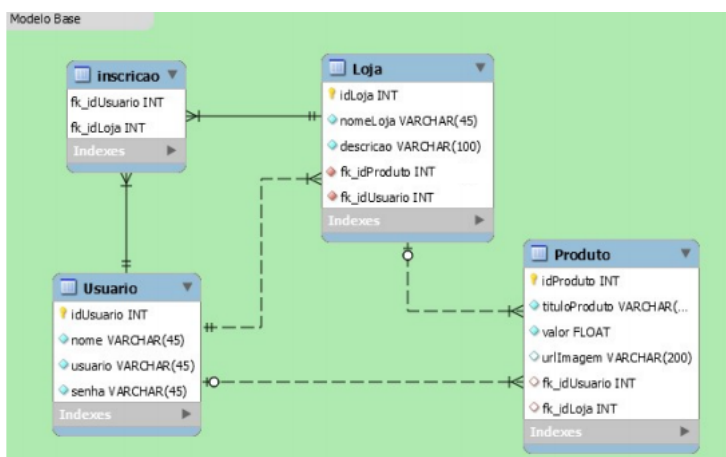


Api de um E-commerce

Boas práticas de construção de uma API rest.

- 0 - configure o application.properties para o seu banco de dados;
- 1 - Crie uma model com os atributos necessários + get setters+ anotações;
- 2 - Teste sua aplicação (verifique se a tabela foi criada no mysql workbench);
- 3 - Crie um repository e um controller com um método findAll;
- 4 - Insira dados via o mysql workbench;
- 5 - Teste o método findAll através do postman;
- 6 - Crie um método findById no controller;
- 7 - Teste o método findById via postman;
- 8 - Crie um método post no controller;
- 9 - Teste o método post via postman;
- 10 - Crie um método put no controller;
- 11 - Teste o método put no postman;
- 12 - Crie um método delete no controller;
- 13 - Teste o método delete no postman. Atividade CRUD Back-end E-commerce .

Construa um back-end para um E-commerce seguindo o DER abaixo



1- Crie todas as camadas necessárias para a criação desse projeto:

- Model
- Repository
- Controller
- Service

2- Crie os relacionamentos entre as tabelas como descrito na imagem;

3- Crie todas as models com os atributos descrito e suas respectivas tipagem na imagem.

4- Crie nos controller de cada uma da models métodos de CRUD completo com;

- findAll
- findById
- post
- put
- delete

5- Crie no controller de ProdutoController métodos de busca específicas nas seguintes categoria:

2 metodos MethodsQueryys

exemplo findAllByPrecoLessThanEqual

exemplo findAllByPrecoGreaterThanEqual

3 metodos @Query

exemplo :

```
@Query(value = "SELECT * FROM blog_pessoal.tb_produto where ativo = :ativo", nativeQuery = true)
```

```
public List findAllByAtivo(@Param("ativo") boolean ativo);
```

```
public List<Produto> findAllByPrecoGreaterThanEqual (float preco);
```

```
@Query(value = "SELECT * FROM tb_produto where ativo = :ativo", nativeQuery = true)
```

```
public List<Produto> findAllByAtivo(@Param("ativo") boolean ativo);
```

[link do projeto guia produtoRepository](#)

<https://github.com/LucasCapSilva/FarmaciaGeneration/blob/main/src/main/java/br/com/famacia/farmaciageneration/repository/ProdutoRepository.java>

[link do projeto guia produtoController](#)

<https://github.com/LucasCapSilva/FarmaciaGeneration/blob/main/src/main/java/br/com/famacia/farmaciageneration/controller/ProdutoController.java>

Material de apoio da atividade

projeto guia para a realização da atividade

[projeto guia](#)

<https://github.com/LucasCapSilva/FarmaciaGeneration>

documentação:

MethodsQueryys:

[MethodsQueryys link](#)

<https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>

@Query:

[MethodsQueryys link](#)

<https://www.baeldung.com/spring-data-jpa-query>

Many-To-Many Relationship in JPA:

[Many-To-Many](#)

<https://www.baeldung.com/jpa-many-to-many>