

Projeto Final Spring Boot

Resumo

Vamos montar um sistema de vendas de veículos no estilo **Webmotors**, onde poderemos consultar carros de acordo com a montadora (marca), modelo, Km e preço.

Nosso sistema também deve informar:

- Se o carro é zero Km ou usado
- O ano de fabricação e ano modelo
- Se possui opcionais (não precisa de muitos itens)
 - Airbag
 - Ar condicionado
 - Direção hidráulica
 - ...

Cadastros

Precisamos das APIs para cadastrar e atualizar os dados básicos do sistema.

- Montadoras
- Veículos (e modelos)
- Opcionais

Criação e atualização

Para enviar dados no corpo da requisição (request body), precisamos usar a anotação **@RequestBody**.

Buscas

Além dos cadastros das entidades, precisamos de API para buscar os carros de acordo com alguns filtros.

Todos os carros

API que retorna todos os carros cadastrados.

OPCIONAL: Usar paginação usando o Pageable.

Todos os carros de uma montadora

API que retorna todos os carros cadastrados para uma determinada montadora.

OPCIONAL: Usar paginação usando o Pageable.

Todos os carros usados

API que retorna apenas os carros usados.

OPCIONAL: Usar paginação usando o Pageable.

Todos os carros novos

API que retorna apenas os carros zero Km.

OPCIONAL: Usar paginação usando o Pageable.

Todos os carros num range de quilometragem (Ex: de 0 até 30mil Km)

API que retorna todos os carros cadastrados no range de Km determinado.

OPCIONAL: Usar paginação usando o Pageable.

Todos os carros num range de preço (Ex: de \$50k até \$100k)

API que retorna todos os carros no range de valor determinado.

OPCIONAL: Usar paginação usando o Pageable.

Paginação

Podemos usar o Pageable do próprio JPA para implementar paginação e de quebra resolvemos o problema de não retornar listas.

A documentação da JPA pode ser encontrada [neste](#) link.