PondiônsTracker: A framework based on GTFS-RT to identify delays and estimate arrivals dynamically in Public Transportation Network

Pedro Pongelupe Lopes

Programa de Pós-Graduação em Informática

December 04 2023

PUC Minas

Contents

PUC Minas

Introduction

Motivation

- Public Transportation Network
- Smart cities
- GTFS and GTFS-RT specifications

PUC Minas

Motivation

GTFS-RT Matching Identifiers Issue

To work with GTFS-RT, it is **required** to track vehicles in real-time. But, in some cases, it is not easy to match the identifier between a real-time record and the GTFS static data. In Rome in 2016, this issue was reported by Raghothama et al. (2016). We still face this issue in Belo Horizonte in 2023.

Objectives

Main Objective

- Proposing and validating PondiônsTracker

Specific Objectives

- Collecting data from the real-time API and combining with the GTFS
- Understanding if Belo Horizonte's delays are spatial and temporal dependent by analyzing delays among bus stops
- Comparing the arrival times defined at the GTFS with the arrival times generated by *PondiônsTracker*.

PUC Minas

Smart Cities

Definition

Nam and Pardo 2011 describe a smart city is a city whose "data infuses information into its physical infrastructure to improve conveniences, facilitate mobility, add efficiencies, conserve energy, improve the quality of air and water, identify problems, and fix them quickly, recover rapidly from disasters, and collect data to make better decisions, deploy resources effectively, and share data to enable collaboration across entities and domains".

Smart Cities

Urban Computing

Interdisciplinary toolkit that uses the data generated in urban spaces by multiple sources to tackle issues that a city could face.

- Transportation, health, and tourism
- Sensors, devices, vehicles, buildings, and humans

Human Mobility

Human mobility aims to study the movement of humans through time and space. And its impacts on the environment.

- **Why do people migrate?**

PUC Minas

General Transit Feed Specification (GTFS)

Definition

Defines a common format for public transportation schedules and associated geographic information

Main ideas

- Open Data
- Many Apps, SDKs, libraries, and tools

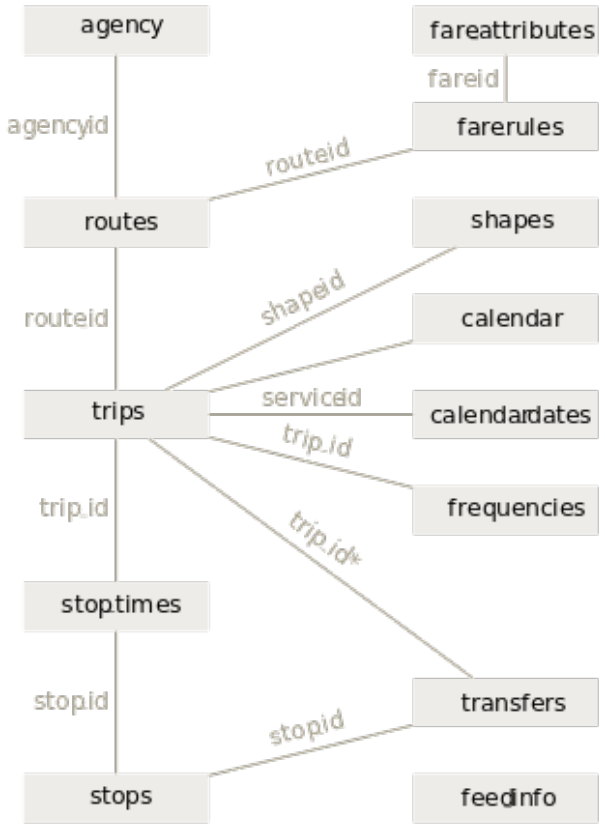


Figura: GTFS class diagram

PUC Minas

General Transit Feed Specification Real-Time (GTFS-RT)

GTFS-RT

GTFS-RT is the real-time extension to GTFS. GTFS-RT allows public transportation agencies to provide real-time updates about:

- Trips
- Services
- Vehicles

PUC Minas

Complex Networks and Graphs

Definition

Bacciu et al., 2020 state that "a graph has a compositional nature, being a compound of atomic information pieces and a relational nature, as the links defining its structure denote relationships between the linked entities".

- $g = (V_g, E_g, X_g, A_g)$

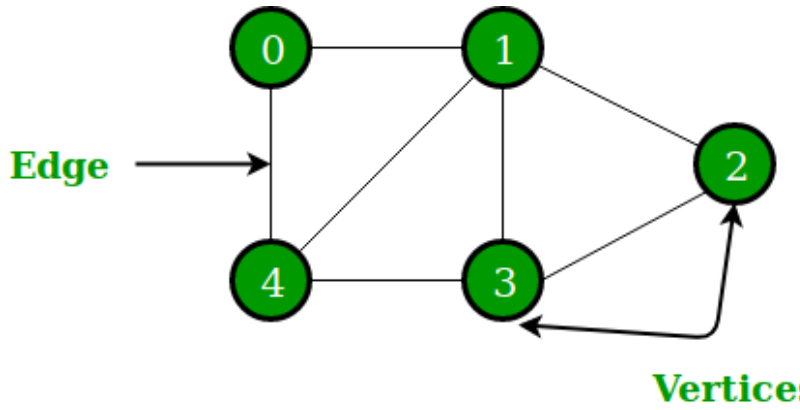
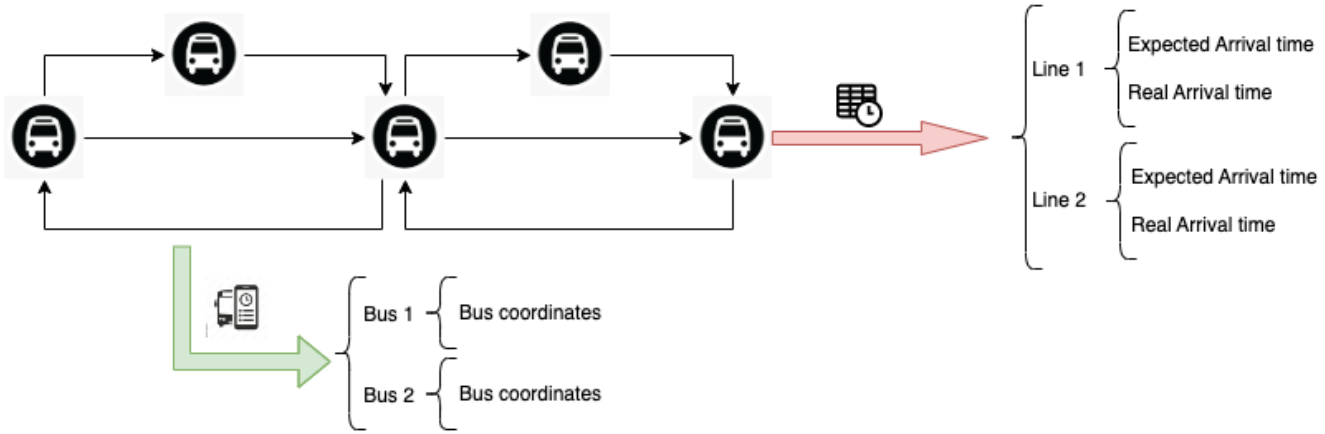


Figura: Example of an undirected graph

PUC Minas

Public Transportation Network as a Complex Network

$$G^{l} = (V_g, E_g, X_g, A_g)$$



G^{l}: Graph G at a given time l

V_g: Bus stops

E_g: Routes connecting two bus stops

X_g: Additional information about bus stops (V_g)

A_g: Additional information about routes connecting two bus stops (E_g)

PUC Minas

Analytics on Public Transport Delays with Spatial Big Data

Rome

- GTFS and Real-Time traffic API
- The link the datasets by comparing the expected arrival time and the real arrival time within a threshold
- Average delay for: stops, trip, trip at stop

Stockholm

- GTFS-RT
- Are the delays in the public transport network spatially dependent?
- What are the factors contributing to delays?
- What methods best suit their analysis of large real-time data streams?

PUC Minas

Discovering the space–time dimensions of schedule padding and delay from GTFS and real-time transit data

Overview

- Public Transportation Network molded as a complex network
- Legal, physical, or social constraints produce delays
- Using schedule padding to mitigate delays

Constructing a routable retrospective transit timetable from a real-time vehicle location feed and GTFS

Overview

- Enrich GTFS with real-time data unifying the two datasets in Toronto
- Algorithm outline after monitoring vehicles in real-time:
  1. Delimiting trips and blocks;
  2. Spatial matching and positional error handling;
  3. Determining stop times;
  4. Constructing the retrospective GTFS package.
- Open Source Routing Machine's map-matching algorithms

PondiônsTracker

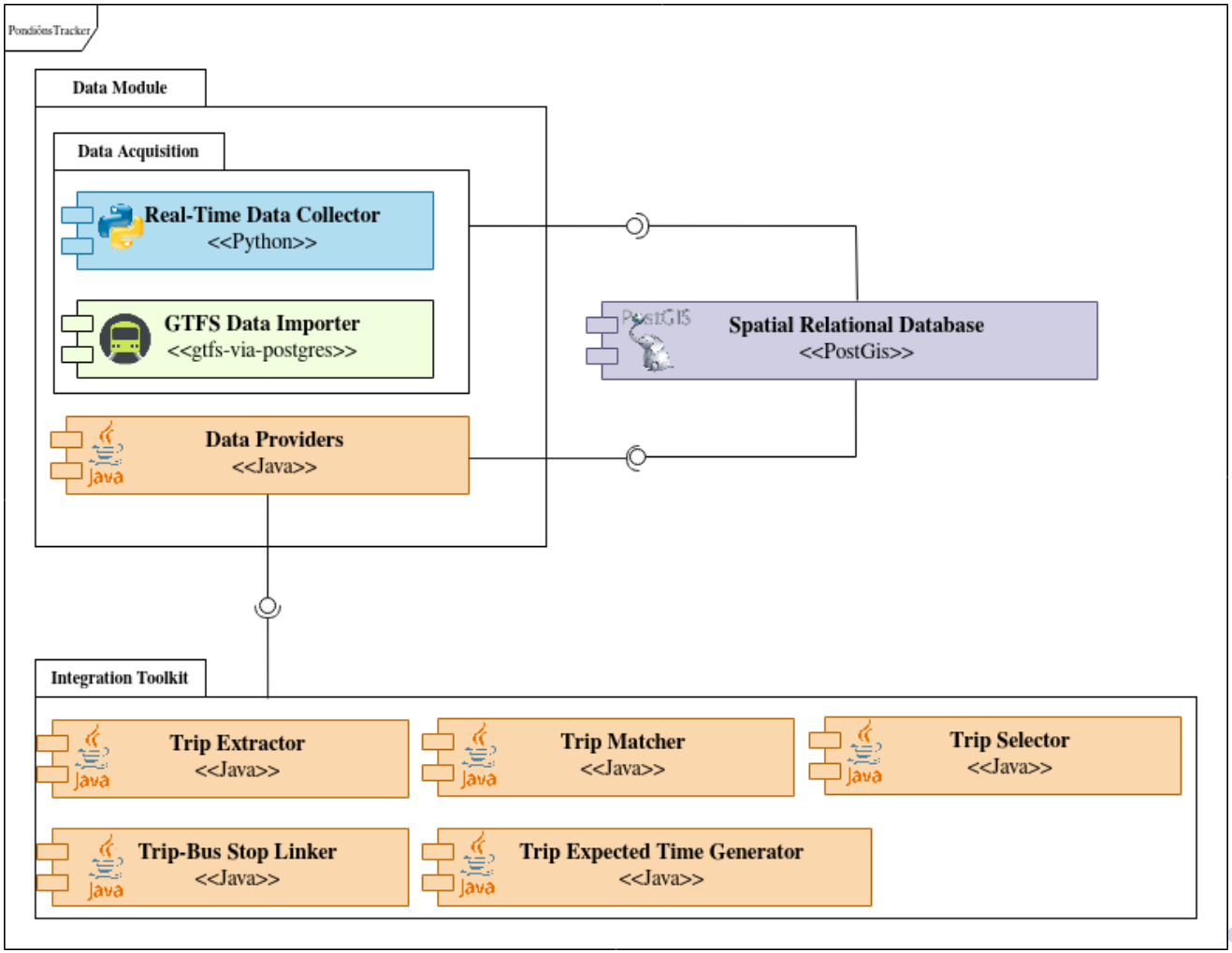
Overview

PondiônsTracker[a] is a framework to enrich GTFS data with real-time data. The name *PondiônsTracker* is a small gag from the sonority of the expression *bus stop* when pronounced in Portuguese with the accent from Minas Gerais.

[a] Available at `https://github.com/Pongelupe/PondionsTracker/`

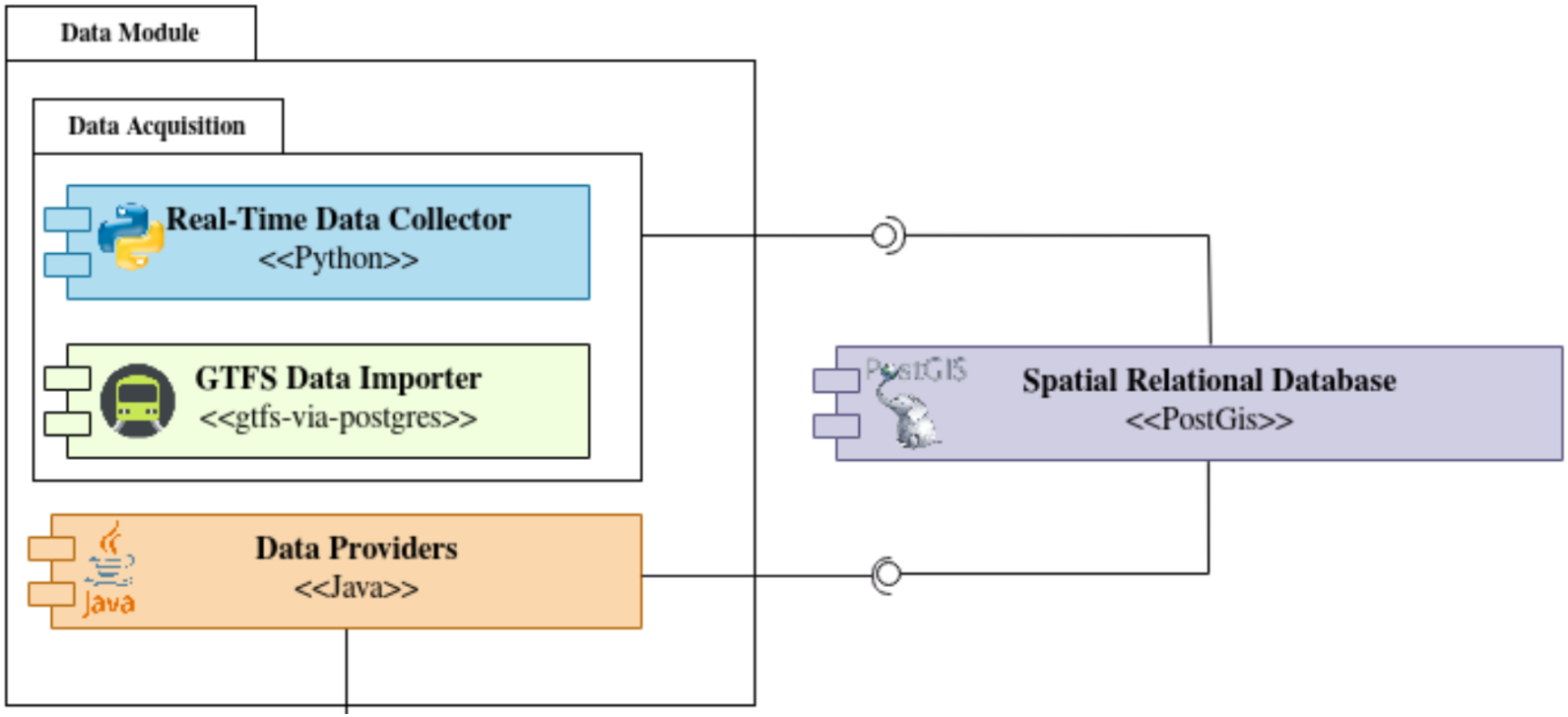
PUC Minas

PondiônsTracker's Architecture



PUC Minas

Data Module Overview



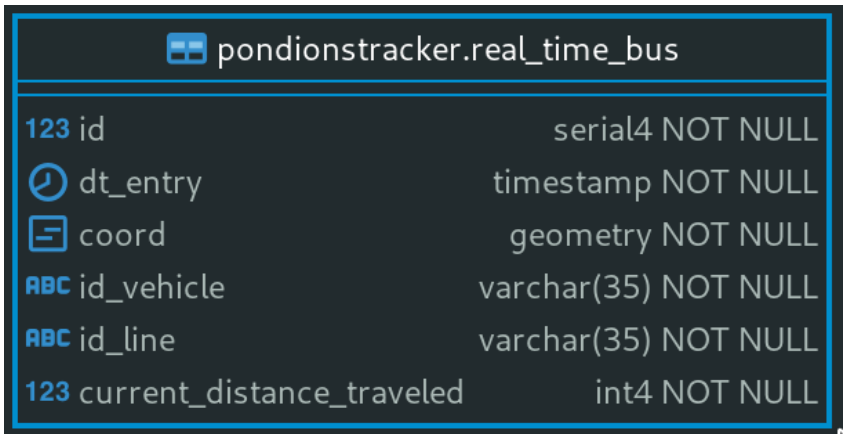
PUC Minas

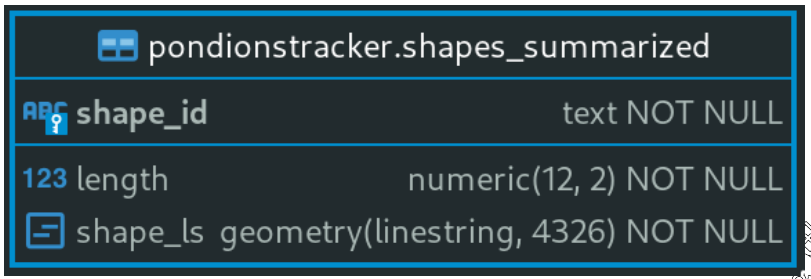
Data Acquisition - GTFS Data Importer

Basic Structure Needed

The *schema.sql* defines the two following tables:

- *real_time_bus*
- *shapes_summarized*





PUC Minas

Data Acquisition - GTFS Data Importer

gtfs-via-postgres

Import GTFS Static/Schedule datasets into a PostgreSQL database, to allow for efficient querying and analysis.

npm v4.8.2 | binary build failing | license Prosperity/Apache | node >=16.17 | support me donate | chat with me on Twitter

- ✅ handles daylight saving time correctly but retains reasonable lookup performance
- ✅ supports `frequencies.txt`
- ✨ joins `stop_times.txt` / `frequencies.txt`, `calendar.txt` / `calendar_dates.txt`, `trips.txt`, `route.txt` & `stops.txt` into views for straightforward data analysis (see below)
- 🚀 is carefully optimised to let PostgreSQL's query planner do its magic, yielding quick lookups even with large datasets (see performance section)
- ✅ validates and imports `translations.txt`
- ✨ exposes (almost) all data via GraphQL using PostGraphile

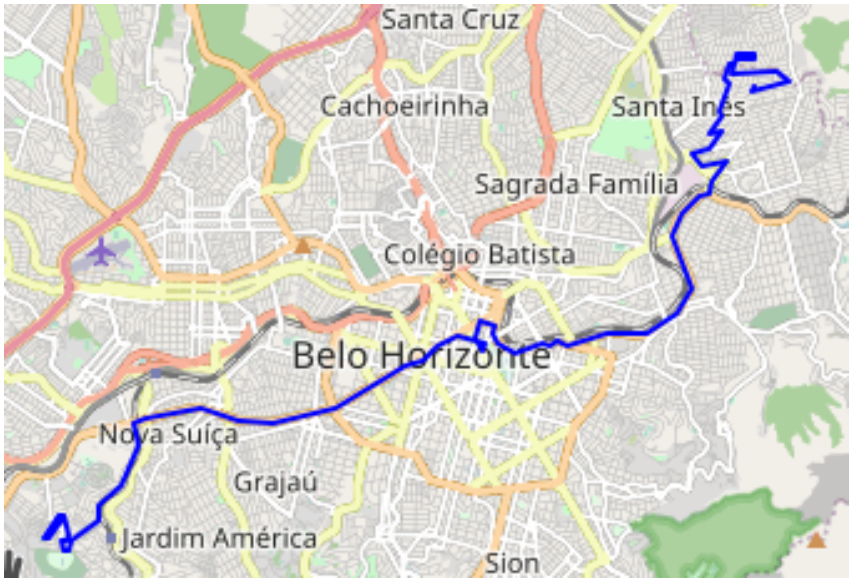
Figura: *gtfs-via-postgres*'s README

PUC Minas

Data Acquisition - GTFS Data Importer

Populate *shapes_summarized*

A SQL script which takes the *shapes* table grouped by the *shape_id* as input to the *PostGIS*'s function *ST_MakeLine* that produces a *LineString* from the given points.



PUC Minas

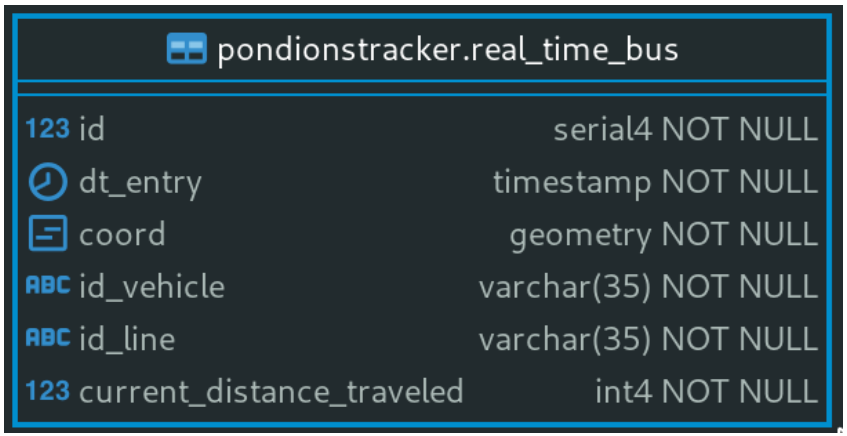
Data Acquisition - Real-Time Data Collector

Considerations

This component incorporates the data from real-time traffic *API* provided by some external source into *real_time_bus*

Example of an entry

A bus *b* of line 123 is at a bus stop near a drugstore and has already traveled 3 km on its route at 4 p.m.



PUC Minas

Data Providers

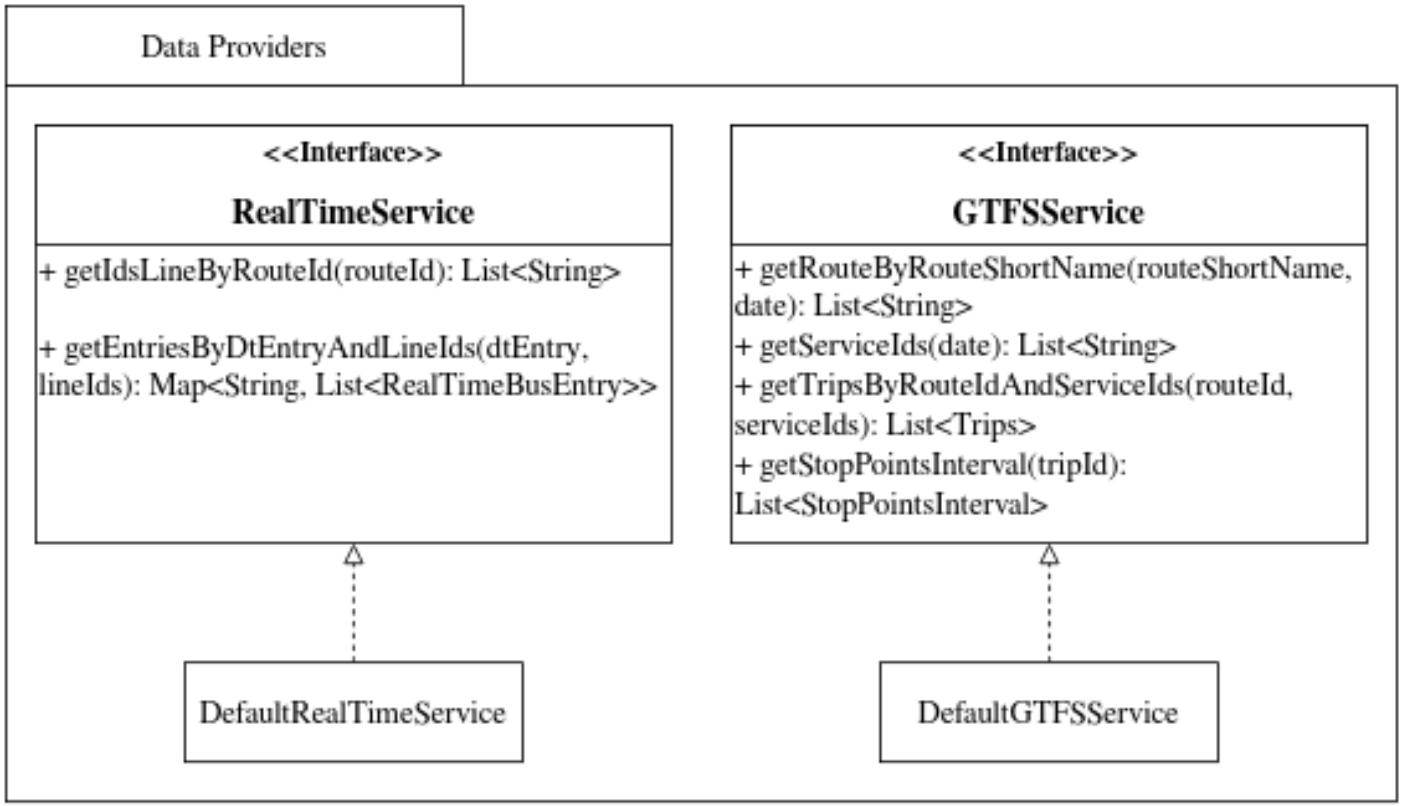


Figura: *PondiônsTracker*'s architecture diagram

PUC Minas

Data Providers

```
1  <dependency>
2    <groupId>br.pondionstracker</groupId>
3    <artifactId>data-module</artifactId>
4    <version>1.0.0</version>
5  </dependency>
```

Figura: *DataModule*'s maven dependency

Integration Module

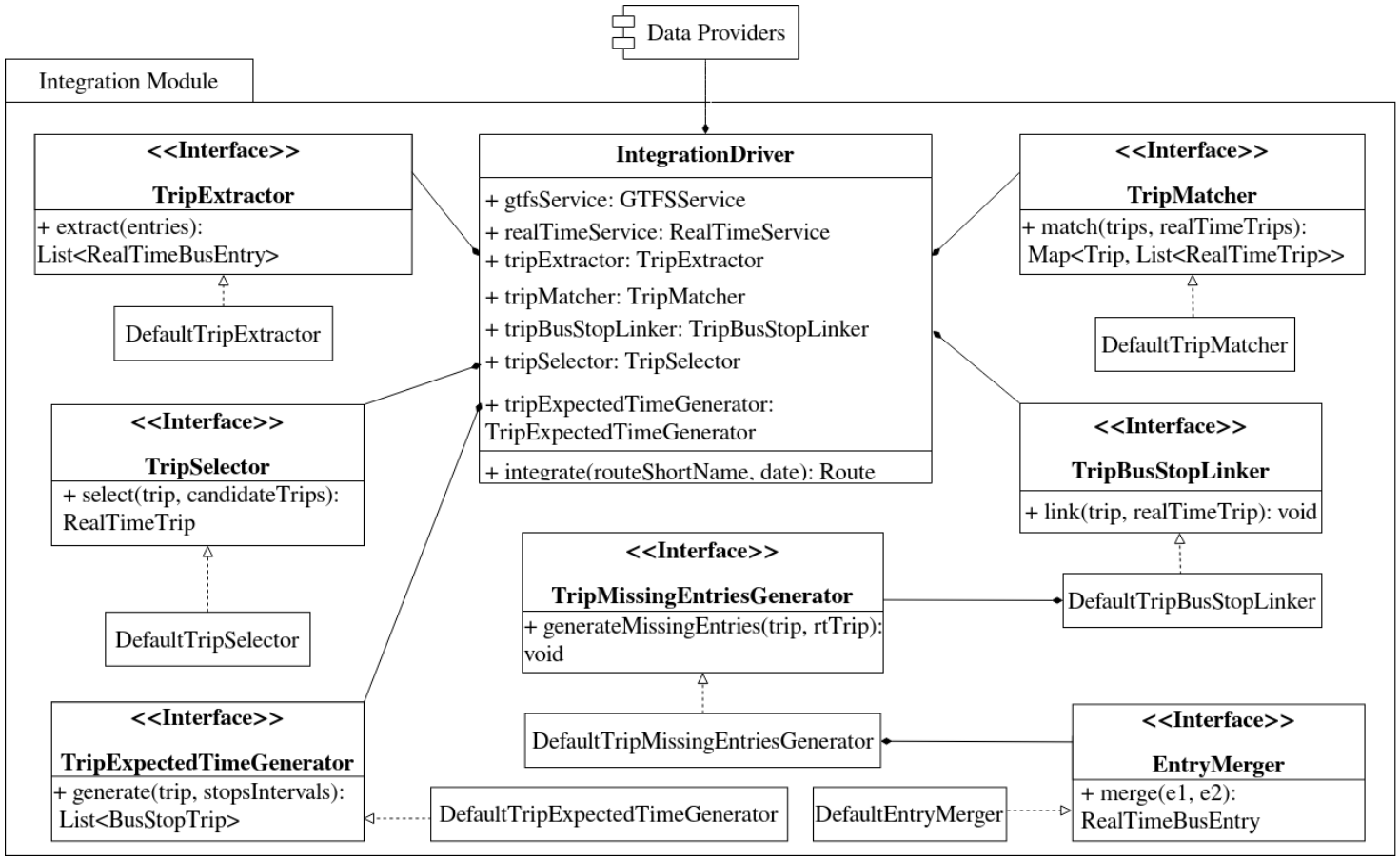


Figura: Integration Module Class Diagram

PUC Minas

Integration Driver

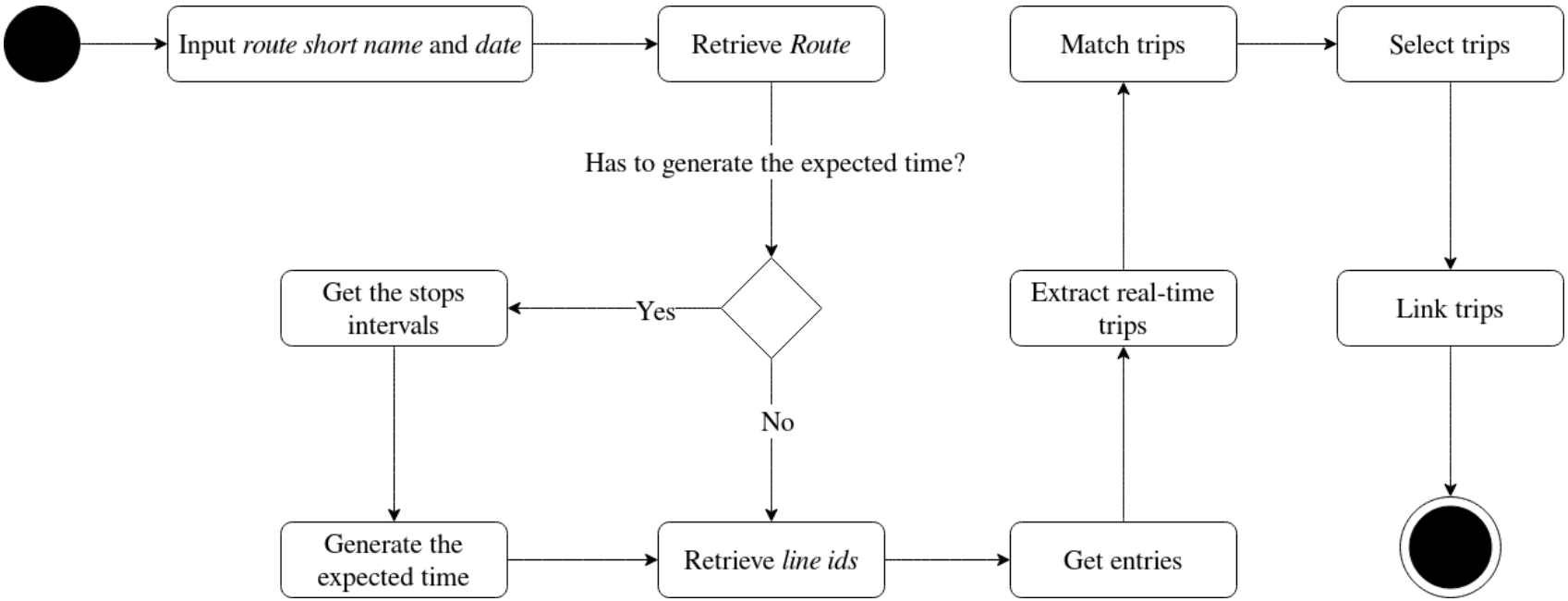


Figura: Integration Driver Activity Diagram

Integration Driver - 1st Step

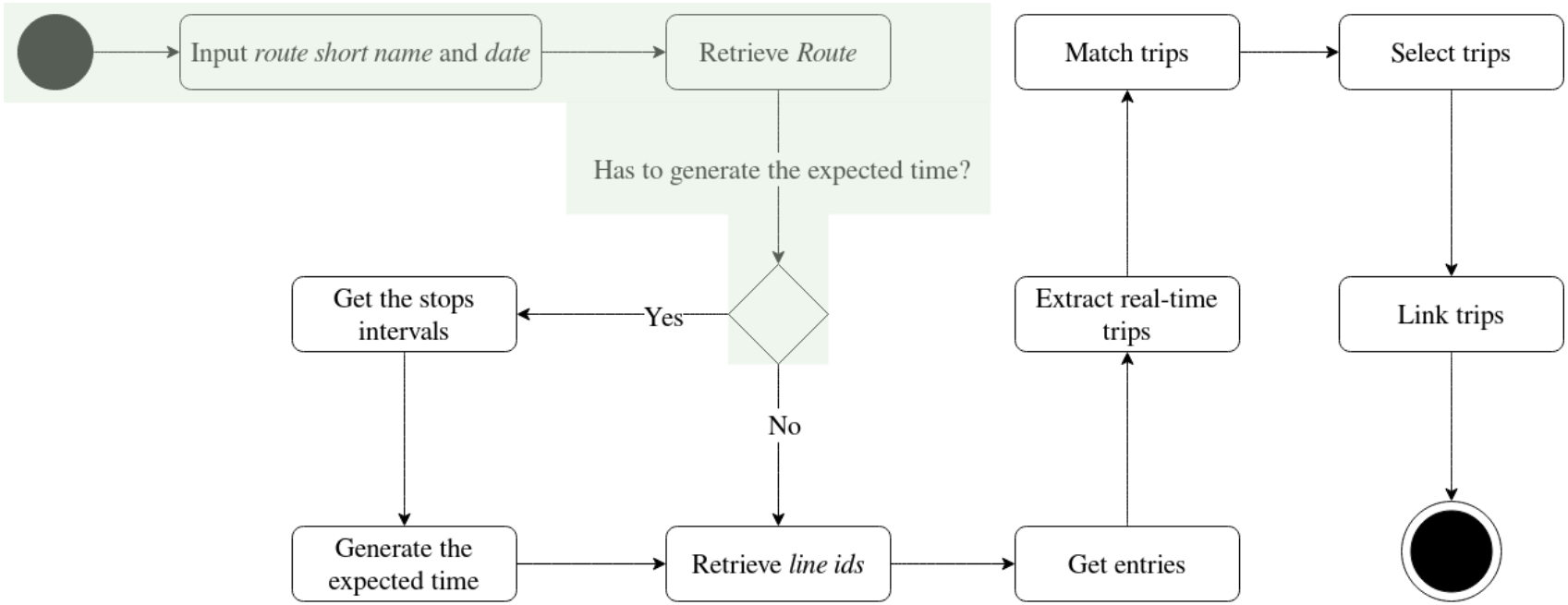


Figura: Integration Driver Activity Diagram

Integration Driver - 1st Step

Overview

1. Given a *route_short_name* and a target date, it uses the GTFSService to retrieve the Route.
2. Has to generate the expected time?

Has to generate the expected time?

`arrival_time`	Time	Conditionally required	Arrival time at a specific stop for a specific trip on a route. If there are not separate times for arrival and departure at a stop, enter the same value for `arrival_time` and `departure_time`. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. Scheduled stops where the vehicle strictly adheres to the specified arrival and departure times are timepoints. If this stop is not a timepoint, it is recommended to provide an estimated or interpolated time. If this is not available, `arrival_time` can be left empty. Further, indicate that interpolated times are provided with `timepoint=0`. If interpolated times are indicated with `timepoint=0`, then time points must be indicated with `timepoint=1`. Provide arrival times for all stops that are time points. An arrival time must be specified for the first and the last stop in a trip.

Figura: *arrival_time* definition from *stop_times.txt*

PUC Minas

Integration Driver - 2nd Step*

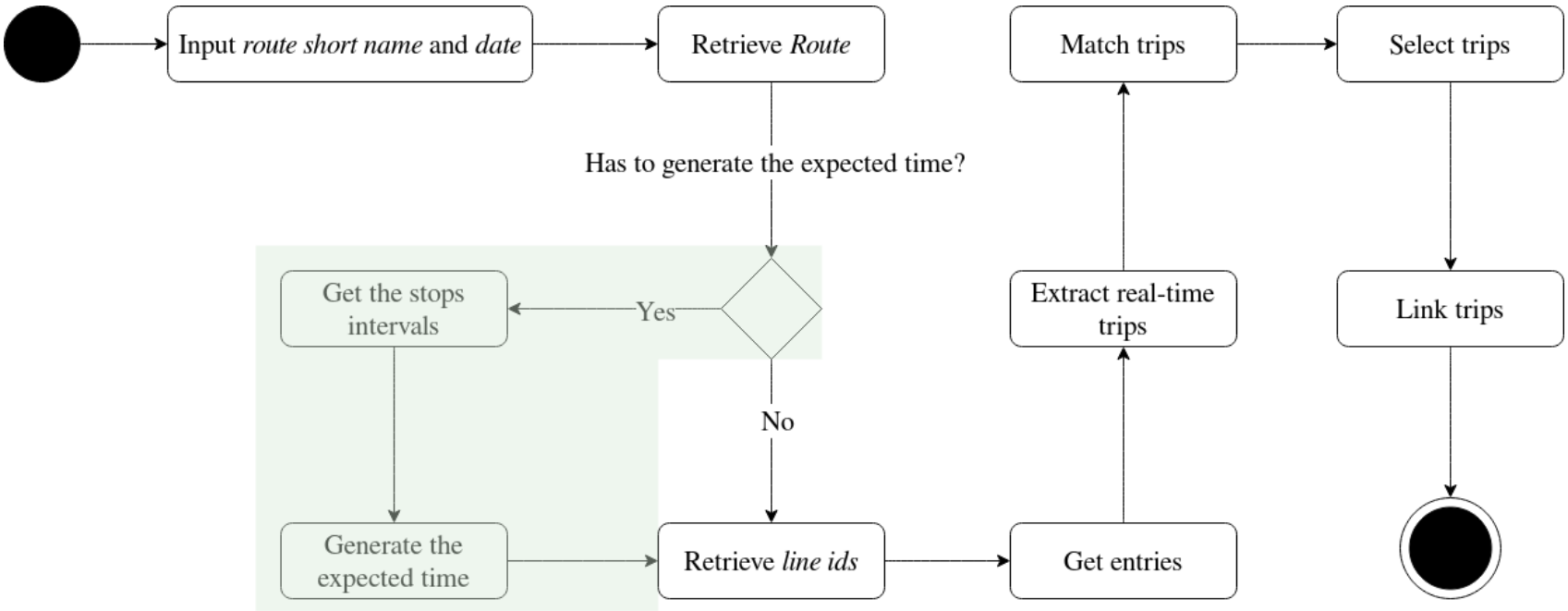


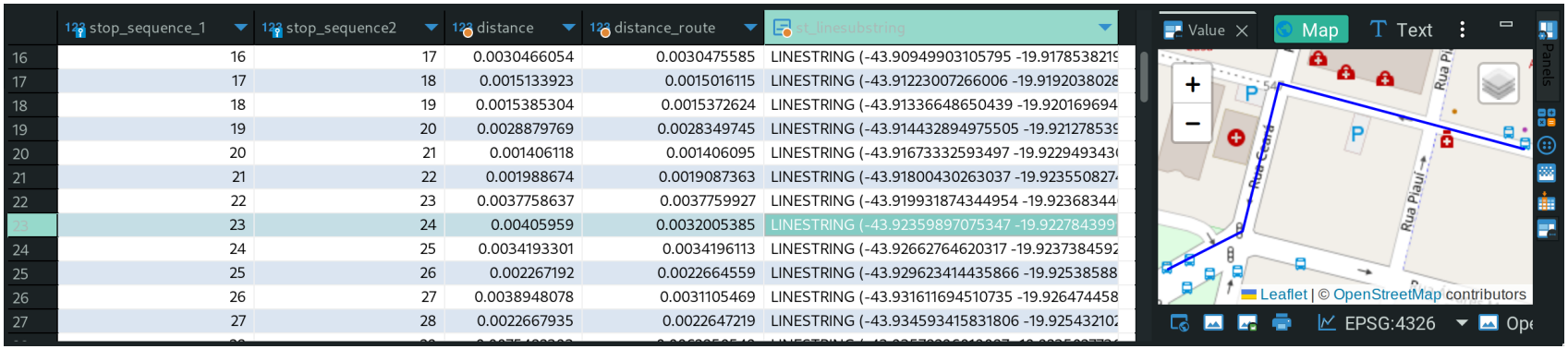
Figura: Integration Driver Activity Diagram

PUC Minas

Integration Driver - 2nd Step*

Overview

1. Get the stop intervals
2. Generate the expected time using *TripExpectedTimeGenerator*

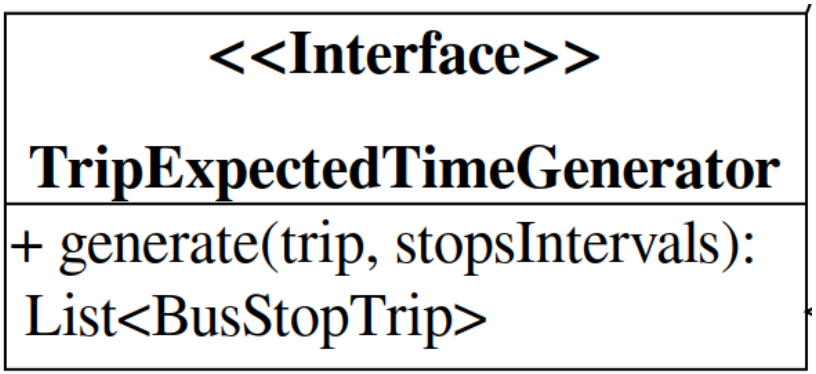


	stop_sequence_1	stop_sequence2	distance	distance_route	st_linesubstring
16	16	17	0.0030466054	0.0030475585	LINESTRING (-43.90949903105795 -19.9178538219
17	17	18	0.0015133923	0.0015016115	LINESTRING (-43.91223007266006 -19.9192038028
18	18	19	0.0015385304	0.0015372624	LINESTRING (-43.91336648650439 -19.920169694
19	19	20	0.0028879769	0.0028349745	LINESTRING (-43.914432894975505 -19.921278539
20	20	21	0.001406118	0.001406095	LINESTRING (-43.91673332593497 -19.922949343(
21	21	22	0.001988674	0.0019087363	LINESTRING (-43.91800430263037 -19.923550827
22	22	23	0.0037758637	0.0037759927	LINESTRING (-43.919931874344954 -19.92368344
23	23	24	0.00405959	0.0032005385	LINESTRING (-43.92359897075347 -19.922784399
24	24	25	0.0034193301	0.0034196113	LINESTRING (-43.92662764620317 -19.923738459
25	25	26	0.002267192	0.0022664559	LINESTRING (-43.929623414435866 -19.92538588
26	26	27	0.0038948078	0.0031105469	LINESTRING (-43.931611694510735 -19.926474458
27	27	28	0.0022667935	0.0022647219	LINESTRING (-43.934593415831806 -19.925432102

Figura: Example of bus stop couples and their distance

PUC Minas

TripExpectedTimeGenerator



Default Implementation key ideas

- Stop times are incremental in the trip and are incremented at each stop.
- $arrival\_time \rightarrow departure\_time$
- The bus *should* travel the trip in average speed
- Minor deviations to the original $departure\_time$

C Minas

Integration Driver - 3rd Step

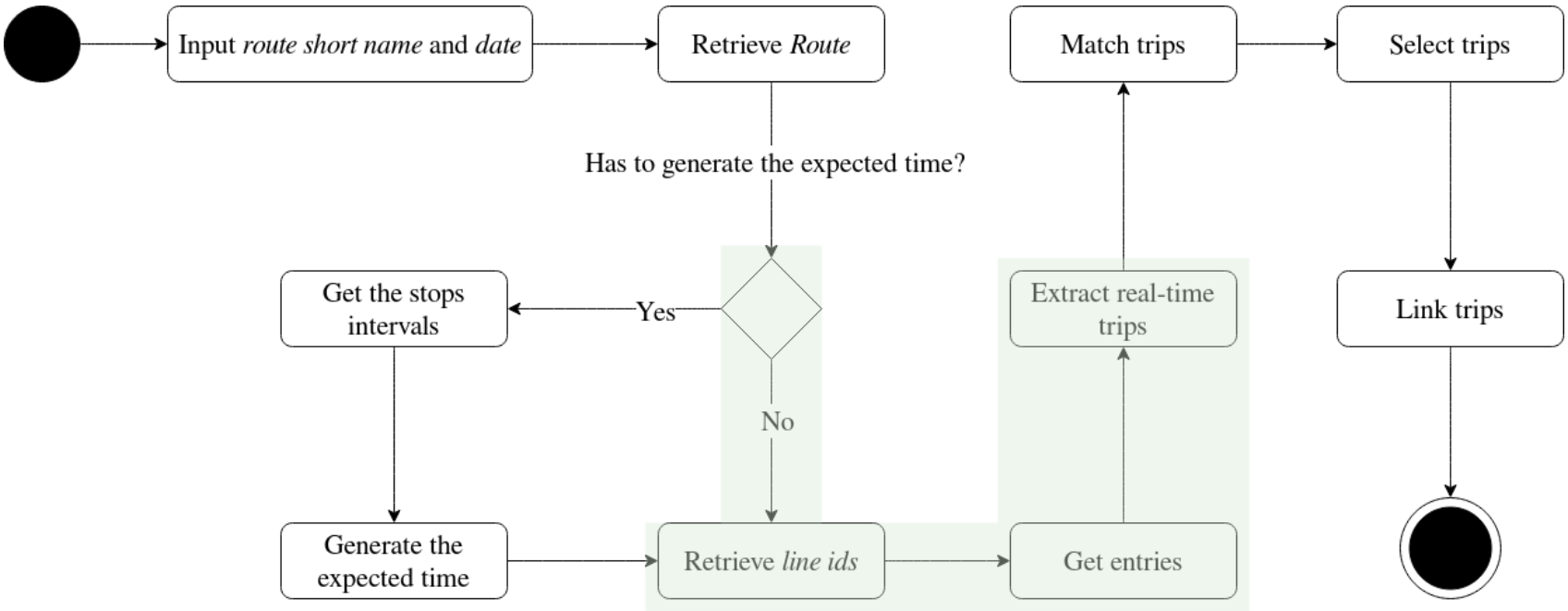


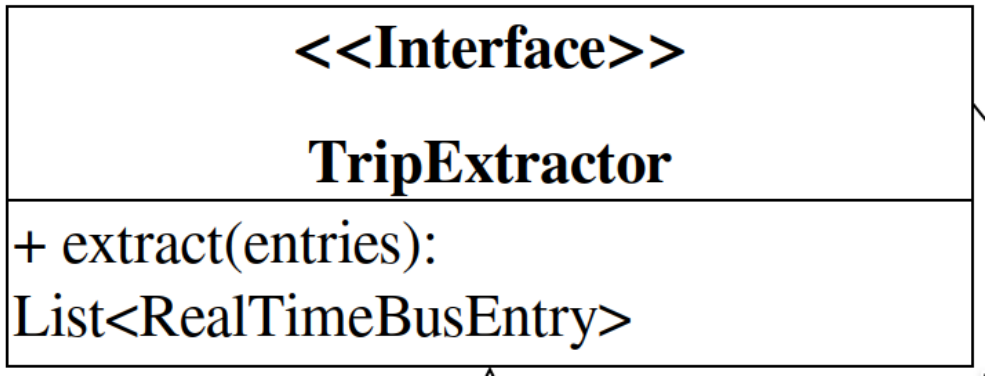
Figura: Integration Driver Activity Diagram

PUC Minas

Integration Driver - 3rd Step

Overview

1. Get line ids from *RealTimeService* using *getRouteByRouteShortName*
2. Get entries using *getEntriesByDtEntryAndLineIds*
3. Extract real-time trips from entries using *TripExtractor*



PUC Minas

TripExtractor

Default Implementation Premises

1. The entries must be from the same *idVehicle*
2. The entries must be sorted by *dtEntry* in order to preserve their chronology
3. The *currentDistanceTraveled* can only increase for the same trip

Integration Driver - 4th Step

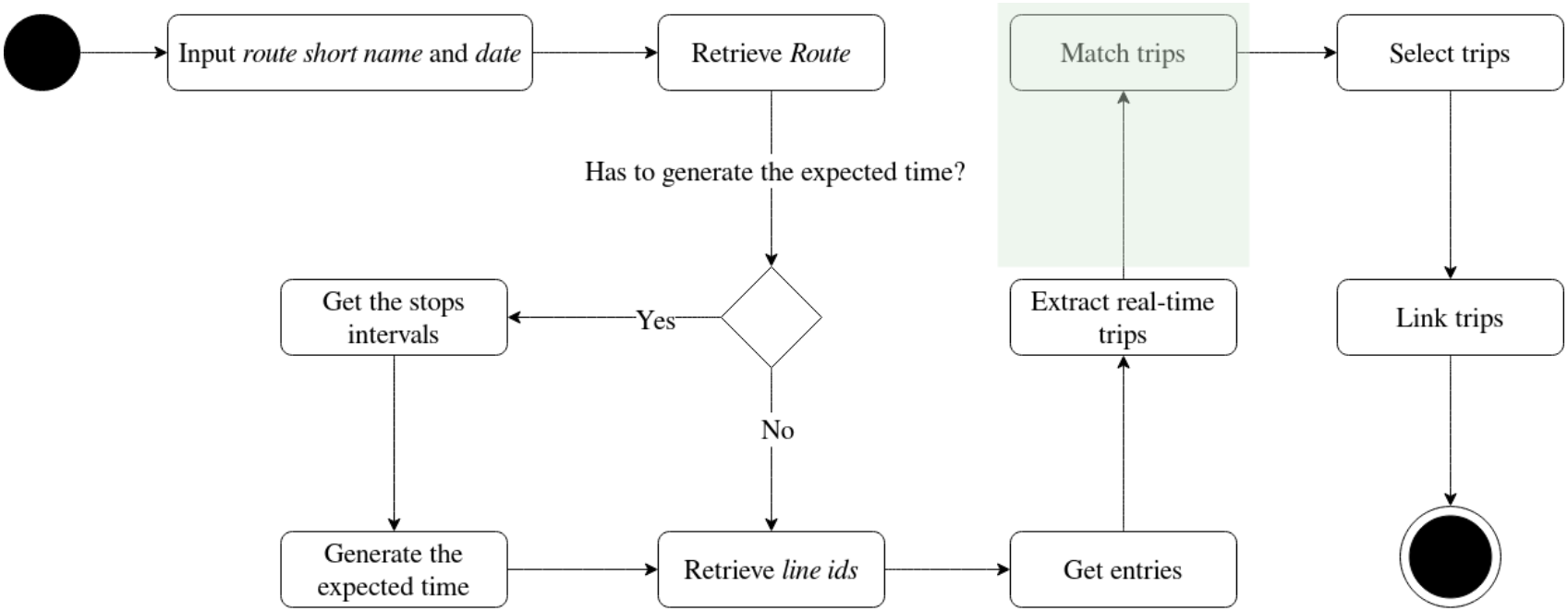
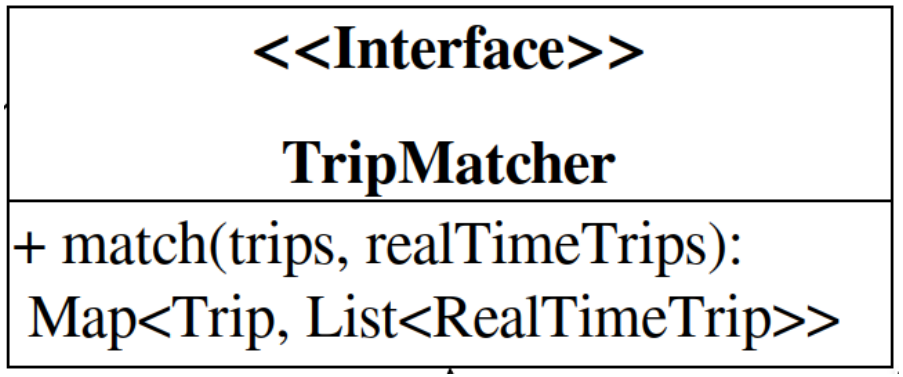


Figura: Integration Driver Activity Diagram

TripMatcher

Default Implementation key ideas

- Linking static data to real-time data
- Compare schedules using padding
- Different values for *maxTripInitialShifting*
- For a scheduled trip there are many *candidate trips*



PUC Minas

Integration Driver - 5th Step

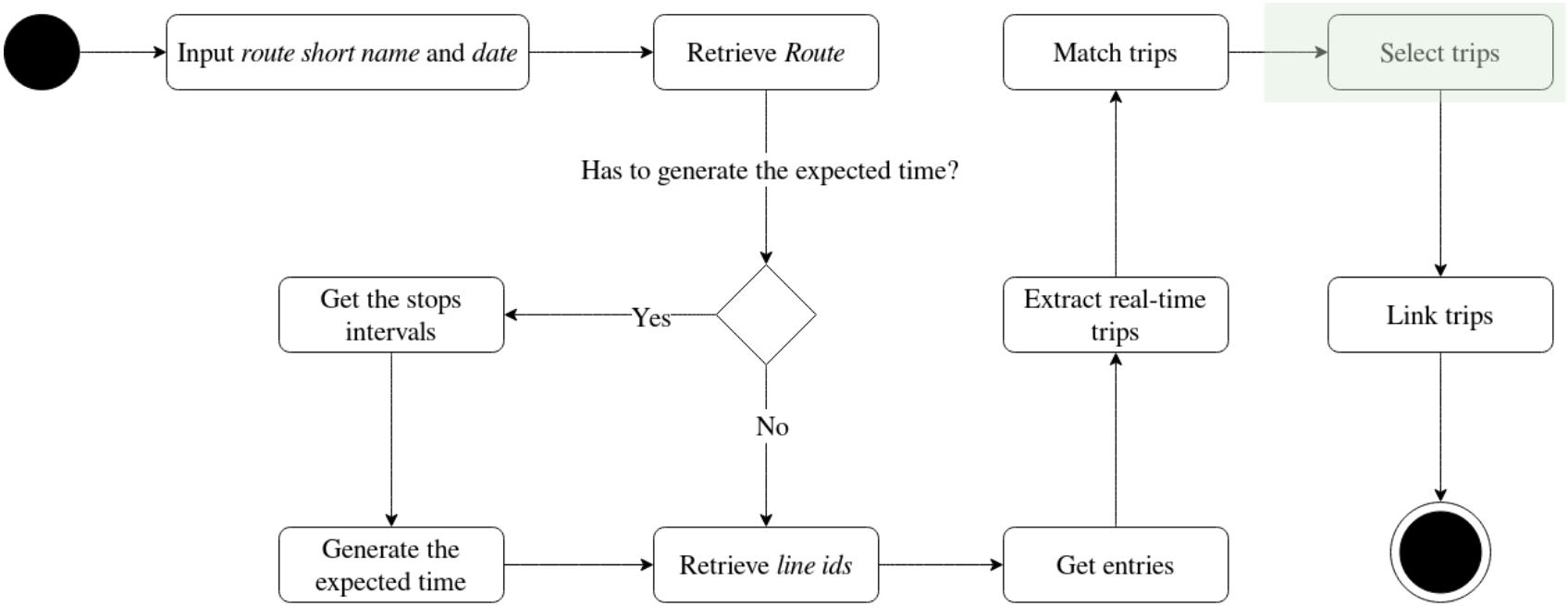


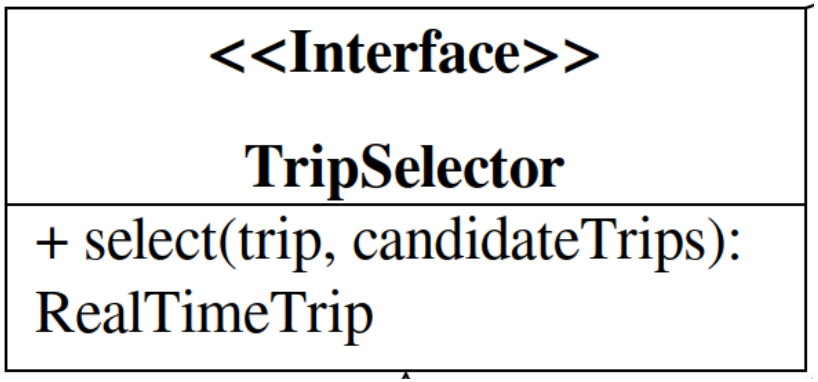
Figura: Integration Driver Activity Diagram

PUC Minas

TripSelector

Default Implementation key ideas

- *Select* the *RealTimeTrip* trip that better fits the *Trip*
- Removing unwanted trips
  - The displacement from the bus garage to a bus stop
  - Bus breaking down.
  - The bus' communication system is malfunctioning.
  - A bus' last entry recorded is still en route.



PUC Minas

TripSelector

Default Implementation Premises

1. The last entries *dtEntry* must be after the *departureTime*;
2. The trip must have traveled at least a *tripMinPercentageTraveled* percentage of the *Trip*'s *length*.

Default Implementation Considerations

- Selects the trip with the latest *departureTime* that fills the preconditions
- 85% for trip minimum percentage traveled

PUC Minas

Integration Driver - 6th Step

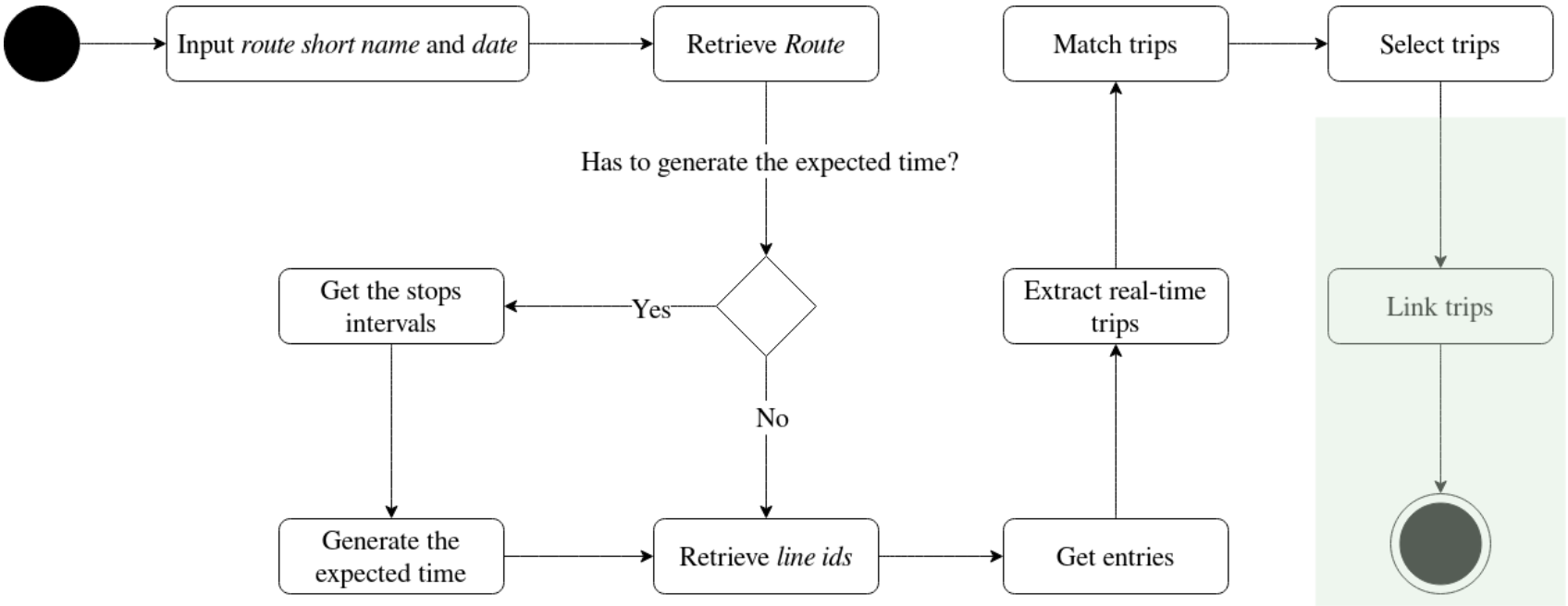


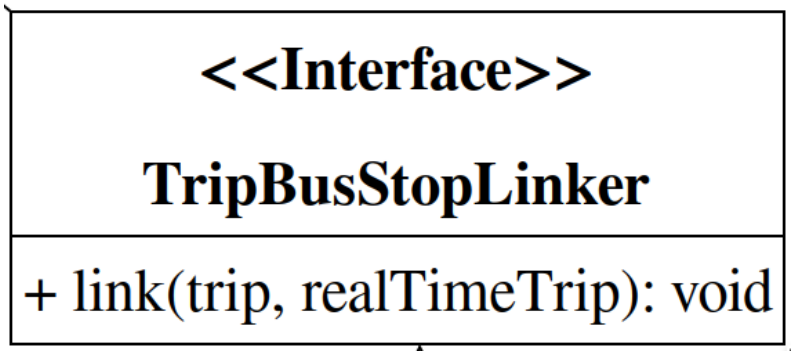
Figura: Integration Driver Activity Diagram

PUC Minas

TripBusStopLinker

Default Implementation Overview

- Link a real-time Trip to the bus stops from a scheduled Trip



PUC Minas

TripBusStopLinker

Default Implementation Premises

- *RealTimeTrip* have been **around** every bus stop from their route
- What is the concept of an entry to be **around** a bus stop?
- Distance threshold d_t

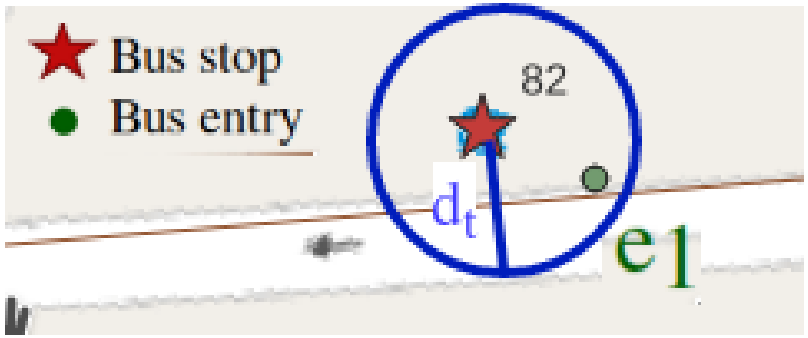


Figura: Entries and d_t representations

PUC Minas

TripBusStopLinker

Default Implementation key ideas

Iterate over *Trip*'s *busStopSequence* and for each bus stop s_x it is searched for all the **valid** entries within d_t*, then scanning the *RealTimeTrip*'s *entries* set.

Relationship between Entries and Bus Stops

An entry e to a bus stop s_x:

1. An entry e can be associated with only one s_x
2. A s_x can be related to more than one entry e
3. A s_x can be related to **zero entries**

PUC Minas

Empty Set of Entries

Why do we have an empty set?

This does not imply that the bus has not been around a stop on the trip, it might be only a GPS positioning error. For example, if a bus is at a certain speed, it passes by a bus stop without stopping because there is no boarding or landing at that given stop.

Figura: A bus stop with no entries related

PUC Minas

TripMissingEntriesGenerator

Default Implementation Overview

- Generate an artificial entry by merging a couple of entry
- The *EntryMerger* is the component in charge of merging two entries
  - **Linear Interpolation**

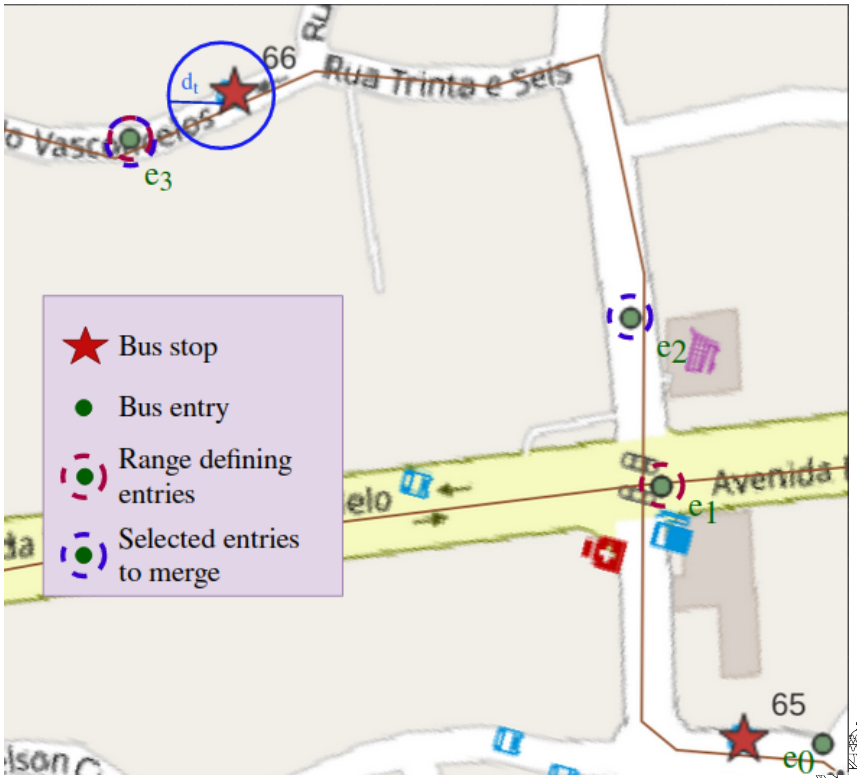
	dt_entry	$coord$	$id\_vehicle$	$current\_distance\_traveled$
e_{n-1}	10:00	coord e_{n-1}	1	5000
e_n	10:02	coord e_n	1	5100
$e_{(n-1,n)}$	10:01	coord $e_{(n-1,n)}$	1	5050

Figura: Example of merge of e_{n-1} and e_n

TripMissingEntriesGenerator

Which entries are going to be merged?

- Defining an interval of candidates entries
- *Lower Bound Entry* and *Upper Bound Entry*
- The closest entries to the bus stop from each side of the interval



PUC Minas

Integration Driver

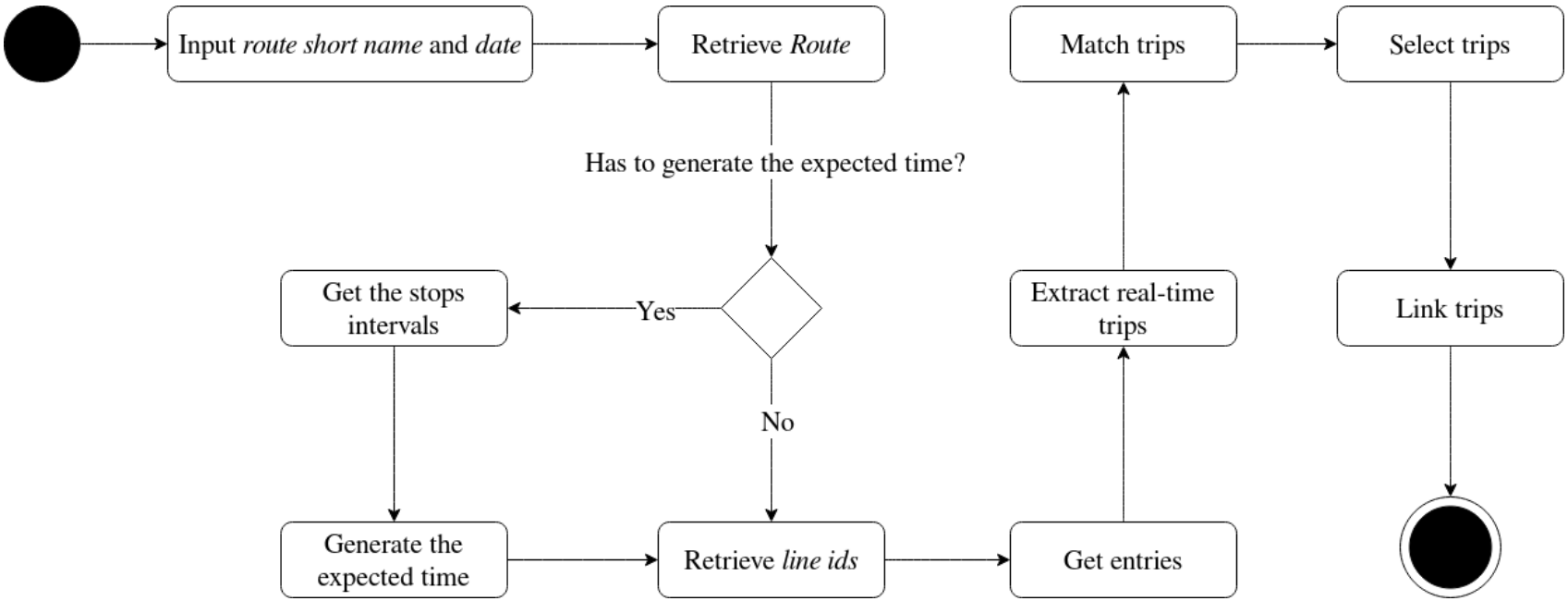


Figura: Integration Driver Activity Diagram

Integration Module

```
1  <dependency>
2    <groupId>br.pondionstracker</groupId>
3    <artifactId>integration-module</artifactId>
4    <version>1.0.0</version>
5  </dependency>
```

Figura: *IntegrationModule*'s maven dependency

PondiônsTracker-BH

Overview

PondiônsTracker-BH[a] is a *PondiônsTracker*'s specialization created to deal with Belo Horizonte's Public Transportation Network particularities. So, we have implemented our own *Real-Time Data collector*, and we have overwritten the method *getIdsLineByRouteId* from the *RealTimeService*.

- *BHTrans* $\rightarrow$ *GTFS*
- *Transfacil* $\rightarrow$ *Traffic API*

[a] Available at `https://github.com/Pongelupe/PondionsTracker-BH`

PUC Minas

Belo Horizonte's RealTimeService

BHRealTimeService

BHRealTimeService extends DefaultRealTimeService and overrides *getIdsLineByRouteId* method. There is a **one-to-many** relationship between the GTFS and the real-time data.

Workload Overview

Workload

- Data collected for 11 days straight in August 2023
- 30 Gigabytes

Date	Day-of-Week	Entries
29-07-23	Saturday	22,319,765
30-07-23	Sunday	22,635,117
31-07-23	Monday	22,583,380
01-08-23	Tuesday	22,432,739
02-08-23	Wednesday	21,970,073
03-08-23	Thursday	22,050,579
04-08-23	Friday	22,402,865
05-08-23	Saturday	22,642,955
06-08-23	Sunday	22,786,254
07-08-23	Monday	22,109,606
08-08-23	Tuesday	22,405,222
Total	-	**246,338,555**

PUC Minas

Schedule Analysis

Schedule-Filled Percentage

Schedule-Filled Percentage = **Matched Trips** / **Scheduled Trips**

- **Total**: 156,628 / 205,884 = 76.08%
- **Weekdays**: 118,559 / 159,418 = 74.37%
- **Saturdays**: 22,796 / 28,200 = 80.84%
- **Sundays**: 15,273 / 18,266 = 83.61%

PUC Minas

Schedule Analysis

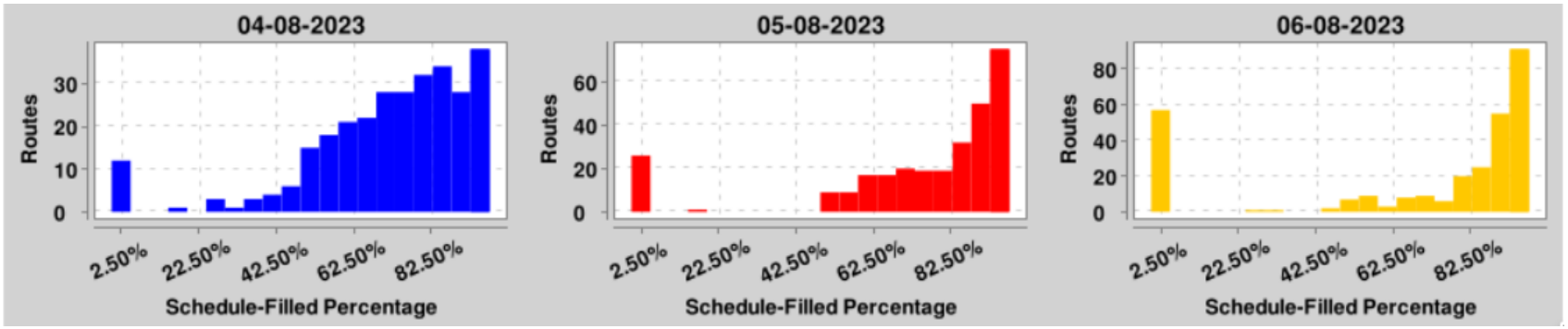


Figura: Histograms Schedule-Filled Percentage per Routes

Schedule Analysis - Schedule Deviations

Schedule Deviations

Regarding the real-time API, there are collected trips which were not defined at Belo Horizonte's GTFS.

82 - Estação São Gabriel / Savassi Via Hospitais

On Sundays, the GTFS does not schedule any trip for route 82, but the API provided entries regarding this route twice during the period observed.

PUC Minas

Delay Analysis

Delay Notation

- **Delay**: $\geqslant$ 1 minute after
- **Ahead-of-Schedule**: $\geqslant$ 1 minute before
- **On time**: $\leqslant$ 59 seconds after OR $\leqslant$ 59 seconds before

	Weekday	Saturday	Sunday
Total trips matched	118,559	22,796	15,273
Trips entirely out of schedule	60,244	10,899	7,148
Trips with departure or arrival on time	39,403	8,731	5,988
Trips with departure and arrival on time	324	95	56
Trips entirely on time	1	2	1

Figura: Delays detailed in whole Public Transportation Network scale

PUC Minas

Delay Analysis

331 - Estação Barreiro/Conjunto Antonio Teixeira Dias Via Upa

Has 32 bus stops, representing a length of almost 9 kilometers.

1. Jul. 29 15:30:00 - 15:56:27 $\longrightarrow$ Jul. 29 15:30:03 - 15:57:03
2. Jul. 30 08:20:00 - 08:46:27 $\longrightarrow$ Jul. 30 08:20:31 - 08:46:15
3. Aug. 04 05:40:00 - 06:06:27 $\longrightarrow$ Aug. 04 05:40:30 - 06:06:00
4. Aug. 05 17:10:00 - 17:36:27 $\longrightarrow$ Aug. 05 17:10:45 - 17:36:49

PUC Minas

Delay Analysis

Distribution of each status over the network

- **Delay**: 89.8%
- **Ahead-of-Schedule**: 6.9%
- **On time**: 3.3%

Attention!

The predominance of *DELAYED* in the Public Transportation Network **does not imply** that the network is not working nor completely stopped!

Delay Analysis

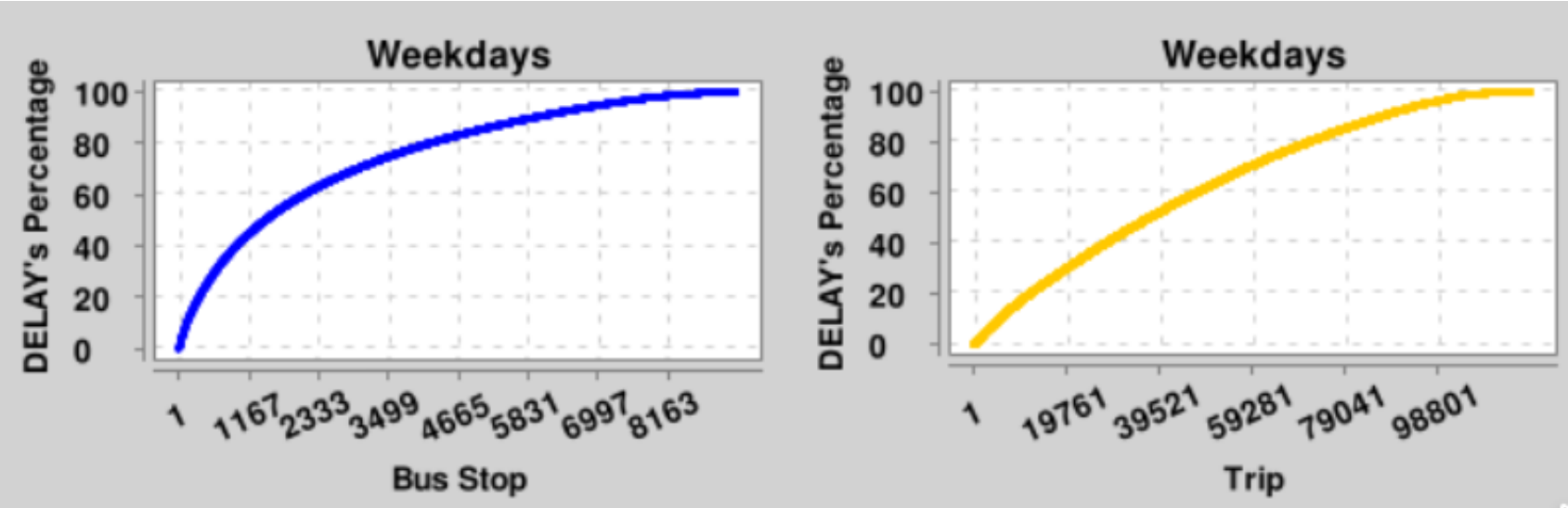


Figura: *DELAY*s Distribution: Bus Stop and Trip

PUC Minas

Delay Analysis

Figura: 300 Most Delayed Stops

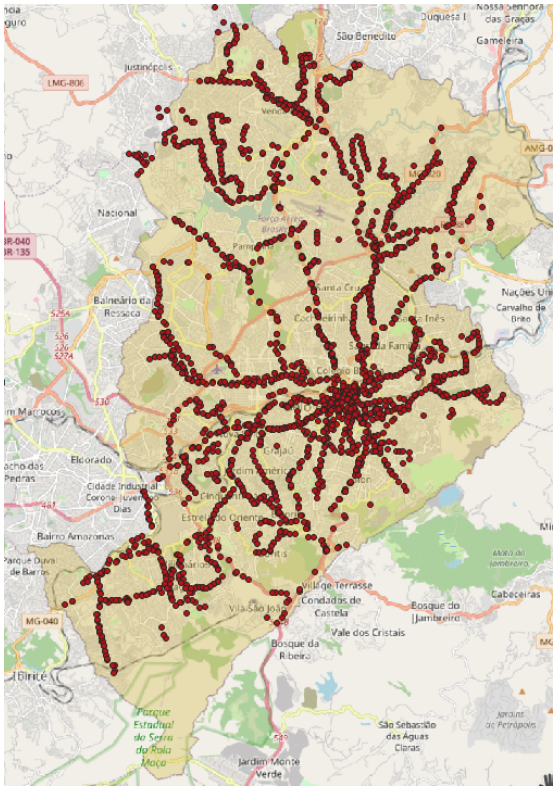
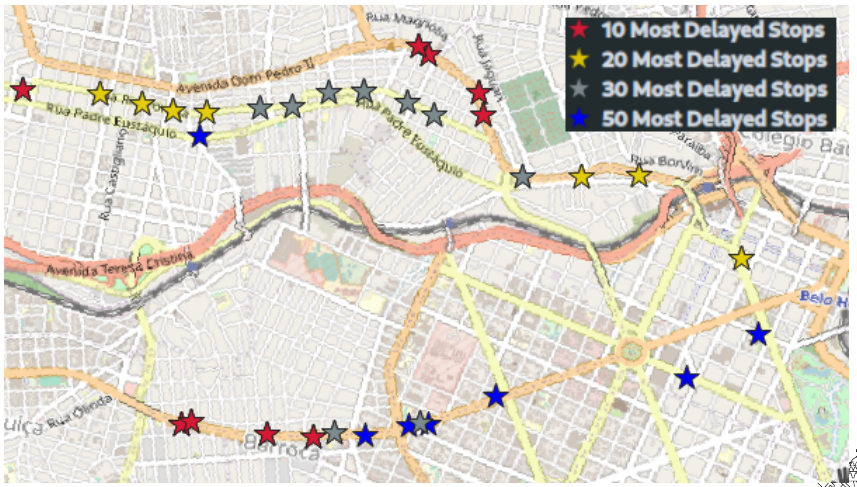


Figura: Fragment of the 50 Most Delayed Stops



PUC Minas

Delay Analysis

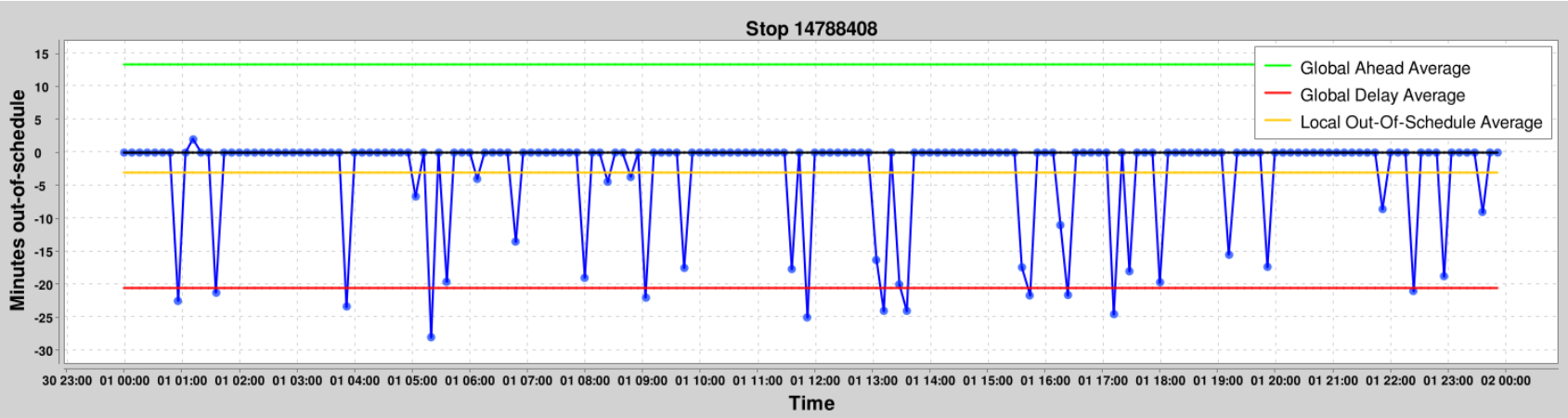
Three Most Delayed Stops for Weekdays

1. #14793268 - *Avenida Dom Pedro II 1520* with 7,309 delays
2. #14791617 - *Avenida Amazonas 7309* with 7,009 delays
3. #14790997 - *Avenida Dom Pedro II 1980* with 6,692 delays

PUC Minas

Delay Analysis

Figura: Minutes out of schedule of bus stops #14788408 distributed throughout the day



Constants

1. *Global Ahead Average*: 13.42 minutes
2. *Global Delay Average*: 20.49 minutes

Delay Analysis

Stops #14793268 and #14790997

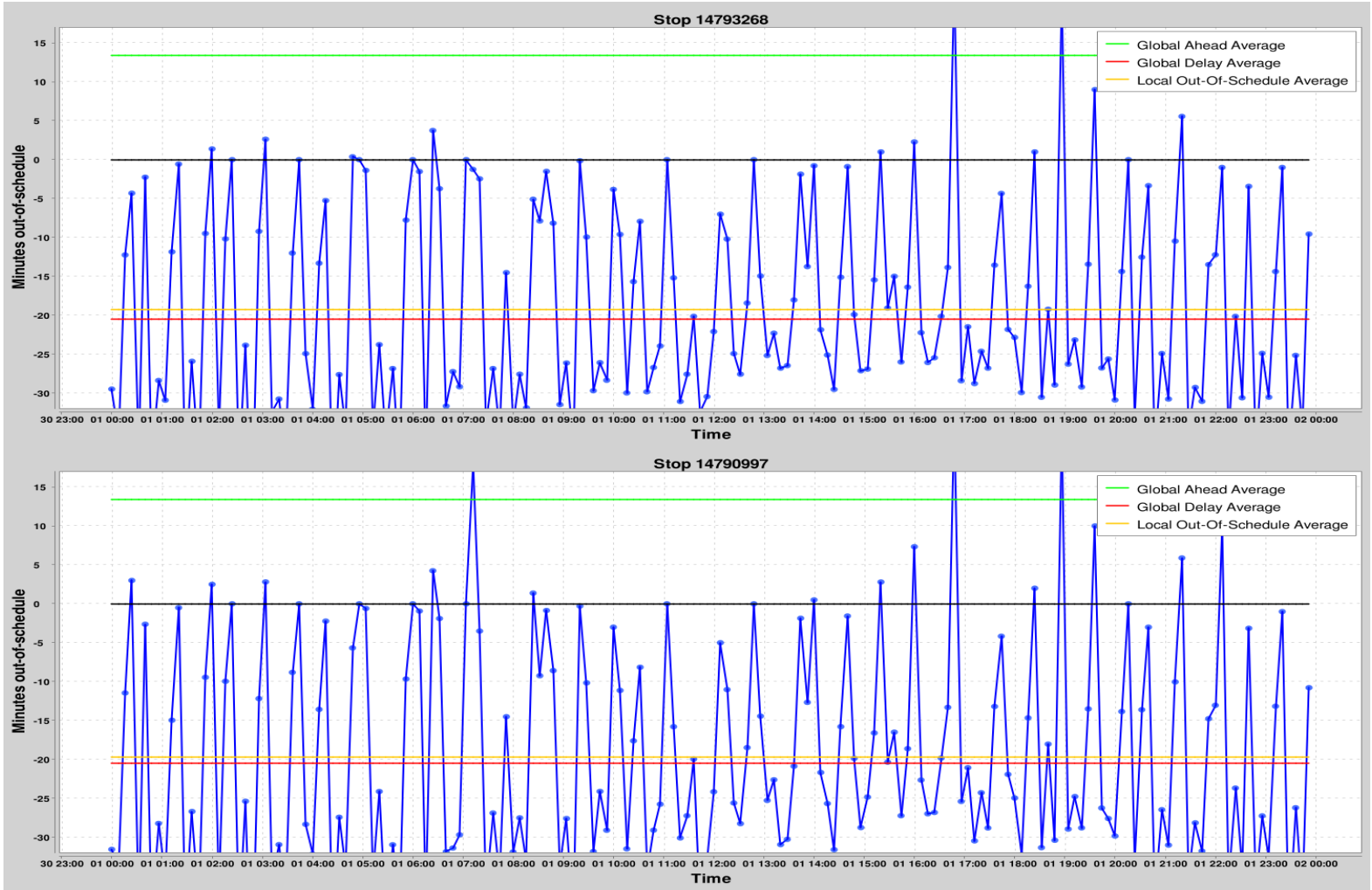
The stops #14793268 and #14790997 are the first and third most delayed in the Public Transportation Network, respectively. Also, these stops are **462** meters from each other on the same avenue, *Avenida Pedro II*, and share **2,590** common trips, so they are spatially related.

Local Out-Of-Schedule Average

- #14793268: 19.29 minutes
- #14790997: 19.68 minutes

PUC Minas

Delay Analysis



PUC Minas

Comparison Between Generated and Real Data

Overview

The previous analysis was only possible because Belo Horizonte's GTFS defines the expected time for all bus stops on every trip. The *Trip Expected Time Generator* generates the expected times when missing, so, we executed this component with Belo Horizonte's data and compared the expected times generated with those defined at the GTFS.

PUC Minas

Comparison Between Generated and Real Data

Trip entirely out of schedule

- GTFS: 78,291
- Generated: 75,073
- **Diff**: 3,218 (**4.29%**)

Trip entirely on time

- GTFS: 4
- Generated: 0
- **Diff**: 4 (**100%**)

Trips with departure **or** arrival on time

- GTFS: 54,122
- Generated: 54,271
- **Diff**: 149 (**0.27%**)

Trips with departure **and** arrival on time

- GTFS: 475
- Generated: 596
- **Diff**: 121 (**25.47%**)

PUC Minas

Comparison Between Generated and Real Data

		GTFS	Generated
Weekday	*ON_TIME*	3.3%	3.2%
	AHEAD_OF_SCHEDULE	6.9%	17.8%
	DELAYED	89.8%	79.0%
Saturday	*ON_TIME*	3.9%	3.5%
	AHEAD_OF_SCHEDULE	6.5%	18.4%
	DELAYED	89.6%	78.1%
Sunday	*ON_TIME*	4.4%	3.9%
	AHEAD_OF_SCHEDULE	5.4%	18.5%
	DELAYED	90.2%	77.6%

Comparison Between Generated and Real Data

Global Averages

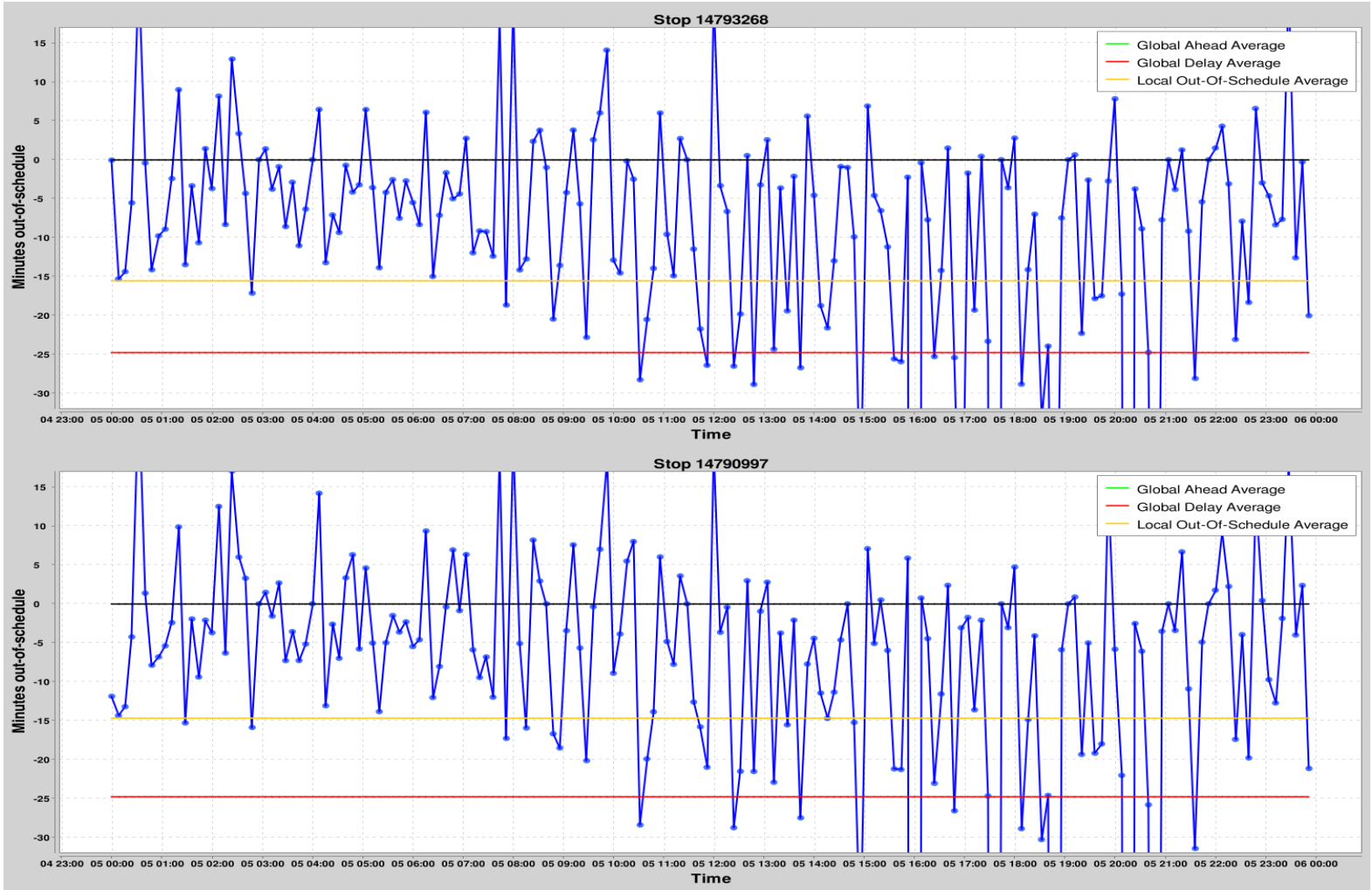
- *Global Ahead Average*
  - GTFS: 13.42 minutes
  - Generated: 38.57 minutes
  - **Diff**: 25.15 minutes
- *Global Delay Average*
  - GTFS: 20.49 minutes
  - Generated: 24.75 minutes
  - **Diff**: 4.26 minutes

Local Out-Of-Schedule Average

- #14793268
  - GTFS: 19.29 minutes
  - Generated: 15.54 minutes
  - **Diff**: 3.75 minutes
- #14790997
  - GTFS: 19.68 minutes
  - Generated: 14.68 minutes
  - **Diff**: 5 minutes

PUC Minas

Comparison Between Generated and Real Data



PUC Minas

Limitations

Limitations

The *Real-Time Data Collector* is the most fragile component due to the third-party real-time traffic API interface.

- Size and quality of the data
- Scheduled routes with no entries reported
  1. *720 - Circular Saúde MG20* missed 175 trips
  2. *912 - Conjunto Taquaril/Praça Che Guevara* missed 210 trips

PUC Minas

Conclusion

Concluding Remarks

- Comparison between the expected schedule with the actual schedule
- Delays in Belo Horizonte follow a *log-normal* distribution
- Analysis using data generated with the *Trip Expected Time Generator*
- *PondiônsTracker* as a viable option when GTFS-RT is unavailable

PUC Minas

Conclusion

Future Work

- Futher explore Belo Horizonte Public Transportation Network using deep learning for graphs approaches
- Reproduce Belo Horizonte's results with other cities
- Explore the delays analysis combining temporal and spatial dimensions

PUC Minas

Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. 2020. A gentle introduction to deep learning for graphs. *Neural Networks* 129 (sep 2020), 203–221. https://doi.org/10.1016/j.neunet.2020.06.006

Taewoo Nam and Theresa A. Pardo. 2011. Conceptualizing Smart City with Dimensions of Technology, People, and Institutions. In *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times* (College Park, Maryland, USA) *(dg.o '11)*. Association for Computing Machinery, New York, NY, USA, 282–291. https://doi.org/10.1145/2037556.2037602

Jayanth Raghothama, Vinutha Magal Shreenath, and Sebastiaan Meijer. 2016. Analytics on Public Transport Delays with Spatial Big Data. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*

PUC Minas

(Burlingame, California) *(BigSpatial '16)*. Association for Computing Machinery, New York, NY, USA, 28–33. https://doi.org/10.1145/3006386.3006387

Conclusion

Thanks!!