



#### Programação Java

# Requisito 6

// Especificações de Requisitos

#### Requisitos US: ml-crud-produto-01

User Story	1
Representação JSON	2
Contratos referentes a User Story	3

#### Importante:

As histórias de usuários são contadas do ponto de vista do representante do armazém de acordo com suas necessidades. Os serviços são expostos a partir do depósito de distribuição para serem consumidos pelo solicitante. Os contratos referem-se à História do Usuário.

Este requisito é livre e é realizado individualmente, partindo do pressuposto de que deve agregar valor à aplicação, utilizando dados existentes ou novos que se considerem necessários. Dito isso, o participante fica livre para fazer um aplicativo separado que se comunica com o primeiro já criado, se desejar. Esclarecimento: você precisa criar um novo aplicativo.





## CRUD da entidade produto

# User Story

User Story Code: ml-crud-produto-06 User Story Name: Realize o CRUD da entidade produto	Horas estimadas
<b>COMO</b> Representante <b>QUERO</b> poder cadastrar, fazer um update, apagar e fazer diferentes tipos de consultas dos produtos do armazém <b>PARA</b> ter um melhor controle sobre os produtos e me planejar melhor.	10
CENÁRIO 1: Um produto é cadastrado  DADO QUE Um produto é cadastrado	
E o produto não exista	
<b>E</b> a categoría existe <b>QUANDO</b> o Representante insere um novo produto	
ENTÃO ele cria um novo produto na base de dados	
<b>E</b> o novo produto tem um id asignado <b>E</b> o produto pode ser alterado	
<b>E</b> o produto pode ser apagado	
<b>E</b> o representante pode ter um relatório de um determinado produto	
VALIDAÇÃO	

- O produto novo n\u00e3o deve estar cadastrado.
- O produto novo deve ter uma categoria válida.
- Nas buscas, update e delete o produto já deve existir.





## Representação JSON

#### Response

```
POST - CADASTRO NOVO PRODUTO: {
 "id": 42,
 "name": "Amendoas",
 "volumen": 0.9,
 "category": "REFRIGERATED"
}
GET - BUSCAR POR NOME DO PRODUTO: {
 "id": 1,
 "name": "rucula",
 "volumen": 0.1,
 "category": "FRESH"
}
GET - RELATÓRIO POR NOME DO PRODUTO: {
 "Nome do produto": "bisteca suína",
 "Quantidade inicial de produtos": 133,
 "Total produtos vendidos": 55,
 "Total produtos disponíveis em estoque": 33,
 "Total produtos que venceram": 45
```





}

**UPDATE**: Update do produto feito com sucesso

**DELETE: Produto deletado com sucesso** 

http://localhost:8080/api/v2/fresh-products/rucula

# Contratos referentes a la User Story

HTTP	Plantilla URI	Descripción	US-code
POST	/api/ <mark>v2</mark> /fresh-products	Cadastre um novo produto com os dados corretos e receba um corpo sem id.	ml-create-product -06
GET	/api/ <mark>v2</mark> /fresh-products/{name}	Faça uma busca do produto pelo nome e retorne un 404 quando ele não é encontrado	ml-findByName- product-06
GET	/api/v2/fresh-products/product/ ?name={name}	Gere um relatório do produto para saber a quantidade de estoque deste produto, a quantidade inicial, quantas unidades foram vendidas e quantas venceram.	ml-do-repoByPro ductName-produ ct-06
PUT	/api/ <mark>v2</mark> /fresh-products	Altere um produto sem a necessidade de passar o id dele, só mandar as novas informações. Retorna um 404 caso o produto não exista.	ml-update-produ ct-06
DELETE	/api/ <mark>v2</mark> /fresh-products/{id}	Apaga o produto com o id mandado, retorna um 404 caso o produto não exista.	ml-[código us]-06

Observação: