

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RICARDO ANTONIO CERVO

**Geração Automática de Mapas
Esquemáticos para o Transporte Público
da Cidade de Porto Alegre**

Monografia apresentada como requisito
parcial para a obtenção do grau de Bacharel
em Ciência da Computação

Orientador: Prof. Dr. Lucas Mello Schnorr

Porto Alegre
2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Ciência de Computação: Prof. Carlos Arthur Lang
Lisbôa

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço ao meu pai Roque Antonio Cervo (*in memoriam*), e à minha mãe Inês Benatti Cervo, por terem sempre me apoiado e me incentivado a seguir em frente; por terem me proporcionado a melhor educação possível, me ensinando sobre valores e virtudes pelos quais devo lutar; por serem pessoas boas e por toda a dedicação que tiveram para comigo; e, também, por todos os bons momentos que tivemos juntos dos quais terei sempre ótimas lembranças.

Aos meus irmãos Mauro André Cervo, Mauri Vianei Cervo e Sandro Volnei Cervo, por todo o empenho que tiveram em me ajudar sempre que precisei; e por estarem sempre presentes em minha vida, tornando menos árduo o caminho que tive de percorrer até aqui. Agradeço, também, ao Mauro, por ter me apresentado o primeiro computador quando eu tinha oito anos, despertando minha paixão por computação desde então.

À minha querida amiga Eneida Rocha, por todo o carinho, amizade, compreensão e parceria; pelos importantes conselhos e pelas longas conversas que me permitiram enxergar através das dificuldades; e, mais ainda, por todos os momentos felizes que tivemos juntos e todas as memórias que construímos. Obrigado, querida.

Ao meu amigo de longa data Fernando Varella, por ter me ajudado com bons conselhos de veterano durante meus primeiros anos na universidade; por ter me indicado à vaga de meu primeiro estágio e por ser um amigo sempre presente.

A todos os amigos que estiveram presentes em minha vida, tornando-a mais agradável.

Agradeço ao professor Lucas Schnorr, por ser um excelente orientador e pela agilidade em responder às minhas dúvidas, tanto presencialmente quanto por email.

Por fim, agradeço a todos os professores que eu tive nesta universidade, por toda a dedicação profissional, com a entrega de ensino com nível de excelência.

RESUMO

Em Porto Alegre ainda não existem mapas que mostram em uma disposição esquemática as informações sobre os trajetos do transporte público. Tais mapas são comuns em várias cidades europeias, asiáticas e norte-americanas. A disponibilidade de mapas esquemáticos é importante para cidades grandes, pois nesse tipo de mapa são exibidas apenas as informações mais relevantes sobre o sistema de transporte público urbano, tornando mais fácil a visualização dos trajetos das linhas de ônibus da cidade. Mapas esquemáticos podem ser impressos e fixados em paradas de ônibus, dentro dos ônibus, e em diversos locais pela cidade, tornando assim mais democrático o acesso a essas informações. Hoje ainda não existe uma ferramenta que permita a geração automática desse tipo de mapa para Porto Alegre. Este trabalho apresenta portanto a implementação de um sistema para geração automática de mapas de transporte público para a cidade de Porto Alegre. Através do sistema implementado é possível gerar dois tipos de mapas esquemáticos: Mapa da Linha e da Parada. O Mapa da Linha exibe todas as paradas do trajeto de ida e volta de uma linha de ônibus, exibindo os endereços de todas as paradas, e também os trechos onde a ida é diferente da volta. O Mapa da Parada exibe todas as linhas que passam em uma parada de ônibus, possibilitando verificar quais destinos são alcançáveis através de uma parada de ônibus da cidade. Os algoritmos desenvolvidos utilizam dados relevantes para geração de mapas de transporte público encontrados em dois portais públicos da cidade: O DataPoa e o ObservaPoa. Os mapas são gerados em formato SVG, possibilitando que os mapas possam ser redimensionadas sem sofrer distorções. Portanto, a futura impressão dos mapas pode ser realizada em qualquer tamanho. A construção dos mapas envolveu a utilização de tecnologias de geoprocessamento, banco de dados e programação orientada a objetos.

Palavras-chave: Transporte público. linhas de ônibus. mapas esquemáticos. Porto Alegre. GTFS.

Automatic Generation of Schematics Maps for Porto Alegre's Public Transportation System

ABSTRACT

In Porto Alegre there are still not many public transportation maps such as those used in several European, Asian and North American cities, which aim to show in a schematic layout the information about the paths of the cities' bus lines or metro. The availability of schematic maps is important for large cities, because this map type displays only the most relevant information of urban public transportation system, simplifying the visualization of the paths of the city's bus lines. Schematic maps can be printed and placed on bus stops, within the buses, and at various locations throughout the city, making more democratic the access to this information. Today there is still no tool that allows automatic generation of this type of map for Porto Alegre. This work therefore presents the implementation of a system for automatic generation of public transportation maps for the city of Porto Alegre. Through the implemented system it is possible to generate two types of schematic maps: Map of Route and Map of Stop. Map of Route displays the addresses of all the stops, and also the stretches where the path of one direction is different to the opposite direction. Map of Stop displays all routes that pass in a bus stop, making it possible to check which destinations are reachable through a city bus stop. The developed algorithms use relevant data to generate public transport maps found in two web portals of the city: DataPoa and ObservaPoa. The maps are built in scalable vector graphics format (SVG), which enables the images of the maps can be resized without suffering distortions. Therefore, future printing of maps may be performed at any size. The construction of maps involved the use of technologies in geoprocessing, database and object-oriented programming.

Keywords: Public transportation. bus routes. schematics maps. Porto Alegre. GTFS.

LISTA DE FIGURAS

Figura 1.1	Fragmento 1 do Mapa Oficial de Porto Alegre – Linhas de Ônibus ..	18
Figura 1.2	Fragmento 2 do Mapa Oficial de Porto Alegre – Legendas	19
Figura 1.3	TubeMap - O mapa esquemático do transporte público de Londres.	20
Figura 2.1	Diferença entre imagem <i>raster</i> e imagem vetorial, quando ampliadas	27
Figura 3.1	Interface de usuário do gerador de mapas desenvolvido.	30
Figura 3.2	Esquema de Funcionamento Geral do Sistema.....	31
Figura 3.3	Diagrama ER da base de dados construída	34
Figura 3.4	Exemplo de Mapa da Linha	36
Figura 3.5	Mapa da Parada com Rastros GPS.....	38
Figura 3.6	Deslocamento das linhas sobrepostas no mapa de parada com GPS.	40
Figura 3.7	Mapa da Parada sem Rastros GPS	41
Figura 3.8	Deslocamento de trechos sobrepostos	42
Figura 4.1	Exemplo de Mapa de Linha com Problemas	46
Figura 4.2	Histograma da quantidade de paradas dos mapas de linha.....	47
Figura 4.3	Mapa da Linha com 329 paradas, a maior quantidade dentre os mapas gerados.....	48
Figura 4.4	Distribuição da quantidade de linhas por parada da cidade.....	50
Figura 4.5	Exemplo de mapa bem sucedido com poucas linhas.....	51
Figura 4.6	Mapa malsucedido: anomalia no desenho	52
Figura 4.7	Mapa malsucedido: muitas linhas dificultam compreensão	53
Figura 4.8	Trecho com muitas paradas aglomeradas dificultando a compre- ensão	54
Figura 4.9	Traços deslocados com padrão zigue-zague e cruzamentos.....	55
Figura 4.10	Traços desconectados quando há muitas linhas sobrepostas.....	57
Figura 4.11	Muitas paradas aglomeradas dificultam compreensão	57
Figura 4.12	Inversão dos traços no desenho.....	58
Figura 4.13	Geração de mapas octalineaes com programação inteira mista.....	60
Figura 4.14	Geração de mapas octalineaes com algoritmo genético	61
Figura C.1	Brooklyn Bus Map - Mapa esquemático das linhas de ônibus do Brooklyn	87
Figura C.2	Rede Integrada do Transporte Público de Curitiba	88
Figura C.3	Mapa do Transporte Metropolitano (São Paulo).....	89

LISTA DE TABELAS

Tabela 2.1	Especificação geral de uma base GTFS.....	24
Tabela 3.1	Resultado obtido ao rodar o código SQL do Exemplo 3.1	35
Tabela B.1	Especificação do Arquivo routes.txt	73
Tabela B.1	Especificação do Arquivo routes.txt (Continuação)	74
Tabela B.1	Especificação do Arquivo routes.txt (Continuação)	75
Tabela B.1	Especificação do Arquivo routes.txt (Continuação)	76
Tabela B.2	Especificação do Arquivo stops.txt.....	77
Tabela B.2	Especificação do Arquivo stops.txt (Continuação).....	78
Tabela B.3	Especificação do Arquivo trips.txt	78
Tabela B.3	Especificação do Arquivo trips.txt (Continuação)	79
Tabela B.3	Especificação do Arquivo trips.txt (Continuação)	80
Tabela B.3	Especificação do Arquivo trips.txt (Continuação)	81
Tabela B.4	Especificação do Arquivo stop_times.txt	82
Tabela B.4	Especificação do Arquivo stop_times.txt (Continuação).....	83
Tabela B.5	Especificação do Arquivo shapes.txt	84
Tabela B.5	Especificação do Arquivo shapes.txt (Continuação)	85

LISTA DE ABREVIATURAS E SIGLAS

SVG	Scalable Vector Graphics
PDF	Portable Document Format
HTML	HyperText Markup Language
XHTML	eXtensible Hypertext Markup Language
CSV	Comma-separated values
GTFS	General Transit Feed Specification
GPS	Global Positioning System
SQL	Structured Query Language
GIS	Geographic Information System
SHP	Shape Format
SHX	Shape Index Format
DBF	dBase Format
PRJ	Projection Format
ER	Entidade Relacionamento
WGS	World Geodetic System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
ID	Identificator

SUMÁRIO

1 INTRODUÇÃO	17
1.1 Objetivo do trabalho	20
1.2 Organização do documento.....	21
2 CONCEITOS BÁSICOS	23
2.1 Registros de Informações Relevantes para Mapas	23
2.2 Imagens de Saída Para Visualização Mapas	26
2.3 Aplicação dos Conceitos Básicos no Trabalho.....	27
3 GERADOR DE MAPAS DE ÔNIBUS PARA PORTO ALEGRE	29
3.1 Obtenção e Modelagem de Dados	32
3.1.1 Importação dos Arquivos GTFS para a Base de Dados.....	32
3.1.2 Cache de dados oriundos do serviço de Geocoding do Google	32
3.1.3 Importação de Shapefiles dos Bairros de Porto Alegre	33
3.2 Geração de Mapas	35
3.2.1 Mapa da Linha.....	35
3.2.2 Mapa da Parada com Rastros GPS	37
3.2.3 Mapa da Parada sem Rastros GPS.....	40
3.3 Sumário do Capítulo	44
4 RESULTADOS OBTIDOS	45
4.1 Análise dos Mapas de Linha.....	45
4.2 Análise dos Mapas de Parada Com Rastros GPS.....	48
4.2.1 Problemas graves levando a rejeição de mapas.....	49
4.2.2 Problemas básicos	51
4.3 Análise dos Mapas de Parada Sem Rastros GPS	55
4.3.1 Vantagens	56
4.3.2 Desvantagens.....	56
4.3.3 Problemas não resolvidos	56
4.4 Trabalhos Futuros.....	59
5 CONCLUSÃO	63
REFERÊNCIAS	65
APÊNDICE A — SCRIPT DE IMPORTAÇÃO DOS ARQUIVOS GTFS	67
APÊNDICE B — ESPECIFICAÇÃO DOS ARQUIVOS GTFS UTILIZADOS	73
APÊNDICE C — EXEMPLOS DE MAPAS ESQUEMÁTICOS	87

1 INTRODUÇÃO

A evolução das tecnologias da informação está tornando possível utilizar a computação para melhorar a qualidade de diversos serviços úteis à população. Um desses serviços que vem recentemente ganhando destaque no universo da computação é o transporte público nas cidades, pois já é possível efetuar consultas *online* sobre diversas informações a respeito de horários e mapas dos trajetos das linhas de ônibus urbanas. Essa evolução é possível com atitudes de transparência dos órgãos governamentais, disponibilizando informações públicas tais como rotas e frequências das linhas de transporte urbano. O desenvolvimento de ferramentas úteis na busca de informações sobre transporte público é facilitado por essas bases de dados que são disponíveis publicamente. Em Porto Alegre, por exemplo, existe o DataPoa (Procempa, 2015a). Essa base de dados aberta é mantida pela empresa Procempa (Procempa, 2015b). Lá é possível encontrar diversas informações sobre a cidade de Porto Alegre, sendo algumas delas a respeito do sistema de transporte público da cidade.

Atualmente, existe em Porto Alegre algumas ferramentas para a busca de informações a respeito do transporte público, dentre as quais podemos destacar: o sistema Viaje Trifácil (BUENO, 2014), o qual permite que uma pessoa planeje a sua viagem de ônibus dentro de Porto Alegre apenas informando os pontos de partida e destino; o aplicativo Moovit (MOOVIT, 2016) que é voltado para *mobile*, permitindo consultas a horários de ônibus e auxílio em tempo real para a realização de uma viagem de ônibus, estando disponível em diversas cidades do mundo e tendo recentemente chegado à Porto Alegre; os sites das empresas de ônibus Carris (Carris, 2015), Conorte (Conorte, 2015), STS (STS, 2015), Unibus (Unibus, 2015) onde cada empresa implementa um sistema de consulta à sua própria maneira; o site da empresa pública municipal EPTC (EPTC, 2015) que também possui alguns recursos de pesquisa *online*; e o mapa oficial da cidade disponibilizado pela prefeitura (Prefeitura de Porto Alegre, 2015), que é um documento em formato PDF contendo muitas informações sobre as linhas de ônibus da cidade.

Dentre as opções anteriormente citadas, a única que poderia ser utilizada sem acesso à Internet é o mapa oficial que, por ser um arquivo PDF, poderia ser impresso. Porém, esse mapa possui muitas informações aglomeradas em um es-

paço pequeno, o que dificulta a sua utilização. As Figuras 1.1 e 1.2 exibem dois fragmentos desse mapa. As outras opções tornam necessário que o usuário acesse o sistema *online*, o que também é difícil em diversas situações, pois é necessário um certo tempo para pesquisa e os mapas *online* também costumam mostrar muitas informações que não são relevantes para quem está apenas precisando embarcar em um ônibus.

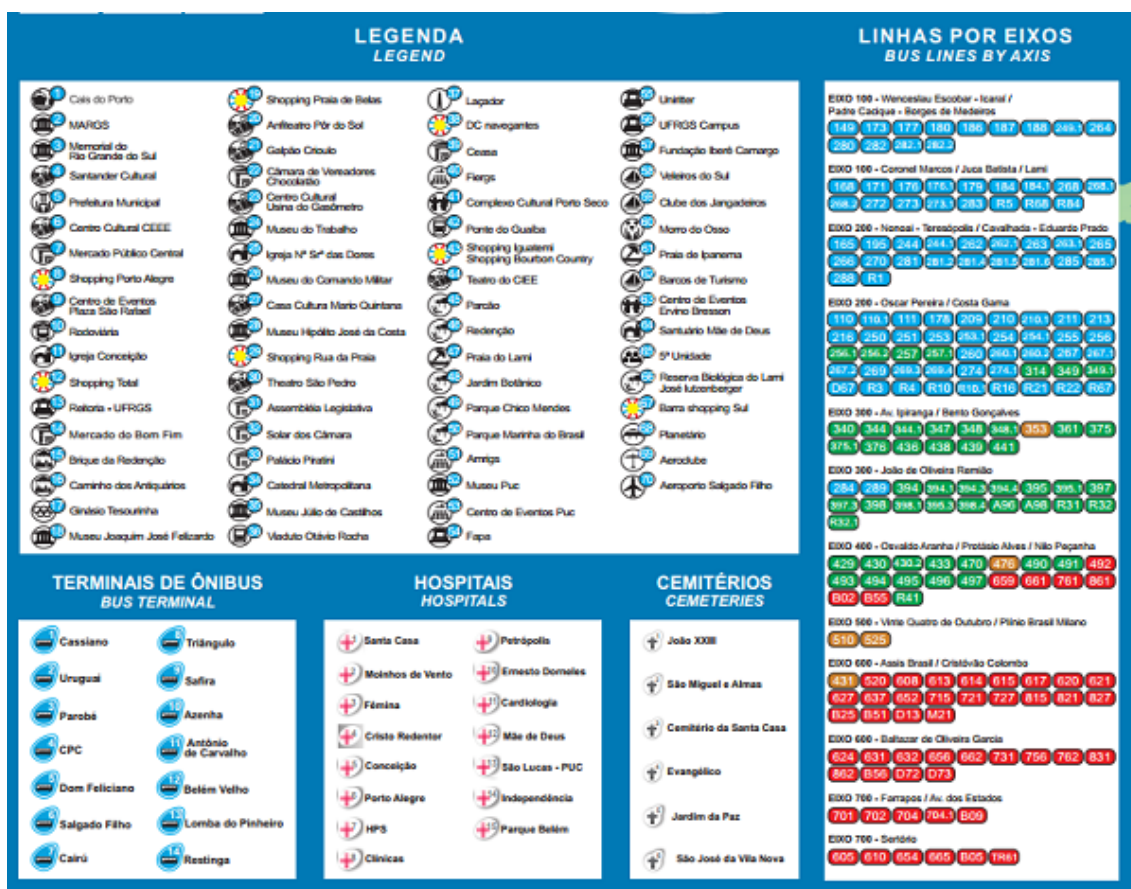
Figura 1.1: Fragmento 1 do Mapa Oficial de Porto Alegre – Linhas de Ônibus



Fonte: (Prefeitura de Porto Alegre, 2015)

Mapas esquemáticos são bastante utilizados em cidades europeias, asiáticas e norte-americanas. Em Londres, por exemplo, é encontrado o “Tube Map” (BILLSON, 2011), sendo um mapa em formato octalinear (onde as rotas são dispostas em apenas 8 direções) que exhibe a circulação das linhas de metrô da cidade.

Figura 1.2: Fragmento 2 do Mapa Oficial de Porto Alegre – Legendas

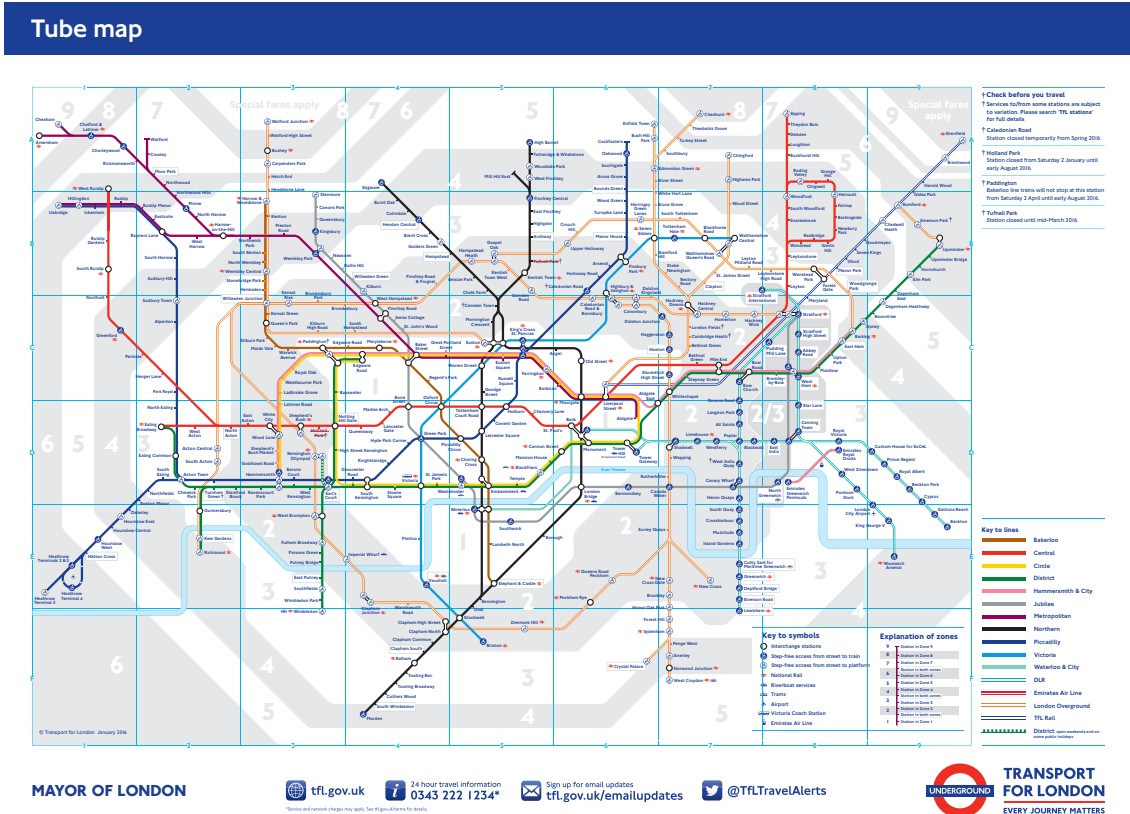


Fonte: (Prefeitura de Porto Alegre, 2015)

A Figura 1.3 exibe esse mapa Londrino. Outro exemplo de mapa esquemático – o mapa das linhas de ônibus do Brooklyn, distrito de Nova Iorque – pode ser visualizado na Figura C.1 do Apêndice C.

No Brasil já existem mapas esquemáticos disponíveis em algumas capitais. Em Curitiba existe a “Rede Integrada de Transporte Coletivo de Curitiba” (URBS, 2015) (Figura C.2 do Apêndice C). Em São Paulo existe o “Mapa do Transporte Metropolitano” (SPTrans, 2011), uma solução que lembra bastante o modelo utilizado em Londres, mostrando um resumo da circulação do metrô de toda a região metropolitana em formato octalinear (Figura C.3 do Apêndice C). Esses mapas mostram que já existe um avanço nos esforços para criação de mapas esquemáticos de transporte público no Brasil. Até onde nosso conhecimento alcança, não existe em Porto Alegre grandes esforços no sentido de se construir mapas esquemáticos do sistema de transporte público da cidade.

Figura 1.3: TubeMap - O mapa esquemático do transporte público de Londres



Fonte: (LONDON, 2016)

1.1 Objetivo do trabalho

Este trabalho tem por objetivo a implementação de um gerador automático de mapas para o transporte público de Porto Alegre. O sistema desenvolvido possibilita a geração automática de mapas em formato de imagem vetorial SVG, sendo que são propostos dois tipos de mapas:

Mapa da Linha: mapa de apenas uma linha de ônibus, o qual irá conter todas as paradas do trajeto de ida e volta de da linha selecionada. Este mapa é ideal para ser fixado no interior dos ônibus, facilitando a visualização das informações pelos passageiros durante a viagem a qualquer momento;

Mapa da Parada: mapa de todas as linhas de ônibus que passam pela parada selecionada, possibilitando que seja visualizado todos os destinos alcançáveis na cidade a partir daquela parada. Mapas desse tipo são ideais para serem fixados nas paradas de ônibus da cidade, facilitando uma rápida consulta pelas pessoas que estão na parada.

O sistema é implementado com a linguagem de programação Java. Os dados utilizados para geração dos mapas em SVG são oriundos dos arquivos em formato CSV contidos no portal do DataPoa, e estão armazenados em um banco de dados local PostgreSQL, o qual faz parte do sistema. Foi utilizada também a extensão do banco de dados PostGIS, para armazenamento e manipulação de alguns dados geográficos da cidade de Porto Alegre.

1.2 Organização do documento

Este documento está organizado da seguinte forma. O Capítulo 2 apresenta os conceitos básicos que são considerados essenciais para o bom entendimento do trabalho desenvolvido. O Capítulo 3 detalha a proposta desenvolvida para solucionar o problema da inexistência de mapas esquemáticos para a cidade de Porto Alegre: o sistema desenvolvido para geração automática de mapas. O Capítulo 4 apresenta uma análise sobre os resultados obtidos com os algoritmos desenvolvidos. Por fim, o Capítulo 5 apresenta a conclusão e uma discussão sobre possíveis trabalhos futuros.

2 CONCEITOS BÁSICOS

Neste capítulo são apresentados alguns conceitos básicos para melhor compreensão do desenvolvimento do trabalho. Inicialmente é apresentado o banco de dados com informações que podem ser utilizadas para geração de mapas; em seguida, é apresentado o conceito de imagens vetoriais para saída de dados.

2.1 Registros de Informações Relevantes para Mapas

O DataPoa é o portal dos dados abertos da cidade de Porto Alegre (Protempa, 2016). Por ele é possível encontrar os mais diversos dados sobre a cidade de Porto Alegre, como despesas públicas, turismo, medições sobre o nível do Rio Guaíba, informações sobre indústria e comércio e os dados utilizados neste trabalho: informações sobre o transporte público da cidade através de conjunto de arquivos GTFS. O formato GTFS (MCHUGH, 2013) é um padrão que foi desenvolvido pela empresa Google – sendo um acrônimo para *Google Transit Feed Specification* (GTFS) – adotado mundialmente para a modelagem dos sistemas de transporte público de cidades. O formato permite conter informações sobre horário da circulação de linhas, localizações de paradas e estações, empresas e agências responsáveis, e o relacionamento entre todas essas informações. A tabela 2.1 exibe todos os arquivos contidos em um conjunto GTFS. Para a realização deste trabalho, foi utilizado apenas um subconjunto de todo o pacote GTFS de Porto Alegre contendo apenas os seguintes arquivos: rotas (`routes.txt`), paradas (`stops.txt`), viagens (`trips.txt`), horários de passagem (`stop_times.txt`), e formas (`shapes.txt`). A especificação detalhada de cada um dos arquivos utilizados pode ser encontrada no Apêndice B. A seguir é apresentado resumidamente o conteúdo encontrado em cada um dos arquivos GTFS utilizados:

- `routes.txt` contém as linhas de ônibus da cidade.
- `stops.txt` contém as paradas de ônibus da cidade.
- `trips.txt` contém informações sobre as viagens, como horários, direção (bairro-centro, centro-bairro), entre outros. Cada viagem pertence a uma linha.

Tabela 2.1: Especificação geral de uma base GTFS

Arquivo	Obrigatório	Descrição
agency.txt	Sim	Uma ou mais agências de transporte público que fornecem os dados nesse <i>feed</i> .
stops.txt	Sim	Locais individuais em que os veículos pegam ou deixam passageiros.
routes.txt	Sim	Trajetos do Google Transit. Um trajeto é um grupo de viagens exibidas aos passageiros como um único serviço.
trips.txt	Sim	As viagens de cada trajeto. Uma viagem é uma sequência de duas ou mais paradas que ocorrem em um horário específico.
stop_times.txt	Sim	Horários de partida e chegada dos veículos em paradas específicas em cada viagem.
calendar.txt	Sim	Datas para IDs de serviço que usam uma programação semanal. Especificam quando o serviço começa e termina, bem como os dias da semana em que o serviço está disponível.
calendar_dates.txt	Não	Exceções para IDs de serviço definidos no arquivo calendar.txt. Se o arquivo calendar_dates.txt inclui TODAS as datas de serviço, ele pode ser especificado no lugar do calendar.txt.
fare_attributes.txt	Não	Informações sobre tarifas dos trajetos de uma empresa de transporte público.
fare_rules.txt	Não	Regras para implementação das informações de tarifa dos trajetos de uma empresa de transporte público.
shapes.txt	Não	Regras para desenhar linhas em um mapa para representar os trajetos de uma empresa de transporte público.
frequencies.txt	Não	Intervalo entre as viagens nos trajetos com frequência variável de serviços.
transfers.txt	Não	Regras para conexões em pontos de baldeação entre os trajetos.
feed_info.txt	Não	Informações adicionais sobre o <i>feed</i> , incluindo editor, versão e informações sobre validade

Fonte: (Google, 2012)

- `stop_times.txt` contém a ligação entre as paradas e as viagens, contendo a sequência das paradas durante uma viagem.
- `shapes.txt` contém os rastros GPS de cada viagem, sendo uma sequência de pontos com coordenadas geográficas de latitude e longitude capturados durante o trajeto da linha.

O portal ObservaPoa (Prefeitura de Porto Alegre, 2016) possui diversas informações georreferenciadas sobre a cidade de Porto Alegre. Uma dessas informações são os arquivos *shapefiles* dos bairros da cidade. O formato *shapefile* (ESRI, 1998), criado pela empresa ESRI, contém definições de três primitivas geométricas: pontos, linhas, e polígonos. Esses dados podem representar informações geoespaciais do mundo real, como rios, poços, ruas, cidades, pontos de interesse ou qualquer objeto do mundo real a ser trabalhado em um ambiente digital georreferenciado. Os *shapefiles* são utilizados em conjuntos de três arquivos obrigatórios com as extensões *Shape Format SHP*, *Shape Index Format SHX* e *dBase Format DBF*. Em situações em que se necessita informações sobre o sistema de coordenadas utilizado, como por exemplo para efetuar conversões entre diferentes sistemas de coordenadas geográficas, é também necessário um quarto arquivo com extensão *Projection Format PRJ*. A seguir estão descritos os dados encontrados em cada um dos arquivos:

- SHP: contém a lista de geometrias, representadas por um conjunto de pontos. Um arquivo desse tipo pode armazenar uma lista somente de um tipo de geometrias de cada vez, podendo ser do tipo pontos, linhas ou polígonos;
- SHX: contém um índice das geometrias, para otimizar as consultas;
- DBF: contém dados em colunas no formato dBase IV, com informações sobre as geometrias. No caso dos bairros, por exemplo, cada polígono contém uma coluna com o nome do bairro;
- PRJ: contém informação sobre o sistema de coordenadas e a projeção utilizada.

Bancos de dados geográficos tem por objetivo permitir a manipulação de objetos do tipo *geometry*, frequentemente inexistente em bancos de dados relacionais comuns. Esse tipo de dados representa as mesmas geometrias que são armazenadas em *shapefiles*, porém em um formato otimizado para as operações

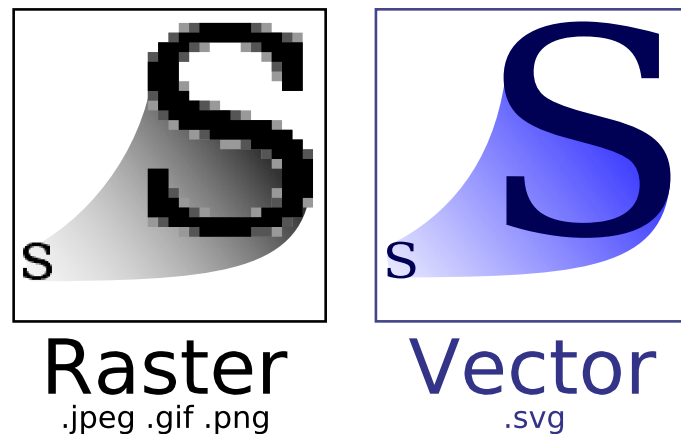
comuns de bancos de dados. O PostGIS é a extensão geográfica GIS (*Geographic Information System*) para o banco de dados PostgreSQL. Essa extensão fornece diversas funções para consulta de dados geométricos, como cálculo de área de um polígono, distância entre geometrias, cálculo do centro geométrico (centroide) de um polígono, entre outras.

2.2 Imagens de Saída Para Visualização Mapas

Em computação gráfica existem dois tipos de imagens digitais: imagem mapa de bits *raster* e imagem vetorial (ARNTSON, 2011). As imagens do tipo *raster* são mapas de *pixels*, onde a única informação contida na imagem são as cores dos *pixels* que a compõe. Dessa forma, possuem um tamanho fixo e, conseqüentemente, podem perder qualidade ao aumentar o nível de zoom, pois não há como compor a informação perdida entre *pixels* nesse processo. As imagens vetoriais, por outro lado, são construídas através da utilização de primitivas geométricas, como pontos, linhas, curvas e texto. Dessa forma os elementos componentes da imagem são desenhados no momento de sua abertura, podendo ter seu tamanho ampliado, ou reduzido, sem perda de qualidade. A utilização de imagens vetoriais é restrita a desenhos que podem ser mapeados para primitivas geométricas, como desenhos esquemáticos. No caso de uma fotografia, é inviável utilizar imagem vetorial. A Figura 2.1 ilustra as diferenças entre uma imagem *raster* (à esquerda) e uma imagem vetorial (à direita). Pelo motivo de não perder qualidade ao redimensionar, e pelo fato de mapas poderem ser definidos com formas geométricas, imagens vetoriais são ideais para desenho de mapas de alta qualidade.

Scalable Vector Graphics (SVG) (FERRAILOLO; JUN; JACKSON, 2003) é um formato de imagem vetorial de padrão aberto. Foi criado pela World Wide Consortium (W3C, 2015), comunidade internacional responsável pela definição de vários padrões para a World Wide Web, como HTML e XHTML. Seu conteúdo é descrito em linguagem XML. Arquivos no formato SVG são compatíveis com todos os navegadores de Internet mais utilizados atualmente: Mozilla Firefox, Google Chrome, Internet Explorer, Apple Safari e Opera. Para visualizar uma imagem SVG basta abrir o arquivo de imagem no navegador. Ao aumentar e reduzir o nível de zoom, ela é automaticamente redesenhada de forma a não ter perda

Figura 2.1: Diferença entre imagem *raster* e imagem vetorial, quando ampliadas



Fonte: (COMMONS, 2016)

de qualidade. Também é possível incorporar facilmente imagens SVG a páginas HTML, bastando adicionar o elemento SVG no código da página. Existe também softwares específicos que permitem editar imagens SVG, como por exemplo o Inkscape (Inkscape, 2016).

2.3 Aplicação dos Conceitos Básicos no Trabalho

Este capítulo apresentou conceitos básicos sobre itens importantes para compreensão do trabalho: os portais DataPoa e ObservaPOA; arquivos GTFS, *shapefiles* e bancos de dados geográficos; e finalmente imagens vetoriais e o formato de imagem SVG. Todos esses conceitos são aplicados no desenvolvimento do trabalho, que é detalhado no próximo capítulo.

3 GERADOR DE MAPAS DE ÔNIBUS PARA PORTO ALEGRE

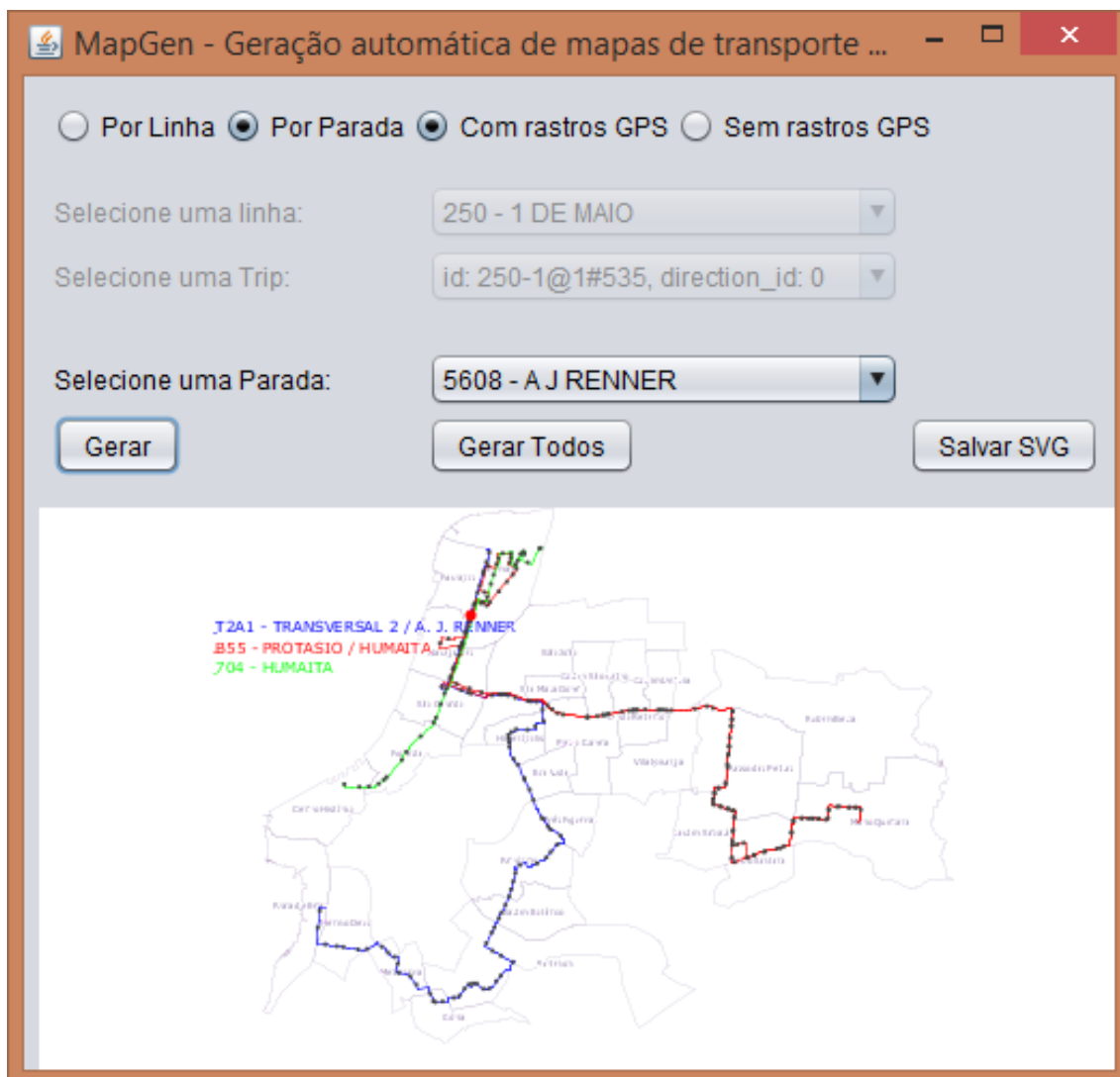
Atualmente, apesar dos avanços computacionais e da existência do portal DataPoa, ainda não existem muitos mapas esquemáticos de transporte público para a cidade de Porto Alegre. Também não existe (não foi encontrado na pesquisa deste trabalho) um programa que permita gerar automaticamente e facilmente mapas para todas as linhas e para todas as paradas de ônibus da cidade.

A proposta deste trabalho é implementar um sistema que permita gerar mapas de transporte público para a cidade de Porto Alegre de forma automática com os dados disponíveis publicamente através dos arquivos GTFS no portal DataPoa. Para isso foi desenvolvido um programa com a linguagem de programação Java, o qual possui uma interface gráfica (veja a Figura 3.1) onde o usuário do sistema pode gerar dois tipos de mapas esquemáticos:

- **Mapa da Linha:** exibe em uma disposição esquemática linear todas as paradas de uma linha de ônibus, onde é possível verificar os trechos em que o caminho de ida é o mesmo do caminho de volta. Também são exibidos os endereços de cada parada obtidos através do sistema de *geocoding* do Google, e os bairros a qual cada parada pertence, obtidos através de uma consulta à base de dados geográfica construída com os bairros da cidade de Porto Alegre.
- **Mapa da Parada:** exibe todas as linhas de ônibus que passam por uma parada, tornando possível visualizar todos os destinos possíveis através de uma parada de ônibus, um tipo de visualização que não existe atualmente em Porto Alegre. Foram feitas duas implementações desse mapa: uma delas utilizando o rastro GPS das linhas de ônibus, e outra utilizando apenas as posições geográficas de cada parada com uma linha reta entre cada par de paradas adjacentes.

Utilizando as funções geométricas do banco de dados geográfico PostGIS, foi possível capturar o nome do bairro ao qual pertence cada uma das paradas, e dessa forma desenhar os polígonos e exibir os nomes dos bairros nos mapas gerados. O funcionamento geral do sistema pode ser observado na Figura 3.2. Os passos desde a obtenção dos dados até a geração das imagens dos mapas ocorre pela seguinte sequência de ações:

Figura 3.1: Interface de usuário do gerador de mapas desenvolvido.



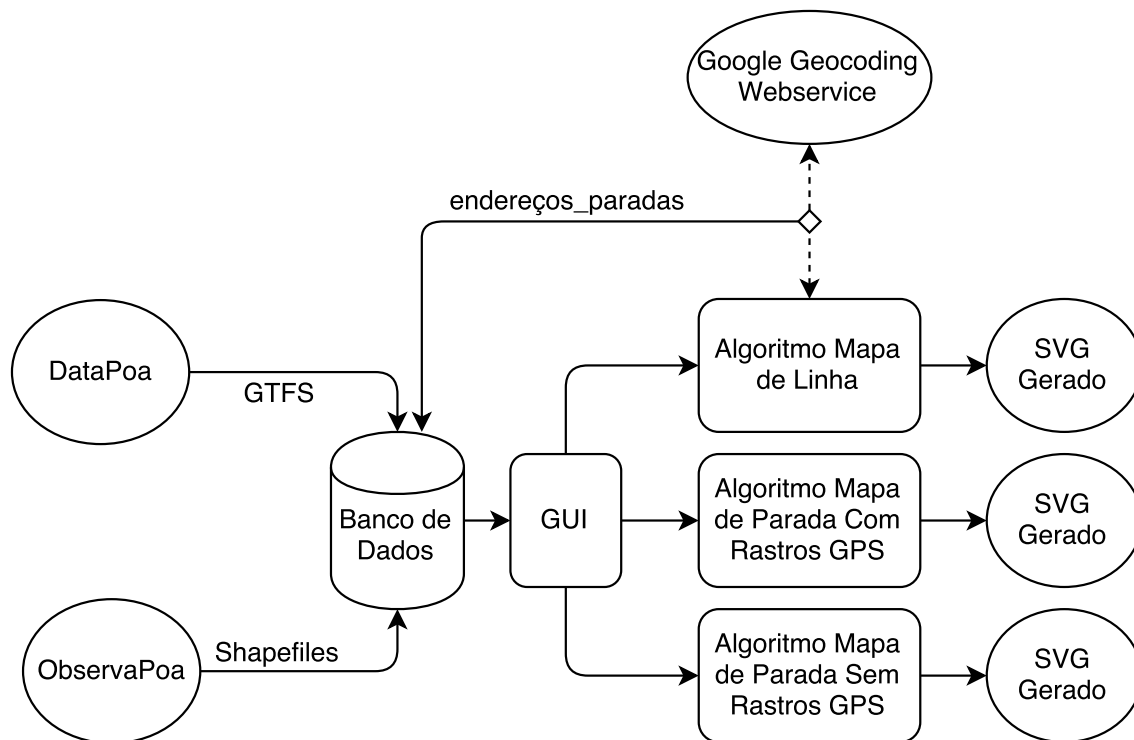
Fonte: Autor

1. Obtenção dos arquivos GTFS no portal DataPoa: deve ser feito o download do conjunto de arquivos para um diretório local. Em seguida é necessário executar o *script* de importação em linguagem SQL, que é apresentado no Apêndice A. O *script* cria todas as tabelas necessárias para o correto funcionamento da aplicação. O detalhamento da criação das tabelas, bem como todos os ajustes necessários durante a importação é apresentado na Subseção 3.1.1.
2. Obtenção dos *shapefiles* no portal ObservaPoa: deve ser feito o download do conjunto de arquivos para um diretório local. Em seguida a importação para o banco de dados deve ser feita manualmente através da ferramenta PGAdminIII (PGADMIN, 2016), para a criação da tabela `bairros` que irá

conter os nomes e os polígonos geográficos dos bairros da cidade. A ferramenta PGAdminIII possui uma funcionalidade específica para criação de tabelas geográficas a partir de *shapefiles*;

3. A execução do programa que exhibe a interface de usuário, que pode ser visualizada na Figura 3.1. A interface permite que o usuário escolha qual dos três tipos de mapas deseja gerar: Mapa da Linha, Mapa da Parada com Rastros GPS ou Mapa de Parada sem Rastros GPS. Cada mapa gerado é exibido na tela do programa e, se o usuário desejar, pode ser salvo em SVG. Também é possível gerar todos os mapas do tipo selecionado e salvar no diretório especificado.

Figura 3.2: Esquema de Funcionamento Geral do Sistema



Fonte: Autor

As próximas seções apresentam o detalhamento de todos os passos descritos anteriormente para a geração de mapas.

3.1 Obtenção e Modelagem de Dados

Esta Seção apresenta o processo de obtenção dos dados utilizados no trabalho e a construção da base de dados a partir desses dados.

3.1.1 Importação dos Arquivos GTFS para a Base de Dados

Os arquivos GTFS foram obtidos no portal DataPoa e importados para uma base de dados local onde cada arquivo foi mapeado para uma tabela relacional através de um *script* SQL. O código SQL que faz a importação dos arquivos GTFS para a base de dados pode ser encontrado no Apêndice A. Pelo diagrama ER do banco de dados, exibido na Figura 3.3, é possível verificar o encadeamento dos relacionamentos existentes entre as entidades GTFS. É notável também a ausência de chaves estrangeiras entre a tabela `shapes` com o restante da base de dados. Isso ocorreu porque existe um relacionamento `muitos para muitos` entre as tabelas `shapes` e `trips` nos arquivos GTFS, mas esse relacionamento não está normalizado. Ou seja, não existe um arquivo intermediário que poderia ser mapeado para uma tabela associativa. Devido a dificuldade em efetuar essa normalização na construção do *script*, ao fato dos registros no banco de dados serem acessados apenas para consulta e a esse problema ser facilmente contornado via código, optou-se por manter a tabela sem chaves estrangeiras explícitas, bastando compreender que existe essa relação de `muitos para muitos` entre as tabelas `shapes` e `trips`. Durante a importação também foi necessário acrescentar a coluna de chave primária para as tabelas `shapes` e `trips`, pois nos arquivos não havia uma coluna com identificador único para os registros.

3.1.2 Cache de dados oriundos do serviço de Geocoding do Google

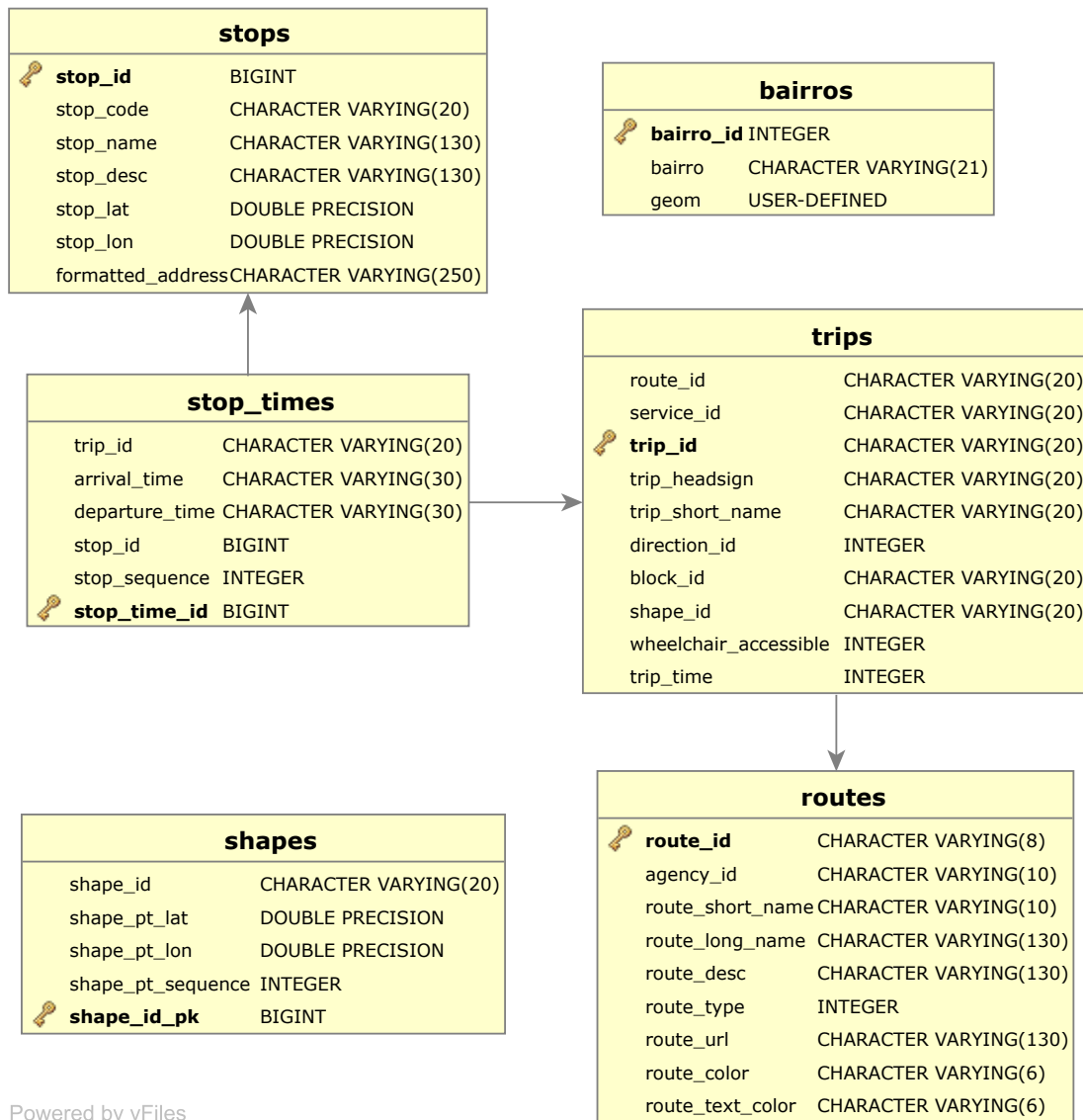
Após a execução do *script* SQL que faz a importação dos arquivos GTFS, foi adicionada a coluna `formatted_address` à tabela `stops`. Os mapas de linha – apresentados na Subseção 3.2.1 – utilizam essa coluna para salvar endereços das paradas de ônibus, conforme os mapas vão sendo gerados, pois o endereço de cada parada é exibido no mapa. A criação dessa coluna foi necessária para re-

duzir ao máximo o número de consultas ao sistema de *geocoding* do Google, pois a versão gratuita do *web service* do Google possui uma limitação de no máximo 2500 consultas por dia. O controle do número de transações para um determinado sistema nos *web services* do Google é feito por meio de uma chave gerada no portal de APIs para desenvolvedores (GOOGLE, 2016), sendo necessário passar essa chave por parâmetro a cada requisição. Por conta dessa limitação, e também do tempo médio necessário para obtenção dos endereços pela Internet (cerca de 1 segundo por endereço), o código foi escrito de forma que primeiro é buscado o endereço no banco de dados local, só acessando o serviço online quando o endereço ainda não existe localmente. Não é necessário executar *script* SQL para criar essa coluna pois a aplicação adiciona essa coluna à tabela `stops` automaticamente na primeira execução.

3.1.3 Importação de Shapefiles dos Bairros de Porto Alegre

Os *shapefiles* contendo os polígonos dos bairros da cidade de Porto Alegre foram obtidos no portal ObservaPoa (Prefeitura de Porto Alegre, 2016). A importação dos *shapefiles* para a tabela `bairros` foi realizada através da ferramenta PGAdminIII, a qual possui uma funcionalidade específica para importação de geometrias com *shapefiles*. Basta selecionar um dos três arquivos que compõe o conjunto de *shapefiles* para que a ferramenta faça a importação. Dessa forma foi criada a tabela `bairros` contendo três campos: `bairro_id`, `bairro` e `geom`, os quais contêm respectivamente a chave primária da tabela, o nome do bairro e a geometria (polígono) do bairro. Pelo diagrama ER, visto na Figura 3.3, pode-se observar que a Tabela `bairros` também não possui relacionamentos com chaves estrangeiras porque é possível obter rapidamente o bairro de uma parada através de funções espaciais do PostGIS, encontrando qual o bairro cujo polígono contém o ponto com as coordenadas geográficas da parada. O código ilustrado no Exemplo 3.1 mostra a obtenção do bairro a partir de um ponto geográfico da cidade com funções espaciais do PostGIS em SQL. A Tabela 3.1 exhibe o resultado obtido ao rodar esse código no banco de dados. Dessa forma, como cada parada possui as coordenadas geográficas armazenadas na base de dados, é possível obter o bairro a qual cada parada pertence e incluir essa informação nos mapas.

Figura 3.3: Diagrama ER da base de dados construída



Exemplo 3.1: SQL para obtenção do bairro pelas coordenadas geográficas no PostGIS

```
select bairro_id, bairro, geom
from bairros
where ST_Contains(geom, ST_MakePoint(-51.2411, -30.0834));
```

Tabela 3.1: Resultado obtido ao rodar o código SQL do Exemplo 3.1

bairro_id	bairro	geom
21	Cristal	0106000000020000000103000000010000000A000000CA (...)

Fonte: Autor

3.2 Geração de Mapas

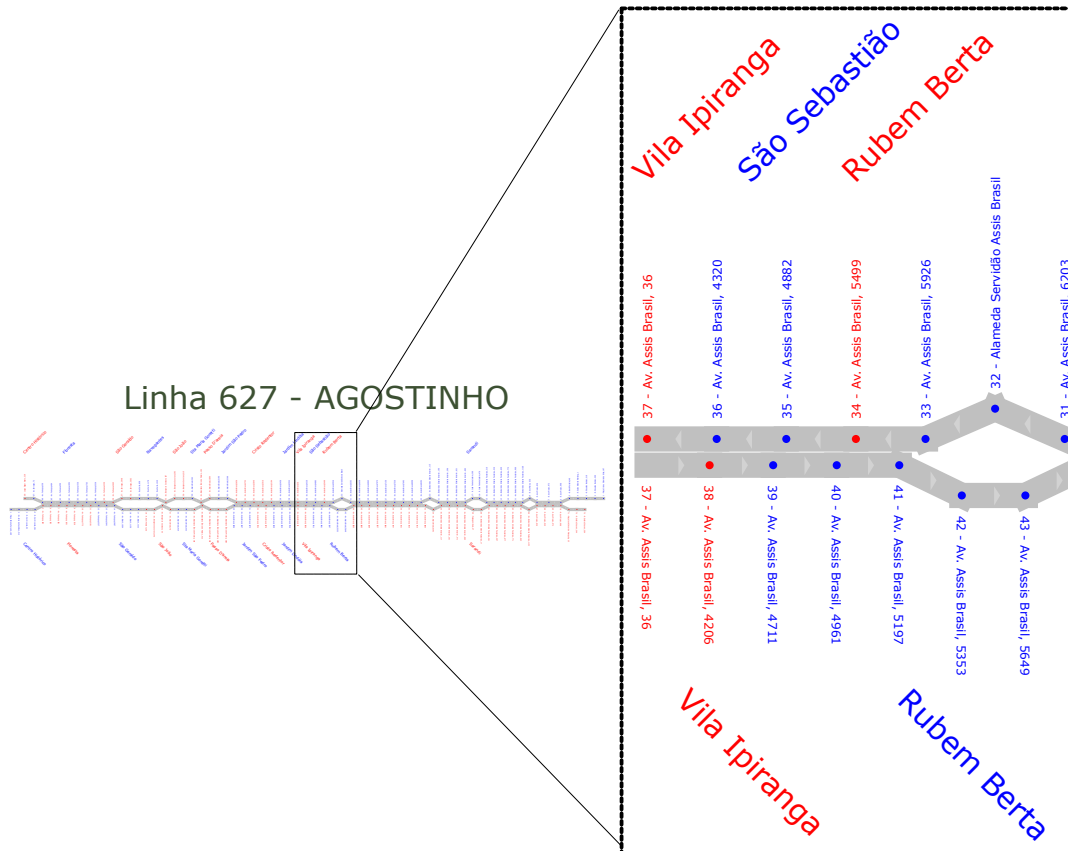
Esta seção apresenta os algoritmos desenvolvidos para a geração dos mapas de cada tipo. Para o desenho em SVG foi utilizada a biblioteca Batik (BATIK, 2011). Essa biblioteca permite utilizar uma classe padrão da linguagem Java para desenhos: a `Graphics2D`, pois o Batik possui a classe `SVGGraphics2D` que estende a classe `Graphics2D`. Dessa forma, todas as funções nativas da linguagem Java para desenhos podem ser utilizadas para a construção dos mapas, sendo que no final do processo é possível salvar a imagem para um arquivo SVG. A seguir são apresentados os diferentes tipos de mapas gerados: mapa de linha e de parada, este em duas versões.

3.2.1 Mapa da Linha

Para o desenvolvimento Mapa da Linha foram utilizados os dados importados dos arquivos GTFS, o serviço de *geocoding* do Google, e os dados da tabela geográfica `bairros`, para desenhar os bairros a que cada parada pertence. A figura 3.4 exibe um exemplar de um mapa gerado automaticamente: no lado esquerdo a visão geral, e do lado direito um detalhe com zoom.

O Algoritmo 1 apresenta os passos da construção do mapa. O funcionamento do algoritmo, em linhas gerais, pode ser compreendido da seguinte maneira: inicialmente é feita a inserção de todas as paradas de ida em uma linha

Figura 3.4: Exemplo de Mapa da Linha



Fonte: Autor

reta, da esquerda para a direita, na parte inferior. As paradas de volta que estão muito próximas de alguma parada de ida (menos que 80 metros) são inseridas no trajeto da volta, logo acima da parada de ida mais próxima. Isso foi um contorno necessário pelo fato de os arquivos GTFS não conterem a informação de quando a parada de ida e de volta são a mesma (caso dos corredores de ônibus). Pelos arquivos, todas as paradas de ida são diferentes de todas as paradas de volta. As únicas informações relevantes contidas em cada parada são os códigos (identificadores únicos), as localizações geográficas e os nome da rua a que cada uma pertence. Após a inserção das paradas de volta que seriam “as mesmas” da ida, as paradas que sobraram são inseridas entre essas paradas que já estão com a posição garantida, no trajeto da volta. Em seguida, é feito o deslocamento de trechos da região central do mapa para acima e para baixo, em pontos onde o trajeto de ida está diferente do trajeto de volta. Por fim, são escritos os nomes dos bairros e os endereços das paradas, na orientação vertical.

Algumas linhas não possuem a informação de ida e volta nos arquivos

Algoritmo 1: Mapa da Linha

- 1 *Armazena em memória a posição X de todas as paradas de ida de forma a ter todas as paradas em linha reta horizontal;*
- 2 **foreach** *parada de ida p do*
- 3 **if** *existe parada de volta p2 tal que DistanciaGeografica (p2, p) < 80 metros then*
- 4 | *armazena posição X de P2 sendo a mesma de P;*
- 5 **end**
- 6 **end**
- 7 *As coordenadas X das paradas de volta que não foram definidas são armazenadas de forma que pudessem ser inseridas entre as posições já definidas, respeitando sempre a sequencia das paradas;*
- 8 *Faz o ajuste das coordenadas Y de todas as paradas: quando uma parada é comum para ida e volta, desloca a coordenada Y para que não fique no eixo central. Uma parada P é considerada comum de ida e volta se a coordenada X mais próxima das paradas do lado o oposto pertencer a uma parada P2 que esteja na mesma rua de P, e se a coordenada X' mais próxima das paradas do sentido oposto de P2 também estiver na mesma rua de P2;*
- 9 *Desenha ligando todos os pontos de acordo com as coordenadas X e Y armazenadas anteriormente;*
- 10 *Escreve o nome dos bairros e os endereços de cada parada;*

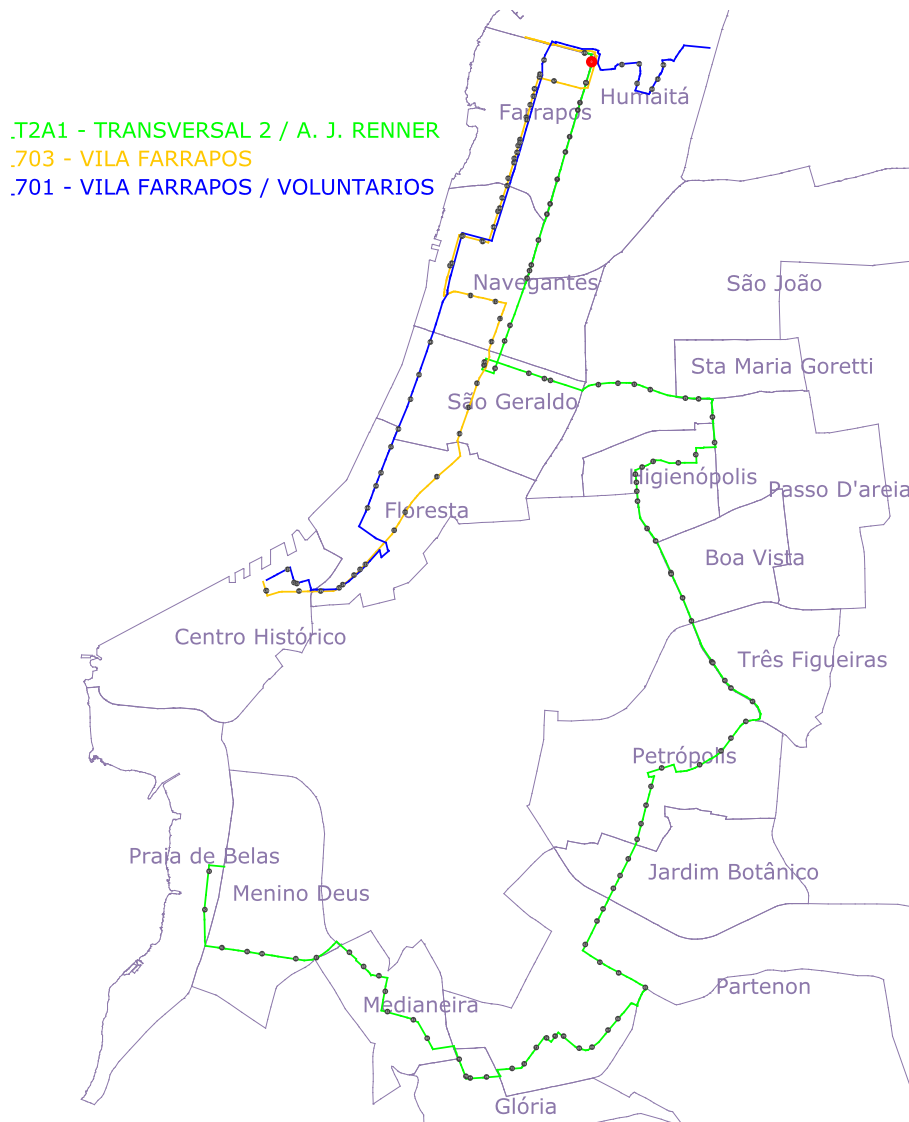
GTFS, seja porque se tratam de linhas circulares, ou porque os arquivos estão com essa informação faltando. A inexistência dessa informação em parte dos arquivos tornou possível que fossem gerados apenas 219 mapas, de um total de 386 linhas de ônibus contidas nos arquivos, o que representa 56% do total de linhas da cidade tendo mapas gerados. A geração de mapas esquemáticos para linhas circulares não foi investigada neste trabalho.

3.2.2 Mapa da Parada com Rastros GPS

O primeiro algoritmo para geração de um mapa de parada de ônibus foi realizado utilizando os rastros GPS das linhas de ônibus, contidos no arquivo `shapes.txt`. Os rastros GPS são pontos geográficos gravados pelas linhas de ônibus e armazenados no arquivo `shapes`. Para cada linha de ônibus é possível obter esses pontos através do relacionamento entre os arquivos `shapes` e `trips`, visto na Subseção 3.1.3. Esses mapas são, portanto, geográficos, contendo a localização das linhas e paradas de ônibus de acordo com suas coordenadas de latitude

e longitude. A Figura 3.5 exibe um exemplar desse mapa.

Figura 3.5: Mapa da Parada com Rastros GPS



Fonte: Autor

A construção do mapa da parada envolve a obtenção de uma lista de objetos do tipo `Trip`, que são registros na tabela `trips`, os quais representam o trajeto de uma linha de ônibus. Cada objeto `Trip` contém o atributo `directionId`, que possui valor binário, indicando o sentido do trajeto: centro-bairro ou bairro-centro. Existem duplicações nos arquivos GTFS dos relacionamentos entre a tabela `routes` e a tabela `trips`, o que exige que seja tratado no código a remoção dessas duplicações, para evitar que uma mesma linha seja desenhada várias vezes, desperdiçando tempo de processamento. Após o carregamento de todos os objetos `Trip`, o processo que realiza o desenho pode ser executado. Inicialmente é desenhado os polígonos de todos os bairros da cidade, contendo o nome de

cada bairro no centro geométrico (centroide) do seu polígono. Em seguida um laço efetua o ajuste dos pontos dos rastros GPS de todos os trajetos. O objetivo é fazer com que os pontos dos rastros de linhas diferentes que estejam muito próximos (menos que dois metros) sejam deslocados na latitude e longitude para que trajetos sobrepostos sejam desenhados lado a lado. O valor escolhido para o deslocamento é de 0.00035 em ambas as coordenadas. Esse valor foi definido baseado em testes para verificar quais valores apresentavam um resultado mais bem sucedido na geração dos mapas.

A Figura 3.6 exhibe o esquema de deslocamento das linhas no mapa. Após o deslocamento das linhas, é executado outro laço onde são efetuados os desenhos dos traços dos trajetos, ligando todos os pontos do rastro GPS de cada linha. O Algoritmo 2 apresenta os passos da construção do mapa.

Algoritmo 2: Mapa da Parada com Rastros GPS

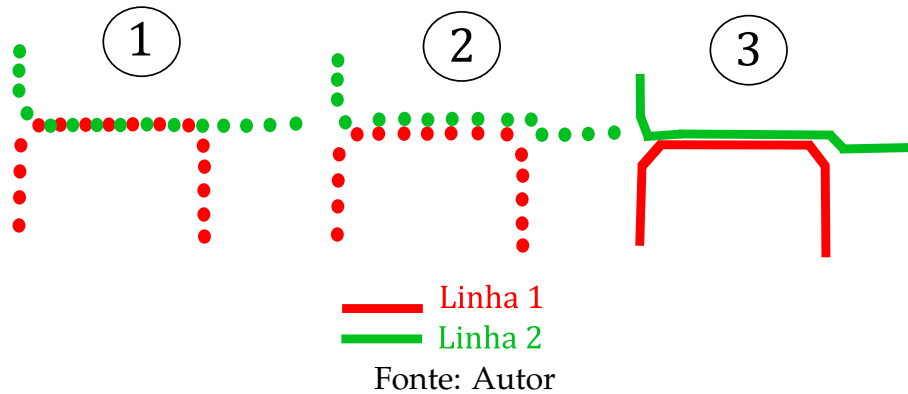
```

1 foreach Bairro b do
2   | DesenhaPoligono(b);
3   | EscreveNomeNoCentroid(b);
4 end
5 foreach Rota r que passa na parada selecionada do
6   | foreach Ponto p que compõe o rastro GPS de r do
7     | foreach Rota r2 diferente de r que passa na parada selecionada do
8       | if existe Ponto p2 do rastro GPS de r2 tal que
9         | DistanciaGeografica (p2, p) < 2 metros then
10        | | p2.x+ = 0.00035;
11        | | p2.y+ = 0.00035;
12        | end
13      | end
14    | end
15  | foreach Rota r que passa na parada selecionada do
16    | | Desenha ligando todos pontos p pertencentes a r com uma cor c;
17    | | Escreve legenda de r com o nome da linha de ônibus na cor c;
18  | end

```

Como o mapa desenvolvido apresentou alguns problemas envolvendo a utilização dos rastros GPS, principalmente em trechos com muitas linhas sobrepostas, foi testado um novo método de desenho, o qual é apresentado na próxima seção.

Figura 3.6: Deslocamento das linhas sobrepostas no mapa de parada com GPS.



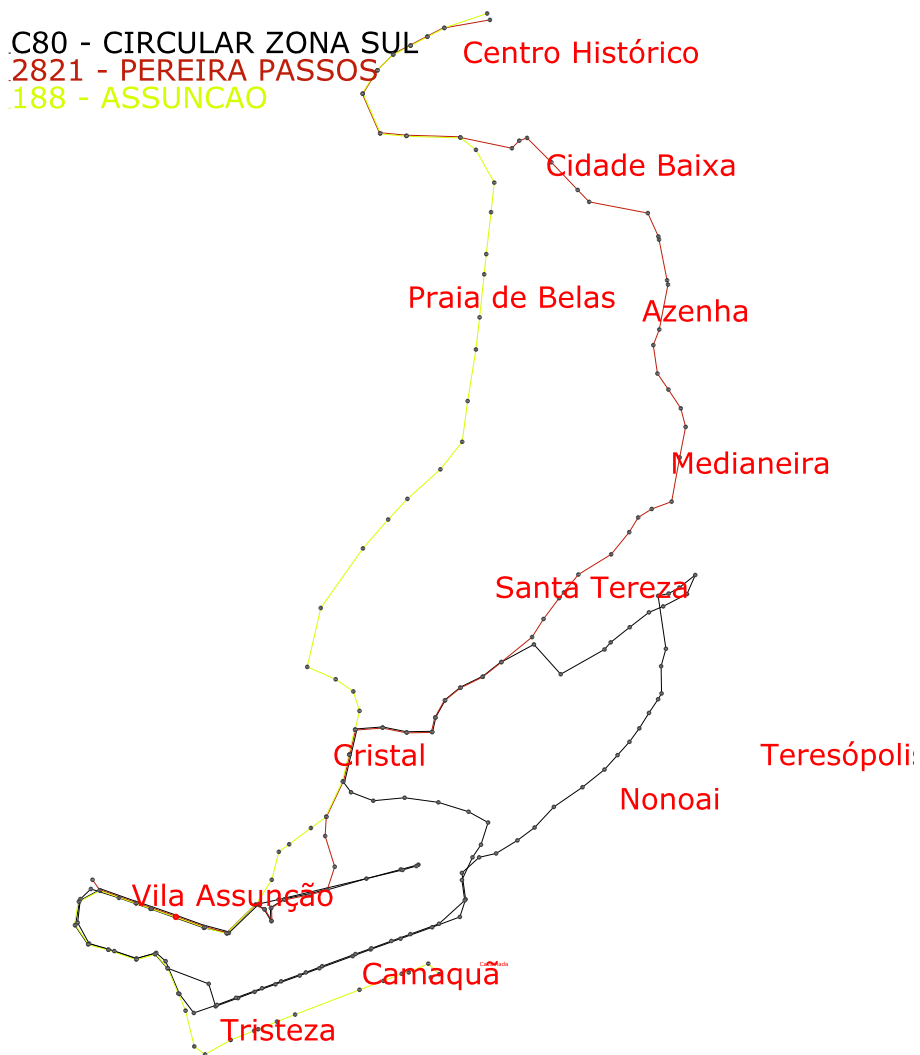
3.2.3 Mapa da Parada sem Rastros GPS

O segundo algoritmo construído para o mapa da parada é semelhante ao primeiro, com uma diferença principal que é a não utilização dos rastros GPS para o desenho das linhas, mas sim apenas a localização geográfica das paradas. Para cada linha de ônibus as rotas são desenhadas apenas ligando as paradas que fazem parte do trajeto. Um exemplo desse mapa pode ser visualizado na Figura 3.7.

A segunda diferença nessa versão – em consequência da primeira – é o deslocamento dos trechos onde as linhas se sobrepõem umas as outras, pois agora não existem mais pontos diversos de rastros GPS entre cada parada, mas sim apenas o ponto de origem e destino de cada segmento de reta a ser desenhado entre uma parada e outra. Para desenhar o deslocamento entre as linhas, foi utilizada uma técnica de álgebra linear para efetuar uma rotação de forma a ter os segmentos que ligam duas paradas posicionados na linha horizontal. Após a rotação, os segmentos sobrepostos são levemente deslocados uns em relação aos outros alterando-se apenas a coordenada y dos pontos de origem e destino. Em seguida, os segmentos são rotacionados para a inclinação original. A seguir é apresentado os passos para efetuar o deslocamento dos trechos sobrepostos.

Inicialmente é necessário encontrar o ângulo de inclinação θ dos segmentos de reta – que ligam as duas paradas – em relação ao eixo X do plano cartesiano. Posteriormente, os segmentos são rotacionados para o ângulo $-\theta$ para que fiquem em linha reta horizontal. Dados dois pontos p_1 e p_2 com as respectivas coordenadas (x_1, y_1) e (x_2, y_2) , o ângulo θ de inclinação – em relação ao eixo X

Figura 3.7: Mapa da Parada sem Rastros GPS



Fonte: Autor

do plano – do segmento que liga p_1 a p_2 é obtido por:

$$\theta = (y_2 - y_1)/(x_2 - x_1)$$

Após o cálculo da inclinação dos segmentos, é necessário encontrar as novas coordenadas (x_1', y_1') e (x_2', y_2') pertencentes respectivamente aos pontos p_1' e p_2' rotacionados ao ângulo $-\theta$. Esse procedimento é realizado através da utilização da matriz de rotação R do plano cartesiano. Dado um ângulo qualquer α , a matriz de rotação $R(\alpha)$ é calculada por:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

Em álgebra linear, cada ponto p do plano cartesiano pode ser representado como um vetor de duas dimensões contendo as coordenadas x, y , que são representadas como colunas de matrizes. Para encontrar os novos pontos $p1'$ e $p2'$ rotacionados ao ângulo $-\theta$ é necessário multiplicar a matriz contendo os pontos originais $p1$ e $p2$ pela matriz de rotação $R(-\theta)$:

$$\begin{bmatrix} x1' & x2' \\ y1' & y2' \end{bmatrix} = \begin{bmatrix} x1 & x2 \\ y1 & y2 \end{bmatrix} * R(-\theta)$$

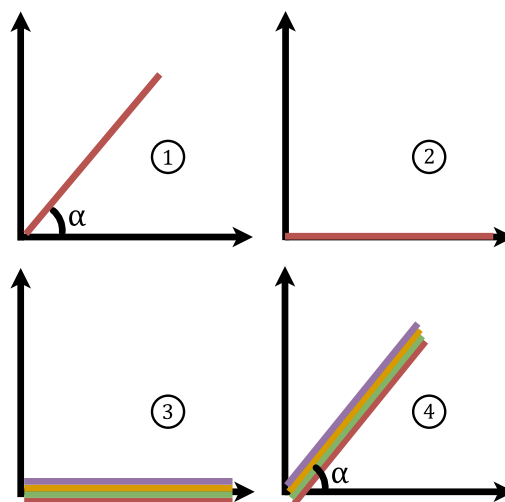
Após a multiplicação, os vetores das colunas da matriz resultante conterão os novos pontos $p1'$ e $p2'$ com as seguintes coordenadas:

$$p1 = (\cos(-\theta)) * x1 + \sin(-\theta) * y1, -\sin(-\theta) * x1 + \cos(-\theta) * y1)$$

$$p2 = (\cos(-\theta)) * x2 + \sin(-\theta) * y2, -\sin(-\theta) * x2 + \cos(-\theta) * y2)$$

Após a rotação dos segmentos para a posição horizontal, é realizado o deslocamento dos segmentos, alterando apenas a coordenada Y dos pontos, de forma a ter todos os segmentos levemente deslocados uns em relação aos outros. Por fim, o processo de rotação é revertido rotacionando-se os segmentos ao ângulo θ para que voltem a ter a inclinação original em relação ao eixo X do plano cartesiano. O algoritmo 3 apresenta os passos da construção do mapa e a Figura 3.8 ilustra o processo de deslocamento dos trechos sobrepostos.

Figura 3.8: Deslocamento de trechos sobrepostos



Fonte: Autor

Algoritmo 3: Mapa da Parada sem Rastros GPS

```

1 foreach Bairro  $b$  do
2   | DesenhaPoligono( $b$ );
3   | EscreveNomeNoCentroid( $b$ );
4 end
5 foreach Rota  $r$  que passa na parada selecionada do
6   | foreach par de paradas adjacentes  $(p_1, p_2)$  que compõe a rota  $r$  do
7     | foreach Rota  $r_2$  diferente  $r$  que passa na parada selecionada do
8       | if existe um par de paradas adjacentes  $(p_1', p_2')$  que compõe
9         |  $r_2$  tal que  $p_1' = p_1$  e  $p_2' = p_2$  then
10        |   | Rotaciona a matriz de forma que o segmento que liga
11          |   |  $p_1$  a  $p_2$  fique na posição horizontal;
12          |   | Desloca minimamente a coordenada  $y$  dos segmentos
13          |   | sobrepostos para que sejam desenhados lado a lado;
14          |   | Rotaciona a matriz de volta à posição original;
15          |   end
16        | end
17      | end
18    | end
19 end
20 foreach Rota  $r$  que passa na parada selecionada do
21   | desenha ligando todas as paradas  $p$  pertencentes a  $r$  com uma cor  $c$ 
22   | (levando em conta os deslocamentos calculados nos trechos
23   | sobrepostos);
24   | Escreve legenda de  $r$  com o nome da linha de ônibus na cor  $c$ ;
25 end

```

3.3 Sumário do Capítulo

Este capítulo apresentou todas as etapas do desenvolvimento da proposta deste trabalho: obtenção e importação para a base de dados dos arquivos GTFS e dos *shapefiles*, os quais contém respectivamente informações sobre o transporte público de Porto Alegre e os polígonos dos bairros da cidade. Todas essas informações foram utilizadas para a geração dos mapas através dos três algoritmos desenvolvidos: Mapa da Linha, Mapa da Parada Com Rastros GPS e Mapa da Parada Sem Rastros GPS. No próximo capítulo serão exibidos os resultados obtidos através dos algoritmos desenvolvidos, bem como o detalhamento das vantagens e desvantagens dos métodos utilizados na construção de cada tipo de mapa.

4 RESULTADOS OBTIDOS

Após o desenvolvimento dos algoritmos para geração dos mapas esquemáticos, vistos no capítulo anterior, foi realizada a geração de todos os mapas possíveis com o programa desenvolvido. Em seguida, foi analisado cada um dos mapas gerados para verificar quais ficaram aceitáveis a partir de parâmetros pré-definidos. Para facilitar o processo de análise, após a geração em formato SVG todas as imagens foram convertidas para o formato de imagem *raster* PNG. Este capítulo irá apresentar um resumo dos dados obtidos após a análise de todos os mapas gerados, começando pelos mapas de linha e, em seguida, os mapas de parada.

4.1 Análise dos Mapas de Linha

A análise dos mapas foi conduzida manualmente através da observação por um humano. Os parâmetros de aceitação de cada mapa de linha gerado pelo sistema são os seguintes:

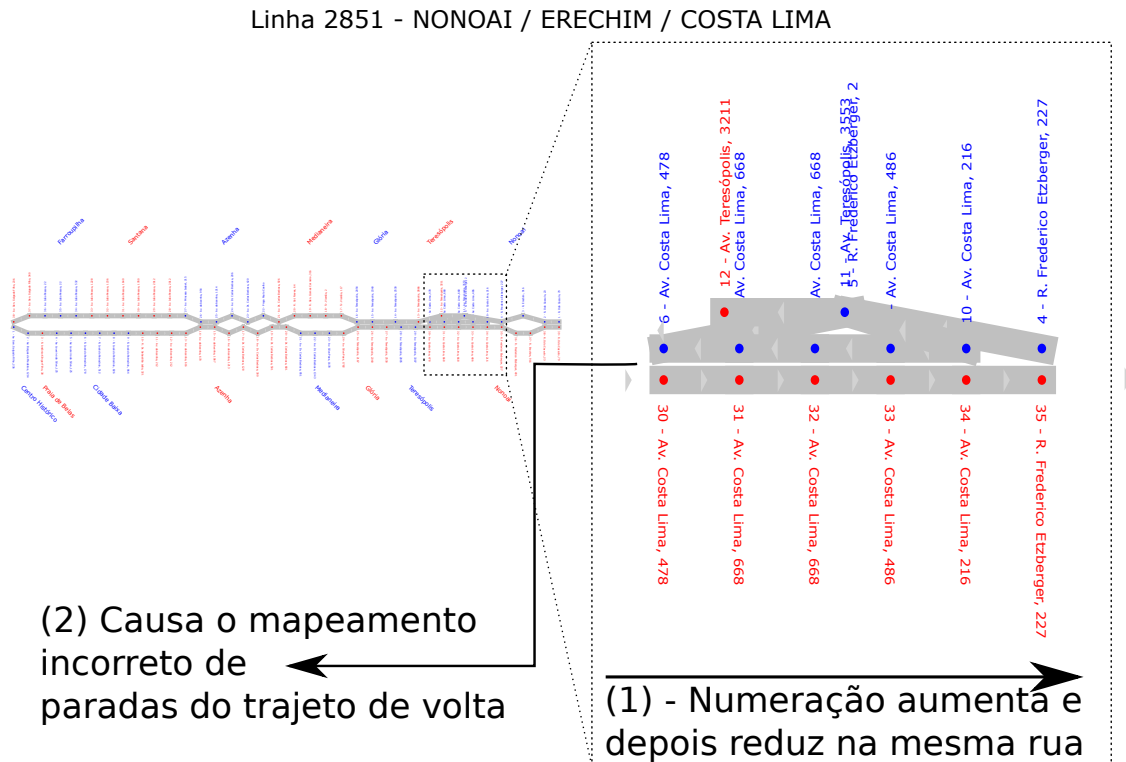
1. O mapa deve respeitar o objetivo principal definido. Deve portanto ter uma linha central que indique o caminho de ida, na parte inferior do mapa, e outra linha central que apresenta o caminho de volta, na parte superior.
2. Em trechos onde o caminho de ida é diferente do caminho de volta, deve haver um distanciamento entre as duas linhas centrais.
3. Em pontos onde as linhas caminham juntas, deve ser verificado se os endereços das paradas de ida e volta que estão próximas no desenho também são próximos geograficamente, ou seja, na mesma rua e com numeração não distante.

Análise Visual dos Mapas

Após a análise visual dos mapas de linha, foi verificado que dos 219 mapas gerados, 199 foram considerados aceitos, o que significa, portanto, uma taxa de aceitação de 90.87%. Dentre os mapas rejeitados, o principal problema ocorreu com anomalias no desenho das paradas da volta. A Figura 4.1 exibe um dos

mapas que foi rejeitado na análise.

Figura 4.1: Exemplo de Mapa de Linha com Problemas



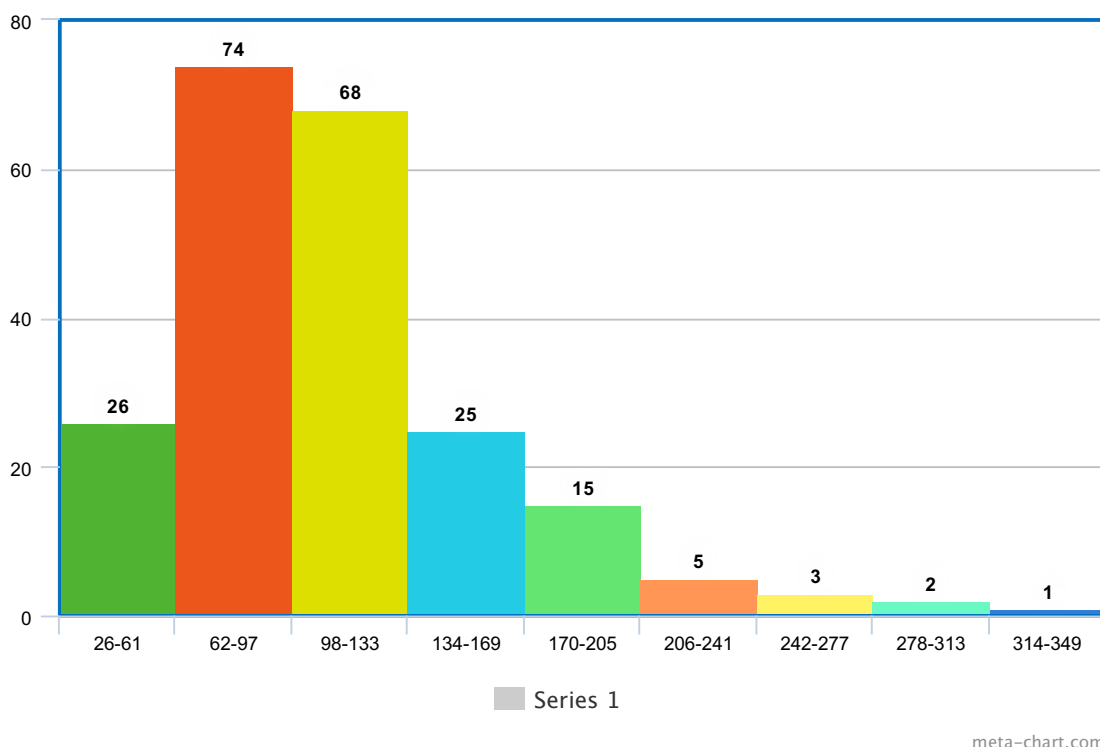
Fonte: Autor

Pela imagem é possível verificar que no trecho após a quarta parada da volta ouve uma quebra na sequência do desenho. Esse problema ocorreu em todos os mapas rejeitados. Esse mau comportamento ocorre em mapas em que o ônibus retrocede um pouco em uma mesma rua, mas mantendo o sentido do trajeto. Por exemplo, a linha 2851 - Nonoai / Erechim / Costa Lima possui um retorno na avenida Costa Lima. Como após esse retorno a trip ainda não encerra (o sentido do trajeto ainda não inverte de acordo com o GTFS) o mapa é desenhado em linha reta para a direita fazendo com que a numeração aumente e em seguida se reduza na parte inferior do mapa. Dessa maneira, ocorre o mapeamento das paradas do trajeto de volta de forma incorreta, causando a quebra no desenho das paradas do sentido oposto. Esse problema ocorre porque não foi previsto que poderia ocorrer esse tipo de situação, em que uma linha de ônibus faça parte do trajeto em direção oposta em uma mesma rua, mas mantendo o sentido principal da viagem (por exemplo, centro-bairro).

Dificuldades pela Quantidade de Paradas

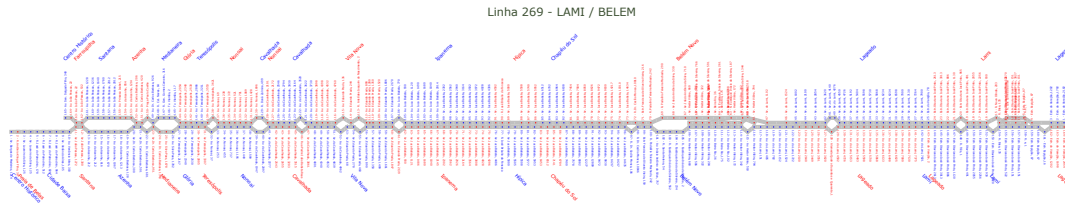
A principal dificuldade no desenvolvimento de uma solução para a geração do mapa de uma linha de ônibus para Porto Alegre é a existência de muitas paradas no trajeto, tornando desafiador encontrar uma maneira de dispor essas informações de forma legível no mapa automaticamente. A Figura 4.2 exibe um histograma da quantidade de paradas dos mapas gerados. Pelo histograma é possível verificar que apenas 26 mapas gerados possuem menos que 62 duas paradas. A Figura 4.3 exibe o exemplo do pior caso nesse quesito, ou seja, o mapa com a maior quantidade total (ida e volta) de paradas: 329. Além do pior caso em número de paradas, esse mapa também apresentou o problema da quebra na sequência das paradas da volta, pois também possui a característica vista anteriormente de a linha trafegar para o sentido oposto em uma mesma rua, sem inverter o sentido principal do trajeto.

Figura 4.2: Histograma da quantidade de paradas dos mapas de linha



A melhor solução encontrada para contornar o problema das muitas paradas foi escrever os endereços das paradas na orientação vertical. Apesar de dificultar um pouco a leitura, dessa forma é possível colocar por extenso o endereço exato de cada parada, facilitando a obtenção de informações mais precisas.

Figura 4.3: Mapa da Linha com 329 paradas, a maior quantidade dentre os mapas gerados



Possíveis Melhorias em Trabalhos Futuros

Dentre as possíveis melhorias futuras para o trabalho realizado, uma delas seria tentar deixar mais limpo o desenho através da realização de um agrupamento dos endereços que são da mesma rua, e em cada grupo exibir apenas os números do endereço. Uma outra melhoria futura também pode ser o acréscimo de informações de pontos de interesse ao longo de uma linha, como praças, shoppings, hospitais e pontos turísticos diversos. Os pontos de interesse podem ser obtidos de forma gratuita através do OpenStreetMap (HAKLAY; WEBER, 2008), que é uma plataforma web de código aberto desenvolvida colaborativamente, para visualização de mapas e fornecimento de ferramentas para desenvolvimento de soluções em geoprocessamento. O OpenStreetMap permite o *download* de *shapefiles* de toda a sua base de dados ou de parte dela. Dessa forma, portanto, pode ser feito o *download* dos *shapefiles* dos pontos de interesse de Porto Alegre e a respectiva importação para a base de dados do PostGIS, da mesma forma que foi realizada a importação dos bairros da cidade, vista na Seção 3.1.3. Na versão atual ainda não foram incluídos os pontos de interesse por ser necessário primeiro resolver o problema das muitas paradas.

4.2 Análise dos Mapas de Parada Com Rastros GPS

A análise de todos os Mapas de Parada da primeira versão foi baseada na seguinte premissa: dada uma parada de ônibus, é necessário ter o traço do trajeto de todas as linhas que passam naquela parada – uma em cada cor – visíveis no mapa, sendo possível verificar todos os destinos alcançáveis a partir da parada inicial. A parada inicial é um círculo pintado de vermelho e as demais paradas são círculos pretos.

Na análise dos mapas não foi considerado o problema das legendas sobrepondo parte do mapa, pois ele faz parte do problema de melhor disposição de legendas, que requer investigação em trabalhos futuros, não ferindo a lógica principal do desenho dos mapas, que é o objetivo principal do algoritmo neste momento.

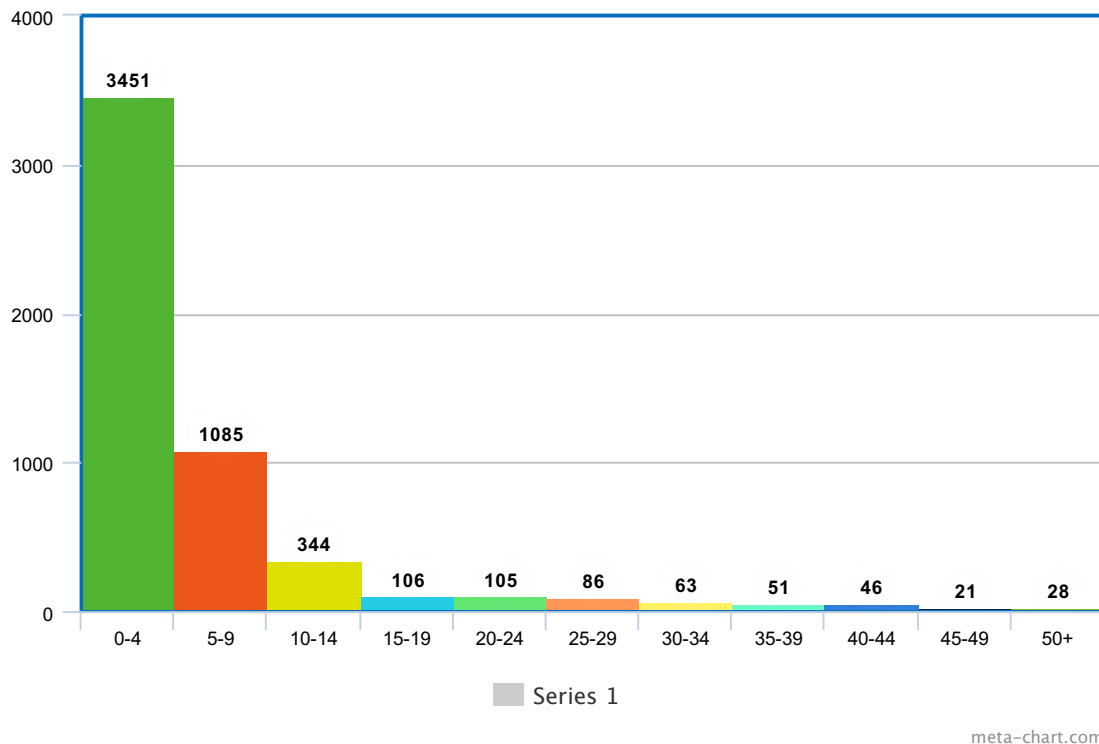
Os resultados obtidos com a análise dos mapas de parada com rastros GPS de acordo com os parâmetros definidos são os seguintes: foram gerados 5386 mapas, sendo um mapa para cada parada da cidade; desse total, 4626 foram considerados como aceitos, tendo, portanto, uma taxa de sucesso de 85.88%; dentre os mapas de parada aceitos, as principais características observadas são a pouca quantidade de linhas, na maioria das vezes não ultrapassando mais do que 10, sendo os mapas mais interessantes, do ponto de vista de facilidade de compreensão, os com no máximo 5 linhas. A Figura 4.4 exibe o histograma dos mapas de parada gerados, onde é possível visualizar a distribuição da quantidade de linhas por mapa. Pelo histograma é possível verificar que 4536 mapas possuem menos que 10 paradas. A geração do histograma foi feita depois da análise humana dos 5386 mapas, e é possível verificar que a porcentagem de mapas com menos de 10 paradas 84.2% é muito parecida com a taxa de sucesso da análise de 85.88%.

A Figura 4.5 exibe um mapa que foi eleito um dos melhores gerados. Nesse mapa é possível visualizar claramente o traço de todas as linhas, pois o método de deslocamento de linhas sobrepostas funcionou bem. A pouca quantidade de linhas também facilita uma melhor distinção dos traços, pois não existem muitas cores diferentes e parecidas, como quando há mais de 10 linhas. Por esses motivos, se houvesse a rotulação de paradas, seria um mapa eleito para utilização em situações reais, podendo ser fixado na parada selecionada.

4.2.1 Problemas graves levando a rejeição de mapas

Os dois principais problemas detectados para a rejeição de mapas são a presença de anomalias nos dados, e o excesso de linhas servindo uma única parada de ônibus. Tais problemas são detalhados a seguir.

Figura 4.4: Distribuição da quantidade de linhas por parada da cidade



Fonte: Autor

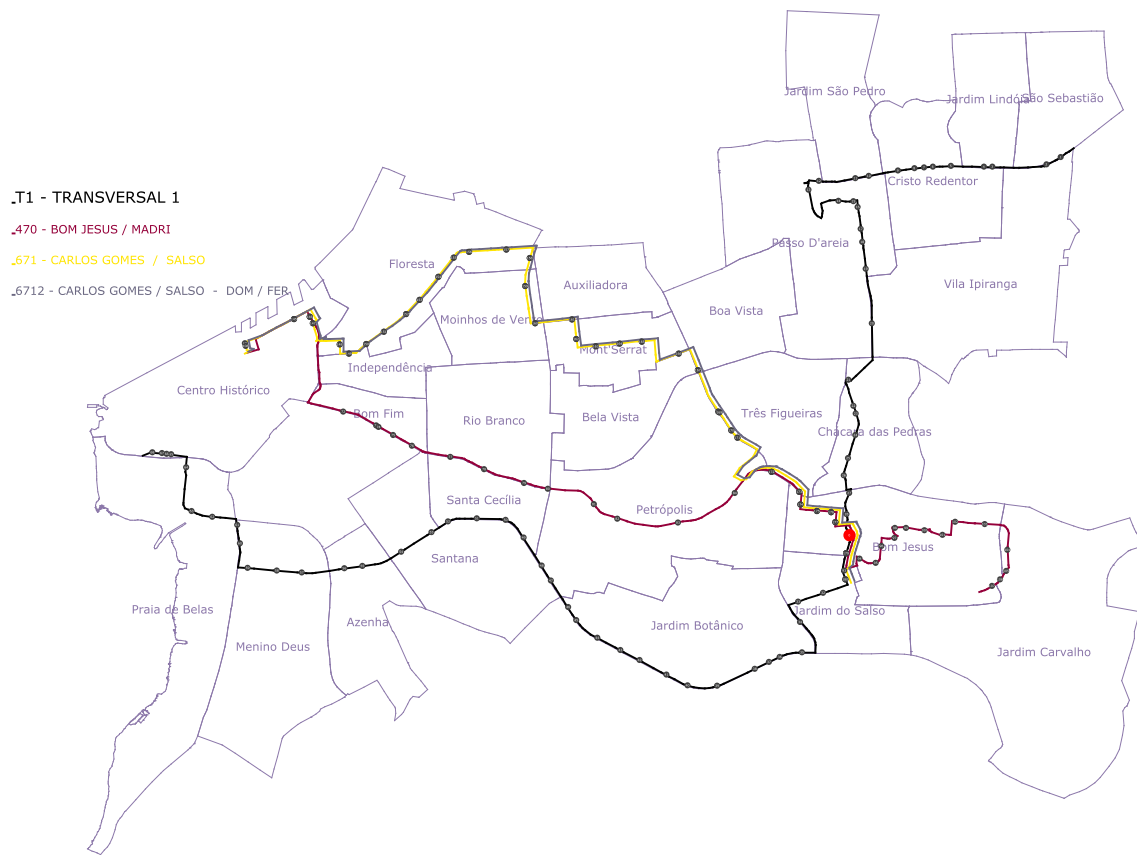
Anomalia no rastro GPS de algumas linhas

A existência de anomalias nos traços GPS, ocorridos sempre nas mesmas linhas, com riscos em linha reta por grandes trechos. A Figura 4.6 exibe um exemplo dessa anomalia em um dos mapas rejeitados. O problema ocorre porque existe, nas linhas em que o problema ocorre, um ponto na sequência do rastro GPS que está muito distante do ponto anterior (as vezes do outro lado da cidade), gerando um risco em linha reta em cima do mapa.

Excesso de linhas servindo uma parada

A existência de muitas linhas torna difícil a compreensão do mapa, não sendo possível visualizar claramente o trajeto de todas as linhas, principalmente em trechos onde todas as linhas passam juntas (como no caso da parada selecionada). A Figura 4.7 exibe um exemplo de mapa com muitas linhas que foi rejeitado por ter a legibilidade muito dificultada. A rejeição desse mapa ocorreu porque no trecho em que há muitas linhas atravessando um mesmo trajeto, é difícil enxergar claramente o trajeto de cada linha, pois algumas somem em meio

Figura 4.5: Exemplo de mapa bem sucedido com poucas linhas



Fonte: Autor

às outras. Também existe um trecho no mapa com muitas paradas próximas, dificultando a visualização de qual parada cada uma das linhas passa. Um terceiro problema observado também foi que, em decorrência do alto número de linhas, as legendas com o nome das linhas ocuparam um espaço vertical maior que o tamanho do desenho do mapa e por isso foram cortadas da imagem.

4.2.2 Problemas básicos

A análise dos mapas mostrou que existem bons resultados. Houve uma grande quantidade de mapas aceitos, devido aos parâmetros adotados. Porém, os mapas desenvolvidos ainda não poderiam ser utilizados em situações reais devido aos seguintes problemas básicos que são detalhados a seguir.

Figura 4.6: Mapa malsucedido: anomalia no desenho



Fonte: Autor

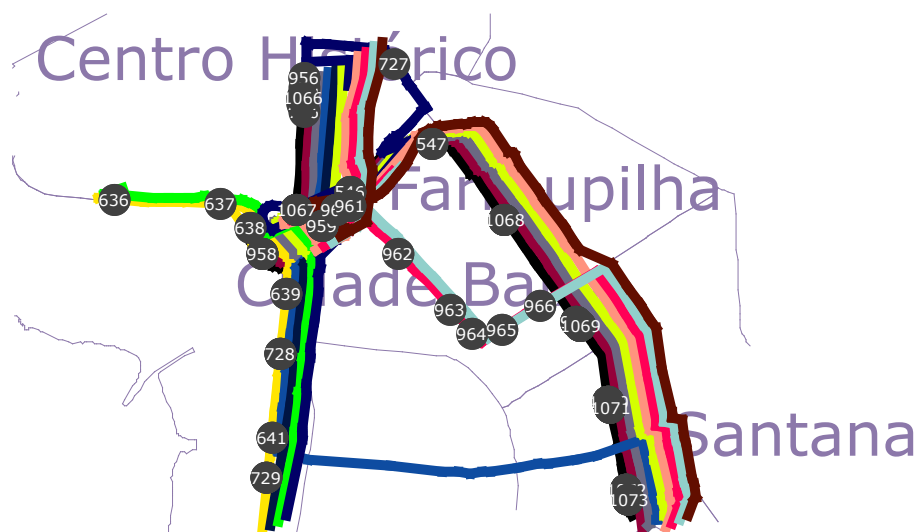
Disposição das paradas

É necessário ter uma melhor disposição esquemática das paradas, pois em alguns trechos existem várias paradas aglomeradas, sendo umas desenhadas por cima das outras. A Figura 4.8 ilustra um trecho de um mapa em que há essa aglomeração de paradas, dificultando a compreensão do mapa.

Rotulação de paradas

Nos dados GTFS de Porto Alegre as paradas não tem um nome único, mas sim apenas o nome da rua e, conseqüentemente, várias paradas com o mesmo

Figura 4.8: Trecho com muitas paradas aglomeradas dificultando a compreensão



Fonte: Autor

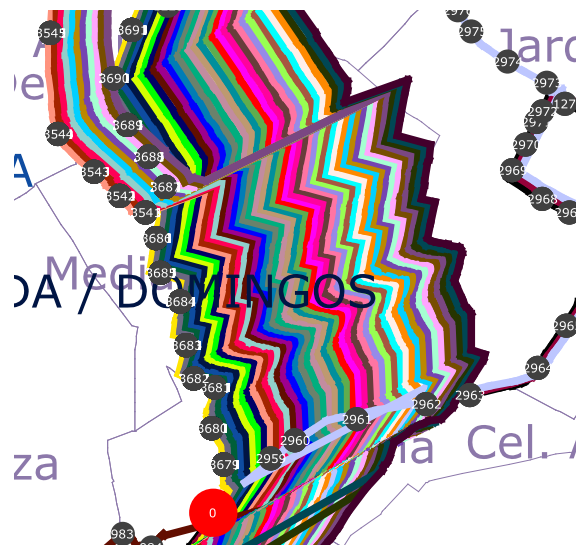
com mais precisão os locais da cidade em que consegue chegar a partir daquela parada. Na versão atual é possível apenas visualizar os bairros.

Deslocamento dos traços sobrepostos

O deslocamento dos traços das linhas nos trechos sobrepostos precisaria ser melhorado, pois em mapas com muitas paradas há trechos em que linhas se cruzam umas por cima das outras, há trechos em que elas se sobrepõem, e outros trechos em que o traço fica em formato de zigue-zague, devido a simplicidade do algoritmo de deslocamento das linhas que apenas desloca pontos dos rastros GPS que são próximos uns dos outros, sem uma avaliação mais global do desenho. A Figura 4.9 exibe um trecho em que há esses problemas. Uma possível melhoria do deslocamento das linhas seria avaliar o distanciamento entre os segmentos de reta dos traços GPS, e não apenas de cada ponto isoladamente. Para cada segmento de reta que liga dois pontos de traço GPS em cada linha de ônibus, deveria ser verificado se existem outros segmentos de reta que passam muito próximos (quase no mesmo trecho), e dessa forma estudar maneiras de deslocar os segmentos nessas situações. Essa solução exige melhor investigação, pois é preciso prever diversas situações possíveis e também aumentaria o tempo de processamento com os cálculos matemáticos e o número de iterações acrescentados.

A próxima seção apresenta os resultados do algoritmo sem a utilização de rastros GPS para construção do mapa.

Figura 4.9: Traços deslocados com padrão zigue-zague e cruzamentos



Fonte: Autor

4.3 Análise dos Mapas de Parada Sem Rastros GPS

A análise dos resultados obtidos com a segunda versão do Mapa da Parada foi feita de forma parecida com as versões anteriores, mas tendo uma mudança principal: desta vez não foram analisados todos os mapas, mas sim apenas um subconjunto do total. Essa mudança foi realizada porque após a análise padrão de 10% dos mapas foi verificado que ocorre um padrão no resultado em função da quantidade de linhas de cada mapa: se o mapa possui menos que 5 linhas, em geral é um bom mapa. Se possui entre 5 e 8 linhas, alguns mapas são bons, outros ruins, ou seja, requer uma análise mais aprofundada. Quando o mapa possui mais que 8 linhas, a regra geral é que todos são ruins, sendo que raramente aparece um bom. Os mapas bons, com mais de 8 linhas, e mapas ruins, com menos de 5 linhas, foram desprezados por serem considerados raras exceções. Foi feita então uma nova geração apenas de mapas entre 5 e 8 linhas, resultando em 920 mapas gerados para serem analisados. Dessa forma, mapas com menos de 5 linhas são considerados aceitos por definição, e mapas com mais de 8 linhas são considerados rejeitados. Por essa metodologia, foram considerados aceitos 4087 mapas. Logo, a taxa de sucesso foi de 75.88%.

4.3.1 Vantagens

As principais vantagens obtidas com o novo algoritmo foram a eliminação de problemas que envolviam os dados dos rastros GPS pelo fato de a nova versão não utilizar esses dados. A nova versão possui o traço do desenho entre cada par de paradas mais limpo, pois são desenhados segmentos de reta com origem em uma parada e destino em outra, não havendo ruído dos rastros GPS em que há um pequeno erro de precisão a cada obtenção da geolocalização e também há vários pontos entre duas paradas. A versão com rastros GPS também apresenta, em algumas situações, uma continuação do desenho do traço da linha de ônibus mesmo após a última parada do trajeto, o que não ocorre na versão atual. Um terceiro problema que também foi resolvido na nova versão foi a eliminação das anomalias com riscos por cima do desenho devido a existência de um ponto de rastro GPS em um local muito longe do restante do trajeto.

4.3.2 Desvantagens

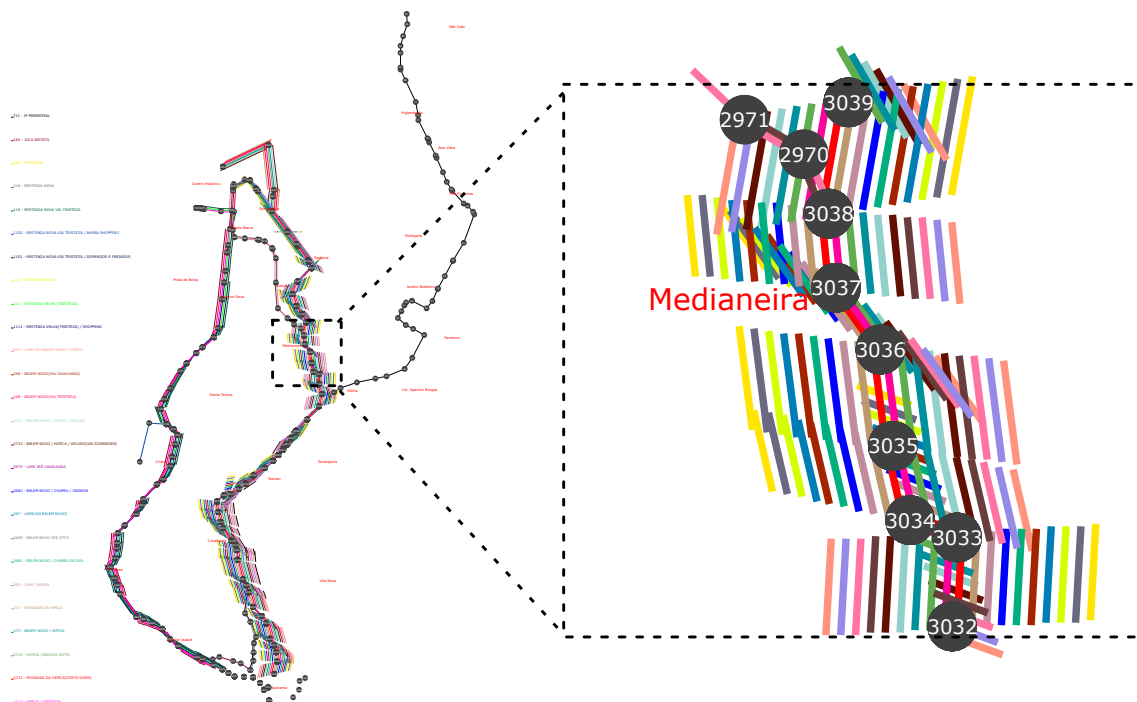
As principais desvantagens do novo método foram relacionadas a problemas no deslocamento das linhas sobrepostas. Quando há muitas linhas de ônibus, ocorrem desconexões dos traços das linhas em trechos que muitas linhas passam juntas, como na própria parada selecionada. A figura 4.10 exhibe um exemplo desse problema. As desconexões dos segmentos das linhas ocorrem porque o método que desloca as paradas através da rotação da matriz é muito simples e não resolve completamente o problema, tendo essa falha para todos os mapas com muitas linhas que andam juntas em alguma parte do trajeto. A solução desse problema exigiria melhor investigação de uma outra maneira de efetuar o deslocamento das linhas de forma mais eficaz.

4.3.3 Problemas não resolvidos

Três problemas não foram resolvidos com a nova versão: rótulos de paradas, aglomeração de paradas e inversão dos traços das linhas em curvas.

Em partes do mapa em que existem muitas paradas próximas, mas em

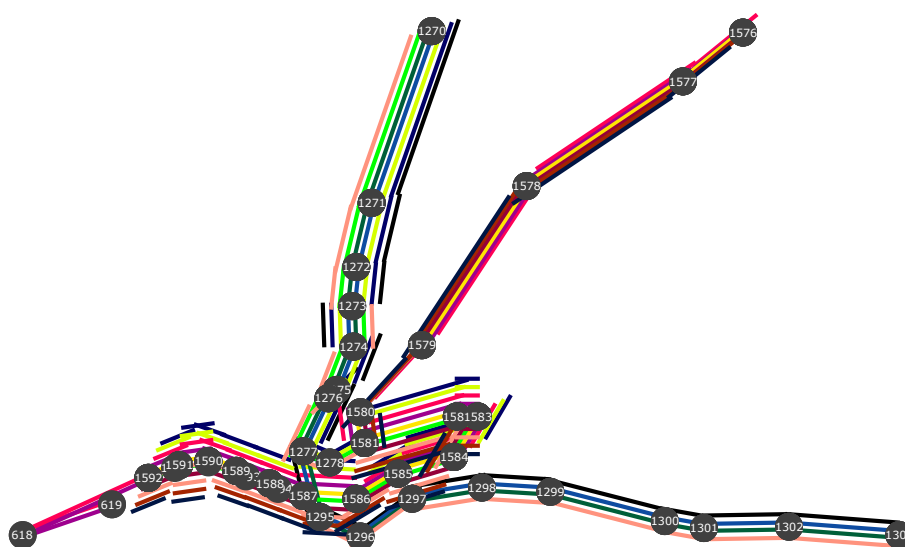
Figura 4.10: Traços desconectados quando há muitas linhas sobrepostas



Fonte: Autor

diferentes ruas, ou às vezes na mesma rua, acaba gerando muita dificuldade e tornando bastante difícil a compreensão da circulação das linhas naquele ponto. Isso pode ser visto na Figura 4.11. É possível que a falta do rastro GPS tenha tornado até um pouco pior essa situação, mas, de qualquer maneira, esse problema já existia na versão anterior.

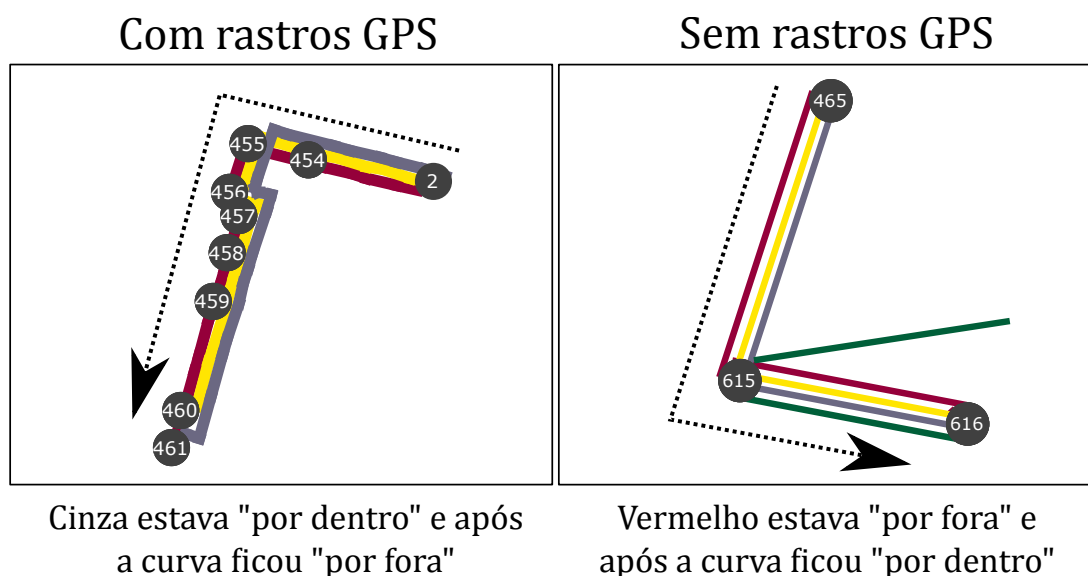
Figura 4.11: Muitas paradas aglomeradas dificultam compreensão



Fonte: Autor

Em alguns trechos são geradas inversões dos traços quando há uma curva no mapa, devido ao comportamento muito simples do algoritmo de deslocamento das linhas. Esse problema ocorria na versão anterior também, e está ilustrado na Figura 4.12, onde pode ser visto a ocorrência na versão antiga (à esquerda) e na nova versão (à direita).

Figura 4.12: Inversão dos traços no desenho



Fonte: Autor

Após a análise dos resultados do mapa sem rastros GPS foi considerado que este apresentou algumas vantagens, e outras desvantagens, equilibrando com as vantagens e desvantagens da versão anterior. A enorme quantidade de linhas e paradas da cidade é a raiz dos problemas das duas versões.

Ao analisar as soluções esquemáticas de outras grandes cidades no mundo, e com as dificuldades enfrentadas na construção dos algoritmos dos mapas gerados neste trabalho, conclui-se que uma possibilidade para alcançar melhores resultados seria utilizar uma abordagem menos geográfica e mais esquemática. Dessa forma, paradas muito próximas poderiam ser reposicionadas para não fiquem aglomeradas, e diversos outros ajustes de reposicionamento de rotas poderiam ser feitos para que o mapa gerado tivesse melhor qualidade. A próxima seção apresenta uma discussão sobre a possibilidade de desenvolvimento de um algoritmo para geração de mapas esquemáticos octalineaes, como o mapa de Londres visto na Figura 1.3.

4.4 Trabalhos Futuros

O formato de mapa do tipo octalinear exige que todas as rotas sejam dispostas em apenas 8 direções, tendo portanto os vetores das rotas em relação ao vetor do eixo x do plano XY -Cartesiano as rotações de 0° , 45° , 90° , 135° , 180° , 225° , 270° e 315° . Nesse formato esquemático, apesar das óbvias distorções em relação ao posicionamento geográfico dos objetos, a topologia da rede é mantida. O ideal, também, é que se mantenha o posicionamento dos locais no mapa o mais próximo possível do layout geográfico.

O desenho automático de mapas em formato octalinear já foi provado ser um problema NP-Completo (NÖLLENBURG, 2005). No artigo pesquisado é apresentada uma solução que utiliza programação inteira mista (*mixed integer programming*), que é uma técnica de otimização combinatória que combina restrições com variáveis inteiras e outras com variáveis reais. A solução de problemas de otimização combinatória com variáveis inteiras costuma ser de grande complexidade de implementação, exigindo a utilização de um programa conhecido como *solver* para encontrar o valor ótimo da função objetivo (respeitando as restrições). No artigo são propostas algumas regras para definir um bom *layout* octalinear. Posteriormente são definidas uma série de restrições do problema, divididas em dois grupos: restrições rígidas (*hard constraints*), que devem ser totalmente atendidas; e restrições fracas (*soft constraints*) que devem ser atendidas o máximo possível. As restrições rígidas incluem a garantia do mapa gerado ser octalinear, e de manter a topologia da rede. As restrições fracas tentam minimizar o máximo possível o número de curvas e manter os objetos o mais próximo possível do posicionamento original geográfico.

Em uma outra publicação mais recente é proposta uma versão atualizada desse mesmo método para a geração de mapas esquemáticos octalineaes de alta qualidade (NÖLLENBURG; WOLFF, 2011). A Figura 4.13 ilustra um dos resultados encontrados no artigo. Pela imagem em a é possível verificar o *layout* geográfico de entrada do algoritmo. As outras três imagens b, c e d exibem três saídas diferentes de acordo com diferentes parâmetros utilizados. Os três resultados mostram que o método utilizado consegue gerar bons mapas octalineaes. Porém, existe uma característica que chama bastante a atenção na imagem: a simplicidade do *layout* geográfico de entrada. Em todos os resultados exibidos nos

dois artigos, sempre são mostrados *layouts* simples de entrada, e já com certa organização na sequência das paradas e das linhas, o que difere bastante do *layout* real do sistema de transporte público de Porto Alegre com suas milhares de paradas e centenas de linhas.

Figura 4.13: Geração de mapas octalineaes com programação inteira mista

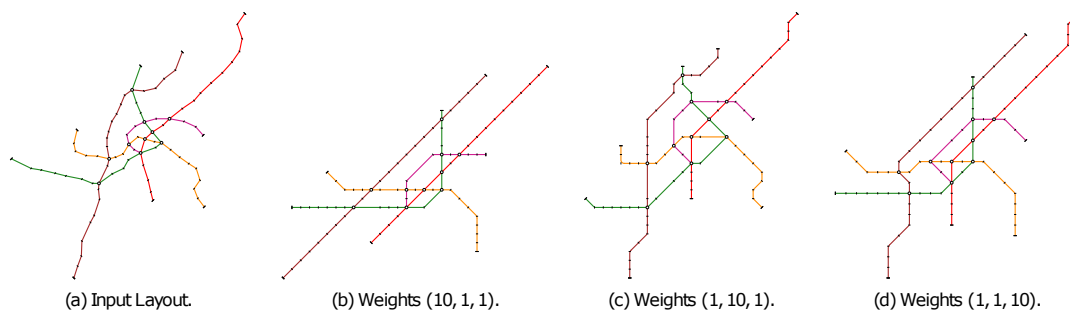


Fig. 6. Layouts of the metro network of Vienna with emphasis on bend minimization (b), preservation of relative positions (c), and length minimization (d) by assigning different weight vectors $(\lambda_{(S1)}, \lambda_{(S2)}, \lambda_{(S3)})$.

Fonte: (NÖLLENBURG; WOLFF, 2011)

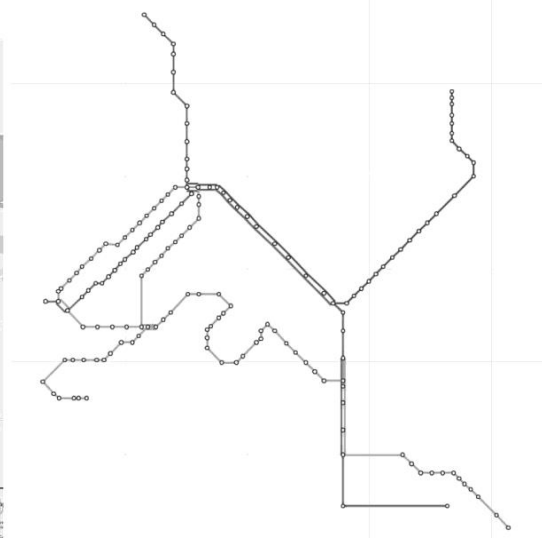
Uma outra possibilidade pensada após a realização dos testes neste trabalho foi a construção de um algoritmo genético para tentar aproximar um mapa octilinear através de regras que definam o quão próximo o mapa gerado está de um *layout* octilinear a cada iteração. Após uma pesquisa sobre esse assunto, descobrimos existir um estudo que apresenta exatamente essa proposta (GALVÃO; LAMAR; TACO, 2014). No artigo são definidas 8 regras para a criação de um bom mapa octilinear, e são efetuados testes na rede de transporte público de Brasília. A Figura ?? exibe um dos resultados do algoritmo desenvolvido no estudo. De acordo com as conclusões do artigo, o método se mostrou bastante promissor para redes com grande número de linhas de ônibus, por gerar bons resultados visíveis, e por ter a geração muito rápida.

Após a análise dos trabalhos encontrados na área de geração automática de mapas esquemáticos, é verificado que o método mais indicado para ser testado em um trabalho futuro na rede de transporte público de Porto Alegre é a construção de um algoritmo genético para a geração de mapas de parada octalineaes. Como nos estudos encontrados não foi visualizado um grande número de linhas e paradas de ônibus, conclui-se que ainda persiste o questionamento sobre a possibilidade dessa geração totalmente automática de bons mapas esquemáticos em cidades com alto número de linhas e paradas de ônibus, como Porto Alegre.

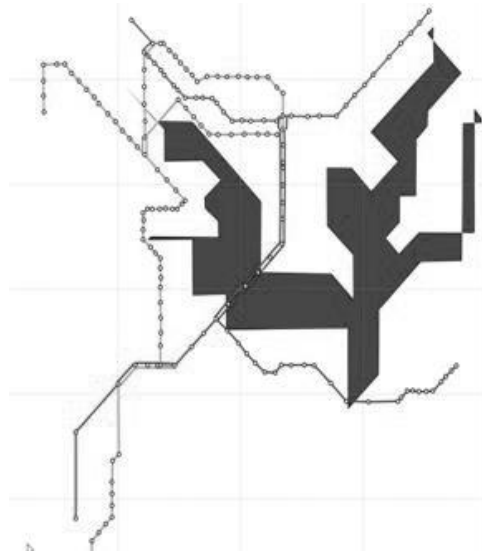
Figura 4.14: Geração de mapas octalineaes com algoritmo genético



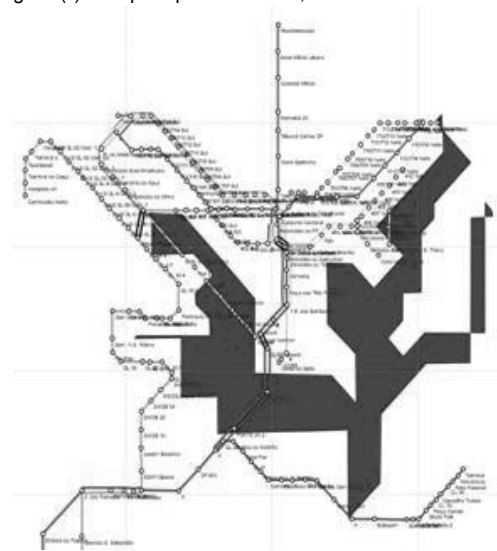
(a) Linhas de ônibus de Brasília. Mapa não esquemático



(b) Esquemática do conjunto de linhas em destaque (cor mais clara) da Figura 6(a) – Tempo de processamento 0,8 s



(c) Mesmo conjunto de linhas de 6(b), com diferente orientação – Tempo de processamento 0,8 s



(d) Todas as linhas esquematizadas, incluindo rótulos – Tempo de processamento 1,1 s

Fonte: (GALVÃO; LAMAR; TACO, 2014)

5 CONCLUSÃO

Este trabalho apresentou todas as etapas do desenvolvimento de um sistema gerador de mapas esquemáticos para o transporte público de Porto Alegre. O trabalho realizado teve como motivação a quase ausência desse tipo de mapa para a cidade de Porto Alegre, pois até hoje somente um mapa é conhecido, e consideramos que ele não é muito bom.

O sistema desenvolvido é capaz de gerar dois tipos de mapas: mapas de linha e mapas de parada, sendo o último tipo com duas versões implementadas. Após a análise humana de todos os mapas possíveis gerados pelo sistema foram identificados pontos fortes e pontos fracos em cada um deles.

O principal problema que precisaria ser melhorado no Mapa da Linha é a busca de uma melhor maneira de rotular as paradas, pois devido ao fato de os mapas possuírem muitas paradas, e do formato escolhido ser em linha reta, acabou sendo necessário escrever os nomes das paradas na orientação vertical. Dessa forma fica um pouco mais difícil para ler o nome de cada parada, mas ao mesmo tempo, é possível ver todas as paradas da linha, o que pode facilitar a visualização precisa dos locais da cidade por onde aquela linha passa. Uma outra possível melhoria também seria o acréscimo de pontos de interesse no mapa, obtidos através de shapefiles no portal do OpenStreetMap.

O problema principal que precisaria ser resolvido no mapa da parada, tanto na versão com rastros GPS como na sem rastros GPS é encontrar uma maneira de dispor mais esquematicamente todas as informações no mapa, pois utilizando a abordagem geográfica como feito nas duas versões, a construção de bons mapas fica comprometida, pois existem muitas linhas que passam em uma mesma parada em Porto Alegre. Também existem muitas paradas de diferentes linhas que são muito próximas umas das outras, o que causa dificuldade para visualização de qual linha passa por qual parada. A proximidade de diversas paradas (de acordo com os dados GTFS) também faz com que, em certas ocasiões, algumas paradas sejam desenhadas praticamente no mesmo lugar umas das outras no mapa.

Os mapas de linha precisariam de alguns ajustes, mas estão próximos de um formato que possa ser utilizado em situações reais, devido a simplicidade de apenas listar as paradas de uma linha de ônibus. Já as dificuldades encontradas

no desenvolvimento dos algoritmos para os mapas de parada e os problemas vistos na análise dos resultados até aqui levantam um questionamento se existe uma maneira automática para gerar bons mapas de trânsito em situações com tantas linhas e paradas de ônibus como a cidade de Porto Alegre.

Após a análise de todos os mapas foi efetuada uma pesquisa sobre a possibilidade de geração automática de mapas octalineaes, como o utilizado em Londres e diversas outras cidades. Foi encontrado um artigo onde é provado que o desenvolvimento desse tipo de mapa é um problema NP-Completo. Em um outro artigo, foi encontrada uma proposta de construção de um algoritmo genético para a aproximação de um mapa octalinear. Essa proposta foi testada na rede de Brasília, e apresentou bons resultados. A partir desses dois artigos, concluímos que o melhor método a ser testado em trabalhos futuros na rede de ônibus de Porto Alegre é a construção de um algoritmo genético, para geração de mapas de parada octalineaes.

REFERÊNCIAS

- ARNTSON, A. **Graphic design basics**. [S.l.]: Cengage Learning, 2011.
- BATIK, S. **Apache Batik SVG Toolkit**. 2011. Available from Internet: <<https://xmlgraphics.apache.org/batik/>>.
- BILLSON, C. **A History of the London Tube Maps**. [S.l.]: Reproduced with permission from the London Transport Museum. NTLworld, 2011.
- BUENO, M. S. Viaje trifácil: um sistema baseado em software livre para consulta de trajetos de ônibus aplicado a cidade de porto alegre. 2014.
- Carris. **Carris**. 2015. Available from Internet: <<http://www.carris.com.br/>>.
- COMMONS, W. **Scalable Vector Graphics**. 2016. Available from Internet: <https://en.wikipedia.org/wiki/Scalable_Vector_Graphics#/media/File:Bitmap_VS_SVG.svg>.
- Conorte. **Conorte**. 2015. Available from Internet: <<http://www.conorte.com.br/>>.
- EPTC. **EPTC**. 2015. Available from Internet: <<http://www2.portoalegre.rs.gov.br/eptc/>>.
- ESRI, E. Shapefile technical description. **An ESRI White Paper**, 1998.
- FERRAILOLO, J.; JUN, F.; JACKSON, D. Scalable vector graphics (svg) 1.1 specification. **World Wide Web Consortium (W3C)**. URL <http://www.w3.org/TR/SVG11>, 2003.
- GALVÃO, M. de L.; LAMAR, M. V.; TACO, P. W. G. Desenho automático de mapas octalineaes de rede de transporte público utilizando algoritmo genético. **TRANSPORTES**, v. 22, n. 1, p. 21–30, 2014.
- Google. **Referência da GTFS (Especificação Geral sobre Feeds de Transporte Público)**. 2012. Available from Internet: <<https://developers.google.com/transit/gtfs/reference?hl=pt-br>>.
- GOOGLE. **Google APIs**. 2016. Available from Internet: <<https://console.developers.google.com>>.
- HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. **Pervasive Computing, IEEE**, IEEE, v. 7, n. 4, p. 12–18, 2008.
- Inkscape. **Inkscape**. 2016. Available from Internet: <<https://inkscape.org>>.
- LONDON, T. F. **Tube Map**. 2016. Available from Internet: <<https://tfl.gov.uk/maps/track/tube>>.
- MCHUGH, B. Pioneering open data standards: The gtfs story. **Edited by Brett Goldstein with Lauren Dyson**, Citeseer, p. 125–135, 2013.

MOOVIT. **Moovit App**. 2016. Available from Internet: <<http://moovitapp.com/>>.

NÖLLENBURG, M. **Automated drawing of metro maps**. [S.l.]: Universität Karlsruhe, Fakultät für Informatik, 2005.

NÖLLENBURG, M.; WOLFF, A. Drawing and labeling high-quality metro maps by mixed-integer programming. **Visualization and Computer Graphics, IEEE Transactions on**, IEEE, v. 17, n. 5, p. 626–641, 2011.

PGADMIN. **PGAdmin**. 2016. Available from Internet: <<https://www.pgadmin.org>>.

Prefeitura de Porto Alegre. **Página do Turista**. 2015. Available from Internet: <http://www2.portoalegre.rs.gov.br/portal_pmpa_turista>.

Prefeitura de Porto Alegre. **ObservaPoa (Observatório da Cidade de Porto Alegre)**. 2016. Available from Internet: <<http://observapoa.com.br/>>.

Procempa. **DataPoa**. 2015. Available from Internet: <<http://www.datapoa.com.br>>.

Procempa. **Procempa**. 2015. Available from Internet: <<http://www.Procempa.com.br>>.

Procempa. **DataPoaAbout**. 2016. Available from Internet: <<http://datapoa.com.br/about>>.

SPTrans. **Mapa do Transporte Metropolitano**. 2011. Available from Internet: <<http://www.mobilize.org.br/mapas/5/mapa-da-rede-de-transporte-publico-de-sao-paulo.html>>.

STS. **STS**. 2015. Available from Internet: <<http://www.sts.com.br/>>.

Unibus. **Unibus**. 2015. Available from Internet: <<http://www.unibus.com.br/>>.

URBS. **Rede Integrada de Transporte Coletivo de Curitiba**. 2015. Available from Internet: <<http://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte>>.

W3C. **W3C - World Wide Consortiom**. 2015. Available from Internet: <<https://www.w3.org>>.

APÊNDICE A — SCRIPT DE IMPORTAÇÃO DOS ARQUIVOS GTFS

Este apêndice apresenta o *script* SQL utilizado para importar os arquivos GTFS para o banco de dados da aplicação.

Exemplo A.1: Obtenção do Bairro Pelas Coordenadas Geográficas no PostGIS

```

1  -- ***** STOPS *****
2  drop table if exists trips;
3  drop table if exists stops_routes;
4  drop table if exists stops;
5  CREATE TABLE stops
6  (
7      stop_id bigint,
8      stop_code varchar(20),
9      stop_name varchar(130),
10     stop_desc varchar(130),
11     stop_lat double precision,
12     stop_lon double precision,
13     CONSTRAINT stops_pkey PRIMARY KEY (stop_id)
14 )
15 WITH (
16     OIDS=FALSE
17 );
18 ALTER TABLE stops
19     OWNER TO postgres;
20
21 COPY stops FROM 'c:\Ricardo\Git\RicardoTCC\datapoa\GTFS\
    ↪ stops.txt' DELIMITER ',' CSV HEADER;
22
23 -- ***** ROUTES *****
24 drop table if exists routes;
25 CREATE TABLE routes
26 (
27     route_id varchar(8),
28     agency_id varchar(10),

```

```

29     route_short_name varchar(10),
30     route_long_name varchar(130),
31     route_desc varchar(130),
32     route_type integer,
33     route_url varchar(130),
34     route_color varchar(6),
35     route_text_color varchar(6),
36     CONSTRAINT routes_pkey PRIMARY KEY (route_id)
37
38 )
39 WITH (
40     OIDS=FALSE
41 );
42 ALTER TABLE routes
43     OWNER TO postgres;
44
45 COPY routes FROM 'c:\Ricardo\Git\RicardoTCC\datapoa\GTFS\
    ↪ routes.txt' DELIMITER ',' CSV HEADER;
46
47
48 -- ***** SHAPES *****
49 drop table if exists shapes;
50 CREATE TABLE shapes
51 (
52     shape_id varchar(20),
53     shape_pt_lat double precision,
54     shape_pt_lon double precision,
55     shape_pt_sequence integer
56 )
57 WITH (
58     OIDS=FALSE
59 );
60 ALTER TABLE shapes
61     OWNER TO postgres;

```



```

62 COPY shapes FROM 'c:\Ricardo\Git\RicardoTCC\datapoa\GTFS\
    ↪ shapes.txt' DELIMITER ',' CSV HEADER;
63
64
65 -- ***** TRIPS *****
66 drop table if exists trips;
67 CREATE TABLE trips
68 (
69     route_id varchar(20),
70     service_id varchar(20),
71     trip_id varchar(20),
72     trip_headsign varchar(20),
73     trip_short_name varchar(20),
74     direction_id integer,
75     block_id varchar(20),
76     shape_id varchar(20),
77     wheelchair_accessible integer,
78     trip_time integer,
79     CONSTRAINT trips_pkey PRIMARY KEY (trip_id),
80     CONSTRAINT trips_routes_fk foreign key (route_id)
81         REFERENCES routes (route_id) MATCH SIMPLE
82         ON UPDATE NO ACTION ON DELETE NO ACTION,
83     CONSTRAINT trips_calendar_fk foreign key (service_id)
84         REFERENCES calendar (service_id) MATCH SIMPLE
85         ON UPDATE NO ACTION ON DELETE NO ACTION
86 )
87 WITH (
88     OIDS=FALSE
89 );
90 ALTER TABLE trips
91     OWNER TO postgres;
92 COPY trips FROM 'c:\Ricardo\Git\RicardoTCC\datapoa\GTFS\
    ↪ trips.txt' DELIMITER ',' CSV HEADER;
93

```

```
94
95 drop table if exists stop_times;
96 CREATE TABLE stop_times
97 (
98     trip_id varchar(20),
99     arrival_time varchar(30),
100     departure_time varchar(30),
101     stop_id bigint,
102     stop_sequence integer
103 )
104 WITH (
105     OIDS=FALSE
106 );
107 ALTER TABLE stop_times
108     OWNER TO postgres;
109 COPY stop_times FROM 'c:\Ricardo\Git\RicardoTCC\datapoa\
    ↪ GTF5\stop_times.txt' DELIMITER ',' CSV HEADER;
110
111 delete from stop_times where stop_id in (
112     select stop_times.stop_id from stop_times
113     left join stops on stop_times.stop_id = stops.stop_id
114     where stops.stop_id is null
115 );
116
117 delete from stop_times where trip_id in (
118     select stop_times.trip_id from stop_times
119     left join trips on stop_times.trip_id = trips.trip_id
120     where trips.trip_id is null
121 );
122
123 -- adiciona uma PK sequencial para a tabela stop_times
124 alter table stop_times add column stop_time_id bigint;
125 drop sequence if exists stop_times_seq;
126 CREATE SEQUENCE stop_times_seq START 1;
```

```
127 update stop_times set stop_time_id = nextval('
    ↪ stop_times_seq');
128 alter table stop_times add constraint stop_times_pkey
    ↪ PRIMARY KEY (stop_time_id);
129
130 -- adiciona as FK's para stop_times:
131 alter table stop_times add constraint stop_times_stop_fk
    ↪ foreign key (stop_id)
132     references stops (stop_id)
133     on update no action;
134 alter table stop_times add constraint stop_times_trip_fk
    ↪ foreign key (trip_id)
135     references trips (trip_id)
136     on update no action;
137
138
139 -- adiciona uma PK sequencial para a tabela shapes
140 alter table shapes add column shape_id_pk bigint;
141 drop sequence if exists shapes_seq;
142 CREATE SEQUENCE shapes_seq START 1;
143 update shapes set shape_id_pk = nextval('shapes_seq');
144 alter table shapes add constraint shapes_pkey PRIMARY KEY (
    ↪ shape_id_pk);
```


APÊNDICE B — ESPECIFICAÇÃO DOS ARQUIVOS GTFS UTILIZADOS

Este apêndice apresenta a especificação dos arquivos GTFS utilizados no trabalho: rotas (`routes.txt`), paradas (`stops.txt`), viagens (`trips.txt`), horários de passagem (`stop_times.txt`), e formas (`shapes.txt`).

Arquivo `routes.txt`

A Tabela B.1 apresenta a especificação do arquivo `routes.txt`. São apresentados apenas os campos contidos no arquivo de Porto Alegre (pois o conjunto GTFS de Porto Alegre não contém todos os campos opcionais da especificação).

Tabela B.1: Especificação do Arquivo `routes.txt`

Campo	Obrig.	Detalhes
<code>route_id</code>	Sim	O campo <code>route_id</code> contém um ID que identifica um trajeto com exclusividade. O <code>route_id</code> é um conjunto de dados exclusivo.
<code>agency_id</code>	Não	O campo <code>agency_id</code> define uma agência para o trajeto especificado. Este valor é indicado no arquivo <code>agency.txt</code> . Use este campo quando estiver fornecendo dados para trajetos de mais de uma agência.
<code>route_short_name</code>	Sim	O <code>route_short_name</code> contém o nome abreviado de um trajeto. Geralmente, será um identificador pequeno e abstrato, como, por exemplo “32”, “100X” ou “Verde”, que os passageiros usam para identificar um trajeto, mas que não fornece nenhuma identificação de quais lugares são atendidos pelo trajeto. Se o trajeto não tem um nome abreviado, especifique um <code>route_long_name</code> e use uma sequência vazia como o valor deste campo.

Fonte: (Google, 2012)

Tabela B.1: Especificação do Arquivo routes.txt (Continuação)

Campo	Obrig.	Detalhes
route_long_name	Sim	O campo <code>route_long_name</code> contém o nome completo de um trajeto. Em geral, esse nome é mais descritivo que <code>route_short_name</code> e incluirá o destino ou a parada do trajeto. Se o trajeto não tem um nome completo, especifique um <code>route_short_name</code> e use uma sequência vazia como o valor deste campo.
route_desc	Não	O campo <code>route_desc</code> contém uma descrição de um trajeto. Forneça informações úteis e de qualidade. Não basta repetir o nome do trajeto. Por exemplo, "Um trem opera entre Inwood-207 St, Manhattan e Far Rockaway-Mott Avenue, Queens em todos os momentos. Além disso, das 6h até meia-noite, o trem adicional A opera entre Inwood-207 St e Lefferts Boulevard (normalmente, os trens se alternam entre Lefferts Blvd e Far Rockaway)."

Fonte: (Google, 2012)

Tabela B.1: Especificação do Arquivo routes.txt (Continuação)

Campo	Obrig.	Detalhes
route_type	Sim	<p>O campo <code>route_type</code> descreve o tipo de transporte usado em um trajeto. Os valores válidos deste campo são:</p> <p>0 - Bonde, ônibus elétrico, veículo leve sobre trilhos. Qualquer veículo leve sobre trilhos ou sistema de transporte de superfície em uma área metropolitana.</p> <p>1 - Metrô, trem subterrâneo. Qualquer sistema de trem subterrâneo em uma área metropolitana.</p> <p>2 - Via férrea. Usada para viagens interurbanas ou de longa distância.</p> <p>3 - Ônibus. Usado para trajetos curtos e de longa distância.</p> <p>4 - Balsa. Usado para serviço de barco de curta e longa distâncias.</p> <p>5 - Teleférico. Usado para teleféricos de nível de rua em que os fios passam debaixo dos carros.</p> <p>6 - Gondola, teleférico suspenso. Normalmente usado para teleféricos, onde o veículo é suspenso pelo cabo.</p> <p>7 - Funicular. Qualquer sistema de trilho projetado para subir ladeiras.</p>
route_url	Não	<p>O campo <code>route_url</code> contém o URL de uma página da sobre esse trajeto específico. Ele deve ser diferente de <code>agency_url</code>. O valor deve ser um URL completo que inclui <code>http://</code> o <code>https://</code>, e quaisquer caracteres especiais no URL devem ser ignorados corretamente. Consulte o <i>link</i>¹ para obter uma descrição de como criar valores de URLs completos.</p>

Fonte: (Google, 2012)

¹<http://www.w3.org/Addressing/URL/4_URI_Recommentations.html>

Tabela B.1: Especificação do Arquivo routes.txt (Continuação)

Campo	Obrig.	Detalhes
route_color	Não	Em sistemas em que cores são atribuídas a traje, o campo <code>route_color</code> define uma cor que corresponda ao trajeto. A cor deve ser informada como um número hexadecimal de seis caracteres, por exemplo, <code>00FFFF</code> . Se nenhuma cor é especificada, a cor padrão de trajetos é branca (<code>FFFFFF</code>). A diferença de cores entre <code>route_color</code> e <code>route_text_color</code> deve fornecer contraste suficiente quando visualizado em uma tela em preto e branco. O documento Técnicas W3C para avaliação de acessibilidade e ferramentas de reparo ² oferece um algoritmo útil para avaliar o contraste de cores. Há também ferramentas on-line úteis para escolher cores de contraste, incluindo o aplicativosnook.ca de verificação de contraste de cores ³ .
route_text_color	Não	O campo <code>route_text_color</code> pode ser usado para especificar uma cor legível para usar em desenho de texto contra um plano de fundo de <code>route_color</code> . A cor deve ser informada como um número hexadecimal de seis caracteres, por exemplo, <code>FFD700</code> . Se nenhuma cor é Se nenhuma cor for especificada, a cor padrão do texto será preto (<code>000000</code>). A diferença de cores entre <code>route_color</code> e <code>route_text_color</code> deve fornecer contraste suficiente quando visualizado em uma tela em preto e branco.

Fonte: (Google, 2012)

²<http://www.w3.org/TR/AERT#color-contrast>

³http://snook.ca/technical/colour_contrast/colour.html

Arquivo stops.txt

A Tabela B.2 apresenta a especificação do arquivo `stops.txt`. São apresentados apenas os campos contidos no arquivo de Porto Alegre (pois o conjunto GTFS de Porto Alegre não contém todos os campos opcionais da especificação).

Tabela B.2: Especificação do Arquivo stops.txt

Campo	Obrig.	Detalhes
<code>stop_id</code>	Sim	O campo <code>stop_id</code> contém um ID que identifica uma parada ou uma estação com exclusividade. Diversos trajetos podem usar a mesma parada. O <code>stop_id</code> é um conjunto de dados exclusivo.
<code>stop_code</code>	Não	O campo <code>stop_code</code> contém um pequeno texto ou um número que identifica a parada para os passageiros com exclusividade. Os códigos das paradas são usados muitas vezes em sistemas de informações sobre transporte público por telefone ou impressos em sinalizações nas paradas para que os passageiros possam obter informações sobre o horário das paradas com mais facilidade ou sobre chegadas de uma parada específica em tempo real. O campo <code>stop_code</code> só deve ser usado para códigos de parada exibidos aos passageiros. Para os códigos internos, use <code>stop_id</code> . Este campo deve ser deixado em branco para as paradas que não têm um código.
<code>stop_name</code>	Sim	O campo <code>stop_name</code> contém o nome de uma parada ou estação. Use um nome compreensível para as pessoas locais e linguagem turística.
<code>stop_desc</code>	Não	O campo <code>stop_desc</code> contém uma descrição de uma parada. Forneça informações úteis e de qualidade. Não basta repetir o nome da parada.

Fonte: (Google, 2012)

Tabela B.2: Especificação do Arquivo stops.txt (Continuação)

Campo	Obrig.	Detalhes
stop_lat	Sim	O campo <code>stop_lat</code> contém a latitude de uma parada ou estação. O valor do campo deve ser uma latitude WGS 84 válida.
stop_lon	Sim	O campo <code>stop_lon</code> contém a longitude de uma parada ou estação. O valor do campo deve ser uma latitude WGS 84 válida entre -180 e 180 .

Fonte: (Google, 2012)

Arquivo trips.txt

A Tabela B.3 apresenta a especificação do arquivo `trips.txt`. São apresentados apenas os campos contidos no arquivo de Porto Alegre (pois o conjunto GTFS de Porto Alegre não contém todos os campos opcionais da especificação).

Tabela B.3: Especificação do Arquivo trips.txt

Campo	Obrig.	Detalhes
route_id	Sim	O campo <code>route_id</code> contém um ID que identifica um trajeto com exclusividade. Este valor é indicado no arquivo <code>agency.txt</code> .
service_id	Sim	O campo <code>service_id</code> contém um ID que identifica um conjunto de datas em que o serviço está disponível para um ou mais trajetos. Este valor é indicado no arquivo <code>calendar.txt</code> ou <code>calendar_dates.txt</code> .
trip_id	Sim	O campo <code>trip_id</code> contém um ID que identifica uma viagem. O <code>trip_id</code> é um conjunto de dados exclusivo.

Fonte: (Google, 2012)

Tabela B.3: Especificação do Arquivo trips.txt (Continuação)

Campo	Obrig.	Detalhes
trip_headsign	Não	O campo <code>trip_headsign</code> contém o texto que aparece em uma sinalização que identifica o destino da viagem para os passageiros. Use este campo para distinguir diferentes padrões de serviço no mesmo trajeto. Se a placa muda durante uma viagem, você pode substituir o campo <code>trip_headsign</code> , especificando valores para o campo <code>stop_headsign</code> em <code>stop_times.txt</code> .
trip_short_name	Não	O campo <code>trip_short_name</code> contém o texto que aparece em programações e placas de sinalização para identificar a viagem para os passageiros, por exemplo, para identificar números de trens para viagens de trens suburbanos. Se os passageiros não recorrem normalmente aos nomes da viagem, deixe este campo em branco. Um valor de <code>trip_short_name</code> , se possível, deve identificar, com exclusividade, uma viagem em um dia de serviço; ele não deve ser usado para nomes de destino ou designações limitadas/expressas.

Fonte: (Google, 2012)

Tabela B.3: Especificação do Arquivo trips.txt (Continuação)

Campo	Obrig.	Detalhes
<code>direction_id</code>	Não	<p>O campo <code>direction_id</code> contém um valor binário que indica a direção de uma viagem. Use este campo para distinguir viagens bidirecionais com o mesmo <code>route_id</code>. Este campo não é usado na criação de trajetos; ele fornece uma maneira de separar viagens por direção durante a publicação de tabelas de horário. Você pode especificar nomes para cada direção com o campo <code>trip_headsign</code>.</p> <p>0 - viagem em uma única direção (por exemplo, só ida)</p> <p>1 - viagem na direção oposta (por exemplo, de volta)</p> <p>Por exemplo, você poderia usar os campos <code>trip_headsign</code> e <code>direction_id</code> juntos para atribuir um nome a uma viagem em cada direção "1234"; o arquivo <code>trips.txt</code> teria estas linhas para usar nas tabelas de horário:</p> <pre>trip_id, ... , trip_headsign, direction_id 1234, ... , to Airport, 0 1505, ... , to Downtown, 1</pre>
<code>block_id</code>	Não	<p>O campo <code>block_id</code> identifica o quadro a que a viagem pertence. Um bloco consiste em duas ou mais viagens sequenciais feitas usando o mesmo veículo, em que um passageiro pode passar de uma viagem para a próxima permanecendo no veículo. O campo <code>block_id</code> deve ser indicado por duas ou mais viagens no arquivo <code>trips.txt</code>.</p>

Fonte: (Google, 2012)

Tabela B.3: Especificação do Arquivo trips.txt (Continuação)

Campo	Obrig.	Detalhes
shape_id	Não	O campo shape_id contém um ID que define a forma da viagem. Este valor é indicado no arquivo shapes.txt. O arquivo shapes.txt permite definir como será traçada uma linha no mapa para representar uma viagem.
wheelchair_accessible	Não	0 (ou vazio) - indica que não há informações sobre acessibilidade para a viagem 1 - indica que o veículo que está sendo usado nesta viagem específica pode acomodar, pelo menos, um passageiro em cadeira de rodas 2 - indica que não é possível acomodar passageiros em cadeiras de rodas nesta viagem

Fonte: (Google, 2012)

No arquivo trips.txt do GTFS de Porto Alegre existe, também, o atributo trip_time, o qual não consta na especificação. Aparentemente, foi um campo acrescentado para conter a duração da viagem..

Arquivo stop_times.txt

A Tabela B.4 apresenta a especificação do arquivo stop_times.txt. São apresentados apenas os campos contidos no arquivo de Porto Alegre (pois o conjunto GTFS de Porto Alegre não contém todos os campos opcionais da especificação).

Tabela B.4: Especificação do Arquivo stop_times.txt

Campo	Obrig.	Detalhes
trip_id	Sim	O campo <code>trip_id</code> contém um ID que identifica uma viagem. Este valor é indicado no arquivo <code>trips.txt</code> .
arrival_time	Sim	O campo <code>arrival_time</code> especifica o horário de chegada em uma parada específica de uma viagem específica de um trajeto. O horário é medido de “meio-dia menos 12h” (efetivamente meia-noite, exceto para dias do horário de verão), no início da data do serviço. No caso de horários que ocorram após a meia-noite na data do serviço, digite o horário como um valor maior que 24:00:00 em horário local HH:MM:SS para o dia em que começa a programação da viagem. Se você não tem horários separados para chegada e partida em uma parada, insira o mesmo valor para <code>arrival_time</code> e <code>departure_time</code> .
departure_time	Sim	O campo <code>departure_time</code> especifica o horário de partida de uma parada específica para uma viagem específica de um trajeto. O horário é medido de “meio-dia menos 12h” (efetivamente meia-noite, exceto para dias do horário de verão), no início da data do serviço. No caso de horários que ocorram após a meia-noite na data do serviço, digite o horário como um valor maior que 24:00:00 em horário local HH:MM:SS para o dia em que começa a programação da viagem. Se você não têm horários diferentes para a chegada e a saída em uma parada, insira o mesmo valor para <code>arrival_time</code> e <code>departure_time</code> .

Fonte: (Google, 2012)

Tabela B.4: Especificação do Arquivo `stop_times.txt` (Continuação)

Campo	Obrig.	Detalhes
<code>stop_id</code>	Sim	<p>O campo <code>stop_id</code> contém um ID que identifica uma parada com exclusividade. Diversos trajetos podem usar a mesma parada. O campo <code>stop_id</code> é indicado no arquivo <code>stops.txt</code>. Se <code>location_type</code> é usado no arquivo <code>stops.txt</code>, todas as paradas indicadas em <code>stop_times.txt</code> deverão ter <code>location_type</code> igual a 0.</p> <p>Onde possível, os valores de <code>stop_id</code> devem permanecer consistentes entre as atualizações de <i>feed</i>. Em outras palavras, a parada A, com <code>stop_id</code> 1, deverá ter <code>stop_id</code> 1 em todas as paradas de dados subsequentes. Se uma parada não está programada, digite valores em branco para <code>arrival_time</code> e <code>departure_time</code>.</p>
<code>stop_sequence</code>	Sim	<p>O campo <code>stop_sequence</code> identifica a ordem das paradas de uma viagem específica. Os valores de <code>stop_sequence</code> devem ser números inteiros positivos e devem aumentar ao longo da viagem.</p> <p>Por exemplo, a primeira parada da viagem teria uma <code>stop_sequence</code> de 1, a segunda parada da viagem teria uma <code>stop_sequence</code> de 23, e a terceira teria uma <code>stop_sequence</code> de 40 e assim por diante.</p>

Fonte: (Google, 2012)

Arquivo `shapes.txt`

A Tabela B.5 apresenta a especificação do arquivo `shapes.txt`. São apresentados apenas os campos contidos no arquivo de Porto Alegre (pois o conjunto GTFS de Porto Alegre não contém todos os campos opcionais da especificação).

Tabela B.5: Especificação do Arquivo shapes.txt

Campo	Obrig.	Detalhes
shape_id	Sim	O campo <code>route_id</code> contém um ID que identifica uma forma com exclusividade.
shape_pt_lat	Sim	<p>O campo <code>shape_pt_lat</code> associa a latitude de um ponto de forma ao ID de uma forma. O valor do campo deve ser uma latitude WGS 84 válida. Cada linha do arquivo <code>shapes.txt</code> representa um ponto de forma em sua definição de formas;</p> <p>Por exemplo, se a forma “A_shp” possui três pontos em sua definição, o arquivo <code>shapes.txt</code> deve conter estas linhas para definir a forma:</p> <pre>A_shp, 37.61956, -122.48161, 0 A_shp, 37.64430, -122.41070, 6 A_shp, 37.65863, -122.30839, 11</pre>
shape_pt_lon	Sim	<p>O campo <code>shape_pt_lon</code> associa a longitude de um ponto de forma ao ID de uma forma. O valor do campo deve ser uma longitude WGS 84 de valor de -180 a 180. Cada linha do arquivo <code>shapes.txt</code> representa um ponto de forma em sua definição de formas.</p> <p>Por exemplo, se a forma “A_shp” possui três pontos em sua definição, o arquivo <code>shapes.txt</code> deve conter estas linhas para definir a forma:</p> <pre>A_shp, 37.61956, -122.48161, 0 A_shp, 37.64430, -122.41070, 6 A_shp, 37.65863, -122.30839, 11</pre>

Fonte: (Google, 2012)

Tabela B.5: Especificação do Arquivo shapes.txt (Continuação)

Campo	Obrig.	Detalhes
shape_pt_sequence	Sim	<p>O campo <code>shape_pt_sequence</code> associa a latitude e a longitude de uma forma de um ponto de formas com sua ordem sequencial juntamente com a forma. Os valores de <code>shape_pt_sequence</code> devem ser números inteiros positivos e devem aumentar com a viagem.</p> <p>Por exemplo, se a forma “A_shp” possui três pontos em sua definição, o arquivo <code>shapes.txt</code> deve conter estas linhas para definir a forma:</p> <pre data-bbox="647 875 1225 1028">A_shp, 37.61956, -122.48161, 0 A_shp, 37.64430, -122.41070, 6 A_shp, 37.65863, -122.30839, 11</pre>

Fonte: (Google, 2012)

APÊNDICE C — EXEMPLOS DE MAPAS ESQUEMÁTICOS

Figura C.1: Brooklyn Bus Map - Mapa esquemático das linhas de ônibus do Brooklyn



Figura C.3: Mapa do Transporte Metropolitano (São Paulo)

Mapa do Transporte Metropolitano

Metropolitan Transport Network



Legenda Legend

	1 Linha 1 - Azul Line 1 - Blue	METRÔ
	2 Linha 2 - Verde Line 2 - Green	METRÔ
	3 Linha 3 - Vermelha Line 3 - Red	METRÔ
	4 Linha 4 - Amarela Line 4 - Yellow	VIAQUATRO
	5 Linha 5 - Lilás Line 5 - Lilac	METRÔ
	7 Linha 7 - Rubi Line 7 - Ruby	CPTM
	8 Linha 8 - Diamante Line 8 - Diamond	CPTM
	9 Linha 9 - Esmeralda Line 9 - Emerald	CPTM
	10 Linha 10 - Turquesa Line 10 - Turquoise	CPTM
	11 Linha 11 - Coral Line 11 - Coral	CPTM
	11 Linha 11 - Coral - Expresso Leste Line 11 - Coral - East Express	CPTM
	12 Linha 12 - Safira Line 12 - Sapphire	CPTM
	15 Linha 15 - Prata Line 15 - Silver	METRÔ
	Expresso Turístico Touristic Express	CPTM
	Pontão ORCA - tarifada Orca Shuttle Service	EMTU
	Corredor Metropolitano de Ônibus ABD ABD Metropolitan Bus Corridor	EMTU
	Terminal Metropolitano de Ônibus Metropolitan Bus Terminal	

	Estação Station		Estações com elevador Station with elevator
	Integração - gratuita Integration - Free interchange		Acesso livre Free access
	Integração - tarifada Integration - Paid interchange		
	Integração - gratuita: Horário Especial (veja no site: METRÔ/CPTM) Integration - Free interchange during off-peak hours (see website: METRÔ/CPTM)		
	Terminal Rodoviário Road Terminal		
	Bicicletário Bike Parking Terminal		Paraciclos Bike Attaching Post
	Bicicletário com empréstimo de bicicleta Bike Parking & Rental Terminal		
	FÁCIL Estacionamento de Carro Integrado Integrated Car Parking		
	FÁCIL Estacionamento de Carro Integrado/ Paraciclos Integrated Car Parking/ Bike Attaching Post		

Informações úteis Useful Information

CPTM	www.cptm.sp.gov.br	0800 055 0121
EMTU	www.emtu.sp.gov.br	0800 724 0555
METRÔ	www.metro.sp.gov.br	0800 770 7722
VIAQUATRO	www.viaquatro.com.br	0800 770 7100

Consulte no site das empresas os horários de funcionamento das estações e transferências entre linhas.
Please address the websites of the metropolitan transport companies for stations service hours and line interchange information.



Secretaria dos Transportes Metropolitanos