

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**TIAGO STAPENHORST MARTINS**

**MAP MATCHING: UMA ANÁLISE DE DADOS STREAMING DE TRAJETÓRIAS  
DE GPS NO TRANSPORTE PÚBLICO**

**CURITIBA**

**2022**

**TIAGO STAPENHORST MARTINS**

**MAP MATCHING: UMA ANÁLISE DE DADOS STREAMING DE TRAJETÓRIAS  
DE GPS NO TRANSPORTE PÚBLICO**

**Map Matching: A Public Transport GPS Trajectory Streaming Data Analysis**

Dissertação apresentada como requisito para obtenção do título de Mestre em Computação Aplicada do Programa de Pós Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Nádia Puchalski  
Kozievitch

**CURITIBA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



TIAGO STAPENHORST MARTINS

**MAP MATCHING: UMA ANÁLISE DE DADOS DE STREAMING DE TRAJETÓRIAS DE GPS NO TRANSPORTE PÚBLICO**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Computação Aplicada da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Sistemas Computacionais.

Data de aprovação: 22 de Agosto de 2022

Dra. Nadia Puchalski Kozievitch, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcelo De Oliveira Rosa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Marcos Sfair Sunye, Doutorado - Universidade Federal do Paraná (Ufpr)

Dra. Tatiana Maria Cecy Gadda, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 22/08/2022.

Dedico este trabalho à minha família, amada e amigos pela paciência, apoio incondicional e compreensão em meus momentos de ausência.

Ao lê-lo eu me leio, melhor me entendo. Somos  
do mesmo sangue, companheiros no mesmo  
mundo.

Rubem Alves

## RESUMO

Garantir um transporte público que supra as necessidades de uma população crescente é um desafio, especialmente em países em desenvolvimento onde os recursos e investimentos são limitados. Com o barateamento e a instalação de dispositivos de *Internet of Things* (IoT) como embarcados, sensores, *Global Positioning System* (GPS) em ônibus de transporte público uma grande quantidade de dados pode ser gerada e utilizada como base para tomadas de decisão. Entretanto se os dados forem afetados por erros e incertezas tais análises podem ser inválidas. Os dados abertos de movimentação dos ônibus de Curitiba é vasto mas apresentam inconsistências e não informam o horário de passagem dos ônibus nos pontos de ônibus. A grande quantidade de dados por si só será valiosa se processamentos e algoritmos extraíam o valor destes dados. Este trabalho tem como objetivo um método de quatro etapas para analisar os dados de *Streaming* de trajetórias de GPS, contendo 1) análise e limpeza dos dados; 2) extração de azimutes; 3) um método para detecção do momento (horário) de passagem dos ônibus nos respectivos pontos de ônibus de sua linha de operação e 4) correlação dos tempos reais e teóricos da passagem nos pontos de ônibus. O trabalho utiliza conceitos de Sistemas de Informação Geográficas, Cidades Inteligentes e Dados Abertos. Testes realizados em dados abertos de *Streaming* de trajetórias de GPS de transporte público de Curitiba ilustraram a eficiência da metodologia dos algoritmos propostos, além de indicar fatores para a melhoria dos dados.

**Palavras-chave:** sistemas de informao geogricas; *map matching*; dados abertos; transporte plico; cidades inteligentes.

## ABSTRACT

Ensuring public transport that meets the needs of a growing population is a challenge, especially in developing countries where resources and investment are limited. With the cheapening and installation of Internet of Things (IoT) devices such as embedded, sensors, Global Positioning System (GPS) in public transport buses, a large amount of data can be generated and used as basis for decision making. However, if the data are affected by errors and uncertainties such analyzes may be invalid. The open data on the movement of buses in Curitiba is vast, but they present inconsistencies and do not inform the time of passage of buses at bus stops. The large amount of data by itself will be valuable if processing and algorithms extract the value of this data. This work presents a four-step method to analyze the Streaming data from GPS trajectories, containing 1) data analysis and cleaning; 2) extraction of azimuths; 3) a method for detecting the moment (time) of buses passing at the respective bus stops of their operating line and 4) correlation of the real and theoretical times of passing at the bus stops. Concepts of Geographic Information Systems, Smart Cities and Open Data are used. Tests performed on open Streaming data from GPS trajectories of public transport in Curitiba illustrated the efficiency of the methodology of the proposed algorithms, in addition to indicating factors for data improvement.

**Keywords:** geographic information systems; map matching; open data; public transport; smart cities.

## LISTA DE FIGURAS

Figura 1 – Trajeto feito pelo ônibus BA125 da linha 216 no dia 03/05/2019 segundo os dados abertos da URBS . . . . .	13
Figura 2 – Todos os pontos de ônibus de Curitiba em 2016 agrupados pelo algoritmo <i>Marker Cluster</i> . . . . .	17
Figura 3 – Todos os pontos de ônibus de Curitiba em 2016 em 40 grupos através do algoritmo <i>K-Means</i> . . . . .	18
Figura 4 – Grafo gerado pelo algoritmo <i>geolayout</i> indicando as conexões entre origens e destinos dos passageiros de ônibus em Curitiba . . . . .	19
Figura 5 – Velocidade média em km/h da linha 40 (Interbairros IV) à esquerda e linha 550 (Ligeirão - Pinheirinho/Carlos Gomes) à direita . . . . .	20
Figura 6 – Tela do <i>dashboard</i> de análise dos dados de ônibus . . . . .	21
Figura 7 – Representação da arquitetura de um SIG . . . . .	22
Figura 8 – Representação geométrica de uma Grade Regular . . . . .	24
Figura 9 – Algoritmo de Map Matching em ação sobre pontos de GPS . . . . .	25
Figura 10 – Componentes de uma <i>Smart City</i> . . . . .	27
Figura 11 – Sequência de execução dos algoritmos . . . . .	32
Figura 12 – Composição da frota de ônibus da URBS . . . . .	35
Figura 13 – Visualização dos trechos de trajeto da linha 216 (Cabral/Portão) no dia 18/04/2019 através do software QGIS . . . . .	36
Figura 14 – Representação da arquitetura da aplicação desenvolvida . . . . .	40
Figura 15 – Exemplo do ônibus BA604 da linha 216 “dando ré” . . . . .	45
Figura 16 – Exemplo do ônibus HA610 da linha 216 “dando ré” . . . . .	45
Figura 17 – Exemplo de ponto de ônibus da linha 216 cadastrado muito distante do meio fio da rua . . . . .	46
Figura 18 – Exemplo de dois pontos de ônibus da linha 216 muito próximos, um em cada lado diferente da rua . . . . .	47
Figura 19 – Metodologia do enriquecimento de dados para descobrir azimutes de cada ponto de ônibus em “pontosLinha” . . . . .	48
Figura 20 – Resultado da aplicação da metodologia descrita pela Figura 19 para a linha 216 . . . . .	49



<b>Figura 21 – Fluxograma explicando a ordem do processamento de dados para encontrar . . . . .</b>	<b>50</b>
<b>Figura 22 – Horários de chegada programados (tabelados) para a linha 216 no dia 03/05/2019 . . . . .</b>	<b>51</b>
<b>Figura 23 – Resultado do algoritmo de correspondência do horário de passagem, em que o trajeto do ônibus foi coincido com a localização real dos pontos de ônibus, utilizando o mapa do Open Street Map . . . . .</b>	<b>52</b>
<b>Figura 24 – Ilustração sobre os intervalos de limites criados (L1, L2, L3, L4) para um conjunto de horários programados de um mesmo ponto de ônibus para um mesmo veículo . . . . .</b>	<b>53</b>
<b>Figura 25 – Gráfico mostrando a frequência da ocorrência de atrasos ou adiantamentos durante o período observado . . . . .</b>	<b>56</b>
<b>Figura 26 – Gráfico de pizza mostrando ocorrência de atrasos e adiantamentos entre 7,5 minutos em comparação ao restante . . . . .</b>	<b>56</b>
<b>Figura 27 – Gráfico de linha mostrando a média dos atrasos (à esquerda) e do desvio padrão (à direita) no eixo vertical Y, juntamente com as horas do dia, expressas no eixo horizontal X . . . . .</b>	<b>57</b>

## LISTA DE TABELAS

<b>Tabela 1 – Tipos de ônibus e suas capacidades . . . . .</b>	<b>33</b>
<b>Tabela 2 – Descrição dos arquivos/tabelas disponibilizadas pela URBS e relação dos quais foram analisados e utilizados neste trabalho . . . . .</b>	<b>36</b>
<b>Tabela 3 – Descrição dos dados contidos na tabela “linhas” . . . . .</b>	<b>37</b>
<b>Tabela 4 – Descrição dos dados contidos no arquivo “pontosLinha” . . . . .</b>	<b>37</b>
<b>Tabela 5 – Descrição dos dados contidos no arquivo “shapeLinha” . . . . .</b>	<b>38</b>
<b>Tabela 6 – Descrição dos dados contidos na tabela “tabelaLinha” . . . . .</b>	<b>38</b>
<b>Tabela 7 – Descrição dos dados contidos na tabela “tabelaVeiculo” . . . . .</b>	<b>39</b>
<b>Tabela 8 – Descrição dos dados contidos na tabela “veiculos” . . . . .</b>	<b>39</b>
<b>Tabela 9 – Arquivos “veiculos” e análise de quantidade de veículos e amostras espaço-temporais na linha 216 . . . . .</b>	<b>41</b>
<b>Tabela 10 – Arquivos “veiculos” e comparação das datas nele contidas com a data do arquivo para a linha 216 . . . . .</b>	<b>42</b>
<b>Tabela 11 – Comparação dos veículos contidos em “veiculos” e “tabelaVeiculo” na linha 216 . . . . .</b>	<b>43</b>
<b>Tabela 12 – Comparação da contagem da quantidade de veículos diferentes em cada dia para o arquivo “tabelaLinha” e “tabelaVeiculo” . . . . .</b>	<b>44</b>
<b>Tabela 13 – Exemplo de dados contidos no arquivo “pontosLinha + azimutes” . . . .</b>	<b>49</b>
<b>Tabela 14 – Exemplo da janela de tempo usada para realizar a correspondência de posições do ônibus com pontos de ônibus (PO) . . . . .</b>	<b>51</b>
<b>Tabela 15 – Quantidade de veículos únicos antes e depois dos algoritmos e filtros .</b>	<b>55</b>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	13
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>15</b>
<b>2.1</b>	<b>Dados Abertos</b>	<b>15</b>
2.1.1	Dados Abertos relacionados ao transporte público em Curitiba	16
<b>2.2</b>	<b>Sistemas de Informações Geográficas (SIGs)</b>	<b>20</b>
2.2.1	Arquitetura de funcionamento de um SIG	21
2.2.2	<i>Map Matching</i>	24
<b>2.3</b>	<b>Cidades Inteligentes</b>	<b>26</b>
2.3.1	Sistemas de Transporte Inteligentes	27
<b>2.4</b>	<b>Desafios das Áreas Relacionadas</b>	<b>28</b>
2.4.1	Desafios em Computação e Sistemas de Informações Geográficas	28
2.4.2	Desafios em Dados Abertos	29
2.4.3	Desafios em Cidades Inteligentes	30
<b>3</b>	<b>METODOLOGIA E DESENVOLVIMENTO</b>	<b>32</b>
<b>3.1</b>	<b>O Funcionamento dos Ônibus</b>	<b>33</b>
<b>3.2</b>	<b>Sobre os Dados Abertos da URBS</b>	<b>35</b>
<b>3.3</b>	<b>Algoritmo 1 - Importação dos dados da URBS</b>	<b>38</b>
3.3.1	Inconsistências, Redundâncias e Peculiaridades dos Dados	40
3.3.1.1	<u>Arquivos vazios e com dados faltantes</u>	41
3.3.1.2	<u>A qual dia o arquivo “veiculo” se refere?</u>	41
3.3.1.3	<u>Quantidade de veículos em operação diferem dos programados</u>	41
3.3.1.4	<u>Pontos de ônibus não fazem parte dos pontos de ônibus comuns ao trajeto da linha</u>	42
3.3.1.5	<u>Quantidade de veículos em “tabelaVeiculo” é diferente de “tabelaLinha”</u>	43
3.3.1.6	<u>Veículos de uma linha operando em outras</u>	43
3.3.1.7	<u>Problemas de amostragem ou precisão de GPS</u>	44
3.3.1.8	<u>Comportamentos estranhos de amostragem - ônibus “dando ré”</u>	44
3.3.1.9	<u>Imprecisões no cadastro dos pontos de ônibus</u>	44
3.3.1.10	<u>Redundância em dados de colunas de arquivos diferentes</u>	45

3.3.1.11	Ausência de direção de passagem do ponto de ônibus . . . . .	46
3.3.1.12	Dados alternativos possíveis de serem analisados . . . . .	46
<b>3.4</b>	<b>Algoritmo 2 - Extração de Azimutes . . . . .</b>	<b>46</b>
<b>3.5</b>	<b>Algoritmo 3 - <i>Map Matching</i> da trajetória dos ônibus com os pontos de ônibus . . . . .</b>	<b>49</b>
<b>3.6</b>	<b>Algoritmo 4 - Comparação Real x Programado . . . . .</b>	<b>51</b>
<b>3.7</b>	<b>Desenvolvimento . . . . .</b>	<b>53</b>
3.7.1	Implementação do Algoritmo 3 - <i>Map Matching</i> da trajetória dos ônibus com os pontos de ônibus . . . . .	53
3.7.2	Implementação do Algoritmo 4 - Comparação Real x Programado . . . . .	54
3.7.3	Limitações ou particularidades encontradas . . . . .	55
3.7.3.1	Limitações do Algoritmo 1 . . . . .	55
3.7.3.2	Limitações do Algoritmo 2 . . . . .	56
3.7.3.3	Limitações do Algoritmo 3 . . . . .	57
3.7.3.4	Limitações do Algoritmo 4 . . . . .	58
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>59</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>60</b>
	<b>ANEXO A ALGORITMO DE DESCOBERTA DE AZIMUTES DOS PONTOS DE ÔNIBUS . . . . .</b>	<b>65</b>
	<b>ANEXO B ALGORITMO DE DESCOBERTA DE PASSAGEM DOS ÔNIBUS POR SEUS PONTOS . . . . .</b>	<b>70</b>

## 1 INTRODUÇÃO

Quando mencionamos o assunto urbanização, especialmente quando diz respeito a variedade de opções de mobilidade e transporte, refletimos sobre a forma que os meios de locomoção desempenham função essencial na qualidade de vida dos cidadãos que necessitam de agilidade nos centros urbanos, destacando-se, nesta conjuntura, o transporte público como o meio de locomoção mais acessível para a população de países em desenvolvimento (KUMAR; GANGULY, 2018).

Ainda sobre o crescimento dos grandes centros urbanos, é importante frisar que as organizações responsáveis por gerenciar o transporte público de ônibus buscam constantemente ferramentas que auxiliem na qualidade, eficácia e eficiência do transporte público (Un-Habitat, 2013). Esta busca por melhorias objetiva a qualidade para os usuários atuais, além de criar atrativos para novos usuários. O aumento da densidade populacional, aliado ao consequente aumento do número de carros particulares nos ambientes urbanos, sem estar acompanhado da melhoria na infraestrutura e no sistema de transporte, traz um impacto negativo no trânsito e na mobilidade urbana em geral (CRISTOBAL *et al.*, 2018; TU; YANG, 2016). A existência de um meio de transporte de alta qualidade, eficiente, que conecte trabalhadores e espaços de demanda de trabalho acaba sendo essencial para o desenvolvimento de uma cidade (ALBINO; BERARDI; DANGELICO, 2015).

Através do aperfeiçoamento na qualidade do serviço, empresas aliam-se à pesquisadores em transporte para direcionar seus investimentos em ferramentas que permitem o uso de Sistemas de Informações Geográficas (SIGs), Sistemas de Transporte Inteligentes (STIs), aliados aos Sistemas de Navegação por Satélite (GNSS, *Global Navigation Satellite System*), termo genérico para o *Global Positioning System* (GPS) utilizado no Ocidente, para ajudar na análise da dinâmica do trânsito urbano. O objetivo é aprimorar a previsibilidade dos serviços e corrigir erros nos dados coletados pelos ônibus, além de proporcionar segurança na tomada de decisões, ferramentas e sugestões para que motoristas de ônibus, urbanistas, engenheiros de tráfego e usuários planejem trajetos e horários para executar de forma otimizada suas funções. (MENEQUETTE; GRANDE; LOUREIRO, 2018).

Conseqüentemente, a popularização e o barateamento de dispositivos IoT, sensores, GPS e embarcados de maneira geral criaram uma nova demanda para que as metrópoles se tornem mais inteligentes (*Smart City*) e interconectadas (AZAMBUJA, 2016), especificamente no que tange ao transporte público: o volume de dados gerados por veículos, aliado à demanda por informações que advém destes dados, permite que novas análises sejam realizadas, para que seja feita uma melhora na eficiência, eficácia e qualidade do transporte sem requerer investimentos adicionais em outras esferas do serviço (compra de veículos, reformas em ruas, etc), além de auxiliar gestores com informações reais e de qualidade, visto que estas serão provenientes de bancos de dados integrados aos veículos. Neste prisma destacam-se o fato de uma Cidade inteligente ser uma característica atrativa do ponto de vista econômico (HÖJER;

WANGEL, 2015; MENEGUETTE; GRANDE; LOUREIRO, 2018) , ainda que um dos grandes desafios desta Cidade se concentre na imensa quantidade de dados colhidos em tempo real.

Em Curitiba grande parte da frota de ônibus do transporte público é operada por empresas privadas por meio de licitação<sup>1</sup>, porém sua gestão é feita pelo município através da empresa estatal Urbanização de Curitiba (URBS)<sup>2</sup> a qual equipou os ônibus, desde 2012<sup>3</sup>, com dispositivos GPS atrelados a um computador de bordo, que enviam dados para o Centro de Controle Operacional (CCO), onde agentes de fiscalização da URBS monitoram a operação do transporte coletivo em tempo real<sup>4</sup>. Desde o ano de 2017 estes dados passaram a ser disponibilizados ao público, sendo processados, armazenados em arquivos diários e acessíveis na internet sem necessidade de qualquer tipo de cadastro ou custo<sup>5</sup> (como ilustra a Figura 1). Define-se esta abertura e disponibilização de dados das empresas e organizações ao público sem restrições de: patentes, *copyright*, Dados Abertos ou *Open Data* (AZAMBUJA, 2016).

A literatura acadêmica sobre o assunto, notadamente no tópico do transporte curitibano, é formada por: pesquisas direcionadas na temática de distribuições dos pontos de ônibus pela capital, intencionando detectar os locais pouco abastecidos pelo transporte (SILVA *et al.*, 2016); calcular o tempo de deslocamento entre a origem o destino dos passageiros (PARCIANELLO, 2020; JUNIOR, 2017); foco na trajetória em unidades de saúde e educação (SOUZA *et al.*, 2020); análise exploratória geral (VILA *et al.*, 2016; VILA, 2016; PARCIANELLO *et al.*, 2018; BARCZYSZYN, 2015) e , por fim, em desenvolvimento de software de caronas solidárias (*car-pooling*) para a comunidade universitária (LARA; MARTINS, 2016; MARTINS; KOZIEVITCH, 2015) (trabalhos anteriores do autor).

O presente trabalho apresenta, de início, um “algoritmo de importação e interpretação (*parse*) dos dados abertos da URBS” presentes no portal de dados abertos da prefeitura de Curitiba. Em seguida, é apresentado um “algoritmo de extração de azimutes”, com a intenção de demonstrar em que direção os ônibus passam por cada ponto. Após isto, é apresentado o “algoritmo de correspondência do momento de passagem”, o qual permite que sejam feitas análises de detecção da chegada e ônibus, analisando todos os fatores de interesse deste trabalho (atrasos, adiantamentos, quais ônibus e linhas, sob quais horários). O diferencial desta análise advém do uso de *Map Matching* para analisar a problemática das imprecisões das fontes de dados georreferenciadas sobre o tempo de chegada do ônibus, conforme exposto na Figura1, em que um dos trajetos gerados pelo GPS ônibus (a linha cinza à esquerda) ultrapassa os limites da rua, atravessando construções. Esta etapa foi publicada em artigo de conferência (MARTINS

<sup>1</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/noticia/contratos-com-empresas-de-transporte-estao-disponiveis-na-internet>> Acesso em: 20 out, 2022.

<sup>2</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/institucional/prestacao-contas>> Acesso em: 20 out, 2022.

<sup>3</sup> Disponível em: <<https://www.gazetadopovo.com.br/vida-e-cidadania/informacao-sobre-linhas-de-oni-bus-e-deficiente-2bd8wsjqxmv4rz5jipboehce/>> Acesso em: 20 out, 2022.

<sup>4</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/transporte/monitoramento-cco>> Acesso em: 20 out, 2022.

<sup>5</sup> Disponível em: <<https://www.curitiba.pr.gov.br/dadosabertos/busca/>> Acesso em: 20 out, 2022.

**Figura 1 – Trajeto feito pelo ônibus BA125 da linha 216 no dia 03/05/2019 segundo os dados abertos da URBS**



**Fonte: Autoria própria (2022).**

*et al.*, 2022). O quarto e último algoritmo compara os tempos reais de passagem nos pontos de ônibus com os teóricos.

### 1.1 Objetivo Geral

O trabalho tem como objetivo um método de quatro etapas para analisar os dados de streaming de trajetórias de GPS, contendo 1) análise e limpeza dos dados; 2) extração de azimutes; 3) um método para detecção do momento (horário) de passagem dos ônibus nos respectivos pontos de ônibus de sua linha de operação e 4) correlação dos tempos reais e teóricos da passagem nos pontos de ônibus. Para isto serão usados conceitos das áreas de Sistemas de Informação Geográficas, Cidades Inteligentes e Dados Abertos.

### 1.2 Objetivos Específicos

- Analisar e desenvolver algoritmo para importar dados abertos da URBS presentes no portal de dados abertos da Prefeitura de Curitiba;
- Realizar uma análise exploratória dos dados históricos de movimentação dos ônibus, linhas e pontos de ônibus para compreender possíveis problemas e desafios presentes em uma linha de ônibus;
- Efetuar limpeza de possíveis inconsistências nos dados encontrados;

- Analisar e desenvolver algoritmo para enriquecer os dados dos pontos de ônibus com a direção de passagem dos ônibus, dado ausente nos dados abertos disponibilizados pela prefeitura;
- Analisar e desenvolver algoritmo para encontrar horário de passagem de um ônibus em todos seus pontos de parada oficiais.

As seções seguintes estão organizadas da seguinte forma:

- Revisão da Literatura, onde explora-se o que a literatura diz sobre os tópicos de Dados Abertos, Sistemas de Informações Geográficas (SIGs) e Cidades Inteligentes (*Smart Cities*).
- Metodologia e Desenvolvimento, onde explica-se de que forma o problema exposto foi atacado, problemas nos dados, soluções encontradas e resultados.
- Conclusão, onde mostra-se um panorama geral do trabalho e das perguntas e respostas a qual pode-se chegar.



## 2 REVISÃO DA LITERATURA

Por conta do aspecto multidisciplinar deste trabalho se torna importante esclarecer conceitos e dar exemplos sobre os temas abordados. Os três assuntos primários que compõem este trabalho são:

- Dados Abertos ou *Open Data*;
- Sistemas de Informações Geográficas (SIGs) ou *Geographic Information Systems* (GIS);
- Cidades Inteligentes ou *Smart Cities*.

### 2.1 Dados Abertos

Dados Abertos são dados que podem ser usados livremente e redistribuídos por e para qualquer pessoa (DIETRICH *et al.*, 2012). Seus termos de uso podem ser tanto totalmente livres, exigindo apenas que crédito seja dado aos autores (chamada de Atribuição 4.0 Internacional, CC BY 4.0) <sup>1</sup> ou podem exigir além dos créditos, toda modificação resultante destes dados sejam também abertas da mesma forma (chamada de Atribuição Compartilhada 4.0 Internacional, CC BY-SA 4.0) <sup>2</sup>. Estes tipos de dados se enquadram na categoria de *copyleft*, em oposição aos dados protegidos e com restrições de uso público e comercial *copyright*.

O conceito de Dados Abertos popularizou-se junto com as iniciativas e legislações governamentais de transparência, ou seja, facilitar o acesso de dados públicos governamentais a jornalistas, auditores e principalmente à população em geral, uma vez que são de interesse coletivo (AZAMBUJA, 2016).

Em 2011, oito países (África do Sul, Brasil, Estados Unidos, Filipinas, Indonésia, México, Noruega e Reino Unido) assinaram em conjunto o documento *Open Government Declaration* <sup>3</sup>, no qual se comprometem a reduzir a corrupção, incentivar participação civil nos processos públicos, empoderar o cidadão e utilizar tecnologias para tornar o governo mais eficiente e auditável através da disponibilização de Dados Abertos. Dentro dos compromissos em que o Brasil se comprometeu no *Open Government Declaration* estão listados os dados de transporte como *Transport Data (BR104)*<sup>4</sup>.

<sup>1</sup> Disponível em: <[https://creativecommons.org/licenses/by/4.0/deed.pt\\_BR](https://creativecommons.org/licenses/by/4.0/deed.pt_BR)> Acesso em: 20 out, 2022.

<sup>2</sup> Disponível em: <[https://creativecommons.org/licenses/by-sa/4.0/deed.pt\\_BR](https://creativecommons.org/licenses/by-sa/4.0/deed.pt_BR)> Acesso em: 20 out, 2022.

<sup>3</sup> Disponível em: <<https://www.opengovpartnership.org/process/joining-ogp/open-government-declaration/>> Acesso em: 20 out, 2022.

<sup>4</sup> Disponível em: <<https://www.opengovpartnership.org/members/brazil/commitments/BR0104/>> Acesso em: 20 out, 2022.

### 2.1.1 Dados Abertos relacionados ao transporte público em Curitiba

Os dados abertos do transporte público de Curitiba gerenciado pela URBS estão disponíveis no *website* da prefeitura de Curitiba desde 2015<sup>5</sup>, sendo disponibilizados em formato JSON, comprimidos com o padrão XZ<sup>6</sup>, são disponibilizados de forma totalmente aberta. Há também uma outra fonte de dados, formatada em *General Transit Feed Specification* (GTFS) estático, mas é acessível apenas por acesso autenticado<sup>7</sup>, sendo necessário entrar em contato com a URBS, identificar-se, e solicitar um login exclusivo para acesso; estes dados são os mesmos utilizados pelas plataformas *Google Maps* e *Moovit*.

Pensando no público geral a própria URBS disponibiliza em seu *website* oficial<sup>8</sup> duas fontes de dados: a primeira compreendida por uma série de tabelas horárias, uma para cada linha de ônibus, com os momentos previstos de chegada dos veículos nas principais paradas (pontos de ônibus) considerados principais<sup>9</sup>; a segunda consolidada em página web interativa contendo um mapa viário de Curitiba com a última posição dos ônibus, atualizáveis a cada minuto<sup>10</sup>. Nesta última o usuário informa uma linha de ônibus e recebe um mapa com a localização de todos os pontos de ônibus, trajeto e última posição dos ônibus em atividade. Ao clicar em um ônibus exibe-se: a direção que o ônibus está indo (horário vs anti-horário; indo para terminal a vs indo para terminal b) e seu estado com relação à tabela horária: atrasado, adiantado ou no horário.

Qualquer tipo de sistema que de alguma forma envolva decisões baseadas em fatos enfrentará um ciclo de transformação, incluindo etapas como Dados (uma sequência de símbolos quantificados ou quantificáveis), dados abertos (onde qualquer pessoa pode ter acesso a eles, ou use-o, modifique-o e compartilhe-o para qualquer propósito (DIETRICH *et al.*, 2012)), dados vinculados/*linked data* (conjunto de boas práticas para publicar e conectar conjuntos de dados estruturados na web (SHETH, 2011)) e conhecimento (camada mais profunda de informações, é necessário que haja processamento de informação ou um conjunto de informações para que o conhecimento possa ser gerado (DAVENPORT; PRUSAK, 1997)).

Diversos trabalhos acadêmicos vêm explorando estes dados desde então. Os pioneiros em 2016 realizaram diversas análises exploratórias dos dados mencionados juntamente com os dados abertos do Instituto de Pesquisa e Planejamento Urbano de Curitiba (IPPUC)<sup>11</sup> (VILA *et al.*, 2016; VILA, 2016; KOZIEVITCH *et al.*, 2016; SILVA *et al.*, 2016), revelando a grande

<sup>5</sup> Disponível em: <<https://www.curitiba.pr.gov.br/dadosabertos/busca/?pagina=9>> Acesso em: 20 out, 2022.

<sup>6</sup> Disponível em: <<https://www.kernel.org/doc/Documentation/xz.txt>> Acesso em: 20 out, 2022.

<sup>7</sup> Disponível em: <[https://mid.curitiba.pr.gov.br/dadosabertos/TransporteColetivo/2015-11-24\\_Documenta%C3%A7%C3%A3o\\_WEB-SERVICE\\_%E2%80%93\\_TRANSPORTE\\_COLETIVO\\_DE\\_CURITIBA.pdf](https://mid.curitiba.pr.gov.br/dadosabertos/TransporteColetivo/2015-11-24_Documenta%C3%A7%C3%A3o_WEB-SERVICE_%E2%80%93_TRANSPORTE_COLETIVO_DE_CURITIBA.pdf)> Acesso em: 20 out, 2022.

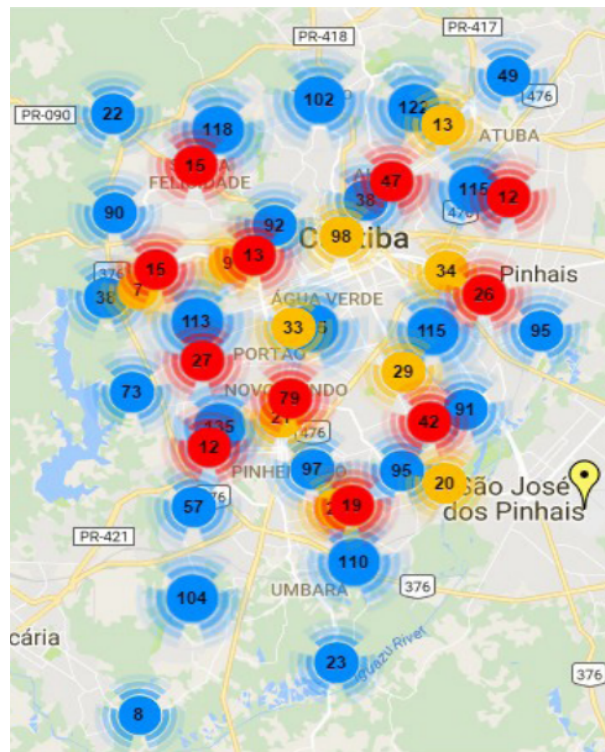
<sup>8</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/>> Acesso em: 20 out, 2022.

<sup>9</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/horario-de-onibus>> Acesso em: 20 out, 2022.

<sup>10</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/mobile/itibus>> Acesso em: 20 out, 2022.

<sup>11</sup> Disponível em: <<http://infocuritiba.ippuc.org.br/imp/>> Acesso em: 20 out, 2022.

**Figura 2 – Todos os pontos de ônibus de Curitiba em 2016 agrupados pelo algoritmo *Marker Cluster***



**Fonte: Vila et al. (2016), Vila (2016).**

quantidade de dados e a riqueza das informações que deles podem ser extraídos. Algumas questões foram analisadas, como a distribuição desigual dos pontos de ônibus ao longo da cidade, especialmente na zona sul, demonstrado pela Figura 3. Outra análise visual dos pontos foi construída utilizando o algoritmo *Marker Cluster*, que permite facilitar o entendimento em mapas onde há uma grande concentração de pontos, demonstrado na Figura 2, onde em azul tem-se os pontos do tipo chapéu chinês, em vermelho as do tipo plataforma e laranja as do tipo tubo (VILA et al., 2016; VILA, 2016).

É válido ressaltar que muitos desafios foram encontrados com o uso desses dados utilizados nos artigos referenciados aqui. Problemas como ambiguidades nos nomes das ruas, dados incorretos e inconsistências são apontados por diversos autores (VILA et al., 2016; BARCZYSZYN, 2015; PARCIANELLO et al., 2018; PARCIANELLO, 2020).

Em outro trabalho (KOZIEVITCH et al., 2016), explorou-se os dados em números e os integrou com outras bases de dados abertas como as do IPPUC e a ferramenta de mapas aberta *OpenStreetMap*<sup>12</sup>. São aproximadamente 9940 os pontos de ônibus incluídos nos dados, dos quais 342 são do tipo "estações tubo", pontos de ônibus especiais que permitem o usuário trocar de ônibus pagando apenas uma passagem.

<sup>12</sup> Disponível em: <<https://www.openstreetmap.org/>> Acesso em: 20 out, 2022.

**Figura 3 – Todos os pontos de ônibus de Curitiba em 2016 em 40 grupos através do algoritmo *K-Means***



**Fonte: Silva *et al.* (2016).**

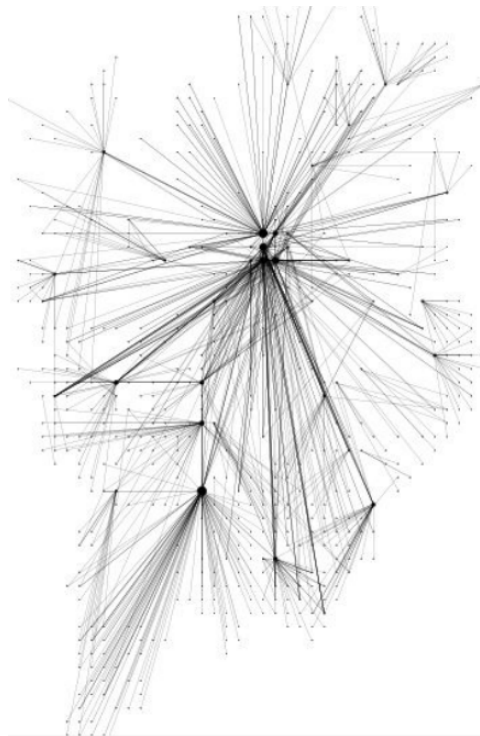
Com a modernização dos sistemas de transporte, também estuda-se a troca de ônibus movidos a diesel por ônibus elétricos com baterias <sup>13</sup>, com o objetivo de se ter um transporte público mais silencioso e com menos emissão de poluentes no ar. Em um estudo de viabilidade técnica para essa troca, usou-se os dados históricos da trajetória dos ônibus para realizar simulações e compreender a autonomia dos ônibus elétricos para alguns tipos de ônibus (biarticulados em especial) em Curitiba (SEBASTIANI; LUDERS; FONSECA, 2016).

Somando as análises dos trabalhos anteriores, Diniz Junior 2017 se propôs a criar um software que permite realizar análise exploratória dos dados abertos da URBS cruzando-os com dados de bilhetagem eletrônica. Os dados de bilhetagem contêm apenas o horário de entrada dos passageiros dentro do ônibus juntamente com o identificador do cartão, o que torna difícil estimar o local de desembarque dos passageiros. Para deduzir os locais de desembarque adotou-se uma abordagem aproximada usando *grids*, fatias retangulares de espaços no mapa, transformando o mapa numa espécie de tabuleiro. A Figura 4 mostra uma perspectiva

<sup>13</sup> Disponível em: <<https://www.curitiba.pr.gov.br/noticias/tecnicos-de-curitiba-visitam-fabricantes-de-ônibus-eletricos-em-sao-paulo/61719>> Acesso em: 20 out, 2022.

dos destinos e origens mais comuns na cidade, mostrando um forte trânsito na região do Centro e Pinheirinho (JUNIOR, 2017). O mesmo trabalho também analisou os ônibus na perspectiva de suas velocidades dentro dos já mencionados *grids*. Uma característica da imagem à direita da Figura 5 é que a velocidade média é menor a medida que o veículo se aproxima do centro da cidade. Em ambas as imagens a velocidade é menor nas extremidades do trajeto (JUNIOR, 2017).

**Figura 4 – Grafo gerado pelo algoritmo *geolayout* indicando as conexões entre origens e destinos dos passageiros de ônibus em Curitiba**



**Fonte: Junior (2017).**

Tendo como base o trabalho de (JUNIOR, 2017), (PARCIANELLO, 2020) desenvolveu uma plataforma em código aberto para exploração visual das aproximações de dados de embarque e desembarque. Diversas perguntas podem ser feitas de forma dinâmica e gráficos são gerados com as respostas conforme ilustrado na Figura 6.

Além disso, pode-se citar eventos como *Open Data Day* em Curitiba, em que equipes se organizaram para criar soluções tecnológicas com o uso de dados abertos da cidade <sup>14</sup>. Uma das equipes criou *onibus.io*, uma página no *github* que agrega informações abertas sobre o transporte público de Curitiba para desenvolvedores e um compilado de cartazes de informações de itinerários com foco em usabilidade e experiência do usuário <sup>15</sup>.

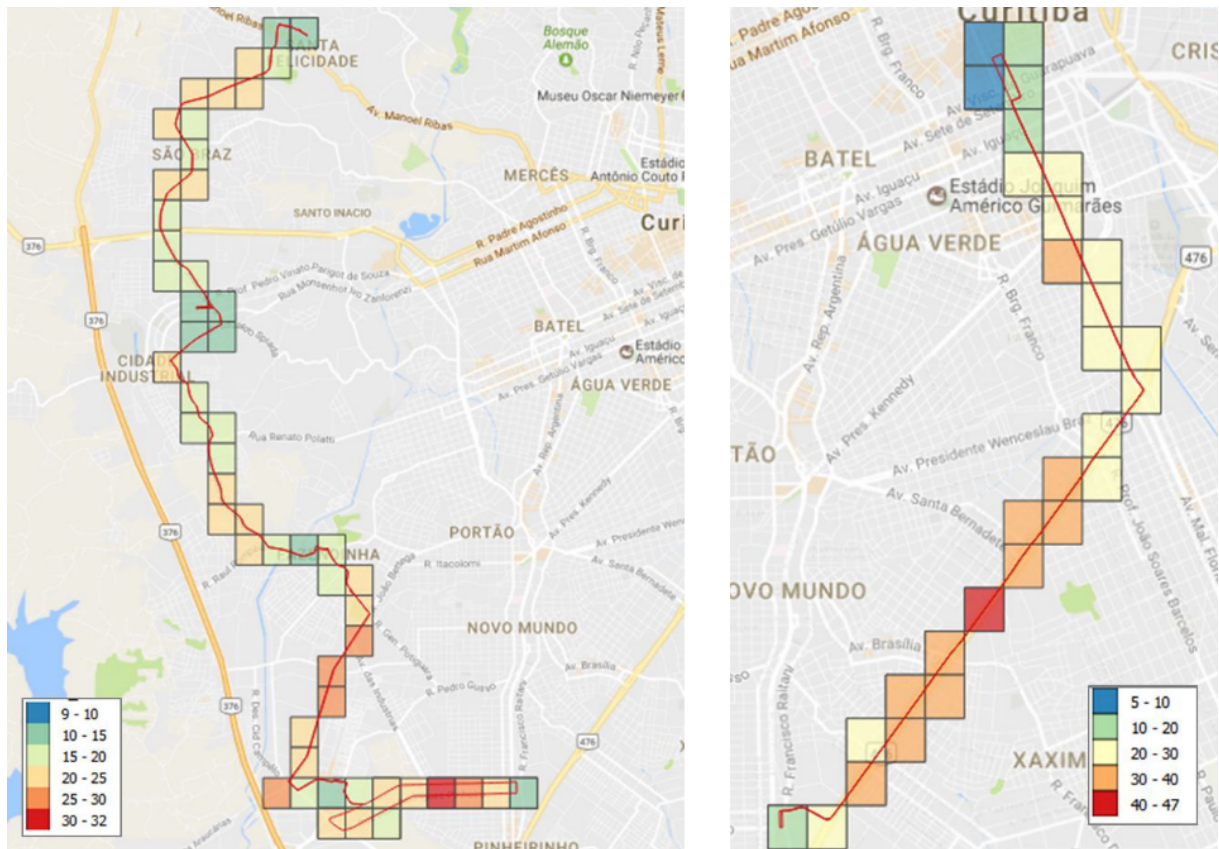
Após analisar todos os trabalhos encontrados que se utilizam do mesmo dado, nenhuma análise ainda foi feita sobre os atrasos dos ônibus com relação as suas tabelas horárias. O

<sup>14</sup> Disponível em: <<https://www.codeforcuritiba.org/2019/04/17/como-foi-o-open-data-day-curitiba-2019/>> Acesso em: 20 out, 2022.

<sup>15</sup> Disponível em: <<https://github.com/CodeForCuritiba/onibus-io>> Acesso em: 20 out, 2022.



**Figura 5 – Velocidade média em km/h da linha 40 (Interbairros IV) à esquerda e linha 550 (Ligeirão - Pinheirinho/Carlos Gomes) à direita**



Fonte: Junior (2017).

presente trabalho intenciona seguir não só como continuação das análises exploratórias feitas pelos pesquisadores que abriram este caminho com os dados abertos de transporte da URBS como também colaborar para uma análise sob outro viés do processamento destes dados.

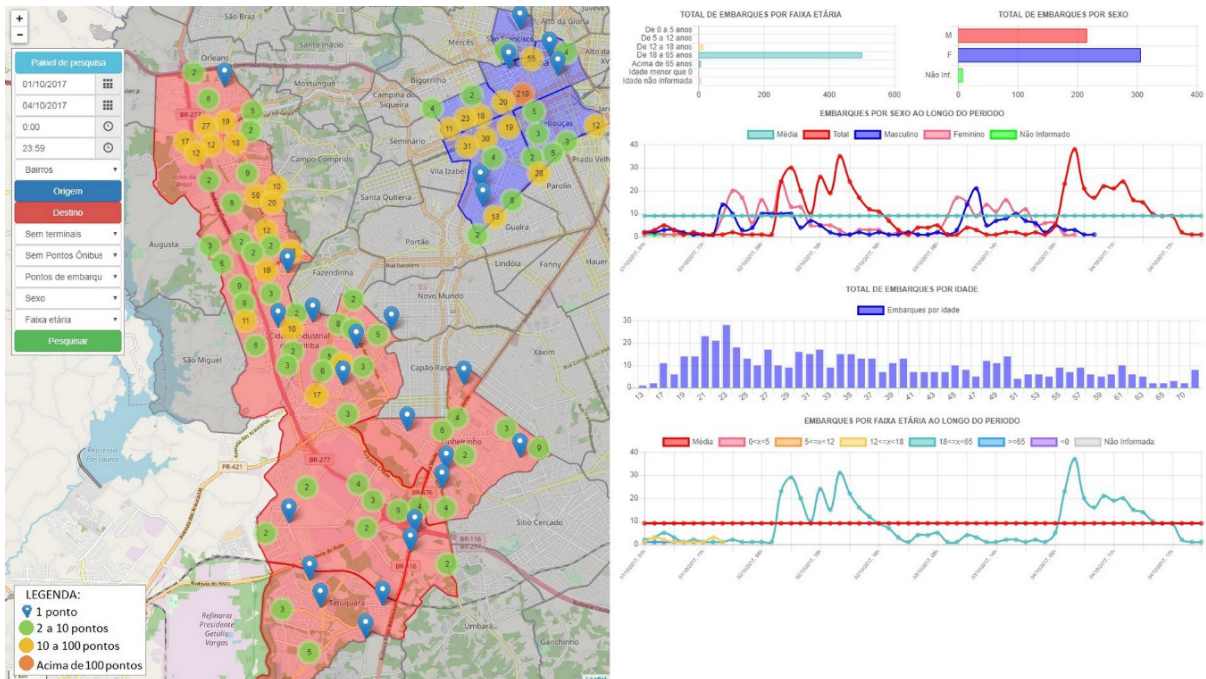
## 2.2 Sistemas de Informações Geográficas (SIGs)

Ao se analisar dados de ônibus que contém coordenadas geográficas, é preciso explicar seus tipos, como eles são compostos e quais operações neles podem ser feitas, tarefa apropriada para um SIGs (HAMADA; GONÇALVES, 2007; BORGES, 2002; DRUCK *et al.*, 2004).

Os Sistemas de Informações Geográficas (SIGs) são ferramentas que permitem o armazenamento, manipulação, processamento de dados comuns e dados com referências geográficas (pontos, linhas, polígonos, etc) como também definem um padrão de modelagem e linguagem comum para estas funções, comumente implementadas através de uma biblioteca de programação, um programa, ou um banco de dados (HAMADA; GONÇALVES, 2007).

Os SIGs podem ter diferentes significados dependendo do contexto em que são usados. As três faces de um SIG podem ser descritas da seguinte forma:

Figura 6 – Tela do *dashboard* de análise dos dados de ônibus



Fonte: Parcianello (2020).

- enquanto **ferramenta** - sendo um sistema que agrega várias ferramentas complementares de coleta, armazenamento, transformação e visualização tanto dados geográficos/geométricos como os convencionais dados de tipo numéricos e textuais (HAMADA; GONÇALVES, 2007).
- enquanto **dados** - sendo um sistema de banco de dados em que dados podem ser de tipos geométricos ou geográficos, contendo funções especiais que permitem fazer operações sobre estes tipos (HAMADA; GONÇALVES, 2007).
- enquanto **organização** - sendo um conjunto de funções automáticas que auxiliam profissionais interessados em armazenamento, recuperação, manipulação e exibição de dados geográficos (HAMADA; GONÇALVES, 2007).

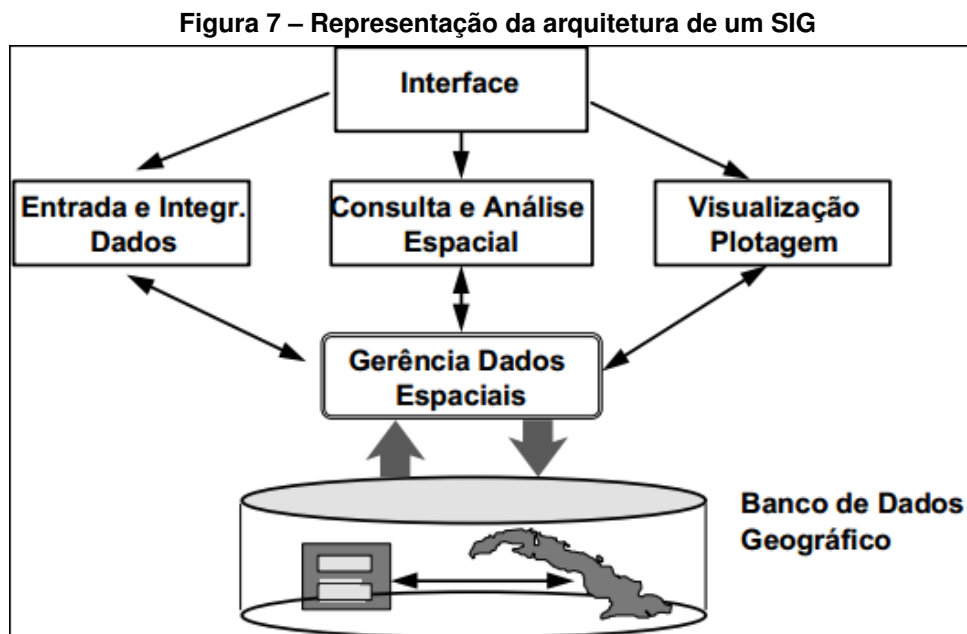
### 2.2.1 Arquitetura de funcionamento de um SIG

Segundo (DRUCK *et al.*, 2004), de forma abrangente, um SIG pode ser dividido em 5 componentes:

- Interface com o Usuário: Ocorre a interação humano-computador e nele são definidos como o sistema é operado e controlado.
- Entrada e Integração de Dados: Os dados são inseridos e organizados na forma adequada para alimentar o banco de dados.

- **Consulta e Análise Espacial:** São as funções e processamentos pré-definidos envolvidos nas consultas e análises espaciais. Cada caso terá suas funções específicas mas a título de exemplo tem-se o cálculo de uma rota, uma área, a quantidade de elementos presentes em tal área, etc.
- **Visualização e Plotagem:** Etapa em que a consulta espacial pré definida é criada é transformada em imagens para auxiliar na interpretação humana.
- **Armazenamento e Recuperação de Dados** (contidos em algum banco ou fonte de dados): Nível mais baixo da hierarquia, no qual os dados espaciais e complementares são gerenciados, organizados e enviados conforme a solicitação dos outros componentes.

Cada um destes componentes se inter-relaciona com o outro como demonstrado na Figura 7.



Fonte: Druck *et al.* (2004).

O modelo mais utilizado de bancos de dados geográficos é o modelo geo-relacional (ou arquitetura dual) (DRUCK *et al.*, 2004), em que há uma união do paradigma relacional dos bancos de dados tradicionais com os atributos/objetos geográficos e arquivos gráficos que armazenam as representações geométricas necessárias para as operações geográficas. Entretanto também existem bibliotecas de programação que podem empoderar softwares com tipos e operações geoespaciais como por exemplo JTS (Java Topology Suite) para Java<sup>16</sup>, GEOS (Geometry Engine Open Source) para C++<sup>17</sup>, Shapely para Python<sup>18</sup> e o TurfJS para Javascript NodeJS e browser<sup>19</sup>.

<sup>16</sup> Disponível em: <<https://locationtech.github.io/jts/>> Acesso em: 20 out, 2022.

<sup>17</sup> Disponível em: <<https://trac.osgeo.org/geos/>> Acesso em: 20 out, 2022.

<sup>18</sup> Disponível em: <<https://github.com/Toblerity/Shapely>> Acesso em: 20 out, 2022.

<sup>19</sup> Disponível em: <<https://turfjs.org/>> Acesso em: 20 out, 2022.

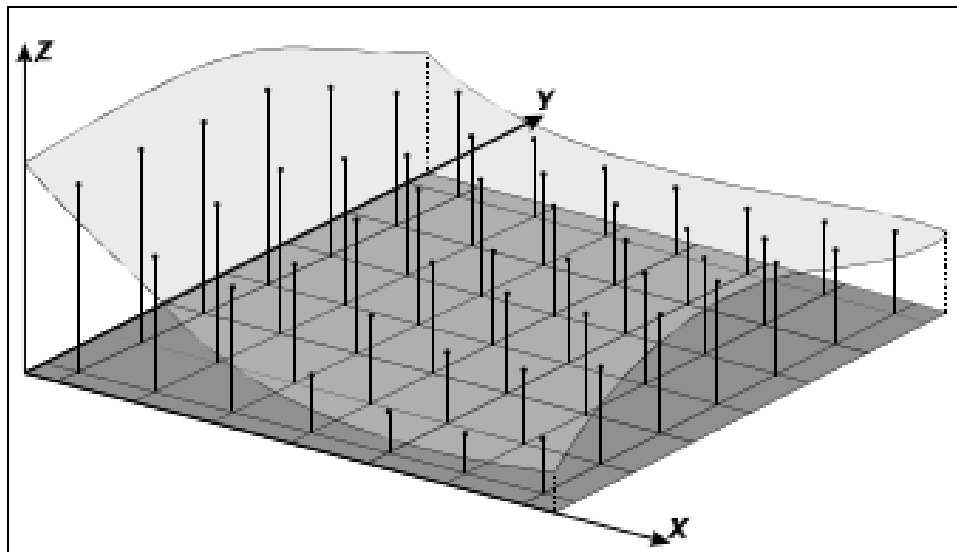


Dentre as representações geométricas padrão utilizadas nos SIGs têm-se:

- Pontos: Caracterizados por um par ordenado  $(x,y)$  de coordenadas espaciais (HAMADA; GONÇALVES, 2007; DRUCK *et al.*, 2004).
- Linhas e Polilinhas: Caracterizados por um n-upla de pontos ou pares ordenados  $(x,y)$  com uma ordem sequencial como por exemplo:  $\{(3,2),(3,5),(1,1),(2,3),(0,2),(1,3)\}$ , que podem representar vias, ruas, rios, divisões políticas, etc (HAMADA; GONÇALVES, 2007; BORGES, 2002).
- Polígonos: Indicados por um conjunto de n pares ordenados  $\{(a,b),(c,d),(x,y),\dots,(a,b)\}$  que graficamente formam diversas linhas interligadas. Nota-se que o último par ordenado deve ser igual ao primeiro, formando uma área fechada no plano geográfico. (HAMADA; GONÇALVES, 2007; DRUCK *et al.*, 2004).
- Amostras: Constituída pela tripla  $(x,y,z)$  onde  $(x,y)$  indica uma coordenada geográfica e  $(z)$  o valor do fenômeno estudado para tal coordenada. É usado em levantamentos de campo geofísicos, geoquímicos e oceanográficos para casos em que se têm múltiplas medidas em uma mesma localidade (DRUCK *et al.*, 2004).
- Grade regular: Caracterizada por uma matriz tridimensional em que cada elemento se associa à um número. Muito utilizada para descrever regiões da superfície terrestre com elevações e depressões, ilustradas na Figura 8 (DRUCK *et al.*, 2004; BORGES, 2002).
- Imagem: Composta por uma matriz onde geralmente cada elemento está associado a um valor numérico inteiro (um mapa de bits), e que pode ser usada em conjunto com uma grade regular ou um polígono, mostrando a textura, vegetação, áreas de precipitação ou o que a imagem indicar sobre a região (HAMADA; GONÇALVES, 2007; DRUCK *et al.*, 2004).

Dentro de um SIG há funções e métodos capazes de realizar operações com dados geográficos usando diferentes sistemas de coordenadas e projeções cartográficas. Uma das operações cartográficas mais comuns, o cálculo de distâncias entre dois pontos, pode ser realizado de diferentes formas com diferentes precisões. As formas mais comuns são: fórmula da distância Euclidiana, simplesmente definida pela distância compreendida por uma linha reta entre dois pontos (LAWHEAD, 2015); a fórmula de *Haversine*, que assume a terra como uma esfera, inclui a curvatura da esfera no cálculo da distância (ignorando as deformações naturais da terra) (PANIGRAHI, 2014); fórmula de Vincenty, que aproxima a forma da terra à um esferoide oblato, sendo uma das fórmulas mais precisas (LAWHEAD, 2015). A fórmula de *Haversine* é preferida em relação a distância Linear Euclidiana por considerar a curvatura da terra e ser relativamente leve de uma perspectiva de custo de processamento e relativamente precisa

**Figura 8 – Representação geométrica de uma Grade Regular**



Fonte: Druck *et al.* (2004).

(LAWHEAD, 2015). A fórmula de Vincenty é a mais precisa das três, mas também exige maior poder computacional para executá-la (LAWHEAD, 2015).

### 2.2.2 Map Matching

*Map Matching* são algoritmos usados para fixar dados de localização em uma rede espacial de estradas, podendo ter diversas aplicações, sendo a mais usual para dispositivos de GPS (PEREIRA; COSTA; PEREIRA, 2009). Os objetivos de um algoritmo de *Map Matching* é determinar a posição dada em um segmento de uma estrada em determinada rede e depois determinar a localização exata desta posição dentro de outro segmento selecionado previamente.

Ao analisar-se a movimentação de veículos com dados de GNSS (GPS) dentro de um SIG alguns problemas ocorrem, e são eles (MILLARD-BALL; HAMPSHIRE; WEINBERGER, 2019):

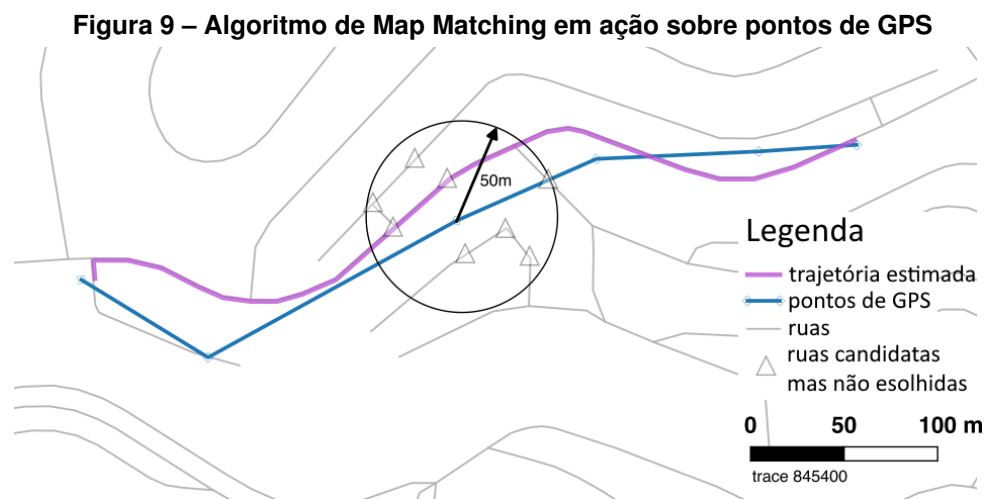
- Falta de dados, quando o veículo não salvou ou não enviou dados em uma frequência adequada para se ter um entendimento correto da sua movimentação geográfica.
- Imprecisões do dispositivo GNSS (GPS). Como toda ferramenta de medição, os dispositivos de GPS sofrem imprecisões e suas causas podem ser: objetos no caminho (prédios, montanhas), interferências eletromagnéticas, falhas de hardware, hardware de baixa qualidade, falhas nos satélites.

Considerando estes problemas é possível dizer que as leituras de GPS de um carro em movimento nem sempre serão registrados no lado correto da rua ou até mesmo dentro dos limites da rua. Isso dificulta alguns tipos de análises onde essas imprecisões podem atrapalhar. (MILLARD-BALL; HAMPSHIRE; WEINBERGER, 2019).

Um dos desafios para a detecção de passagem do ônibus em pontos de parada (com os dados abertos de ônibus da URBS) ocorre nos casos em que há duas paradas de ônibus de uma mesma linha em uma rua de mão dupla: uma parada no sentido "ida" e outra no sentido "volta". Uma forma de resolver este impasse é através do *Map Matching*.

Através da Figura 9 percebe-se a possibilidade de encaixar os pontos de leitura de GPS nas arestas reais de uma rua. Nela temos uma típica representação do problema; a linha azul mostra a polilinha gerada a partir dos pontos reais coletados pelo GPS; nota-se que ela ultrapassa os limites físicos das ruas (representadas pelas linhas pretas), percorrendo sobre canchais e possivelmente construções. Os algoritmos de *map matching* resolvem este problema estimando qual seria a trajetória, mais realista, do veículo dentro dos limites das ruas, também representada pela polilinha rosa.

Em outras palavras, os algoritmos de *Map Matching* são alimentados com uma lista de posições de GPS e mapa de ruas, devolvendo as posições de GPS ajustadas dentro daquela rua (PEREIRA; COSTA; PEREIRA, 2009).



Fonte: Modificado de Millard-Ball, Hampshire e Weinberger (2019).

O *Map Matching* pode ser realizado de diferentes formas por algoritmos diferentes. Alguns algoritmos funcionam em tempo real (*online*) enquanto outros funcionam em dados históricos (*offline*) ou em ambos. Os modelos por trás de um algoritmo de *Map Matching* podem ser diversos e difíceis de categorizar, entretanto alguns autores criaram um modelo para facilitar este trabalho (CHAO *et al.*, 2019), sendo:

**Modelos de Similaridade (*Similarity Model*):** Os modelos de similaridade escolhem o *matching* baseando-se na similaridade da polilinha real da rua com a polilinha gerada pelo GPS do veículo. Esse grau de similaridade pode ser medido de duas formas: baseado em distância e baseado em padrões (CHAO *et al.*, 2019).

**Modelos de Transição de Estado (*State-Transition Model*):** Dada uma posição do veículo no mapa cria-se um grafo com todas as possibilidades de rotas que o veículo provavelmente fez, onde os vértices são os estados (lugar onde o veículo provavelmente esteve em um

determinado intervalo de tempo) e as arestas as transições de um estado para outro (um trecho da rua para outro trecho da rua, no decorrer do tempo). Neste modelo tem-se 3 variações: *Hidden Markov Model* (HMM), *Conditional Random Field* (CRF), *Weighted Graph Technique* (WGT) (CHAO *et al.*, 2019).

**Modelo Candidato Evolutivo (*Candidate-Evolving Model*):** Quando inicia-se o *Map Matching* durante o primeiro ponto de GPS, inicia-se uma lista com o início dos possíveis trajetos (trajetos "candidatos"). Na medida que um novo ponto de GPS é adicionado, cria-se a continuação dos possíveis trajetos anteriores, baseando-se no novo ponto, dando um voto aos trajetos mais similares aos pontos de GPS ao mesmo tempo eliminando os trajetos que se tornaram irrelevantes (CHAO *et al.*, 2019). Uma vantagem deste com relação aos Modelos de Transição de Estado é a que não leva em consideração apenas a leitura de GPS anterior (estado anterior), mas todas as outras leituras e candidatos. Há dois tipos de variações desde modelo: Filtro de Partícula (*Particle Filter, PF*) e Múltiplas Hipóteses (*Multiple Hypothesis Technique, MHT*) (CHAO *et al.*, 2019).

**Modelo de Pontuação (*Scoring Model*):** Para cada ponto de GPS, define-se diferentes grupos de ruas que possivelmente representam o trajeto real. Então, para cada grupo de rua, calcula-se uma pontuação com base em uma função de pontuação. O grupo de rua que receber a maior pontuação (maximizar a função) é o caminho vencedor (CHAO *et al.*, 2019).

Além dos modelos e técnicas descritos na literatura também encontram-se inúmeros *softwares* prontos para realização de *Map Matchings*. Há uma boa variedade de *softwares* de código aberto oferecendo tal ferramenta, em sua maioria são utilizáveis de forma *blackbox*. Os mais conhecidos na comunidade <sup>20</sup> são: Mapzen Valhalla <sup>21</sup>; Open Source Routing Machine (OSRM) <sup>22</sup>; GraphHopper <sup>23</sup>; pgMapMatch <sup>24</sup> (MILLARD-BALL; HAMPSHIRE; WEINBERGER, 2019).

## 2.3 Cidades Inteligentes

Embora o conceito tecnológico de Cidades Inteligentes ser um dos mais utilizados não há consenso na literatura em relação ao significado do termo (CAVADA; ROGERS; HUNT, 2014; FERNANDEZ-ANEZ, 2016), pretende-se, para fins objetivos deste trabalho, utilizar-se da interpretação genérica descrita em (MENDES, 2020), no qual *smart cities* são cidades que, com o uso de recursos tecnológicos, buscam promover o crescimento inclusivo e sustentável de forma a maximizar a eficiência da distribuição destes recursos para melhor qualidade de vida da população.

<sup>20</sup> Disponível em: <<https://gis-ops.com/open-source-routing-engines-and-algorithms-an-overview/>> Acesso em: 20 out, 2022.

<sup>21</sup> Disponível em: <<https://github.com/valhalla/valhalla>> Acesso em: 20 out, 2022.

<sup>22</sup> Disponível em: <<http://project-osrm.org/>> Acesso em: 20 out, 2022.

<sup>23</sup> Disponível em: <<https://github.com/graphhopper/graphhopper>> Acesso em: 20 out, 2022.

<sup>24</sup> Disponível em: <<https://github.com/amillb/pgMapMatch>> Acesso em: 20 out, 2022.

**Figura 10 – Componentes de uma Smart City**



**Fonte: Mendes (2020).**

Para Mendes (2020) o conceito de Cidades Inteligentes é composto por seis pilares: Economia Inteligente, Governança Inteligente, Habitação Inteligente, Pessoas Inteligentes, Meio-Ambiente Inteligente e finalmente Mobilidade Inteligente (MENDES, 2020), cujo é o foco do presente trabalho, conforme representado pela Figura 10. A Mobilidade Inteligente contextualiza a importância em se ter inovação em sistemas de transporte com soluções modernas, sustentáveis e que atraiam a adesão da população, apoiada pela disponibilidade de infraestrutura de tecnologia da informação (GIFFINGER *et al.*, 2007). Dar mais transparência sobre o funcionamento do transporte público à população pode ser um meio de aumentar a adesão.

### 2.3.1 Sistemas de Transporte Inteligentes

Aliado ao conceito de Cidades Inteligentes, os Sistemas de Transporte Inteligentes (*Intelligent Transport Systems*, ITS) buscam melhorar a eficiência, segurança e conforto dos meios de transporte utilizando-se de softwares, computadores e sensores instalados em veículos, sinalizadores, e centrais de controle de tráfego. No escopo do transporte público, a inteligência por trás de um ITS busca facilitar a vida dos usuários com informações sobre tráfego, atualizações em tempo real sobre o serviço, disponibilidade de assentos no transporte desejado, etc.(MENEQUETTE; GRANDE; LOUREIRO, 2018)

Dentro do escopo de um ITS existem um grupo de softwares e dispositivos denominados Assistentes de Viagem Inteligentes (*Intelligent Travel Assistants*, ITA)(MENEQUETTE; GRANDE; LOUREIRO, 2018). Estes softwares servem para auxiliar seus usuários com informações durante o uso de algum tipo de transporte, seja carro, trem, ônibus, etc. Algumas de

suas funcionalidades podem contemplar: encontrar um menor caminho de um ponto até outro, ou no contexto do transporte público, descobrir quais ônibus podem ser usados para chegar-se mais rapidamente de um ponto A até um ponto B, informar quanto tempo se levará para chegar em um destino, informar quanto tempo o ônibus demorará pra chegar, informar sobre acidentes, congestionamentos, etc. Softwares servidos via website como os do Google Maps<sup>25</sup>, Moovit<sup>26</sup>, e ItiBus<sup>27</sup> são exemplos de ITAs que funcionam com os dados abertos providenciados pela URBS.

A equipe do *Google Maps* criou um serviço para prover informações históricas e em tempo real para empresas de transporte público mas em contrapartida os dados precisam estar no padrão GTFS (*General Transit Feed Specification*). Há um formato para consultas em tempo real, chamado GTFS *Realtime* e outro formato para consultas de dados estáticos chamado GTFS *Static* (WONG, 2013).

## 2.4 Desafios das Áreas Relacionadas

Dentre os desafios das áreas relacionadas, pode-se citar:

### 2.4.1 Desafios em Computação e Sistemas de Informações Geográficas

Com o crescente barateamento de sensores, dispositivos com GPS, *smartphones* e das redes móveis (como 3g, 4g e *low power Wide Area Network*), uma enorme quantidade de dados se torna disponível aos usuários, empresas privadas e instituições públicas. Automóveis contam com uma enorme quantidade de sensores; dispositivos GPS acoplados em automóveis, *smartphones* permitem rastrear sua localização geográfica no decorrer do tempo; *smartphones* permitem que pessoas se tornem sensores ambulantes (incluindo sons, imagens, movimentação, redes sem fio circundantes, temperatura e vários outros parâmetros ao seu redor); utensílios domésticos (geladeiras, lâmpadas, fornos elétricos, máquinas de café, despertadores, ar-condicionado, entre outros<sup>28</sup>), equipamentos urbanos (semáforos, radares de trânsito, iluminação pública, equipamentos meteorológicos) tem sido fabricados com conectividade em rede e componentes eletrônicos integrados, capazes de coletar, processar e transmitir dados, movimento também chamado de *Internet of Things* (IoT) (DAVENPORT, 2014). Nesta direção, vários desafios podem ser citados:

- **Processamento e armazenamento de grande quantidade de dados** - Dada a enorme quantidade de dados gerados por diferentes fontes de dados torna-se um grande desafio armazená-los, gerenciá-los e processá-los. Ferramentas e métodos

<sup>25</sup> Disponível em: <<https://maps.google.com/>> Acesso em: 20 out, 2022.

<sup>26</sup> Disponível em: <<https://moovitapp.com/>> Acesso em: 20 out, 2022.

<sup>27</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/mobile/itibus>> Acesso em: 20 out, 2022.

<sup>28</sup> Disponível em: <<http://iotlineup.com/>> Acesso em: 20 out, 2022.

bem consolidados, utilizados para tratar dados em escala menor, precisam ser, re-  
vistos, reestruturados para que seja possível operar dados em grande escala e de  
distintas fontes (DAVENPORT, 2014; KADIR *et al.*, 2018).

- **Ausência de fontes públicas de informações georreferenciadas** - É um problema comum em computação lidar com dados quantitativos numéricos, entretanto quando se entra na área de SIG percebe-se um desafio ligado à inexistente ou pouca disponibilização de dados georreferenciados públicos para análise (VILA, 2016).
- **Visualização em grande volume de dados georreferenciados** - Somando os desafios da grande quantidade de dados e de Sistemas de Informações Geográficas surge um novo desafio que é o da visualização de grandes volumes de dados georreferenciados. Para gerar visualizações e gráficos geográficos relevantes se faz necessário empregar técnicas para remediar a poluição visual causada pela grande quantidade de dados geográficos e reduzir o alto tempo de renderização das imagens, sendo algumas delas: clusterização, mapas de calor e *data binning* (JUNIOR, 2017; VILA, 2016; VILA *et al.*, 2016).

#### 2.4.2 Desafios em Dados Abertos

Dentro do escopo de Dados Abertos alguns desafios são descritos pela literatura, sendo eles:

- **Dados Incompletos e qualidade de dados** - Embora, em alguns casos, exista uma boa disponibilidade de dados abertos, não há garantia de que estes dados estejam completos ou corretos, análises exploratórias de dados abertos, descritos na literatura, frequentemente apontam problemas de incompletude e qualidade (KONO, 2016; PARCIANELLO, 2020; PARCIANELLO *et al.*, 2018; BELUZO, 2015).
- **Ausência de metadados e documentação adequada** - É essencial que documentações, metadados e explicações sejam disponibilizadas abertamente junto com os dados (KONO, 2016), permitindo a total compreensão sobre o que cada dado representa, de que forma foram coletados, em que momento, quais possíveis imprecisões, como decodificá-los com ferramentas gratuitas (abertas, acessíveis) e quais as relações com outros dados (AZAMBUJA, 2016; PARCIANELLO *et al.*, 2018).
- **Inconsistência de dados provenientes de diferentes fontes** - Mesmo quando há uma gama de dados abertos disponíveis, ocasionalmente percebe-se que dados provenientes de uma fonte entram em conflito com dados de outra (BELUZO, 2015; KONO, 2016). Um exemplo disso ocorre com os dados dos pontos de ônibus de Curitiba disponibilizados pela URBS em comparação com os providenciados pelo Instituto

de Pesquisa e Planejamento Urbano de Curitiba (IPPUC) e os presentes através da plataforma OpenStreetMap<sup>29</sup>, onde as coordenadas geográficas de pontos de ônibus de mesmo código acabam sendo diferentes (PARCIANELLO *et al.*, 2018).

### 2.4.3 Desafios em Cidades Inteligentes

Com a crescente urbanização das cidades e surgimento de metrópoles alguns problemas começam a surgir, principalmente em desenvolvimentos urbanos rápidos e mal planejados (SCHRANK; LOMAX; EISELE, 2011), e eles se tornam grandes temas de discussão dentro da temática de Cidades Inteligentes. São eles:

- **Superpopulação e alta densidade populacional** - Quando o número de indivíduos em um mesmo território danifica o meio ambiente mais rapidamente do que este pode ser reparado pela natureza, possivelmente acarretando um colapso ecológico e social (YIN *et al.*, 2015; DIRKS; KEELING, 2009; SCHRANK; LOMAX; EISELE, 2011; HÖJER; WANGEL, 2015) - é uma das fontes causadoras dos próximos desafios da área, e intimamente relacionado com a busca por sustentabilidade e desenvolvimento sustentável;
- **Poluição e degradação ambiental** - Há um grande desafio e necessidade imediata para reduzir a poluição do ar e disponibilizar água limpa nos centros urbanos, situação geralmente desencadeada pelo aumento da densidade populacional (YIN *et al.*, 2015). Com o aumento do número de veículos movidos a combustíveis fósseis e indústrias poluentes, naturalmente a qualidade do ar será degradada (YIN *et al.*, 2015; CAVADA; ROGERS; HUNT, 2014; ALBINO; BERARDI; DANGELICO, 2015; AZAMBUJA, 2016). Empresas tem vendido sensores de poluição PM 2.5, PM 10, CO2 para uso consumidor e profissional, coletando dados e disponibilizando-os ao público pela internet<sup>30,31,32</sup>;
- **Criminalidade e segurança** - Alguns estudos mostraram indícios de que a crescente urbanização seja um fator dominante no crescimento da criminalidade (HALICIOGLU; ANDRÉS; YAMAMURA, 2012; YIN *et al.*, 2015; AZAMBUJA, 2016);
- **Problemas de transporte e tráfego** - Com o aumento crescente da quantidade de carros em trânsito em uma cidade, problemas de congestionamento se tornam mais frequentes, aumenta-se o tempo em que os passageiros ficam ociosos em seus automóveis. Um estudo de 2011 nos Estados Unidos da América documentou que 493 regiões em cidades passavam por constante engarrafamentos, o que custou à população 4,8 bilhões de horas a mais em veículos de transporte a um custo extra de 1,9

<sup>29</sup> Disponível em: <<https://www.openstreetmap.org/>> Acesso em: 20 out, 2022.

<sup>30</sup> Disponível em: <<https://www2.purpleair.com/>> Acesso em: 20 out, 2022.

<sup>31</sup> Disponível em: <<https://air.plumelabs.com/en/>> Acesso em: 20 out, 2022.

<sup>32</sup> Disponível em: <<https://www.iqair.com/air-quality-map>> Acesso em: 20 out, 2022.



bilhões de galões de combustível, a um custo de 101 bilhões de dólares (YIN *et al.*, 2015; SCHRANK; LOMAX; EISELE, 2011).

Todos estes desafios e problemas, discutidos no âmbito de Cidades Inteligentes, chamam atenção do setor público, privado e comunidade para buscar formas tecnológicas de reduzi-los, trazendo uma melhor qualidade de vida à população.

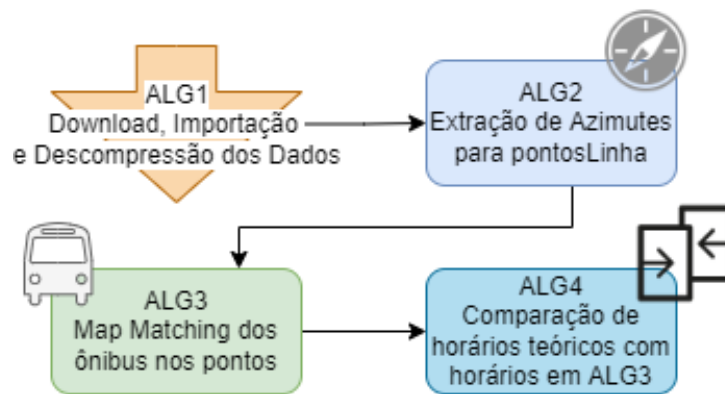
Este trabalho inspirou-se em Junior (2017), Parcianello (2020), Parcianello *et al.* (2021). Eles buscam de forma criativa, através de ferramentas tecnológicas, extrair informações interessantes nunca antes exploradas nos dados brutos e abertos do transporte público de Curitiba operado pela URBS. Estes trabalhos revelam que, nestes dados, há uma gama de informações ocultas, mas não serão reveladas sem um esforço investigativo, técnico e curioso para explorá-los e deles extrair valor para os gestores do transporte público, comunidade acadêmica e população.

### 3 METODOLOGIA E DESENVOLVIMENTO

Nesta seção é apresentada a metodologia e o desenvolvimento. Primeiro são apresentados detalhes sobre o funcionamento dos ônibus, os dados abertos disponibilizados, seguido pelos algoritmos utilizados para tratar e processar os dados mencionados.

Para que os resultados fossem atingidos dividiu-se o problema em 4 algoritmos diferentes, conforme apresentado na Figura 11. De forma resumida, os algoritmos são responsáveis por baixar os dados, limpá-los, enriquecê-los, explorá-los para extrair horários estimados de passagem e compará-los com horários teóricos de passagem.

**Figura 11 – Sequência de execução dos algoritmos**



**Fonte: Autoria própria (2022).**

Para a análise e desenvolvimento da solução utilizou-se o banco de dados Geo-Relacional PostgreSQL com extensão PostGIS, a máquina virtual de Javascript V8 e NodeJS para importação e processamento de dados, o sistema de informação geográfico para visualização QGIS, e o *software* de *business intelligence* Metabase para visualização de dados numéricos, categóricos e temporais.

O foco deste trabalho são dados históricos entre 01/05/2019 e 14/05/2019 para a linha Cabral Portão de código 216. A escolha deste período se deu por corresponder a duas semanas completas em um mês que não é normalmente de férias e em um período típico de trabalho do ano. Decidiu-se não escolher uma data mais recente de dados por conta das particularidades e redução de frota causadas pela pandemia do COVID-19 no transporte público de Curitiba. A linha Cabral Portão foi escolhida por ser uma linha antiga, de fluxo de passageiros relativamente alto (contendo ônibus articulados) e que opera em pontos de ônibus em ruas comuns, compartilhadas com os demais veículos (carros e motos), em contraste com o BRT. Este período foi escolhido por conter duas semanas completas de dados, um dia de feriado, 2 finais de semana (sábado e domingo) e 9 dias úteis, possibilitando compreender as particularidades dos diferentes tipos de dias.

Para a execução dos algoritmos, banco de dados, importação e de todo o sistema de maneira geral utilizou-se um notebook com processador i7-4710HQ, memória RAM 16GB 1600MHz e um SSD mSATA Samsung 840 EVO, 500GB.

### 3.1 O Funcionamento dos Ônibus

Dentro da frota da URBS encontram-se 10 modelos diferentes de veículos, cada um com distintas capacidades de passageiros, tamanho do veículo, presença de articulações, forma de abertura das portas, cor, entre outras. Cada um destes modelos e suas respectivas capacidades estão presentes na Tabela 1. Além dos modelos há 10 categorias de linhas de ônibus diferentes<sup>1</sup>, cada uma atendendo demandas diferentes no trânsito da cidade, algumas trafegam apenas por ruas exclusivas para ônibus (chamadas de *Bus Rapid Transit* - BRT), outro por vias normais juntamente com os carros, outros apenas pelos bairros centrais, outros circundam apenas bairros periféricos.

**Tabela 1 – Tipos de ônibus e suas capacidades**

<b>Tipo do Ônibus</b>	<b>Capacidade (passageiros)</b>
Micro	47
Microespecial	65
Comum	87
Padron	92
Padron Linha Direta	95
Articulado (18,6 m)	138
Articulado Linha Direta (18,6 m)	150
Articulado Expresso (20 m)	180
Biarticulado (24,5 m)	224
Biarticulado (28 m)	250

**Fonte: Aatoria própria (2022).**

A frota de ônibus da URBS é dividida em 9 categorias de linhas, descritas pela Figura 12 e detalhadas a seguir:

- **Expresso Ligeirão**, sua frota é composta por veículos biarticulados nas cores azul ou vermelha, em canaletas exclusivas, com número reduzido de paradas e por consequência proporcionando deslocamentos mais rápidos em comparação às outras categorias. Tanto embarque como desembarque são feitos em plataformas elevadas em terminais e estações-tubo. Fazem parte do *Bus Rapid Transit* (BRT) da cidade.
- **Expresso**, tem em sua composição veículos tipo biarticulados, de cor vermelha, que ligam os terminais (grandes estações de ônibus que agregam linhas diferentes) ao

<sup>1</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte/33>> Acesso em: 20 out, 2022.

















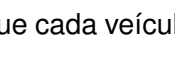



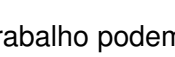
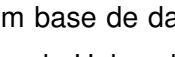
centro da cidade, através das canaletas exclusivas. Embarques e desembarques são feitos em nível nos terminais e nas estações-tubo existentes no trajeto. Também fazem parte do *Bus Rapid Transit* (BRT) da cidade.

- **Linha Direta (Ligeirinho)**, tem em sua frota veículos nas cores prata ou cinza, com paradas em média a cada 3 km, com embarque e desembarque em nível nas estações-tubo. São linhas complementares, principalmente das linhas expressas e interbairros. Também fazem parte do *Bus Rapid Transit* (BRT) da cidade.
- **Interbairros**, são operados por veículos tipo padron ou articulados, na cor verde, que ligam os diversos bairros e terminais sem passar pelo Centro da cidade.
- **Alimentador**, utiliza veículos tipo micro, comum ou articulados, nas cores laranja ou amarela, que ligam os terminais de integração aos bairros de cada região.
- **Troncal**, trabalha com veículos tipo articulados, nas cores amarela ou laranja, que ligam os terminais de integração ao Centro da cidade, utilizando vias compartilhadas.
- **Circular Centro (Linha Especial)**, operada com veículos tipo micro-ônibus, atende os principais pontos atrativos da região central de Curitiba, tais como praças, shoppings, Rodoviária e Biblioteca Pública.
- **Convencional (Linha Especial)**, composta por veículos tipo micro ou comum, nas cores amarela ou laranja, que ligam os bairros ao Centro da cidade, sem integração.
- **Linha Turismo (Linha Especial)**, usa veículos de dois andares, diferentes de todas as outras linhas por ser um ônibus voltado ao turismo na cidade, passa pelos principais parques, pontos turísticos, portanto sua tarifa é mais elevada.

Cada veículo é identificado por um código único (JL305, KA602, LL300, LR802 ...) e é capaz de operar em uma ou mais linhas de ônibus. Cada linha, que também conta seu código único (462, 532, X20, 915, 521), dita o caminho que os veículos devem fazer, quais pontos de ônibus precisam visitar e em qual ordem. Cada linha é formada por um ou mais trechos de trajetos, cada um com nomes que geralmente se referem ao ponto de ônibus final do trecho. É grande a quantidade de linhas com exatamente dois trechos (um de ida e outro de volta), formando uma ligação entre diferentes terminais de ônibus. Os terminais de ônibus são grandes estações, localizadas em pontos estratégicos da cidade, que agregam vários pontos de ônibus em um só lugar. Durante cada trecho de trajeto existem os pontos de ônibus, que devem ser visitados sempre na mesma ordem pré estabelecida. Uma visualização completa da linha 216 (Cabral Portão) é mostrada na Figura 13; em azul há o trecho sentido terminal Cabral, em vermelho o trecho sentido terminal Portão e as placas em preto todos os pontos de ônibus.

Em um mesmo dia, cada linha de ônibus é operada por diferentes veículos, cada um iniciando seus trechos de trajeto em horários diferentes e com um cronograma a ser cumprido.

Figura 12 – Composição da frota de ônibus da URBS

Categoria de Linhas	Tipos de Veículo	Capacidade dos Veículos	Frota Operante Subtotal	Total	Quantidade de Linhas
EXPRESSO LEIGIRÃO	 BIARTICULADO	250	44	44	03
					
EXPRESSO	 BIARTICULADO	230/250	97	128	05
	 ARTICULADO	170	31		
LINHA DIRETA	 ARTICULADO	150	38	219	15
	 PADRON	110	181		
INTERBAIROS	 ARTICULADO	140	91	102	08
	 PADRON	100	1		
	 HÍBRIDO	79	10		
ALIMENTADOR	 ARTICULADO	140	71	425	129
	 COMUM	85	325		
	 MICRO ESPECIAL	70	29		
TRONCAL	 ARTICULADO	140	5	77	15
	 COMUM	85	59		
	 HÍBRIDO	79	10		
	 MICRO ESPECIAL	70	3		
CONVENCIONAL	 COMUM	85	102	217	74
	 HÍBRIDO	79	10		
	 MICRO ESPECIAL	70	102		
	 MICRO	40	3		
CIRCULAR	 MICRO	40	5	5	01
TURISMO	 DOUBLE-DECK	65	12	12	01
<b>TOTAL</b>			<b>1.229</b>	<b>251</b>	

Fonte: Urbanização de Curitiba (2021).

A URBS define antecipadamente que cada veículo deve passar em certos pontos de ônibus em horários pré estabelecidos<sup>2</sup>.

### 3.2 Sobre os Dados Abertos da URBS

Os dados utilizados neste trabalho podem ser encontrados através da página de dados abertos da Prefeitura de Curitiba em base de dados do “Transporte Coletivo de Curitiba”<sup>3</sup>. Os arquivos dos dados<sup>4</sup> são mantidos pela Universidade Federal do Paraná através do Centro de Computação Científica e Software Livre (C3SL), que armazena *backups* diários de 8 tabelas utilizadas pela URBS, e há um arquivo por dia para cada tabela.

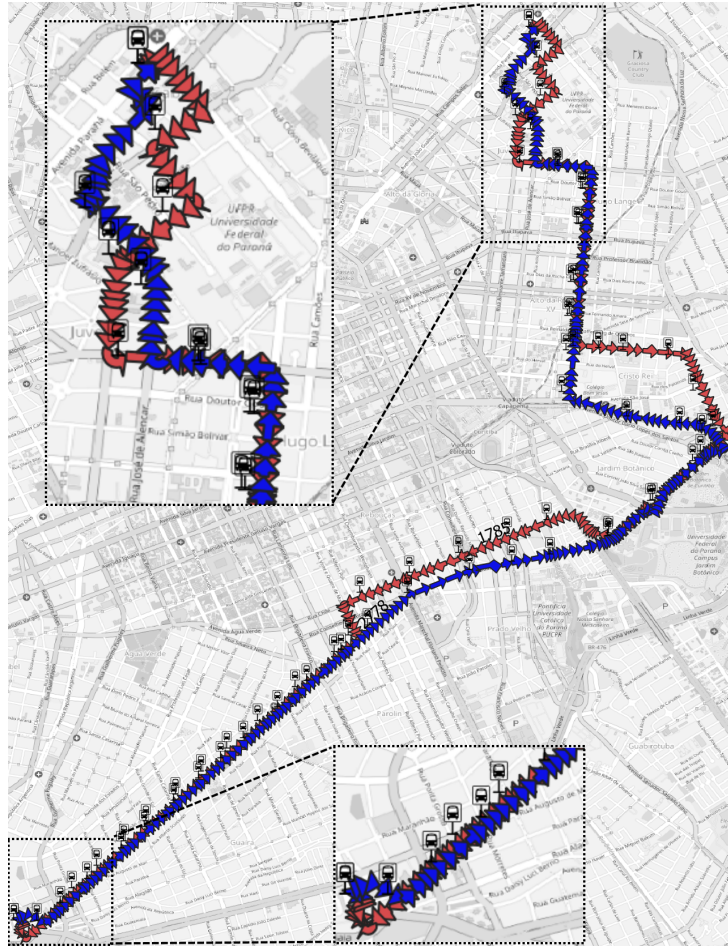
As tabelas disponíveis são: “linhas”, “pontosLinha”, “shapeLinha”, “tabelaLinha”, “tabelaVeiculo”, “veiculos”, “pois” e “trechosItinerarios”, conforme presente na Tabela 2. Os arquivos “pois” e “trechosItinerarios” não foram utilizados.

<sup>2</sup> Disponível em: <<https://www.urbs.curitiba.pr.gov.br/horario-de-onibus/216>> Acesso em: 20 out, 2022.

<sup>3</sup> Disponível em: <<https://www.curitiba.pr.gov.br/dadosabertos/busca/?pagina=9>> Acesso em: 20 out, 2022.

<sup>4</sup> Disponível em: <<http://dadosabertos.c3sl.ufpr.br/curitibaurbs/>> Acesso em: 20 out, 2022.

**Figura 13 – Visualização dos trechos de trajeto da linha 216 (Cabral/Portão) no dia 18/04/2019 através do software QGIS**



**Fonte: Autoria própria (2022).**

**Tabela 2 – Descrição dos arquivos/tabelas disponibilizadas pela URBS e relação dos quais foram analisados e utilizados neste trabalho**

Nome do arquivo	Descrição	Analisado?
linhas	dados sobre todas as linhas de ônibus	✓
pontosLinha	dados sobre todos pontos de ônibus	✓
shapeLinha	polilinhas que representam cada segmento trajeto das linhas de ônibus	✓
tabelaLinha	horários programados de chegada dos ônibus nos pontos de ônibus	✓
tabelaVeiculo	horários programados de chegada dos ônibus nos pontos de ônibus	✓
veiculos	trajetória espaço-temporal dos ônibus	✓
pois	pontos de interesse na cidade	X
trechosItinerarios	dados sobre os pontos de ônibus	X

**Fonte: Autoria própria (2022).**

Nos próximos parágrafos são explicados cada um destes 6 arquivos/tabelas, seu conteúdo (suas colunas), tipos de dados, 3 suas inter-relações.

A primeira tabela é a “linhas”, ela descreve as propriedades de cada linha de ônibus tipo a cor de seus ônibus e a categoria que se encaixa, suas colunas são descritas na Tabela 3.

**Tabela 3 – Descrição dos dados contidos na tabela “linhas”**

Coluna	Novo nome	Tipo	Significado
COD	bus_line_id	texto	código da linha de ônibus
NOME	bus_line_name	texto	nome da linha de ônibus
NOME_COR	color	texto	cor dos ônibus desta linha
SOMENTE_CARTAO	card_only	texto (S ou N)	se o ônibus aceita apenas cartão
CATEGORIA_SERVICO	bus_line_category	texto	categoria de linha, as mesmas descritas na Figura 12

**Fonte: Autoria própria (2022).**

A segunda tabela relevante é a “pontos linha”, ela lista todos os pontos de ônibus presentes em cada linha. As propriedades da tabela estão descritas na Tabela 4.

**Tabela 4 – Descrição dos dados contidos no arquivo “pontosLinha”**

Coluna	Novo nome	Tipo	Significado
COD	bus_line_id	texto	código da linha de ônibus
NUM	bus_stop_id	número inteiro	código do ponto de ônibus
NOME	bus_stop_name	texto	nome do ponto de ônibus
LAT	lat	número real	se o ônibus aceita apenas cartão
LON	lon	número real	longitude da amostra espaço-temporal
SENTIDO	way	texto	o nome do trecho de trajeto (geralmente o nome do terminal de destino)
SEQ	seq	número inteiro	a ordem de passagem deste ponto dentro do trecho de trajeto (refere-se ao SENTIDO)
TIPO	bus_stop_type	texto	o tipo de cobertura do ponto de ônibus
GRUPO	bus_stop_group	número inteiro	código compartilhado entre outros pontos de ônibus que podem ser utilizados gratuitamente a partir deste

**Fonte: Autoria própria (2022).**

Por terceiro há a “shape linha”, que contém os caminhos exatos que os ônibus devem fazer em cada trecho de trajeto de uma respectiva linha. Suas colunas são descritas pela Tabela 5.

Em quarto, a “tabelaLinha”, que contém o cronograma com os horários em que todos os ônibus de cada linha devem passar em seus respectivos pontos de ônibus, também chamados de horários teóricos, tabelados ou programados. A descrição dos dados contidos no arquivo estão na Tabela 6. Linhas que caem na categoria BRT (Expresso Ligeirão, Expresso, Linha Direta) contam com horário tabelado em todos seus pontos, entretanto isto não é válido para as demais linhas.

**Tabela 5 – Descrição dos dados contidos no arquivo “shapeLinha”**

Coluna	Novo nome	Tipo	Significado
ÍNDICE	id	número inteiro	a ordem deste ponto geográfico no arquivo
COD	bus_line_id	texto	código da linha de ônibus
SHP	path_id	número inteiro	o código do trecho de trajeto
LAT	lat	número real	a latitude deste ponto no trajeto
LON	lon	número real	a longitude deste ponto no trajeto

**Fonte: Autoria própria (2022).**

**Tabela 6 – Descrição dos dados contidos na tabela “tabelaLinha”**

Coluna	Novo nome	Tipo	Significado
COD	bus_line_id	texto	código da linha de ônibus
TABELA	schedule_id	texto	o código da tabela horária que este veículo está operado neste dia
DIA	day_category	texto	categoria de dia, onde 1 = dia útil, 2 = sábado, 3 = domingo, 4 = feriado
HORA	time	hora	horário (no horário oficial de Brasília) de passagem do ônibus neste ponto
NUM	bus_stop_id	número inteiro	código do ponto de ônibus
PONTO	bus_stop_name	texto	nome do ponto de ônibus
ADAPT	wheelchair_accessibility_type	texto	se o ponto de ônibus contém plataforma de acesso para cadeirantes

**Fonte: Autoria própria (2022).**

Em quinto, a “tabelaVeiculos”, é muito semelhante à “tabelaLinha”, mencionada anteriormente, que também descreve o horário tabelado (teórico, programado) que cada ônibus deve desempenhar em determinada linha. Entretanto, nela cada veículo é identificado explicitamente por seu código. Conseqüentemente é possível saber, através dela, em que momento cada veículo deveria passar por certos pontos de ônibus. A descrição de suas colunas está presente na Tabela 7.

Por último, o arquivo “veiculos” apresenta uma série temporal da posição geográfica de cada ônibus (em média em amostras a cada 20 segundos) e qual linha ele está operando. Suas colunas são descritas na Tabela 8.

Na próxima seção é apresentado o Algoritmo 1 para o download, importação e descompressão dos dados.

### 3.3 Algoritmo 1 - Importação dos dados da URBS

Nesta etapa foi criado um script na linguagem *Typescript* através do *runtime* de *Javascript NodeJS*, o qual recebe como entrada uma lista de datas (tipo 05/05/2019, 06/05/2019...),



Tabela 7 – Descrição dos dados contidos na tabela “tabelaVeiculo”

Coluna	Novo nome	Tipo	Significado
COD_LINHA	bus_line_id	texto	código da linha de ônibus
NOME_LINHA	bus_line_name	texto	nome da linha de ônibus
VEICULO	vehicle_id	texto	código do veículo
COD_PONTO	bus_stop_id	número inteiro	código do ponto de ônibus
HORARIO	time	hora	horário (no horário oficial de Brasília) de passagem do ônibus neste ponto
TABELA	schedule_id	texto	o código da tabela horária que este veículo está operando neste dia

Fonte: Autoria própria (2022).

Tabela 8 – Descrição dos dados contidos na tabela “veiculos”

Coluna	Novo nome	Tipo	Significado
VEIC	vehicle_id	texto	código do veículo
COD_LINHA	bus_line_id	texto	código da linha de ônibus
DTHR	timestamp	data e horário	data e horário da amostra espaço-temporal
LAT	lat	número real	latitude da amostra espaço-temporal
LON	lon	número real	longitude da amostra espaço-temporal

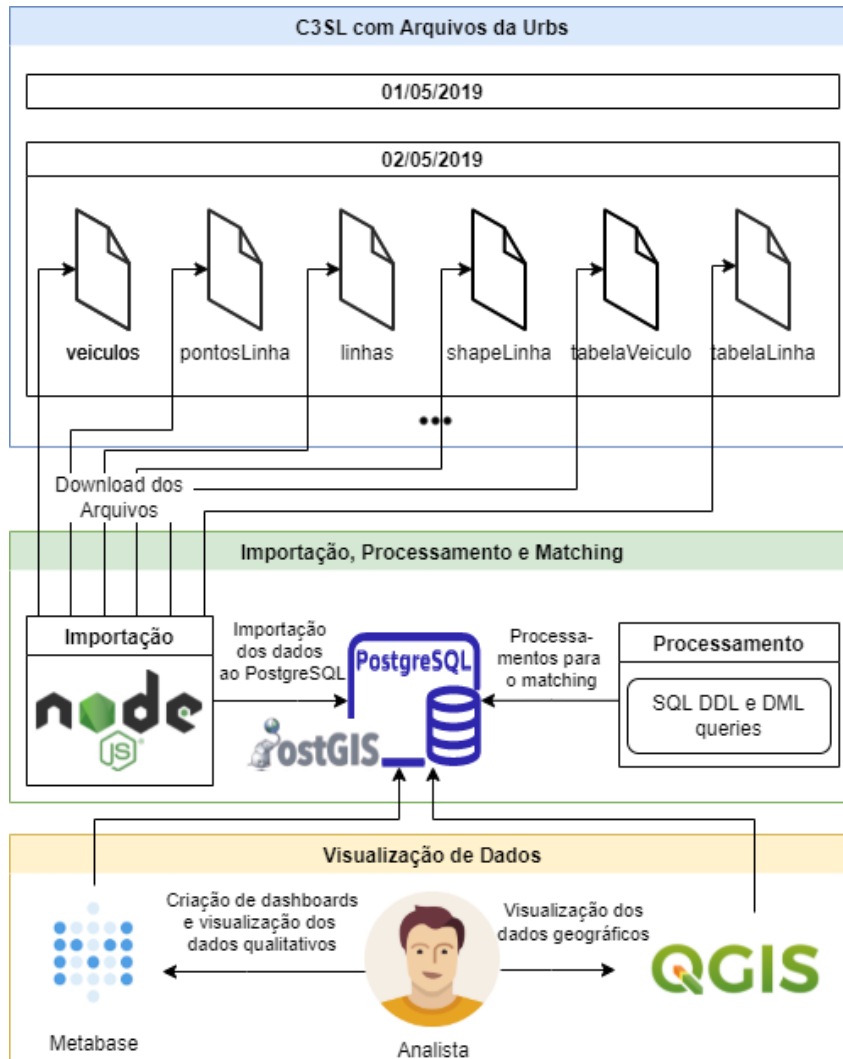
Fonte: Autoria própria (2022).

realiza o download dos arquivos mencionados anteriormente, processa-os e os armazena em tabelas de um banco de dados PostgreSQL. Uma visualização do processo de importação é representada na Figura 14. O passo a passo é descrito abaixo:

- solicita ao usuário as datas desejadas para importação;
- realiza o download dos 6 arquivos relevantes (*veiculos*, *pontosLinha*, *linhas*, *shapeLinha*, *tabelaVeiculo*, *tabelaLinha*) para as datas desejadas;
- efetua a descompressão de cada um com o algoritmo LZMA;
- realiza o *parse* e desserialização de seu conteúdo descomprimido, transformando-os em dados entendíveis e manipuláveis pelo *Typescript*;
- efetua a limpeza e conversão de dados (transformando alguns dados de texto em datas ou números e latitude/longitude em coordenadas geográficas WGS84);
- conecta-se a um banco de dados PostgreSQL + PostGIS, executa-se comandos de criação de tabelas (uma por arquivo) e inserção dos dados processados.

Para o arquivo “veiculos” o método de importação é diferente dos outros, uma vez que seu arquivo contém objetos JSON separados por quebras de linha e que seu tamanho descomprimido pode ter em média 500 megabytes. Um filtro é executado antes da inserção dos dados no banco, salvando apenas aquelas linhas de ônibus desejadas.

**Figura 14 – Representação da arquitetura da aplicação desenvolvida**



Fonte: Autoria própria (2022).

### 3.3.1 Inconsistências, Redundâncias e Peculiaridades dos Dados

Os dados abertos disponibilizados pela URBS contam com algumas inconsistências, redundâncias, peculiaridades que serão abordadas nesta seção. As seguintes análises foram feitas do dia 01/05/2019 até 14/05/2019 para a linha 216 (Cabral/Portão).

### 3.3.1.1 Arquivos vazios e com dados faltantes

A tabela “veiculos” apresenta falta de dados perceptíveis. De 01/05/2019 até 15/05/2019, quatro dias (dias 01, 04, 09, 10) apresentaram ausência total de dados e um dia (dia 08) mostrou amostras espaço-temporais de apenas um ônibus, ambos para a linha 216. A Tabela 9 mostra quantidade de veículos da linha 216 presentes em cada arquivo e também sua relação de amostras espaço-temporais (quantas linhas estão presentes no arquivo).

**Tabela 9 – Arquivos “veiculos” e análise de quantidade de veículos e amostras espaço-temporais na linha 216**

<b>Dia do Arquivo</b>	<b>Quantidade de Veículos</b>	<b>Quantidade de Amostras</b>
2019-05-01	0	0
2019-05-02	4	17647
2019-05-03	13	36213
2019-05-04	0	0
2019-05-05	6	26676
2019-05-06	4	18782
2019-05-07	16	49430
2019-05-08	1	1052
2019-05-09	0	0
2019-05-10	0	0
2019-05-11	3	13349
2019-05-12	6	26465
2019-05-13	3	18428
2019-05-14	16	47796

**Fonte: Autoria própria (2022).**

### 3.3.1.2 A qual dia o arquivo “veiculo” se refere?

Os dados de tempo presentes nas amostras espaço-temporais da tabela “veiculo”, em sua maioria, têm problemas de correspondência com a data contida no nome do arquivo. Dentro do período analisado, a maioria dos dias apresentou dados do dia anterior ao arquivo, enquanto dois apresentaram dia igual. Nos arquivos do dia 11 e 12, as amostras de séries temporais são todas registradas no dia 11. As amostras estão descritas na Tabela 10. Em negrito identifica-se os arquivos que fogem do padrão, onde a data das amostras é igual a data do arquivo.

### 3.3.1.3 Quantidade de veículos em operação diferem dos programados

Ao extrair os identificadores dos veículos para a linha 216 contidos em “veiculos” e compará-los com os existentes em “tabelaVeiculo”, em cada respectiva data, criou-se a Tabela 11. Pode-se perceber que na maioria dos casos houve mais veículos em operação que o programado. Os dois casos em que se verifica o contrário (em que há mais veículos programados

**Tabela 10 – Arquivos “veículos” e comparação das datas nele contidas com a data do arquivo para a linha 216**

<b>Data do Arquivo</b>	<b>Data das Amostras</b>	<b>Veículos em Operação</b>
2019-05-01		
2019-05-02	2019-05-01	DR406, DR408, HA617, HI604
2019-05-03	2019-05-02	BA125, BA131, BA601, BA604, CA600, CA601, CA602, CA603, DA300, DR400, DR402, DR403, DR405
2019-05-04		
2019-05-05	2019-05-04	BA600, BA601, BA602, CA601, CA603, DR405
2019-05-06	2019-05-05	DR406, DR408, HA615, HA616
2019-05-07	2019-05-06	BA600, BA601, BA602, BA604, CA300, CA601, CA602, CA603, DR401, DR403, DR405, DR406, HA601, HA606, HI602, HI603
<b>2019-05-08</b>	<b>2019-05-08</b>	BA121
2019-05-09		
2019-05-10		
<b>2019-05-11</b>	<b>2019-05-11</b>	BA600, BA601, BA602
2019-05-12	2019-05-11	BA600, BA601, BA602, CA601, CA603, DR405
2019-05-13	2019-05-12	DR405, HI602, HI603
2019-05-14	2019-05-13	BA136, BA601, BA602, BI600, CA012, CA600, CA601, CA602, CA603, DR403, DR405, DR406, HA601, HA605, HA606, HA611

**Fonte: Autoria própria (2022).**

do que reais) ocorrem justamente onde a data das amostras foram iguais às datas dos arquivos, um indicativo de que os dados em “veículos” sofreram algum tipo de inconsistência nestes dias específicos.

Outro detalhe é o de que os veículos em operação não são necessariamente os mesmos presentes no planejamento - a coluna “Veículos em comum” da Tabela 11 mostra esta comparação. Nota-se que tanto no dia 8 quanto no dia 11, dias em que a data das amostras “veículos” foram iguais à do arquivo houve poucos veículos em comum em comparação com os outros dias.

#### 3.3.1.4 Pontos de ônibus não fazem parte dos pontos de ônibus comuns ao trajeto da linha

A “tabelaLinha” é usada para mostrar horários programados (teóricos) de chegada nos pontos de ônibus, entretanto existem nela horários cadastrados para pontos de ônibus não existentes na linha, por exemplo, a linha 216 (para o dia 01/05/2019) contém registros de horários programados para o “Terminal Pinheirinho” e “Bairro Pompeia”.

**Tabela 11 – Comparação dos veículos contidos em “veiculos” e “tabelaVeiculo” na linha 216**

<b>Data</b>	<b>Veículos em “veiculos”</b>	<b>Veículos em “tabelaVeiculo”</b>	<b>Veículos em comum</b>
2019-05-01	4	3	
2019-05-02	13	13	8
2019-05-04	6	6	5
2019-05-05	4	3	
2019-05-06	16	13	10
<b>2019-05-08</b>	1	13	
<b>2019-05-11</b>	3	6	2
2019-05-11	6	6	5
2019-05-12	3	3	3
2019-05-13	16	13	8

**Fonte: Autoria própria (2022).**

### 3.3.1.5 Quantidade de veículos em “tabelaVeiculo” é diferente de “tabelaLinha”

Comparando a quantidade de veículos diferentes, em um mesmo dia, para os arquivos “tabelaVeiculo” (coluna VEICULO) e “tabelaLinha” (coluna TABELA) percebe-se que também há inconsistências: esta comparação está presente na Tabela 12. Uma análise rápida sobre a tabela mostra um padrão: sempre quando há 13 veículos em “tabelaVeiculo” ocorrem 18 em “tabelaLinha”.

Além desta inconsistência, percebe-se que no dia 01/05/2019, um feriado nacional (“DIA” código 4), não houve veículos programados para operar de acordo com “tabelaLinha”. Alternativamente um teste mostrou que ao considerar a tabela de domingo (“DIA” código 3) para este dia de feriado a quantidade de veículos se torna igual em ambos os arquivos.

Embora tenha-se gasto tempo e esforço em fazer ambos os arquivos se equivalerem, há uma dificuldade grande em resolver inconsistências entre eles, concluindo-se que suas propostas de correção e equivalência trazem um grau de incerteza em sua assertividade. Desta forma decidiu-se ignorar o arquivo “tabelaLinha” e usar apenas a “tabelaVeiculo” para as análises deste trabalho, uma vez que esta apresenta também a identificação do veículo e traz algum dado programado mesmo em feriados sem necessidade de realizar suposições para resolver inconsistências.

### 3.3.1.6 Veículos de uma linha operando em outras

O veículo de código BA125 aparece na tabela “veiculos” realizando o trajeto da linha 216, entretanto segundo a tabela “tabelaVeiculo” o mesmo veículo aparece apenas com programação e horários para outra linha, de código 233.

**Tabela 12 – Comparação da contagem da quantidade de veículos diferentes em cada dia para o arquivo “tabelaLinha” e “tabelaVeiculo”**

<b>Data</b>	<b>Veículos em “tabelaLinha”</b>	<b>Veículos em “tabelaVeiculo”</b>
2019-05-01	0	3
2019-05-02	18	13
2019-05-03	18	13
2019-05-04	6	6
2019-05-05	3	3
2019-05-06	18	13
2019-05-07	18	13
2019-05-08	18	13
2019-05-09	18	13
2019-05-10	18	13
2019-05-11	6	6
2019-05-12	3	3
2019-05-13	18	13
2019-05-14	18	13

**Fonte: Autoria própria (2022).**

#### 3.3.1.7 Problemas de amostragem ou precisão de GPS

Foram encontrados pontos GPS de trajetória dos ônibus que ultrapassam os limites físicos das ruas (atravessando terrenos, prédios, casas, etc), além de problemas de direção (como o lado incorreto das portas), conforme ilustrado pelas Figuras 1, 15 e 16.

#### 3.3.1.8 Comportamentos estranhos de amostragem - ônibus “dando ré”

Outro caso de problemas nos dados geo-temporais é capturado pelas Figuras 15 e 16, onde o veículo parece dar ré ou fazer curvas fechadas em espaços muito pequenos. Instrumentos de medição (incluindo GPSs) podem sofrer imprecisões, fazendo o equipamento parecer se movimentar mesmo estando parado ou estar a metros de distância da localização geográfica original.

#### 3.3.1.9 Imprecisões no cadastro dos pontos de ônibus

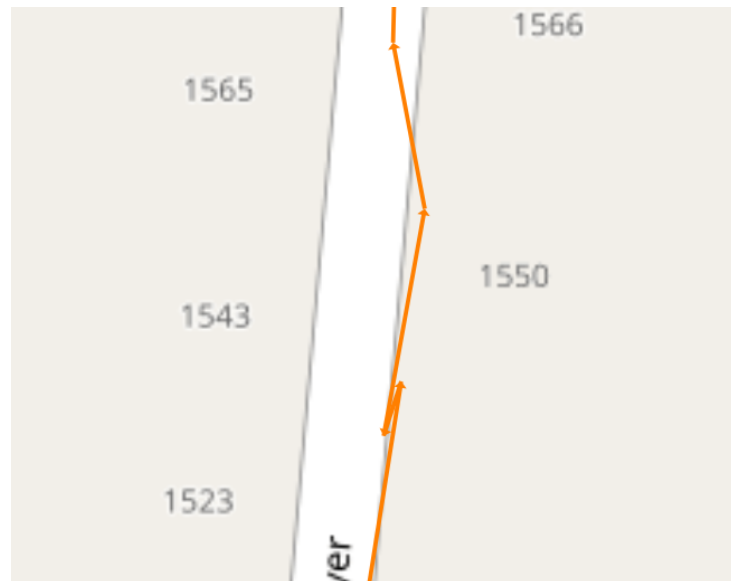
Sobre os dados geográficos de pontos de ônibus, alguns estão cadastrados a distâncias muito afastadas das ruas (com base nos dados de arruamento do *Open Street Maps*). O ponto de ônibus da linha 216 localizado na rua Oyapock (representado pela figura do ônibus em vermelho) está a 15,5 m de distância do meio fio de sua rua, como mostrado na Figura 17. Essa característica dificulta a precisão de algoritmos que fazem suposições das passagem de ônibus baseadas em distâncias que forem menor de 15,5 m.

**Figura 15 – Exemplo do ônibus BA604 da linha 216 “dando ré”**



**Fonte: Aatoria própria (2022).**

**Figura 16 – Exemplo do ônibus HA610 da linha 216 “dando ré”**



**Fonte: Aatoria própria (2022).**

### 3.3.1.10 Redundância em dados de colunas de arquivos diferentes

Colunas redundantes foram detectadas em: “tabelaLinha” para a coluna PONTO, que podem ser substituídas pela coluna NOME da tabela “pontosLinha”; “tabelaVeiculo” coluna NOME LINHA, que pode ser substituída pelo NOME da tabela “linhas”. Uma sugestão para trabalhos que venham usar estes dados é que a exclusão destas colunas podem ajudar a reduzir o espaço ocupado em memória e disco rígido.

**Figura 17 – Exemplo de ponto de ônibus da linha 216 cadastrado muito distante do meio fio da rua**



**Fonte: Autoria própria (2022).**

#### 3.3.1.11 Ausência de direção de passagem do ponto de ônibus

Uma informação importante para realizar a análise neste trabalho está ausente, que é a direção (ou azimute) de passagem do ônibus por cada ponto de ônibus. Esta característica é importante para discernir de qual lado da rua o ponto de ônibus está, principalmente em ruas de mão dupla com pontos de ônibus em mãos distintas muito próximos, exemplificado pela Figura 18. Os ícones dos ônibus simbolizam os pontos de ônibus, um em cada lado diferente da rua.

#### 3.3.1.12 Dados alternativos possíveis de serem analisados

Uma outra forma de revisão das inconsistências é utilizar uma base de dados no formato GTFS (*General Transit Feed Specification*), acessíveis através da API fechada da URBS. Estes dados são os mesmos enviados aos softwares *Google Maps* e *Moovit* para prover informações aos usuários, entretanto não caem na categoria de dados totalmente abertos por conta de sua restrição de acesso.

### **3.4 Algoritmo 2 - Extração de Azimutes**

Para que seja possível detectar a passagem de um ônibus por um ponto de ônibus, eliminando ambiguidades causadas por ruas de mão dupla com pontos em sentidos diferentes, é necessário saber em qual azimute (sentido, ângulo, ou direção) da rua os ônibus trafegam. Tal informação não está explícita nos dados abertos da URBS, mas ela pode ser descoberta através de enriquecimento de dados. Uma metodologia foi criada para enriquecer os dados



**Figura 18 – Exemplo de dois pontos de ônibus da linha 216 muito próximos, um em cada lado diferente da rua**



**Fonte: Autoria própria (2022).**

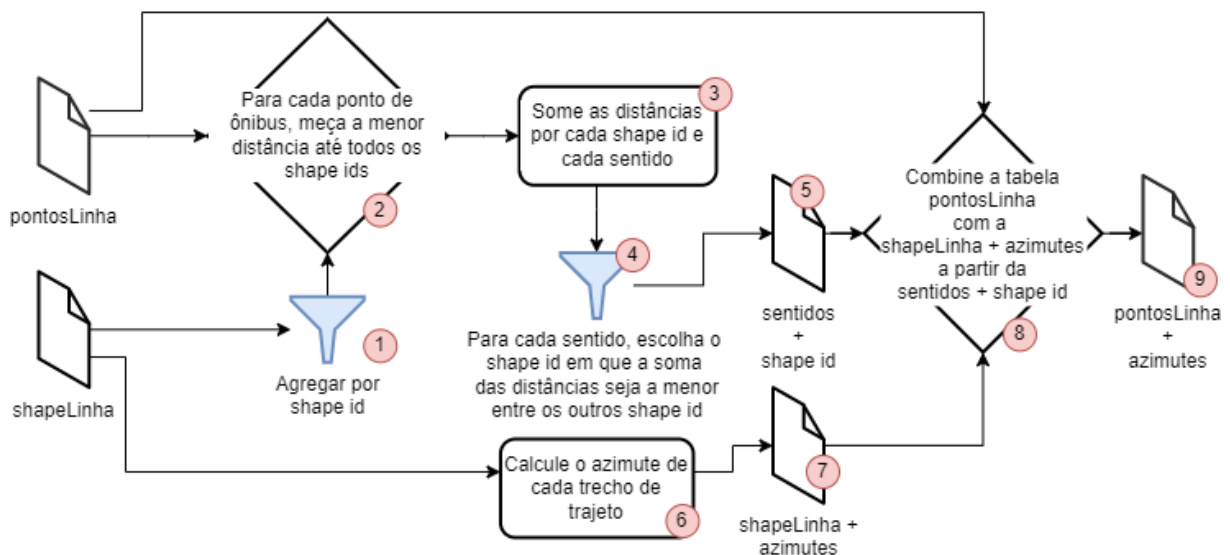
de pontos de ônibus contidos no arquivo “pontosLinha”, agregando a cada ponto de ônibus o azimute, em outras palavras, em qual ângulo ou direção (de 0 360 graus) um ônibus têm no momento em que está passando por determinado ponto. A implementação do algoritmo foi feita na linguagem *SQL* utilizando o banco de dados PostgreSQL.

O processo de enriquecimento utiliza os arquivos “shapeLinha” e “pontosLinha” como entrada. Sua metodologia está descrita graficamente na Figura 19. Seus passo a passo é descrito abaixo:

1. com a tabela “shapeLinha”, transforme todos os pontos de um mesmo identificador de shape (SHP, shape id) em uma polilinha, que representará todo o segmento de trajeto de uma linha (o caminho completo do terminal X até terminal Y), representado pelas polilinhas de cores distintas da Figura 13;
2. a partir da tabela “pontosLinha”, meça a menor distância de cada ponto de ônibus para cada polilinha em “shapeLinha” identificadas por seu SHP;
3. com o dado anterior, realize uma agregação e some as distâncias agrupadas em cada shape id (SHP) e cada sentido (SENTIDO);
4. para cada sentido (SENTIDO), escolha o shape id (SHP) em que a soma das distâncias seja a menor entre os outros shape ids, neste ponto já é possível saber qual SHP em “shapeLinha” equivale a cada sentido (SENTIDO) em “pontosLinha”;
5. crie uma tabela “sentidos + shape id” com o resultado anterior;

6. utilizando a tabela “shapeLinha”, calcule o azimuth de cada segmento de trajeto, semelhante às setas da Figura 13 e armazene-o;
7. crie uma nova tabela chamada “shapeLinha + azimutes” com os dados da etapa anterior;
8. correlacione a tabela “pontosLinha” com a “shapeLinha + azimutes” usando a “sentidos + shape id”. Neste momento cada ponto de ônibus já contém um azimuth; e
9. salve os dados da etapa anterior em uma nova tabela chamada “pontosLinha + azimutes”.

**Figura 19 – Metodologia do enriquecimento de dados para descobrir azimutes de cada ponto de ônibus em “pontosLinha”**



**Fonte: Autoria própria (2022).**

O resultado do algoritmo aplicado na linha de ônibus 216 pode ser visto na Figura 20. A direção das setas mostram o azimuth para o ponto de ônibus mais próximo (representado pelo ícone do ônibus). O comprimento das setas mostra apenas o segmento da linha em “shapeLinha” que mais correspondeu a passagem pelo ponto de ônibus. A informação do azimuth é utilizada para enriquecer os dados disponibilizados pela URBS, indicando a direção de um veículo dentro de seu trajeto na linha.

A nova tabela “pontosLinha + azimutes” criada no final do processo servirá como entrada para o Algoritmo 3 - “Algoritmo do Momento de Correspondência de Passagem”, descrito na próxima seção.

A Tabela 13 mostra um exemplo dos dados de saída gerados pelo algoritmo. A única diferença desta para a tabela “pontosLinha” é a adição da coluna AZIMUTE que contém o valor do azimuth (ou direção) em radianos.

Figura 20 – Resultado da aplicação da metodologia descrita pela Figura 19 para a linha 216



Fonte: Autoria própria (2022).

Tabela 13 – Exemplo de dados contidos no arquivo “pontosLinha + azimutes”

COD	SENTIDO	SEQ	NOME	NUM	LAT	LON	TIPO	IT_ID	AZIMUTE
216	Terminal Cabral	1	Terminal Port?o - 216 - Cabral / Port?o	150017	-25.476	-49.292	Plataforma	559	5.905
216	Terminal Cabral	2	Av. Pres. Kennedy, 3683 - Port?o	150023	-25.474	-49.288	Novo mobiliario	559	0.841
216	Terminal Cabral	3	Av. Pres. Kennedy, 3399 - Agua Verde	150021	-25.472	-49.286	Novo mobiliario	559	0.851
...									
216	Terminal Cabral	33	Terminal Cabral - 216 - Cabral / Port?o	104404	-25.406	-49.252	Plataforma	559	0.856
216	Terminal Port?o	1	Terminal Cabral - 216 - Cabral / Port?o	104404	-25.406	-49.252	Plataforma	558	0.837
216	Terminal Port?o	2	Av. Munhoz da Rocha, 706 - Cabral	130866	-25.408	-49.251	Novo mobiliario	558	0.809
216	Terminal Port?o	3	Rua Professor Artur Loliola, 145 - Cabral	130101	-25.411	-49.251	Novo mobiliario	558	0.849
...									
216	Terminal Port?o	37	Terminal Port?o - 216 - Cabral / Port?o	105806	-25.476	-49.292	Plataforma	558	0.847

Fonte: Autoria própria (2022).

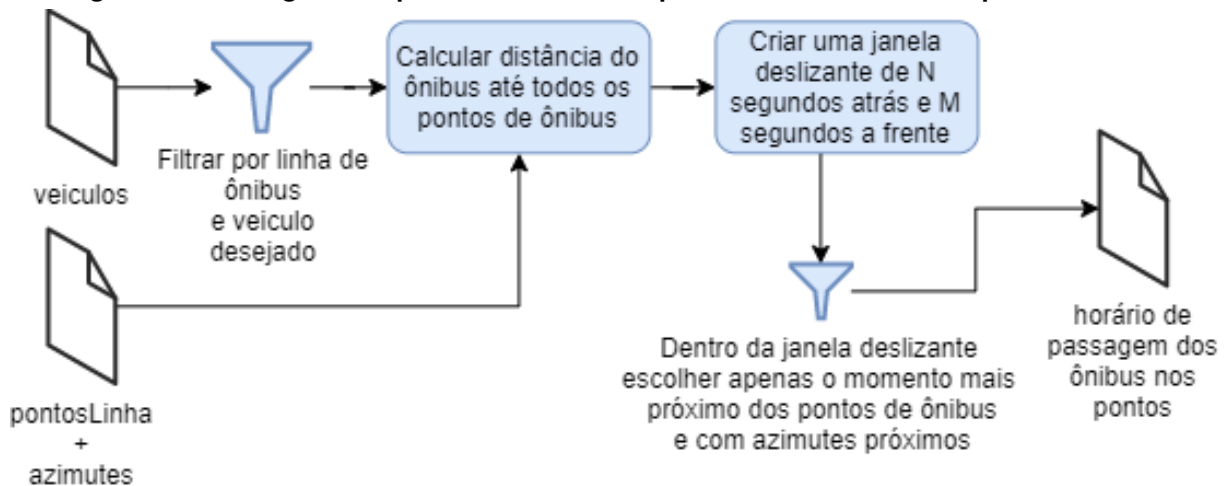
### 3.5 Algoritmo 3 - Map Matching da trajetória dos ônibus com os pontos de ônibus

O algoritmo 3 requer duas tabelas como entrada: “veiculos” e “pontosLinha + azimutes” (recém criada no Algoritmo 2) e devolve como saída a tabela “momento de passagem do onibus” com os horários de passagem detectados, id da linha de ônibus, id do ponto de ônibus, ordem de visitaçao do ponto de ônibus no trajeto para todos os ônibus de uma linha escolhida.

O processamento ocorre em 6 etapas, descritas pela Figura 21:

1. Carrega-se a tabela “veiculos”, contendo os dados espaço-temporais de movimentação de todos os ônibus o dia desejado;
2. Dado um veículo e uma linha de ônibus, seleciona-se apenas as linhas que correspondem a eles, removendo todos os outros da análise;

Figura 21 – Fluxograma explicando a ordem do processamento de dados para encontrar

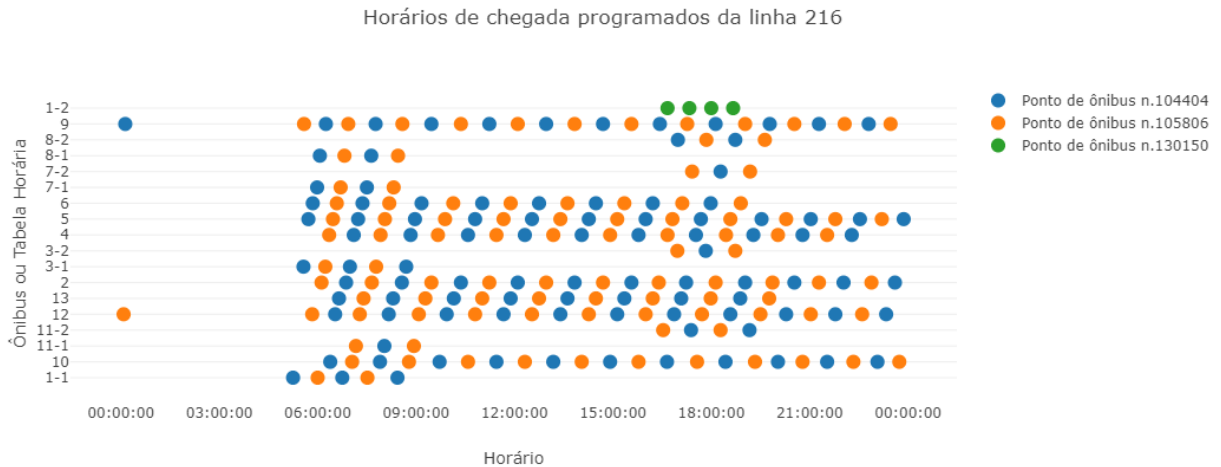


Fonte: Autoria própria (2022).

- Carrega-se a tabela “pontosLinha + azimutes” criada previamente. Para cada posição do ônibus no mapa, calcula-se a distância Euclidiana, *Haversine* ou *Vincenty* (a desejar) para todos os pontos de ônibus daquela linha e utiliza-as no próximo passo;
- Cria-se uma janela móvel baseada em tempo ordenado de forma crescente, com seu centro em uma das amostras espaço-temporais, à esquerda todas as amostras de DELTA-JANELA segundos atrás do centro e à direita todas as amostras de DELTA-JANELA segundos adiante. A janela se move de um em um, crescente ao longo do tempo. Um exemplo é descrito na tabela 14;
- Compara-se o elemento do centro da janela (terceiro, caso o tamanho da janela seja cinco) com os outros elementos. Se o elemento do centro for o ponto mais próximo de algum ponto de ônibus daquela linha e também menor que DIST-MAX e se a diferença absoluta do azimuth do ponto entre o azimuth da trajetória espaço-temporal for menor que DELTA-AZIMUTE, ele é o ponto vencedor, logo seu horário representa o horário de chegada daquele ônibus no pontos de ônibus. Percebe-se que na tabela 14 a leitura de id 4 apresentou maior proximidade com o ponto de ônibus 1, logo ela será escolhida como correspondência do momento de passagem; e
- Salva-se os horários reais de chegada do ônibus nos pontos em uma nova tabela.

A Figura 23 apresenta a estimativa de chegada para o ônibus BA125 da linha 216 (Cabrál/Portão) para o dia 03/05/2019. Os círculos vermelhos mostram os pontos de ônibus da linha e as setas roxas mostram a correspondência da passagem de um ônibus por eles. Internamente estão armazenados nas setas roxas alguns dados como direção em que o ônibus se movia (também chamado de azimuth ou *bearing*), o horário de passagem do ônibus naquele

**Figura 22 – Horários de chegada programados (tabelados) para a linha 216 no dia 03/05/2019**



**Fonte: Autoria própria (2022).**

**Tabela 14 – Exemplo da janela de tempo usada para realizar a correspondência de posições do ônibus com pontos de ônibus (PO)**

Id	Horário leitura	Distância Vincenty até o PO 1	Distância Vincenty até o PO 2	Distância Vincenty até o PO 3
1	10:00:00	100m	500m	700m
2	10:00:30	60m	503m	710m
3	10:01:05	40m	510m	740m
4	10:01:35	2m	550m	770m
5	10:02:00	20m	590m	790m

**Fonte: Autoria própria (2022).**

ponto, a distância mínima do ônibus até o ponto de ônibus referenciado, identificador do veículo e linha de ônibus.

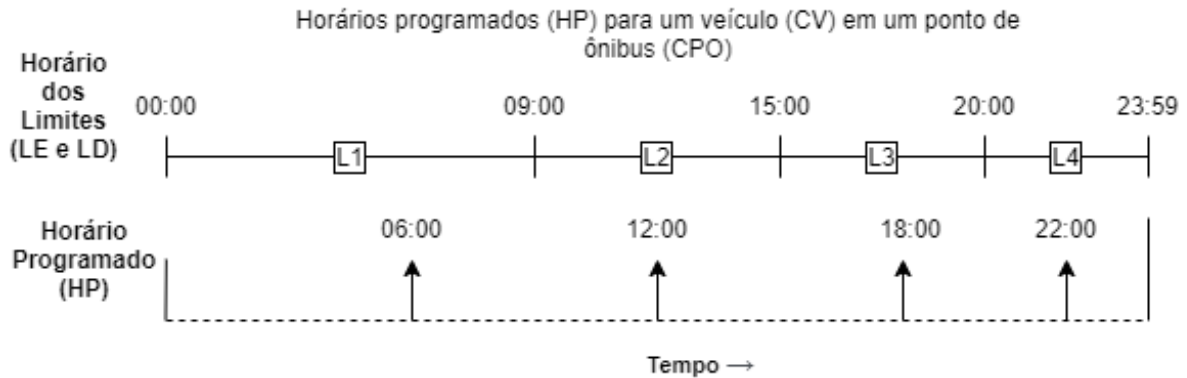
### 3.6 Algoritmo 4 - Comparação Real x Programado

O quarto algoritmo deste trabalho compara os horários programados de passagem dos ônibus com os horários estimados mencionados anteriormente. O foco do algoritmo é mostrar a diferença de tempo entre o horário programado e o horário de passagem estimado. O algoritmo é ilustrado pela Figura 24:

1. Carrega-se a tabela "tabelaVeiculo", contendo os horários programados, filtrando-a pela data desejada (DD) e pelo código da linha de ônibus (CLO);
2. Os dados são ordenados de forma crescente com base nas colunas CLO, CV (código do veículo) e HP (horário programado);



**Figura 24 – Ilustração sobre os intervalos de limites criados (L1, L2, L3, L4) para um conjunto de horários programados de um mesmo ponto de ônibus para um mesmo veículo**



**Fonte: Autoria própria (2022).**

*outliers* do processo. O motivo de remoção dos veículos que ocorre na etapa 8 é para melhorar a confiança das análises, uma vez que veículos que não estão seguindo o trajeto planejado podem prejudicar precisão das observações realizadas.

### 3.7 Desenvolvimento

As análises nesta etapa foram realizadas entre os dias 01/05/2019 e 14/05/2019. Os código fonte de todos dos algoritmos em NodeJS e SQL para enriquecimento de dados, desenvolvidos neste trabalho estão presentes no GitHub do autor<sup>5</sup>.

#### 3.7.1 Implementação do Algoritmo 3 - *Map Matching* da trajetória dos ônibus com os pontos de ônibus

O algoritmo 3 foi implementado na linguagem SQL, em PostgreSQL 13. Seu funcionamento depende de três parâmetros, que precisam ser atribuídos antes do seu início. Alguns testes e observações feitos na linha de ônibus 216 para o dia escolhido mostraram que os parâmetros razoáveis estão nesta faixa:

- DIST-MAX = 40 metros;
- DELTA-JANELA = 20 minutos;
- DELTA-AZIMUTE = 45 graus ( $\pi \div 4$  rad).

<sup>5</sup> Disponível em: <<https://github.com/tiagosmx/master-thesis-public-transportation>> Acesso em: 20 out, 2022.

Em alguns momentos do dia, o GPS parece perder precisão e os ônibus passam mais de 30 metros de distância dos pontos de ônibus correspondentes. Estes casos justificam este valor alto de 40 metros para DIST-MAX.

A razão do valor 45 graus em DELTA-AZIMUTE está em permitir que o algoritmo remova da correspondência os pontos de ônibus que estão no lado oposto da rua durante o trajeto do veículo.

### 3.7.2 Implementação do Algoritmo 4 - Comparação Real x Programado

O Algoritmo 4 foi implementado na linguagem SQL, em PostgreSQL 13. Como mencionado anteriormente, seu objetivo é cruzar as informações geradas pelo Algoritmo 3 e a “tabelaVeiculo”, permitindo descobrir o tempo de atraso ou adiantamento dos ônibus. Outra parte de sua função é realizar uma limpeza nos dados, removendo elementos que possivelmente trariam problemas para análises.

A Tabela 15 mostra a quantidade de códigos de veículos presentes em cada etapa do processo do algoritmo. Nota-se que dos 14 dias de dados importados, apenas 5 dias retornaram dados limpos após a execução do Algoritmo 4. Análises exploratórias mostraram que esta discrepância dos dados brutos em relação aos limpos está relacionada a: a ausência de dados na tabela “veiculos”; a diferença dos códigos de veículos em operação e dos códigos de veículos programados; e a inconsistência do trajeto e horário de passagem de certos veículos em comparação ao trajeto programado em “tabelaVeiculo”.

Ao final da execução do algoritmo, uma tabela nova com os resultados é gerada e foi analisada através da ferramenta Metabase.

A primeira análise é a da frequência dos atrasos ou adiantamentos observados ao longo dos dias. A Figura 25 mostra que a incidência mais comum é a da categoria entre adiantamentos de 7,5 minutos até 0, com 213 ocorrências; em segundo lugar, adiantamentos entre 15 minutos e 7,5 minutos com 118 ocorrências; e em terceiro lugar, atrasos entre 0 e 7,5 minutos com 75 ocorrências.

Considerando apenas as ocorrências de adiantamentos e atrasos entre 7,5 minutos, percebe-se através da Figura 26 que elas representam a maioria, 63,7%, em contraste com 36,3% das ocorrências com atrasos e adiantamentos mais extremos, de acima de 7,5 minutos.

A Figura 27 apresenta a média dos atrasos e do desvio padrão. Nota-se que há um aumento do desvio padrão entre às 08h e 09h, e das 17h às 19h, um indício de que nestes horários ocorrem mais incertezas e variações nos horários de chegada dos ônibus em suas respectivas paradas. Os períodos de menor variabilidade nos horários ocorrem durante o começo da manhã, das 04h às 06h, e no fim da manhã para o começo da tarde, das 11h às 15h.

Na perspectiva dos atrasos, percebe-se que aumentam muito no final da tarde entre às 18h e 19h, com uma média de 8 minutos, embora acompanhada também de um pico no desvio



Tabela 15 – Quantidade de veículos únicos antes e depois dos algoritmos e filtros

Data arquivo “veiculos”	Quantidade veículos presentes em “veiculos”	Data arquivos	Quantidade de veículos presentes em “tabela_veiculo”	Quantidade de veículos presentes após algoritmo de Comparação com filtro
2019-05-02	4	2019-05-01	3	0
2019-05-03	13	2019-05-02	13	4
2019-05-04	0	2019-05-03	13	0
2019-05-05	6	2019-05-04	6	5
2019-05-06	4	2019-05-05	3	0
2019-05-07	16	2019-05-06	13	7
2019-05-08	1	2019-05-07	13	0
2019-05-09	0	2019-05-08	13	0
2019-05-10	0	2019-05-09	13	0
2019-05-11	3	2019-05-10	13	0
2019-05-12	6	2019-05-11	6	0
2019-05-13	3	2019-05-12	3	3
2019-05-14	16	2019-05-13	13	4
-	-	2019-05-14	13	-

Fonte: Autoria própria (2022).

padrão. Uma característica interessante do gráfico é que a média está muito mais situada na parte inferior do que superior, indicando que os ônibus costumam, mais adiantar sua chegada do que atrasar. É importante salientar que por esta análise ser feita apenas nas paradas de ônibus com horários programados (geralmente localizadas nos pontos extremos da linha) estes dados não refletem os atrasos e adiantamentos na visão de quem utiliza o transporte em paradas intermediárias.

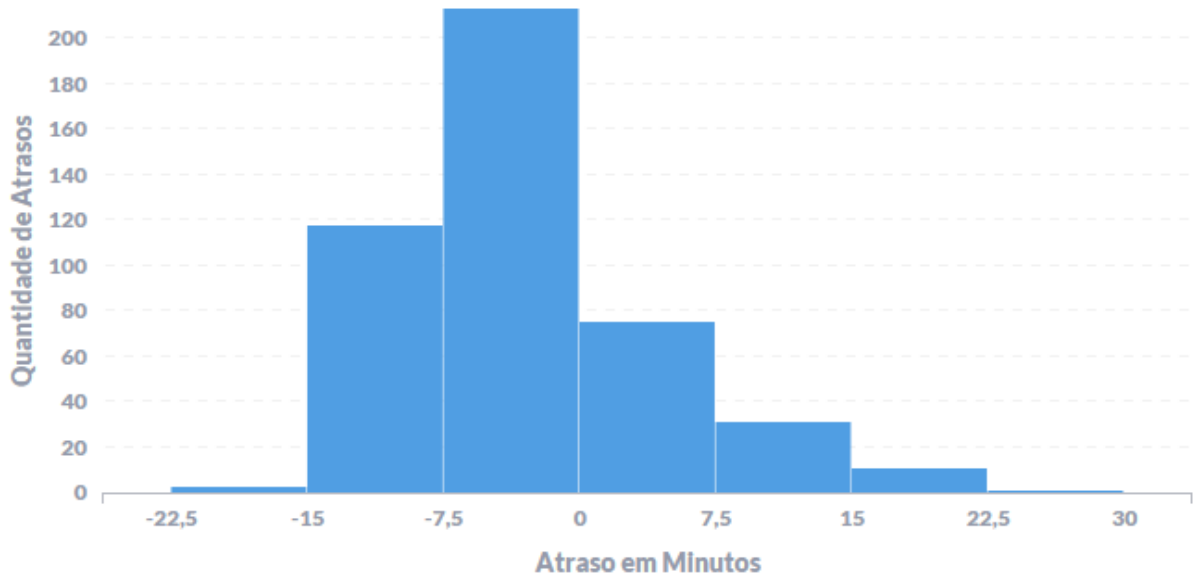
### 3.7.3 Limitações ou particularidades encontradas

Nesta subseção discute-se as limitações e particularidades dos algoritmos encontradas através de análises exploratórias realizadas durante o desenvolvimento do trabalho.

#### 3.7.3.1 Limitações do Algoritmo 1

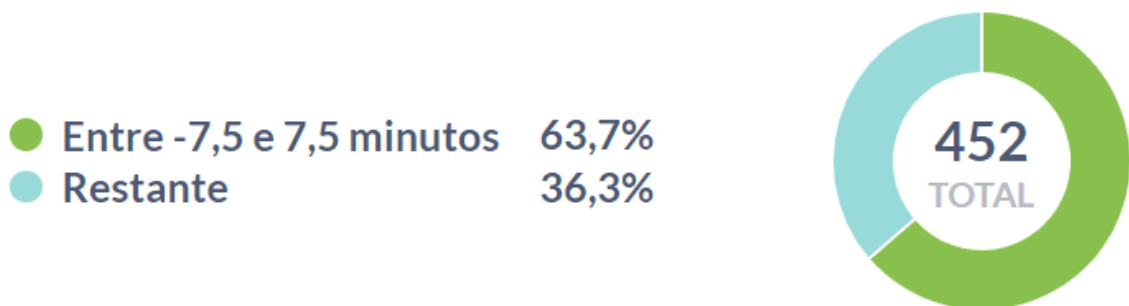
O algoritmo cria uma tabela para cada dia importado seguindo o padrão da própria fonte dos dados, de um arquivo por dia. Um ponto positivo desta abordagem é a de permitir consultas de varreduras completas de tabelas (*full table scan*) serem executadas mais rapidamente, já que a quantidade de dados a ser analisada é menor. Em contrapartida, análises que requerem de vários dias podem exigir maior esforço, sendo necessário unir as tabelas de mesmo tipo através de alguma consulta/*view* ou executar os algoritmos uma vez para cada data.

Figura 25 – Gráfico mostrando a frequência da ocorrência de atrasos ou adiantamentos durante o período observado



Fonte: Autoria própria (2022).

Figura 26 – Gráfico de pizza mostrando ocorrência de atrasos e adiantamentos entre 7,5 minutos em comparação ao restante



Fonte: Autoria própria (2022).

Algumas tabelas como “pontosLinha” e “linhas” contêm dados idênticos em dias consecutivos, o que acaba resultando em uma série de dados duplicados espalhados por várias tabelas.

### 3.7.3.2 Limitações do Algoritmo 2

Nem todas as linhas de ônibus contêm uma quantidade igual de segmentos de trajeto em “shapeLinha” com relação ao número de sentidos em “pontosLinha”. Análises realizadas para o dia 03/05/2019 mostram que das 315 linhas cadastradas, 55 contêm diferenças nesta

**Figura 27 – Gráfico de linha mostrando a média dos atrasos (à esquerda) e do desvio padrão (à direita) no eixo vertical Y, juntamente com as horas do dia, expressas no eixo horizontal X**



**Fonte: Autoria própria (2022).**

contagem e 2 estão ausentes de qualquer informação de ambos, esta proporção representa 17% de todas as linhas.

Para obter um resultado adequado do algoritmo é necessário excluir das opções as linhas com contagens divergentes.

### 3.7.3.3 Limitações do Algoritmo 3

As situações de imprecisões encontradas nas Figuras 15 e 16 causam correspondências falsas no algoritmo proposto. No momento em que o ônibus “dá ré”, o algoritmo considera que o ônibus está fazendo uma passagem no ponto de ônibus localizado no lado oposto da rua, uma suposição incorreta. Uma forma de driblar estes problema é realizar mais um filtro nos dados das correspondência a fim de excluir as do momento de passagem de um mesmo veículo mas em pontos de ônibus de um outro trajeto em um espaço de tempo tão curto.

O algoritmo encontra apenas um momento de passagem para cada ponto de ônibus. Isso significa que se o ônibus permanecer por um longo período de tempo estacionado em um ponto de ônibus, o momento registrado será o momento de sua chegada. O instante de sua saída não é considerado, o que poderia ser uma informação interessante em algumas análises.

#### 3.7.3.4 Limitações do Algoritmo 4

A base deste algoritmo é a “tabelaVeiculo”, que contém os horários programados de passagem nos pontos de ônibus, juntamente com os códigos dos veículos e das linhas. Como esta tabela contém horários programados para apenas os terminais de ônibus da linha 216 (semelhante ao caso da grande maioria dos ônibus não-BRT), o algoritmo se limita apenas a estes dados.

A comparação entre o horário real e teórico ocorre principalmente por meio do código do veículo. Isso significa que são desconsideradas as situações em que um ônibus precisou ser substituído por outro, com outro código de veículo, e essas alterações não estão representadas “tabelaVeiculo”.

## 4 CONCLUSÃO

Este trabalho teve como objetivo um método de quatro etapas para analisar os dados de streaming de trajetórias de GPS, contendo 1) análise e limpeza dos dados; 2) extração de azimutes; 3) um método para detecção do momento (horário) de passagem dos ônibus nos respectivos pontos de ônibus de sua linha de operação e 4) correlação dos tempos reais e teóricos da passagem nos pontos de ônibus.

Para compreender as grandes áreas que contribuem para esta pesquisa, apresentou-se um sumário dos conceitos básicos sobre Sistemas de Informações Geográficas, Dados Abertos e Cidades Inteligentes com foco na exploração de dados abertos de ônibus de transporte público e movimentação de GPS.

Nas análises preliminares notou-se algumas inconsistências nos dados (como imprecisões de GPS), mas também inconsistências de cadastro (como ônibus operando em linhas das quais não pertencem, ausência de tabelas de horários teóricos para certos veículos e presença de horários teóricos em pontos de ônibus não condizentes com os da linha).

Um protótipo para as quatro etapas foi implementado e sua metodologia de funcionamento foi explicada, juntamente com exemplos e desafios encontrados no seu funcionamento.

Como trabalhos futuros, podemos citar a análise de métricas para atrasos de ônibus, a estimativa de horários reais mais compatíveis com os reais, e a correlação de atrasos de horários com efeitos climáticos, dias da semana, feriados (entre outros). Espera-se que, com os resultados deste trabalho, abram-se caminhos para novas análises envolvendo atraso e adiantamento de ônibus de previsões com base em dados históricos ou até mesmo sistemas para tomada de ações em tempo real visando uma melhoria na qualidade do transporte público da cidade.

## REFERÊNCIAS

- ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. **Journal of Urban Technology**, Taylor & Francis, v. 22, n. 1, p. 3–21, 1 2015. ISSN 1063-0732. Disponível em: <http://dx.doi.org/10.1080/10630732.2014.942092>.
- AZAMBUJA, L. S. D. **Dados Abertos em Cidades Inteligentes: Portais de Dados Abertos Possibilitando o Acesso e Uso da Informação**. 2016. Monografia (TCC) — UFRGS, 2016.
- BARCZYSZYN, G. L. **Integração de dados geográficos para planejamento urbano da cidade de Curitiba**. 2015. Monografia (TCC) — Universidade Tecnológica Federal do Paraná, 2015.
- BELUZO, J. R. **IDEO – Integrador de dados da Execução Orçamentária Brasileira : Um estudo de caso da integração de dados das receitas e despesas nas Esferas Federal , Estadual – Governo de São Paulo , e Municipal – Municípios do Estado de São Paulo**. 2015. Monografia (TCC) — USP, 2015.
- BORGES, K. A. d. V. **Modelagem de Dados Geográficos**. 1. ed. Belo Horizonte: UFMG, 2002. 66 p.
- CAVADA, M.; ROGERS, C.; HUNT, D. Smart Cities: Contradicting Definitions and Unclear Measures. *In: Proceedings of The 4th World Sustainability Forum*. Basel, Switzerland: MDPI, 2014. (4, November), p. f004. Disponível em: <http://www.sciforum.net/conference/wsf-4/paper/2454>.
- CHAO, P. *et al.* A Survey on Map-Matching Algorithms. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 12008 LNCS, p. 121–133, 10 2019. ISSN 16113349. Disponível em: <http://arxiv.org/abs/1910.13065>.
- CRISTOBAL, T. *et al.* Systematic Approach to Analyze Travel Time in Road-Based Mass Transit Systems Based on Data Mining. **IEEE Access**, v. 6, n. c, p. 32861–32873, 2018. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/8360420/>.
- DAVENPORT, T. H. **Big data at work: dispelling the myths, uncovering the opportunities**. 1. ed. Boston: Harvard Business Review Press, 2014. 241 p. ISBN 9783800648153. Disponível em: <http://elibrary.vahlen.de/index.php?doi=10.15358/9783800648153>.
- DAVENPORT, T. H.; PRUSAK, L. **Information Ecology: Mastering the Information and Knowledge Environment**. São Paulo: Editora Futura, 1997. 312 p. ISBN 85-86082-72-4. Disponível em: <http://www.uel.br/revistas/uel/index.php/informacao/article/view/1649>.
- DIETRICH, D. *et al.* **Open Data Handbook**. 2012. Disponível em: <https://opendatahandbook.org/guide/en/>.
- DIRKS, S.; KEELING, M. **A vision of smarter cities**. [S.l.], 2009. 18 p. Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+vision+of+smarter+cities#1>.
- DRUCK, S. *et al.* **Análise Espacial de Dados Geográficos**. Brasília: EMBRAPA, INPE, 2004. 209 p. ISBN 85-7383-260-6. Disponível em: <http://www.dpi.inpe.br/gilberto/livro/analise/>.
- FERNANDEZ-ANEZ, V. Stakeholders Approach to Smart Cities: A Survey on Smart City Definitions. *In: Lecture Notes in Computer Science (including subseries Lecture Notes*

in **Artificial Intelligence and Lecture Notes in Bioinformatics**). 1. ed. Madrid: Springer International Publishing Switzerland, 2016. v. 9704, p. 157–167. ISBN 9783319395944. Disponível em: [http://link.springer.com/10.1007/978-3-319-39595-1\\_16](http://link.springer.com/10.1007/978-3-319-39595-1_16).

GIFFINGER, R. *et al.* City-ranking of European medium-sized cities. **Centre of Regional Science, Vienna UT**, n. October, 2007.

HALICIOGLU, F.; ANDRÉS, A. R.; YAMAMURA, E. Modeling crime in Japan. **Economic Modelling**, Elsevier B.V., v. 29, n. 5, p. 1640–1645, 9 2012. ISSN 02649993. Disponível em: <http://dx.doi.org/10.1016/j.econmod.2012.05.026>.

HAMADA, E.; GONÇALVES, R. R. do V. **Introdução ao Geoprocessamento: princípios básicos e aplicação**. 1st. ed. Jaguariúna, São Paulo: EMBRAPA, 2007. 52 p. ISBN 1516-4691. Disponível em: [http://www.cnpma.embrapa.br/download/documentos\\_67.pdf](http://www.cnpma.embrapa.br/download/documentos_67.pdf).

HÖJER, M.; WANGEL, J. Smart Sustainable Cities: Definition and Challenges. *In*: HILTY, L. M.; AEBISCHER, B. (Ed.). **Advances in intelligent systems and computing**. Cham: Springer International Publishing, 2015, (Advances in Intelligent Systems and Computing, v. 310). p. 333–349. ISBN 978-3-319-09227-0. Disponível em: <http://link.springer.com/10.1007/978-3-319-09228-7>.

JUNIOR, P. C. D. **Serviços Telemáticos em uma Rede de Transporte Público Baseados em Veículos conectados e Dados Abertos**. 2017. 116 p. Dissertação (Mestrado) — UTFPR, Curitiba, 2017. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/2691>.

KADIR, R. A. *et al.* Clustering of public transport operation using K-means. *In*: **2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)**. Shangai: IEEE, 2018. (3, Ivi), p. 427–432. ISBN 978-1-5386-4794-3. Disponível em: <https://ieeexplore.ieee.org/document/8367721/>.

KONO, F. A. M. **Um Modelo de Representação Computacional Baseado em Conceitos de Crescimento Urbano Associados a Alvarás e Primitivas em Banco de Dados Espacial**. 2016. 1–163 p. Dissertação (Mestrado) — UTFPR, 2016. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/2035>.

KOZIEVITCH, N. P. *et al.* Exploratory Analysis of Public Transportation Data in Curitiba. **43o Seminário Integrado de Software e Hardware, Porto Alegre**, v. 43, n. November, p. 1656–1666, 2016.

KUMAR, C.; GANGULY, A. Travelling Together But Differently: Comparing Variations in Public Transit User Mode Choice Attributes Across New Delhi and New York. **Theoretical and Empirical Researches in Urban Management**, v. 13, n. 3, p. 22–36, 2018.

LARA, F. R. D.; MARTINS, T. S. **Protótipo de sistema de caronas aplicado na Universidade Tecnológica Federal do Paraná**. 2016. Monografia (TCC) — UTFPR, 2016.

LAWHEAD, J. **Learning Geospatial Analysis with Python**. 2nd. ed. Birmingham: Packt Publishing, 2015. v. 7. 394 p. ISSN 00167061. ISBN 9780128143919.

MARTINS, T. S.; KOZIEVITCH, N. P. Os desafios de uma aplicação de Carpooling no contexto de uma comunidade universitária brasileira. **XI Escola Regional de Banco de Dados**, XI, p. 10, 2015. Disponível em: <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Autor?id=34455>.

MARTINS, T. S. *et al.* Map Matching: Uma análise de Dados Streaming de trajetórias do GPS no Transporte Público. *In*: **SBSI 2022 - Trilha Emergente em Cidades Inteligentes**. [S.l.]: SBC, 2022. p. 1–8.

MENDES, T. C. M. **Smart Cities : Solução Para As Cidades Ou Aprofundamento Das Desigualdades Sociais?** Rio de Janeiro: [s.n.], 2020. 23 p.

MENEGUETTE, R. I.; GRANDE, R. E. D.; LOUREIRO, A. A. F. **Intelligent Transport System in Smart Cities**. 1. ed. Cham: Springer International Publishing, 2018. 191 p. (Urban Computing). ISBN 978-3-319-93331-3. Disponível em: <http://link.springer.com/10.1007/978-3-319-93332-0>.

MILLARD-BALL, A.; HAMPSHIRE, R. C.; WEINBERGER, R. R. Map-matching poor-quality GPS data in urban environments: the pgMapMatch package. **Transportation Planning and Technology**, v. 42, n. 6, p. 539–553, 2019. ISSN 10290354.

PANIGRAHI, N. **Computing in Geographic Information Systems**. 1st. ed. Boca Raton, Florida: CRC Press, 2014. 294 p. ISBN 9781482223163.

PARCIANELLO, Y. **Análise de Origem-Destino do uso do Sistema de Transporte Coletivo de Curitiba Sob o Ponto de Vista de Regions of Interest**. 2020. 1–89 p. Dissertação (Mestrado) — UTFPR, Curitiba, 2020. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/4821>.

PARCIANELLO, Y. *et al.* Transportation: An Overview from Open Data Approach. *In: 2018 IEEE International Smart Cities Conference (ISC2)*. Kansas City: IEEE, 2018. p. 1–8. ISBN 978-1-5386-5959-5. Disponível em: <https://ieeexplore.ieee.org/document/8656937/>.

PARCIANELLO, Y. *et al.* Origin-Destination Data: a prototype and related scenarios. **Revista Brasileira de Computação Aplicada**, v. 13, n. 2, p. 16–27, 2021. ISSN 2176-6649.

PEREIRA, F. C.; COSTA, H.; PEREIRA, N. M. An off-line map-matching algorithm for incomplete map databases. **European Transport Research Review**, v. 1, n. 3, p. 107–124, 2009. ISSN 18670717.

SCHRANK, D.; LOMAX, T.; EISELE, B. **2011 Urban Mobility Report**. College Station, 2011. 147 p. Disponível em: [https://nacto.org/docs/usdg/2011\\_urban\\_mobility\\_report\\_schrank.pdf](https://nacto.org/docs/usdg/2011_urban_mobility_report_schrank.pdf).

SEBASTIANI, M. T.; LUDERS, R.; FONSECA, K. V. O. Evaluating Electric Bus Operation for a Real-World BRT Public Transportation Using Simulation Optimization. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, n. 10, p. 2777–2786, 10 2016. ISSN 1524-9050. Disponível em: <https://ieeexplore.ieee.org/document/7415957/>.

SHETH, A. **Semantic Services, Interoperability and Web Applications**. IGI Global, 2011. 420 p. ISBN 9781609605933. Disponível em: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-593-3>.

SILVA, E. L. Chiesse da *et al.* Combining K-means method and complex network analysis to evaluate city mobility. *In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro: IEEE, 2016. v. 19, p. 1666–1671. ISBN 978-1-5090-1889-5. Disponível em: <https://ieeexplore.ieee.org/document/7795782/>.

SOUZA, F. X. D. *et al.* Visualização de Dados de Origem-Destino - Foco em Unidades de Saúde e Educação. **Geoinfo**, v. 1, n. 21, p. 6, 2020.

TU, Y.; YANG, J. Analysis and Forecast of Passenger Flow Based on Public Transportation IC Card and GPS Data. **International Conference on Computer Science and Network Technology**, p. 281–285, 2016.

Un-Habitat. **Planning and Design for Sustainable Urban Mobility**. Routledge, 2013. 1–317 p. ISSN 1098-6596. ISBN 9781317932871. Disponível em: <https://www.taylorfrancis.com/books/9781317932871>.



Urbanização de Curitiba. **Composição da Frota da URBS**. 2021. Disponível em: <https://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte>.

VILA, J. J. F. R. **Clusterização e Visualização Espaço-temporal de Dados Georreferenciados Adaptando o Algoritmo Marker Clusterer - um caso de uso em Curitiba**. 2016. 87 p. Dissertação (Mestrado) — UTFPR, Curitiba, 2016. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/2832>.

VILA, J. J. R. *et al.* Urban Mobility Challenges - An Exploratory Analysis of Public Transportation Data in Curitiba. **Revista de Informática Aplicada**, v. 12, n. 1, p. 14, 2016.

WONG, J. Leveraging the General Transit Feed Specification for Efficient Transit Analysis. **Transportation Research Record: Journal of the Transportation Research Board**, v. 2338, n. 1, p. 11–19, 1 2013. ISSN 0361-1981. Disponível em: <http://journals.sagepub.com/doi/10.3141/2338-02>.

YIN, C. *et al.* A literature survey on smart cities. **Science China Information Sciences**, v. 58, n. 10, p. 1–18, 10 2015. ISSN 1674-733X. Disponível em: <http://link.springer.com/10.1007/s11432-015-5397-4>.

## **ANEXO A – Algoritmo de descoberta de azimutes dos pontos de ônibus**

```

1 WITH
2 chosen_bus_lines AS (
3     SELECT
4         *
5     FROM
6         (
7             VALUES
8                 ('216')
9         ) bus_lines (bus_line_id)
10 ),
11 chosen_dates AS (
12     SELECT *
13     FROM (
14     VALUES
15         ('2021-03-25'::DATE)
16     ) AS d(file_date)
17 ),
18 pontos_linha AS (
19     SELECT
20         -- (index) índice do ponto
21         file_index,
22         -- (nome) nome do ponto
23         bus_stop_name,
24         -- (num) código do ponto
25         bus_stop_id,
26         -- (seq) índice do ponto no percurso
27         seq,
28         -- (grupo) grupo a qual este ponto de ônibus faz parte, onde pessoas podem
29         pegar outros ônibus com uma mesma passagem
30         bus_stop_group,
31         -- (sentido) nome do último ponto de ônibus no final do percurso
32         way,
33         -- (tipo) tipo do ponto de ônibus
34         bus_stop_type,
35         -- (itinerary_id) id do itinerário
36         itinerary_id,
37         -- (cod) código da linha
38         bus_line_id,
39         -- (geom) coordenada geográfica do ponto de ônibus
40         bus_stop_point_geom,
41         -- a data do arquivo que originou o dado
42         file_date
43     FROM
44         pontos_linha_2021_03_25
45 ),
46 shape_linha AS (
47     SELECT
48         id,
49         shp,
50         lat,
51         lon,
52         shape_point_geom,
53         bus_line_id,
54         file_date
55     FROM
56         shape_linha_2021_03_25
57 ),
58 shapes_as_polylines AS (

```

```

58 SELECT
59     file_date,
60     bus_line_id,
61     shp,
62     st_makeline (
63         shape_point_geom
64         ORDER BY
65             id
66     ) shape_polyline_geom
67 FROM
68     shape_linha
69 GROUP BY
70     file_date,
71     bus_line_id,
72     shp
73 ),
74 shapes_and_sentidos AS (
75     SELECT
76         file_date,
77         bus_line_id,
78         shp,
79         way
80     FROM
81         (
82             SELECT
83                 ROW_NUMBER() OVER (
84                     PARTITION BY (file_date, bus_line_id)
85                     ORDER BY
86                         COUNT(*) DESC
87                 ) rank,
88                 COUNT(*) OVER (PARTITION BY (file_date, bus_line_id)) rank_max,
89                 COUNT(*) OVER (PARTITION BY (file_date, bus_line_id)) / 2 top_ranks,
90                 COUNT(*),
91                 file_date,
92                 bus_line_id,
93                 shp,
94                 way
95             FROM
96                 (
97                     SELECT
98                         x1.st_distance,
99                         ROW_NUMBER() OVER (
100                             PARTITION BY pl.file_date, pl.file_index
101                             ORDER BY
102                                 x1.st_distance ASC
103                         ) rank,
104                         pl.file_date,
105                         pl.bus_line_id,
106                         pl.way,
107                         sap.shp
108                     FROM
109                         pontos_linha pl
110                     JOIN shapes_as_polylines sap ON pl.file_date = sap.file_date
111 and pl.bus_line_id = sap.bus_line_id,
112                     LATERAL (
113                         SELECT
114                             st_distance (pl.bus_stop_point_geom,
115                             sap.shape_polyline_geom) AS st_distance

```

```

114         ) x1
115     WHERE
116         pl.bus_line_id IN (
117         SELECT
118             bus_line_id
119         FROM
120             chosen_bus_lines
121         )
122     ORDER BY
123         pl.file_date,
124         pl.bus_line_id,
125         pl.way,
126         pl.seq
127     ) q1
128 WHERE
129     rank = 1
130 GROUP BY
131     (
132         file_date,
133         bus_line_id,
134         shp,
135         way
136     )
137 ) q2
138 WHERE
139     rank <= top_ranks
140 ),
141 shapes_and_azimuths AS (
142     SELECT
143         *
144     FROM
145         (
146             SELECT
147                 *,
148                 LAG(shape_point_geom) OVER w,
149                 st_makeline (
150                     LAG(shape_point_geom) OVER w,
151                     shape_point_geom
152                 ) shape_line_geom,
153                 st_azimuth (
154                     LAG (shape_point_geom) OVER w,
155                     shape_point_geom
156                 ) shape_line_azimuth
157             FROM
158                 shape_linha
159             WINDOW w AS (PARTITION BY (file_date, bus_line_id, shp) ORDER BY id)
160         ) AS q1
161     WHERE
162         shape_line_azimuth IS NOT NULL
163 ),
164 pontos_linha_and_azimuths AS (
165     SELECT
166         *
167     FROM
168         (
169             SELECT
170                 pl.*,
171                 sa.id,

```

```

172     sa.shp,
173     shape_line_geom,
174     shape_line_azimuth,
175     st_distance (bus_stop_point_geom, shape_line_geom),
176     ROW_NUMBER() OVER (
177         PARTITION BY pl.file_date, pl.file_index
178         ORDER BY
179             st_distance (bus_stop_point_geom, shape_line_geom) ASC
180     )
181 FROM
182     pontos_linha pl
183     JOIN shapes_and_sentidos ss ON ss.bus_line_id = pl.bus_line_id
184     AND ss.way = pl.way
185     JOIN shapes_and_azimuths sa ON sa.bus_line_id = ss.bus_line_id
186     AND sa.shp = ss.shp
187 WHERE
188     pl.bus_line_id IN (
189         SELECT
190             bus_line_id
191         FROM
192             chosen_bus_lines
193     )
194 ) AS q1
195 WHERE
196     row_number = 1
197 ORDER BY
198     bus_line_id ASC,
199     way ASC,
200     seq ASC
201 )
202 SELECT
203     *
204 FROM
205     pontos_linha_and_azimuths
206 ;

```

**ANEXO B – Algoritmo de descoberta de passagem dos ônibus por seus pontos**

```

1 -- CREATE TABLE nome_da_sua_tabela AS
2 WITH
3 chosen_bus_lines AS (
4     SELECT
5         *
6     FROM
7         (
8             VALUES
9                 ('216')
10        ) bus_lines (bus_line_id)
11 ),
12 chosen_dates AS (
13     SELECT *
14     FROM (
15     VALUES
16         ('2021-03-25'::DATE)
17     ) AS d(file_date)
18 ),
19 veiculos AS (
20     SELECT
21         -- codigo da linha de ônibus
22         bus_line_id,
23         -- codigo do veiculo
24         vehicle_id,
25         -- data hora da mensuração
26         timestamp,
27         -- localização
28         bus_location_point_geom geom,
29         -- data do arquivo
30         file_date
31     FROM
32         veiculos_2021_03_25_bus_line_216
33 ),
34 pontos_linha_and_azimuths AS (
35     ...
36 ),
37 -- ***** VEICULOS WITH AZIMUTHS *****
38 veiculos_with_azimuth AS (
39     select
40         file_date,
41         bus_line_id,
42         vehicle_id,
43         LAG(timestamp) OVER w AS prev_dthr,
44         timestamp,
45         timestamp - LAG (timestamp) OVER w time_dif,
46         st_makeline (LAG (geom) OVER w, geom) AS trajectory_line,
47         st_azimuth (LAG (geom) OVER w, geom) AS trajectory_azimuth
48     FROM
49         veiculos WINDOW w AS (
50             PARTITION BY
51                 file_date,
52                 bus_line_id,
53                 vehicle_id
54             ORDER BY
55                 timestamp ASC
56         )
57     ORDER BY
58     (

```



```

59         file_date,
60         bus_line_id,
61         vehicle_id,
62         timestamp
63     ) --LIMIT 100
64 ),
65 va_pa AS (
66     select
67         va.file_date,
68         va.bus_line_id,
69         va.vehicle_id,
70         va.prev_dthr,
71         l1.bus_arrival_time,
72         va.timestamp,
73         va.time_dif,
74         va.trajectory_line ,
75         va.trajectory_azimuth,
76         file_index bus_stop_index,
77         pa.bus_stop_name,
78         pa.bus_stop_id,
79         pa.seq,
80         pa.bus_stop_group,
81         pa.way,
82         pa.bus_stop_type,
83         pa.itinerary_id,
84         bus_stop_point_geom,
85         pa.id shape_sequence,
86         pa.shp shape_id,
87         pa.shape_line_geom,
88         pa.shape_line_azimuth bus_stop_azimuth,
89         st_distance distance_from_bus_stop_to_shape,
90         l1.distance_bus_to_stop,
91         closest_point_vehicle_bus_stop,
92         ratio_closest_point_vehicle_bus_stop,
93         MIN (l1.distance_bus_to_stop) OVER w_preceding
min_distance_bus_to_stop_preceding,
94         MIN (l1.distance_bus_to_stop) OVER w_following
min_distance_bus_to_stop_following
95     FROM
96         veiculos_with_azimuth va -- 0 ônibus e o ponto de ônibus precisam ser da
mesma linha
97         JOIN pontos_linha_and_azimuths pa ON va.bus_line_id = pa.bus_line_id and
va.file_date = pa.file_date,
98         lateral (
99             select
100                 -- Calcula o ponto geográfico onde o ônibus mais perto ficou do ponto de
ônibus
101                 st_closestpoint (va.trajectory_line, pa.bus_stop_point_geom)
closest_point_vehicle_bus_stop,
102                 -- Calcula a proporção sobre a linha de trajetória do ponto geográfico onde
o ônibus mais perto ficou do ponto de ônibus
103                 st_linelocatepoint (va.trajectory_line, pa.bus_stop_point_geom)
ratio_closest_point_vehicle_bus_stop
104             ) l0,
105         LATERAL (
106             SELECT
107                 -- Calcula a distância entre o ônibus e o ponto de ônibus
108                 st_distance (

```

```

109         l0.closest_point_vehicle_bus_stop :: geography,
110         pa.bus_stop_point_geom :: geography
111     ) distance_bus_to_stop,
112     -- Calcula a diferença entre o azimute da trajetória e o azimute do
ponto de onibus
113     (
114         va.trajectory_azimuth - pa.shape_line_azimuth + pi () + pi () * 2
115     ) :: numeric % (pi () * 2) :: numeric - pi () angle_dif,
116     -- Calcula a estimativa do momento passado pelo onibus
117     va.prev_dthr + (va.time_dif *
l0.ratio_closest_point_vehicle_bus_stop) bus_arrival_time
118 ) l1
119 WHERE
120     TRUE -- A distância do ônibus até o ponto de ônibus precisa ser menor que 20m
121     AND distance_bus_to_stop <= 40 -- A diferença em graus entre os azimutes
precisa estar entre -45 e +45
122     AND angle_dif BETWEEN - pi () / 4
123     AND pi () / 4
124     WINDOW w_preceding AS (
125         PARTITION BY
126         va.file_date,
127         va.bus_line_id,
128         va.vehicle_id,
129         pa.bus_stop_id,
130         pa.seq
131         ORDER BY
132             timestamp ASC RANGE BETWEEN '20 minutes' PRECEDING
133             AND CURRENT ROW EXCLUDE CURRENT ROW
134     ),
135     w_following AS (
136         PARTITION BY
137         va.file_date,
138         va.bus_line_id,
139         va.vehicle_id,
140         pa.bus_stop_id,
141         pa.seq
142         ORDER BY
143             timestamp ASC RANGE BETWEEN CURRENT ROW
144             AND '20 minutes' FOLLOWING EXCLUDE CURRENT ROW
145     )
146 ),
147 chegadas AS (
148     SELECT
149         *
150     FROM
151         va_pa
152     WHERE
153         TRUE
154         AND distance_bus_to_stop < COALESCE(min_distance_bus_to_stop_preceding,
'+Infinity')
155         AND distance_bus_to_stop <= COALESCE(min_distance_bus_to_stop_following,
'+Infinity')
156     ORDER by
157         file_date,
158         bus_line_id,
159         vehicle_id,
160         timestamp
161 )

```

```
162 SELECT
163     *
164 FROM
165     chegadas
166 ;
```