

Testr 2.0

# Supermercado em Java

 [Voltar](#)

**Deadline:** Dia 1 de dezembro de 2025 às 00:00 hora

**Linguagem de Programação:**

**Descrição:**

Considere um supermercado em que usuários podem realizar várias compras. Cada compra possui um ou mais produtos adquiridos em diferentes quantidades. Cada produto possui nome, preço e quantidade em estoque. O mesmo produto pode aparecer em várias compras. Além dos produtos e suas quantidades, as compras registram o cliente que fez a compra. Cada cliente possui nome e CPF. Para simular este contexto, faça um programa em Java que leia e execute uma sequência de comandos até que seja digitado o comando ENCERRAR. Os comandos possíveis são:

- REGISTRAR\_PRODUTO : adiciona o produto com o preço indicado e 0 itens no estoque.
- REGISTRAR\_CLIENTE : adiciona um cliente.
- ADQUIRIR\_PRODUTO : incrementa a quantidade de produto no estoque.
- INICIAR\_COMPRA : inicia o processo de uma nova compra da pessoa cujo cpf foi indicado. Os próximos comandos serão necessariamente ADD\_ITEM ou FINALIZAR\_COMPRA.
- ADD\_ITEM : adiciona um produto na compra com a quantidade indicada. Se o produto já existir, incrementa a quantidade.
- FINALIZAR\_COMPRA: finaliza a compra. Mostra na tela o valor total da compra. As quantidades de produtos vendidas devem ser decrementadas do estoque.
- REVENUE: mostra a soma dos valores de todas as compras feitas no supermercado até o momento.
- MAIS\_VENDIDO: mostra o nome do produto com mais itens vendidos

considerando todas as compras.

- MELHOR\_CLIENTE: mostra o nome do cliente que fez mais compras.
- EM\_FALTA: lista os produtos com menos de 5 itens em estoque.

**REGRAS:** Para tornar o projeto mais divertido, não é válido armazenar como atributos das classes o número de compras feitas por cada cliente e o número de itens vendidos de cada produto. As quantidades dos relatórios devem ser computadas a partir das compras realizadas.

**REGRAS:** Pode ser útil utilizar o tipo [Pair](#) para armazenar pares de valores durante o processo de gerar os relatórios.

Tente organizar o código e isolar as responsabilidades das classes o máximo que você conseguir. Por exemplo, crie uma classe Sistema que realize as interações com usuário e, dependendo do comando, invoque métodos das outras classes.

## Arquivos

Não existem arquivos

## Enviar Resposta

Procurar... Nenhum arquivo selecionado.

Upload

## Entradas e Saídas

Entrada 1

Saída 1

```
REGISTRAR_PRODUTO arroz  
10.0  
ADQUIRIR_PRODUTO arroz 20  
REGISTRAR_CLIENTE joao 111  
INICIAR_COMPRA 111  
ADD_ITEM arroz 17  
FINALIZAR_COMPRA  
REVENUE  
MAIS_VENDIDO  
MELHOR_CLIENTE
```

```
170.00  
170.00  
arroz  
joao  
arroz
```

### Entrada 2

```
REGISTRAR_PRODUTO arroz  
5.0  
REGISTRAR_PRODUTO feijao  
7.0  
REGISTRAR_CLIENTE maria 765  
ADQUIRIR_PRODUTO feijao 10  
REGISTRAR_PRODUTO agua 3.0  
REGISTRAR_CLIENTE isabel 123  
ADQUIRIR_PRODUTO agua 7  
ADQUIRIR_PRODUTO feijao 5
```

### Saída 2

```
44.00  
10.00  
25.00  
isabel  
feijao  
arroz  
agua  
79.00
```