

## Relatório de implementação

Para a implementação do modelo na nuvem, nós optamos por subir na AWS, em uma instância de EC2, utilizando os requisitos padrões que encaixavam no plano gratuito:

Sistema Operacional	Ubuntu Server 24.04 LTS, amd64
Tipo de Instância	t2.micro
RAM	1 GB
Armazenamento	8 GB

Além disso, foi necessário configurar o grupo de segurança da instância, dando acesso a protocolos TCP para a porta 5000 de todos os IPs. Isso para que o cliente consiga fazer uma requisição ao servidor.

Nossa API foi desenvolvida em python, com a lib Flask. Ela conta com dois endpoints:

- GET / : Mensagem para verificar se a API está sendo reconhecida. Retorna uma mensagem “Hello”
- GET /calcularCorrida : é o método usado para calcular os preços previstos pelo modelo. Retorna uma lista de JSON, cada uma representando uma categoria de corrida.

Chamadas à API devem ser feitas usando:

```
/calcularCorrida?origem=<origem>&destino=<destino>
```

O Dockerfile anexado junto a esta entrega importa uma imagem python 3.10, adicionando para a imagem da api o arquivo “meu\_modelo\_treinado.joblib”, que se refere ao modelo. Além disso, o arquivo “requirements.txt” é lido e as bibliotecas necessárias são instaladas.

A imagem gerada foi publicada no DockerHub, e então importada na instância do EC2. Uma vez importada, basta utilizar o comando:

```
docker run -d -p 5000:5000 vitortokunaga/api-previsao
```

Assim, a API Flask pode ser acessada acessando a URL:

```
http://ip-do-EC2:5000/
```

Caso todos os passos tenham sido feitos corretamente, a mensagem “Hello” deve aparecer na página de resposta.