# Fundação Getulio Vargas

## Estruturas de Dados e Algoritmos

---

**Final Project - Mini Waze**

Prof. Jorge Poco

**Participants:**
Each group will have a maximum of 3 students.

**Goal:**
In this project you will implement a Mini-Waze, i.e. a web application that calculates the shortest path between two points in a city using two algorithms. The result should be an application similar to a Waze or Google Maps, but reducing the functionalities.

**Dataset:**

In this project we will use the road system of the city of Rio de Janeiro. The graph with the road system will be obtained with the OSMnx library, where we can specify the name of the city and it returns the graph directly. The command used will be the following:

```
RJ = ox.graph_from_place('Rio de Janeiro', network_type='drive')
```

Notice that this graph has 69,630 nodes and 170,484 edges. All your projects must work with this graph and this configuration without exception, it is not valid to make simplifications of this graph.

If you wish to earn extra points, you can use the graph with this command:

```
RJ = ox.graph_from_place('Rio de Janeiro', simplify=False, network_type='drive')
```

Which has 256,929 nodes and 487,035 edges.

**Algorithms**
You will have to implement two efficient versions of the Dijkstra and A* algorithms. The backend part has to be implemented in C++. All data structures (e.g. priority queues) must be implemented by you, no libraries or public code is allowed. This is the most important part of the project, and that's why we will evaluate the efficiency of the algorithm. To get the maximum grade you must return the result in less than 1s.

**Web-based Interface**

You will also have to implement a web interface (e.g., in html and javascript) with the following functionalities:

- Display a map and allow the user to select two points (origin and destination).
- Communicate with the server by sending the two coordinates and receiving the shortest path.
- The interface must draw the shortest path on the map.
- We must have a menu where we can select which algorithm to use for the shortest path.

This is the minimum functionality to have. However, you can gain an extra point for adding new functionality. For example:

- Write the address in a textfield and have it communicate with Google Maps to convert an address to GPS coordinate.
- Calculate the fastest path instead of the shortest path. OSMNx allows you to get the average speed for each street segment, this can help you with this problem.

**Products:**

You will have to upload a PDF file with the following name **project-grupoX.pdf**. I will assign you a group number (listed at the end of this file). Only one of the members of the group should send the work, remember, I only want one pdf file in the system. This document should be between 4 and 6 pages long. With the following content:

- Title
- Abstract
- Link to a GitHub repository where your source code is located. You will have to create a TAG with the following name "FINAL". I will verify the date of creation of the tag is June 28 until midnight otherwise it will not grade your work. Moreover, this repository should have a README, where it clearly explains how to run your program, datasets, etc. If I can't run your program, then it won't be evaluated either. Those instructions should also be in this document.
- Link to a video explaining your whole project and showing me how it works. You can take inspiration from the following video. You can record using zoom or any tool that is simpler for you. The objective is not the quality of the recording, what interests me is how you explain your project to me in a clear and simple way.
- Description: What did you do? What data structure did you use? Did you do any pre-processing?
- Results: Do some experiments to show me the complexity of your solution. Memory space? How much disk space did you use? How long does the queries take? Make some charts showing the results.
- Limitations of your work and possible improvements as future work
- Conclusions
- Distribution of work: Indicate clearly which parts of the work were done by each of the members of the group.

**Grading:**

| | Score | Comentarios |
|---|---|---|
| Did you meet the goal? | 7 pts | If you accomplish the goal.<br>- Dijkstra's algorithm (2 pontos)<br>- A* search algorithm (2 pontos)<br>- Query: less than a second (1 pontos)<br>- Minimum Web interface (2 pontos) |
| Presentation | 1 pt | Each one will have to present what you did, I want to see how each one performs in presenting your work. Also, if I ask questions and you do not answer, you will not get this point. This score is individual. |
| Video | 1 pt | Watch the video that was sent to you previously. It is very important to know how to sell your work. |
| Report | 1 pt | The clarity in the document. Even if you have sent the document, it is possible that you get 0 if it does not comply with the minimum quality. |
| Additionals | 2 pts | Until 2 points, depending in the functionalities |

**Technical details:**

- The core of the search engine (i.e. the data structure and algorithms) must be implemented in C++.
- If you don't have a web interface, then the queries should be in the console.
- For the interface, you can only use HTML, Javascript, and CSS. The fewer requirements you have, the better. But remember to be explicit in the document on how to run your program.
- In case you have done the web interface, remember that there must also be an executable that allows me to run queries per console.
- Your problem must run in the following way:

```
> ./shortest
> … Loading index, done!
> Which algorithm to use?
> [0] Dijkstra
> [1] A*
> 0
> Enter the origin: -22.939885645157645, -43.17952481161723
> Enter the destination: -22.954798201237953, -43.194277749662845
> The shortest path using Dijkstra is 9999 meters (0.02 seconds)
> [SHOW THE COORDINATES OF THE SHORTEST PATH]
```

## Groups:

| Grupo | Estudantes |
|---|---|
| 1 | Fernanda |
| | João Lucas |
| | João Pedro |
| 2 | Lívia |
| | Luiz Luz |
| | Ari Oliveira |
| 3 | Gianlucca |
| | Juliana |
| | Raphael Levy |
| 4 | Breno |
| | Tulio |
| | Joanne |
| 5 | Juliane |
| | Caio Lins |
| | Ademir |
| 6 | Emanuel |
| | Vinícius |
| | Denner |
| 7 | Andre Ribeiro |
| | Carlos César |
| | Erlon Kelvim |
| 8 | Erick |
| | Germano |
| | Patrick Saul |
| 9 | Lucas |
| | Maisa |
| | Victor |
| 10 | Gabriel Machado |
| | Eduardo Junqueira |
| | Tiago |
| 11 | João Alcindo |
| | Sávio |
| | André Costa |
| 12 | Wellington José |
| | Daniel Csillag |
| | Rafael Gomes |