

1. Objetivos:

Criar um sistema simples de controle de estoque utilizando os conceitos de Programação Orientada a Objetos (POO), como **classes**, **objetos**, **encapsulamento**, **métodos**, **construtores** e **listas (ArrayList)**.

2. Enunciado:

Você foi contratado para desenvolver um sistema básico de controle de estoque para uma loja. O sistema deverá permitir:

- (I) Cadastrar produtos no estoque (cadastrar em uma lista).
- (II) A lista de produtos deverá ter os seguintes campos: código, nome, quantidade e preço
- (III) Listar todos os produtos cadastrados.
- (IV) Atualizar o estoque de um produto (entrada ou saída).
- (V) Remover um produto do estoque.
- (VI) Buscar um produto pelo código.

3. Especificações:

- (I) Crie uma classe Produto com os seguintes atributos:
 - int codigo
 - String nome
 - double preco
 - int quantidade
- (II) Inclua:
 - Construtor
 - Métodos getters e setters
 - Um método `exibirInfo()` que retorna as informações do produto.
- (III) Crie uma classe Estoque que gerencie uma lista de produtos:
 - Atributo: `ArrayList<Produto> produtos`
 - Métodos:
 - `void adicionarProduto(Produto p)`
 - `void removerProduto(int codigo)`
 - `Produto buscarProduto(int codigo)`
 - `void listarProdutos()`

- `void atualizarEstoque(int codigo, int quantidade, boolean entrada)`
(se `entrada = true`, aumenta o estoque; se `false`, reduz)

(IV) **Crie uma classe Main com um menu interativo no console:**

Exemplo de menu:

```
1 - Adicionar Produto
2 - Listar Produtos
3 - Atualizar Estoque
4 - Remover Produto
5 - Buscar Produto
0 - Sair
```

Use Scanner para ler as opções e interagir com o usuário.

4. Requisitos mínimos:

- Aplicar encapsulamento corretamente.
- Usar listas para armazenar os produtos.
- Usar switch para o menu.
- Validar entradas básicas (como produto não encontrado).

5. Dica para os alunos:

Você pode ir além e adicionar:

- Salvar os dados em arquivo texto.
- Atualizar preços.
- Relatórios com produtos com estoque baixo.