

# Seminário Final de **Análise Preditiva BIG 008**

Prof Kendji Eduardo Wolf

## **Vehicle Routing Problem**

Evander Siqueira

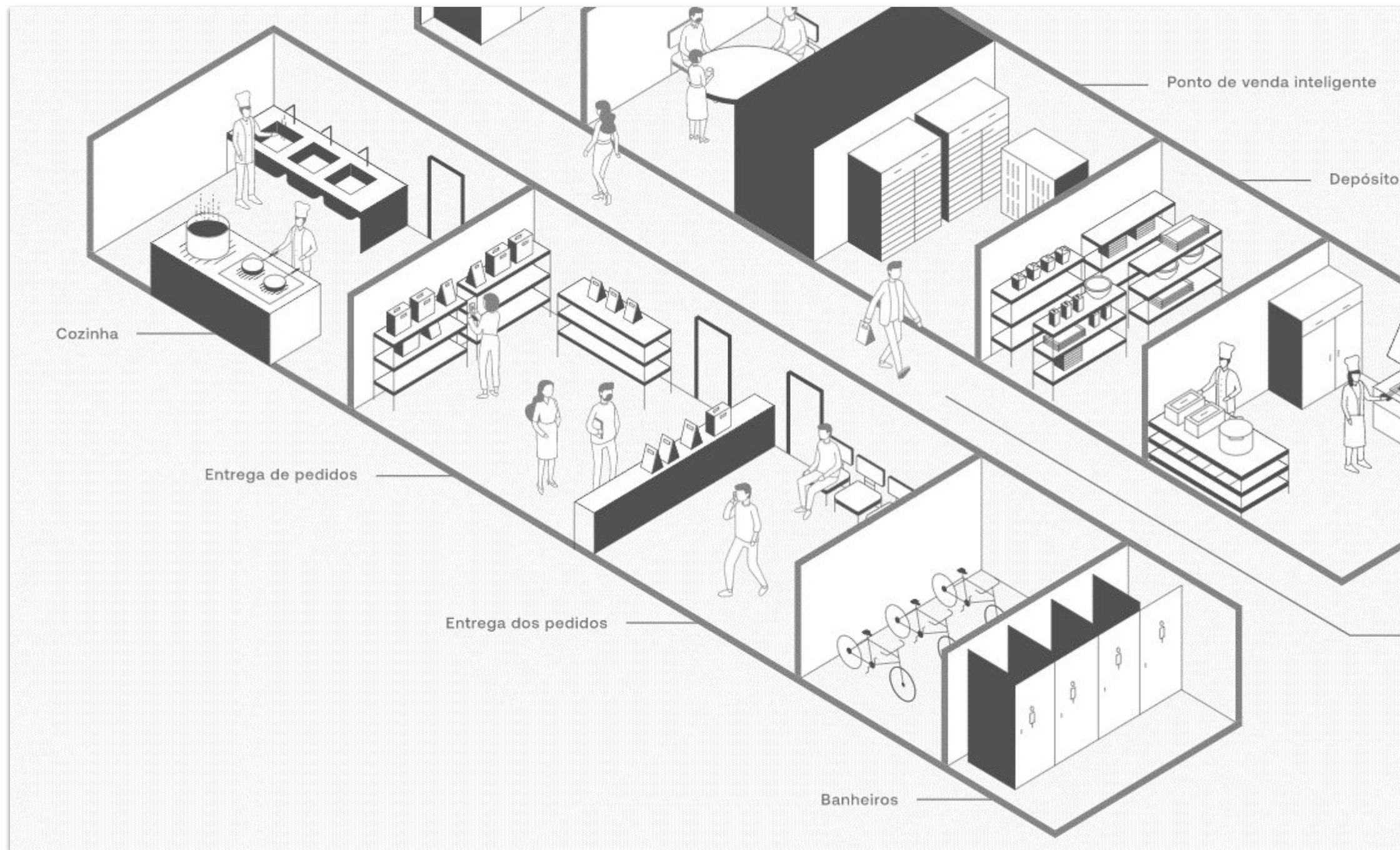
Thiago Pereira

Tiago Galinari





# introdução



## Deliv. Eat

A startup Deliv.Eat é uma dark kitchen que está iniciando suas operações gradualmente em São Paulo.

Sua base inicial fica na região do Jaguaré, por ser uma região estratégica e de fácil acesso às marginais e bairros como Pinheiros e Lapa, mas também como as regiões Sul e Centro, de onde vem a grande maioria de seus pedidos, via app.

Por atuar, em sua maioria, com entregas de refeições nas regiões citadas acima, os entregadores contratados são motoboys ou bikers.

Por hora, a Deliv.Eat pretende operar apenas nessas regiões para realizar testes e gradualmente ir aumentando sua capacidade operacional e tecnológica.

Mas o objetivo central é expandir sua operação a ponto de oferecer todo serviço, inclusive logístico, para outras empresas que queiram se estabelecer no ramo de dark kitchen e entregas.

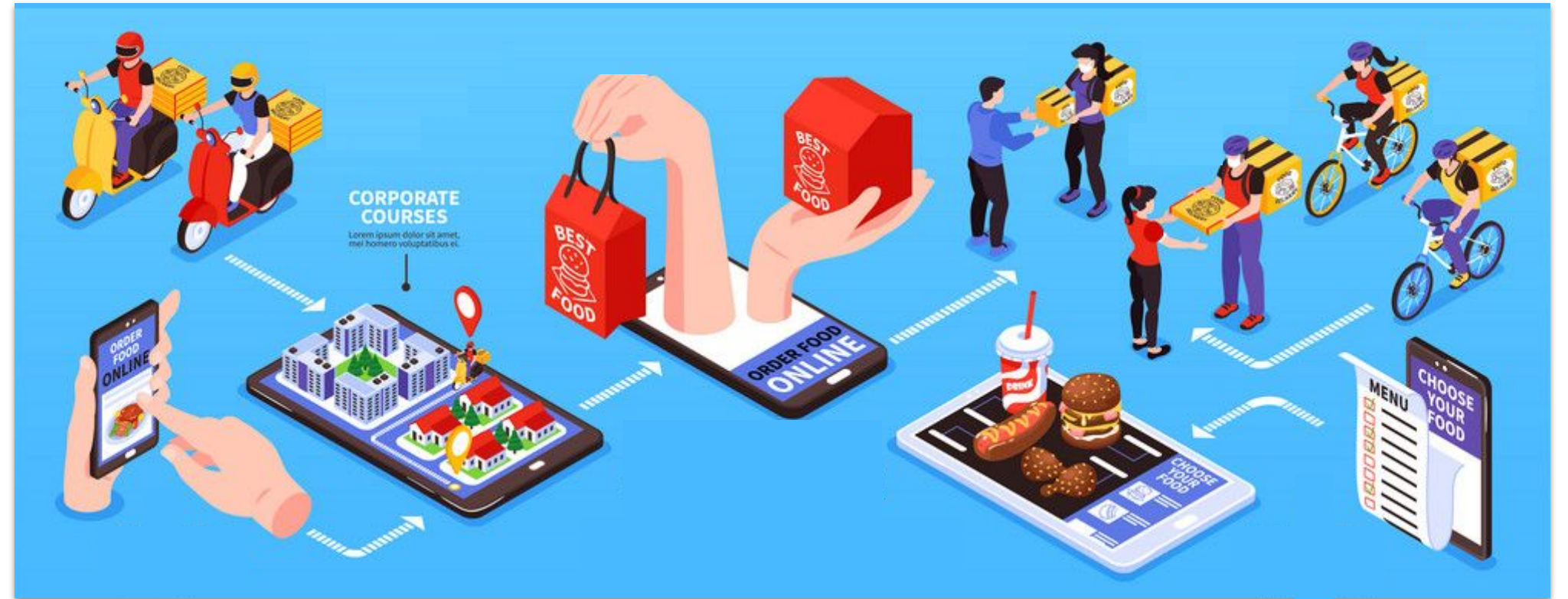


# contextualização

Com o crescimento do volume das entregas a serem realizadas, surgiu a necessidade de otimizar as entregas para os clientes, melhorando assim a dinâmica dos entregadores para atender a todos os pedidos no menor tempo possível, como também otimizar os custos.

Atualmente, os entregadores realizam mais de uma entrega devido ao crescente volume de pedidos. Porém, fica a cargo dele escolher o melhor caminho e a sequência das entregas.

Como também, está ficando cada vez mais complexo definir qual o entregador deve efetuar determinada entrega.



Desse modo, surgiu a necessidade de fazer a roteirização desses pedidos a fim de:

- Melhorar o fluxo de entregas;
- Otimizar a rota do entregador:
  - Melhor caminho;
  - No menor tempo;
  - No melhor custo benefício
- Melhorar a disponibilidade dos entregadores;

# definição do modelo

Para otimizar o processo de entregas, utilizamos um modelo de otimização combinatória baseado na técnica de teoria de grafos, chamado problema do caixeiro viajante.

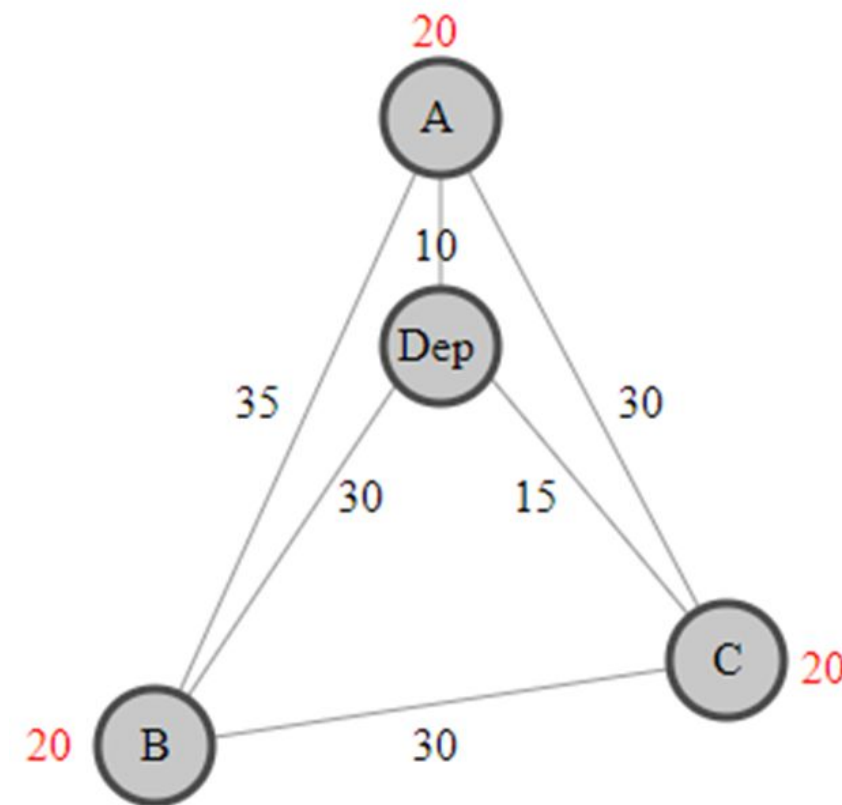
O problema do caixeiro viajante consiste em se descobrir o número de rotas que torna mínima a viagem total para um determinado número  $n$  de pontos de entrega, através de uma notação fatorial do tipo  $R(n) = (n - 1)!$

Além da definição da melhor rota para realizar uma determinada entrega, temos o problema de restrição de capacidade, que também deve ser considerado no modelo.

# definição do modelo

Para solução deste problema foi utilizado o OR-Tools, um software do Google de código aberto para otimização combinatória.

O algoritmo utiliza o conceito de penalidade, que é ação de dispensar uma determinada visita em função da demanda de entrega superior à capacidade, baseado na maior distância a ser percorrida.





# proposta (mvp)

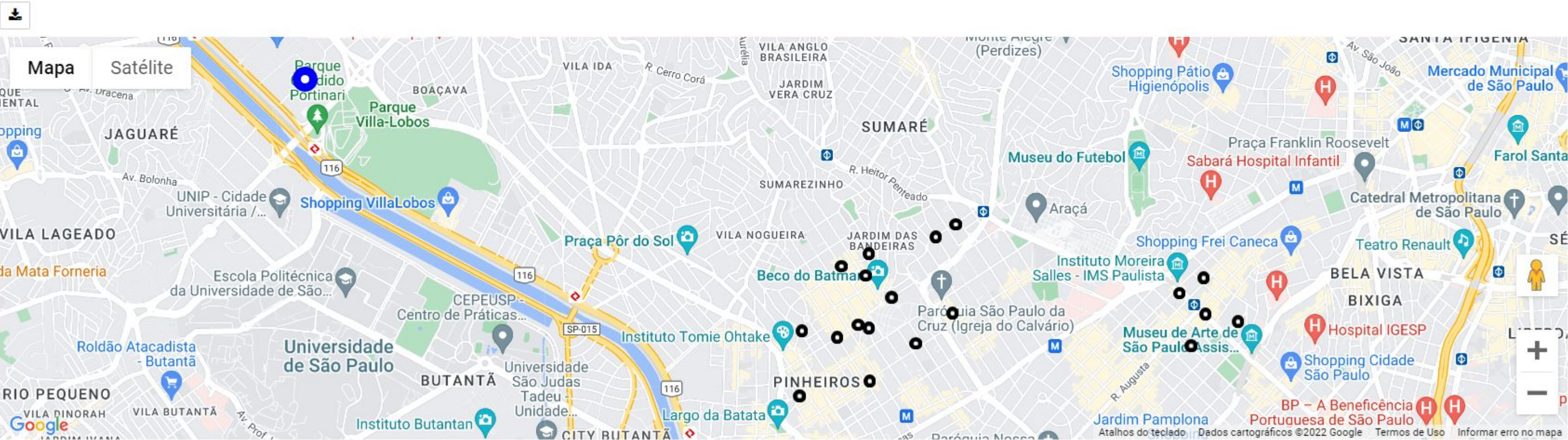
[VRP Entregas.ipynb - Colaboratory \(google.com\)](#)

Visualizando os pontos de entrega e do armazém

Agora, vamos mostrar o armazém e todas as entregas a serem realizadas

```
[ ] mapa_de_pontos = fvrp.gerar_mapa_pontos('Armazém e Pontos de Entrega', localizacao_base, nome_base, localizacao_entregas, nome_entregas)
mapa_de_pontos
```

Armazém e Pontos de Entrega



**Obrigado!**