

SCC0630 | Inteligência Artificial

# Trabalho 1: Busca de Doadores Compatíveis de Sangue

Leonardo Gueno Risetto	13676482
Lucas Lima Romero	13676325
Luciano Gonçalves Lopes Filho	13676520
Marco Antonio Gaspar Garcia	11833581
Thiago Kashivagui Gonçalves	13676579

2025



# Sumário

## 0.1 Sobre o projeto

a. Objetivos

b. Contexto

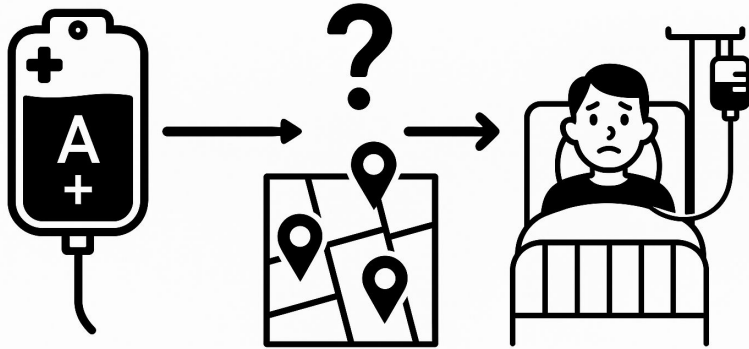
## 0.2 Buscas implementadas

## 0.3 Funcionamento do projeto



## Objetivo: sistema em tempo real

Qual o problema que desejamos solucionar...



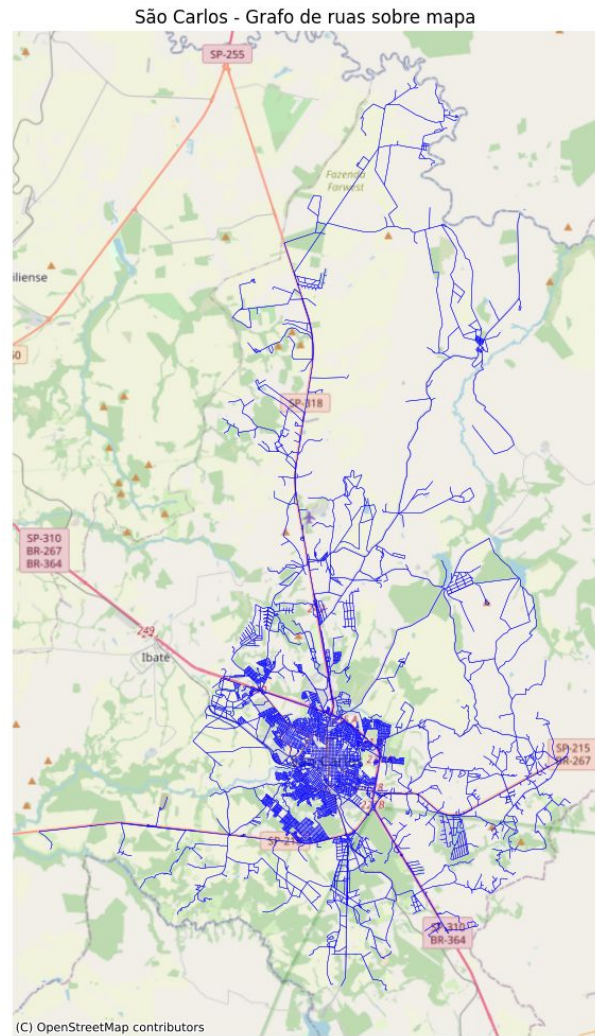
Qual a nossa solução e como ela pode ajudar...



A ideia é que exista uma **base de dados em tempo real** com as informações de hemocentros para a busca de **sangue compatível** com o do usuário...

## Contexto: modelagem da cidade em um grafo

Modelando o problema: gerando um grafo da cidade



# Localização e Validação

## Definindo os Hemocentros e Localização do Usuário

```
from utils.helper_functions import BancoDeHemocentros
```

```
# Instanciando o banco de hemocentros  
H = BancoDeHemocentros(hemocentros, G.graph)
```

```
# Exemplo, estoque de bolsas de sangue um único hemocentro  
H.consultar_estoque(hemocentros[0])
```

Python

```
{'A+': 0, 'A-': 27, 'B+': 13, 'B-': 0, 'AB+': 4, 'AB-': 10, 'O+': 44, 'O-': 0}
```





# Localização e Validação

Filtragem dos hemocentros com sangue compatível

```
# Definindo o tipo de sangue do usuário e filtrando os HCs válidos
blood_type = "O-"
hcs_validos = H.hemocentros_validos(blood_type)

print(hcs_validos)
```

Python

```
[528971858, 1013811199, 1014884447]
```

Usuário e Hemocentros Válidos (com Zoom)



# Sumário

0.1 Sobre o projeto

**0.2 Buscas implementadas**

a. BFS

b. A\*

0.3 Funcionamento do projeto



## Justificativa da escolha das buscas

### Busca Não-Informada

---

- A **DFS** pode ser muito ineficiente perante o nosso problema...
- Para garantir um caminho menor, escolhemos a **BFS**

### Busca Informada

---

- As Buscas **Hill-Climbing** e **Best-First** podem ficar presas em ruas sem saída...
- Por isso, precisamos garantir uma rota confiável com **A\***



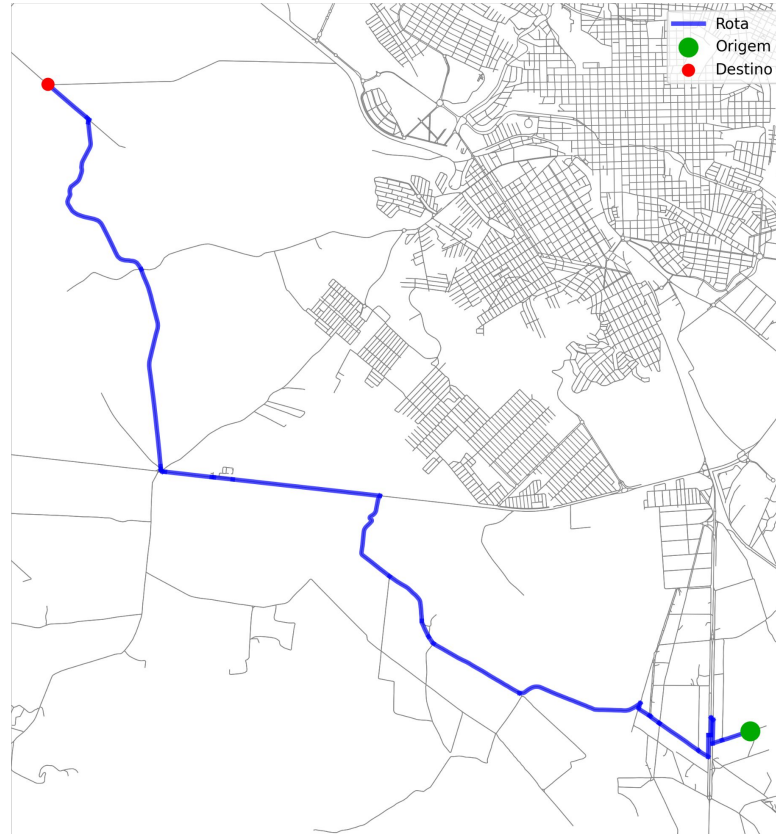
## Busca em Largura (BFS) como algoritmo de busca cega...

### Sobre o BFS

---

- No BFS **não somos capazes de dizer** se de fato estamos percorrendo a **menor distância!**
- O BFS encontrará o menor caminho em termos de **arestas percorridas...**
- Tem **execução mais rápida** que a do  $A^*$

## Exemplo de Rota encontrada pela BFS



## Exemplo de Rota encontrada pela BFS

Análise dos resultados do BFS

---

	BFS	Rota mais curta (Gabarito)
Número de nós visitados	34	87
Distância em metros	18246.66	17332.67

- ⚠ BFS encontrou uma rota **diferente** da mais curta em distância — esperado!

## A\* como busca informada...

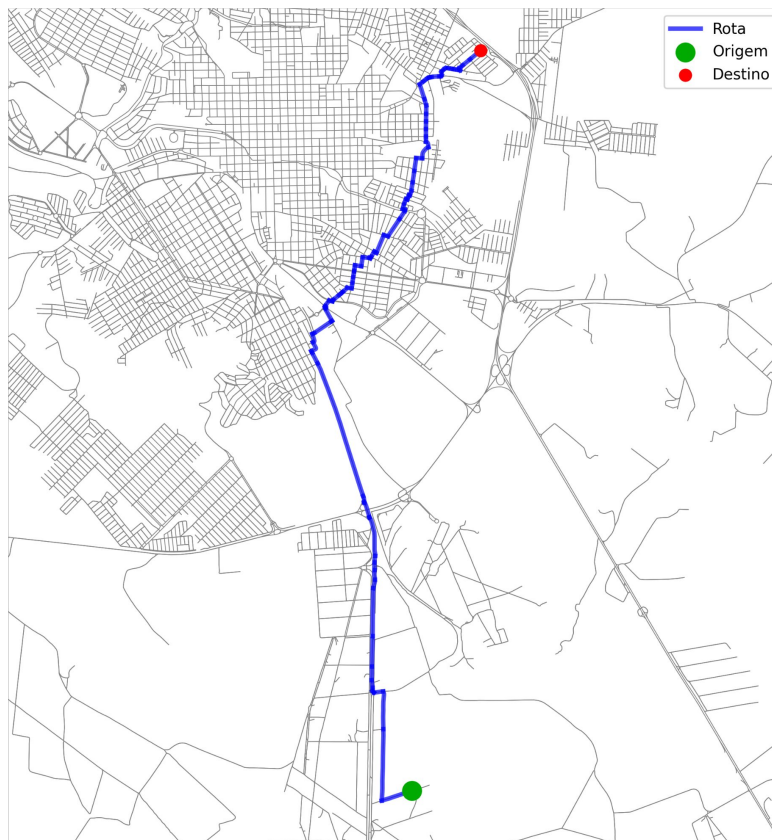
Sobre o algoritmo A\*

---

- Utiliza uma **heurística** para guiar a exploração dos nós.
- Avalia os caminhos com base na função  $f(n) = g(n) + h(n)$ .
- **SEMPRE** encontra o caminho de menor custo se a heurística for admissível.



## Exemplo de Rota encontrada pelo A\*



Em nosso trabalho implementamos uma Busca utilizando A\* como algoritmo de busca informada...

Análise dos resultados do A\*

---

	A*	Rota mais curta (Gabarito)
Número de nós visitados	99	99
Distância em metros	12321.37	12321.37

-  A\* encontrou **a mesma rota** da menor distância (esperado).

# Sumário

0.1 Sobre o projeto

0.2 Buscas implementadas

**0.3 Funcionamento do projeto**

