

Teste Java

API Rest para gerenciamento de veículos.****

Esta é uma avaliação básica de código.

O objetivo é conhecer um pouco do seu conhecimento/prática de **RESTful, Spring e Java**.

Representação de produto(objeto):

```
{  
  "id": 12345,  
  "chassi": "9BWZZZ377VT004251",  
  "name": "Lamborghini Urus",  
  "manufacturer": "Automobili Lamborghini S.P.A.",  
  "year": 2021,  
  "color": "white",  
  "status": "ACTIVATED",  
  "placa": "OPA0148"  
}
```

Status do objeto:

- * ACTIVATED
- * DEACTIVATED
- * RESERVED
- * RENTED

Crie endpoints para as seguintes ações:

1. Cadastrar Carro
2. Listar Carros
3. Ver Detalhes
4. Trocar Status
5. Deleção Lógica

Requisitos

- Caso o carro já exista em memória e ao ser criado com o mesmo ****Chassi**** ou ****Placa****, uma exceção deverá ser lançada.
- Ao atualizar um carro, o antigo deve ser sobrescrito com o que está sendo enviado na requisição.
 - A requisição deve receber o ID para efetuar a atualização.
- Caso o status passado no body não exista, uma exceção deverá ser lançada.
- A listagem de carros deverá ser agrupada por fabricante, e o agrupamento deverá ser ordenado por nome e ano (do menor até o mais atual).
- O sistema deverá apresentar a opção de realizar consultas com filtro: nome do veículo, ou fabricante, ou ano de fabricação.
- Qualquer pesquisa que não retorne um resultado, deverá ser lançado uma exceção.
- Montar desenho da solução utilizando Draw.io ou ferramenta similar

Configuração para o Projeto

Para o projeto, devesse utilizar:

- Java 11
- Spring Boot
- Maven

Dicas

- Documente sua API
- Sinta-se a vontade para utilizar docker
- Testes são sempre bem-vindos 😊
- Os produtos podem ficar em memória, não é necessário persistir os dados