

PRD — Carrinho de compras

- **Projeto:** Simulador de Carrinho de Compras
 - **Disciplina:** Design Patterns & Clean Code
 - **Objetivo:** Desenvolver um sistema simples em **PHP** aplicando **PSR-12**, **KISS** e **DRY**, para simular a operação de um carrinho de compras em um e-commerce.
-

Objetivo de Aprendizagem

- Praticar **boas práticas de programação** (PSR-12, DRY, KISS).
 - Aplicar **validações e regras de negócio** simples.
 - Entender a importância de **organização do código** e **documentação mínima**.
 - Simular um **fluxo real de negócio** (checkout de e-commerce).
-

Escopo do Projeto

O sistema deve simular um carrinho de compras básico. Todos os dados serão fixos e representados em **arrays PHP**. Não é necessário banco de dados.

1. Lista de Produtos

- Cada produto deve conter:
 - **id** (inteiro)
 - **nome** (string)
 - **preco** (float)
 - **estoque** (int)

Exemplo:

```
$produtos = [  
    ['id' => 1, 'nome' => 'Camiseta', 'preco' => 59.90, 'estoque' => 10],  
    ['id' => 2, 'nome' => 'Calça Jeans', 'preco' => 129.90, 'estoque' =>  
5],  
    ['id' => 3, 'nome' => 'Tênis', 'preco' => 199.90, 'estoque' => 3]  
];
```

2. Carrinho de Compras

- Deve ser representado como um array.
- Cada item do carrinho deve conter:
 - **id_produto**
 - **quantidade**

- **subtotal** (calculado automaticamente).
-

Funcionalidades Requeridas

1. Adicionar Item ao Carrinho

- Validar se o produto existe.
- Validar se há estoque suficiente.
- Atualizar carrinho e reduzir estoque.

2. Remover Item do Carrinho

- Validar se o item existe no carrinho.
- Atualizar carrinho e devolver estoque.

3. Listar Itens do Carrinho

- Mostrar os itens adicionados com quantidade, subtotal e total.

4. Calcular Total

- Somar todos os subtotais.

5. Aplicar Desconto Fixo

- Regra simples: cupom **DESCONT010** → 10% no total.
 - O desconto deve ser aplicado no cálculo do total final.
-

Casos de Uso

Caso 1 — Usuário adiciona um produto válido

- Entrada: produto **id=1**, quantidade **2**.
- Resultado esperado: produto é adicionado ao carrinho, estoque atualizado.

Caso 2 — Usuário tenta adicionar além do estoque

- Entrada: produto **id=3**, quantidade **10**.
- Resultado esperado: mensagem de erro → "Estoque insuficiente".

Caso 3 — Usuário remove produto do carrinho

- Entrada: produto **id=2**.
- Resultado esperado: produto removido, estoque restaurado.

Caso 4 — Aplicação de cupom de desconto

- Entrada: cupom **DESCONT010**.
 - Resultado esperado: valor total reduzido em 10%.
-

Limitações

- Não precisa persistir dados (uso apenas de arrays).
 - Não precisa interface de login/usuário.
 - Não precisa input via formulário → valores podem estar fixos em arrays/variáveis.
 - Não usar frameworks externos. Apenas **PHP puro rodando no XAMPP**.
-

Entregáveis

1. Repositório no GitHub com:

- Código fonte em PHP seguindo **PSR-12**.
- README.md contendo:
 - Nome e RA da dupla.
 - Instruções de execução (XAMPP).
 - Breve documentação (funcionalidades, regras de negócio, limitações).
- Pastas organizadas (**src/**, **docs/**).

2. README.md deve conter:

- Como rodar o projeto (**<http://localhost/carrinho>**).
 - Lista de funcionalidades implementadas.
 - Exemplos de uso (casos de teste).
-

Critérios de Avaliação

- Organização do código (PSR-12).
- Aplicação de **DRY e KISS**.
- Cobertura das funcionalidades mínimas.
- Clareza e completude da documentação.
- Criatividade na apresentação da solução.