

Projeto — Sistema de Controle de Estoque

1. Objetivo

O projeto pretende gerenciar um estoque de produtos, permitindo:

- Cadastro de novos produtos.
 - Alteração de dados de produtos.
 - Exclusão de produtos.
 - Emissão de relatórios de produtos cadastrados.
 - Registro de vendas e cálculo automático de lucro.
-

2. Funcionalidades

- **Inclusão de produtos:** cadastro de novos itens, limitados a 48 produtos no total.
 - **Alteração de produtos:** atualização de quantidade e preço de venda.
 - **Exclusão de produtos:** remoção de produtos pelo código.
 - **Relatórios:**
 - **Relatório Parcial:** lista de códigos e nomes dos produtos.
 - **Relatório Específico:** detalhes de um produto específico.
 - **Relatório Completo:** todos os produtos, incluindo cálculos de valor total, valor vendido e lucro.
 - **Venda de produtos:** diminuição do estoque e registro de vendas para cálculo posterior de lucro.
-

3. Estrutura de Pastas

```
Trabalho2semestre/  
├── Estoque.java  
├── InOut.java
```

```
|— Produtos.java
|— main.java
```

4. Classes e Responsabilidades



Produtos

- Representa um produto no estoque.
- Atributos: código, nome, preço de custo, preço de venda, quantidade em estoque, quantidade vendida.
- Métodos:
 - Assessores (getters e setters)
 - `vender(int quantidade)` — realiza uma venda
 - Cálculos de:
 - Valor total em estoque
 - Valor total vendido
 - Lucro obtido



Estoque

- Controla a lista de produtos.
- Métodos:
 - `adicionarProdutos(Produtos p)` — adiciona produto (verifica duplicidade e limite máximo de 48 produtos)
 - `alterarProduto(String codigo, int novaQtd, float novoPreco)` — altera estoque/preço.
 - `excluirProduto(String codigo)` — remove um produto.
 - `relatorioPr()` — lista parcial de produtos.
 - `relatorioEspecifico(String codigo)` — informações de um produto.
 - `relatorioCompleto()` — informações e cálculos de todos os produtos.



InOut

- Interface de entrada/saída com o usuário via caixas de diálogo (`JOptionPane`).
 - Métodos:
 - Leitura de vários tipos (`String` , `int` , `float` , etc.)
 - Mensagens de erro, informação, aviso e mensagens simples.
-



main

- Controle de execução do programa.
 - Exibe menu de opções para o usuário.
 - Chama as funções correspondentes de `Estoque` e `Produtos` .
-

5. Fluxo de Funcionamento

1. O usuário inicia o programa.
 2. Escolhe uma opção no menu:
 - Cadastrar, alterar, excluir, vender, ou emitir relatórios.
 3. O programa interage via janelas (`JOptionPane`) para entrada e saída de dados.
 4. Todos os dados são manipulados em memória (não há persistência em arquivos).
-

6. Tecnologias Utilizadas

- **Java** (versão 8 ou superior).
- **Swing** (`JOptionPane`) para janelas de diálogo.
- **IDE sugerida:** Eclipse, NetBeans, IntelliJ ou qualquer editor de sua escolha.