**Generation Brasil – Projeto Integrador**

**Projeto:** Plataforma WEB – ColaborArt.

**Tipo:** E-commerce.

**Representantes:** Alana Rodrigues Teixeira / Bruno Falango / Gustavo Broch /

Laíse Lima Galvão de Souza / Vitor Carneiro Borela.

***Back-End***

**Framework:** Spring Boot

**Link:** <https://start.spring.io/>

**Versão:** 2.4.1

**Projeto:** Maven Project

**Linguagem:** Java

**Versão:** 8

**Dependências Adicionadas:**

* Spring Web
* Spring Boot DevTools
* Spring Data JPA
* Validation
* MySQL Driver

**Banco de Dados:** MySQL

***Model***

**Produto - Atributos:**

* **private long idProduto;**

Número de identificação do produto dentro da *tb\_produto*.

(Exemplo: 1, 2, 3, etc.)

* **private String descricaoProduto;**

Descrição do produto dentro da *tb\_produto*.

(Exemplo: Camiseta Personalizada, com estampa Star Wars)

* **private String tamanho;**

Tamanho do Produto.

(Exemplo: Camisas: PP, P, M, G, GG / Caneca: 250ml)

* **private BigDecimal valor;**

Valor do Produto.

(Exemplo: R$22,50)

* **private boolean disponivel;**

Disponibilidade do Produto.

(Exemplo: true ou false)

* **private int estoque;**

Quantidade de produto no estoque.

(Exemplo: 20)

* **private String urlProduto;**

URL das imagens hospedadas.

(Exemplo: www.imgur.com/starwars)

* **private String nome;**

Nome do produto.

(Exemplo: Camiseta Star Wars Yoda)

* **private Categoria categoria;**

Relacionamento Produto/Categoria

**Produto – Métodos:**

* ***getters / setters* dos atributos do *model*;**

**public** **long** getIdProduto() {

**return** idProduto;

}

**public** **void** setIdProduto(**long** idProduto) {

**this**.idProduto = idProduto;

}

**public** String getDescricaoProduto() {

**return** descricaoProduto;

}

**public** **void** setDescricaoProduto(String descricaoProduto) {

**this**.descricaoProduto = descricaoProduto;

}

**public** String getTamanho() {

**return** tamanho;

}

**public** **void** setTamanho(String tamanho) {

**this**.tamanho = tamanho;

}

**public** BigDecimal getValor() {

**return** valor;

}

**public** **void** setValor(BigDecimal valor) {

**this**.valor = valor;

}

**public** **boolean** isDisponivel() {

**return** disponivel;

}

**public** **void** setDisponivel(**boolean** disponivel) {

**this**.disponivel = disponivel;

}

**public** **int** getEstoque() {

**return** estoque;

}

**public** **void** setEstoque(**int** estoque) {

**this**.estoque = estoque;

}

**public** String getUrlProduto() {

**return** urlProduto;

}

**public** **void** setUrlProduto(String urlProduto) {

**this**.urlProduto = urlProduto;

}

**public** Categoria getCategoria() {

**return** categoria;

}

**public** **void** setCategoria(Categoria categoria) {

**this**.categoria = categoria;

}

**public** String getNome() {

**return** nome;

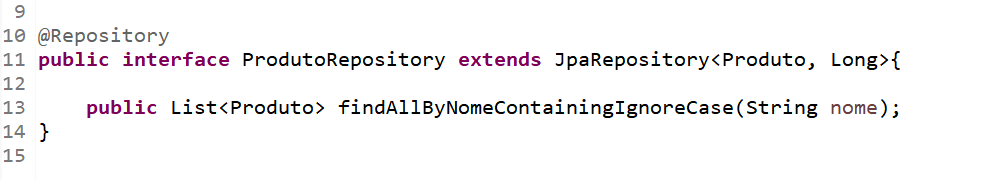
}

**public** **void** setNome(String nome) {

**this**.nome = nome;

}

***Repository***



**Interface – ProdutoRepository**

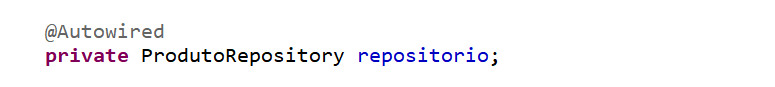
Criação do repositório do produto, estendendo da Classe ***JpaRepository***, onde recebe a **Produto** Model e o tipo primitivo **Long**.

**Método – Query Method**

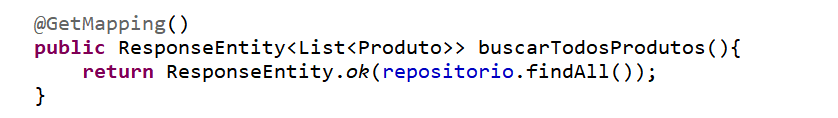
* findAllByNomeContainingIgnoreCase

Lista todo tipo de produto pelo nome presente na *tb\_produto*, de acordo com a texto passado pelo usuário.

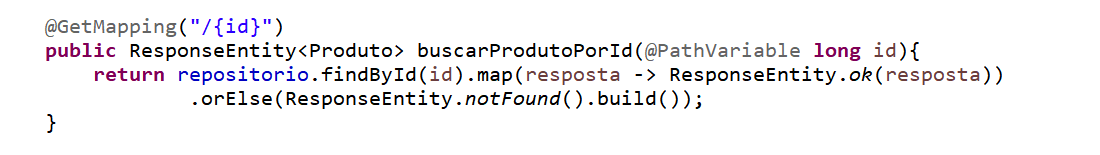
***Controller***



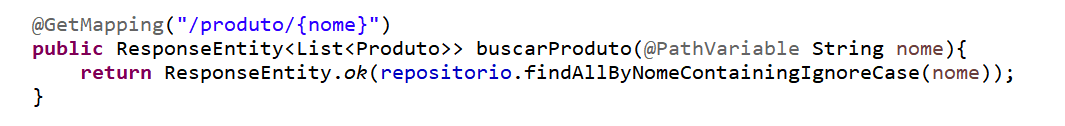
Atributo para implementação do repositório, através da injeção de dependências(@Autowired).

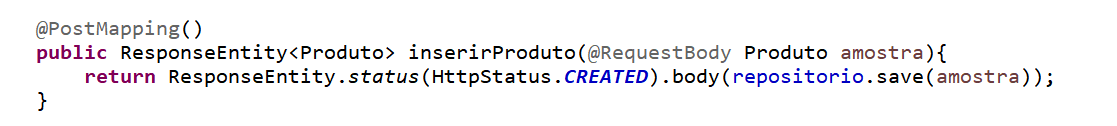


Retorna todos os produtos encontradas no banco de dados.

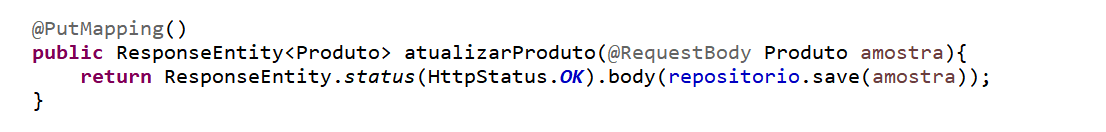


Retorna o Produto de acordo com Id fornecido pelo usuário.

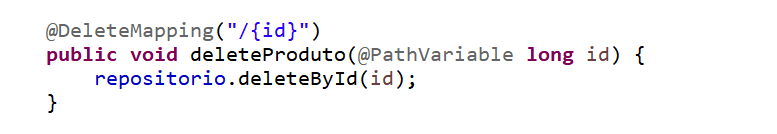
Retorna valores dos Produtos, de acordo com o nome do produto fornecido (Query Method).



Insere informações do Produto no banco de dados.



Atualiza informações do Produto no banco de dados, de acordo com o Id.

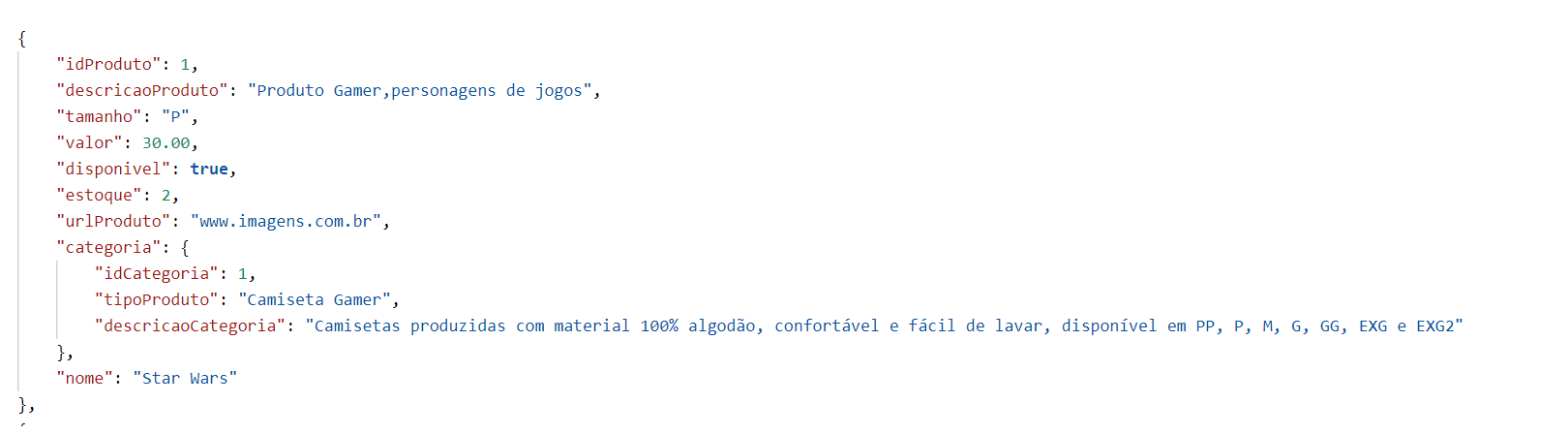


Apaga informações do Produto no banco de dados.

***CRUD – Postman***

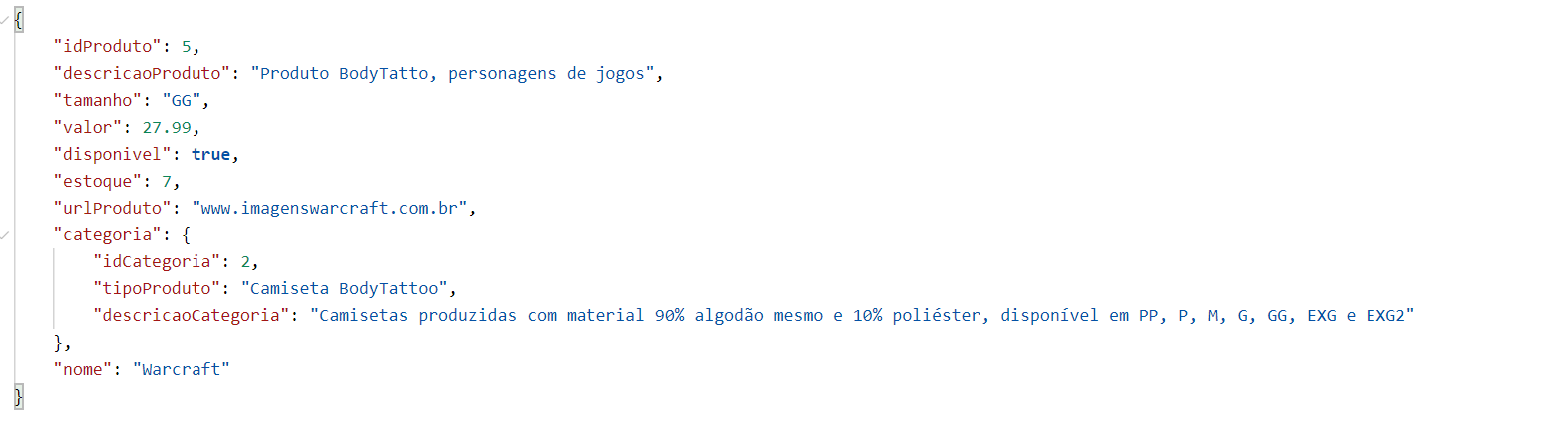
**GET**

<http://localhost:8080/produto>

****

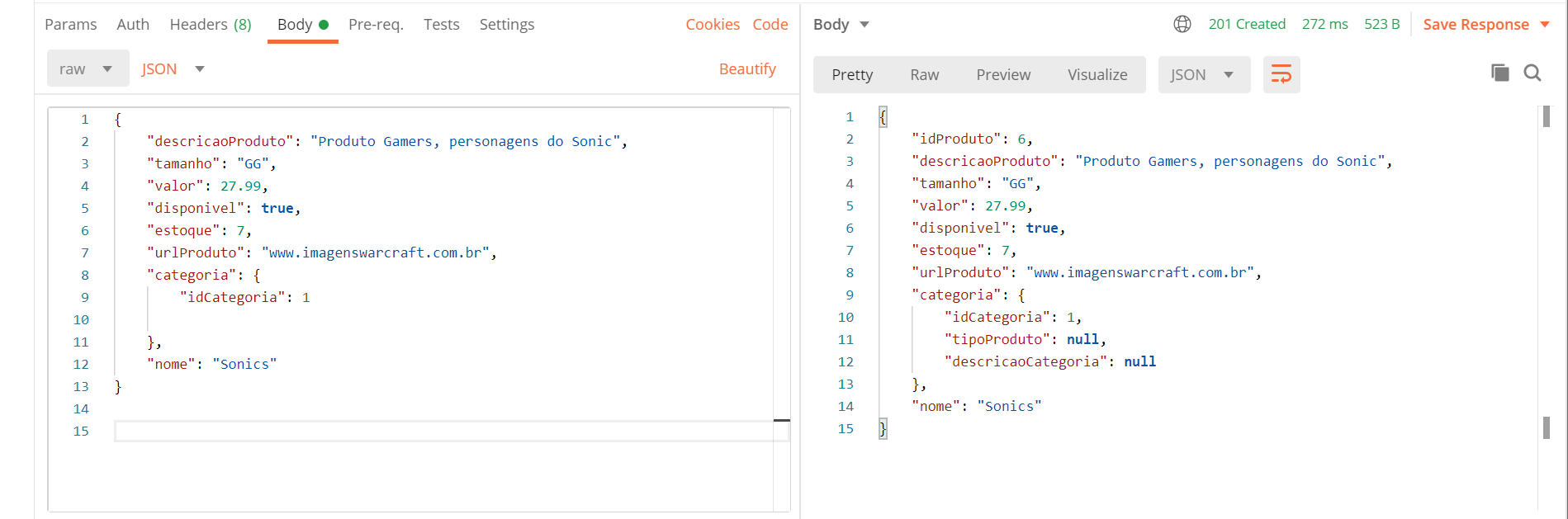
**GET**

<http://localhost:8080/produto/5>



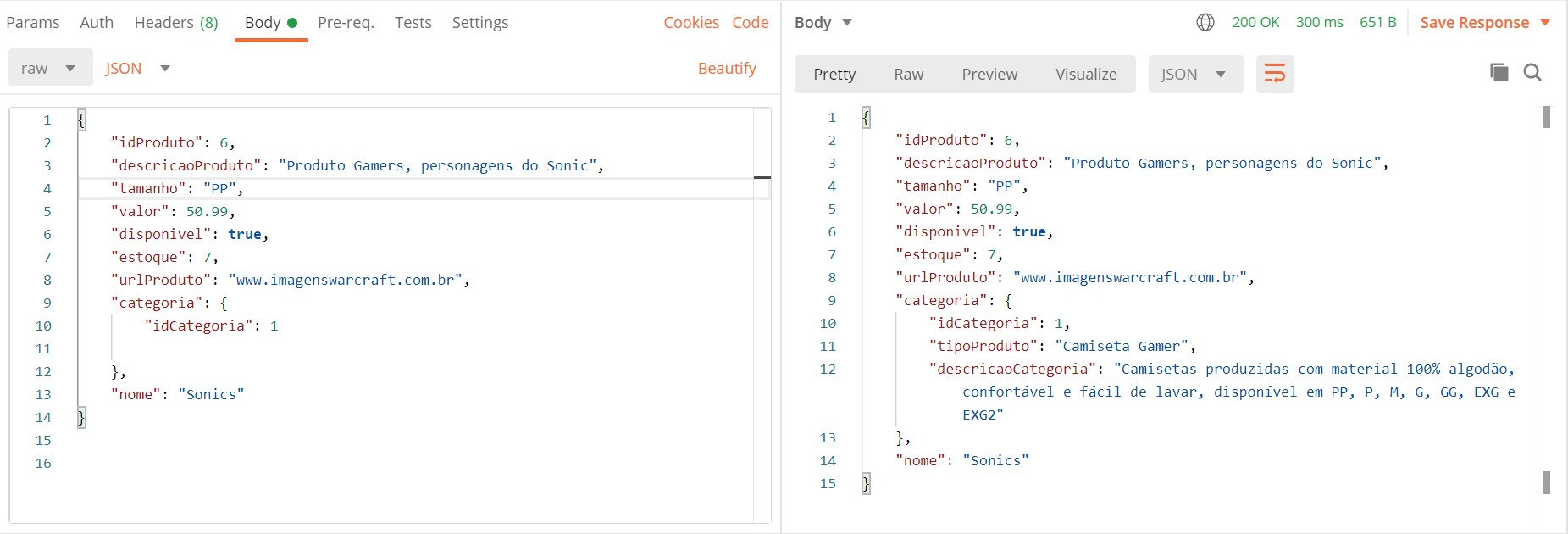
**POST**

<http://localhost:8080/produto/>

****

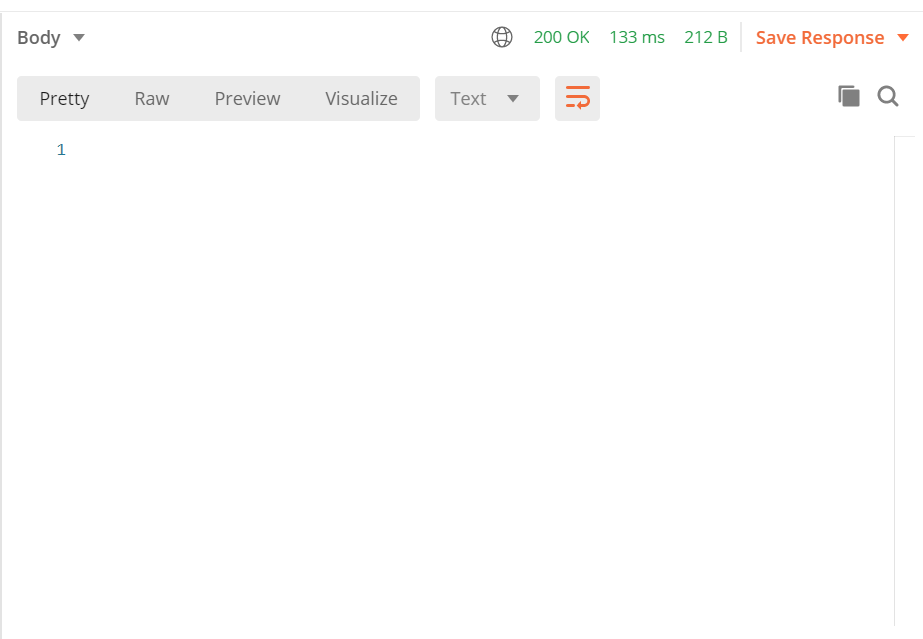
**PUT**

<http://localhost:8080/produto/>

****

**DELETE**

<http://localhost:8080/produto/6>

****