

Documentação - Teste Prático

1) Classes, atributos e métodos do cenário:

Comanda	Consumido	Produto
<ul style="list-style-type: none">- idComanda : int- total : float	<ul style="list-style-type: none">- idComanda : int- idProduto : int- criadoem : data- quantidade : int	<ul style="list-style-type: none">- idProduto : int- nome : string- valor : float
<ul style="list-style-type: none">+ criaComanda+ getTotalComanda	<ul style="list-style-type: none">+ criaConsumido+ getMenosVendido+ getMaisVendido+ getTicketDia+ getTotalDia	<ul style="list-style-type: none">+ getListaProdutos

2) Diagrama de classes especificando relacionamentos e multiplicidade:

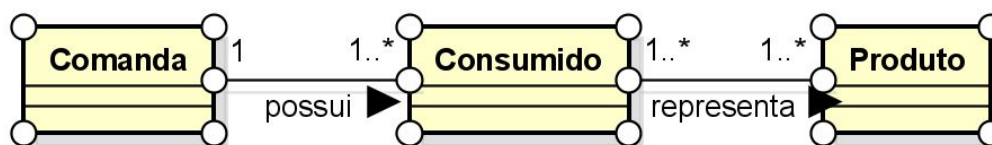


Diagrama de classes representando o cenário proposto.

Uma comanda vai possuir um ou mais produtos consumidos, enquanto os produtos consumidos representam os produtos existentes na padaria (é possível consumir mais de um produto na mesma comanda). A comanda só vai existir se houver algo a ser consumido.

3) Tecnologias utilizadas:

- NetBeans 12.3
- Java
- MySQL 8.0
- Conector JDBC: mysql-connector-java-8.0.22.jar

4) Consultas SQL:

a) Os valores totais das vendas diárias:

```
SELECT P.valor, C.quantidade FROM consumido C INNER JOIN produto P ON P.idProduto = C.idProduto  
WHERE criadoem BETWEEN '2021-03-10 00:00:00' AND '2021-03-10 23:59:59';
```

(exemplo no dia 10/03/2021)

b) O ticket médio diário:

O ticket médio diário é calculado como resultado da divisão do método anterior (valor total de vendas do dia) e do método abaixo (quantidade de vendas do dia), basicamente 'total/vendas'.

```
SELECT SUM(C.quantidade) FROM consumido C  
WHERE criadoem BETWEEN '2021-03-10 00:00:00' AND '2021-03-10 23:59:59';
```

(exemplo no dia 10/03/2021)

c) Três produtos mais vendidos e os seus respectivos totais de venda:

```
SELECT P.nome, SUM(quantidade) FROM consumido C INNER JOIN produto P ON P.idProduto = C.idProduto  
GROUP BY C.idProduto ORDER BY SUM(quantidade) DESC;
```

d) Três produtos menos vendidos e os seus respectivos totais de venda:

```
SELECT P.nome, SUM(quantidade) FROM consumido C INNER JOIN produto P ON P.idProduto = C.idProduto  
GROUP BY C.idProduto ORDER BY SUM(quantidade) ASC;
```