

Atividade: Sistema Básico de Cadastro de Produtos (PHP e HTML)

Enunciado da Atividade

Você deve desenvolver um sistema básico para o cadastro de produtos utilizando as tecnologias PHP e HTML. Cada produto a ser cadastrado deverá possuir as seguintes informações:

- **Nome:** O nome descritivo do produto.
- **Preço:** O valor unitário do produto.
- **Quantidade:** A quantidade disponível do produto em estoque.

Requisitos:

1. **Criação de Classe `Produto`** : Desenvolva uma classe em PHP denominada `Produto` . Esta classe deve encapsular os atributos `nome` , `preço` e `quantidade` .
2. **Formulário HTML:** Crie um formulário HTML intuitivo que permita ao usuário inserir os dados de um novo produto (Nome, Preço, Quantidade).
3. **Métodos da Classe `Produto`** : Implemente métodos dentro da classe `Produto` para:
 - **Calcular o Valor Total em Estoque:** Um método que retorne o valor monetário total do produto em estoque, calculado pela multiplicação do `preço` pela `quantidade` .
 - **Exibir Dados Formatados:** Um método que apresente todos os dados do produto (Nome, Preço, Quantidade e o Valor Total em Estoque) de forma clara e formatada.
4. **Exibição Pós-Cadastro:** Após o envio do formulário HTML, o sistema deve processar os dados e exibir as informações do produto recém-cadastrado, utilizando o método de exibição formatada da classe `Produto` .

Como Resolver a Atividade (Sem Código)

Para abordar esta atividade, você deve seguir uma sequência lógica de passos, focando na separação de responsabilidades entre o HTML (interface do usuário) e o PHP (lógica de negócio e processamento).

Passo 1: Estrutura da Classe PHP (`Produto`)

Comece definindo a estrutura da sua classe `Produto` em PHP. Pense nos atributos (`nome` , `preço` , `quantidade`) e como eles serão armazenados (propriedades da classe). Você precisará de um construtor para inicializar esses atributos quando um novo objeto `Produto`

for criado. Além disso, para acessar e modificar esses atributos de forma controlada, considere a implementação de métodos

getters (para obter os valores) e setters (para definir os valores, se necessário).

Passo 2: Desenvolvimento do Formulário HTML

Em seguida, crie o arquivo HTML que conterá o formulário de cadastro. Este formulário deve ser simples e direto, com campos de entrada (`<input>`) para o nome, preço e quantidade do produto. Certifique-se de que os campos tenham atributos `name` apropriados, pois eles serão usados pelo PHP para coletar os dados enviados. O formulário deve ter um método de envio (`method="post"` é o mais comum para envio de dados) e uma ação (`action`) que aponte para o script PHP que irá processar esses dados.

Passo 3: Implementação dos Métodos na Classe `Produto`

Dentro da sua classe `Produto` , você precisará adicionar a lógica para os dois métodos principais solicitados:

- **`calcularValorTotal()`** : Este método deve acessar as propriedades `preço` e `quantidade` do próprio objeto `Produto` e retornar o resultado da multiplicação entre elas. Lembre-se de que o preço pode ser um número decimal, então use tipos de dados apropriados.
- **`exibirDados()`** : Este método deve construir uma string ou um array formatado que contenha todas as informações do produto: nome, preço (formatado como moeda), quantidade e o valor total em estoque (obtido através do método `calcularValorTotal()`). O objetivo é que a saída seja legível e informativa para o usuário.

Passo 4: Script PHP de Processamento

Crie um arquivo PHP separado (por exemplo, `processar.php`) que será o `action` do seu formulário HTML. Este script terá as seguintes responsabilidades:

1. **Incluir a Classe `Produto`** : Certifique-se de que o script de processamento tenha acesso à definição da sua classe `Produto` . Isso geralmente é feito com `require_once` ou `include_once` .
2. **Coletar Dados do Formulário**: Quando o formulário HTML for enviado, os dados estarão disponíveis na superglobal `$_POST` . Você precisará coletar o `nome` , `preço` e `quantidade` desses dados. É crucial validar e sanitizar esses dados para garantir a segurança e a integridade da aplicação (por exemplo, garantir que preço e quantidade sejam números).
3. **Instanciar a Classe `Produto`** : Com os dados coletados, crie uma nova instância da sua classe `Produto` , passando o nome, preço e quantidade para o construtor.

4. **Exibir os Dados do Produto:** Após criar o objeto `Produto`, chame o método `exibirDados()` desse objeto para obter a representação formatada do produto. Em seguida, exiba essa informação na página HTML que o script PHP irá gerar como resposta ao envio do formulário.

Ao seguir esses passos, você construirá um sistema funcional que atende a todos os requisitos da atividade, separando a lógica de negócios da apresentação e demonstrando o uso de classes e objetos em PHP para gerenciar dados de produtos.