

Relatório Técnico de Implementação e Deploy

Visão Geral do Projeto

Este projeto é uma API desenvolvida em Flask que estima o preço de corridas com base em dados de origem e destino, utilizando a API de rotas do [OpenRouteService \(ORS\)](#) e um modelo de machine learning treinado com dados reais. O sistema também realiza **monitoramento de desempenho** (tempo de resposta, uso de CPU e memória) e retorna essas informações via endpoint.

Componentes Principais

- **Linguagem:** Python 3.13
- **Framework:** Flask
- **Machine Learning:** Scikit-learn (LinearRegression)
- **Monitoramento:** `psutil`
- **Bibliotecas Adicionais:** pandas, requests, matplotlib (dependências completas no requirements)
- **API Externa:** OpenRouteService (direções e distância entre pontos)
- **Ambiente:** Docker + Railway

Estrutura para Deploy

```
/api
├── api.py
├── ia.py
├── Procfile
├── Dockerfile
└── requirements.txt
```

Conteúdo do Dockerfile

```
FROM python:3.13-slim
WORKDIR /app
COPY . /app
RUN pip install --upgrade pip && pip install -r requirements.txt
EXPOSE 5000
CMD ["gunicorn", "-b", "0.0.0.0:5000", "api:app"]
```

Subindo no Railway

1. Vá até <https://railway.app> e clique em "New Project".
2. Escolha "Deploy from GitHub repo" e conecte seu repositório.

3. Certifique-se de que o projeto tem um Dockerfile na raiz.
4. Railway irá automaticamente iniciar o build.
5. Aguarde o deploy ser concluído.

Teste da API

Após o deploy, você receberá uma URL pública

Teste com ferramentas como Postman

Exemplo:

GET - <https://insightmob.up.railway.app/dados>

GET - <https://insightmob.up.railway.app/monitoramento>

POST - <https://insightmob.up.railway.app/calcular>

JSON Para a rota calcular

```
{
  "lonOri": -46.684403,
  "latOri": -23.589674,
  "lonDes": -46.655881,
  "latDes": -23.561414
}
```