

Título do capítulo	CAPÍTULO 6 COMPARANDO A ACESSIBILIDADE ENTRE DOIS CENÁRIOS DE TRANSPORTE
Autor(es)	Rafael H. M. Pereira Daniel Herszenhut
DOI	DOI: http://dx.doi.org/10.38116/9786556350547cap6

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.

6 COMPARANDO A ACESSIBILIDADE ENTRE DOIS CENÁRIOS DE TRANSPORTE

Neste capítulo, vamos ilustrar como usar de forma conjunta o material ensinado nos capítulos anteriores para medir o impacto de um projeto de infraestrutura de transporte sobre a acessibilidade urbana. Para avaliar o impacto de projetos de transporte, precisaremos comparar os níveis de acessibilidade antes e depois da implementação do projeto. Para isso, vamos:

- utilizar diferentes conjuntos de arquivos GTFS e fazer as edições necessárias para representar os cenários antes e depois da implementação do projeto;
- calcular duas matrizes de tempo de viagem, uma antes e outra depois do investimento;
- calcular os níveis de acessibilidade, tanto antes quanto depois do investimento; e
- comparar as condições de acessibilidade em cada cenário, examinando como os impactos se distribuem espacialmente e entre populações de diferentes níveis socioeconômicos.

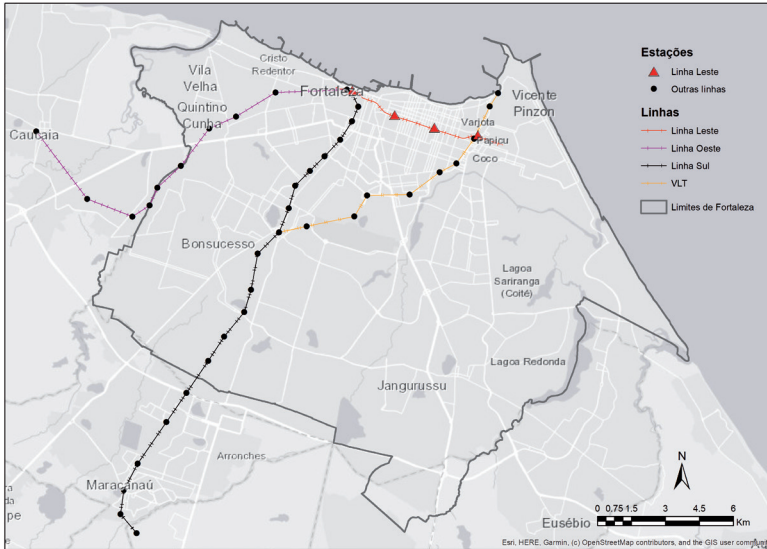
Vamos cobrir esse passo a passo em detalhes, começando por uma breve apresentação do nosso estudo de caso.

6.1 Apresentação do estudo de caso

Como estudo de caso, vamos fazer uma avaliação do projeto de construção da linha Leste do metrô de Fortaleza (figura 10). O município de Fortaleza é a capital do estado do Ceará, localizado na região Nordeste do Brasil. Com uma população estimada de 2,7 milhões habitantes, Fortaleza é a quinta cidade mais populosa do país.

Um dos grandes investimentos recentes no sistema de transporte de Fortaleza é a linha Leste do metrô. O traçado da linha Leste possui 7,3 km de extensão e liga o centro da cidade ao bairro Papicu, permitindo a integração das linhas de metrô Sul e Oeste com corredores de veículo leve sobre trilhos (VLT) e com o terminal de ônibus no Papicu (figura 11). Como a linha Leste ainda se encontrava em construção durante a elaboração deste livro, a análise feita neste capítulo trata de uma avaliação *ex ante*, ou seja, em que avaliamos o futuro impacto de um projeto sobre a acessibilidade urbana. Esse tipo de análise se contrapõe a avaliações *ex post*, usadas para avaliar o impacto causado por projetos já implementados.

FIGURA 10

Sistema de transporte urbano de média e alta capacidade de Fortaleza

Fonte: Braga *et al.* (2022).

Obs.: Figura reproduzida em baixa resolução e cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

FIGURA 11

Detalhamento da futura linha Leste do metrô de Fortaleza

Fonte: Braga *et al.* (2022).

Obs.: Figura reproduzida em baixa resolução e cujos leiaute e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

BOX 4

O cenário de intervenção analisado

É importante notar que a implementação desse projeto também será acompanhada de mudanças nas frequências das linhas Sul e Oeste do metrô e no corredor de VLT Parangaba-Mucuripe, além do racionamento do sistema de ônibus municipais, conforme o Plano de Acessibilidade Sustentável de Fortaleza (Pasfor).¹ Para fins didáticos, no entanto, não incorporamos as mudanças relativas ao racionamento do sistema de ônibus aos cenários da nossa análise. Portanto, o estudo de caso que apresentamos aqui representa um cenário simplificado, que considera apenas a implementação da linha Leste e as mudanças nas frequências das demais linhas do metrô e do VLT.²

Elaboração dos autores.

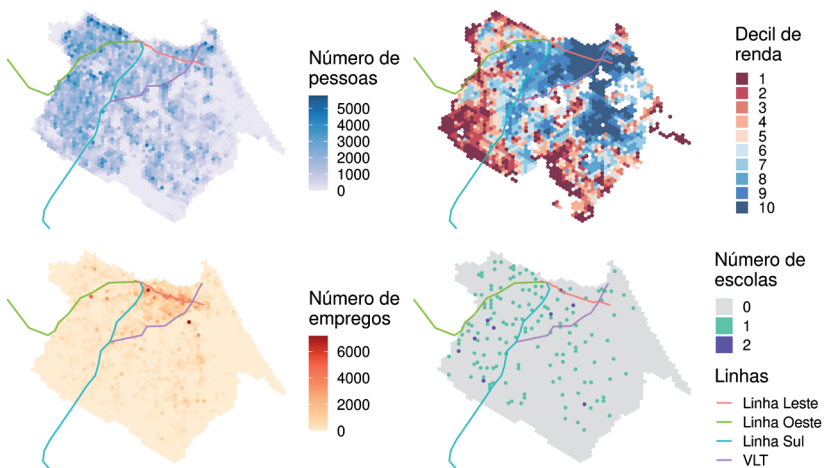
Notas: ¹ Disponível em: <<https://www.pasfor.com.br/>>.

² Uma avaliação mais completa, que leva em consideração todas as mudanças previstas no Pasfor, pode ser encontrada em Braga *et al.* (2022).

Como mostra a figura 12, a população de Fortaleza está majoritariamente distribuída na região central e na porção oeste da cidade, embora haja alguns pontos de alta densidade populacional no sudeste da cidade. Via de regra, a população de mais alta renda (em tons de azul no mapa da distribuição de decis de renda) está localizada na região do centro expandido e no sudeste do município, enquanto as populações de menor renda (em tons de vermelho) estão principalmente localizadas nas regiões periféricas ao sul e a oeste. Os empregos formais na cidade se distribuem principalmente ao longo de grandes avenidas, com maior concentração no centro da cidade. Em contraste, as escolas públicas de nível médio têm uma distribuição espacial mais equilibrada por todo o território.

FIGURA 12

Distribuição sociodemográfica, de oportunidades de emprego e de educação e de corredores de transporte público em Fortaleza



Elaboração dos autores.

Obs.: Figura cujos lajeautes e textos não puderam ser padronizados e revisados em virtude das condições técnicas dos originais (nota do Editorial).

6.2 Dados GTFS usados na análise

Nesta análise, usaremos os arquivos GTFS disponibilizados pela Etufor e pelo Metrofor. Esses dados descrevem o sistema de transporte público operante na cidade de Fortaleza em outubro de 2019. Para ter acesso a esses dados, usamos o código a seguir, que baixa os *feeds* usando o pacote {httr}:

```
# cria o endereço de arquivos temporários
end_metrofor <- tempfile("metrofor", fileext = ".zip")
end_etufor <- tempfile("etufor", fileext = ".zip")

# baixa dados da metrofor
httr::GET(
  "https://github.com/ipeaGIT/intro_access_book/releases/
  download/data_1st_edition/gtfs_for_metrofor_2021-01.zip",
  httr::write_disk(end_metrofor)
)

# baixa dados da etufor
httr::GET(
  "https://github.com/ipeaGIT/intro_access_book/releases/
  download/data_1st_edition/gtfs_for_etufor_2019-10.zip",
  httr::write_disk(end_etufor)
)
```

Para simularmos a implementação da nova linha Leste do metrô, precisamos também de um *feed* que descreva como deverá ser a sua operação. Esse *feed* deve conter algumas informações-chave, como o traçado da linha, a localização de suas estações, o tempo de viagem entre elas e a frequência planejada dos serviços. Nesse exemplo, vamos utilizar o arquivo GTFS do projeto da linha Leste criado anteriormente por Braga *et al.* (2022). Assim como os *feeds* da Etufor e do Metrofor, este arquivo está disponível para *download* no repositório do livro no GitHub e pode ser baixado com o código a seguir:

```
# cria o endereço do arquivo temporário
end_linha_leste <- tempfile("linha_leste", fileext = ".zip")

# baixa o GTFS da Linha Leste
httr::GET(
  "https://github.com/ipeaGIT/intro_access_book/releases/
  download/data_1st_edition/gtfs_linha_leste.zip",
  httr::write_disk(end_linha_leste)
)
```

Os *feeds* da Etufor e do Metrofor não refletem as mudanças do sistema de transporte público previstas no Pasfor. Para levá-las em consideração no cenário pós-implementação, portanto, precisamos editar esses arquivos usando o pacote `{gtfstools}`.

Em nosso cenário simplificado, vamos considerar as mudanças de frequência no metrô e no VLT relatadas em Braga *et al.* (2022), com base no Pasfor: i) aumento da frequência da linha Sul do metrô de quatro para dez viagens por hora; ii) aumento da frequência da linha Oeste do metrô de duas para cinco viagens por hora; e iii) aumento da frequência do VLT Parangaba-Mucuripe de duas para oito viagens por hora. Como estamos considerando apenas mudanças em linhas de metrô e VLT, vamos nos ater a edições no *feed* do Metrofor. Em primeiro lugar, precisamos ler esse arquivo com a função `read_gtfs()` e entender como as linhas estão descritas. Para isso, vamos olhar como as tabelas `routes`, `trips` e `calendar` estão estruturadas.

```
library(gtfstools)

gtfs_metrofor <- read_gtfs(end_metrofor)

gtfs_metrofor$routes[, .(route_id, route_long_name)]

  route_id      route_long_name
1:         8 VLT Parangaba Papicu
2:          6          Linha Sul
3:          7          Linha Oeste

gtfs_metrofor$trips[, .N, by = .(route_id, direction_id,
  service_id)]

  route_id direction_id service_id  N
1:         7             0         4 15
2:         7             1         4 15
3:         6             0         4 63
4:         6             1         4 64
5:         8             0         4 29
6:         8             1         4 29

gtfs_metrofor$calendar

  service_id monday tuesday wednesday thursday friday saturday sunday
1:          4       1       1         1         1       1         1       0
  start_date   end_date
1: 2020-01-01 2021-12-31
```

O *feed* descreve três linhas distintas: os dois corredores de metrô e o corredor de VLT. Como o *feed* não contém uma tabela *frequencies*, cada rota é descrita por diversas viagens, que partem em diferentes horários do dia. Essas viagens são divididas em viagens de ida e de volta e estão todas associadas a um mesmo serviço, que opera em dias úteis e no sábado.

A estratégia que vamos usar para fazer as mudanças necessárias no *feed* é composta pelas etapas descritas a seguir.

- 1) Filtrar o *feed* de forma a manter apenas uma viagem por sentido para cada linha, que servirá para nos dizer o tempo que cada viagem leva entre as suas paradas.
- 2) Adicionar uma tabela *frequencies* ao objeto GTFS, em que descreveremos a frequência de cada viagem.
- 3) “Converter” os registros da recém-adicionada tabela *frequencies* para cronogramas descritos na tabela *stop_times*. Essa conversão servirá para manter a característica do *feed* original, que descreve as viagens usando apenas a *stop_times*.

Para manter apenas uma viagem por sentido para cada linha, precisamos filtrar o *feed* usando a `filter_by_trip_id()`. Vamos usar a função para manter apenas o primeiro registro de viagem por linha e por sentido.

```
# identifica a linha na tabela trips em que os primeiros
registros por rota e
# por sentido estão localizados
indice <- gtfs_metrofor$trips[, .I[1], by = .(route_id,
direction_id)]$V1

# seleciona o identificador de cada linha acima
viagens_selecionadas <- gtfs_metrofor$trips[indice]$trip_id

# filtra o gtfs para manter apenas as viagens acima
gtfs_filtrado <- filter_by_trip_id(
  gtfs_metrofor,
  trip_id = viagens_selecionadas
)

gtfs_filtrado$trips
```

	trip_id	trip_headsign	direction_id	block_id	shape_id	service_id	route_id
1:	4	Caucaia	0			4	7
2:	19	Moura Brasil	1			4	7

3:	34	Carlito Benevides	0	4	6
4:	96	Chico da Silva	1	4	6
5:	159	Iate	0	4	8
6:	181	Parangaba	1	4	8

A fim de facilitar nossa edição, vamos mudar o id das viagens, identificando o corredor e o sentido em que elas operam. Essa mudança precisa ser feita tanto na tabela `trips` quanto na `stop_times`.

```
gtfs_filtrado$stop_times[
  ,
  trip_id := data.table::fcase(
    trip_id == "4", "metro_oeste_0",
    trip_id == "19", "metro_oeste_1",
    trip_id == "34", "metro_sul_0",
    trip_id == "96", "metro_sul_1",
    trip_id == "159", "vlt_0",
    trip_id == "181", "vlt_1"
  )
]

gtfs_filtrado$trips[
  ,
  trip_id := data.table::fcase(
    trip_id == "4", "metro_oeste_0",
    trip_id == "19", "metro_oeste_1",
    trip_id == "34", "metro_sul_0",
    trip_id == "96", "metro_sul_1",
    trip_id == "159", "vlt_0",
    trip_id == "181", "vlt_1"
  )
]

gtfs_filtrado$trips
```

	trip_id	trip_headsign	direction_id	block_id	shape_id	service_id
1:	metro_oeste_0	Caucaia	0			4
2:	metro_oeste_1	Moura Brasil	1			4
3:	metro_sul_0	Carlito Benevides	0			4
4:	metro_sul_1	Chico da Silva	1			4
5:	vlt_0	Iate	0			4
6:	vlt_1	Parangaba	1			4


```

route_id
1:      7
2:      7
3:      6
4:      6
5:      8
6:      8

```

Precisamos agora adicionar uma tabela `frequencies` que descreva a frequência de cada viagem. A especificação requer, no entanto, que listemos o *headway* de cada linha, e não a sua frequência. Como o *headway* é o inverso da frequência, precisamos dividir o intervalo de uma hora (3.600 segundos) pela frequência de cada linha (10 viagens/hora para a linha Sul, 5 viagens/hora para a linha Oeste e 8 viagens/hora para o VLT). Como resultado, temos que os *headways* da linha Sul, da linha Oeste e do VLT passarão a ser, respectivamente, 360, 720 e 450 segundos. Com o código a seguir, criamos uma tabela `frequencies` com os pacotes `{tibble}` e `{data.table}`.

```

frequencies <- tibble::tribble(
  ~trip_id,      ~start_time, ~end_time, ~headway_secs, ~exact_times,
  "metro_oeste_0", "06:00:00", "09:00:00", 720L,      1,
  "metro_oeste_1", "06:00:00", "09:00:00", 720L,      1,
  "metro_sul_0",   "06:00:00", "09:00:00", 360L,      1,
  "metro_sul_1",   "06:00:00", "09:00:00", 360L,      1,
  "vlt_0",         "06:00:00", "09:00:00", 450L,      1,
  "vlt_1",         "06:00:00", "09:00:00", 450L,      1
)

# converte tabela para data.table
data.table::setDT(frequencies)

# cria tabela frequencies no objeto gtfs com base na tabela acima
gtfs_filtrado$frequencies <- frequencies

```

A fim de simplificar este estudo de caso, assumimos que os *headways* listados são válidos no período de 6h às 9h. Essa premissa é válida no nosso caso, pois iremos calcular as matrizes de tempo de viagem apenas durante o pico da manhã. No entanto, caso desejássemos calcular os tempos de viagens em outros períodos do dia, ou mesmo utilizar este arquivo GTFS na operação do dia a dia desses corredores, precisaríamos listar os *headways* para os demais períodos do dia, como fora do pico, madrugada etc. O valor 1 da coluna `exact_times` estabelece que o cronograma das viagens durante o período especificado deve seguir o *headway* à risca, e não de forma aproximada.¹⁹

19. Para mais detalhes, conferir a descrição da tabela `frequencies` no capítulo 4.

O objeto GTFS que resulta das modificações feitas até aqui já pode perfeitamente ser utilizado para o cálculo de matrizes de tempo de viagem. Buscando retomar a característica inicial do *feed* de não possuir uma tabela *frequencies*, no entanto, vamos “converter” os registros dessa tabela em cronogramas descritos na tabela *stop_times*. Para isso, usamos a função `frequencies_to_stop_times()`. Como todas as viagens do *feed* são convertidas, a tabela *frequencies* é removida do objeto.

```
gtfs_filtrado <- frequencies_to_stop_times(gtfs_filtrado)

gtfs_filtrado$frequencies
```

```
NULL
```

Para verificar se a operação deu certo, vamos olhar para as viagens da linha Oeste no sentido Caucaia (cujo *direction_id* é 0). Como essa linha deve ter uma frequência de cinco viagens por hora entre 6h e 9h, a tabela *trips* deve conter exatamente dezesseis viagens relacionadas a ela (cinco viagens por hora durante três horas, mais uma viagem começando às 9h).

```
# seleciona apenas as viagens da linha oeste do metrô
metro_linha_oeste <- gtfs_filtrado$trips[grepl("metro_oeste_0",
trip_id)]

# checa número de viagens
nrow(metro_linha_oeste)
```

```
[1] 16
```

```
# checa identificador de cada viagem
metro_linha_oeste$trip_id
```

```
[1] "metro_oeste_0_1" "metro_oeste_0_2" "metro_oeste_0_3" "metro_oeste_0_4"
[5] "metro_oeste_0_5" "metro_oeste_0_6" "metro_oeste_0_7" "metro_oeste_0_8"
[9] "metro_oeste_0_9" "metro_oeste_0_10" "metro_oeste_0_11" "metro_oeste_0_12"
[13] "metro_oeste_0_13" "metro_oeste_0_14" "metro_oeste_0_15" "metro_oeste_0_16"
```

A tabela *stop_times*, por sua vez, deve listar essas viagens partindo a cada doze minutos (o equivalente a um *headway* de 450 segundos). Vamos verificar, portanto, o primeiro registro do cronograma de cada uma das viagens listadas anteriormente.

```

viagens_metro_oeste <- metro_linha_oeste$trip_id

# identifica a linha na tabela stop_times em que estão os
# primeiros registros de
# cada uma das viagens acima
indice_viagens <- gtfs_filtrado$stop_times[
  trip_id %in% viagens_metro_oeste,
  .I[1],
  by = trip_id
]$V1

gtfs_filtrado$stop_times[indice_viagens, .(trip_id, departure_time)]

```

	trip_id	departure_time
1:	metro_oeste_0_1	06:00:00
2:	metro_oeste_0_2	06:12:00
3:	metro_oeste_0_3	06:24:00
4:	metro_oeste_0_4	06:36:00
5:	metro_oeste_0_5	06:48:00
6:	metro_oeste_0_6	07:00:00
7:	metro_oeste_0_7	07:12:00
8:	metro_oeste_0_8	07:24:00
9:	metro_oeste_0_9	07:36:00
10:	metro_oeste_0_10	07:48:00
11:	metro_oeste_0_11	08:00:00
12:	metro_oeste_0_12	08:12:00
13:	metro_oeste_0_13	08:24:00
14:	metro_oeste_0_14	08:36:00
15:	metro_oeste_0_15	08:48:00
16:	metro_oeste_0_16	09:00:00

A operação de “conversão” da tabela frequências para a `stop_times`, portanto, funcionou corretamente, e podemos utilizar o nosso *feed* modificado no cálculo da matriz de tempo de viagem. Para isso, no entanto, precisamos salvar o objeto GTFS na memória em formato `.zip`, assim como estão salvos os demais dados GTFS que serão utilizados no estudo de caso. Para isso, usamos a função `write_gtfs()`.

```

end_metrofor_modificado <- tempfile("metrofor_modificado",
fileext = ".zip")

write_gtfs(gtfs_filtrado, end_metrofor_modificado)

```

Temos, agora, quatro arquivos de GTFS distintos:

- o *feed* da Etufor, que descreve o sistema de ônibus que operava em outubro de 2019;
- o *feed* do Metrofor, que descreve a operação em outubro de 2019 das linhas Sul e Oeste do metrô e do VLT Parangaba-Mucuripe;
- o *feed modificado* do Metrofor, que descreve a futura operação das linhas Sul e Oeste do metrô e do VLT Parangaba-Mucuripe, como previsto no Pasfor; e
- o *feed* da linha Leste, que descreve a futura operação dessa linha.

Esses quatro arquivos GTFS serão usados em conjunto para calcular as condições de acessibilidade de Fortaleza antes e depois da implementação da linha Leste. No cenário pré-linha Leste, vamos calcular as matrizes de tempo de viagem com base apenas nos *feeds* de outubro de 2019 do Metrofor e da Etufor, que representam a típica operação de transporte público da cidade antes da implementação do novo corredor. No cenário pós-implementação, consideraremos o *feed* da Etufor, o *feed modificado* do Metrofor, com as frequências do metrô e do VLT alteradas, e o arquivo GTFS da linha Leste, incorporando à análise a operação planejada dessa linha após sua finalização.

6.3 Cálculo das matrizes de tempo de viagem

Feitas as edições necessárias ao *feed* do Metrofor e definidos os dados GTFS que vamos usar em cada um dos cenários de transporte público, o próximo passo é calcular as matrizes de tempo de viagem, que posteriormente serão utilizadas para estimar os níveis de acessibilidade. Para isso, vamos utilizar a função `travel_time_matrix()` do pacote `{r5r}`, como apresentado anteriormente no capítulo 3.

Antes de calcular as matrizes, contudo, precisamos organizar os nossos arquivos na estrutura que o `{r5r}` requer. Com o código a seguir, criamos uma pasta separada para cada um dos nossos cenários (antes e depois) e salvamos nelas os dados necessários para o roteamento:

```
# cria pasta raiz da análise de dados
pasta_analise <- "analise_de_impacto"
dir.create(pasta_analise)

# cria pasta dos cenários
pasta_antes <- file.path(pasta_analise, "antes")
pasta_depois <- file.path(pasta_analise, "depois")
```

```

dir.create(pasta_antes)
dir.create(pasta_depois)

# copia os arquivos para pasta do cenário "antes"
file.copy(from = end_etufor, to = file.path(pasta_antes,
"etufor.zip"))
file.copy(from = end_metrofor, to = file.path(pasta_antes,
"metrofor.zip"))

# copia os arquivos para pasta do cenário "depois"
file.copy(from = end_etufor, to = file.path(pasta_depois,
"etufor.zip"))
file.copy(
  from = end_metrofor_modificado,
  to = file.path(pasta_depois, "metrofor_modificado.zip")
)
file.copy(
  from = end_linha_leste,
  to = file.path(pasta_depois, "linha_leste.zip")
)

# visualiza esquema de arquivos na pasta
fs::dir_tree(pasta_analise)

```

```

analise_de_impacto
+-- antes
|   +-- etufor.zip
|   \-- metrofor.zip
\-- depois
    +-- etufor.zip
    +-- linha_leste.zip
    \-- metrofor_modificado.zip

```

Para estimarmos o tempo de viagem na nossa área de estudo, precisamos ainda de um arquivo com os dados do OSM representando a rede viária local, em formato `.pbfl`. Opcionalmente, iremos utilizar também um arquivo representando a topografia local, em formato `.tif`. Esses arquivos, assim como os dados GTFS, estão disponíveis para *download* no repositório do livro. Partindo do pressuposto de que a implementação da linha Leste não vai afetar o traçado das ruas e calçadas na região, bem como a topografia local, podemos usar os mesmos arquivos para o cálculo das duas matrizes de tempo de viagem. Com o código

apresentado a seguir, baixamos esses dados e copiamos os arquivos para as pastas dos dois cenários de transporte:

```
# cria endereço temporário dos arquivos na máquina local
end_pbf <- tempfile("rede_viaria", fileext = ".osm.pbf")
end_tif <- tempfile("topografia", fileext = ".tif")

# download dos dados de OSM
httr::GET(
  "https://github.com/ipeaGIT/intro_access_book/releases/
  download/data_1st_edition/fortaleza.osm.pbf",
  httr::write_disk(end_pbf)
)

# download dos dados de topografia
httr::GET(
  "https://github.com/ipeaGIT/intro_access_book/releases/
  download/data_1st_edition/topografia3_for.tif",
  httr::write_disk(end_tif)
)

# copia arquivo para as pastas dos cenários antes e depois
file.copy(from = end_pbf, to = file.path(pasta_antes,
"rede_viaria.osm.pbf"))
file.copy(from = end_pbf, to = file.path(pasta_depois,
"rede_viaria.osm.pbf"))

file.copy(from = end_tif, to = file.path(pasta_antes,
"topografia.tif"))
file.copy(from = end_tif, to = file.path(pasta_depois,
"topografia.tif"))
```

```
fs::dir_tree(pasta_analise)
```

```
analise_de_impacto
+-- antes
|   +-- etufor.zip
|   +-- metrofor.zip
|   +-- rede_viaria.osm.pbf
|   \-- topografia.tif
```

```

\-- depois
  +-- etufor.zip
  +-- linha_leste.zip
  +-- metrofor_modificado.zip
  +-- rede_viaria.osm.pbf
  \-- topografia.tif

```

Com os dados organizados nas pastas, podemos começar o cálculo das matrizes de tempo de viagem. A primeira etapa é construir a rede de transporte multimodal usada pelo {r5r} no roteamento a partir dos dados da rede viária, do sistema de transporte público e de topografia. Isso é feito com o comando `setup_r5()`, que também retorna uma conexão com o R5. Com o código a seguir, criamos duas redes, uma para cada cenário:

```

# aloca a memória disponível para o Java
options(java.parameters = "-Xmx4G")

# carrega a biblioteca
library(r5r)

# cria a rede de transporte multimodal de cada cenário
con_r5r_antes <- setup_r5(pasta_antes, verbose = FALSE)
con_r5r_depois <- setup_r5(pasta_depois, verbose = FALSE)

```

Criadas as redes de transporte de cada cenário, prosseguimos para o cálculo das matrizes de tempo de viagem. Nesta etapa, vamos utilizar como origens e destinos os centroides de uma grade espacial de hexágonos de Fortaleza, disponibilizada pelo pacote {aopdata}.²⁰ Cada hexágono tem uma área de 0,11 km², aproximadamente um quarteirão, o que permite uma análise espacial bem detalhada.

Para comparar adequadamente os dois cenários, precisamos calcular as duas matrizes considerando os mesmos parâmetros de viagem. Aqui, vamos considerar viagens a pé ou por transporte público, permitir distâncias de caminhada de até trinta minutos nas pernas de acesso e egresso das paradas de transporte público e limitar o tempo máximo de viagem em até sessenta minutos. Vamos considerar o horário de partida de 7h, durante o horário de pico de uma típica segunda-feira de operação:

20. Mais detalhes sobre o pacote são apresentados na seção 5.

```
# baixa os dados da grade espacial
grade_fortaleza <- aopdata::read_grid(city = "Fortaleza")

# calcula os centroides das células da grade
pontos <- sf::st_centroid(grade_fortaleza)

# renomeia o nome da coluna com o id das células
names(pontos)[1] <- "id"

# calcula a matriz de tempo de viagem do cenário "antes"
matriz_antes <- travel_time_matrix(
  con_r5r_antes,
  origins = pontos,
  destinations = pontos,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = as.POSIXct(
    "02-03-2020 07:00:00",
    format = "%d-%m-%Y %H:%M:%S"
  ),
  max_walk_time = 30,
  max_trip_duration = 60,
  verbose = FALSE,
  progress = FALSE
)

# calcula a matriz de tempo de viagem do cenário "depois"
matriz_depois <- travel_time_matrix(
  con_r5r_depois,
  origins = pontos,
  destinations = pontos,
  mode = c("WALK", "TRANSIT"),
  departure_datetime = as.POSIXct(
    "02-03-2020 07:00:00",
    format = "%d-%m-%Y %H:%M:%S"
  ),
  max_walk_time = 30,
  max_trip_duration = 60,
  verbose = FALSE,
  progress = FALSE
)

head(matriz_antes)
```



```

      from_id          to_id travel_time_p50
1: 89801040323ffff 89801040323ffff          2
2: 89801040323ffff 89801040327ffff         22
3: 89801040323ffff 8980104032bffff         32
4: 89801040323ffff 8980104032ffff         15
5: 89801040323ffff 89801040333ffff         10
6: 89801040323ffff 89801040337ffff         19

```

```
head(matriz_depois)
```

```

      from_id          to_id travel_time_p50
1: 89801040323ffff 89801040323ffff          2
2: 89801040323ffff 89801040327ffff         22
3: 89801040323ffff 8980104032bffff         32
4: 89801040323ffff 8980104032ffff         15
5: 89801040323ffff 89801040333ffff         10
6: 89801040323ffff 89801040337ffff         19

```

À primeira vista, nossas matrizes parecem iguais: todos os tempos de viagem na amostra de pares mostrados anteriormente são idênticos. Isso ocorre porque o projeto de expansão do metrô fica restrito a uma área relativamente pequena, no centro da cidade de Fortaleza, e as mudanças nas frequências do VLT e das linhas Sul e Oeste do metrô afetam principalmente as imediações desses corredores. Assim, muitos deslocamentos entre regiões da cidade de fato não são afetados pela implementação do corredor e das mudanças de frequência. Diversos pares origem-destino, no entanto, têm os tempos de viagem entre eles impactados:

```

# une os tempos de viagem dos dois cenários
comparacao <- merge(
  matriz_antes,
  matriz_depois,
  by = c("from_id", "to_id"),
  suffixes = c("_antes", "_depois")
)

# mostra apenas os pares OD cujo tempo que os distancia diminuiu
comparacao[travel_time_p50_antes < travel_time_p50_depois]

```

```

      from_id          to_id travel_time_p50_antes
1: 8980104092fffff 8980104531bffff          48
2: 8980104092fffff 8980104536bffff          47
3: 8980104092fffff 898010453c7ffff          44

```

```

4: 8980104092ffffff 8980104e803ffff 48
5: 8980104092ffffff 8980104e807ffff 57
---
10499: 8980104eecffffff 8980104e80bffff 53
10500: 8980104eecffffff 8980104e80ffffff 55
10501: 8980104eecffffff 8980104e863ffff 56
10502: 8980104eecffffff 8980104e87bffff 51
10503: 8980104f1abffff 8980104e87bffff 58
travel_time_p50_depois
1: 50
2: 48
3: 46
4: 50
5: 59
---
10499: 54
10500: 57
10501: 57
10502: 53
10503: 59

```

6.4 Cálculo da acessibilidade nos cenários antes e depois

O cálculo dos níveis de acessibilidade nos dois cenários é muito simples, exigindo apenas um processamento básico dos dados e a aplicação de uma das funções de cálculo de acessibilidade do pacote `{accessibility}`. Para facilitar o tratamento dos dados, vamos empilhar as matrizes de tempo de viagem dos dois cenários em uma única tabela, identificando cada cenário com a coluna `cenario`:

```

# combina as matrizes de tempo de viagem dos cenários antes e depois
matriz <- rbind(matriz_antes, matriz_depois, idcol = "cenario")
matriz[, cenario := factor(cenario, labels = c("antes", "depois"))]

matriz

```

```

cenario from_id to_id travel_time_p50
1: antes 89801040323ffff 89801040323ffff 2
2: antes 89801040323ffff 89801040327ffff 22
3: antes 89801040323ffff 8980104032bffff 32
4: antes 89801040323ffff 8980104032ffffff 15
5: antes 89801040323ffff 89801040333ffff 10
---

```

3773154:	depois	8980107b6dbffff	8980107b6cbffff	8
3773155:	depois	8980107b6dbffff	8980107b6cfffff	15
3773156:	depois	8980107b6dbffff	8980107b6d3ffff	9
3773157:	depois	8980107b6dbffff	8980107b6d7ffff	16
3773158:	depois	8980107b6dbffff	8980107b6dbffff	0

Para calcular os níveis de acessibilidade, precisamos de uma tabela com os dados de uso do solo da cidade de Fortaleza. Podemos baixar uma tabela com esses dados usando a função `read_landuse()` do pacote `{aopdata}`, que traz dados de contagem populacional e de oportunidades em cada um dos hexágonos que compõem a grade espacial baixada anteriormente.

```
# baixa dados de uso do solo em fortaleza
dados_fortaleza <- aopdata::read_landuse(
  city = "Fortaleza",
  showProgress = FALSE
)
```

A fim de demonstração, vamos calcular a acessibilidade a postos de trabalho e a escolas públicas de ensino médio na nossa área de estudo. Os dados do total de empregos e de escolas públicas de nível médio em cada hexágono estão listados nas colunas `T001` e `E004`, respectivamente. Vamos renomeá-las para facilitar sua identificação e manter apenas as colunas necessárias na tabela de dados de uso do solo. Vamos manter também as colunas `P001`, de população total em cada hexágono, e `R003`, do decil de renda em que cada hexágono se encontra, que serão úteis mais à frente:

```
# renomeia colunas
colunas_mantidas <- c("id", "empregos", "escolas", "populacao",
"decil")
data.table::setnames(
  dados_fortaleza,
  old = c("id_hex", "T001", "E004", "P001", "R003"),
  new = colunas_mantidas
)

# deleta as demais colunas que não serão usadas
dados_fortaleza[, setdiff(names(dados_fortaleza),
colunas_mantidas) := NULL]

dados_fortaleza
```

	id populacao	decil	empregos	escolas
1:	89801040323ffff	30	1	0
2:	89801040327ffff	318	1	7
3:	8980104032bffff	0	NA	0
4:	8980104032fffff	103	1	98
5:	89801040333ffff	43	1	0

2558:	8980107b6cbffff	2575	4	124
2559:	8980107b6cfffff	2997	3	4
2560:	8980107b6d3ffff	1751	8	14
2561:	8980107b6d7ffff	2032	4	134
2562:	8980107b6dbffff	1896	9	193

Uma decisão-chave no cálculo de acessibilidade é a escolha da medida a ser utilizada. É extremamente importante pesar as vantagens e desvantagens de cada medida e compreender quais indicadores se adequam às oportunidades para as quais desejamos calcular os níveis de acessibilidade. Nesse exemplo, utilizaremos duas medidas distintas.

- 1) No cálculo da acessibilidade a empregos, vamos usar a medida de oportunidades cumulativas. Essa métrica nos permite entender quantos empregos são acessíveis dentro de um determinado custo de viagem. Apesar das suas limitações, discutidas no capítulo 2, esse é um dos indicadores mais amplamente utilizados. Isso se deve, em grande medida, à facilidade de comunicar e interpretar seus resultados. Nesse exemplo, vamos estabelecer como limite de custo um tempo de viagem de sessenta minutos, valor muito próximo do tempo médio de deslocamento casa-trabalho por transporte público de Fortaleza em 2019 (cerca de 58 minutos, segundo o Pasfor).
- 2) No cálculo da acessibilidade a escolas públicas, vamos usar a medida de custo mínimo de viagem. Essa métrica é especialmente útil para avaliar a cobertura de serviços públicos essenciais, como educação e saúde básica. Com ela, podemos, por exemplo, identificar as parcelas da população que estão a uma distância maior do que a considerada razoável para acessar esses serviços essenciais.

Como mostrado anteriormente, no capítulo 3, o cálculo dessas medidas pode ser feito com as funções `cumulative_cutoff()` e `cost_to_closest()`, respectivamente, do pacote `{accessibility}`:

```
# carrega a biblioteca
library(accessibility)

# calcula a medida de oportunidades cumulativas
acesso_a_empregos <- cumulative_cutoff(
  matriz,
  land_use_data = dados_fortaleza,
  opportunity = "empregos",
  travel_cost = "travel_time_p50",
  cutoff = 60,
  group_by = "cenario"
)

acesso_a_empregos
```

```
      id cenario empregos
1: 89801040323ffff antes   48049
2: 89801040327ffff antes   26044
3: 8980104032bffff antes   25862
4: 8980104032fffff antes   69361
5: 89801040333ffff antes   48049
---
5120: 8980107b6cbffff depois  378840
5121: 8980107b6cfffff depois  286878
5122: 8980107b6d3ffff depois  339878
5123: 8980107b6d7ffff depois  359648
5124: 8980107b6dbffff depois  372565
```

```
# calcula a medida de tempo mínimo de viagem
tempo_a_escolas <- cost_to_closest(
  matriz,
  land_use_data = dados_fortaleza,
  opportunity = "escolas",
  travel_cost = "travel_time_p50",
  group_by = "cenario"
)

tempo_a_escolas
```

```
      id cenario travel_time_p50
1: 89801040323ffff antes           36
2: 89801040323ffff depois          36
3: 89801040327ffff antes           41
```

4:	89801040327ffff	depois	41
5:	8980104032bffff	antes	41

5120:	8980107b6d3ffff	depois	19
5121:	8980107b6d7ffff	antes	14
5122:	8980107b6d7ffff	depois	14
5123:	8980107b6dbffff	antes	15
5124:	8980107b6dbffff	depois	15

O resultado da função de custo mínimo de viagem inclui alguns valores Inf (abreviação de infinito). Esse valor é utilizado para sinalizar origens que não conseguem alcançar nenhuma oportunidade, dadas as viagens que compõem a matriz. Em nosso caso, isso significa que as origens listadas com esse valor não conseguem alcançar nenhuma escola pública de nível médio dentro de sessenta minutos (limite de tempo de viagem imposto no cálculo da matriz). Para simplificar os cálculos daqui em diante, vamos considerar que essas regiões estão a oitenta minutos de viagem de uma escola:

```
# substituí valores Inf por tempos de viagem de 80 minutos
tempo_a_escolas[
  ,
  travel_time_p50 := ifelse(is.infinite(travel_time_p50),
  80, travel_time_p50)
]
```

Feito isso, podemos calcular a diferença dos níveis de acessibilidade entre os dois cenários. Esta informação é útil para comunicar de forma mais direta o efeito da implementação da linha Leste e das mudanças de frequência dos outros corredores sobre as condições de acessibilidade na cidade. Para isso, usamos o código adiante:

```
acesso_a_empregos[
  ,
  diferenca := data.table::shift(empregos, type = "lead") - empregos,
  by = id
]

tempo_a_escolas[
  ,
  diferenca := data.table::shift(travel_time_p50, type = "lead") -
  travel_time_p50,
  by = id
]
```

6.5 Análise dos níveis de acessibilidade antes e depois

Agora que calculamos os níveis de acessibilidade em cada cenário e a diferença entre eles, precisamos compreender como a futura implementação da linha Leste em conjunto com o aumento da frequência das linhas de metrô e VLT deverá impactar as condições de acessibilidade em nossa área de estudo. Como primeira análise exploratória, podemos investigar o impacto dessas mudanças sobre a acessibilidade média da cidade. Olhando primeiro para a acessibilidade a oportunidades de emprego, vamos calcular a quantidade média de empregos acessíveis em cada cenário. Aqui, é importante calcular a média de acessibilidade ponderada pela população de cada célula da grade espacial que estamos usando, visto que células com maiores populações contribuem mais para o nível médio da população como um todo do que células com poucas pessoas.

```
# carrega bibliotecas de visualização de dados
library(ggplot2)
library(patchwork)

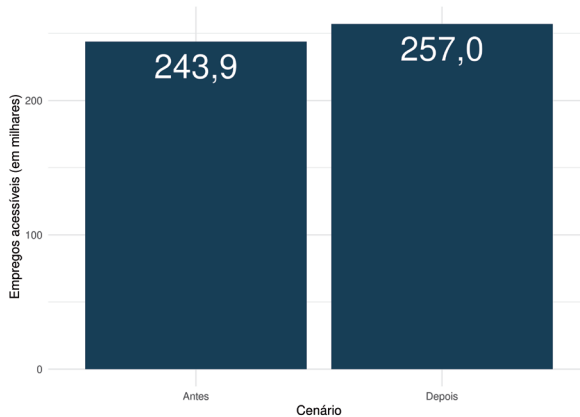
# une tabela de acessibilidade com informações de pessoas
residentes e renda em
# renda nos hexágonos
acesso_a_empregos <- merge(
  acesso_a_empregos,
  dados_fortaleza,
  by = "id"
)

# renomeia colunas com nomes ambíguos pós-união
data.table::setnames(
  acesso_a_empregos,
  old = c("empregos.x", "empregos.y"),
  new = c("acesso_a_empregos", "empregos_no_hexagono")
)

# calcula a média ponderada de acessibilidade em cada cenário
acesso_medio <- acesso_a_empregos[
  ,
  .(acesso = weighted.mean(acesso_a_empregos,
    w = as.numeric(populacao)),
  by = cenario
]
```

```
ggplot(data = acesso_medio, aes(x = cenario, y = acesso / 1000)) +  
  geom_col(fill = "#0f3c53") +  
  geom_text(  
    aes(  
      label = formatC(  
        acesso / 1000,  
        digits = 1,  
        decimal.mark = ",",  
        format = "f"  
      )  
    ),  
    vjust = 1.5,  
    color = "white",  
    size = 10  
  ) +  
  ylab("Empregos acessíveis (em milhares)") +  
  scale_x_discrete(name = "Cenário", labels = c("Antes",  
"Depois")) +  
  theme_minimal()
```

FIGURA 13
Média de acessibilidade ao emprego em Fortaleza por cenário de transporte



Fonte: Figura gerada pelo código supracitado.

Os resultados mostram que a população de Fortaleza conseguia acessar em média 243.859 empregos por transporte público em até sessenta minutos de viagem antes da expansão do metrô, em 2019. A implementação da linha Leste e os aumentos nas frequências do metrô e do VLT aumentam esse valor em 5,4%, para em média 257.035 empregos.

Analisando o tempo médio de acesso à escola pública de ensino médio mais próxima, notamos que as mudanças no sistema de transportes pouco alteram a acessibilidade a essas escolas. Em média, a população de Fortaleza demorava cerca de 12,5 minutos para chegar à escola pública de ensino médio mais próxima de sua casa em 2019. Após a conclusão da extensão do metrô e o aumento da frequência das linhas de metrô e VLT, esse resultado permanece praticamente inalterado.

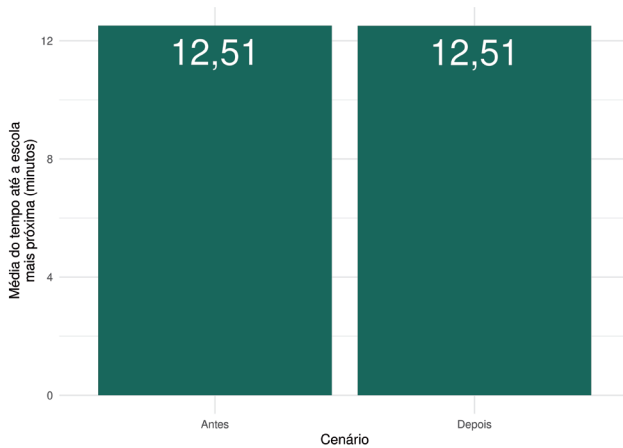
```
# une tabela de tempo a escolas com informações de pessoas
residentes nos
# hexágonos
tempo_a_escolas <- merge(
  tempo_a_escolas,
  dados_fortaleza,
  by = "id"
)

# calcula a média ponderada de acessibilidade em cada cenário
tempo_medio <- tempo_a_escolas[
  ,
  .(tempo = weighted.mean(travel_time_p50,
    w = as.numeric(populacao))),
  by = cenario
]

ggplot(data = tempo_medio, aes(x = cenario, y = tempo)) +
  geom_col(fill = "#0d6556") +
  geom_text(
    aes(label = formatC(tempo, digits = 2, decimal.mark = ",",
      format = "f")),
    vjust = 1.5,
    color = "white",
    size = 10
  ) +
  ylab("Média do tempo até a escola\nmais próxima (minutos)") +
  scale_x_discrete(name = "Cenário", labels = c("Antes",
    "Depois")) +
  theme_minimal()
```

FIGURA 14

Média do tempo até a escola pública de nível médio mais próxima em Fortaleza por cenário de transporte



Fonte: Figura gerada pelo código supracitado.

Em síntese, os resultados mostram que o plano de construção da linha Leste do metrô e de ajuste das frequências de outros corredores de transporte público da cidade deverá afetar mais significativamente a acessibilidade a oportunidades de emprego do que a acessibilidade a escolas públicas de nível médio. Isso se explica principalmente pela distribuição espacial desses dois tipos de oportunidade na cidade de Fortaleza: enquanto os empregos se distribuem de forma muito mais concentrada no centro da cidade, as escolas estão muito mais bem distribuídas em todo o território. As mudanças no sistema de transporte público, portanto, devem facilitar o acesso de moradores de regiões mais afastadas aos empregos localizados no centro. Em contrapartida, a distribuição mais equitativa de escolas públicas resulta em condições relativamente boas de acesso a esses locais mesmo antes das mudanças, explicando por que tais mudanças deverão ter impacto tão pequeno no tempo médio de acesso a escolas.

Esses resultados podem ser melhor compreendidos quando observamos a sua distribuição espacial. Antes de fazer isso, no entanto, vamos criar um objeto espacial que descreve o traçado dos corredores de transporte da cidade – o que torna ainda mais evidente o impacto das mudanças no sistema de transportes sobre os níveis de acessibilidade.

```
# lê arquivos gtfs necessários para gerar as geometrias de
cada corredor
gtfs_metrofor <- read_gtfs(end_metrofor)
gtfs_linha_leste <- read_gtfs(end_linha_leste)
```

```
# gtfs da metrofor não contém tabela shapes, logo vamos criar
a geometria
# a partir das tabelas stops e stop_times com a função
get_trip_geometry()
viagens_corredores <- c("4", "34", "159")

# a sequência de paradas de uma das viagens não está ordenada
corretamente,
# então precisamos ordená-las manualmente
gtfs_metrofor$stop_times <- gtfs_metrofor$stop_times[
  order(trip_id, stop_sequence)
]
traj_metrofor <- gtfstools::get_trip_geometry(
  gtfs_metrofor,
  trip_id = viagens_corredores
)

# converte o trajeto em um dos sentidos da linha leste em
geometria espacial
traj_linha_leste <- gtfstools::convert_shapes_to_sf(
  gtfs_linha_leste,
  shape_id = "LL_0"
)

# nomeia cada uma das linhas e une as tabelas
traj_linha_leste$corredor <- "Linha Leste"
traj_metrofor$corredor <- data.table::fcase(
  traj_metrofor$trip_id == 4, "Linha Oeste",
  traj_metrofor$trip_id == 34, "Linha Sul",
  traj_metrofor$trip_id == 159, "VLT"
)

traj_metrofor$origin_file <- NULL
traj_metrofor$trip_id <- NULL
traj_linha_leste$shape_id <- NULL

traj_corredores <- rbind(traj_metrofor, traj_linha_leste)

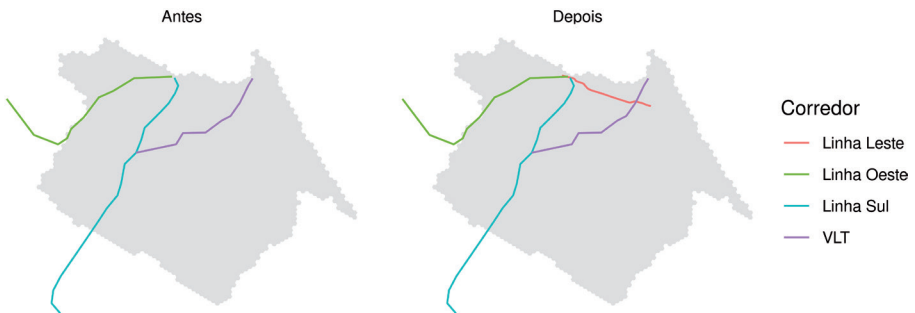
# duplica a tabela, adiciona coluna identificando cada
cenário e remove a linha
# leste do cenário pré-implementação das mudanças
traj_corredores <- rbind(traj_corredores, traj_corredores)
traj_corredores$cenario <- rep(c("antes", "depois"), each = 4)
```

```
traj_corredores <- subset(
  traj_corredores,
  corredor != "Linha Leste" | cenario != "antes"
)

ggplot() +
  geom_sf(data = grade_fortaleza, fill = "gray90", color = NA) +
  geom_sf(data = traj_corredores, aes(color = corredor)) +
  scale_color_manual(
    name = "Corredor",
    values = c("#F8766D", "#7CAE00", "#00BFC4", "#C77CFF")
  ) +
  facet_wrap(
    ~ cenario,
    nrow = 1,
    labeller = as_labeller(c(antes = "Antes", depois = "Depois"))
  ) +
  theme_void()
```

FIGURA 15

Distribuição dos corredores de transporte de média e alta capacidade em Fortaleza por cenário de transporte



Fonte: Figura gerada pelo código supracitado.

Podemos, então, analisar a distribuição espacial dos níveis de acessibilidade nos cenários antes e depois, bem como a diferença de acessibilidade entre eles. Para isso, temos que juntar as estimativas de acessibilidade com a grade espacial da nossa área de estudo. Primeiro, para a acessibilidade a empregos, como mostrado a seguir.

```
# une os dados de acessibilidade com a grade espacial de
# fortaleza e os converte
# em objetos espaciais
acesso_a_empregos <- merge(
  acesso_a_empregos,
  grade_fortaleza,
  by.x = "id",
  by.y = "id_hex"
)
acesso_a_empregos_sf <- sf::st_sf(acesso_a_empregos)

# configura mapas da distribuição de acessibilidade nos
# cenários antes e depois
dist_acesso <- ggplot() +
  geom_sf(
    data = acesso_a_empregos_sf,
    aes(fill = acesso_a_empregos),
    color = NA
  ) +
  facet_wrap(
    ~ cenario,
    nrow = 1,
    labeller = as_labeller(c(antes = "Antes", depois = "Depois"))
  ) +
  scale_fill_viridis_c(
    option = "inferno",
    label = scales::label_number(scale = 1 / 1000)
  ) +
  labs(fill = "Empregos\nnecessíveis\n(em milhares)",
    color = "Corredores") +
  geom_sf(
    data = traj_corredores,
    aes(color = corredor),
    alpha = 0.8,
    show.legend = FALSE
  ) +
  scale_color_manual(values = c("#F8766D", "#7CAE00",
    "#00BFC4", "#C77CFF")) +
  theme_void() +
  theme(legend.key.size = unit(0.4, "cm"))

# configura mapa de diferença entre cenários
dist_diferenca <- ggplot() +
  geom_sf(
    data = subset(acesso_a_empregos_sf, !is.na(diferenca)),
```

```

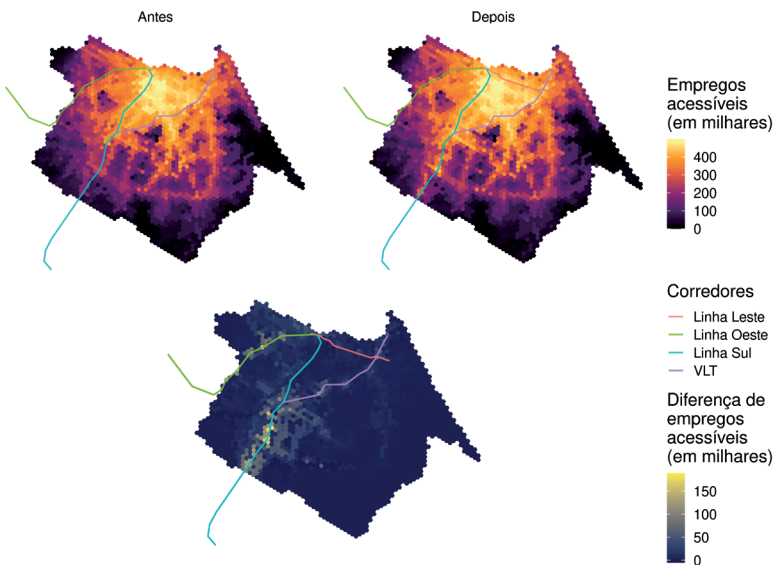
    aes(fill = diferenca),
    color = NA
  ) +
  scale_fill_viridis_c(
    option = "cividis",
    label = scales::label_number(scale = 1 / 1000)
  ) +
  labs(
    fill = "Diferença de\nempregos\nnecessíveis\n(em milhares)",
    color="Corredores"
  ) +
  geom_sf(data = traj_corredores, aes(color = corredor),
    alpha = 0.8) +
  scale_color_manual(values = c("#F8766D", "#7CAE00",
    "#00BFC4", "#C77CFF")) +
  theme_void() +
  theme(legend.key.size = unit(0.4, "cm"))

# combina as duas figuras
dist_acesso / dist_diferenca + plot_layout(ncol = 1,
  heights = c(1, 1))

```

FIGURA 16

Distribuição espacial dos níveis de acessibilidade ao emprego por cenário de transporte e da diferença entre eles



Fonte: Figura gerada pelo código supracitado.

A figura 16 mostra que as regiões que mais se beneficiam das mudanças no sistema de transportes são aquelas afastadas do centro da cidade, mas que ainda se encontram próximas a estações de média e alta capacidade. Os ganhos de acessibilidade ao emprego se concentram principalmente próximos às estações dos corredores das linhas Sul e Oeste do metrô e, em menor medida, em torno de algumas estações do corredor de VLT Parangaba-Mucuripe. Mesmo regiões próximas a esses corredores, embora não imediatamente adjacentes a eles, mostram grandes ganhos de acessibilidade, ressaltando o papel da conectividade da rede na garantia de boas condições de acessibilidade. Por sua vez, a região no entorno da linha Leste, que já concentrava os maiores níveis de acessibilidade da cidade mesmo antes da implementação do novo corredor, apresenta melhora nas condições de acessibilidade muito menos expressiva.

Os mapas de distribuição do tempo de acesso até a escola de nível médio mais próxima, no entanto, apresentam uma situação diferente, como mostrado a seguir.

```
# une os dados de tempo de acesso com a grade espacial de
# fortaleza e os
# converte em objetos espaciais
tempo_a_escolas <- merge(
  tempo_a_escolas,
  grade_fortaleza,
  by.x = "id",
  by.y = "id_hex"
)
tempo_a_escolas_sf <- sf::st_sf(tempo_a_escolas)

# configura mapas da distribuição de tempo a escolas nos cenários
# antes e depois
dist_tempo <- ggplot() +
  geom_sf(data = tempo_a_escolas_sf, aes(fill = travel_time_p50),
  color = NA) +
  facet_wrap(
    ~ cenario,
    nrow = 1,
    labeller = as_labeller(c(antes = "Antes", depois = "Depois"))
  ) +
  scale_fill_viridis_c(option = "plasma", direction = -1) +
  labs(fill = "Tempo até\n a escola\nmais próxima\n(em minutos)") +
  geom_sf(
    data = traj_corredores,
    aes(color = corredor),
```

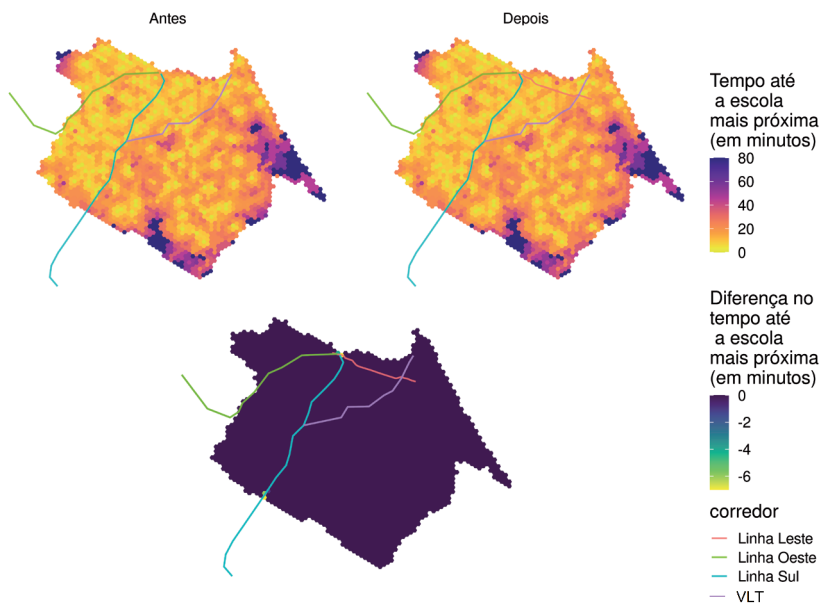
```
    alpha = 0.8,
    show.legend = FALSE
  ) +
  scale_color_manual(values = c("#F8766D", "#7CAE00",
    "#00BFC4", "#C77CFF")) +
  theme_void() +
  theme(legend.key.size = unit(0.4, "cm"))

# configura mapa de diferença entre cenários
dist_dif_tempo <- ggplot() +
  geom_sf(
    data = subset(tempo_a_escolas_sf, !is.na(diferenca)),
    aes(fill = diferenca),
    color = NA
  ) +
  scale_fill_viridis_c(option = "viridis", direction = -1) +
  labs(
    fill = "Diferença no tempo até a escola mais próxima (em minutos)"
  ) +
  geom_sf(data = traj_corredores, aes(color = corredor), alpha =
    0.8) +
  scale_color_manual(values = c("#F8766D", "#7CAE00",
    "#00BFC4", "#C77CFF")) +
  theme_void() +
  theme(legend.key.size = unit(0.4, "cm"))

# combina as duas figuras
dist_tempo / dist_dif_tempo + plot_layout(ncol = 1, heights = c(1, 1))
```


FIGURA 17

Distribuição espacial dos tempos de acesso à escola pública de nível médio mais próxima por cenário de transporte e da diferença entre eles



Fonte: Figura gerada pelo código supracitado.

A linha Leste e as mudanças de frequência nas outras linhas do metrô e do VLT praticamente não alteram o panorama da acessibilidade a escolas de nível médio em Fortaleza, o que faz com que pouquíssimos hexágonos apresentem algum tipo de ganho de acessibilidade entre os dois cenários – apenas alguns poucos nas imediações de estações de metrô apresentam alguma redução no tempo de acesso a escolas. Como podemos ver, a distribuição da acessibilidade a escolas na cidade segue um padrão muito menos concêntrico que o da distribuição da acessibilidade a empregos. Novamente, isso é uma consequência da distribuição de escolas públicas na cidade: ao contrário da distribuição de empregos, que tende a seguir critérios econômico-financeiros, a distribuição de equipamentos públicos tende a seguir critérios equitativos, buscando priorizar regiões periféricas e de maior concentração de grupos populacionais vulneráveis. Mesmo antes da implementação das mudanças no sistema de transportes, áreas com baixíssima acessibilidade ao emprego, como a região no extremo sul da cidade, já possuíam boas condições de acessibilidade a escolas públicas de nível médio. No entanto, o grau de sucesso das políticas de educação em promover um acesso equitativo pode variar muito entre cidades e para diferentes níveis de ensino, como infantil ou fundamental (Saraiva *et al.*, 2023).

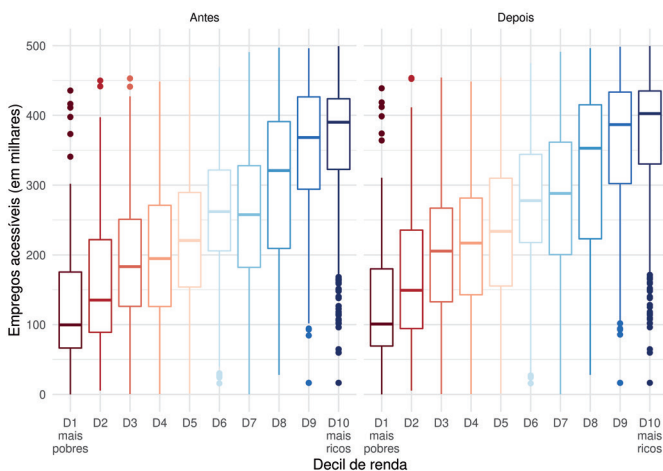
6.6 Desigualdade de acessibilidade

Uma dimensão-chave para a avaliação de políticas de transporte está relacionada aos seus aspectos distributivos. Quem são os principais beneficiados e prejudicados por essas políticas? Sob uma perspectiva de equidade, espera-se que políticas governamentais priorizem melhorar as condições de acessibilidade das pessoas em piores condições socioeconômicas e que mais dependem do transporte público (Pereira, Schwanen e Banister, 2017; Van Wee, 2022).

Nesta seção, vamos examinar como os ganhos de acessibilidade ao emprego decorrentes da construção da linha Leste e do aumento das frequências das linhas do metrô e do VLT estão distribuídos entre pessoas de diferentes níveis de renda. Para isso, precisamos entender como era a distribuição do nível de acesso a empregos entre a população em 2019, antes das mudanças no sistema de transportes, e como ela passará a ser após implementadas essas mudanças. Com o código a seguir, utilizamos a informação do decil de renda *per capita* em que cada célula da nossa grade espacial se encontra para entender a distribuição dos níveis de acessibilidade entre grupos socioeconômicos antes e depois das mudanças.

```
ggplot(data = acesso_a_empregos[populacao > 0]) +
  geom_boxplot(
    aes(
      x = as.factor(decil),
      y = acesso_a_empregos / 1000,
      color = as.factor(decil),
      weight = populacao,
      group = decil
    ),
    show.legend = FALSE
  ) +
  facet_wrap(
    ~ cenario,
    nrow = 1,
    labeller = as_labeller(c(antes = "Antes", depois = "Depois"))
  ) +
  scale_colour_brewer(palette = "RdBu") +
  labs(x = "Decil de renda", y = "Empregos acessíveis (em milhares)") +
  scale_x_discrete(
    labels = c("D1\nmais\npobres", paste0("D", 2:9),
              "D10\nmais\nricos")
  ) +
  theme_minimal()
```

FIGURA 18
Distribuição dos níveis de acessibilidade ao emprego entre decis de renda por cenário de transporte



Fonte: Figura gerada pelo código supracitado.

A figura 18 mostra claramente que as pessoas mais ricas de Fortaleza possuem maiores níveis de acesso a empregos do que seus pares mais pobres, tanto antes quanto depois das mudanças no sistema de transportes. Em Fortaleza, assim como na maior parte das grandes cidades brasileiras, os mais ricos tendem a morar próximo às grandes concentrações de emprego e do centro da cidade, enquanto os mais pobres tendem a ocupar regiões periféricas (Pereira *et al.*, 2022b). Consequentemente, os mais ricos costumam ter melhores condições de acessibilidade urbana do que os mais pobres, não apenas porque moram mais perto dos empregos, mas também porque, geralmente, moram em regiões centrais mais bem servidas de transporte público do que as periferias.

Entretanto, é difícil enxergar na figura 18 grandes mudanças nas condições de acessibilidade entre os dois cenários. Lançando mão do mesmo recurso que usamos anteriormente, apresentamos na figura 19 como os *ganhos* de acessibilidade entre os cenários se distribuem por decil de renda:

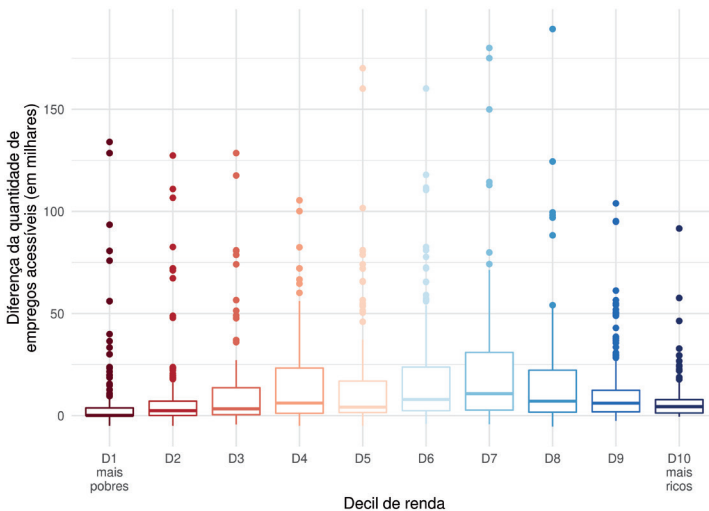
```
ggplot(subset(acesso_a_empregos, populacao > 0 & !is.na(
  diferenca))) +
  geom_boxplot(
    aes(
      x = as.factor(decil),
      y = diferenca / 1000,
      color = as.factor(decil),
      weight = populacao,
```

```

    group = decil
  ),
  show.legend = FALSE
) +
scale_colour_brewer(palette = "RdBu") +
labs(
  x = "Decil de renda",
  y = "Diferença da quantidade de\nempregos acessíveis
(em milhares)"
) +
scale_x_discrete(
  labels = c("D1\nmais\npobres", paste0("D", 2:9),
"D10\nmais\nricos")
) +
theme_minimal()

```

FIGURA 19
Distribuição da diferença dos níveis de acessibilidade ao emprego entre cenários de transporte



Fonte: Figura gerada pelo código supracitado.

Como podemos ver, a distribuição de ganhos de acessibilidade segue o formato de U invertido, com maiores ganhos entre as camadas médias da população e menores entre as camadas mais ricas e mais pobres. A célula na qual houve maior aumento de acessibilidade aparece como *outlier* do oitavo decil, apresentando um ganho de quase 200 mil empregos em relação ao cenário antes das mudanças.

Gráficos como os das figuras 18 e 19 não são os mais simples de comunicar, exatamente por serem ricos em nuances e interpretações. Visando facilitar essa comunicação, medidas sintéticas de desigualdade são frequentemente utilizadas como uma maneira simples de medir o impacto de políticas de transporte sobre a desigualdade de acessibilidade. Esse tipo de medida busca resumir a distribuição dos níveis de acessibilidade entre grupos populacionais (aqui, decis de renda) em um único indicador que facilite a compreensão dos resultados e que possa ser usado, por exemplo, na elaboração de planos e metas. As medidas de desigualdade mais frequentemente utilizadas na literatura de acessibilidade são a razão de Palma e o índice de Gini (Lucas, Van Wee e Maat, 2016; Guzman e Oviedo, 2018; Pritchard *et al.*, 2019).

Nesse exemplo, vamos calcular a razão de Palma dos cenários antes e depois das mudanças. Essa medida é o resultado da divisão da acessibilidade média dos 10% mais ricos pela acessibilidade média dos 40% mais pobres:

$$P = \frac{\overline{A_{tp10}}}{\overline{A_{bt40}}} \quad (7)$$

Em que P é a razão de Palma, $\overline{A_{tp10}}$ é a acessibilidade média dos 10% mais ricos e $\overline{A_{bt40}}$ é a acessibilidade média dos 40% mais pobres.

BOX 5

Por que usar a razão de Palma?

Uma das vantagens da razão de Palma em relação ao índice de Gini é a sua facilidade de comunicação e interpretação. Valores maiores do que 1 indicam um cenário em que os ricos possuem níveis médios de acessibilidade maiores do que os dos pobres, e valores menores do que 1, a situação inversa. Outra vantagem da razão de Palma é que ela reflete claramente como a desigualdade varia entre dois grupos de especial interesse para nós: os mais privilegiados e os mais vulneráveis de uma população. O índice de Gini, por sua vez, estima o quanto uma distribuição desvia de uma situação hipotética em que todos possuem o mesmo nível de acesso, mas nada diz sobre as condições socioeconômicas daqueles que detêm os maiores e menores níveis de acessibilidade. Se uma determinada política aumenta os níveis de acessibilidade de pessoas de alta renda que moram em locais de baixa acessibilidade, por exemplo, o índice de Gini indicaria que houve uma diminuição da desigualdade, mesmo que nenhuma população vulnerável tenha se beneficiado dessa política. Dificilmente uma política como essa poderia ser considerada equitativa.

Elaboração dos autores.

Ao calcularmos a razão de Palma nos cenários antes e depois da implementação da linha Leste e das mudanças de frequência das linhas de metrô e VLT, conseguimos entender o impacto dessas políticas na desigualdade de acesso a empregos em Fortaleza:

```
# calcula a acessibilidade média dos mais ricos nos dois cenários
acesso_mais_ricos <- acesso_a_empregos[
  decil == 10,
  .(acesso = weighted.mean(acesso_a_empregos, w = as.numeric
    (populacao))),
  by = cenario
]

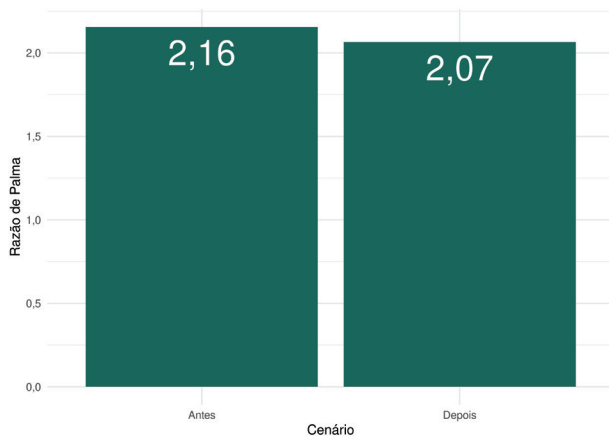
# calcula a acessibilidade média dos mais pobres nos dois cenários
acesso_mais_pobres <- acesso_a_empregos[
  decil %in% 1:4,
  .(acesso = weighted.mean(acesso_a_empregos, w = as.numeric
    (populacao))),
  by = cenario
]

# combina a acessibilidade dos mais ricos e mais pobres
razao_de_palma <- merge(
  acesso_mais_ricos,
  acesso_mais_pobres,
  by = "cenario",
  suffixes = c("_mais_ricos", "_mais_pobres")
)

# calculando Razão de Palma
razao_de_palma[, palma := acesso_mais_ricos / acesso_mais_pobres]

ggplot(data = razao_de_palma, aes(x = cenario, y = palma)) +
  geom_col(fill = "#0d6556") +
  geom_text(
    aes(label = formatC(palma, digits = 2, decimal.mark = ",",
      format = "f")),
    vjust = 1.5,
    color = "white",
    size = 10
  ) +
  scale_y_continuous(
    name = "Razão de Palma",
    labels = scales::label_number(decimal.mark = ",")
  ) +
  scale_x_discrete(name = "Cenário", labels = c("Antes", "Depois")) +
  theme_minimal()
```

FIGURA 20

Razão de Palma da acessibilidade ao emprego em Fortaleza por cenário de transporte

Fonte: Figura gerada pelo código supracitado.

A figura 20 aponta que em 2019 a população mais rica de Fortaleza conseguia acessar, em média, 2,16 vezes mais empregos do que a população mais pobre em até sessenta minutos por transporte público. Ela também mostra que a desigualdade medida pela razão de Palma sofreu uma pequena queda entre os cenários pré e pós-intervenções. Ou seja, podemos concluir que, neste cenário simplificado de avaliação da linha Leste, a política de expansão do metrô de Fortaleza combinada com o aumento da frequência das linhas Sul e Oeste do metrô e do VLT seria progressiva. Em outras palavras, essas mudanças, quando analisadas sem considerar as demais intervenções previstas no Pasfor, contribuem positivamente para a redução da desigualdade de acesso a oportunidades de emprego em Fortaleza.²¹

Neste capítulo, focamos na avaliação do impacto de uma política de transporte sobre as condições de acessibilidade da população. Vale ressaltar, no entanto, que a avaliação completa de uma política pública deve considerar também critérios como a participação social em sua formulação, além de outros possíveis impactos ambientais, econômicos e sociais. Embora uma avaliação de impacto de acessibilidade seja muito importante para identificar quais grupos se beneficiam de uma política de transporte e para entender como essa política impacta o desempenho do sistema de transportes como um todo, esse tipo de análise é apenas uma dimensão de avaliação e deve complementar ou ser complementada por outras análises.

21. É importante lembrar que a avaliação apresentada neste capítulo considera um cenário de intervenção simplificado para fins didáticos. Para uma avaliação mais completa da linha Leste do metrô de Fortaleza e das mudanças previstas no Pasfor, que também considera alterações nas linhas de ônibus da cidade, conferir o trabalho de Braga *et al.* (2022).