



Desenvolver uma API REST com um método de criação, edição, remoção lógica e listagem com filtro e paginação.

A API deve possuir autenticação em todos os métodos.

A API deve possuir uma documentação bem explicativa de como funciona cada método com os possíveis parâmetros de entrada e os tipos de retorno.

A API deve ser publicada, configurada e disponibilizada em uma URL temporária.

Os modelos dessa API devem ser espelhados nas duas tabelas presentes no arquivo 'db-structure.sql' em anexo.

Criação:

- Deve receber como parâmetro todas as colunas da tabela 'service\_orders'.
- É importante fazer validações nos parâmetros de entrada a fim de evitar erros de inserção no banco de dados e inconsistência de dados.

Edição:

- Deve receber como parâmetro apenas as colunas da placa do veículo e valor de estadia para edição do registro na tabela 'service\_orders'.
- É importante fazer validações nos parâmetros de entrada a fim de evitar erros de inserção no banco de dados e inconsistência de dados.

Remoção lógica:

- Deve receber como parâmetro o ID de um registro na tabela 'service\_orders'.

Lista:

- Deve listar todas as colunas presentes na tabela 'service\_orders'.
- A coluna 'userId' deve ser relacionada na tabela 'users' e retornar o nome do usuário.
- Deve conter um mecanismo de paginação.
- Cada página deve ter apenas 5 itens.
- Deve ter um filtro para buscar por placa de veículo.

**Requisitos:**

- Deve usar o framework PHP Laravel.
- Deve usar um banco de dados MySQL.
- Deve possuir autenticação
- Deve estar disponível numa URL temporária.
- Deve possuir uma documentação bem explicativa.
- Uma requisição de criação de ordem de serviço.
- Uma requisição de edição de ordem de serviço.
- Uma requisição de remoção lógica da ordem de serviço.
- Uma requisição de listagem com filtro e paginação.
- Tanto os parâmetros de entrada quanto os retornos devem ser transitados apenas no formato JSON.

**Entrega:**

O desenvolvedor tem **dez dias corridos** para retornar a solução, contando a partir do recebimento do teste.

A solução deve ser **disponibilizada em qualquer repositório Git com acesso público e a API concluída deve estar disponível em uma URL temporária.**