

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo Henrique Garcia Silva

**Aplicação Backend para Consulta de Dados
de Geoposicionamento de Ônibus e
Previsão de Chegada**

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo Henrique Garcia Silva

**Aplicação Backend para Consulta de Dados de
Geoposicionamento de Ônibus e Previsão de
Chegada**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Flávio de Oliveira Silva, Ph.D

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2017

Gustavo Henrique Garcia Silva

Aplicação Backend para Consulta de Dados de Geoposicionamento de Ônibus e Previsão de Chegada

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 28 de julho de 2017:

Prof. Flávio de Oliveira Silva, Ph.D
Orientador

Prof. Pedro Frosi Rosa, Ph.D

**Prof. Ronaldo Castro de Oliveira,
Ph.D**

Uberlândia, Brasil
2017

Este trabalho é dedicado primeiramente à Deus por ter me dado todas as oportunidades de ter chegado até aqui. Em segundo lugar agradeço aos meus pais Lázara Helena Cândida Garcia e Glaycon Rosa Silva e irmã Anaby Garcia Rosa pelo apoio de sempre, e por todo o sacrifício que fiseram para que eu tivesse condições de lutar pelos meus sonhos. À toda a minha família, tios, padrinhos em especial Roselena, Noel, Romerson, Valéria por sempre me apoiarem em toda a minha jornada. Aos primos mais próximos Gabriela, Leon e Semoni que me incentivaram a continuar entendendo mais claramente todos os desafios que me eram apresentados. Por fim aos meus amigos e namorada por terem compartilhado comigo momentos únicos durante todo este período, que certamente nunca me esquecerei. Dedico também ao corpo docente da Faculdade de Computação da Universidade Federal de Ubêrlandia por serem meus mestres durante um dos períodos mais transformadores da minha vida, de muito trabalho e muito aprendizado.

“Eu acredito muito em sorte. E tenho constatado que, quanto mais duro eu trabalho, mais sorte eu pareço ter.”

- Thomas Jefferson

Resumo

O tema mobilidade urbana tem ganhado cada vez mais espaço dentre às questões que mais influenciam no modo de viver de cada indivíduo em todo o globo. Com isso, frequentemente é alvo de pesquisas e enfoque de tecnologias que tentam agregar valor ou resolver os problemas que este objeto de estudo traz consigo. Uma das principais abordagens tem foco em aprimorar a performance do fluxo no trânsito das cidades, através de meios alternativos de transporte, uso de tecnologia na gestão do tráfego e até mesmo nos serviços de transporte ofertados. Este trabalho apresenta o estudo, análise e construção de uma aplicação que coleta, centraliza e disponibiliza dados para consumo de outros sistemas, relacionados ao transporte público e coletivo da cidade de Uberlândia, estendendo para o estudo e aplicação de algoritmos de predição de tempo de chegada de ônibus, informação esta que pode ser trabalhada para amenizar certos transtornos relacionados ao contexto da mobilidade em centros urbanos.

Palavras-chave: mobilidade urbana, buzapp, time prediction, estimated time, tempo de chegada, ônibus .

Lista de ilustrações

Figura 1	– Telas do aplicativo CittaMobi na versão para dispositivos Android	19
Figura 2	– Telas do aplicativo Moovit na versão para dispositivos Android . .	20
Figura 3	– <i>Representação BuzApp Backend</i>	24
Figura 4	– <i>Representação BuzApp Predictor</i>	29
Figura 5	– <i>Representação da plataforma BuzApp</i>	30
Figura 6	– <i>Tempo estimado de chegada comparado ao tempo real.</i>	32
Figura 7	– <i>Erro do tempo estimado de chegada comparado ao tempo real.</i> . .	34

Lista de tabelas

Tabela 1 – Análise comparativa dos algoritmos baseado no estudo bibliográfico e contexto da cidade de Uberlândia.	18
---	----

Lista de abreviaturas e siglas

MaaS	<i>Mobility as a Service</i>
ITS	<i>Intelligent Transport Systems</i>
REST	<i>Representational State Transfer</i>
RNA	Rede Neural Artificial
GPS	<i>Global Positioning System</i>
JS	JavaScript
API	<i>Aplication Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
NPM	<i>Node Package Manager</i>

Sumário

1	INTRODUÇÃO	11
1.1	Objetivo	12
1.1.1	Objetivo Específico	12
1.2	Método	12
1.3	Organização do Trabalho	13
2	REVISÃO BIBLIOGRÁFICA	14
2.1	Aplicações Servidoras ou Backend	14
2.2	Predição de tempo de chegada de ônibus	14
2.2.1	Problema	14
2.2.2	Algoritmos de Predição	15
2.2.2.1	Predição baseada em dados históricos	15
2.2.2.2	Predição baseada em modelos matemáticos	16
2.2.2.3	Predição baseada em Redes neurais artificiais	17
2.3	Comparação e escolha do Algoritmo	17
2.3.1	Conclusão da análise	18
2.4	Aplicações de mercado Correlatas	19
2.4.1	CittaMobi	19
2.4.2	Moovit	20
3	DESENVOLVIMENTO DA APLICAÇÃO SERVIDORA	21
3.1	Tecnologias Utilizadas	21
3.1.1	Framework JavaScript	21
3.1.1.1	Visão Geral	21
3.1.1.2	Justificativa e Uso	22
3.1.2	Node.js	22
3.1.2.1	Visão Geral	22
3.1.2.2	Justificativa e Uso	22
3.1.3	MongoDB	22
3.1.3.1	Visão Geral	22
3.1.3.2	Justificativa e Uso	23
3.1.4	API e REST	23
3.1.4.1	Visão Geral	23
3.1.4.2	Justificativa e Uso	23
3.2	Desenvolvimento da Aplicação Servidora	23
3.2.1	Arquitetura (Diagrama De Componentes)	24
3.3	Desenvolvimento do Algoritmo de Previsão de Tempo de Chegada	27
3.3.1	Arquitetura (Diagrama De Componentes)	29
3.4	Visão Geral da Arquitetura	30
3.4.1	Componentes	30

4	AVALIAÇÃO	32
4.1	Cenário de Testes	32
4.2	Limitações	33
4.3	Resultados da Análise	34
	Conclusão	35
	REFERÊNCIAS	36

1 Introdução

É cada vez maior, o número de pessoas que se mudam para zonas urbanas. De acordo com ([GOODALL et al., 2017](#)) no ano de 2014, 54% da população global já vivia em centros urbanos, e esse número é previsto para aumentar com o passar dos anos, onde estima-se que 66% dos habitantes estarão vivendo em zonas urbanas. Esse movimento tem como implicação o aumento dos congestionamentos nos grandes centros urbanos.

Por esses e outros motivos, agências governamentais e instituições privadas responsáveis por garantir as boas condições de trânsito e a mobilidade nos centros urbanos, tem procurado progressivamente por soluções que atendam essa problemática. Seja na busca por novas abordagens no modelo do serviço, na atribuição de tecnologia à esse universo ou um arranjo dos dois.

Junto com o crescimento da necessidade de garantir um bom serviço de mobilidade urbana e sua infraestrutura, novos conceitos começaram a surgir e ganham popularidade neste universo de estudo. Alguns destes conceitos serviram como motivação para este trabalho.

Algumas dessas abstrações como o [MaaS](#) (Mobility as a Service) que é a integração de serviços de transporte de fontes públicas e privadas integradas em uma única plataforma que cria e gerencia a jornada de um usuário do seu ponto de partida até o seu destino ([MARSH; PADAMSEY, 2017](#)), o [ITS](#) (Intelligent Transport Systems) que prega por usar da tecnologia através de aplicações ou plataformas que aprimoram a qualidade do transporte ou traga novas soluções baseado no monitoramento e gerenciamento dos sistemas de tráfego ([TECHOPEDIA, 2017](#)), [Smart Cities](#) que define que pessoas interagindo com os serviços, energia e materiais de uma cidade devem ser catalizados para empoderar seu crescimento econômico e melhoria da qualidade de vida ([PROJETOS, 2017](#)) e outros buscam resolver alguns problemas que são relacionados à mobilidade urbana, como a questão dos congestionamentos em grandes centros, a segurança e a eficiência no transporte e até mesmo a preservação do meio ambiente ([FIGUEIREDO et al., 2001](#)). Um fator em comum dentre estes tópicos tem um papel fundamental para a realização deste estudo, que é o aprimoramento constante da tecnologia e a sua utilização. Nos dias atuais, a velocidade na comunicação e troca de informação, além de outros fatores de extrema importância como a quantidade crescente do número de indivíduos conectados à tecnologia, por meio de dispositivos móveis como smartphones, tablets e wearables, possibilitam uma melhor disseminação destes novos serviços em todo o mundo.

Acompanhando todo este cenário, foi desenvolvida uma plataforma, aqui denominada Buzapp, composta por duas aplicações principais, que, coletam informações em tempo real de geoposicionamento de ônibus, utilizando algoritmos de previsão para calcular estimativas de chegadas de ônibus e disponibiliza para uma aplicação consumidora através de um serviço REST. Estas aplicações tem o obje-

tivo comum de oferecer uma experiência aproximada ao que é observado neste novo contexto apresentado, relacionado à mobilidade urbana e tecnologias empregadas, tendo como base de teste e validação a cidade de Uberlândia e seu sistema de transporte público.

1.1 Objetivo

O objetivo deste trabalho é construir uma aplicação que permita a disponibilização de informações relacionadas ao transporte público local, de maneira a instruir o usuário do sistema, apresentando dados em tempo real das posições do ônibus que percorrem uma rota fornecendo uma previsão de chegada em cada ponto desta rota.

1.1.1 Objetivo Específico

No intuito de agregar valor às informações disponibilizadas ao usuário, foi feito um estudo sobre algoritmos de predição de tempo de chegada de ônibus. Um desses algoritmos foi adaptado para o cenário de Uberlândia e implementado para verificar, posteriormente, se a taxa de erro relacionada à sua previsão pode ser considerado relevante quando comparado ao tempo de chegada de ônibus em estações ao longo de sua rota real.

Para processamento e disponibilização destes dados, as coletas de informações chave são feitas a partir de uma plataforma de uma empresa privada que é responsável pelas medições em tempo real de toda a frota do transporte coletivo de Uberlândia.

1.2 Método

Com o objetivo de entender mais sobre o cenário dos algoritmos de previsão de tempo estimado de chegada, foi feito um levantamento bibliográfico para entender como funcionam, se mostram um grau aceitável de precisão e se são aplicáveis ao contexto de Uberlândia, isto é, se a partir dos dados coletados no cenário local é possível implementar e validar o algoritmo.

Juntamente ao estudo, foi feita uma análise comparativa dos algoritmos encontrados, verificando-se através das comparações de resultados oferecidos na própria bibliografia, quais foram os algoritmos que desempenharam melhor resultado, isto é, aqueles que apresentaram uma estimativa de tempo de chegada mais próxima ao contexto real.

Posteriormente a análise, o algoritmo escolhido foi implementado e adaptado para atender ao cenário da cidade de Uberlândia, recebendo os dados referentes ao seu sistema de transporte público, processando assim suas informações e disponibilizando-as em uma aplicação servidora.

1.3 Organização do Trabalho

Este trabalho é constituído de cinco capítulos. No primeiro capítulo é apresentado a introdução, contextualizando o cenário de mobilidade urbana atual, expondo uma parte da problemática abordada, além de evidenciar os fatores de impulso que levaram à construção deste trabalho, bem como seus objetivos e métodos.

No segundo capítulo, são apresentados de forma mais profunda alguns conceitos necessários para melhor entendimento do trabalho, conceitos relacionados tanto ao contexto de aplicações provedoras de serviços quanto aos algoritmos de predição de tempo de chegada, posteriormente citando algumas tecnologias que já atuam neste ambiente em mercado.

No capítulo terceiro, são apresentadas, de forma abrangente, algumas tecnologias que foram utilizadas na concepção da aplicação, bem como seus resultados.

No quarto capítulo, é feita uma avaliação sobre os testes empregados na aplicação demonstrando também quais foram os resultados encontrados.

No quinto e último capítulo, são apresentadas as conclusões e quais são os trabalhos que podem ser exercidos a partir deste projeto.

2 Revisão Bibliográfica

Este capítulo apresenta um estudo mais detalhado sobre os conceitos necessários para o melhor entendimento deste trabalho, como conceitos de aplicações servidoras ou backend. É exposto ademais, um refinamento das definições relacionadas à problemática envolvendo algoritmos de predição de chegada de ônibus, posteriormente abordando quais são algumas das soluções encontradas hoje. Por fim, são apresentadas ferramentas que estão presentes no mercado, usando desta e demais tecnologias para agregar valor aos serviços relacionados a mobilidade urbana.

2.1 Aplicações Servidoras ou Backend

Uma aplicação servidora ou aplicação de backend em computação é o sistema responsável por armazenar e/ou processar e fornecer dados para que outras aplicações ou sistemas possam consumir de diversas formas. Uma aplicação servidora, geralmente é aquela em que os dados de todo o sistema são armazenados através de bancos de dados e trabalhados por outros processos computacionais, para que possam gerar novas informações posteriormente ([BACKENDLESS, 2017](#)).

É também comum encontrar toda a estrutura de regras de negócio de um sistema sob a responsabilidade de uma aplicação backend, uma vez que esta é a que disponibiliza de mais recursos computacionais quando a comparamos com aplicações cliente, como páginas web sendo requisitadas de notebooks de uso pessoal, aplicativos móveis rodando em smartphones comuns e outros ([PLURALSIGHT, 2015](#)).

Em um contexto geral, o papel fundamental das aplicações de backend é a disponibilização de informação remota, através de protocolos de comunicação, utilizando a internet, bluetooth e outros padrões de transporte de informação.

2.2 Predição de tempo de chegada de ônibus

Nesta seção são apresentadas as abstrações relacionadas primeiramente ao contexto e uma visão geral da problemática envolvendo a Previsão de chegada de ônibus, para posterior apresentação dos algoritmos abordados que tentam atender a esse quesito.

2.2.1 Problema

A previsão do tempo de chegada de ônibus nas condições do ambiente do tráfego urbano, é uma informação altamente probabilística e irregular, o que traz grande dificuldade para tentar chegar em um número que seja preciso, dado às flutuações à qual o tráfego é exposto, com diferentes condições de circulação, em

diferentes horários do dia, em diferentes dias da semana, causados por lentidão em cruzamentos, acidentes, condições do tempo e outros fatores (GONG; LIU; ZHANG, 2013).

Como é discutido em (SUN et al., 2014), fornecer previsões precisas de tempo de chegada de ônibus nos seus pontos de parada no percorrer de sua rota tem um valor muito relevante tanto para usuários deste tipo de transporte quanto para as operadoras de trânsito responsáveis. Em uma análise de curto prazo, com este tipo de dado, entidades de trânsito podem responder de maneira quase que imediata às situações de interrupção ou atraso no serviço inesperadas, concebendo estratégias para melhor controle e performance da sua operação.

Analizando os benefícios a longo prazo, os órgãos administradores de tráfego podem identificar mais facilmente rotas com problemas complexos e mudanças que historicamente levariam um maior tempo e dedicação de pessoal técnico e administrativo para serem apontados, como por exemplo ajuste de horários e análise da sinalização do trânsito no intuito de melhorar o fluxo e a performance (SUN et al., 2014).

A informação de previsão de tempo de chegada, quando difundida para os usuários do transporte, através de aplicativos, internet, quadros eletrônicos em pontos de parada, tvs, podem reduzir a ansiedade de espera dos passageiros quando esperam pela chegada da sua condução, além de poupar seu tempo de viagem, uma vez que estes podem planejar melhor quando devem chegar no ponto de ônibus (SCHWEIGER; L, 2003) e (SUN; FU, 2004). Consequentemente, essa transparência na informação, ajuda a reter os viajantes neste meio de transporte e atrai novos usuários que não estão habituados com o uso deste meio locomotivo.

De modo geral, ainda de acordo com (SUN et al., 2014), a qualidade do próprio tráfego urbano tende a melhorar, uma vez que o aprimoramento na qualidade do serviço de transporte público faz com que a sua utilidade seja mais atrativa e amigável ao usuário, promovendo o abandono do uso de outros meios de transporte como veículos pessoais.

2.2.2 Algoritmos de Predição

No passado, um número variado de estudos foram introduzidos para tentar endereçar os distúrbios da predição de tempo de chegada de ônibus (SUN et al., 2014). Em conclusão destes esforços, três tipos de modelos de predição foram determinados, o primeiro deles é conhecido como um modelo baseado em dados históricos, o segundo tem seu fundamento em métodos de modelos de regressão e o último deles baseia-se em padrões de redes neurais.

2.2.2.1 Predição baseada em dados históricos

O primeiro tipo de modelo de predição, infere o tempo estimado de trajeto de um ônibus, presente e futuro, baseados no histórico do tempo de viagem deste mesmo ônibus ou de outros ônibus. Uma variedade de algoritmos de predição de tempo de chegada de ônibus foram propostos por Lin e Zeng, para oferecer para os

passageiros e usuários do sistema de trânsito da cidade de Blacksburg, na Virgínia, Estados Unidos da América (LIN; ZENG, 1999). Foram propostos quatro algoritmos diferentes, todos eles com um modelo de entrada de dados distintos entre si. Esses algoritmos se mostraram melhores na predição do que os algoritmos já conhecidos pela literatura da época, no entanto, não foi considerado nestas soluções, as condições arbitrárias de trânsito, como exemplo os engarrafamentos e o tempo em que um ônibus fica inerte em um ponto de parada, esperando para que seus passageiros se juntem a viagem ou saiam dela.

Outro algoritmo de previsão de chegada de ônibus proposto foi o apresentado por Kidwell. O algoritmo, que é baseado na localização em tempo real do veículo, funciona da seguinte forma: cada rota em que um ônibus percorre é dividida em zonas menores, a cada zona, é atribuído um tempo em que o ônibus leva para percorrer esse trecho. As predições eram fundamentadas na coleta de dados mais recente relacionada ao trecho em que o ônibus percorreu, o que fez com que esse algoritmo não conseguisse atender às expectativas de cenários de grandes centros urbanos, onde tanto o tempo de viagem de um ônibus quanto o seu tempo de espera em pontos de parada, tendem a variar bastante (KIDWELL, 2001).

De modo geral, os modelos atribuídos ao primeiro tipo de predição, baseada em dados históricos, desempenham boas estimativas somente em condições onde o padrão do tráfego é relativamente estável, um cenário normalmente encontrado em cidades de médio e pequeno porte. Um dos maiores entraves dessas abordagens é que ela exige um conjunto de dados históricos consideravelmente amplo para poder prover dados mais apurados, premissa essa que pode não ser atendida principalmente quando os padrões de tráfego urbano variam significativamente ao longo do tempo (SUN et al., 2014).

2.2.2.2 Predição baseada em modelos matemáticos

Uma segunda alternativa apresentada para a resolução dos problemas envolvendo estimativas de tempo de chegada de ônibus é a aplicação de modelos matemáticos para encontrar os tempos estimados de viagem dos ônibus entre os pontos de paradas de sua rota para posteriormente chegar a um tempo de chegada esperado em cada um desses pontos. Frequentemente, estes modelos são estabelecidos pela regressão dos tempos de viagem em um conjunto de variáveis independentes, como as condições do trânsito, motorista, número de pontos de parada na rota e condições climáticas. Um conjunto de modelos de regressão foi desenvolvido por (PATNAIK; CHIEN; BLADIKAS, 2004) para estimar o tempo de chegada de um ônibus com dados coletados automaticamente por contadores de passageiros instalados nas conduções. Essa abordagem se mostrou válida somente quando essas variáveis independentes bem como suas equações podem ser calculadas e estabelecidas com um certo grau de garantia, o que pode não ser uma opção para todos os cenários encontrados onde muitas dessas variáveis estão relacionadas. Salvo as colocações feitas como premissa, os resultados obtidos pelos experimentos são promissantes e indicam que o modelo desenvolvido pode ser usado para simular um diverso número de condições externas ao trânsito.

2.2.2.3 Predição baseada em Redes neurais artificiais

O terceiro e último tipo de modelo de predição, é uma abordagem aplicada à redes neurais artificiais(RNAs), que são capazes de capturar correspondências complexas não lineares entre os agentes no sistema. As RNAs quando usadas, requerem um longo treinamento e teste da rede no intuito de obter a estrutura certa e encontrar quais são os parâmetros de entrada que demonstram melhores resultados (SUN et al., 2014).

Uma RNA simula o processo de aprendizado de um cérebro humano, elas apresentam um bom resultado, quando delegamos à elas tarefas relacionadas à reconhecimentos de padrões, predições, classificações e outros. De forma abrangente, RNAs são calibradas em dois passos, treinando e testando. No período de treinamento, a RNA utiliza-se de princípios de aprendizado indutivo para capacitar-se usando um conjunto de dados de entrada no sistema (JEONG; RILETT, 2004).

Existem dois tipos de técnicas de aprendizado usadas na construção de RNAs, a primeira técnica é conhecida como aprendizagem não supervisionada, onde a rede tenta separar o conjunto de dados de entrada de treinamento em diferentes grupos, baseados nos padrões de entrada. No segundo método, o supervisionado, a resposta desejada do sistema, encontrada na camada de neurônios de saída do sistema é conhecida, e a rede ajusta o peso das ligações entre seus neurônios, camada a camada para reproduzir o resultado esperado. Durante este processo, o erro de cada saída do sistema, é propagado de forma recursiva entre suas camadas, através desse ajuste entre os pesos das conexões. Este processo é conhecido como *back-propagation*, e é um dos métodos mais usados em aplicações voltadas ao transporte. O processo de aprendizado das RNAs pode ser contínuo, fazendo com que os modelos se adequem às mudanças características do ambiente exposto (JEONG; RILETT, 2004).

Dado esse contexto, os modelos RNA podem ser vistos também como predições dinâmicas, devido ao fato de eles poderem ser atualizados ou modificados usando novos dados de entrada no sistema online (CHIEN; DING; WEI, 2002).

2.3 Comparação e escolha do Algoritmo

Feito um estudo mais detalhado da bibliografia, e seus conceitos apresentados na seção anterior, foi realizada uma análise comparativa entre os modelos de algoritmos propostos para auxiliar na escolha do método a ser implantado neste trabalho.

Os critérios de decisão estabelecidos são específicos do contexto deste trabalho, e não necessariamente representam as melhores escolhas para uma aplicação em um cenário real. Os preceitos que foram analisados para o desenvolvimento deste trabalho foram:

1. Complexidade de implementação;
2. Bibliografia de fácil acesso;

3. Tempo de implementação;
4. Tempo para treinamento do algoritmo;
5. Adaptável ao contexto de Uberlândia;
6. Aplicável às condições de trânsito de uma cidade de médio porte;
7. Apresenta uma baixa taxa de erro;
8. Não Requer extensa quantidade de dados históricos.

Finalizada essa relação dos preceitos a serem analisados de cada conceito de algoritmo para predição de tempo de chegada propostos, pontuamos a tabela abaixo, **1** para *atende* e **0** para *não atende*.

	Dados Históricos	Modelos Matemáticos	RNAs
1 - Baixa Complexidade	1	1	0
2 - Bibliografia	1	1	1
3 - Tempo de implementação	1	1	0
4 - Tempo para treinamento	1	1	0
5 - Adaptável Uberlândia	1	1	1
6 - Aplicável	1	1	1
7 - Baixa taxa de erro	0	1	1
8 - Não Requer dados históricos	0	1	0

Tabela 1 – Análise comparativa dos algoritmos baseado no estudo bibliográfico e contexto da cidade de Uberlândia.

2.3.1 Conclusão da análise

Dados os itens listados que serviram para análise de aplicabilidade e desenvolvimento dos modelos de algoritmos de predição estudados, para a construção deste trabalho o modelo que melhor se encaixa nas premissas é um modelo baseado em modelos matemáticos, ou seja, apresentam baixa complexidade de implementação, tem uma bibliografia de fácil acesso, não demanda grande tempo para ser implementado, não requer período para treinamento extenso, é adaptável à cidade de Uberlândia e seu contexto, entrega resultados com uma taxa de erro considerada aceitável (SUN et al., 2014) e não requer dados históricos.

2.4 Aplicações de mercado Correlatas

Nesta sessão são apresentadas dois produtos que são reconhecidas em cenário nacional e mundial, se utilizando, dentre outros, dos conceitos e motivações apresentadas neste trabalho, entregando um serviço de valor agregado para usuários do sistema de transporte público e coletivo de centros urbanos.

2.4.1 CittaMobi

O **CittaMobi** é um produto da empresa **Cittati**, empresa integradora de soluções voltada para o desenvolvimento de tecnologia de informação e comunicação com foco na gestão de sistemas de transporte, que através do CittaMobi entrega à usuários de transporte público uma plataforma acessível para usuários de transporte público coletivo através de um aplicativo informando mapeamento, cadastros, monitoramento, previsões e informações de alterações sobre ônibus, rotas e linhas percorridas, tudo isso através da coleta de dados de um dispositivo instalado em cada veículo da plataforma. Além de dar a possibilidade para o usuário contribuir com a plataforma gerando suas próprias informações sobre informações de linha e ônibus e pessoais, como localização e nome (**CITTAMOBIL**, 2017).

Abaixo são apresentadas algumas telas da versão Android do aplicativo CittaMobi.

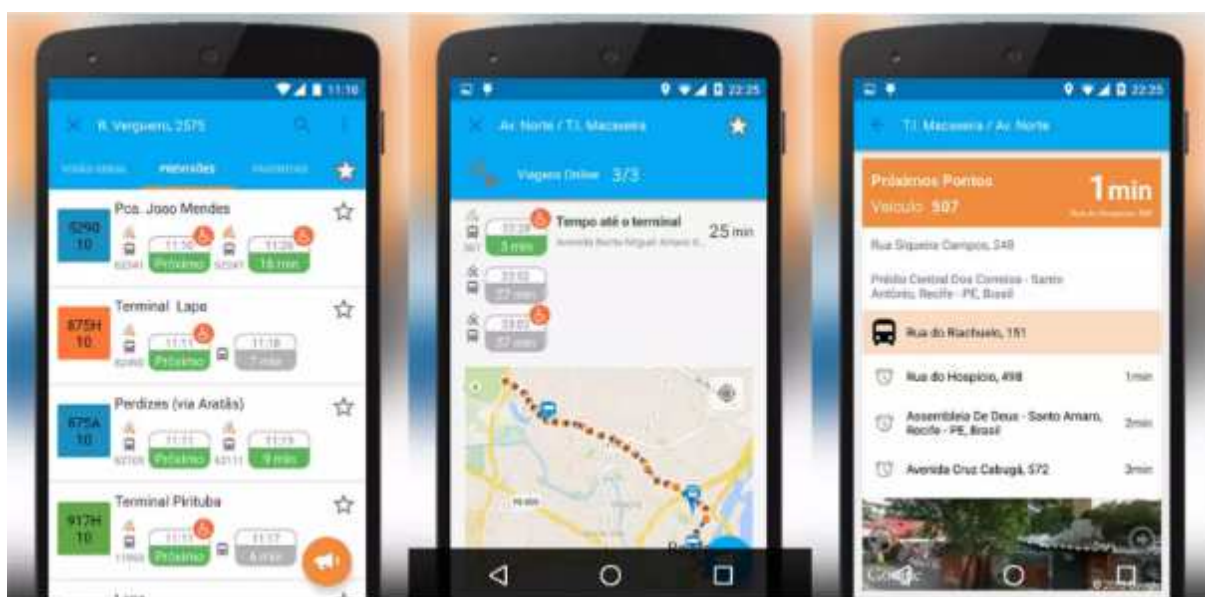


Figura 1 – Telas do aplicativo CittaMobi na versão para dispositivos Android
Fonte: [PlayStore](#)

A primeira tela exibida na figura acima, refere-se à tela que exibe as previsões de chegada de ônibus em referidos pontos de parada de ônibus ao longo de suas respectivas rotas. A segunda tela, apresenta um acompanhamento em tempo real de um veículo específico no percorrer de seu trajeto. Finalmente, o terceiro painel exibe um acompanhamento do trajeto percorrido do próprio usuário.

2.4.2 Moovit

O Moovit é um aplicativo de transporte público e mapeamento desenvolvido pela empresa [Moovit Inc.](#), com foco em planejamento de viagens, chegadas em tempo real, previsões de partidas, linhas rodoviárias atualizadas, mapas de estações locais e serviços de alerta voltados para o transporte público coletivo. O Moovit tem versões para Android e iOS, além de uma versão web e é o aplicativo de trânsito mais usado no mundo com mais de 60 milhões de usuários, presente em mais de 1.200 cidades em 77 países e 43 línguas. Como um diferencial de outras plataformas voltadas para acompanhamento de tráfego urbano, o Moovit utiliza a técnica de *crowdsourcing* para coletar dados de seus usuários para que eles mesmos alimentem a plataforma, além de coletar dados das próprias indústrias de transporte locais ([MOOVIT, 2017](#)).

Abaixo demonstradas algumas das telas do aplicativo na versão Android do Moovit.

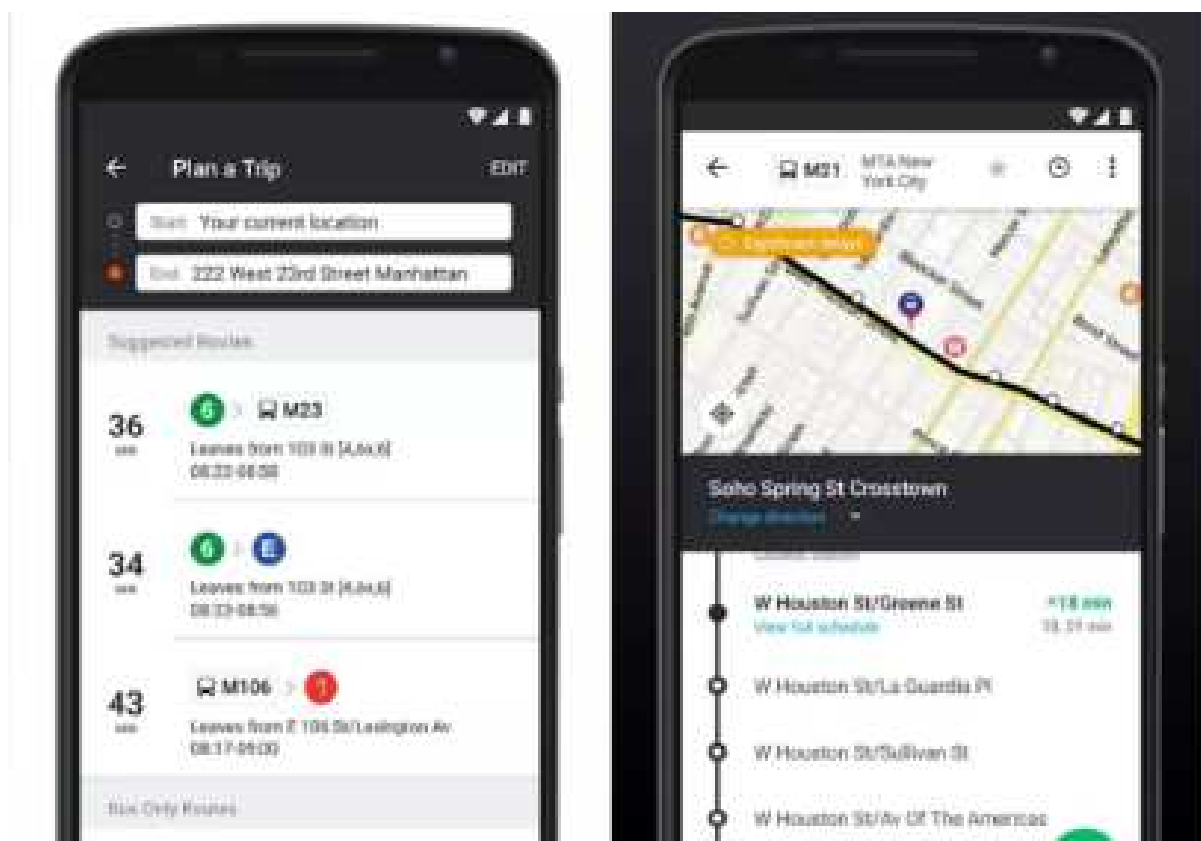


Figura 2 – Telas do aplicativo Moovit na versão para dispositivos Android
Fonte: [PlayStore](#)

Na figura apresentada acima a primeira tela demonstra as estimativas de tempo de chegada de ônibus no planejamento de viagem de um usuário, seguida por uma tela onde são mostrados todo o acompanhamento do percurso da viagem.

3 Desenvolvimento da Aplicação Servidora

Neste capítulo são apresentadas as metodologias e conceitos aplicados ao desenvolvimento da aplicação obra fruto deste trabalho. Nele são evidenciados infraestrutura, arquitetura, tecnologias, metodologias e padrões de desenvolvimento, algoritmos propostos e uma visão geral da composição

Para alcançar os objetivos do trabalho anteriormente apresentados, foi construída uma aplicação servidora, que em sua estrutura demonstra o que foi proposto neste trabalho, que é a construção de uma aplicação que consiga prover informações referentes ao transporte público da cidade de Uberlândia, e a partir dos dados da frota através de GPS em tempo real, enriquecer essa informação com previsões de chegada de ônibus nos seus respectivos pontos de parada.

3.1 Tecnologias Utilizadas

Nesta seção, são apresentadas as tecnologias e ferramentas que foram utilizadas para a construção da aplicação objetivo deste trabalho, bem como definições abrangentes relacionadas às mesmas, adicionada de uma breve justificativa da sua escolha, apresentando seu papel no contexto deste projeto.

3.1.1 Framework JavaScript

3.1.1.1 Visão Geral

O JavaScript, comumente abreviado para JS é uma linguagem de programação, caracterizada por ser leve, interpretada e fundamentada em objetos. É mundialmente conhecida por ser uma linguagem de script para páginas Web e vem ganhando força no uso em ambientes sem browsers, como o [Node.js](#) para aplicações de serviços ou como o [Apache CouchDB](#) para banco de dados. O JS, além de ser uma linguagem de script que oferece suporte a diferentes paradigmas de programação(orientada à objetos, funcional, imperativa), baseia-se na dinâmica de protótipos, que permite o uso de herança entre objetos ([MOZILLA, 2017](#)).

O JS é uma das linguagens de programação mais conhecidas ao redor do globo, se não for a mais usada no desenvolvimento do frontend de aplicações Web, o que contribuiu para o massivo número de frameworks criados para a linguagem JavaScript. Um framework JS, descreve como uma aplicação deve ser construída, e faz com que o código seja reutilizável e mais organizado, o que implica em mais flexibilidade e escalabilidade desta aplicação. Os frameworks podem ser utilizados tanto para a construção de aplicações para ambientes em browsers (React, AngularJS, BackboneJS) tanto para ambientes de serviços como o próprio [Node.js](#) ([MARIANO, 2017](#)).

3.1.1.2 Justificativa e Uso

A linguagem JavaScript foi escolhida para este projeto pelo fato de ser uma linguagem leve, a sua sintaxe pode ser replicada tanto para construir aplicações provedoras de serviços quanto para construção de bancos de dados. De fato, neste trabalho todo o backend foi construído utilizando o Node.js e o banco de dados aplicado para o armazenamento de informação foi o MongoDB, ambos escritos com a sintaxe JS.

3.1.2 Node.js

3.1.2.1 Visão Geral

O Node.js, também conhecido como Node, é um ambiente JavaScript *server-side* (backend), baseada na engine da empresa Google, conhecida como V8 (TILKOV; VINOSKI, 2010). A V8, é um mecanismo open source de alta performance JS, escrito em linguagem de programação C++, usado no Chromium, Node.js e outros embarcados. A V8, compila e executa códigos JavaScript, lida com as alocações de memória para objetos e faz a remoção dos objetos que não serão mais usados (GOOGLE, 2017).

O Node, se utiliza de um mecanismo com uma lógica de entrada/saída dirigida a eventos não bloqueante, para dar suporte a execução de múltiplas operações e tarefas em seu ambiente, o que a torna muito leve e eficiente. O universo de bibliotecas disponíveis para aplicações Node é o [npm](#), que é o maior ecossistema de bibliotecas *open source* do globo (NODEJS, 2017).

3.1.2.2 Justificativa e Uso

A justificativa de se usar o Node para o desenvolvimento deste projeto deve-se ao fato de se utilizar da linguagem JS, à sua alta performance na execução de aplicações, e a sua vasta lista de bibliotecas open source que, trazem implementações prontas de tarefas que foram fundamentais para este trabalho, como a criação de schedulers e comunicação com banco de dados mongo, com uma extrema facilidade na integração e utilização das mesmas.

3.1.3 MongoDB

3.1.3.1 Visão Geral

De acordo com (MONGODB, 2016), o MongoDB ou Mongo, é um banco de dados gratuito, open source e NoSQL, isto é, os dados não estão estruturados necessariamente para serem armazenados, que oferece escalabilidade e flexibilidade na busca e indexação de informação, fundamentado num modelo de dado descrito por documentos. O MongoDB salva a informação em documentos no formato [JSON\(JavaScript Object Notation\)](#), isto significa que as propriedades e estruturas dos documentos armazenados podem variar entre si. Os documentos armazenados no Mongo, são mapeados em objetos JavaScript, tornando o seu processamento em aplicações mais fácil.

3.1.3.2 Justificativa e Uso

O MongoDB foi de grande valia para este projeto por duas razões principais. Primeiramente, ele sendo um banco baseado na sintaxe JS, a tradução de informação de documentos é facilmente realizada para objetos JavaScript através de bibliotecas para Node.js como o [mongoosejs](#). A segunda e principal motivação deve-se ao fato de que da concepção do trabalho, foi entendido que ao coletar as informações de GPS em tempo real dos ônibus selecionados na cidade de Uberlândia, uma grande quantidade de dados seria gerada, pois essas informações são coletadas em intervalos de tempos muito pequenos e sempre variam. Para este cenário, um banco de dados não relacional é o mais adequado em grande parte dos casos, como é visto em ([LI; MANOHARAN, 2013](#)).

3.1.4 API e REST

3.1.4.1 Visão Geral

Uma API, é entendido por um conjunto de processos e padrões que uma aplicação estabelece e documenta, com o objetivo de que outras aplicações, externas a essa possam consumir de seus serviços, comunicando-se por um protocolo pré-estabelecido, sem necessariamente conhecer os detalhes da implementação desta aplicação. Ou seja, uma API é uma forma de comunicação entre aplicações e usuários ([PIRES, 2015](#)).

Já o REST(*Representational State Transfer*), refere-se a um conjunto de regras, padrões ou princípios que, são claramente definidas na concepção de um projeto, deixando documentadas todas os seus pontos de interface, isto é, relata e estrutura como as aplicações vão se comunicar.

3.1.4.2 Justificativa e Uso

Para melhor entendimento e abstração deste projeto, foi criada uma API para que os dados referentes aos ônibus pudessem ser futuramente consumidos e utilizados por uma aplicação cliente. Este foi estruturado juntamente com um padrão REST, que para este exemplo, utiliza de um método [GET](#), passando como parâmetro o nome da rota a ser consultada, e espera-se uma resposta com os dados no modelo JSON.

3.2 Desenvolvimento da Aplicação Servidora

Para que os objetivos expostos pudessem ser alcançados, foi desenvolvida uma aplicação backend, utilizando dentre outras tecnologias citadas o Node.js, que tem com responsabilidade algumas tarefas. Primeiro, ela é responsável pela coleta de dados de posicionamento de ônibus. A coleta de dados é feita através de uma parceria com duas empresas privadas que atuam no transporte público de Uberlândia, a primeira delas, a [Ubertrans](#), é a empresa responsável por todo o gerenciamento de transporte público coletivo do município, e foi quem nos concedeu a permissão para acessar os referentes dados.

A segunda empresa, a [InLog](#), é a instituição responsável pelo *hardware* instalado em cada veículo da frota de Uberlândia, ou seja, toda a informação que serviu como base para este trabalho estava sob posse da InLog, que fornece uma API para que de tempos em tempos o BuzApp pudesse consultar os dados referentes à frota de ônibus da cidade de Uberlândia em tempo real.

A segunda tarefa de incumbência dessa aplicação, é o armazenamento destes dados. O BuzApp backend armazena todas os logs recebidos em tempo real em um banco de dados MongoDB, para que estes mais tarde possam ser usados para, entre outros, o cálculo de tempo de chegada de ônibus estimado.

A terceira e fundamental missão desta aplicação é prover através de uma API REST, toda essa informação para que esta possa ser consumida por outras aplicações. Para isso os dados que são coletados da InLog, são coletados, processados e padronizados em tempo real e após isso são fornecidos através de um serviço REST.

3.2.1 Arquitetura (Diagrama De Componentes)

Como é representado visualmente na Figura 3 abaixo, o BuzApp Backend executa uma aplicação escrita em JavaScript, que é interpretada pelo Node.js instalado na máquina. Esta aplicação faz a coleta dos dados utilizando um serviço disponibilizado pela empresa InLog aplicando uma chamada HTTP GET, processa e armazena esses dados localmente em um banco MongoDB, e os fornecendo novamente para o consumo de aplicações clientes dado o exemplo de um aplicativo móvel (A) ou até mesmo uma aplicação desktop (B). Para que o BuzApp Backend fosse exposta como uma API, foi aproveitada uma biblioteca conhecida no npm chamada de [Hapi.js](#). Para o recolhimento constante de informações de ônibus aconteça, outro pacote npm foi agregado ao projeto, com o nome de [node-schedule](#), ela auxilia na criação e manutenção de tarefas executadas por agendamento no ambiente Node.

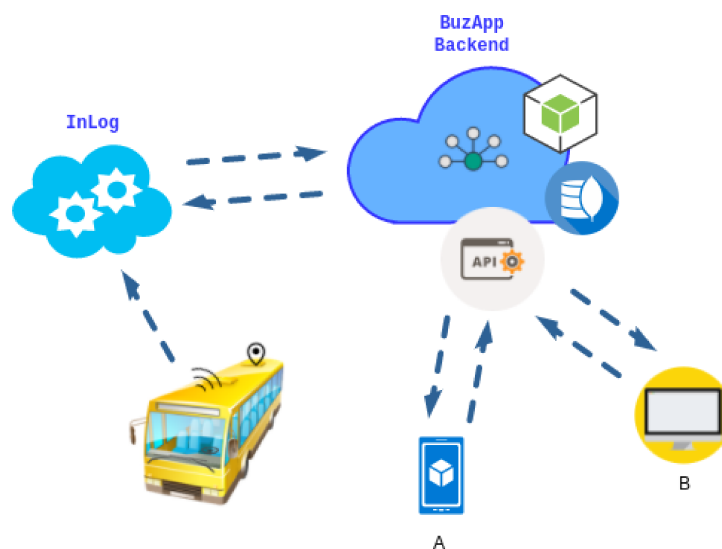


Figura 3 – Representação BuzApp Backend

Abaixo é descrito um exemplo de uma chamada do serviço do BuzApp Backend. No resultado, são informados todos os ônibus que estão ativos, isto é, estão em operação na rota no momento, para este caso a rota usada é a T-131, que na cidade de Uberlândia movimentam passageiros com destino entre os terminais Santa Luzia e Central. Além disso são recuperados todos os pontos de parada desta rota, bem como suas informações gerais e de localização.

Para o caso dos veículos, são exibidas informações como dados de última velocidade e localização recolhidos pelo equipamento da empresa InLog, além de dados providos pelos cálculos do BuzApp Backend e Predictor, como sentido do ônibus (Ida ou Volta) bem como seu tempo estimado de chegada no próximo ponto de parada.

Já nas informações referentes aos pontos de parada, são exibidas as informações de nome da estação, seu georeferenciamento que foi cadastrado de maneira manual, além de algumas rotas que também passam por este ponto de parada e outros.

```
1 {
2   "activeBuses": [
3     {
4       "busPlateId": "07510",
5       "routeId": "T131",
6       "isRunning": true,
7       "lastPosition": {
8         "latitude": -18.938482,
9         "longitude": -48.229862
10      },
11      "lastUpdateTime": "05-06-2017 20:27:13",
12      "velocity": 21,
13      "direction": "Ida",
14      "timeToNextStop": "3.25 minutos",
15      "isActive": true
16    },
17    {
18      "busPlateId": "07504",
19      "routeId": "T131",
20      "isRunning": true,
21      "lastPosition": {
22        "latitude": -18.937353,
23        "longitude": -48.230453
24      },
25      "lastUpdateTime": "05-06-2017 22:37:30",
26      "velocity": 6,
27      "direction": "Volta",
28      "timeToNextStop": "2.08 minutos",
29      "isActive": true
30    },
31  ],
32 }
```

```
31     ...
32 ],
33 "busStops": [
34     {
35         "id": "5939abcc0193b0789e349e59",
36         "busStopId": 0,
37         "stationName": "Terminal Santa Luzia",
38         "isStation": true,
39         "description": "Terminal Santa Luzia",
40         "v": 0,
41         "position": {
42             "latitude": "-18.937165",
43             "longitude": "-48.229884"
44         },
45         "routeIds": [
46             "T131"
47         ]
48     },
49     {
50         "_id": "5939abf730390079008ec040",
51         "busStopId": 1,
52         "stationName": "Terminal Central",
53         "isStation": true,
54         "description": "Terminal Central",
55         "v": 0,
56         "position": {
57             "latitude": "-18.913883",
58             "longitude": "-48.275496"
59         },
60         "routeIds": [
61             "T131"
62         ]
63     },
64     {
65         "_id": "593a148095f715598ff97286",
66         "busStopId": 2,
67         "stationName": "Cabine 13",
68         "isStation": false,
69         "description": "Cabine 13",
70         "v": 0,
71         "position": {
72             "latitude": "-18.933115",
73             "longitude": "-48.235060"
74         }
75     },...
76 ],
```

```
77 "initialStop": {
78   "_id": "5939abcc0193b0789e349e59",
79   "busStopId": 0,
80   "stationName": "Terminal Santa Luzia",
81   "isStation": true,
82   "description": "Terminal Santa Luzia",
83   "v": 0,
84   "position": {
85     "latitude": "-18.937165",
86     "longitude": "-48.229884"
87   },
88   "routeIds": [
89     "T131",
90     "T132",
91     "A339"
92   ]
93 },
94 "finalStop": {
95   "_id": "5939abf730390079008ec040",
96   "busStopId": 1,
97   "stationName": "Terminal Central",
98   "isStation": true,
99   "description": "Terminal Central",
100   "_v": 0,
101   "position": {
102     "latitude": "-18.913883",
103     "longitude": "-48.275496"
104   },
105   "routeIds": [
106     "T131",
107     "T132"
108   ]
109 }
110 }
```

3.3 Desenvolvimento do Algoritmo de Previsão de Tempo de Chegada

Após um estudo bibliográfico sobre os algoritmos de predição e a escolha da solução proposta, a sua lógica foi implementada em uma aplicação separada ao BuzApp Backend, uma vez que subentende-se que estes algoritmos demandam um custo computacional que pode ser alto dependendo das operações a serem realizadas. A esta aplicação daremos o nome de BuzApp Predictor.

Assim como o BuzApp Backend, o BuzApp Predictor roda no ambiente Node,

operando com o uso do Hapi.js e do node-scheduler para expor serviços em sua camada de API e para salvar as estimativas calculadas no banco para serem usadas posteriormente, de maneira periódica, respectivamente.

O BuzApp Predictor, recebe uma chamada do BuzApp Backend para calcular uma estimativa de chegada, de determinado ônibus, em uma determinada rota, em determinada hora do dia, enviando também a última localização (latitude e longitude) conhecida do veículo. A partir dessas informações, o BuzApp Predictor faz os seguintes cálculos para obter uma estimativa:

1. Busca no banco de dados uma rota que seja compatível com a rota atual do ônibus em questão. (O algoritmo só procede se for encontrada a rota, pois é necessário que exista como conhecimento quais são os pontos de parada de cada rota, para determinar quais são segmentos deste trajeto, isto é, os caminhos que interligam todos os pontos de parada de uma dada rota.)
2. Através da localização do ônibus em coordenadas de latitude e longitude, é feito um cálculo, uma distância simples entre 2 pontos, determina-se em qual segmento da rota, isto é, entre quais dois pontos de parada o ônibus se encontra neste registro de localização.

Para este cálculo, suponha que P e Q são os dois pontos de parada cujos quais compõem os limites do segmento da rota pela qual o veículo percorre. Se o veículo está localizado em qualquer ponto M, sabemos que M só será um ponto localizado no segmento PQ se e somente se, dado que $d(x, y)$ equivale a distância de dois pontos, $[(d(p, Q) \leq d(P, M) \text{ e } d(P, Q) \leq d(Q, M)) \text{ ou } [(d(P, Q) = d(P, M) \text{ e } d(P, Q) > d(Q, M))]]$. Assim sendo, ao receber a localização de um dado veículo, o cálculo acima referido é testado para todos os pontos de parada sequenciais, até que a condição acima seja atendida, significando que o ônibus está entre os referidos pontos de parada (WEIGANG et al., 2002).

A seguir, podemos calcular o tempo estimado de chegada T de um ônibus em um determinado ponto de parada n por (SUN et al., 2014):

$$T = \frac{S_{if}}{v_i} + t_{(i+1)} + t_{(i+2)} + \dots + t_{(i+n)} + td$$

onde:

$$v_i = \frac{av_r + bv_{ai}}{a + b}$$

e

v_r = velocidade atual dada pelo GPS

v_{ai} = velocidade média do ônibus no segmento i no horário desejado

$a = S_{if}$ = distância do ônibus até o ponto de parada do final do segmento i

$b = S_{ib}$ = distância do ônibus até o ponto de parada do inicial do segmento i

n = número de segmentos de uma rota

t_{i+n} = tempo de viagem de cada segmento baseado na média individual de cada segmento no período escolhido

td = soma da média de tempo em que um ônibus fica estacionado em suas estações de parada para embarque e desembarque de passageiros

Para este trabalho td foi desconsiderado, pois essa informação é baseada em dados históricos e não é conhecida no cenário da cidade de Uberlândia.

Outro requisito também é necessário para que o algoritmo funcione da maneira desejada, que é a capacidade de determinar se um ônibus está indo ou vindo em uma rota. Para o cálculo deste problema, foi utilizado um método simples, ao receber a atual localização do ônibus em um momento t_1 e em um segundo momento t_2 , dado que seu ponto de parada inicial p_i é conhecido, assim como seu ponto de parada final da rota p_f , à partir da coleta da localização em t_2 , é feito o cálculo de distância entre pontos das geolocalização do veículo em relação à p_f e p_i , desta forma, inferimos que, se o ônibus v no momento t_2 se aproxima de p_f , ele está indo pela rota, caso contrário ele está voltando em direção oposta ao sentido da rota.

3.3.1 Arquitetura (Diagrama De Componentes)

Como discutido acima, é apresentado abaixo, visualmente, como está distribuída a organização do BuzApp Predictor em relação ao BuzApp Backend.

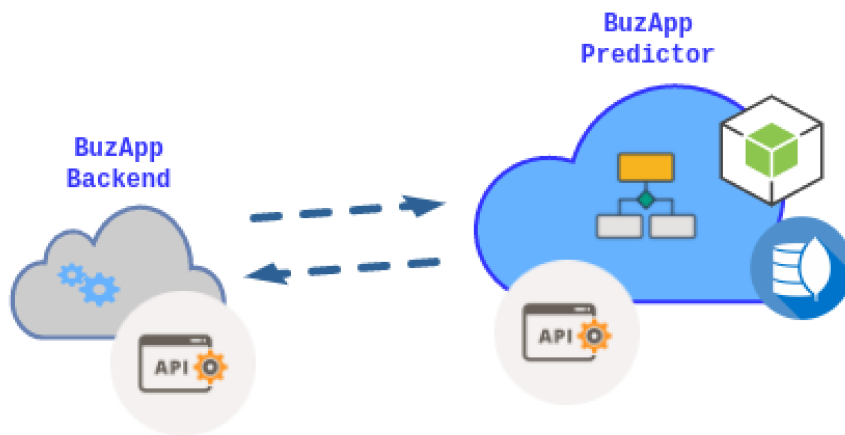


Figura 4 – Representação BuzApp Predictor

Como é representado acima na figura 4, a aplicação BuzApp Backend faz uma requisição para BuzApp Predictor pedindo a informação de tempo estimada de chegada de um determinado ônibus pertencente à uma determinada rota. BuzApp Predictor por sua vez recebe de BuzApp Backend o identificador do veículo, sua rota pertencente, a direção em que o veículo está indo, sua velocidade atual, sua última localização conhecida e a hora em que a mesma foi coletada, logo que BuzApp Predictor termina o cálculo da previsão do tempo estimado de chegada retorna esta previsão para BuzApp Backend, para que este por sua vez passe a disponibilizá-lo como informação complementar em seu serviço.

3.4 Visão Geral da Arquitetura

Nesta seção, são apresentados de forma abstrata todos os componentes que foram utilizados para a construção da plataforma inteira, bem como representados alguns de seus fluxos nas tarefas realizadas para entregar as informações que validem o objetivo do projeto.

3.4.1 Componentes

Abaixo é apresentado um diagrama demonstrando como os componentes da plataforma BuzApp se relacionam, e posteriormente são discutidos os processos neste cenário.

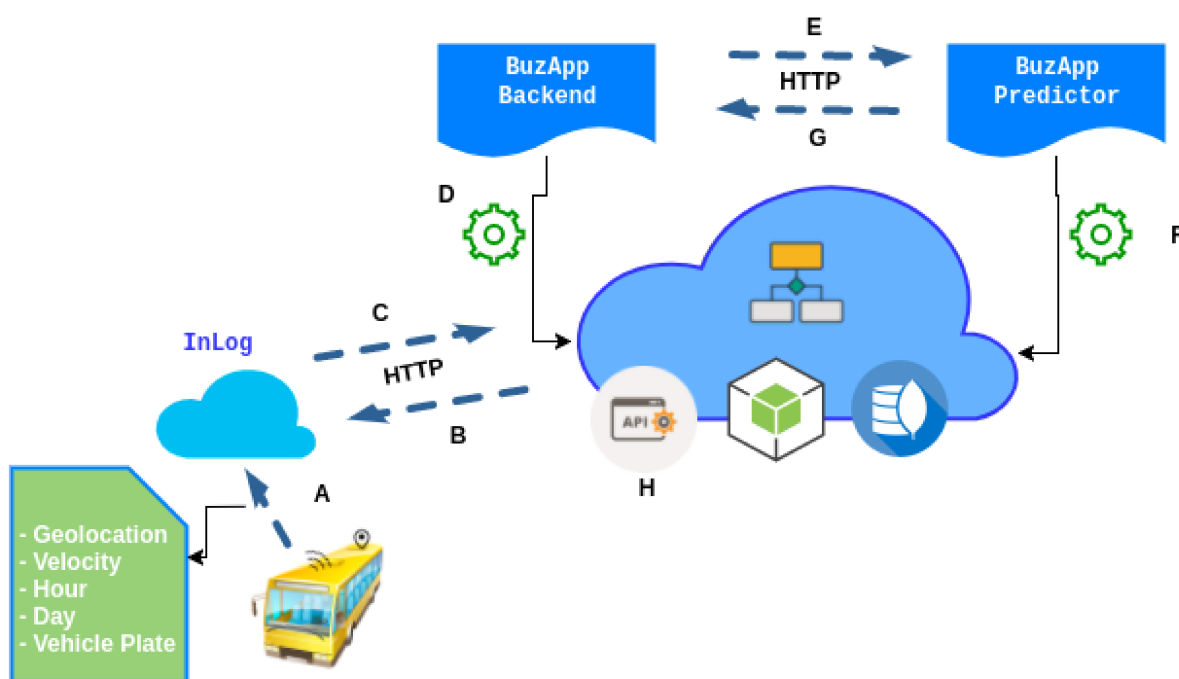


Figura 5 – Representação da plataforma BuzApp

Os processos exibidos acima são respectivamente representados por:

- A - O veículo percorre sua rota e envia seus dados para a plataforma da empresa InLog;
- B - A aplicação BuzApp Backend consome um serviço disponibilizado pela InLog para consultar e os dados coletados;
- C - A InLog retorna os dados para o BuzApp Backend;
- D - O BuzApp Backend processa estes dados e para cada ônibus ativo calcula se este está "Indo" ou "Vindo" em sua rota;

- E - O BuzApp Backend requisita do BuzApp Predictor o tempo estimado de chegada de um ônibus em seus próximos pontos de parada, até o final da rota.
- F - BuzApp Predictor calcula os tempos estimados baseados na localização, velocidade e histórico de cada ônibus e rota individualmente;
- G - BuzApp Predictor retorna os cálculos para BuzApp Backend;
- H - BuzApp Backend disponibiliza todos estes dados em uma API REST.

4 Avaliação

4.1 Cenário de Testes

Para avaliar o algoritmo proposto, foram coletados durante 10 dias consecutivos, dados sobre o comportamento de um ônibus específico, que tinha seu trajeto delimitado pela rota T-131 da cidade de Uberlândia, que transporta passageiros de origem Terminal Santa Luzia à destino Terminal Central, e vice-versa, e pontos desta rota. Os dados usados para o teste foram retirados do banco de dados após a sua coleta, no intervalo dos dias 20/06/2017 até o dia 26/06/2017, em horários igualmente distribuídos de forma aleatória. Cada amostra, representa um *log* coletado dentre o período mencionado, selecionado de forma aleatória para ser usado na comparação.

A figura 6, mostra os tempos calculados pelo algoritmo proposto neste trabalho para o tempo estimado de chegada de um ônibus na sua próxima parada, bem como o tempo real de chegada para cada amostra selecionada aleatoriamente.

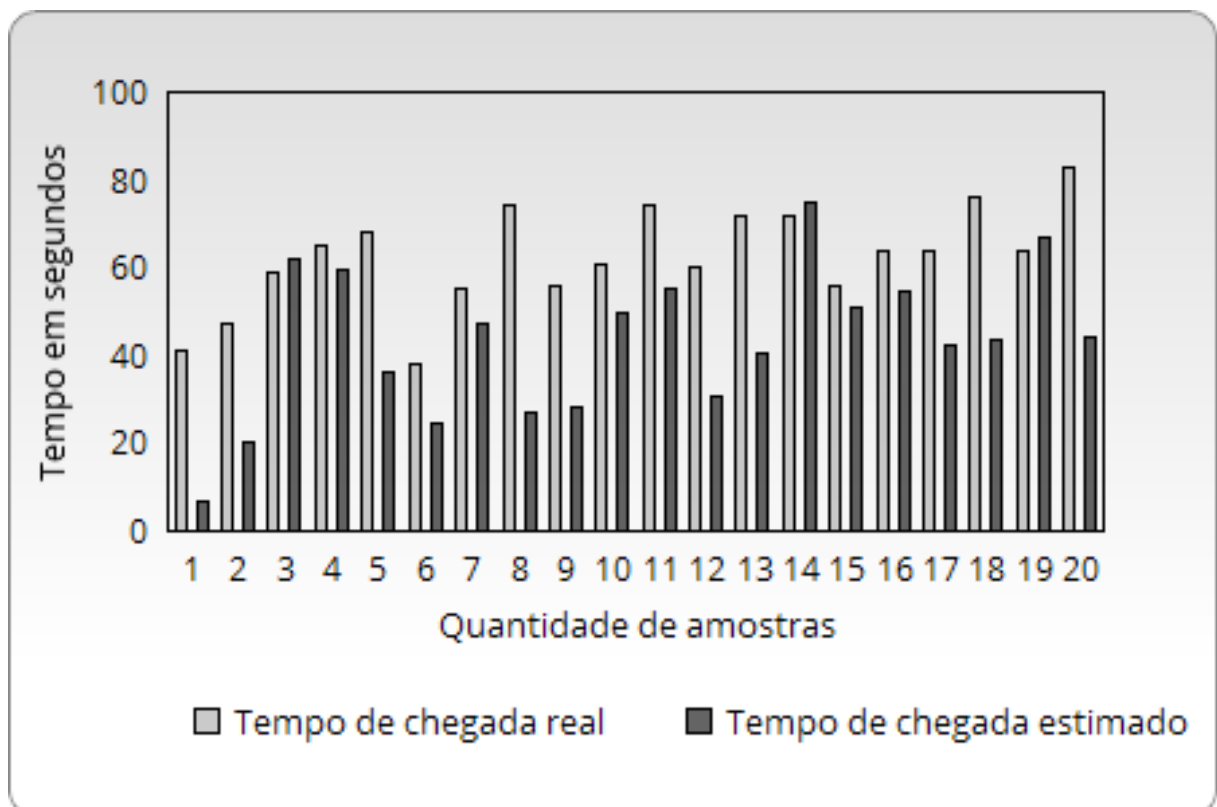


Figura 6 – *Tempo estimado de chegada comparado ao tempo real.*

Para verificar a eficácia do algoritmo apresentado, e comparar o tempo estimado com o tempo real de chegada, as seguintes considerações foram feitas para

cada amostra coletada:

1. Seleciona-se na base de dados uma rota específica, neste caso a T-131 e um ônibus específico para ser analisado em todo o experimento;
2. Uma amostra selecionada é composta por 3 *logs* (P_1, P_2, P_3) recuperados de forma sequencial de acordo com a data de registro, assumindo assim uma ordem cronológica nos eventos coletados;
3. Cada *log*, é composto de informações como Data e hora da coleta, posição atual em coordenadas cartesianas do veículo, velocidade atual do veículo (*km/h*), identificador único do veículo e outras.
4. O algoritmo proposto, assume que P_2 é o *log* que identifica o momento atual do ônibus a ser analisado, levando em conta informação agregada de velocidade e localização atuais para a estimativa, assim P_1 é o ponto inicial do segmento de rota a qual o ônibus pertence, e P_3 o final.
5. O cálculo da estimativa, portanto, é relativo ao tempo em que P_2 leva para chegar no ponto P_3 .
6. Quando o algoritmo apresenta o resultado referente à uma amostra, este é comparado com a diferença dos tempos de medição de P_2 e P_3 real, contidos em seu próprio documento de *log*.
7. Por fim, calcula-se a taxa de erro relacionada ao tempo estimado, dado pelo algoritmo proposto e o tempo real de chegada de um ônibus P_2 no ponto P_3 .

4.2 Limitações

O método proposto para a análise da eficácia do algoritmo, tem algumas limitações que se refletiram no desempenho do algoritmo apresentado na seção subsequente.

O primeiro fator limitante, que pode ter causado um impacto na eficácia do algoritmo, foi o fato de se desconsiderar o tempo de parada, ou seja, o período em que um veículo fica estacionado nos seus pontos de parada no percorrer de sua rota para que os passageiros embarquem e desembarquem de suas viagens.

Uma segunda condição a ser considerada, foi o fato de que não aconteceu em nenhuma das amostras coletadas para o experimento, de P_1 , P_2 e P_3 pertencerem a um mesmo segmento de rota, o que gera uma divergência no uso da velocidade histórica a ser utilizada pelo algoritmo proposto, que é a variante que tem forte peso no resultado da estimativa de acordo com (SUN et al., 2014).

Por fim, outro fato que deve ser considerado na análise do estudo apresentado, este relacionado ao próprio contexto de transporte da cidade de Uberlândia, é o fato de os pontos de parada da rota analisada, T-131, estarem distribuídos em distâncias que nunca ultrapassam 800 metros entre si([Mapa](#)), em uma avenida da cidade que contém muitos semáforos e um intenso fluxo de veículos e pessoas.

4.3 Resultados da Análise

A figura 7 representa o erro calculado para a análise acima. Como foi apontado na seção anterior, nota-se que o algoritmo performou com maior taxa de erro do que o apresentado em (SUN et al., 2014), com uma taxa média de erro absoluto de 0.32738%, dado por:

$$\frac{1}{n} \sum_{n=1}^n \frac{|t_i - t_o|}{t_o} \times 100\%$$

onde:

n = quantidade de predições

t_i = tempo de chegada previsto pelo algoritmo

t_o = tempo de chegada real

Essa taxa de erro, é a taxa atribuída a cada amostra coletada e analisada na avaliação feita, e indica a diferença de tempo, em segundos, do prazo real em que um ônibus levou para chegar a um determinado ponto de chegada, com o tempo estimado pelo algoritmo apresentado.

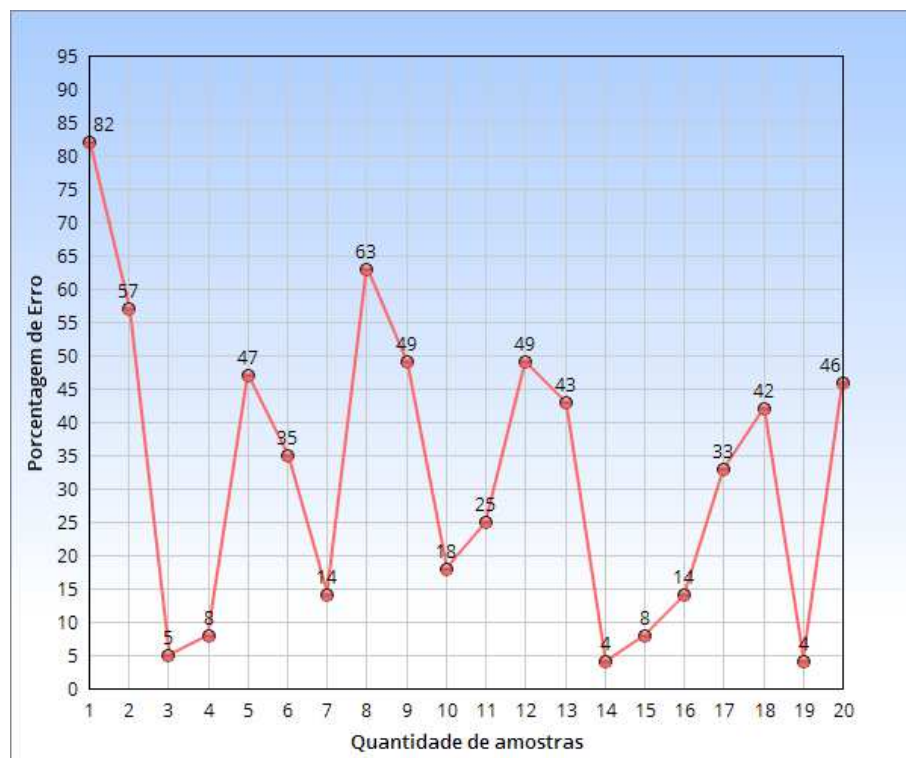


Figura 7 – Erro do tempo estimado de chegada comparado ao tempo real.

Conclusão

A importância de uma predição de tempo estimado de chegada em pontos de parada de ônibus, é uma informação que ajuda não só passageiros como também autoridades a usufruir e prover uma melhor mobilidade nas cidades.

Apesar de o algoritmo de previsão tempo estimado de chegada proposto ter apresentado uma taxa de erro superior quando comparada à bibliografia apresentada, ainda sim pode reduzir o tempo de espera de passageiro para segundos, o que pode trazer um impacto significativo no ir e vir dos passageiros da cidade de Uberlândia.

O algoritmo adaptado para este trabalho, assim como sua implementação foram construídas de maneira relativamente simples, para serem avaliados no cenário proposto. Nos trabalhos subsequentes, no intuito de melhorar a precisão das predições, deve-se adicionar o parâmetro de tempo em que um ônibus permanece estacionado em um ponto de parada para o cálculo das estimativas de chegada, verificando posteriormente se há melhora na taxa de erro apresentada.

Ao final do desenvolvimento deste trabalho, portanto, presume-se que este tenha demonstrado por meios analíticos e exemplificados, que a informação de previsão de tempo de chegada de ônibus, mesmo que com uma taxa de erro atribuída e calculada de maneira simples, poderia estar presente de maneira mais disseminada nos centros urbanos, onde ônibus circulam diariamente provendo viagens de passageiros em suas jornadas.

Referências

- BACKENDLESS. *WHAT IS BACKEND AS A SERVICE?* 2017. Disponível em: [<https://backendless.com/what-is-backend-as-a-service/>](https://backendless.com/what-is-backend-as-a-service/). Acesso em: 24 jul. 2017. Citado na página 14.
- CHIEN, S. . J.; DING, Y.; WEI, C. Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering*, p. 429–438, 2002. Citado na página 17.
- CITTAMOBIL. 2017. Disponível em: [<http://www.cittati.com.br/>](http://www.cittati.com.br/). Acesso em: 24 jun. 2017. Citado na página 19.
- FIGUEIREDO, L. et al. Towards the development of intelligent transportation systems. *2001 IEEE Intelligent Transportation Systems Conference Proceedings*, p. 1206–1211, 2001. Citado na página 11.
- GONG, J.; LIU, M.; ZHANG, S. Hybrid dynamic prediction model of bus arrival time based on weighted of historical and real-time gps data. *Control and Decision Conference (CCDC), 2013 25th Chinese*, n. 8, p. 5, 2013. Citado na página 15.
- GOODALL, W. et al. The rise of mobility as a service. *Deloitte Review*, n. 20, p. 20, 2017. Citado na página 11.
- GOOGLE. *Chrome V8*. 2017. Disponível em: [<https://developers.google.com/v8/>](https://developers.google.com/v8/). Acesso em: 18 jun. 2017. Citado na página 22.
- JEONG, R.; RILETT, L. R. Bus arrival time prediction using artificial neural network model. *IEEE Intelligent Transportation Systems Conference*, v. 14, n. 8, p. 988–993, 2004. Citado na página 17.
- KIDWELL, B. Predicting transit vehicle arrival times. *GeoGraphics Laboratory, Bridgewater State College, Bridgewater, Mass.*, 2001. Citado na página 16.
- LI, Y.; MANOHARAN, S. A performance comparison of sql and nosql databases. *Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on*, p. 5, 2013. Citado na página 23.
- LIN, W.-H.; ZENG, J. Experimental study on real-time bus arrival time prediction with gps data. *Journal of the Transportation Research Board*, No. 1666, TRB, National Research Council, Washington, D.C, p. 101–109, 1999. Citado na página 16.
- MARIANO, C. L. Benchmarking javascript frameworks. *Dublin Institute of Technology*, p. 114, 2017. Citado na página 21.
- MARSH, N.; PADAMSEY, N. *Designing the next generation of transport platforms*. 2017. Disponível em: <https://www.theguardian.com/atkins-partner-zone/2017/mar/09/designing-the-next-generation-of-transport-platforms>. Acesso em: 04 ago. 2017. Citado na página 11.

- MONGODB. *What Is MongoDB*. 2016. Disponível em: <<https://www.mongodb.com/what-is-mongodb>>. Acesso em: 20 jun. 2017. Citado na página 22.
- MOOVIT. 2017. Disponível em: <<https://www.company.moovitapp.com/>>. Acesso em: 24 jun. 2017. Citado na página 20.
- MOZILLA. *About JavaScript*. 2017. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript>. Acesso em: 16 jun. 2017. Citado na página 21.
- NODEJS. 2017. Disponível em: <<https://nodejs.org/>>. Acesso em: 18 jun. 2017. Citado na página 22.
- PATNAIK, J.; CHIEN, S.; BLADIKAS, A. Estimation of bus arrival times using apc data. *Journal of Public Transportation*, v. 7, n. 1, p. 1–20, 2004. Citado na página 16.
- PIRES, J. *O que é API? REST e RESTful? Conheça as definições e diferenças!* 2015. Disponível em: <<https://becode.com.br/o-que-e-api-rest-e-restful/>>. Acesso em: 21 jun. 2017. Citado na página 23.
- PLURALSIGHT. *What's the Difference Between the Front-End and Back-End?* 2015. Disponível em: <<https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>>. Acesso em: 24 jul. 2017. Citado na página 14.
- PROJETOS, F. *O que é uma cidade inteligente?* 2017. Disponível em: <<http://fgvprojetos.fgv.br/noticias/o-que-e-uma-cidade-inteligente>>. Acesso em: 04 ago. 2017. Citado na página 11.
- SCHWEIGER, L. C. Tcrp synthesis 48: Real-time bus arrival information systems: A synthesis of transit practice. *Transportation Research Board of the National Academies, Washington, D.C.*, 2003. Citado na página 15.
- SUN, D.; FU, L. Cellular phone based real-time bus arrival information system. *Eighth International Conference on Applications of Advanced Technologies in Transportation Engineering, Beijing*, 2004. Citado na página 15.
- SUN, D. et al. Predicting bus arrival time on the basis of global positioning system data. *Transportation Research Record: Journal of the Transportation Research Board*, v. 2034, n. 8, p. 62–72, 2014. Citado 7 vezes nas páginas 15, 16, 17, 18, 28, 33 e 34.
- TECHOPEDIA. *Intelligent Transportation System*. 2017. Disponível em: <<https://www.techopedia.com/definition/4462/intelligent-transportation-system-its>>. Acesso em: 04 ago. 2017. Citado na página 11.
- TILKOV, S.; VINOSKI, S. Node.js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, n. 6, p. 80–83, 2010. Citado na página 22.
- WEIGANG, L. et al. Algorithms for estimating bus arrival times using gps data. *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*, p. 868–873, 2002. Citado na página 28.