

PROJETO DE JAVA

Sistema de Controle de Estoque

Um projeto prático para aplicar os conteúdos de Java estudados até agora

Individual ou em grupo | Execução em console

1. O DESAFIO

Uma pequena loja de informática precisa de um sistema simples para controlar seus produtos.

Atualmente, o funcionário registra as informações manualmente e tem dificuldade para saber:

- quais produtos estão disponíveis;
- quantas unidades existem de cada produto;
- quais produtos estão com estoque baixo;
- quanto dinheiro está armazenado em produtos;
- quais produtos possuem maior quantidade em estoque;
- quais produtos precisam de reposição.

Sua missão será desenvolver um Sistema de Controle de Estoque em Java, executado no console. O sistema deverá permitir que o usuário cadastre produtos, consulte informações, registre entradas e saídas de mercadorias e obtenha informações sobre o estoque.

Objetivo

Construir um programa que utilize os conteúdos de Java estudados em aula para resolver um problema semelhante ao encontrado em uma empresa real.

2. CONTEÚDOS QUE DEVERÃO SER UTILIZADOS

Conteúdo	Aplicação no projeto
Variáveis	Quantidades, preços, posições, opções do menu etc.
Scanner	Entrada de dados pelo usuário
if / else	Validações e regras do estoque
switch	Menu principal
for	Percorrer produtos e estruturas
while	Repetições durante determinadas operações
do...while	Manter o menu funcionando
Vetores	Armazenamento dos nomes, preços etc.
Arrays.toString()	Exibição de informações de um vetor

Conteúdo	Aplicação no projeto
Arrays.sort()	Ordenação de valores
Matrizes	Organização das informações de estoque

⚠ Importante

O objetivo é utilizar os conteúdos que já foram estudados. Não é necessário utilizar classes, objetos, banco de dados, arquivos ou interfaces gráficas.

3. SITUAÇÃO-PROBLEMA

A loja trabalha inicialmente com 5 produtos. Cada produto deverá possuir, no mínimo: nome, preço e quantidade em estoque.

O sistema deverá controlar as informações durante sua execução. Por exemplo:

Produto	Preço	Estoque
Teclado	R\$ 80,00	15
Mouse	R\$ 45,00	20
Monitor	R\$ 900,00	8
Webcam	R\$ 150,00	10
Headset	R\$ 120,00	12

Os produtos acima são apenas exemplos.

Seu programa deverá permitir que os dados sejam informados pelo usuário.

4. ESTRUTURA DOS DADOS

Você deverá pensar em uma forma de organizar as informações utilizando vetores e matriz. Uma possibilidade é utilizar:

```
String[] produtos;
double[] precos;
int[][] movimentacoes;
```

A matriz poderá armazenar, por exemplo:

	Entrada	Saída
Produto 1	20	5
Produto 2	15	3
Produto 3	10	2

...

Você deverá decidir como utilizar essas estruturas para representar o estoque.

Atenção

Não existe apenas uma maneira correta de organizar os dados. O importante é que:

- cada produto possa ser identificado;
- seu preço possa ser consultado;
- seu estoque possa ser controlado;
- as entradas e saídas possam ser registradas;
- o programa mantenha as informações consistentes.

5. MENU PRINCIPAL

O programa deverá apresentar um menu semelhante a:

```
=====
      CONTROLE DE ESTOQUE
=====

1 - Cadastrar produtos
2 - Exibir produtos
3 - Registrar entrada
4 - Registrar saída
5 - Consultar estoque
6 - Produtos com estoque baixo
7 - Valor total do estoque
8 - Relatório do estoque
9 - Ranking de preços
0 - Sair

Escolha uma opção:
```

O menu deverá continuar sendo apresentado até que o usuário escolha a opção 0 - Sair.

Requisitos: o menu deverá utilizar `do...while`, `switch` e `Scanner`.

6. MISSÕES DE DESENVOLVIMENTO

O sistema deverá ser desenvolvido em etapas. Não tente desenvolver tudo de uma vez. Faça uma missão funcionar antes de avançar para a próxima.

MISSÃO 1 — Criar a estrutura dos produtos

Crie os vetores necessários para armazenar os produtos. O sistema deverá trabalhar inicialmente com 5 produtos. Você deverá armazenar pelo menos: nome, preço e quantidade em estoque.

Desafio: defina corretamente o tamanho dos vetores.

● MISSÃO 2 — Cadastrar os produtos

Crie uma opção no programa para cadastrar os 5 produtos. Para cada produto, o usuário deverá informar:

Nome:
Preço:
Quantidade inicial:

Exemplo:

Produto 1
Nome: Teclado
Preço: 80
Quantidade: 15

Repita o processo para os demais produtos. Conteúdos envolvidos: Scanner, variáveis, vetores, for.

● MISSÃO 3 — Exibir os produtos

Crie uma opção para mostrar todos os produtos cadastrados. A saída poderá ser semelhante a:

```
===== PRODUTOS =====  
  
Código: 0  
Produto: Teclado  
Preço: R$ 80,00  
Estoque: 15  
  
Código: 1  
Produto: Mouse  
Preço: R$ 45,00  
Estoque: 20  
  
...
```

Utilize um for para percorrer os produtos.

● MISSÃO 4 — Criar o menu principal

Agora transforme o programa em um verdadeiro sistema. Crie um menu utilizando do...while e switch. O usuário deverá poder escolher as operações do sistema. O programa somente deverá terminar quando for escolhida a opção 0 - Sair.

● MISSÃO 5 — Registrar entrada de produtos

A loja recebe novos produtos. Crie a opção 3 - Registrar entrada. O sistema deverá perguntar qual produto recebeu mercadoria. Exemplo:

Digite o código do produto: 2
Digite a quantidade recebida: 10

Se o produto possuía Estoque: 8 e recebeu Entrada: 10, o novo estoque deverá ser Estoque: 18.

Validações: o programa deverá verificar se o código informado existe e se a quantidade é válida.

● **MISSÃO 6 — Registrar saída de produtos**

Crie a opção 4 - Registrar saída. O usuário deverá informar o código do produto e a quantidade retirada. O sistema deverá diminuir a quantidade disponível.

Regra importante

O sistema não poderá permitir uma saída maior que o estoque disponível. Exemplo: estoque atual 10, saída solicitada 15 → o programa deverá informar "Quantidade indisponível! Estoque atual: 10 unidades." O estoque não poderá ficar negativo.

● **MISSÃO 7 — Consultar estoque**

Crie uma opção para consultar um produto específico. O usuário deverá informar o código do produto. Exemplo:

Digite o código do produto: 3

Produto: Webcam

Preço: R\$ 150,00

Estoque atual: 7 unidades

Caso o código não exista, informe: "Produto não encontrado."

● **MISSÃO 8 — Produtos com estoque baixo**

Defina um limite de estoque baixo, por exemplo Estoque ≤ 5 . O programa deverá percorrer todos os produtos e mostrar aqueles que estiverem abaixo ou no limite definido. Exemplo:

===== ESTOQUE BAIXO =====

Monitor - 4 unidades

Mouse - 5 unidades

Utilize: for, if.

● **MISSÃO 9 — Calcular o valor total do estoque**

A loja quer saber quanto dinheiro está armazenado em produtos. Para cada produto: valor = preço $\times$ quantidade. Depois, some o valor de todos os produtos. Exemplo:

Teclado

R\$ 80 $\times$ 15 = R\$ 1.200

Mouse

R\$ 45 $\times$ 20 = R\$ 900

...

Valor total do estoque: R\$ 2.100,00

Desafio: não calcule somente um produto. O programa deverá calcular o valor total de todos os produtos.

● MISSÃO 10 — Descobrir o produto com maior estoque

Crie uma opção para descobrir qual produto possui a maior quantidade de unidades armazenadas. Exemplo:

```
===== MAIOR ESTOQUE =====
```

Produto: Mouse

Quantidade: 35 unidades

Utilize: variável para guardar o maior valor encontrado, for, if.

● MISSÃO 11 — Utilizar uma matriz para controlar movimentações

Agora entra o desafio relacionado às matrizes. Crie uma matriz para registrar as movimentações de cada produto. Uma sugestão:

```
int[][] movimentacoes = new int[5][2];
// coluna 0 -> entradas
// coluna 1 -> saídas
```

Produto	Entrada	Saída
Teclado	20	5
Mouse	30	12
Monitor	10	3
Webcam	15	4
Headset	25	10

Sempre que ocorrer uma entrada, atualize a coluna correspondente. Sempre que ocorrer uma saída, atualize a outra coluna.

Objetivo: ao final, o sistema deverá saber não apenas quanto existe no estoque, mas também quanto entrou e quanto saiu.

● MISSÃO 12 — Criar um relatório completo

Crie a opção 8 - Relatório do estoque. O relatório deverá apresentar informações como:

```
=====
```

RELATÓRIO DE ESTOQUE

```
=====
```

Produto	Preço	Entrada	Saída	Estoque
---------	-------	---------	-------	---------

```

-----
Teclado      R$ 80,00      20      5      15
Mouse       R$ 45,00      30     12     18
Monitor     R$ 900,00     10      3      7
Webcam      R$ 150,00     15      4     11
Headset     R$ 120,00     25     10     15

-----
Valor total do estoque: R$ XXXXX,XX
-----

```

Você deverá montar esse relatório utilizando os vetores e a matriz criados anteriormente.

● **MISSÃO 13 — Utilizar Arrays.toString()**

Crie uma opção para visualizar alguma informação armazenada em um vetor utilizando Arrays.toString(). Por exemplo, você poderá mostrar os nomes dos produtos:

```
[Teclado, Mouse, Monitor, Webcam, Headset]
```

Ou outra informação que faça sentido para o sistema.

● **MISSÃO 14 — Ranking de preços**

Agora vamos utilizar Arrays.sort(). Crie um vetor contendo os preços dos produtos e utilize Arrays.sort() para ordená-los. O sistema deverá apresentar os valores ordenados. Exemplo:

```

===== PREÇOS ORDENADOS =====

R$ 45,00
R$ 80,00
R$ 120,00
R$ 150,00
R$ 900,00

```

Desafio: tente apresentar os preços do maior para o menor.

🏆 **DESAFIO FINAL — SISTEMA COMPLETO**

Agora chegou a hora de juntar tudo. Seu programa deverá funcionar como um verdadeiro pequeno sistema de controle de estoque, permitindo que o usuário navegue pelo sistema sem precisar reiniciar o programa a cada operação:

```

=====
                SISTEMA DE ESTOQUE
=====

1 - Cadastrar produtos
2 - Exibir produtos
3 - Registrar entrada
4 - Registrar saída

```

- 5 - Consultar produto
- 6 - Produtos com estoque baixo
- 7 - Valor total do estoque
- 8 - Relatório completo
- 9 - Ranking de preços
- 0 - Sair

7. REQUISITOS OBRIGATÓRIOS

Para considerar o projeto completo, seu programa deverá possuir:

- ☐ 5 produtos cadastrados
- ☐ Vetor para armazenar nomes
- ☐ Vetor para armazenar preços
- ☐ Estrutura para controlar quantidade em estoque
- ☐ Matriz para registrar entradas e saídas
- ☐ Menu principal
- ☐ Scanner
- ☐ switch
- ☐ if / else
- ☐ for
- ☐ while
- ☐ do...while
- ☐ Registro de entrada
- ☐ Registro de saída
- ☐ Validação para impedir estoque negativo
- ☐ Consulta de produto
- ☐ Identificação de estoque baixo
- ☐ Cálculo do valor total do estoque
- ☐ Identificação do produto com maior estoque
- ☐ Uso de Arrays.toString()
- ☐ Uso de Arrays.sort()
- ☐ Relatório completo

8. DESAFIOS EXTRAS

Terminou tudo? Então melhore seu sistema. Escolha um ou mais desafios abaixo.

★ Desafio 1 — Aumentar a quantidade de produtos

Faça o sistema trabalhar com 10 produtos em vez de apenas 5.

★ Desafio 2 — Sistema de reposição

Crie uma opção "Recomendar reposição". O sistema deverá identificar produtos com estoque baixo e informar uma quantidade sugerida para reposição.

★ Desafio 3 — Produto mais caro

Descubra qual produto possui o maior preço. Exemplo: Produto mais caro: Monitor — R\$ 900,00.

★ Desafio 4 — Produto mais barato

Faça o contrário: Produto mais barato: Mouse — R\$ 45,00.

★ Desafio 5 — Total de entradas e saídas

Calcule o total de unidades que entraram e o total que saíram, utilizando a matriz.

★ Desafio 6 — Estoque zerado

Identifique produtos que chegaram a 0 unidades e mostre "⚠ PRODUTOS ESGOTADOS".

★ Desafio 7 — Relatório estatístico

Mostre maior estoque, menor estoque, maior preço, menor preço, quantidade total de produtos armazenados e valor total do estoque.

9. DESAFIO DE RACIOCÍNIO



Pergunta do gerente

"Quero saber quanto dinheiro a loja perdeu em produtos que saíram do estoque."

Como você poderia calcular isso? Pense: preço do produto × quantidade que saiu. Depois, some os valores de todos os produtos.

Não implemente imediatamente. Primeiro tente descobrir sozinho como resolver.

10. TRABALHO EM EQUIPE

O projeto poderá ser desenvolvido individualmente ou em grupos, conforme orientação do professor. Se for realizado em grupo, cada integrante deverá participar do desenvolvimento e deverá ser capaz de explicar o funcionamento do programa.

Uma sugestão de divisão:

Integrante 1. Cadastro e consulta de produtos.

Integrante 2. Entrada e saída de estoque.

Integrante 3. Relatórios e cálculos.

Integrante 4. Menu, testes e integração.

Atenção

A divisão de tarefas não significa que cada integrante ficará responsável apenas por sua parte. No final, todos deverão conhecer o funcionamento do projeto.

11. CRITÉRIOS DE AVALIAÇÃO

Critério	Valor
Estrutura correta de vetores e matriz	1,5
Menu e interação com o usuário	1,0
Cadastro e consulta de produtos	1,0
Entrada e saída de estoque	1,5
Validações e uso de estruturas condicionais	1,0
Uso das estruturas de repetição	1,0
Relatórios e cálculos	1,0
Arrays.toString() e Arrays.sort()	0,5
Organização, criatividade e funcionamento	0,5
TOTAL	10,0

12. REGRAS DO PROJETO

- O programa deverá ser executado no console.
- O código deverá estar organizado e indentado.
- Utilize nomes de variáveis que façam sentido.
- Evite repetir código desnecessariamente.
- O programa não deverá encerrar inesperadamente durante uma operação válida.
- O estoque nunca poderá ficar negativo.
- O grupo deverá testar diferentes situações.
- Todos os integrantes deverão conseguir explicar o código.
- Recursos que ainda não foram estudados poderão ser utilizados somente com autorização do professor.

13. TESTES OBRIGATÓRIOS

Antes de entregar, teste pelo menos estas situações:

Teste 1. Cadastrar todos os produtos.

Teste 2. Consultar cada produto.

Teste 3. Registrar uma entrada.

Teste 4. Registrar uma saída válida.

Teste 5. Tentar retirar uma quantidade maior que o estoque.

Teste 6. Consultar um código inexistente.

Teste 7. Verificar produtos com estoque baixo.

Teste 8. Calcular o valor total do estoque.

Teste 9. Verificar o maior estoque.

Teste 10. Executar o ranking de preços.

Teste 11. Executar o relatório completo.

14. CHECKLIST FINAL

Antes de entregar, confira:

- ☐ Meu programa compila sem erros.
- ☐ O menu funciona.
- ☐ Posso cadastrar os produtos.
- ☐ Posso consultar os produtos.
- ☐ Posso registrar entradas.
- ☐ Posso registrar saídas.
- ☐ O estoque não fica negativo.
- ☐ O programa identifica estoque baixo.
- ☐ O programa calcula o valor total do estoque.
- ☐ O programa identifica o maior estoque.
- ☐ Utilizei uma matriz.
- ☐ Utilizei vetores.
- ☐ Utilizei Arrays.toString().
- ☐ Utilizei Arrays.sort().
- ☐ Utilizei for.
- ☐ Utilizei while.
- ☐ Utilizei do...while.
- ☐ Utilizei if / else.
- ☐ Utilizei switch.
- ☐ Testei situações de erro.
- ☐ Consigo explicar meu código.