UNIVERSIDADE FEDERAL FLUMINENSE

ARLINDO SARAIVA PEREIRA JUNIOR

GERAÇÃO DE ARQUIVO GTFS A PARTIR DE DADOS DO *OPENSTREETMAP*

**Niterói**

2019

ARLINDO SARAIVA PEREIRA JUNIOR

ESPECIFICANDO LINHAS DE TRANSPORTE PÚBLICO NO PADRÃO GTFS A PARTIR DE DADOS DO *OPENSTREETMAP*

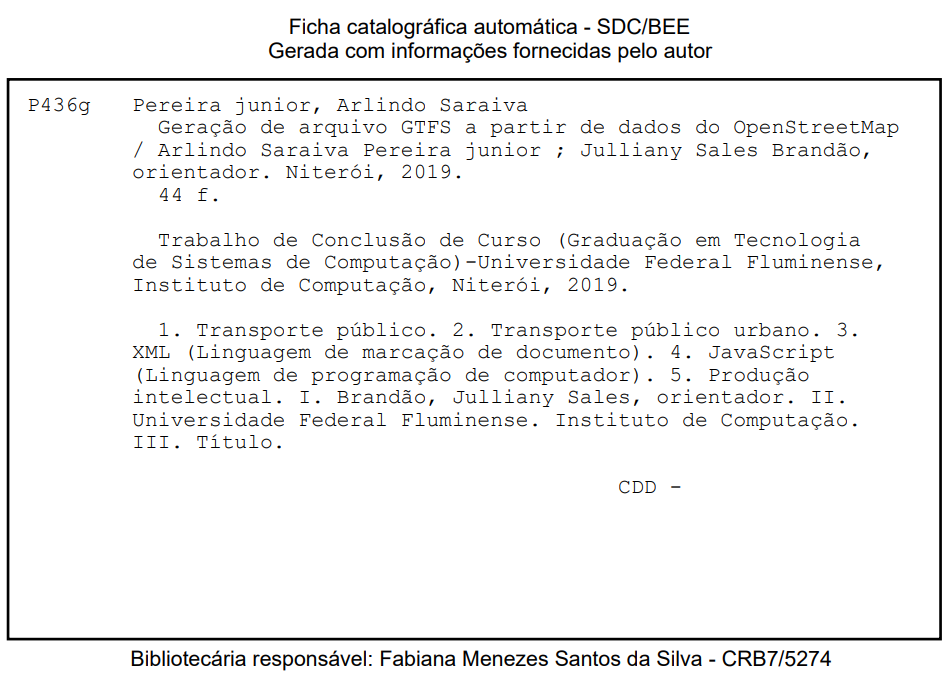
Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

**Orientadora:**

**Julliany Sales Brandão**

NITERÓI

2019



**ARLINDO SARAIVA PEREIRA JUNIOR**

ESPECIFICANDO LINHAS DE TRANSPORTE PÚBLICO NO PADRÃO GTFS A PARTIR DE DADOS DO *OPENSTREETMAP*

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, \_\_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ de 2019.

Banca examinadora:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profa. Julliany Sales Brandão, DSc. Orientadora

Cefet/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Felipe Pereira do Carmo, MSc. Avaliador

IFF - Instituto Federal Fluminense

AGRADECIMENTOS

Agradeço à minha esposa Dandarah e à minha filha Letícia pela perseverança com os desafios acadêmicos e compreensão com o frequente avançar das horas ao preparar o trabalho.

Aos meus pais, Maria e Arlindo, pela minha criação e sacrifícios para que desde tenra idade eu tivesse acesso a computadores.

À minha então pediatra, Lara, pelo primeiro computador doado.

Aos professores dos locais onde tive ensino de informática, no Colégio Pedro Segundo, do CEFET/RJ, da UNIRIO e da UFF.

# 

RESUMO

O trabalho apresentou o padrão de informações sobre transporte público GTFS - *General Transport Feed Specification* -, detalhando os arquivos e campos que o compõe; apresentou o *OpenStreetMap*, plataforma de dados cartográficos colaborativa e em licença livre, detalhando seu modelo de dados; traçou uma relação entre as duas plataformas e propôs um algoritmo para geração de um *feed* GTFS a partir de dados do *OpenStreetMap* e entradas do usuário. Dessa maneira, visa contribuir para a melhoria da publicização de dados de transporte público no país em um formato aberto e legível por máquinas.

**Palavras-chave: *gtfs, openstreetmap*, transporte público.**

# 

ABSTRACT

This work presented the public transport standard GTFS - General Transport Feed Specification -, detailing the files and fields that compose it; presented OpenStreetMap, a collaborative project with cartographic data under an open license; showed the relation between the two platforms and proposed an algorithm to generate a GTFS feed based on OpenStreetMap data and user input. By doing so, it helps to improve public transportation data available in an open format and machine readable.

**Keywords:gtfs, openstreetmap, public transport.**

# 

LISTA DE TABELAS

|  |  |
| --- | --- |
| [Tabela 1: Relação de arquivos obrigatórios e opcionais de um arquivo GTFS](#hj6ydmoroi2) | 12 |
| [Tabela 2: Campos obrigatórios do arquivo agency.txt](#qty1ap3yi3uk) | 14 |
| [Tabela 3: Campos obrigatórios do arquivo stops.txt](#jvmzaxcekifk) | 14 |
| [Tabela 4: Campos do arquivo routes.txt](#r67eby4prx1o) | 15 |
| [Tabela 5: Campos do arquivo trips.txt](#7i4j9z17pqiq) | 16 |
| [Tabela 6: Campos obrigatórios do arquivo stop\_times.txt](#ummqfqwe0xwg) | 17 |
| [Tabela 7: Campos obrigatórios do arquivo calendar.txt](#1yyfsx4w96jb) | 18 |
| Tabela 8: Campos obrigatórios do arquivo opcional shapes.txt | 19 |
| [Tabela 9: Etiquetas do nó que define a estação de trem Recife no *OpenStreetMap*](#9p2fvk46ci0l) | 21 |
| [Tabela 10: Propriedades do elemento *node* no OSM XML](#6unchnoms7bk) | 22 |
| [Tabela 11: Etiquetas da via que define parte do trecho do metrô de Recife no *OpenStreetMap*](#wbutbzhifrqj) | 24 |
| [Tabela 12: Propriedades do elemento *way* no OSM XML](#v8akco45t9yv) | 25 |
| [Tabela 13: Etiquetas da relação que define a rota Linha Sul do metrô de Recife](#8kwq1zvsh10t), sentido Cajueiro Seco → Recife | 27 |
| [Tabela 14: Etiquetas da relação que define a rota mestre Linha Sul do metrô de Recife](#ec6o8vs5lawe) | 29 |
| [Tabela 15: Comparação de dados entre o GTFS e o *OpenStreetMap*](#glusl4i5vryf) | 31 |

# 

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*, Interface de Programação de Aplicações

CBTU – Companhia Brasileira de Trens Urbanos

CSV – *Comma-separated Values*, Valores Separados por Vírgula

GTFS – *General Transit Feed Specification*, Especificação Geral de *Feed* de Transporte Público

OSM – *OpenStreetMap*

VLT – Veículo Leve sobre Trilhos

WGS – *World Geodetic System*

XML – *eXtensible Markup Language*, Linguagem Extensível de Marcação

SUMÁRIO

RESUMO.....................................................................................................................6.  
ABSTRACT..................................................................................................................7  
LISTA DE TABELAS....................................................................................................8  
LISTA DE ABREVIATURAS E SIGLAS.......................................................................9  
[1.INTRODUÇÃO 11](#_Toc12461041)

[2.PANORAMA SOBRE O FORMATO GTFS 12](#_Toc12461042)

[2.1.ESPECIFICANDO AGÊNCIAS DE TRANSPORTE PÚBLICO 13](#_Toc12461043)

[2.2.ESPECIFICANDO PARADAS DE TRANSPORTE PÚBLICO 14](#_Toc12461044)

[2.3.ESPECIFICANDO ROTAS DE TRANSPORTE PÚBLICO 15](#_Toc12461045)

[2.4.ESPECIFICANDO VIAGENS DE TRANSPORTE PÚBLICO 16](#_Toc12461046)

[2.5.ESPECIFICANDO HORÁRIOS DE PARTIDA E CHEGADA 17](#_Toc12461047)

[2.6.ESPECIFICANDO CALENDÁRIOS DE OPERAÇÃO 18](#_Toc12461048)

[2.7.ESPECIFICANDO TRAJETOS DE TRANSPORTE PÚBLICO 19](#_Toc12461049)

[3.*OPENSTREETMAP* E SEU MODELO DE DADOS 20](#_Toc12461050)

[3.1.NÓ – CONCEITO E REPRESENTAÇÃO EM OSM XML 20](#_Toc12461051)

[3.2.VIA – CONCEITO E REPRESENTAÇÃO EM OSM XML 23](#_Toc12461052)

[3.3.RELAÇÃO – CONCEITO E REPRESENTAÇÃO EM OSM XML 26](#_Toc12461053)

[4. GERANDO UM FEED GTFS A PARTIR DE DADOS DO OPENSTREETMAP 31](#_Toc12461054)

[4.1 COMPARANDO DADOS DO GTFS E DO *OPENSTREETMAP* 31](#_Toc12461055)

[4.2.LENDO OS DADOS DA API DO *OPENSTREETMAP* 32](#_Toc12461056)

[4.3.ESCREVENDO O ARQUIVO AGENCY.TXT 34](#_Toc12461057)

[4.4.ESCREVENDO O ARQUIVO ROUTES.TXT 34](#_Toc12461058)

[4.5.ESCREVENDO O ARQUIVO STOPS.TXT 35](#_Toc12461059)

[4.6.ESCREVENDO O ARQUIVO SHAPES.TXT 35](#_Toc12461060)

[4.7.CRIANDO FORMULÁRIOS DE ENTRADA PARA PREENCHIMENTO DAS INFORMAÇÕES DE PARTIDAS 36](#_Toc12461061)

[4.8.ESCREVENDO O ARQUIVO CALENDARS.TXT 37](#_Toc12461062)

[4.9.ESCREVENDO O ARQUIVO TRIPS.TXT 38](#_Toc12461063)

[4.10.ESCREVENDO O ARQUIVO STOP\_TIMES.TXT 38](#_Toc12461064)

[CONCLUSÕES E TRABALHOS FUTUROS 40](#_Toc12461065)

[REFERÊNCIAS BIBLIOGRÁFICAS 41](#_Toc12461066)

## INTRODUÇÃO

Este trabalho se propõe a dar um rápido panorama geral sobre o modelo de dados GTFS (*General Transit Feed Specification*), um padrão de comunicação de dados sobre a operação de serviços de transporte público [1], do *OpenStreetMap* (OSM), uma plataforma colaborativa de dados geográficos abertos [2] e, posteriormente, apresentar uma implementação de geração de um arquivo GTFS baseado nos dados do *OpenStreetMap*.

A motivação para desenvolver uma solução neste sentido, além do desenvolvimento acadêmico, se dá por querer contribuir com a divulgação dos serviços de transporte ferroviário de passageiros da estatal Companhia Brasileira de Trens Urbanos (CBTU) [3] no formato GTFS, o que permitiria incluí-los em plataformas como o *Google Maps*. Desta forma, o presente trabalho pretende contribuir uma pequena melhoria da divulgação do transporte público nas seis regiões metropolitanas de nosso país que a CBTU opera.

O Capítulo 2 traz um panorama geral sobre o formato GTFS, qual a sua utilidade e quais são e as informações nele contidas, mencionando atributos obrigatórios e opcionais. A principal referência é a página de documentação no *Google Transit*, uma vez que o formato é criação da empresa Google [4].

O terceiro capítulo é dedicado a uma explanação sobre o que é o *OpenStreetMap* e como é seu modelo de dados, descrevendo as feições que compõem o mapa, bem como a sua representação em *eXtended Markup Language* (XML) [5]. As referências são a documentação do projeto, o *OpenStreetMap Wiki* [6].

Já o quarto capítulo faz a ligação entre os dois assuntos, descrevendo um programa em JavaScript que captura esses dados geográficos do *OpenStreetMap* e gera um arquivo GTFS contendo as informações sobre a operação do transporte público sobre trilhos numa dada região.

Finalmente, o Capítulo 5 traz as conclusões e indicações para futuros trabalhos.

## PANORAMA SOBRE O FORMATO GTFS

A Especificação Geral de *Feeds* de Transporte Público (*General Transport Feeds Specification* – GTFS) "define um formato comum para horários de transportes públicos e informações geográficas associadas" [1]. Tem como objetivo ser um formato padrão de comunicação de informações de transporte público, incluindo linhas, serviços e horários de partida.

Tecnicamente, um arquivo GTFS é um conjunto de arquivos de texto reunidos em um pacote compactado no formato zip. Cada um destes arquivos de texto é uma tabela seguindo o padrão *Comma-Separated Values* (CSV), tendo a primeira linha como cabeçalho descritivo dos campos e possuindo um nome de arquivo específico. Alguns destes arquivos são obrigatórios, outros são opcionais. De forma semelhante, cada um deles possui atributos obrigatórios e opcionais.

A Tabela a seguir mostra a relação de arquivos obrigatórios e opcionais:

Tabela 1: Relação de arquivos obrigatórios e opcionais de um arquivo GTFS [7]

|  |  |  |
| --- | --- | --- |
| **Nome do arquivo** | **Obrigatório** | **Definição** |
| agency.txt | sim | Uma ou mais agências de transporte público que fornecem os dados desse *feed*. |
| stops.txt | sim | Locais individuais em que os veículos pegam e 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 *feed*, incluindo editor, versão e informações sobre validade. |

As páginas a seguir tratam dos arquivos obrigatórios do padrão GTFS, bem como arquivos opcionais que podem ser gerados a partir dos dados do *OpenStreetMap*.

### ESPECIFICANDO AGÊNCIAS DE TRANSPORTE PÚBLICO

As agências de transporte público – neste contexto, as empresas que operam o transporte público, ou as secretarias ou autarquias responsáveis por este – são representadas no GTFS no arquivo agency.txt. Este arquivo é uma tabela no formato CSV com campos obrigatórios e opcionais. Os campos obrigatórios estão descritos na Tabela 2.

Tabela 2: Campos obrigatórios do arquivo agency.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| agency\_name | Nome completo da agência de transporte público. |
| agency\_url | Endereço web da agência de transporte público. |
| agency\_timezone | Fuso horário de onde a agência de transporte público está localizada. |

Um exemplo de arquivo agency.txt pode ser visto a seguir:

agency\_name,agency\_url,agency\_timezone

CBTU,https://www.cbtu.gov.br,America/Sao\_Paulo

### ESPECIFICANDO PARADAS DE TRANSPORTE PÚBLICO

As paradas de transporte público – estações, pontos de ônibus etc. são representadas no GTFS no arquivo stops.txt, que como os demais é uma tabela no formato CSV com campos obrigatórios e opcionais. Os campos obrigatórios são:

Tabela 3: Campos obrigatórios do arquivo stops.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| stop\_id | Identificador alfanumérico único da parada ou estação – **chave primária**. |
| stop\_name | Nome da parada. |
| stop\_lat | Latitude da parada no formato de coordenada *World Geodetic System* (WGS) 84 [8]. |
| stop\_lon | Longitude da parada no formato de coordenada WGS 84. |

Um exemplo de arquivo stops.txt pode ser visto a seguir:

stop\_id,stop\_name,stop\_lat,stop\_lon

stop1,Recife,-8.06917,-34.88537

### ESPECIFICANDO ROTAS DE TRANSPORTE PÚBLICO

As rotas de transporte público, linhas de ônibus, metrô etc., são especificadas no arquivo routes.txt. Serviços distintos de uma mesma rota (por exemplo, Linha 1 do metrô sentido centro ou sentido subúrbio) são especificadas como uma única entrada. A Tabela 4 traz alguns campos do arquivo routes.txt que podem ser especificados a partir de dados do *OpenStreetMap* (nomes de campo marcados com asterisco são opcionais):

Tabela 4: Campos do arquivo routes.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| route\_id | Identificador alfanumérico único da rota – **chave primária**. |
| route\_short\_name | Nome abreviado de uma rota, por exemplo "2" ou "Verde", ou vazio caso não haja um nome abreviado. |
| route\_long\_name | Nome completo de uma rota, por exemplo "Linha 2", ou vazio caso não haja um nome completo. |
| route\_type | Descreve o tipo de transporte usado em uma rota. Os valores válidos são:   * 0 - Bonde, ônibus elétrico, veículo leve sobre trilhos; * 1 - Metrô ou trem em área metropolitana; * 2 - Via férrea, usado para viagens interurbanas ou de longa distância; * 3 - Ônibus * 4 - Balsa * 5 - Teleférico * 6 - Gôndola, teleférico suspenso * 7 - Funicular, sistema de trilho projetado para subir ladeiras |
| route\_color\* | Cor da linha de transporte público, em código hexadecimal de seis dígitos – por exemplo, 00FFFF. |

Um exemplo de arquivo routes.txt pode ser visto a seguir:

route\_id,route\_short\_name,route\_long\_name,route\_type,route\_color

route1,Centro 1,Recife - Camaragibe,1,00FFFF

### ESPECIFICANDO VIAGENS DE TRANSPORTE PÚBLICO

As viagens de transporte público são listadas no arquivo trips.txt. Este arquivo traz uma linha para cada partida de cada linha no sistema de transporte público. Por exemplo, se há dez partidas diárias de um trem em um sentido e dez no sentido oposto, este arquivo terá 20 linhas (além da linha do cabeçalho).

Neste arquivo há a primeira referência a outras tabelas (chaves estrangeiras). Alguns dos campos estão na tabela a seguir, sendo opcionais os nomes marcados com asterisco.

Tabela 5: Campos do arquivo trips.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| route\_id | Identificador alfanumérico da linha, especificado no arquivo routes.txt – **chave estrangeira**. |
| service\_id | Identificador alfanumérico do conjunto de datas em que o serviço está disponível, especificado no arquivo calendar.txt – **chave estrangeira**. |
| trip\_id | Identificador alfanumérico único de uma viagem – **chave primária**. |
| shape\_id\* | Identificador alfanumérico do trajeto feito pela linha, especificado no arquivo shapes.txt – **chave estrangeira**. Opcional. |

Segue exemplo de arquivo trips.txt:

route\_id,service\_id,trip\_id,shape\_id

route1,service1,trip1,shape1

route1,service1,trip2,shape1

route1,service2,trip3,shape1

route1,service2,trip4,shape1

### ESPECIFICANDO HORÁRIOS DE PARTIDA E CHEGADA

No arquivo stop\_times.txt listamos os horários de partida e chegada em cada estação do trajeto, para cada viagem realizada. Por exemplo, para uma linha de metrô com 12 paradas, há 24 linhas no arquivo considerando os dois sentidos para cada viagem realizada, desconsiderando o cabeçalho.

Neste arquivo também há chaves estrangeiras fazendo referência a outras tabelas. Os campos obrigatórios são:

Tabela 6: Campos obrigatórios do arquivo stop\_times.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| trip\_id | Identificador alfanumérico da viagem no arquivo trips.txt – **chave estrangeira**. |
| arrival\_time | Horário de chegada na parada, no formato hh:mm:ss.  Caso seja após meia noite, somar 24 na hora (por exemplo, 1:23 da madrugada = 25:23:00). |
| departure\_time | Horário de partida na parada, no formato hh:mm:ss. Caso seja após meia noite, somar 24 na hora (por exemplo, 1:23 da madrugada = 25:23:00). |
| stop\_id | Identificador alfanumérico da parada no arquivo stops.txt – **chave estrangeira**. |
| stop\_sequence | Ordem das paradas em uma viagem específica, com valores numéricos não necessariamente sequenciais que aumentam ao longo da viagem. |

Arquivo stop\_times.txt de exemplo:

trip\_id,arrival\_time,departure\_time,stop\_id,stop\_sequence

trip1,5:59:00,6:00:00,stop1,1

trip1,6:03:00,6:03:30,stop2,23

### ESPECIFICANDO CALENDÁRIOS DE OPERAÇÃO

Os dias da semana e horários de operação de uma dada linha são informados no arquivo calendar.txt. Exceções (feriados, por exemplo) podem ser informados no arquivo opcional calendar\_dates.txt. Todos os campos são obrigatórios e estão listados a seguir:

Tabela 7: Campos obrigatórios do arquivo calendar.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| service\_id | Identificador alfanumérico único de uma entrada de calendário – **chave primária**. Caso haja operações distintas para dias úteis e finais de semana, haverá duas entradas na tabela. |
| monday | Indica com 0 ou 1 se o serviço opera normalmente às segundas-feiras. |
| tuesday | Indica com 0 ou 1 se o serviço opera normalmente às terças-feiras. |
| wednesday | Indica com 0 ou 1 se o serviço opera normalmente às quartas-feiras. |
| thursday | Indica com 0 ou 1 se o serviço opera normalmente às quintas-feiras. |
| friday | Indica com 0 ou 1 se o serviço opera normalmente às sextas-feiras. |
| saturday | Indica com 0 ou 1 se o serviço opera normalmente aos sábados. |
| sunday | Indica com 0 ou 1 se o serviço opera normalmente aos domingos. |
| start\_date | Indica uma data de início da operação, no formato YYYYMMDD (ano com quatro dígitos, mês com dois dígitos, dia com dois dígitos). Deve ser especificada mesmo se for desconhecida. |
| end\_date | Indica uma data de fim da operação, no formato YYYYMMDD (ano com quatro dígitos, mês com dois dígitos, dia com dois dígitos). Deve ser especificada mesmo se for indeterminada. |

Arquivo calendar.txt de exemplo:

service\_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start\_date,end\_date

service1,1,1,1,1,1,1,1,20000101,20991231

### ESPECIFICANDO TRAJETOS DE TRANSPORTE PÚBLICO

Os trajetos percorridos pelos veículos podem ser especificados no arquivo opcional shapes.txt. Cada linha neste arquivo, após o cabeçalho, representa uma coordenada de cada trajeto. Uma vez incluído, este arquivo traz os campos obrigatórios a seguir:

Tabela 8: Campos obrigatórios do arquivo shapes.txt

|  |  |
| --- | --- |
| **Nome do campo** | **Descrição** |
| shape\_id | Identificador alfanumérico único de um trajeto – **chave primária**. |
| shape\_pt\_lat | Latitude de um ponto do trajeto no padrão WGS 84 [8]. |
| shape\_pt\_lon | Longitude de um ponto do trajeto no padrão WGS 84. |
| shape\_pt\_sequence | Ordem dos pontos em um trajeto, com valores numéricos não necessariamente sequenciais que aumentam ao longo do mesmo. |

Arquivo shapes.txt de exemplo:

shape\_id,shape\_pt\_lat,shape\_pt\_lon,shape\_pt\_sequence

shape1,-8.0686599,-34.8848571,1

shape1,-8.0688418,-34.8849784,2

shape1,-8.0692442,-34.8852618,3

shape1,-8.0697629,-34.8856296,5

## *OPENSTREETMAP* E SEU MODELO DE DADOS

O *OpenStreetMap* é uma plataforma colaborativa de dados geográficos em licença aberta. Como definido em sua página *Sobre*, "é desenvolvido por uma comunidade voluntária de mapeadores que contribuem e mantêm atualizados os dados sobre estradas, trilhos, cafés, estações ferroviárias e muito mais por todo o mundo" [9]. De forma resumida, pode ser definida como "A Wikipédia dos Mapas" [10].

Os componentes que compõem o modelo de dados conceitual do *OpenStreetMap* são três: nós (do inglês *nodes*), vias (*ways*) e relações (*relations*) [11]. Cada uma delas pode ter uma ou mais etiquetas (*tags*) que descrevem aquele elemento.

De uma forma geral, nós definem pontos no espaço; vias definem linhas ou áreas; e relações são utilizadas para explicar como outros elementos interagem entre si. Nos subcapítulos a seguir será possível compreender um pouco mais o funcionamento desses elementos e sua relação com o modelo de transporte público que se pretende representar no *feed* GTFS.

Atualmente, mais de 5 milhões de usuários mapearam mais de 5 bilhões de nós, mais de 500 milhões de vias e mais de 6 milhões de relações em todo o mundo [12]. É um banco de dados grande: quando exportado para *OpenStreetMap eXtensible Markup Language* (OSM XML) – um dos formatos de representação da base em arquivos de texto – ocupa mais de 1TB de espaço em disco [13].

Esse banco de dados é centralizado nos servidores do *OpenStreetMap*, porém pode ser replicado livremente, pois os arquivos são fornecidos para baixar. Além disso, é possível fazer consultas através de *Application Programming Interfaces* – APIs, para que não seja necessário transferir todo o banco de dados para realizar consultas [14].

### NÓ – CONCEITO E REPRESENTAÇÃO EM OSM XML

Nós, do inglês *nodes*, são a representação geográfica mais simples no modelo conceitual do *OpenStreetMap*, e representam um ponto no espaço. Cada nó tem um par de coordenadas – latitude e longitude – e um identificador numérico único (chave primária) [15], além de metadados comuns aos demais elementos (vias e relações), tais como: autor, data e hora da última alteração, versão de edição, descrição do conjunto de alterações etc. Um exemplo de nó seria um ponto representando uma padaria, um semáforo ou uma estação ferroviária.

Cada nó pode existir de forma independente, desconectado dos demais elementos, ou fazer parte de uma via. Por exemplo, um nó fazendo parte de uma linha pode representar um quebra-molas em uma rua. Pode, ainda, independentemente de pertencer ou não a uma linha, fazer parte de uma relação, onde pode adotar um papel (*role*) – por exemplo, uma estação numa rota de transporte público.

Finalmente, cada nó pode conter uma ou mais etiquetas (*tags*), que se compõem de campos chave-valor de texto livre, porém padronizados pela comunidade. São as etiquetas que fazem o entendimento do *significado* de um nó, ou seja, descrevem o que aquele nó representa.

Por exemplo, o nó que representa a estação de trem Recife no *OpenStreetMap* [16] tem o conjunto de etiquetas representadas na tabela a seguir:

Tabela 9: Etiquetas do nó que define a estação de trem Recife no *OpenStreetMap*

|  |  |  |
| --- | --- | --- |
| **Chave** | **Valor** | **Descrição** |
| railway | station | Denota que o nó é uma estação ferroviária. |
| public\_transport | station | Denota que o nó é uma estação de passageiros. |
| station | subway | Denota que a estação é de trens metropolitanos. |
| subway | yes | Marcação obsoleta de trens metropolitanos. |
| name | Recife | Nome da estação. |

Este nó tem a seguinte representação em OSM XML (as chaves do elemento-raiz <osm> do XML foram suprimidas para melhor legibilidade) [17]:

<osm>

<node id="5279929237" visible="true" version="3"

changeset="56041441" timestamp="2018-02-03T22:33:08Z"

user="cantor34" uid="7496544" lat="-8.0691737"  
 lon="-34.8853656">

<tag k="name" v="Recife"/>

<tag k="public\_transport" v="station"/>

<tag k="railway" v="station"/>

<tag k="station" v="subway"/>

<tag k="subway" v="yes"/>

</node>

</osm>

Os atributos do nó no OSM XML são os seguintes:

Tabela 10: Propriedades do elemento *node* no OSM XML

|  |  |  |
| --- | --- | --- |
| **Nome** | **Regra** | **Descrição** |
| id | Número inteiro >= 1 | Número identificador sequencial único do nó – **chave primária**. |
| visible | true ou false | Define se um nó é visível ou não. |
| version | Número inteiro >= 1 | Define quantas vezes um nó foi editado. Por exemplo, se version for igual a 1, significa que o nó nunca foi editado após ser criado. |
| changeset | Número inteiro >= 1 | Número identificador sequencial único do conjunto de edições que introduziu essa mudança no mapa – **chave estrangeira**. |
| timestamp | Data e hora no padrão ISO 8601 [18] | Data e hora da última edição, no formato YYYY-MM-DDTHH:MM:SSZ. |
| user | Campo de texto | Nome do usuário que realizou a última edição. |
| uid | Número inteiro >= 1 | Número identificador sequencial do usuário que realizou a última edição – **chave estrangeira**. |
| lat | Número >= -90 e <= 90 com até 7 casas decimais | Latitude (ângulo formado com a Linha do Equador) do nó, no formato WGS 84. Positivo para locais ao Norte da Linha do Equador e negativo para locais ao Sul desta linha. |
| lon | Número >= -180 e <= 180 com até 7 casas decimais | Longitude (ângulo formado com a Linha do Meridiano de Greenwich) do nó, no formato WGS 84. Negativo para locais a Oeste da Linha do meridiano (caso do Brasil) e positivo para locais a Leste desta linha. |

Além disso, cada elemento de nó <node> pode abrigar como elementos filhos no XML um ou mais elementos de etiqueta <tag>, com as propriedades k (chave, *key*) e v (valor, *value*) como texto livre não-nulo. Essas chaves e valores possuem valores padronizados que podem ser encontrados na documentação do *OpenStreetMap* [19].

No contexto da geração do *feed* GTFS de serviços de trem, é de especial interesse os nós que contenham as etiquetas railway=station (estações ferroviárias) [20], railway=halt (paradas ferroviárias de menor porte) [21] e railway=tram\_stop (paradas ou estações de Veículo Leve sobre Trilhos – VLT ou bondes) [22].

### VIA – CONCEITO E REPRESENTAÇÃO EM OSM XML

Vias, do inglês *ways*, são a representação no modelo conceitual do *OpenStreetMap* que conecta diferentes nós formando um caminho. São um conjunto *ordenado* de nós. Considerando um paralelo com a geometria, enquanto os nós são os vértices, as vias são as arestas.

Cada via é composta de no mínimo dois, no máximo 2.000 nós [23], não contendo nenhuma informação geográfica em si – a latitude e longitude de cada um dos pontos que a compõe reside nos pontos, não nas vias. Dessa forma, para conhecer o *trajeto* que uma linha representa, é preciso verificar as coordenadas de cada um dos pontos que a compõe.

Uma vez que os nós nas vias são ordenados, além de definir uma direção, a via também definida um sentido. Dessa forma, para mudar o sentido de uma via, é preciso inverter a ordem em que os pontos estão nela representados.

As vias podem ser abertas ou fechadas. Vias abertas representam um trajeto que não é circular, como tipicamente são as ruas ou ferrovias. Vias fechadas representam um trajeto fechado, como o contorno de um prédio, e possuem um nó repetido, de forma que o primeiro e o último nó são o mesmo.

De forma similar aos nós, as vias também podem ter etiquetas, compostas de pares chave-valor de texto livre, padronizados na documentação do projeto. Por exemplo, a via que define um trecho da Linha Sul do metrô de Recife [24] tem as etiquetas a seguir:

Tabela 11: Etiquetas da via que define parte do metrô de Recife no *OpenStreetMap*

|  |  |  |
| --- | --- | --- |
| **Chave** | **Valor** | **Descrição** |
| railway | subway | Denota que a via é de trilhos metropolitanos. |
| usage | main | Denota que a via é um trecho principal. |
| name | Linha Sul | Nome da via. |
| layer | 1 | Marcação de desambiguidade de camadas (ao cruzar com outra via, define a ordem física das vias). |
| colour | blue | Define a cor utilizada pelo operador para esta via. |

Esta via tem a seguinte representação em OSM XML (as chaves do elemento-raiz <osm> do XML e a maioria dos nós foram suprimidos para melhor legibilidade) [25]:

<osm>

<way id="128374220" visible="true" version="9"

changeset="59023145" timestamp="2018-05-16T16:23:07Z"

user="UsuárioPar" uid="367813">

<nd ref="1418908060"/>

<nd ref="1418908057"/>

<nd ref="5280081387"/>

(...)

<tag k="colour" v="blue"/>

<tag k="layer" v="1"/>

<tag k="name" v="Linha Sul"/>

<tag k="railway" v="subway"/>

<tag k="usage" v="main"/>

</way>

</osm>

Os atributos da via no OSM XML são os seguintes:

Tabela 12: Propriedades do elemento *way* no OSM XML

|  |  |  |
| --- | --- | --- |
| **Nome** | **Regra** | **Descrição** |
| id | Número inteiro >= 1 | Número identificador sequencial único da via – **chave primária**. |
| visible | true ou false | Define se uma via é visível ou não. |
| version | Número inteiro >= 1 | Define quantas vezes uma via foi editada. Por exemplo, se version for igual a 1, significa que a via nunca foi reeditada após ser criada. |
| changeset | Número inteiro >= 1 | Número identificador sequencial único do conjunto de edições que introduziu essa mudança no mapa – **chave estrangeira**. |
| timestamp | Data e hora no padrão ISO 8601 [18] | Data e hora da última edição, no formato YYYY-MM-DDTHH:MM:SSZ. |
| user | Campo de texto | Nome do usuário que realizou a última edição. |
| uid | Número inteiro >= 1 | Número identificador sequencial do usuário que realizou a última edição – **chave estrangeira**. |

Note que as propriedades do elemento <way> são exatamente as mesmas do elemento <node>, à exceção da latitude e longitude. Abaixo deste elemento, como elementos filhos, estão os nós, representados com elementos XML <nd> (de *node*, mas escrito dessa maneira para se diferenciarem dos <node>) com um atributo ref trazendo a chave estrangeira, o número identificador único de cada nó que compõe a via. Após os elementos XML dos nós, vem os elementos XML das etiquetas, seguindo o mesmo padrão chave-valor já visto anteriormente.

É importante destacar dois pontos sobre a representação de entidades no modelo conceitual do *OpenStreetMap*. Primeiro, alguns elementos possuem mais de uma forma de representação, considerando nós, vias e relações. Por exemplo, uma estação ferroviária pode ser representada tanto como um único nó no centro do prédio, quanto como uma via fechada, denotando o contorno do prédio da estação. Dessa forma, é preciso considerar a existência de alguns objetos não só como nós mas também como vias (por exemplo, as estações ferroviárias) e calcular a centróide da área [26] uma vez que a representação das paradas no GTFS deve se dar com um único ponto.

Além disso, não há a exigência de uma representação um-para-um entre entidades do mundo real e a sua representação geográfica no *OpenStreetMap*. Por exemplo, uma estrada pode ser composta de várias vias diferentes, tanto pelo limite de 2.000 nós, quanto pela necessidade de dividir a via para representar diferentes características. No exemplo da estrada, caso haja um trecho em ponte, esta estrada será representada com pelo menos três vias: uma para o trecho da ponte (com etiquetas diferentes representando a ponte) e dois para os trechos antes e depois da obra de arte. As vias são divididas à medida que seja necessário representar características diferentes.

No contexto da geração do *feed* GTFS de serviços de trem, são de especial interesse as vias que definem os trilhos, que contém as etiquetas railway=rail (trilhos de trem) [27], railway=light\_rail (trilhos de VLT) [28], railway=tram (trilhos de bonde) [29], railway=monorail (vias de monotrilho) [30] ou railway=subway (trilhos de metrô) [31], assim como railway=station (estações ferroviárias) [20] conforme mencionado anteriormente.

### RELAÇÃO – CONCEITO E REPRESENTAÇÃO EM OSM XML

Relações, do inglês *relations*, são um grupo ordenado de elementos, chamados demembros. Esses membros podem ser nós, vias ou mesmo outras relações.

As relações têm metadados de forma similar aos nós e vias, podem ter uma ou mais etiquetas, e cada membro pode, opcionalmente, ter um papel (*role*) que melhor descreve aquele elemento. Servem para dar significado a um grupo de elementos com alguma característica em comum [32].

Dentre as diferentes relações, é possível destacar alguns casos especiais interessantes. Os multipolígonos combinam diferentes vias para formar uma área complexa [33], seja definindo furos (por exemplo, um prédio com um vão central), seja definindo partes externas ao contorno principal (por exemplo, os arquipélagos brasileiros após o mar territorial [34], como Fernando de Noronha, que fazem parte da fronteira do Brasil) [35]. Neste caso, as vias que fazem parte da relação têm como papéis os termos outer (se definem uma área externa) ou inner (se definem um furo ou enclave).

Outro caso simples de utilização de relações são restrições de conversão [36]. Com este tipo de relação, se define uma conversão proibida (por exemplo, proibido virar à direita). Neste caso, as vias de origem e destino são indicadas respectivamente com os papéis from e to, enquanto o nó que representa a interseção das vias ganha o papel via.

Mais concretamente a relação que nos interessa é aquela que modela rotas de transporte público. Uma relação de linha de transporte público, por exemplo, pode agrupar as várias ruas ou trilhos (vias) pelas quais o veículo percorre, bem como as várias paradas ou estações (nós) aonde o veículo faz embarque e desembarque de passageiros [37]. Por exemplo, A relação que define a Linha Sul do metrô de Recife [38] tem as seguintes etiquetas:

Tabela 13: Etiquetas da relação que define a rota Linha Sul do metrô de Recife, sentido Cajueiro Seco → Recife

|  |  |  |
| --- | --- | --- |
| **Chave** | **Valor** | **Descrição** |
| type | route | Define que a relação é do tipo rota. |
| route | subway | Define que a rota é do tipo trem metropolitano. |
| public\_transport:version | 2 | Define o esquema utilizado para o mapeamento, para facilitar o consumo dos dados [39]. |
| network | Metrô do Recife | Define o nome da rede de transportes a qual esta rota faz parte. |
| ref | Sul | Define um código abreviado para esta rota. |
| name | Linha Sul (Cajueiro Seco → Recife) | Define um nome para esta rota. |
| operator | CBTU | Define um nome do operador da rota. |

Esta relação tem a seguinte representação em OSM XML (as chaves do elemento-raiz <osm> do XML, a maioria dos nós e das vias foram suprimidos para melhor legibilidade) [40]:

<osm>

<relation id="7869373" visible="true" version="5"  
 changeset="65863423" timestamp="2018-12-29T12:03:17Z"  
 user="danctc" uid="8107931">

<member type="node" ref="3746482927" role="stop"/>

<member type="node" ref="1551255017" role="stop"/>

<member type="node" ref="1551255016" role="stop"/>

(...)

<member type="way" ref="370452573" role=""/>

<member type="way" ref="370452575" role=""/>

<member type="way" ref="141710727" role=""/>

(...)

<tag k="name" v="Linha Sul (Cajueiro Seco --> Recife)"/>

<tag k="network" v="Metrô do Recife"/>

<tag k="operator" v="CBTU"/>

<tag k="public\_transport:version" v="2"/>

<tag k="ref" v="Sul"/>

<tag k="route" v="subway"/>

<tag k="type" v="route"/>

</relation>

</osm>

É possível verificar que os atributos do elemento <relation> são exatamente os mesmos do <way> e dos que existem no <node>: identificador, visibilidade, versão, changeset, timestamp, usuário e id do usuário. Os elementos que definem as etiquetas – <tag> – também são os mesmos.

O elemento novo na relação é a tag <member>, com o atributos type, que define o tipo do membro – node, way ou relation –, ref, chave estrangeira com o identificador do elemento referido, e role, que define o papel, se houver.

Outra relação importante é a do tipo rota mestre (route\_master), que agrupa relações do tipo rota, para denotar que duas rotas são a ida e a volta de uma mesma linha. Por exemplo, a rota mestre que define a Linha Sul do metrô de Recife tem como membros duas relações, a relação Linha Sul (Cajueiro Seco → Recife) e a relação Linha Sul (Recife → Cajueiro Seco) [41]. Esta relação tem as seguintes etiquetas:

Tabela 14: Etiquetas da relação que define a rota mestre Linha Sul do metrô de Recife

|  |  |  |
| --- | --- | --- |
| **Chave** | **Valor** | **Descrição** |
| type | route\_master | Define que a relação é do tipo rota mestre. |
| route\_master | subway | Define que a rota mestre é de trens metropolitanos. |
| name | Linha Sul | Define o nome da rota mestre. |
| ref | Sul | Define o nome abreviado da rota mestre. |
| network | Metrô do Recife | Define o nome da rede a qual esta rota mestra faz parte. |
| operator | CBTU | Define o nome do operador da rota mestre. |
| colour | #1E90FF | Define a cor utilizada pelo operador para se referir a esta rota mestra (em mapas ou diagramas, por exemplo). |
| wikidata | Q10329288 | Referência desta rota mestra no Wikidata, um projeto que visa fornecer uma base de dados para oferecer suporte a projetos como a Wikipédia [42]. |
| wikipedia | pt:Linha Sul do Sistema de Trens Urbanos do Recife | Referência ao artigo da Wikipédia que traz informações sobre esta rota mestra. |

Definidos estes conceitos, é possível traçar um paralelo entre os termos e dados do GTFS e do *OpenStreetMap*, o que será feito no próximo capítulo.

# GERANDO UM FEED GTFS A PARTIR DE DADOS DO OPENSTREETMAP

### 4.1 COMPARANDO DADOS DO GTFS E DO *OPENSTREETMAP*

Uma vez conhecidos os tipos de dados do GTFS e do *OpenStreetMap*, é possível traçar um paralelo entre eles, como descrito na Tabela 15.

Tabela 15: Comparação de dados entre o GTFS e o *OpenStreetMap*

|  |  |  |  |
| --- | --- | --- | --- |
| **GTFS** | | ***OpenStreetMap*** | |
| **Arquivo** | **Campo** | **Elemento** | **Etiqueta** |
| agency.txt | agency\_name | Rota mestre | operator |
| agency.txt | agency\_url | Não há | *Não há* |
| agency.txt | agency\_timezone | Não há | *Não há* |
| stops.txt | stop\_id | Nó indicado na relação da rota com papel stop | id |
| stops.txt | stop\_name | Nó indicado na relação da rota com papel stop | name |
| stops.txt | stop\_lat | Nó indicado na relação da rota com papel stop | lat |
| stops.txt | stop\_lon | Nó indicado na relação da rota com papel stop | lon |
| routes.txt | route\_id | Rota mestre | id |
| routes.txt | route\_short\_name | Rota mestre | ref |
| routes.txt | route\_long\_name | Rota mestre | name |
| routes.txt | route\_type | Rota mestre | route\_master |
| routes.txt | route\_color | Rota mestre | colour |
| shapes.txt | shape\_id | Rota | id |
| shapes.txt | shape\_pt\_lat | Nós das vias que fazem parte da rota | lat |
| shapes.txt | shape\_pt\_lon | Nós das vias que fazem parte da rota | lon |
| shapes.txt | shape\_pt\_sequence | Não há | Não há |

Não há correlação possível (exceto chaves estrangeiras de elementos já existentes) para os campos dos arquivos trips.txt, stop\_times.txt e calendar.txt, de forma que se faz necessária uma fonte externa para estes dados (por exemplo, entrada do usuário).

### LENDO OS DADOS DA API DO *OPENSTREETMAP*

Partindo do princípio de que a entrada do algoritmo será uma ou mais relações do tipo rota mestre (a partir de suas chaves identificadoras únicas), um algoritmo em pseudo-linguagem para a leitura dos dados a partir da API do *OpenStreetMap* seria:

para cada id de rota mestre:

lê e processa dados da relação tipo rota mestre

para cada rota que faz parte da relação tipo rota mestre:

lê e processa dados da relação tipo rota

para cada nó indicado como parada:

lê e processa dados do nó

para cada via que faz parte da relação tipo rota:

lê e processa dados da via

para cada nó que faz parte da via:

lê e processa dados do nó

Assim, há quatro níveis de profundidade para realizar leituras na API. Após realizar de maneira recursiva as consultas à API do *OpenStreetMap*, haverá na memória um objeto com o seguinte formato JSON (simplificado com as propriedades essenciais):

routeMasters: [

{ // uma relação routemaster

id,

tags: [...],

members: [

{ // uma relação route

role,

data: {

id,

tags: [...],

members: [

{ // um nó que faz parte da relação de rota

type: "node",

role: "stop",

data: {

id,

lat,

lon

tags: [...]

}

},

{ // uma via que faz parte da relação de rota

type: "way",

data: {

id,

tags: [...],

nds: [

{ // um nó que faz parte da via

data: {

id,

lat,

lon

}

}

]

}

}

]  
 }  
 }

]

}

Dessa forma os arquivos do GTFS podem ser montados com os algoritmos em pseudocódigo a seguir.

### ESCREVENDO O ARQUIVO AGENCY.TXT

As informações do arquivo agency.txt podem ser geradas a partir do seguinte algoritmo em pseudocódigo:

Para cada relação routemaster:

Se existe a etiqueta "operator":

Adicione uma linha no arquivo agency.txt contendo:

agency\_name: valor da etiqueta "operator"

agency\_url: URL do operador (entrada do usuário)

agency\_timezone: Fuso horário do operador (idem)

Note que as informações de URL do operador e fuso horário do operador não fazem parte do conjunto de informações disponíveis no *OpenStreetMap* e por isso devem ser objeto de entrada manual de dados por parte do usuário.

### ESCREVENDO O ARQUIVO ROUTES.TXT

As informações do arquivo routes.txt podem ser geradas a partir do seguinte algoritmo em pseudocódigo:

Para cada relação do tipo routemaster:

Adicione uma linha no arquivo routes.txt contendo:

route\_id: routemaster.id

route\_short\_name: valor da etiqueta "ref"

route\_long\_name: valor da etiqueta "name"

route\_type: número baseado na etiqueta "route\_master"

route\_color: valor da etiqueta "colour"

### ESCREVENDO O ARQUIVO STOPS.TXT

As informações do arquivo stops.txt podem ser geradas a partir do seguinte algoritmo em pseudocódigo:

Para cada relação do tipo routemaster:

Para cada relação do tipo route:

Para cada nó com papel "stop":

Adicione uma linha no arquivo stops.txt contendo:

stop\_id: nó.data.id

stop\_name: valor da etiqueta "name" do nó

stop\_lat: nó.data.lat

stop\_lon: nó.data.lon

Há duas limitações neste algoritmo. Primeiro, e mais importante, ele não verifica a existência prévia de uma parada, de forma que uma estação atendida por duas linhas ficará duplicada. Segundo, ele não agrupa paradas similares (de mesmo nome, com coordenadas próximas), um atributo opcional do GTFS que permite agrupar paradas em sentidos opostos num mesmo elemento.

### ESCREVENDO O ARQUIVO SHAPES.TXT

As informações do arquivo shapes.txt podem ser geradas a partir do seguinte algoritmo em pseudocódigo:

Para cada relação do tipo routemaster:

Para cada relação do tipo route:

Seja a variável shapeSequence = 1

Para cada via desta relação:

Para cada nó desta via:

Crie uma linha no arquivo shapes.txt contendo:

shape\_id: relação\_route.data.id

shape\_pt\_lat: nó.data.lat

shape\_pt\_lon: nó.data.lon

shape\_pt\_sequence: shapeSequence

Incrementa variável shapeSequence

DeduplicaPontos(shapes.txt)

Primeiro é criado um ponto para cada via presente em cada relação do tipo rota. Com isso, haverá uma série de pontos duplicados, uma vez que dadas duas vias que se conectem, o último ponto da primeira via e o primeiro ponto da segunda via são necessariamente os mesmos, isto é, são pontos que têm as mesmas coordenadas de latitude e longitude, e portanto precisam ser desduplicados.

A função de desduplicação de pontos verifica remove entradas que possuam as mesmas informações de latitude e longitude, mantendo apenas uma cópia.

### CRIANDO FORMULÁRIOS DE ENTRADA PARA PREENCHIMENTO DAS INFORMAÇÕES DE PARTIDAS

Os arquivos calendars.txt e trips.txt são inteiramente baseados em informações preenchidas pelo usuário, uma vez que informações sobre datas de funcionamento e horários de partidas não constam do *OpenStreetMap*. É preciso criar formulários para a entrada das informações com o seguinte algoritmo:

Para cada relação do tipo routemaster:

Para cada relação do tipo route:

Criar formulário

O formulário de cada rota possui um campo de texto para entrada das partidas; um conjunto de *checkboxes* - uma para cada dia da semana - para associar estas partidas a um conjunto de dias (dias úteis ou finais de semana, por exemplo); um botão para duplicação destes campos (para entrada de horários diferentes para dias úteis e finais de semana, por exemplo); além de campos para data de validade inicial e final.

### ESCREVENDO O ARQUIVO CALENDARS.TXT

Uma vez que o usuário entrou os dados do item anterior, poderemos gerar o arquivo calendars.txt com o seguinte algoritmo:

Para cada rota no arquivo de rotas:

Para cada calendário criado para esta rota:

Crie uma linha no arquivo calendars.txt contendo:

service\_id: concatenação do ID da rota com o índice do calendário

monday: 1 se o campo a caixa referente à segunda-feira está ativa, 0 se inativa

tuesday: 1 se o campo a caixa referente à terça-feira está ativa, 0 se inativa

wednesday: 1 se o campo a caixa referente à quarta-feira está ativa, 0 se inativa

thursday: 1 se o campo a caixa referente à quinta-feira está ativa, 0 se inativa

friday: 1 se o campo a caixa referente à seta-feira está ativa, 0 se inativa

saturday: 1 se o campo a caixa referente ao sábado está ativa, 0 se inativa

sunday: 1 se o campo a caixa referente à segunda-feira está ativa, 0 se inativa

start\_date: data de início de validade

end\_date: data de fim de validade

Uma vez que os parâmetros start\_date e end\_date são obrigatórios na especificação do GTFS, mesmo que o usuário tenha deixado os campos em branco eles devem estar especificados com valores padrão (19000101 e 20991231 respectivamente).

### ESCREVENDO O ARQUIVO TRIPS.TXT

O arquivo trips.txt conecta entradas em outros arquivos, como uma tabela de relação do tipo "n para m", baseado na lista de partidas criadas pelo usuário. O algoritmo tem a seguinte forma:

Para cada rota:

Para cada calendário desta rota:

Para cada horário na lista de partidas:

Cria uma linha no arquivo trips.txt contendo:

route\_id: ID da rota

calendar\_id: Concatenação do ID da rota com o índice do calendário

trip\_id: Concatenação do ID da rota com o índice do calendário e o índice do horário

### ESCREVENDO O ARQUIVO STOP\_TIMES.TXT

O arquivo stop\_times.txt lista os horários de chegada e partida em cada estação, baseados nos horários de partida iniciais entrados pelo usuário. O algoritmo tem a seguinte forma:

Para cada rota:

Para cada calendário:

Para cada horário:

Seja variável stopSequence = 1

Para cada parada:

Cria uma linha no arquivo stop\_times.txt contendo:

trip\_id: identificador único da viagem, sendo a concatenação do ID da rota, índice do calendário e índice do horário

arrival\_time: horário de chegada na estação, calculado por uma função auxiliar

departure\_time: horário de partida da estação, calculado por uma função auxiliar

stop\_id: id da parada

stop\_sequence: variável stopSequence  
 Incrementa variável stopSequence

A função auxiliar calcula os horários da seguinte forma:

horário de chegada = horário inicial + (índice das paradas x tempo padrão entre paradas)

horário de partida = horário de chegada + tempo de abertura de portas padrão

Versões futuras do trabalho poderiam calcular o tempo entre paradas de forma variável - considerando a distância, velocidade máxima do trecho e a curva de aceleração dos veículos, por exemplo - bem como permitir especificar tempo de abertura de portas diferenciado - para estações ou faixas de horários mais ou menos movimentadas, por exemplo.

# CONCLUSÕES E TRABALHOS FUTUROS

O trabalho conclui que sim, é possível traçar um paralelo entre as duas plataformas e converter dados do *OpenStreetMap* para a geração de um *feed* GTFS, ainda que de forma semiautomática, com alguma entrada do usuário, uma vez que nem todos os dados obrigatórios do GTFS constam do rol de dados disponíveis no *OpenStreetMap*.

Trabalhos futuros podem seguir em duas direções: primeiro, trabalhar no agrupamento de paradas repetidas, uma vez que estações com serviço de duas linhas, ou da mesma linha em sentido contrário - caso típico das estações ferroviárias, que via de regra possuem serviços em dois sentidos - figurarão no arquivo GTFS como paradas separadas. Neste caso, será preciso que um algoritmo verifique paradas de mesmo nome e as junte em um agrupamento de estações. O padrão GTFS especifica no arquivo stops.txt o campo opcional parent\_station justamente com o objetivo de criar uma entidade que agrupe diferentes pontos de parada [7]. Por outro lado, este agrupamento deveria ser parametrizável - por exemplo por distância e/ou nome de sistema - ou ainda ser sugerido porém aceito ou não pelo usuário, uma vez que poderia gerar falsos positivos. Por exemplo, no sistema ferroviário do Rio de Janeiro existem duas estações ferroviárias de nome Maracanã, uma operada pela concessionária MetrôRio e outra operada pela concessionária SuperVia, que são estações separadas, ainda que ligadas por um corredor de integração, de forma que o *feed* GTFS deveria trazer duas estações distintas.

A segunda possibilidade de trabalho futuro seria parametrizar melhor o tempo entre paradas e o tempo na parada (tempo de abertura de portas) especificado como função auxiliar no capítulo 4.9. Ao invés de utilizar um valor fixo - uma aproximação muito bruta do observado na realidade - poderia haver um cálculo levando em consideração a velocidade média ou máxima do trecho, além da curva de aceleração e frenagem do veículo, assim como levar em conta diferentes tempos de abertura de portas para estações mais movimentadas, ou horário de pico.

# REFERÊNCIAS BIBLIOGRÁFICAS

1. O que é GTFS?. **Google Transit**, 2019. Disponível em: <<https://developers.google.com/transit/gtfs/?hl=pt-br>>. Acesso em: 18 de fev. de 2019.
2. OPENSTREETMAP. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/about>>. Acesso em: 25 de fev. de 2019.
3. PORTAL CBTU - Home. **CBTU**, 2019. Disponível em <<https://www.cbtu.gov.br>>. Acesso em: 25 de fev. de 2019.
4. A Brief History of GTFS | Institute of Transportation Studies Library. **LEVINE, Kendra K.**, 2019. Disponível em <<http://library.its.berkeley.edu/node/265>>. Acesso em: 25 de fev. de 2019.
5. OSM file formats. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/OSM_file_formats>>. Acesso em: 25 de fev. de 2019.
6. OPENSTREETMAP Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Main_Page>>. Acesso em: 25 de fev. de 2019.
7. REFERÊNCIA da GTFS (Especificação Geral de Feeds de Transporte Público). **Google Transit**, 2019. Disponível em: <<https://developers.google.com/transit/gtfs/reference?hl=pt-br>>. Acesso em 18 de fev. de 2019.
8. WORLD Geodetic System website of the NGA. **National Geospatial-Intelligence Agency**. Arquivado do original em abr. de 2012. Disponível em: <<https://web.archive.org/web/20120402143802/https://www1.nga.mil/ProductsServices/GeodesyandGeophysics/WorldGeodeticSystem/Pages/default.aspx>> . Acesso em: 25 de fev. de 2019.
9. OPENSTREETMAP [About]. **OpenStreetMap**, 2019. Disponível em: <<https://www.openstreetmap.org/about>>. Acesso em 26 de fev. de 2019.
10. OPENSTREETMAP: A Wikipédia dos Mapas. **Dados Abertos Brasil**, 2018. Disponível em <<http://www.dadosabertosbrasil.com.br/node/64>>. Acesso em 4 de mar. de 2019.
11. ELEMENTS - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em: <<https://wiki.openstreetmap.org/wiki/Elements>>. Acesso em 26 de fev. de 2019.
12. OPENSTREETMAP Statistics. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/stats/data_stats.html>>. Acesso em 26 de fev. de 2019.
13. PLANET.OSM. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Planet.osm>>. Acesso em 26 de fev. de 2019.
14. API - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/API>>. Acesso em 26 de fev. de 2019.
15. NODE - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Node>>. Acesso em 26 de fev. de 2019.
16. PONTO: Recife (5279929237) | OpenStreetMap. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/node/5279929237>>. Acesso em 26 de fev. de 2019.
17. [SEM TÍTULO]. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/api/0.6/node/5279929237>>. Acesso em 26 de fev. de 2019.
18. DATE and Time Formats. **W3 Consortium**, 1997. Disponível em <<https://www.w3.org/TR/NOTE-datetime>>. Acesso em 26 de fev. de 2019.
19. MAP Features - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Map_Features>>. Acesso em 26 de fev. de 2019.
20. TAG:railway=station - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dstation>>. Acesso em 5 de mar. de 2019.
21. TAG:railway=halt - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dhalt>>. Acesso em 5 de mar. de 2019.
22. TAG:railway=tram\_stop - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dtram_stop>>. Acesso em 5 de mar. de 2019.
23. WAY - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Way>>. Acesso em 4 de mar. de 2019.
24. LINHA: Linha Sul (128374220) | OpenStreetMap. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/way/128374220>>. Acesso em 4 de mar. de 2019.
25. [SEM TÍTULO]. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/api/0.6/way/128374220>>. Acesso em 4 de mar. de 2019.
26. DAN Scientia: A centróide de um polígono. **Daniel Madeira**, 2009. Disponível em <<http://dan-scientia.blogspot.com/2009/10/centroide-de-um-poligono.html>>. Acesso em 5 de mar. de 2019.
27. TAG:railway=rail - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Drail>>. Acesso em 5 de mar. de 2019.
28. TAG:railway=light\_rail - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dlight_rail>>. Acesso em 5 de mar. de 2019.
29. TAG:railway=tram - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dtram>>. Acesso em 5 de mar. de 2019.
30. TAG:railway=monorail - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dmonorail>>. Acesso em 5 de mar. de 2019.
31. TAG:railway=subway - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Tag:railway%3Dsubway>>. Acesso em 5 de mar. de 2019.
32. RELATION - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Relation>>. Acesso em 6 de mar. de 2019.
33. RELATION:multipolygon - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Relation:multipolygon>>. Acesso em 6 de mar. de 2019.
34. BRASIL. LEI Nº 8.617, DE 4 DE JANEIRO DE 1993. **Dispõe sobre o mar territorial, a zona contígua, a zona econômica exclusiva e a plataforma continental brasileiros, e dá outras providências**, Brasília, DF, jan. de 1993. Disponível em <<http://www.planalto.gov.br/ccivil_03/LEIS/L8617.htm>>. Acesso em 6 de mar. de 2019.
35. RELAÇÃO: Brasil (59470). **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/relation/59470>>. Acesso em 6 de mar. de 2019.
36. RELATION:restriction - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Relation:restriction>>. Acesso em 6 de mar. de 2019.
37. RELATION:route - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Relation:route>>. Acesso em 6 de mar. de 2019.
38. RELAÇÃO: Linha Sul (Cajueiro Seco → Recife) (7869373). **OpenStreetMap**, 2019. Disponível em <[https://www.openstreetmap.org/relation/7869373>](https://www.openstreetmap.org/relation/7869373). Acesso em 6 de mar. de 2019.
39. PUBLIC transport - OpenStreetMap Wiki. **OpenStreetMap**, 2019. Disponível em <<https://wiki.openstreetmap.org/wiki/Public_transport#Different_tagging_schemas>>. Acesso em 6 de mar. de 2019.
40. [SEM TÍTULO]. **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/api/0.6/relation/7869373>>. Acesso em 6 de mar. de 2019.
41. RELAÇÃO: Linha Sul (7869374). **OpenStreetMap**, 2019. Disponível em <<https://www.openstreetmap.org/relation/7869374>>. Acesso em 6 de mar. de 2019.
42. WIKIDATA - Wikipédia, a enciclopédia livre. **Wikipédia**, 2019. Disponível em <<https://pt.wikipedia.org/wiki/Wikidata>>. Acesso em 6 de mar. de 2019.