1

Documentação Técnica da   
Aplicação Web   
 Homework  
  
  
Gabriel Araujo de Figueiredo  
  
(sonicultratv@gmail.com)

Nova Iguaçu  
Novembro/2024

**Índice Completo - Carrinho de Compras**

1. **Introdução**  
   1.1 Propósito do Documento  
   1.2 Contextualização do Projeto
2. **Descrição Geral**  
   2.1 O que é um Carrinho de Compras?  
   2.2 Benefícios e Utilização
3. **Requisitos do Sistema**  
   3.1 Funcionalidades Essenciais  
   3.2 Tecnologias Utilizadas
4. **Projeto Lógico e Implementação**  
   4.1 Estrutura de Arquivos e Diretórios  
   4.2 Descrição do Fluxograma  
   4.3 Descrição do Algoritmo
5. **Código do Sistema**  
   5.1 Arquivo PHP Principal  
   5.2 Arquivo HTML para Interface  
   5.3 Arquivo CSS para Estilo
6. **Explicação Detalhada do Código**  
   6.1 Funções e Lógica  
   6.2 Fluxo do Sistema
7. **Manual do Usuário**  
   7.1 Instruções para o Administrador  
   7.2 Instruções para o Cliente
8. **Considerações Finais**  
   8.1 Desafios Encontrados  
   8.2 Possíveis Melhorias Futuras

**1. Introdução**

**1.1 Propósito do Documento**

Este documento descreve o processo de construção e implementação de um carrinho de compras simples, integrando HTML, CSS e PHP. Inclui uma simulação de banco de dados para manipulação de produtos.

**1.2 Contextualização do Projeto**

O projeto visa facilitar a experiência do usuário ao realizar compras online, apresentando uma interface funcional e intuitiva.

**2. Descrição Geral**

**2.1 O que é um Carrinho de Compras?**

Um carrinho de compras é uma funcionalidade essencial em plataformas de e-commerce, permitindo que usuários adicionem produtos, visualizem preços e finalizem compras.

**2.2 Benefícios e Utilização**

* **Organização**: Armazena informações temporárias sobre produtos.
* **Interatividade**: Fornece uma experiência interativa com o cliente.
* **Escalabilidade**: Base para sistemas mais complexos.

**3. Requisitos do Sistema**

**3.1 Funcionalidades Essenciais**

* Adicionar produtos ao carrinho.
* Exibir conteúdo do carrinho.
* Remover produtos.
* Consultar banco de dados para preços.

**3.2 Tecnologias Utilizadas**

* **HTML**: Estrutura da interface.
* **CSS**: Estilização e design.
* **PHP**: Lógica do sistema.
* **Simulação de Banco de Dados**: Gerenciamento de produtos.

**4. Projeto Lógico e Implementação**

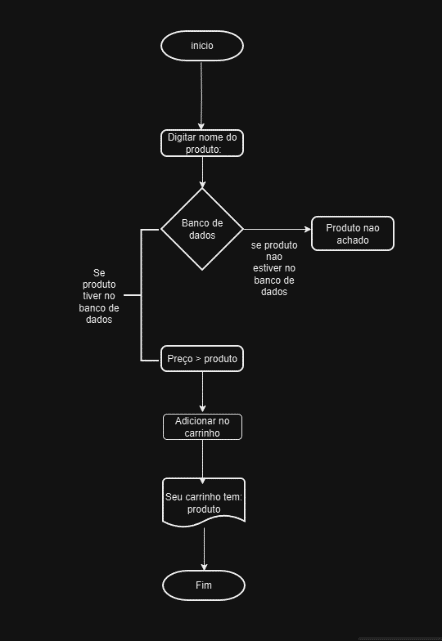
**4.1 Estrutura de Arquivos e Diretórios**

/projeto-carrinho/

│── index.php (Arquivo PHP principal)

│── styles.css (Arquivo CSS)

**4. Projeto Lógico e Implementação**

**4.2 Descrição do Fluxograma**

O **fluxograma** do sistema segue o fluxo lógico de uma aplicação básica de carrinho de compras. Ele é organizado da seguinte forma:

1. **Início**: O sistema é iniciado, aguardando a entrada do usuário.
2. **Entrada do Nome do Produto**: O usuário digita o nome do produto que deseja adicionar ao carrinho.
3. **Consulta ao Banco de Dados (Simulado)**:
   * O sistema verifica se o produto está disponível no banco de dados.
   * Caso o produto não seja encontrado, exibe uma mensagem informando que o item não foi localizado.
4. **Produto Encontrado**:
   * Se o produto está no banco de dados, o sistema verifica o preço e continua com a adição ao carrinho.
5. **Adição ao Carrinho**: O produto é adicionado ao carrinho, e uma mensagem de confirmação é exibida ao usuário.
6. **Exibição do Carrinho**: O sistema apresenta os produtos atualmente no carrinho.
7. **Fim**: O processo termina.

**Detalhes do Fluxograma:**

1. O **decisor** no fluxograma é o ponto onde o sistema verifica se o produto está no banco de dados.
2. O fluxo garante que os usuários recebam feedback imediato (ex.: mensagens de "Produto não encontrado").
3. A funcionalidade "Adicionar no Carrinho" atualiza a sessão, representando a persistência temporária.

**4.3 Descrição do Algoritmo**

A lógica do algoritmo segue a estrutura do fluxograma, implementando cada etapa em pseudocódigo. Aqui está o detalhamento do algoritmo:

Algoritmo CarrinhoDeCompras

Variáveis:

produto: texto

bancoDeDados: tabela [nome: texto, preco: real]

carrinho: lista [nome: texto, preco: real]

Início:

Escreva("Digite o nome do produto:")

Leia(produto)

Se produto está no bancoDeDados Então

Escreva("Produto encontrado!")

Adicione produto no carrinho

Escreva("Produto adicionado ao carrinho.")

Exiba conteúdo do carrinho

Senão

Escreva("Produto não encontrado no banco de dados.")

FimSe

Fim.

**Explicação Passo a Passo:**

1. **Declaração de Variáveis**:
   * produto: Nome do produto que o usuário deseja adicionar.
   * bancoDeDados: Representa os produtos disponíveis e seus preços.
   * carrinho: Lista onde serão armazenados os produtos selecionados.
2. **Leitura do Produto**:  
   O sistema solicita o nome do produto e o armazena na variável produto.
3. **Verificação no Banco de Dados**:  
   Uma condicional verifica se o produto está presente no bancoDeDados.
4. **Produto Encontrado**:  
   Caso o produto exista:
   * Exibe uma mensagem informando que o produto foi encontrado.
   * Adiciona o produto ao carrinho.
5. **Produto Não Encontrado**:  
   Caso o produto não seja encontrado no bancoDeDados, uma mensagem de erro é exibida.
6. **Exibição do Carrinho**:  
   Lista os produtos atualmente no carrinho, com seus respectivos preços.

**5. Código do Sistema**

**5.1 Arquivo PHP Principal**

<?php

session\_start();

$produtos = [

"Teclado" => 150.00,

"Mouse" => 100.00,

"Monitor" => 800.00

];

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$nomeProduto = $\_POST['nomeProduto'];

if (array\_key\_exists($nomeProduto, $produtos)) {

$\_SESSION['carrinho'][] = [

"nome" => $nomeProduto,

"preco" => $produtos[$nomeProduto]

];

$mensagem = "Produto adicionado com sucesso!";

} else {

$mensagem = "Produto não encontrado!";

}

}

if (isset($\_POST['remover'])) {

unset($\_SESSION['carrinho'][$\_POST['remover']]);

}

**5.2 Arquivo HTML para Interface**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Bem-vindo ao Carrinho de Compras</h1>

<form method="POST" action="index.php">

<label>Digite o nome do produto:</label>

<input type="text" name="nomeProduto" required>

<button type="submit">Adicionar</button>

</form>

<?php if (!empty($mensagem)) echo "<p>$mensagem</p>"; ?>

<h2>Seu Carrinho:</h2>

<ul>

<?php

if (!empty($\_SESSION['carrinho'])) {

foreach ($\_SESSION['carrinho'] as $indice => $item) {

echo "<li>{$item['nome']} - R$ {$item['preco']}";

echo "<form method='POST' action='index.php'>";

echo "<button name='remover' value='$indice'>Remover</button>";

echo "</form></li>";

}

} else {

echo "<p>Seu carrinho está vazio.</p>";

}

?>

</ul>

</body>

</html>

**5.3 Arquivo CSS para Estilo**

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 20px;

}

h1 {

color: #333;

}

form {

margin-bottom: 20px;

}

ul {

list-style-type: none;

padding: 0;

}

li {

margin: 10px 0;

}

button {

background-color: #007bff;

color: white;

border: none;

padding: 5px 10px;

cursor: pointer;

}

**6. Explicação Detalhada do Código**

* **session\_start**: Gerencia as variáveis de sessão.
* **Simulação de banco de dados**: Array $produtos representa os dados.
* **Lógica POST**: Processa adições e remoções de produtos.

**7. Manual do Usuário**

**7.1 Instruções para o Administrador**

* Adicione novos produtos no array $produtos.
* Personalize estilos no arquivo styles.css.

**7.2 Instruções para o Cliente**

* Digite o nome do produto e clique em "Adicionar".
* Para remover, clique no botão "Remover".

**8. Considerações Finais**

**8.1 Desafios Encontrados**

* Integração de lógica com interface.

**8.2 Possíveis Melhorias Futuras**

* Uso de banco de dados real.
* Sistema de autenticação.