

Título do capítulo	CAPÍTULO 4 DADOS GTFS
Autor(es)	Rafael H. M. Pereira Daniel Herszenhut
DOI	DOI: http://dx.doi.org/10.38116/9786556350547cap4

Título do livro	Introdução à Acessibilidade Urbana: um guia prático em R
Organizadores(as)	Rafael H. M. Pereira Daniel Herszenhut
Volume	1
Série	-
Cidade	Rio de Janeiro
Editora	Instituto de Pesquisa Econômica Aplicada (Ipea)
Ano	2023
Edição	1a
ISBN	9786556350547
DOI	DOI: http://dx.doi.org/10.38116/9786556350547

© Instituto de Pesquisa Econômica Aplicada – ipea 2023

As publicações do Ipea estão disponíveis para *download* gratuito nos formatos PDF (todas) e EPUB (livros e periódicos). Acesso: <http://www.ipea.gov.br/portal/publicacoes>

As opiniões emitidas nesta publicação são de exclusiva e inteira responsabilidade dos autores, não exprimindo, necessariamente, o ponto de vista do Instituto de Pesquisa Econômica Aplicada ou do Ministério do Planejamento e Orçamento.

É permitida a reprodução deste texto e dos dados nele contidos, desde que citada a fonte. Reproduções para fins comerciais são proibidas.

4 DADOS GTFS

O formato GTFS é uma especificação aberta e colaborativa que visa descrever os principais componentes de uma rede de transporte público. Originalmente criada em meados dos anos 2000 por uma parceria entre Google e TriMet, a agência de transporte de Portland, em Oregon, nos Estados Unidos, a especificação hoje é utilizada por agências de transporte em milhares de cidades, espalhadas por todos os continentes do globo (McHugh, 2013). Atualmente, essa especificação é dividida em dois componentes distintos:

- GTFS Schedule, ou GTFS Static, que contém o cronograma planejado de linhas de transporte público, informações sobre suas tarifas e informações espaciais sobre os seus itinerários; e
- GTFS Realtime, que contém informações de localização de veículos em tempo real e alertas de possíveis atrasos, de mudanças de percurso e de eventos que possam interferir no cronograma planejado.

Ao longo desta seção, focaremos no formato GTFS Schedule, por ser o mais amplamente utilizado por agências de transporte e em análises de acessibilidade.⁹

Por ser uma especificação aberta e colaborativa, o formato GTFS tenta abarcar em sua definição um grande número de usos distintos que agências de transporte e desenvolvedores de ferramentas possam lhe dar. No entanto, agências e *softwares* podem ainda assim depender de informações que não constem na especificação oficial. Surgem, dessa forma, extensões da especificação, algumas das quais podem eventualmente se tornar parte da especificação oficial, caso isso seja aceito pela comunidade. Nesta seção, focaremos em um subconjunto de informações presentes no formato GTFS Schedule “puro” e, portanto, não cobriremos suas extensões.

4.1 Estrutura de arquivos GTFS

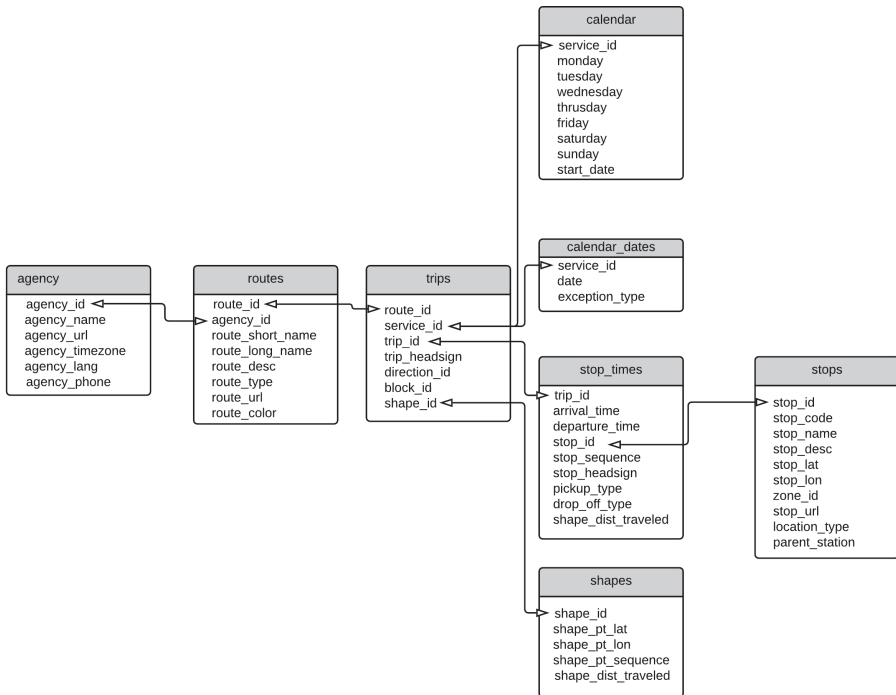
Arquivos no formato GTFS Schedule (daqui em diante chamado apenas de GTFS) também são conhecidos pela denominação *feed*.¹⁰ Um *feed* é nada mais do que um arquivo comprimido em formato .zip que contém um conjunto de tabelas, salvas em formato .txt, com algumas informações sobre a rede de transporte público (localização das paradas, frequências das viagens, traçado das rotas, entre outras). Como em uma base de dados relacional, as tabelas de um *feed* possuem colunas-chaves que permitem vincular os dados de rotas, viagens e tabelas de

9. Mais informações sobre o formato GTFS Realtime estão disponíveis em: <<https://gtfs.org/realtime/>>.

10. Neste livro, utilizaremos os termos *feed*, arquivo GTFS e dados GTFS como sinônimos.

horários entre si. O esquema geral do formato GTFS é apresentado na figura 4, que mostra algumas das principais tabelas que compõem a especificação e destaca como elas se relacionam a partir de suas colunas-chaves.

FIGURA 4
Esquema do formato GTFS



Fonte: Pereira, Andrade e Vieira (2022).

Ao todo, 22 tabelas compõem o formato GTFS.¹¹ Nem todas, no entanto, devem estar obrigatoriamente presentes para que um *feed* seja considerado válido, sendo consideradas, portanto, opcionais. A especificação classifica cada tabela conforme sua obrigatoriedade em três possíveis categorias: obrigatórias, opcionais e condicionalmente obrigatórias (quando a obrigatoriedade de uma tabela depende da existência de uma determinada tabela, coluna ou valor). Para fins de simplicidade, neste livro consideraremos apenas as duas primeiras categorias e faremos comentários quanto à obrigatoriedade de cada tabela quando apropriado. Dessa forma, as tabelas ficam classificadas conforme a seguir.

11. Conforme a [especificação oficial](#), versão da revisão de 9 de maio de 2022.

- 1) Obrigatórias: `agency.txt`; `stops.txt`; `routes.txt`; `trips.txt`; `stop_times.txt`; `calendar.txt`.
- 2) Opcionais: `calendar_dates.txt`; `fare_attributes.txt`; `fare_rules.txt`; `fare_products.txt`; `fare_leg_rules.txt`; `fare_transfer_rules.txt`; `areas.txt`; `stop_areas.txt`; `shapes.txt`; `frequencies.txt`; `transfers.txt`; `pathways.txt`; `levels.txt`; `translations.txt`; `feed_info.txt`; `attributions.txt`.

Ao longo desta seção, aprenderemos sobre a estrutura básica de um arquivo GTFS e das tabelas que o compõem. Portanto, vamos olhar apenas para as tabelas obrigatórias e para as tabelas opcionais mais frequentemente utilizadas por produtores e consumidores desses arquivos.¹²

Na demonstração que será feita aqui, utilizaremos um subconjunto de dados provenientes do *feed* da cidade de São Paulo criado pela São Paulo Transporte (SPTrans)¹³ e baixado em outubro de 2019. O *feed* contém as seis tabelas obrigatórias e mais duas tabelas opcionais bastante utilizadas, a `shapes.txt` e a `frequencies.txt`, o que permite uma boa visão geral sobre o formato GTFS.

4.1.1 `agency.txt`

Arquivo utilizado para descrever as operadoras de transporte que atuam no sistema descrito pelo arquivo GTFS. Embora o termo *agency* (agência) seja usado em lugar de *operators* (operadoras), por exemplo, fica a cargo do produtor do *feed* definir quais instituições serão listadas na tabela.

Por exemplo: múltiplas concessionárias de ônibus atuam em um determinado local, mas todo o planejamento de cronograma e de tarifa é realizado por uma única instituição, em geral uma secretaria de transporte ou empresa pública específica. Essa instituição é também entendida pelos usuários do sistema como a operadora, de fato. Nesse caso, devemos listar a instituição responsável pelo planejamento na tabela.

Agora, imagine um sistema em que a agência de transporte público local transfere a responsabilidade da operação de um sistema multimodal a diversas empresas, por meio de concessões. Cada uma dessas empresas é responsável pelo planejamento de cronogramas e tarifas dos modos que operam, desde que sejam seguidos determinados parâmetros preestabelecidos em contrato. Sendo assim, devemos listar as operadoras (concessionárias) na tabela, e não a agência de transporte público em si.

12. Para mais informações sobre as tabelas e as colunas não abordadas neste texto, pode-se verificar a [especificação oficial](#).

13. Disponível em: <<https://www.sptrans.com.br/desenvolvedores/>>.

A tabela 1 mostra o arquivo `agency.txt` do *feed* da SPTrans. Como podemos ver, os responsáveis pelo *feed* optaram por listar a própria empresa no arquivo, e não as concessionárias que operam os ônibus e o metrô da cidade.

TABELA 1
Exemplo de arquivo `agency.txt`

agency_id	agency_name	agency_url	agency_timezone	agency_lang
1	SPTRANS	http://www.sptrans.com.br/?versao=011019	America/Sao_Paulo	pt

Fonte: SPTrans.

É necessário notar que, embora estejamos apresentando o `agency.txt` em formato de tabela, o arquivo deve ser formatado como se fosse salvo em formato `.csv`. Ou seja, os valores de cada célula da tabela devem ser separados por vírgulas e cada linha da tabela deve constar em uma linha no arquivo. A tabela 1, por exemplo, é definida da seguinte forma:

```
agency_id,agency_name,agency_url,agency_timezone,agency_lang
1,SPTRANS,http://www.sptrans.com.br/?versao=011019,America/
Sao_Paulo,pt
```

Por uma questão de comunicação e interpretação dos dados, apresentaremos os exemplos em formato de tabela. É importante ter em mente, porém, que essas tabelas são organizadas como mostrado anteriormente.

4.1.2 `stops.txt`

Arquivo usado para descrever as paradas de transporte público que compõem o sistema. Os pontos listados neste arquivo podem fazer menção a paradas mais simples (como pontos de ônibus), estações, plataformas, entradas e saídas de estações etc. A tabela 2 mostra o `stops.txt` do *feed* da SPTrans.

TABELA 2
Exemplo de arquivo `stops.txt`

stop_id	stop_name	stop_desc	stop_lat	stop_lon
706325	Parada 14 Bis B/C	Viad. Dr. Plínio De Queiroz, 901	-23.55593	-46.65011
810602	R. Sta. Rita, 56	Ref.: R. Bresser/R. João Boemer	-23.53337	-46.61229
910776	Av. Do Estado, 5854	Ref.: Rua Dona Ana Néri	-23.55896	-46.61520
1010092	Parada Caetano Pinto	Av. Rangel Pestana, 1249 Ref.: Rua Caetano Pinto/rua Prof. Batista De Andrade	-23.54615	-46.62218
1010093	Parada Piratininga	Av. Rangel Pestana, 1479 Ref.: Rua Monsenhor Andrade	-23.54509	-46.62006
1010099	R. Xavantes, 612	Ref.: Rua Joli	-23.53545	-46.61368

Fonte: SPTrans.

As colunas `stop_id` e `stop_name` servem como identificadores de cada parada, porém cumprem papéis distintos. O principal propósito da `stop_id` é identificar relações entre esta tabela e outras que compõem a especificação (como veremos mais à frente no arquivo `stop_times.txt`, por exemplo). Já a coluna `stop_name` cumpre o papel de um identificador facilmente reconhecido pelo passageiro. Seus valores, portanto, costumam ser nomes de estações, nomes de pontos de interesse da cidade ou endereços (como no caso do *feed* da SPTrans).

A coluna `stop_desc`, presente no *feed* da SPTrans, é opcional e permite à agência de transporte adicionar uma descrição de cada parada e de seu entorno. As colunas `stop_lat` e `stop_lon`, por fim, são as responsáveis por associar cada parada a uma posição espacial, por meio de suas coordenadas geográficas de latitude e longitude.

Entre as colunas opcionais não presentes no `stops.txt` deste *feed* estão a `location_type` e a `parent_station`. A `location_type` é utilizada para denotar o tipo de localização a que cada ponto se refere. Quando ausente, todos os pontos são interpretados como paradas de transporte público, mas valores distintos podem ser usados para distinguir uma parada (`location_type = 0`) de uma estação (`location_type = 1`) ou uma área de embarque (`location_type = 2`), por exemplo. A coluna `parent_station`, por sua vez, é utilizada para descrever relações de hierarquia entre dois pontos. Por exemplo, uma área de desembarque deve dizer a qual parada/plataforma ela pertence, assim como uma parada/plataforma pode também, opcionalmente, listar a qual estação ela pertence.

4.1.3 routes.txt

Arquivo usado para descrever as linhas de transporte público que rodam no sistema, incluindo os modos de transporte utilizados em cada uma. A tabela 3 mostra o `routes.txt` do *feed* da SPTrans.

TABELA 3
Exemplo de arquivo `routes.txt`

route_id	agency_id	route_short_name	route_long_name	route_type
CPTM L07	1	CPTM L07	JUNDIAI - LUZ	2
CPTM L08	1	CPTM L08	AMADOR BUENO - JULIO PRESTES	2
CPTM L09	1	CPTM L09	GRAJAU - OSASCO	2
CPTM L10	1	CPTM L10	RIO GRANDE DA SERRA - BRÁS	2
CPTM L11	1	CPTM L11	ESTUDANTES - LUZ	2
CPTM L12	1	CPTM L12	CALMON VIANA - BRAS	2

Fonte: SPTrans.

Assim como no caso do arquivo `stops.txt`, a tabela do `routes.txt` também possui diferentes colunas que apontam o identificador de cada linha (`route_id`) e o seu nome. Nesse caso, no entanto, existem duas colunas de nome: a `route_short_name` e a `route_long_name`. A primeira diz respeito ao nome da linha, usualmente utilizado por passageiros no dia a dia, enquanto o segundo tende a ser um nome mais descritivo. A SPTrans, por exemplo, optou por destacar os pontos finais de cada linha nessa coluna. Podemos notar também que os mesmos valores se repetem nas colunas `route_id` e `route_short_name`, o que não é obrigatório nem proibido – nesse caso, o produtor do *feed* julgou que os nomes das linhas poderiam funcionar satisfatoriamente como identificadores por serem razoavelmente curtos e não se repetirem.

A coluna `agency_id` é a chave que permite relacionar a tabela das rotas com a tabela descrita no `agency.txt`. Ela faz menção a uma agência descrita naquele arquivo, a agência de id 1 (a própria SPTrans). Essa coluna é opcional no caso de *feeds* em que existe apenas uma agência, porém é obrigatória nos casos em que existe mais de uma. Imaginemos, por exemplo, um *feed* que descreve um sistema multimodal que conta com um corredor de metrô e diversas linhas de ônibus: uma configuração possível de `routes.txt` descreveria as linhas de metrô como de responsabilidade da operadora do metrô, e as de ônibus como de responsabilidade da empresa responsável pelo planejamento das linhas de ônibus, por exemplo.

A coluna `route_type` é utilizada para descrever o modo de transporte utilizado em cada linha. Essa coluna aceita diferentes números, cada um representando um determinado modo. Esse exemplo descreve linhas de trem, cujo valor numérico correspondente é 2. Os valores correspondentes para outros modos de transporte são listados na [especificação](#).

4.1.4 `trips.txt`

Arquivo usado para descrever as viagens realizadas no sistema. A viagem é a unidade básica de movimento do formato GTFS: cada viagem é associada a uma linha de transporte público (`route_id`), a um serviço que opera em determinados dias da semana (como veremos mais à frente, no arquivo `calendar.txt`) e a uma trajetória espacial (como será mostrado no arquivo `shapes.txt`). A tabela 4 mostra o `trips.txt` do *feed* da SPTrans.

TABELA 4
Exemplo de arquivo `trips.txt`

trip_id	route_id	service_id	trip_headsign	direction_id	shape_id
CPTM L07-0	CPTM L07	USD	JUNDIAI	0	17846
CPTM L07-1	CPTM L07	USD	LUZ	1	17847
CPTM L08-0	CPTM L08	USD	AMADOR BUENO	0	17848
CPTM L08-1	CPTM L08	USD	JULIO PRESTES	1	17849
CPTM L09-0	CPTM L09	USD	GRAJAU	0	17850
CPTM L09-1	CPTM L09	USD	OSASCO	1	17851

Fonte: SPTTrans.

A coluna `trip_id` identifica cada uma das viagens descritas na tabela, assim como a `route_id` faz referência a uma linha de transporte público identificada no arquivo `routes.txt`. A coluna `service_id` identifica serviços que determinam os dias da semana em que cada uma das viagens opera (dias úteis, finais de semana, uma mistura dos dois etc.), descritos detalhadamente no arquivo `calendar.txt`. A última coluna à direita na tabela 4 é a `shape_id`, que identifica a trajetória espacial de cada uma das viagens, descrita em detalhes no arquivo `shapes.txt`.

As duas colunas restantes, `trip_headsign` e `direction_id`, são opcionais e devem ser utilizadas para descrever o sentido/destino da viagem. A primeira, `trip_headsign`, é utilizada para ditar o texto que aparece no letreiro de veículos (no caso de um ônibus, por exemplo) ou em painéis informativos (como em metrô e trens) destacando o destino da viagem. Já a coluna `direction_id` é frequentemente utilizada em conjunto com a primeira para dar uma conotação de ida ou volta para cada viagem, em que 0 representa ida e 1, volta, ou vice-versa (assim como ida e volta são conceitos que mudam conforme o referencial, os valores 0 e 1 podem ser usados como desejado, desde que um represente um sentido e o outro, o contrário). No exemplo, as duas primeiras linhas são viagens que fazem menção à mesma rota de transporte público (CPTM L07), porém em sentidos opostos: uma corre em direção a Jundiaí e a outra, à estação da Luz.

4.1.5 `calendar.txt`

Arquivo usado para descrever os diferentes tipos de serviço existentes no sistema. Um serviço, nesse contexto, denota um conjunto de dias da semana em que viagens são realizadas. Cada serviço também é definido pela data em que começa a valer e pela data a partir da qual ele não é mais válido. A tabela 5 mostra o `calendar.txt` do *feed* da SPTTrans.

TABELA 5
Exemplo de arquivo `calendar.txt`

service_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday	start_date	end_date
USD	1	1	1	1	1	1	1	20080101	20200501
U__	1	1	1	1	1	0	0	20080101	20200501
US_	1	1	1	1	1	1	0	20080101	20200501
_SD	0	0	0	0	0	1	1	20080101	20200501
__D	0	0	0	0	0	0	1	20080101	20200501
__S	0	0	0	0	0	1	0	20080101	20200501

Fonte: SPTTrans.

A coluna `service_id` identifica cada um dos serviços descritos na tabela. Como mostrado anteriormente, esse identificador é usado também no arquivo `trips.txt` e é o responsável por associar cada viagem a um determinado serviço.

As colunas `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday` e `sunday` (segunda-feira a domingo, em inglês) são utilizadas para delimitar os dias em que cada serviço funciona. O valor 1 significa que o serviço opera naquele dia, enquanto o valor 0 significa que ele não opera. Como podemos ver na tabela 5, o serviço USD opera em todos os dias da semana. Já o serviço U__ opera apenas em dias úteis.

Por fim, as colunas `start_date` e `end_date` delimitam o intervalo em que cada serviço é válido. As datas do formato GTFS são sempre formatadas segundo a regra YYYYMMDD, em que os primeiros quatro números definem o ano, os dois subsequentes definem o mês e os últimos dois, o dia. O valor 20220428, por exemplo, representa o dia 28 de abril de 2022.

4.1.6 `shapes.txt`

Arquivo usado para descrever a trajetória espacial de cada viagem listada no *feed*. Esse arquivo é opcional, mas é fortemente recomendado que agências de transporte o incluam em seus arquivos GTFS. A tabela 6 mostra o `shapes.txt` do *feed* da SPTTrans.

TABELA 6
Exemplo de arquivo `shapes.txt`

shape_id	shape_pt_lat	shape_pt_lon	shape_pt_sequence
17846	-23.53517	-46.63535	1
17846	-23.53513	-46.63548	2
17846	-23.53494	-46.63626	3
17846	-23.53473	-46.63710	4
17846	-23.53466	-46.63735	5
17846	-23.53416	-46.63866	6

Fonte: SPTTrans.

A coluna `shape_id` identifica cada uma das trajetórias descritas na tabela. Como mostrado anteriormente, esse identificador é usado também no arquivo `trips.txt` e é o responsável por associar cada viagem à sua trajetória espacial. Diferentemente de todos os outros identificadores que vimos até então, no entanto, o identificador `shape_id` se repete em diversas observações da tabela. Isso porque o arquivo associa cada `shape_id` a uma série de pontos espaciais, cujas coordenadas geográficas são descritas nas colunas `shape_pt_lat` e `shape_pt_lon`. A coluna `shape_pt_sequence` lista a sequência na qual os pontos se conectam para formar a trajetória de cada `shape_id`. Os valores listados nessa coluna devem ser ordenados de forma crescente.

4.1.7 `stop_times.txt`

Arquivo usado para descrever a tabela de horários de cada viagem, incluindo o horário de chegada e partida em cada uma das paradas. A formatação desse arquivo depende da existência ou não de um arquivo `frequencies.txt`, detalhe que cobriremos mais adiante. Por enquanto, olharemos para o `stop_times.txt` do *feed* da SPTrans, que também conta com um `frequencies.txt`, na tabela 7.

TABELA 7
Exemplo de arquivo `stop_times.txt`

<code>trip_id</code>	<code>arrival_time</code>	<code>departure_time</code>	<code>stop_id</code>	<code>stop_sequence</code>
CPTM L07-0	04:00:00	04:00:00	18940	1
CPTM L07-0	04:08:00	04:08:00	18920	2
CPTM L07-0	04:16:00	04:16:00	18919	3
CPTM L07-0	04:24:00	04:24:00	18917	4
CPTM L07-0	04:32:00	04:32:00	18916	5
CPTM L07-0	04:40:00	04:40:00	18965	6

Fonte: SPTrans.

A viagem cuja tabela de horários está sendo descrita é identificada pela coluna `trip_id`. De forma análoga ao que acontece na tabela de trajetórias, um mesmo `trip_id` se repete em muitas observações da tabela. Isso porque, assim como a trajetória é composta por uma sequência de pontos espaciais, a tabela de horários é composta por uma sequência de diversos horários de partida/chegada em diversas paradas de transporte público.

As colunas seguintes, `arrival_time`, `departure_time` e `stop_id`, são as responsáveis por descrever o cronograma de cada viagem, associando um horário de chegada e um horário de partida a cada uma das paradas da viagem. As colunas de horário são formatadas segundo a regra HH:MM:SS, em que os dois primeiros números definem a hora, os dois seguintes, os minutos e os últimos dois, os

segundos. Essa formatação aceita valores de hora maiores que 24: por exemplo, se uma viagem parte às 23h, mas só chega a uma determinada estação 1h da manhã do dia seguinte, seu horário de chegada deve ser registrado como 25:00:00, e não 01:00:00. A coluna `stop_id`, por sua vez, associa os horários de chegada e partida a uma parada descrita no arquivo `stops.txt`. Por fim, a coluna `stop_sequence` lista a sequência na qual cada parada se conecta às demais para formar o cronograma da viagem. Seus valores devem ser sempre ordenados de forma crescente.

Vale destacar aqui a diferença entre os arquivos `shapes.txt` e `stop_times.txt`. Embora os dois descrevam uma viagem espacialmente, eles o fazem de forma diferente. O `stop_times.txt` descreve a sequência de paradas e horários que compõem um cronograma, mas nada diz sobre o trajeto percorrido pelo veículo entre cada uma das paradas. Já o `shapes.txt` traz a trajetória detalhada da viagem como um todo, mas não descreve em que ponto do espaço estão as paradas da viagem. Quando usamos os dois arquivos em conjunto, portanto, sabemos tanto o cronograma de cada viagem quanto a trajetória espacial da viagem entre paradas.

4.1.8 `frequencies.txt`

Arquivo opcional usado para descrever a frequência de cada viagem dentro de um determinado período do dia. A tabela 8 mostra o `frequencies.txt` do *feed* da SPTrans.

TABELA 8
Exemplo de arquivo `frequencies.txt`

<code>trip_id</code>	<code>start_time</code>	<code>end_time</code>	<code>headway_secs</code>
CPTM L07-0	04:00:00	04:59:00	720
CPTM L07-0	05:00:00	05:59:00	360
CPTM L07-0	06:00:00	06:59:00	360
CPTM L07-0	07:00:00	07:59:00	360
CPTM L07-0	08:00:00	08:59:00	360
CPTM L07-0	09:00:00	09:59:00	480

Fonte: SPTrans.

A viagem cuja frequência está sendo descrita é identificada pela coluna `trip_id`. Novamente, um mesmo identificador pode aparecer em várias observações da tabela, pois a especificação prevê que uma mesma viagem pode ter frequências diferentes ao longo do dia (como em horários de pico e fora do pico, por exemplo). Assim, cada linha da tabela se refere à frequência de uma determinada viagem dentro de um intervalo de tempo especificado pelas colunas `start_time` e `end_time`.

Dentro do período especificado por essas duas colunas, a viagem possui um *headway* detalhado na coluna *headway_secs*. O *headway* é o tempo que separa a passagem de dois veículos que operam a mesma linha de transporte público. No caso da tabela *frequencies.txt*, esse tempo deve ser especificado em segundos. Um valor de 720 entre 4h e 5h, portanto, significa que a viagem CPTM L07-0 ocorre de doze em doze minutos dentro desse período.

Usando as tabelas frequencies.txt e stop_times.txt conjuntamente

É importante entender, agora, como a presença da tabela *frequencies.txt* altera a especificação da tabela *stop_times.txt*. Como podemos ver no exemplo da tabela *stop_times.txt* (tabela 7), a viagem CPTM L07-0 parte da primeira parada às 4h e chega na segunda às 4h08. O cronograma de chegada e saída de uma mesma parada de uma viagem, no entanto, não pode ser definido mais de uma vez na tabela. Como então definir o cronograma das viagens que partem às 4h12, 4h24, 4h36 etc. (lembrando que o *headway* dessa viagem é de doze minutos)?

No caso em que a frequência de uma viagem é especificada no *frequencies.txt*, o cronograma (a tabela de horários) de uma viagem definido no *stop_times.txt* deve ser entendido como uma referência que descreve o tempo entre paradas. Isto é, os horários ali definidos não devem ser interpretados à risca. Por exemplo, o cronograma listado estabelece que o tempo de viagem entre a primeira e a segunda parada é de oito minutos, e o tempo entre a segunda e a terceira também. Ou seja, a viagem que parte da primeira parada às 4h chega na segunda às 4h08, e, na terceira, às 4h16. A próxima viagem, que parte da primeira parada às 4h12, por sua vez, chega na segunda parada às 4h20, e, na terceira, às 4h28.

Entretanto, poderíamos descrever as mesmas viagens no *stop_times.txt* sem fazer uso do arquivo *frequencies.txt*. Para isso, poderíamos adicionar um sufixo que identificasse cada uma das viagens referentes à linha CPTM L07 no sentido 0 ao longo do dia. A viagem (*trip_id*) com identificador CPTM L07-0_1, por exemplo, seria a primeira viagem no sentido 0 do dia e partiria da primeira parada às 4h e chegaria na segunda às 4h08. A viagem CPTM L07-0_2, por sua vez, seria a segunda viagem e partiria da primeira parada às 4h12 e chegaria na segunda às 4h20, e assim por diante. Cada uma dessas viagens deveria ser também adicionada ao arquivo *trips.txt* e a quaisquer outros que possuam a coluna *trip_id* como identificador.

Outro elemento que influencia na forma como o *frequencies.txt* afeta as tabelas de horários na tabela *stop_times.txt* é a coluna opcional *exact_times*. Um valor de 0 nesta coluna (ou quando ela está ausente do *feed*, como no caso do arquivo GTFS da SPTrans) indica que a viagem não necessariamente segue um cronograma fixo ao longo do período. Em vez disso, operadores tentam se ater a um determinado *headway* durante o período. Usando o mesmo exemplo de

uma viagem cujo *headway* é de doze minutos entre 4h e 5h, isso significa que não necessariamente a primeira partida sairá exatamente às 4h, a segunda às 4h12 e por aí em diante. A primeira pode, por exemplo, sair às 4h02. A segunda, às 4h14 ou 4h13 etc. Caso desejemos definir um cronograma que é seguido à risca, obtendo o mesmo resultado que seria obtido se definíssemos diversas viagens semelhantes partindo em diferentes horários no `stop_times.txt` (como mostrado no parágrafo anterior), devemos utilizar o valor 1 na coluna `exact_times`.

4.2 Onde encontrar dados GTFS de cidades brasileiras

Os dados de GTFS de diversas cidades do mundo podem ser baixados com o pacote de R `{tidytransit}` ou no *site* Transitland. No Brasil, diversas cidades usam dados GTFS no planejamento e operação de seus sistemas de transportes. Em muitos casos, no entanto, esses dados são de propriedade de empresas operadoras e concessionárias, e não do poder público. Infelizmente, esses arquivos raramente são disponibilizados abertamente e publicamente, contrariando boas práticas de gestão e compartilhamento de dados de interesse público. A tabela 9 mostra as fontes dos dados GTFS de algumas das poucas cidades do Brasil que disponibilizam seus *feeds* abertamente.¹⁴

TABELA 9
Fontes de dados GTFS publicamente disponíveis no Brasil

Cidade	Fonte	Informações
Belo Horizonte	Empresa de Transportes e Trânsito de Belo Horizonte (BHTrans).	Dado aberto: transporte convencional e transporte suplementar .
Fortaleza	Empresa de Transporte Urbano de Fortaleza (Etufor).	Dado aberto, disponível em: https://dados.fortaleza.ce.gov.br/dataset/gtfs .
Fortaleza	Metrô de Fortaleza (Metrofor).	Dado aberto, disponível em: https://www.metrofor.ce.gov.br/gtfs/ .
Porto Alegre	Empresa Pública de Transporte e Circulação de Porto Alegre (EPTC).	Dado aberto, disponível em: https://dados.portoalegre.rs.gov.br/dataset/gtfs .
Rio de Janeiro	Secretaria Municipal de Transportes (SMTR).	Dado aberto, disponível em: https://www.data.rio/datasets/gtfs-do-rio-de-janeiro/about .
São Paulo	Empresa Metropolitana de Transportes Urbanos de São Paulo (EMTU).	<i>Download</i> disponível em: https://www.emtu.sp.gov.br/emtu/dados-abertos/dados-abertos-principal/gtfs.fss . Necessário cadastro.
São Paulo	SPTans.	<i>Download</i> disponível em: https://www.sptrans.com.br/desenvolvedores/perfil-desenvolvedor/ . Necessário cadastro.

Elaboração dos autores.

Obs.: Os dados de GTFS disponibilizados pela SMTR não incluem os dados dos sistemas de trem e de metrô.

14. Levantamento realizado durante a elaboração deste livro.