

Brenna de Mesquita Amorim (202205526)

O trabalho teve como foco criar uma API RESTful para gerenciar produtos, usando Spring Boot para a parte da lógica e MongoDB como banco de dados NoSQL. Para conectar tudo com o banco, foi usado o Spring Data MongoDB, que facilita bastante esse processo. A API conta com as operações básicas de CRUD: criar (POST), ler (GET), atualizar (PUT) e deletar (DELETE).

O desenvolvimento foi feito com JDK 24.0.2, Apache Maven 3.9.11 e Visual Studio Code. O MongoDB ficou rodando localmente na porta padrão 27017.

A primeira etapa foi configurar o application.properties, definindo o endereço do banco, a porta e o nome da base. Depois disso, foi criada a classe Produto.java, que representa os dados dos produtos. Ela foi anotada com @Document (pra indicar a coleção no MongoDB) e com @Id (pra identificar cada produto de forma única).

Em seguida, foi criada a interface ProdutoRepository.java, que estende MongoRepository<Produto, String>. Com isso, já foi possível usar os métodos CRUD sem precisar escrever código adicional para eles.

A classe ProdutoController.java ficou responsável por expor os endpoints da API. Ela foi anotada com @RestController e @RequestMapping("/api/produtos"). O repositório foi injetado usando @Autowired, permitindo que o controller interaja diretamente com o banco. Os métodos para cada operação (@PostMapping, @GetMapping, @PutMapping e @DeleteMapping) retornam ResponseEntity, permitindo controlar melhor as respostas HTTP.

Depois de tudo pronto, o projeto foi executado com o comando mvn spring-boot:run. A aplicação iniciou com sucesso, conectando ao MongoDB e subindo o servidor Tomcat na porta 8080. Os testes da API foram feitos com o curl, direto no terminal, e todas as operações funcionaram como esperado.

## POST

O comando foi usado para criar um novo produto na API, enviando os dados do arquivo produto.json para o endpoint /api/produtos. A resposta com o ID gerado confirma que o produto foi salvo com sucesso no MongoDB e que a criação está funcionando corretamente.

```
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe -X POST -H "Content-Type: application/json" -d '@produto.json' http://localhost:8080/api/produtos
>>
{"id":"687dcb7c4d87c15351e009bc","nome":"Blusa de Frio","descricao":"Blusa de frio azul","preco":35.0}
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe -X POST -H "Content-Type: application/json" -d '@produto.json' http://localhost:8080/api/produtos
>>
{"id":"687dcb8f4d87c15351e009bd","nome":"Celular Top Ultra (2025)","descricao":"Nova versão com 512GB e tela 120Hz","preco":3500.0}
PS C:\Users\Brenna\Downloads\java-mongodb-odm> |
```

## PUT

O comando foi utilizado para atualizar um produto existente na API. Ele enviou os novos dados do arquivo produto.json para o endpoint /api/produtos/{id}. A resposta confirma que o produto com o ID informado foi atualizado com sucesso no MongoDB.

```
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe -X PUT -H "Content-Type: application/json" -d '@produto.json' http://localhost:8080/api/produtos/687dcb7c4d87c15351e009bc
{"id":"687dcb7c4d87c15351e009bc","nome":"Blusa de Frio","descricao":"Blusa de frio verde","preco":20.0}
PS C:\Users\Brenna\Downloads\java-mongodb-odm> |
```

## GET

O comando foi usado para consultar todos os produtos cadastrados na API. Ele fez uma requisição GET para o endpoint /api/produtos, e a resposta retornou uma lista com todos os produtos armazenados no MongoDB, confirmando que a leitura dos dados está funcionando corretamente.

```
[{"id":"687dc70154f7e5751c0ac93a","nome":"Produto CRUD ATUALIZADO","descricao":"Item para testar as funcoes CRUD, agora atualizado e funcionando","preco":320.0}, {"id":"687dc90c54f7e5751c0ac93b","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}, {"id":"687dc91b54f7e5751c0ac93c","nome":"Notebook Gamer Pro","descricao":"RTX 4060, 16GB RAM, SSD 1TB","preco":5500.0}, {"id":"687dcb074d87c15351e009bb","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}, {"id":"687dcb7c4d87c15351e009bc","nome":"Blusa de Frio","descricao":"Blusa de frio verde","preco":20.0}, {"id":"687dcb8f4d87c15351e009bd","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}]
PS C:\Users\Brenna\Downloads\java-mongodb-odm>
```

## GET por ID

O comando foi usado para buscar um produto espec fico pelo seu ID no endpoint `/api/produtos/{id}`. A resposta trouxe os dados do produto correspondente, confirmando que a busca individual est  funcionando corretamente

```
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe http://localhost:8080/api/produtos/687dcb7c4d87c15351e009bc
{"id":"687dcb7c4d87c15351e009bc","nome":"Blusa de Frio","descricao":"Blusa de frio verde","preco":20.0}
PS C:\Users\Brenna\Downloads\java-mongodb-odm>
```

## DELETE

O comando enviou uma requis  o DELETE para o endpoint `/api/produtos/{id}` com o objetivo de remover um produto do banco. A aus ncia de erros na resposta indica que o produto foi exclu do com sucesso do MongoDB.

```
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe -X DELETE http://localhost:8080/api/produtos/687dcb7c4d87c15351e009bc
PS C:\Users\Brenna\Downloads\java-mongodb-odm> curl.exe http://localhost:8080/api/produtos
>>
[{"id":"687dc70154f7e5751c0ac93a","nome":"Produto CRUD ATUALIZADO","descricao":"Item para testar as funcoes CRUD, agora atualizado e funcionando","preco":320.0}, {"id":"687dc90c54f7e5751c0ac93b","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}, {"id":"687dc91b54f7e5751c0ac93c","nome":"Notebook Gamer Pro","descricao":"RTX 4060, 16GB RAM, SSD 1TB","preco":5500.0}, {"id":"687dcb074d87c15351e009bb","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}, {"id":"687dcb7c4d87c15351e009bc","nome":"Blusa de Frio","descricao":"Blusa de frio verde","preco":20.0}, {"id":"687dcb8f4d87c15351e009bd","nome":"Celular Top Ultra (2025)","descricao":"Nova vers o com 512GB e tela 120Hz","preco":3500.0}]
PS C:\Users\Brenna\Downloads\java-mongodb-odm>
```