



Teste-Arcos

O que é?

O teste arcos é composto de um sistema de criação e listagem de requisições, onde será possível efetuar a consulta de relatórios compostos por um sistema empreendedor de produtos e vendas.



O sistema é feito em um protótipo* de linguagem Spring boot, Java 17, e armazenamento feito no banco de dados MySql server para guardar as informações de cadastro de dados.



O Software

O sistema software é composto por 3 tabelas: Sendo elas, a tabela de produtos, a tabela de estoques, e a tabela de vendas. Nelas, serão armazenados desde o valor de cada produto, bem como sua descrição, preços e quantidade.

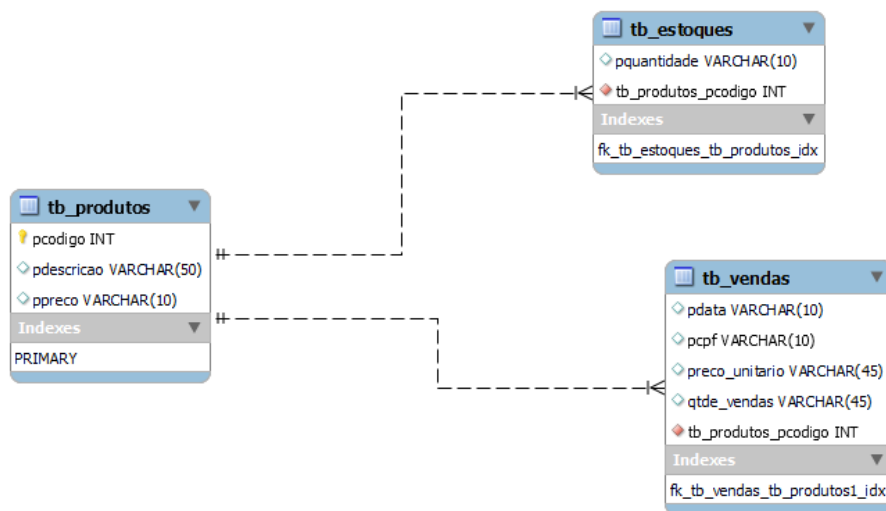


Figura 1 - Estrutura de tabelas no MYSQL (DER)

O sistema também contará com o suporte de requisições para a consulta de dados mencionados anteriormente promovendo maior flexibilidade e clareza de informações.



Objetivo



Perceber a capacitação do candidato com relação a estruturação de sistema conforme o desafio solicitado.



Funcionalidades do Usuário



Promover um cadastro de vendas para auxiliar os usuários que queiram cadastrar informações de produtos, bem como suas quantidades, preços e descrições.



Funcionalidades Técnicas

A. Formulação de cadastro para consulta de dados.

B. Estrutura de dados feito em MYSQL

C. Para aprimoramentos futuros, o sistema também poderá ser feito em Java, Springboot com integração ao banco.

D. Sistema de estrutura de Dados com a possibilidade de realização de consultas e adições de dados para armazenamentos.

E. O sistema poderá ser aprimorado com springboot + java 17 para promover maior sofisticação, caso o cliente solicite.



Teste-Arcos

The screenshot shows an IDE with a project named 'demo' in the left sidebar. The project structure includes 'src/main/java' with packages 'Teste_Arco.demo' and 'Teste_Arco.demo.model', and 'src/test/java'. The main code file is 'VendaController.java'. The code defines a REST API for a product repository with methods for getting, creating, and updating products.

```
44 }
45
46 @GetMapping("/{pcodigo}")
47 public ResponseEntity<ProdutoModel> getByPcodigo(@PathVariable Long pcodigo) {
48     return produtoRepository.findById(pcodigo)
49         .map(resposta -> ResponseEntity.ok(resposta))
50         .orElse(ResponseEntity.notFound().build());
51 }
52
53 @GetMapping("/pdescricao/{pdescricao}")
54 public ResponseEntity<List<ProdutoModel>> getByPdescricao(@PathVariable String pdescricao){
55     return ResponseEntity.ok(produtoRepository.findAllByPdescricaoContainingIgnoreCase(pdescricao));
56 }
57
58 @PostMapping
59 public ResponseEntity<ProdutoModel> postProduto (@Valid @RequestBody ProdutoModel produto){
60
61     if (estoqueRepository.existsById(produto.getPcodigo()))
62         return ResponseEntity.status(HttpStatus.CREATED).body(produtoRepository.save(produto));
63
64     return ResponseEntity.status(HttpStatus.BAD_REQUEST).build();
65 }
66
67
68 @PutMapping
69 public ResponseEntity<ProdutoModel> putPostagem (@Valid @RequestBody ProdutoModel produto){
70
71     if (produtoRepository.existsById(produto.getPcodigo())){
```

Figura 2 - Prévia CRUD no springbootem JAVA 17

Estrutura do código – Prevista do código para integração ao banco



Teste-Arcos

```
insert into tb_estoques(tb_produtos_pcodigo, pquantidade )
values
('1', '50');
```

Figura 3 - Inserção de dados no tb_estoques

```
SELECT T1.pcodigo,
T1.pdescricao,
T2.pquantidade
FROM tb_produtos AS T1
INNER JOIN tb_estoques AS T2
ON(T1.pcodigo=t2.tb_produtos_pcodigo);
```

Figura 4 - Filtro de dados do relatório de produtos cadastrados

```
insert into tb_vendas(tb_produtos_pcodigo, pdata, pcpf, preco_unitario, qtde_vendas )
values
('1', '04/09/2021', '2658672781', '22.50', '15');
```

Figura 5 - Inserção de dados no tb_vendas

```
SELECT T1.pcodigo,
T1.pdescricao,
sum(T2.qtde_vendas) as qtd_vendida
FROM tb_produtos AS T1
INNER JOIN tb_vendas AS T2
ON(T1.pcodigo=t2.tb_produtos_pcodigo) Where T2.pdata Between '04/09/2021' and '05/09/2021'
group by
T1.pcodigo,
T1.pdescricao;
```

Figura 6 - Requisição do relatório de Vendas



Teste-Arcos

```
insert into tb_produtos(pdescricao, ppreco)
values
('Máscara covid', '22.00');
```

```
select pcodigo, pdescricao, ppreco
from tb_produtos;
```

Figura 6 - Requisição do relatório de Vendas