

## Desafio Parcial BACK-END

Nome:

### Pergunta:

1. Com base no banco de dados (**locadora\_veiculos**) estruturado no Desafio Parcial No 6:

1.1 Utilize o gerenciador de requests “Postman” para implementar as rotas e crie um **CRUD** que permita gerenciar as tabelas:

- **veiculos**
- **marcas**
- **categorias**

### Requisitos:

- **Rotas CRUD para "veiculos"**
  - **GET /veiculos:** Retorna todos os veículos.
  - **GET /veiculos/{id}:** Retorna um veículo pelo ID.
  - **POST /veiculos:** Adiciona um novo veículo ao banco de dados.
  - **PUT /veiculos/{id}:** Atualiza as informações de um veículo pelo ID.
  - **DELETE /veiculos/{id}:** Remove um veículo do banco de dados.
- **Rotas CRUD para "marcas"**
  - **GET /marcas:** Retorna todas as marcas.
  - **GET /marcas/{id}:** Retorna uma marca pelo ID.
  - **POST /marcas:** Adiciona uma nova marca ao banco de dados.
  - **PUT /marcas/{id}:** Atualiza as informações de uma marca pelo ID.
  - **DELETE /marcas/{id}:** Remove uma marca do banco de dados.
- **Rotas CRUD para "categorias"**
  - **GET /categorias:** Retorna todas as categorias.
  - **GET /categorias/{id}:** Retorna uma categoria pelo ID.
  - **POST /categorias:** Adiciona uma nova categoria ao banco de dados.
  - **PUT /categorias/{id}:** Atualiza as informações de uma categoria pelo ID.
  - **DELETE /categorias/{id}:** Remove uma categoria do banco de dados.

---

### Instruções:

#### 1. Configuração do Backend

- Configure um servidor com **Node.js** e **Express**.
- Conecte o backend ao banco de dados PostgreSQL (**locadora\_veiculos**) usando o módulo **pg**.

## 2. Criação das Rotas

- Implemente cada uma das rotas mencionadas acima, organizando-as em arquivos separados para cada entidade (**veiculos**, **marcas**, **categorias**).

## 3. Testes no Postman

- Para cada rota, configure requests no Postman com os métodos correspondentes (**GET**, **POST**, **PUT**, **DELETE**).
- Adicione exemplos de payloads JSON para os requests **POST** e **PUT**.

## 4. Validação e Erros

- Adicione validações para garantir que os dados inseridos estão no formato correto.
- Implemente tratamento de erros para responder com mensagens apropriadas em casos de falhas (ex.: ID inexistente).

## 5. Desafios Extras (Opcional)

- Adicione paginação na rota **GET /veiculos**.
- Implemente buscas por atributos nas tabelas, como **modelo** na tabela **veiculos** e **nome** na tabela **marcas**.
- Inclua uma funcionalidade para filtrar veículos por faixa de preço ou ano.