ATCLL MOBILE

Milestone 4 – Final Presentation

Project in Computer and Informatics Engineering – (LECI),
Department of Electronics, Telecommunications and Informatics,
University of Aveiro

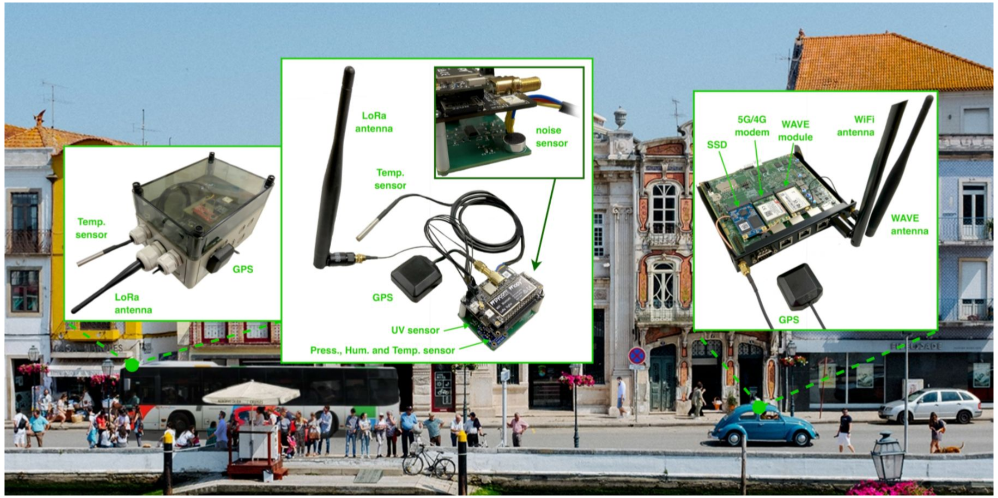
Alexandre Gago – 98123
Ana Rosa – 98678
Bernardo Kaluza – 97521
Filipe Silveira – 97981
Nuno Cunha – 98124
Pedro Lima – 97860

Supervisors:
Dr.a Susana Sargento
Dr. Pedro Rito
Dr. Miguel Luís

Context

- The Aveiro Tech City Living Lab (ATCLL) is composed of an advanced communications infrastructure and an urban data management platform in the city of Aveiro.

- Aveiro therefore contains several stations composed of various communication technologies installed in Smart Lamp Posts and buildings in the city. It also contains environmental sensors, mobility sensors (e.g., radars, LIDARs, video cameras).



Problem

- ❑ Enrich the Aveiro Tech City Living Lab (ATCLL) infrastructure in order to improve the lifestyle of citizens.

- ❑ The people in the city of Aveiro that use public transports have to rely on the static schedules and these schedules usually don't account for the delays. We can use our infrastructure to provide better schedule results, through prediction of the buses arrival.

- ❑ The streets of the city sometimes have a lot of traffic at key times of the day, we can use the radars of the infrastructure to inform the citizens of such traffic as a warning to avoid those streets.

Objectives

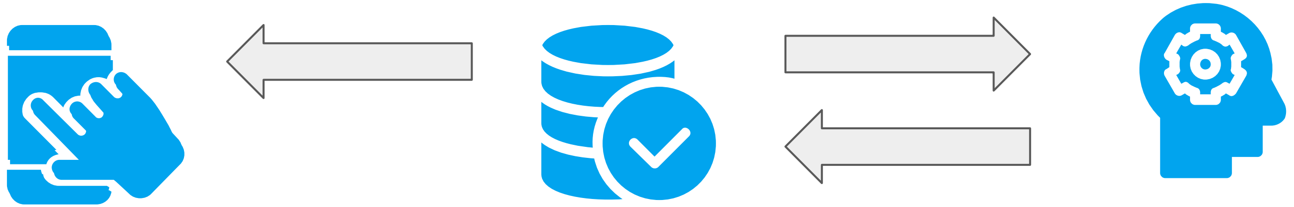
Bus tracking and arrival prediction

- ❑ Through the application it is possible to track the positions of buses in real time and its line, as well as the prediction of the time of arrival at the stops of the line.

Live Traffic

- ❑ Using the information provided by the radars, the application makes a traffic evaluation through a color system, and allows a visualization of the average speed of the vehicles on the roads.

Methodology



UI

Interface that allows the users to visualise the produced data

Server

Server hosted in IT that stores and sends data to the app

Line Detection, Prediction, Live Traffic

Responsible to produce the needed data to the app

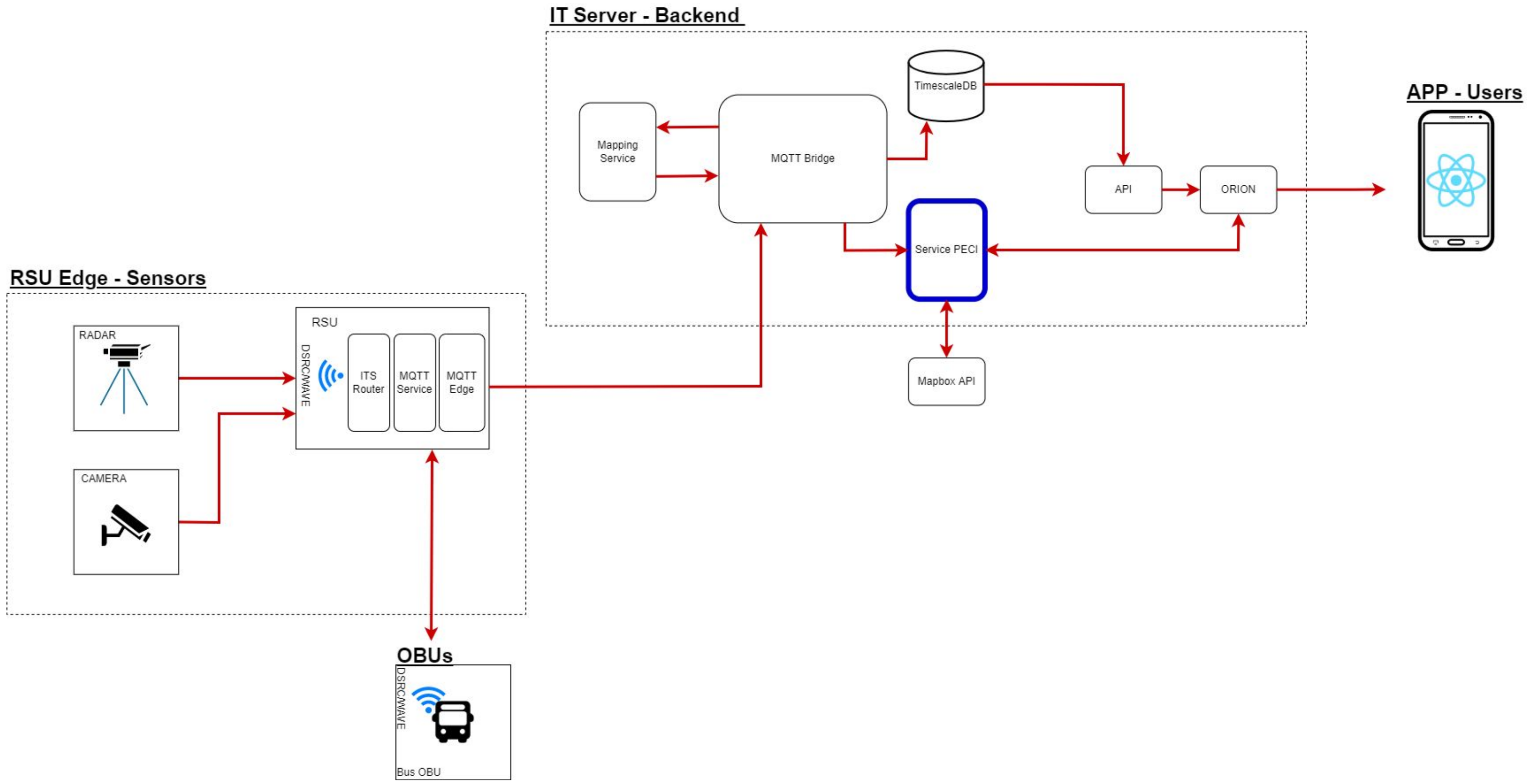
Architecture

- ❑ **MQTT (Message Queue Telemetry Transport)**
  - ❑ MQTT is a protocol that communicates data over low-bandwidth, unreliable networks. The protocol uses a publish/subscribe mechanism to exchange messages. The broker is the one that receives and sends the messages received from the publishers to the subscribers.

Architecture

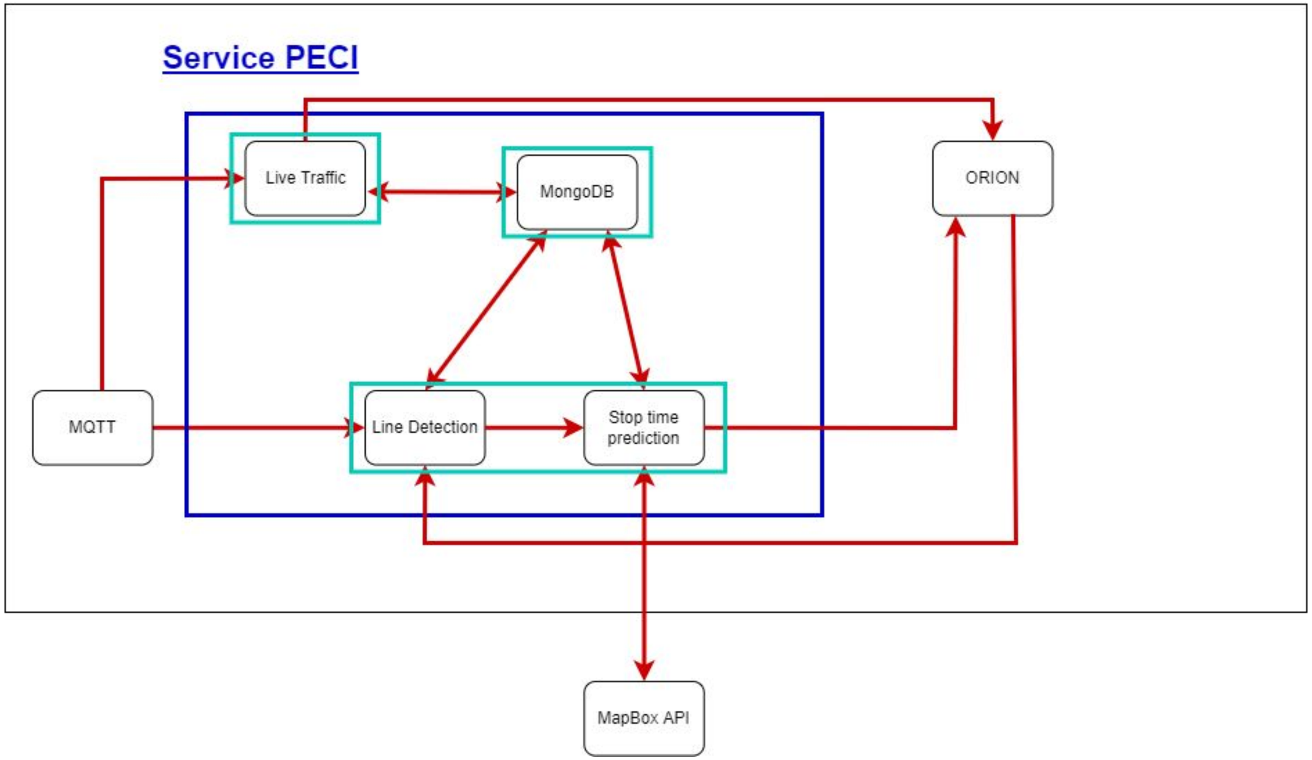
- ❑ **ORION**
  - ❑ Orion is a service that allows to collect information from various types of data (e.g. Traffic) and services (e.g. Transdev). The information is collected either historically or in real time through a URL to provide the type of data and the service required.

Architecture



Architectur e

IT Server - Backend



Architecture

Communication with the App

To communicate with the app, we had to create 2 topics in the orion API to differentiate our data with the main IT's server data.

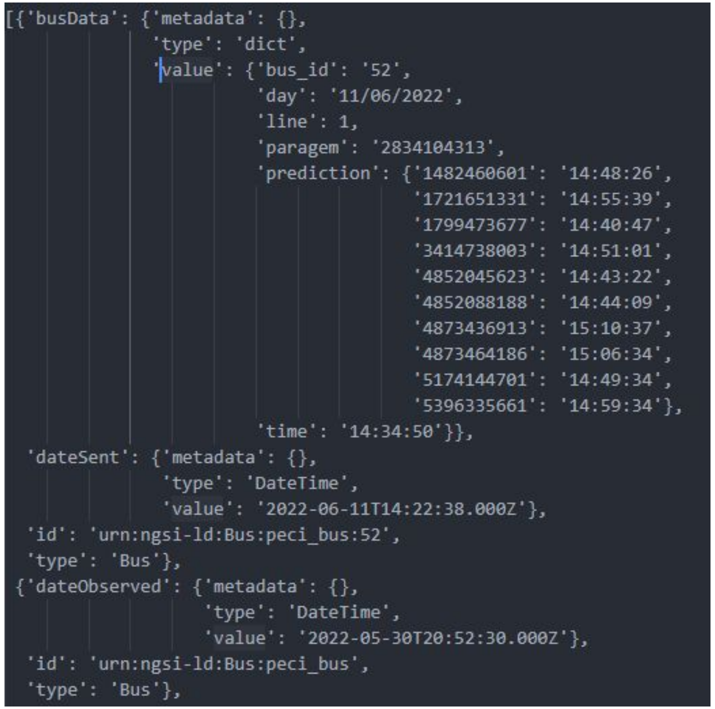
These were:

- bus_line
- aveiro_livetraffic

Architecture

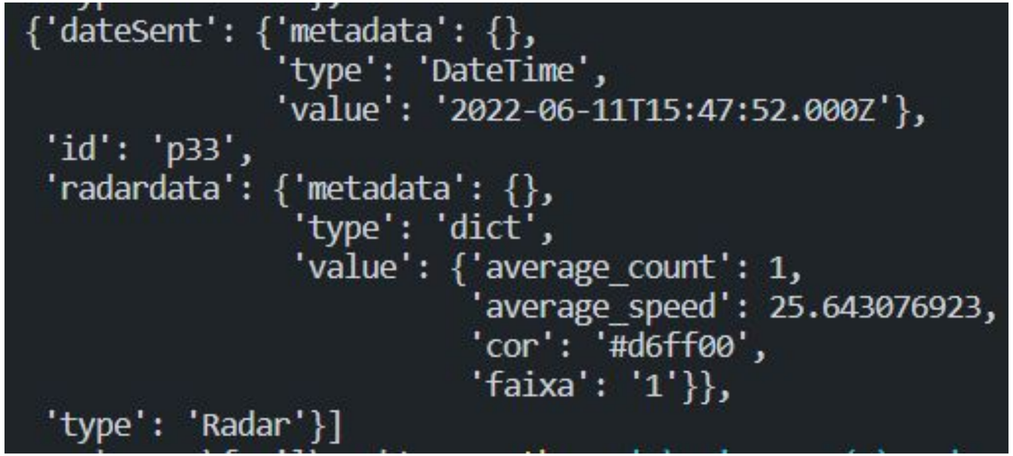
Communication with the App

Bus_line



```
[{'busData': {'metadata': {},
              'type': 'dict',
              'value': {'bus_id': '52',
                        'day': '11/06/2022',
                        'line': 1,
                        'paragem': '2834104313',
                        'prediction': {'1482460601': '14:48:26',
                                       '1721651331': '14:55:39',
                                       '1799473677': '14:40:47',
                                       '3414738003': '14:51:01',
                                       '4852045623': '14:43:22',
                                       '4852088188': '14:44:09',
                                       '4873436913': '15:10:37',
                                       '4873464186': '15:06:34',
                                       '5174144701': '14:49:34',
                                       '5396335661': '14:59:34'},
                        'time': '14:34:50'}},
  'dateSent': {'metadata': {},
               'type': 'DateTime',
               'value': '2022-06-11T14:22:38.000Z'},
  'id': 'urn:ngsi-ld:Bus:peci_bus:52',
  'type': 'Bus'},
 {'dateObserved': {'metadata': {},
                   'type': 'DateTime',
                   'value': '2022-05-30T20:52:30.000Z'},
  'id': 'urn:ngsi-ld:Bus:peci_bus',
  'type': 'Bus'},
```

Aveiro_livetraffic



```
{'dateSent': {'metadata': {},
              'type': 'DateTime',
              'value': '2022-06-11T15:47:52.000Z'},
 'id': 'p33',
 'radardata': {'metadata': {},
               'type': 'dict',
               'value': {'average_count': 1,
                         'average_speed': 25.643076923,
                         'cor': '#d6ff00',
                         'faixa': '1'}},
 'type': 'Radar'}]
```

Server

Server

Features:

- Use of Orion to obtain location (latitude and longitude) and ID of OBU (On Board Unit) of each bus from historical data in IT’s database.
- Use of MQTT to get bus information in real-time.
- Communication between the app and IT’s server using ORION.
- Host of a MongoDB database used to store data produced by the algorithms.

Server

Problem	Solution
ORION broker documentation not up to date and/or not functional.	Update of documentation by the group in IT.
The current and UTC timezone from the ORION timestamps did not match.	Before processing the data, the timezone was changed for each timestamp value in order to match Lisbon timezone.
The historical data in the ORION broker was not being updated correctly to the cloud.	This is being addressed in IT.
When receiving the MQTT data from buses, multiple messages were received when the bus was in radar range.	A filter was implemented, so that line detection starts only when the first bus enters the radar range.

Line Detection

Line Detection

Features:

- Information about lines and bus stops stored in JSON format.
- Confidence based algorithm that, given the positions of a bus in the previous hour, returns its possible line.
- Estimated time of Arrival prediction for the next stops of each bus in real time using the MapBox API.
- Store the algorithm’s output in a Mongo Database so the data collected can be used in future work.
- The line detection service is running in a docker container continuously in a IT server.

Line Detection

Problem	Solution
Old Code base non functional	Make our own line detection algorithm and Estimated time of Arrival
Although each bus has an identifier, each bus changes its line multiple times a day	The algorithm filters the received data of each bus
History data is not reliable	Adapt the line detection algorithm to be confidence-based
Some lines are nested inside others	Group those lines and treat them as one

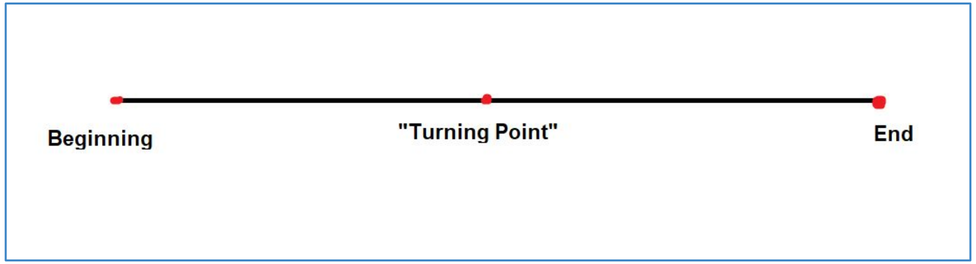
Line Detection

Problems found:

To calculate the ETA the direction of the bus had to be determined.

Solution:

- We filtered the History data in search of a stop that could assign the direction of the bus.
- If no such stop is found, count the number of times the current stop of the bus appears in the line, if it is only once there's no need to determine the direction.



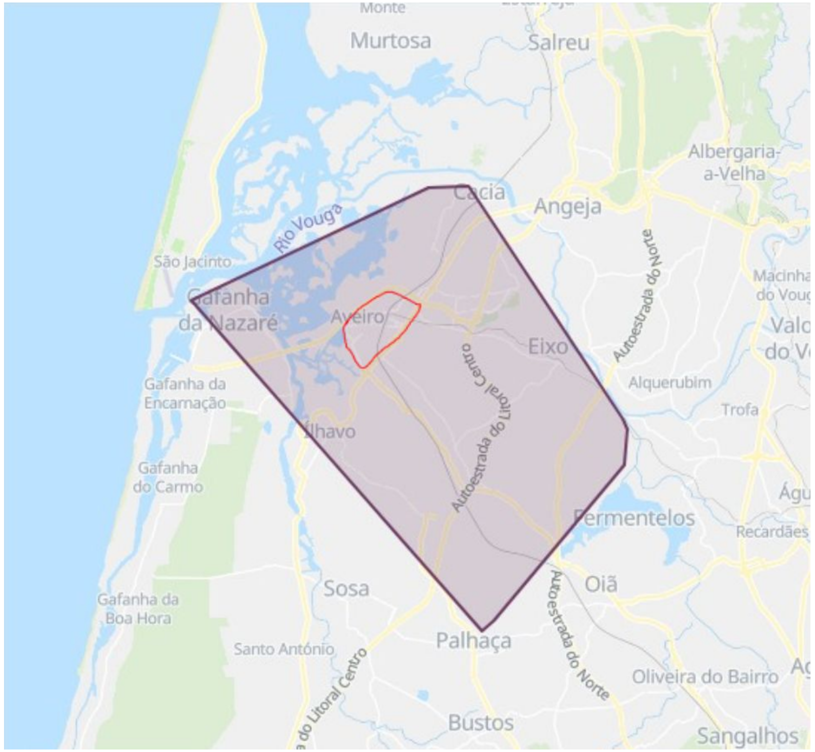
Line Detection

Problems found:

The coverage of the Smart Lamp posts and Murals is small compared to the coverage of the Aveiro Bus's lines. Furthermore most of the posts and Murals are in the middle of the city in which the bus lines merge and the same stop can belong to various different lines.

Solution:

- We had to get the history of the last hour of the bus to determine its lines because outside of the city the lines diverge and isolate themselves.



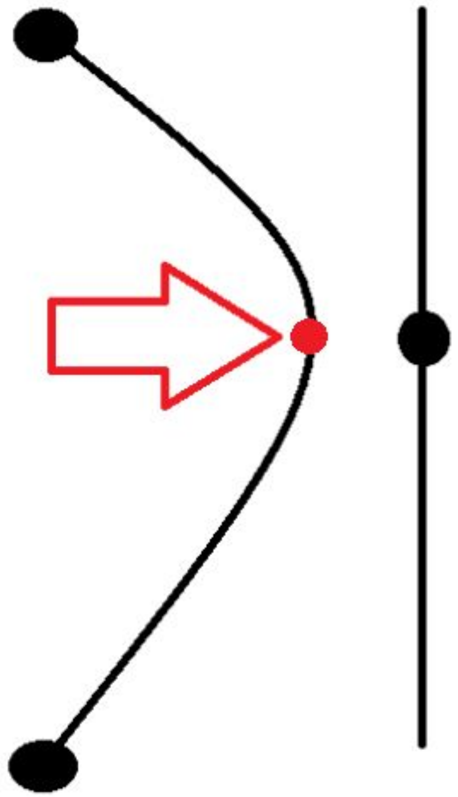
Line Detection

Problems found:

There are some stops that only belong to one line so we associated that line to the bus. However in situations like the one on the image, the algorithm would associate an incorrect line to the bus.

Solution:

We implemented a confidence system to choose the most probable line and assign it to the bus.



Live Traffic

Live Traffic

Features:

- Process real-time radar data and compare it to the average of the previous day.
- Allocate a color to the processed data, this color is posted in Orion to be retrieved by the App
- Mongo Database storing the traffic data each day so data can be used in future projects.

UI

UI

Framework used - React Native

Features:

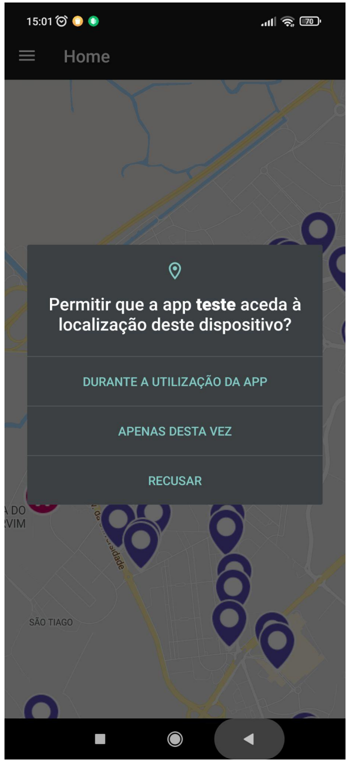
- Main map updated with the buses.
- Lines menu showing stops of line with estimated time of arrival and respective bus in a map.
- Receiving data from main IT server and our own server and generating access token.
- Bus schedule menu.
- Live traffic map.
- About page.

UI

Problem	Solution
Problem with receiving the data and saving it, even though the request was being made.	We used javascript Promises so that we would only store data when the Promise confirmed the arrival of data.
Due to the amount of requests done, the app would get progressively slower in certain screens.	We switch the places where the requests were made to reduce extra unwanted requests.
Some of the component libraries we tried to use had problems during installation or didn’t work as intended.	We researched various libraries and we also made backups with relative frequency to ensure the app stayed stable.

DEMO

DEMO

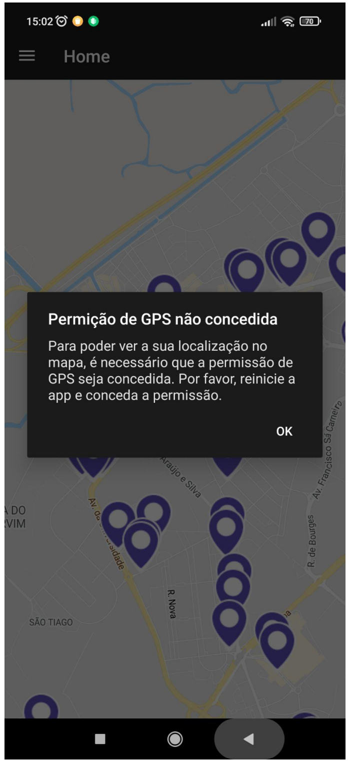


Permitir que a app **teste** aceda à localização deste dispositivo?

DURANTE A UTILIZAÇÃO DA APP

APENAS DESTA VEZ

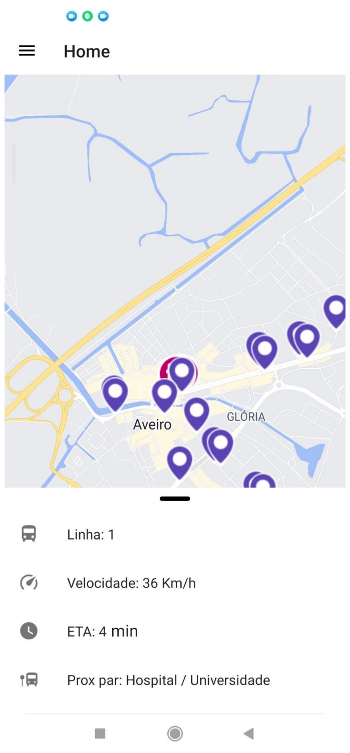
RECUSAR



Permição de GPS não concedida

Para poder ver a sua localização no mapa, é necessário que a permissão de GPS seja concedida. Por favor, reinicie a app e conceda a permissão.

OK

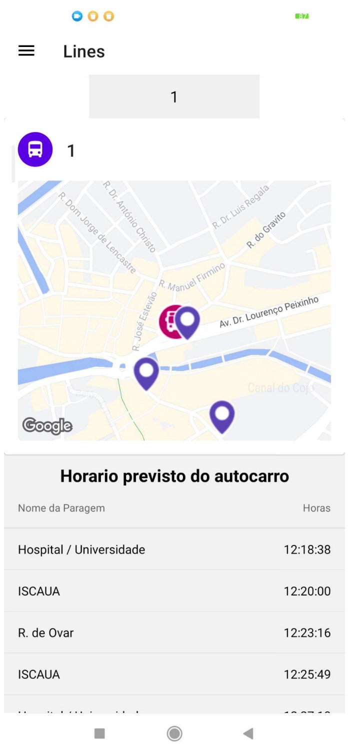


Linha: 1

Velocidade: 36 Km/h

ETA: 4 min

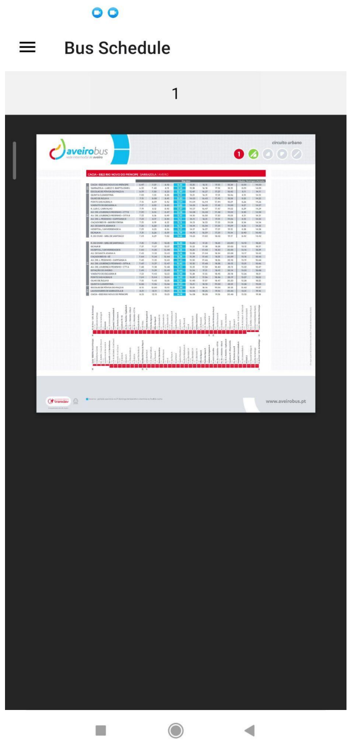
Prox par: Hospital / Universidade



Horario previsto do autocarro

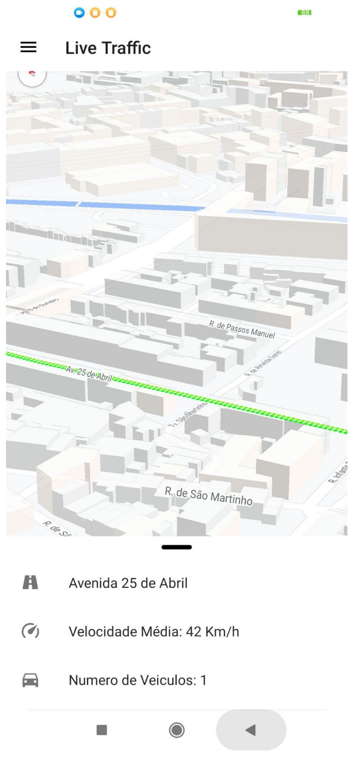
Nome da Paragem	Horas
Hospital / Universidade	12:18:38
ISCAUA	12:20:00
R. de Ovar	12:23:16
ISCAUA	12:25:49

DEMO



Bus Schedule

1

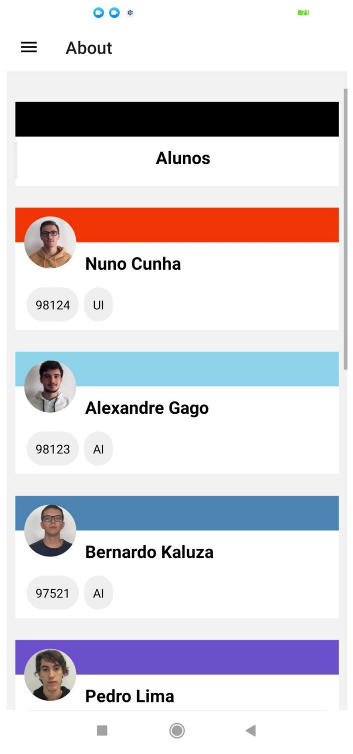


Live Traffic

Avenida 25 de Abril

Velocidade Média: 42 Km/h

Numero de Veiculos: 1



About

Alunos

Nuno Cunha

98124 UI

Alexandre Gago

98123 AI

Bernardo Kaluza

97521 AI

Pedro Lima

Results Discussion

- We were able to achieve our two main objectives, which include the Bus tracking and arrival prediction and the Live traffic, thus adding new features to the ATCLL infrastructure already implemented and deployed.

Vision for the future

How the product will be used:

- ❑ This will be used though an smartphone Android/IOS app by any citizen of the city of Aveiro that uses buses as his main way of transportation or that wants to know the current traffic of the city

What can be improved:

- ❑ The line detection, in the future, will be an API provided by Transdev
- ❑ For the Line Detection, in the future we will have more active smart lamp posts; therefore, our prediction will be improved and there's always room to improve the prediction algorithm
- ❑ The live traffic feature will also be improved with the installation of more radars, but contrary to the other features, this one must be done through an update of the app.