

## Projeto Backend Taxis

---

### Objetivo

Implementar o backend de um aplicativo de smartphone que mostra um mapa com os taxistas ativos da 99Taxis. O sistema deve assumir que tem diversos taxistas cadastrados e permitir atualizar o status desses taxistas, consultar status deles e encontrar taxistas em uma dada área. Toda comunicação deve ser feita com JSON.

### Endpoints:

1. Grava a posição atual e estado de um taxista.
  - a. Parâmetros:
    - i. latitude
    - ii. longitude
    - iii. driverId (id único de taxista no sistema)
    - iv. driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas.
  - b. Resposta:
    - i. Apenas status code HTTP
    - c. Exemplo:
      - i. `POST /drivers/8475/status '{"latitude":-23.60810717,"longitude":-46.67500346,"driverId":5997,"driverAvailable":true}'`
2. Lista os taxistas ativos dentro de uma retângulo geográfico.
  - a. Parâmetros:
    - i. sw: Ponto extremo sul, extremo oeste do retângulo, no formato "latitude,longitude". Ex: -23.612474,-46.702746
    - ii. ne: Ponto extremo norte, extremo leste do retângulo, no formato "latitude,longitude". Ex: -23.589548,-46.673392
  - b. Resposta: Um array em formato json, de um objeto com atributos:
    - i. latitude
    - ii. longitude
    - iii. driverId (id único de taxista no sistema)
    - iv. driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas. Neste endpoint retorna sempre true.
  - c. Exemplo:

```
GET /drivers/inArea?sw=-23.612474,-46.702746&ne=-23.589548,-46.673392
[{"latitude":-23.60810717,"longitude":-46.67500346,"driverId":5997,"driverAvailable":true},{"latitude":-23.59065045044675,"longitude":-46.68837101634931,"driverId":63446,"driverAvailable":true},{"latitude":-23.60925506,"longitude":-46.69390415,"driverId":1982,"driverAvailable":true},{"latitude":-23.599871666666665,"longitude":-46.680903333333326,"driverId":9106,"driverAvailable":true},{"latitude":-23.59492613,"longitude":-46.69024011,"driverId":16434,"driverAvailable":true}]
```

3. Estado de um taxista. Recebe o id de um taxista e retorna os dados dele:
  - a. Parâmetros:
    - i. driverId
  - b. Resposta:
    - i. latitude
    - ii. longitude
    - iii. driverId (id único de taxista no sistema)
    - iv. driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas.
  - c. Exemplo:
    - i. GET /drivers/73456/status  
'{"latitude":-23.60810717,"longitude":-46.67500346,"driverId":5997,"driverAvailable":true}'

Objetivos Bonus:

1. Subir a aplicação "na nuvem" e enviar o link para testarmos os endpoints.
2. Descrever algoritmo ótimo de procura dos taxistas.
3. Autenticação nos endpoints, explicando a solução.
4. Criar outros endpoints que possibilitem a mesma funcionalidade mas com outro fluxo.
5. Endpoint Cria taxista:
  - a. Parâmetros:
    - i. name
    - ii. carPlate: placa do carro
  - b. Resposta:
    - i. Apenas status code HTTP
  - c. Exemplos:
    - i. POST /drivers  
'{"name":"Pedro","carPlate":"RPC-9999}"'