**CONSTRUINDO UM MINI E-COMMERCE COM HTML, CSS E JAVASCRIPT**

**1. Introdução**

Neste projeto, desenvolvemos uma página de **loja virtual** que permite:

* Cadastrar produtos
* Visualizar produtos disponíveis
* Adicionar produtos ao carrinho
* Atualizar quantidade no carrinho
* Remover produtos do catálogo ou do carrinho
* Limpar o carrinho
* Finalizar a compra

Tudo foi feito de forma modular, com arquivos separados:

* index.html (estrutura da página)
* style.css (aparência da página)
* script.js (funcionalidades dinâmicas)

**2. Estrutura de Arquivos**

|-- index.html

|-- style.css

|-- script.js

**3. Estrutura HTML (index.html)**

O HTML define a **estrutura** da página:

**3.1 Cabeçalho (header)**

* Exibe o nome da loja.
* Possui um menu de navegação (Início, Produtos, Contato).
* Botão "Carrinho" que mostra o número de itens no carrinho.

<header>

<h1>🛒 Loja Virtual</h1>

<nav>...</nav>

<button id="open-cart">Carrinho (<span id="cart-count">0</span>)</button>

</header>

**3.2 Formulário de Cadastro de Produto (section#product-form)**

* Inputs para nome e preço.
* Botão para cadastrar produto.

<section id="product-form">

<h2>Cadastro de Produto</h2>

<input type="text" id="product-name" placeholder="Nome do Produto" />

<input type="number" id="product-price" placeholder="Preço do Produto" />

<button id="add-product">Cadastrar Produto</button>

</section>

**3.3 Listagem de Produtos (section#product-list)**

* Espaço onde os produtos cadastrados serão exibidos dinamicamente.

<section id="product-list">

<h2>Produtos Disponíveis</h2>

</section>

**3.4 Carrinho de Compras (aside#cart)**

* Lista de itens adicionados.
* Botões para limpar carrinho, finalizar compra e fechar o carrinho.

<aside id="cart" class="hidden">

<h2>Carrinho</h2>

<ul id="cart-items"></ul>

<div class="cart-footer">

<p id="cart-total">Total: R$ 0,00</p>

<button id="clear-cart">Limpar Carrinho</button>

<button id="checkout">Finalizar Compra</button>

<button id="close-cart">Fechar</button>

</div>

</aside>

**3.5 Rodapé (footer)**

* Rodapé simples com direitos autorais.

<footer>

<p>&copy; 2025 Loja Virtual Completa</p>

</footer>

**4. Estilização CSS (style.css)**

O CSS define a **aparência visual**:

* **Reset básico**:

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

* **Header**:
  + Fundo escuro
  + Menu horizontal
  + Botão do carrinho alinhado à direita
* **Formulário de Produto**:
  + Caixa branca com sombra
  + Inputs alinhados
* **Listagem de Produtos**:
  + Exibição em **grade** (flexbox)
  + Cada produto em um **card** com borda e botão
* **Carrinho**:
  + Sidebar fixa no lado direito
  + Animada com transform
  + Itens organizados em lista
* **Botões**:
  + Botões azuis para ações normais
  + Botões vermelhos para ações de remoção

**5. Funcionalidades JavaScript (script.js)**

O JavaScript traz **interatividade**:

**5.1 Cadastro de Produtos**

* Clicar no botão "Cadastrar Produto" cria um novo objeto de produto.
* Adiciona esse produto ao array products.

document.getElementById('add-product').addEventListener('click', () => {

const name = document.getElementById('product-name').value;

const price = parseFloat(document.getElementById('product-price').value);

products.push({ id: Date.now(), name, price });

renderProducts();

});

**5.2 Renderizar Produtos**

* Percorre o array products.
* Cria dinamicamente um card para cada produto, com:
  + Nome
  + Preço
  + Botão "Adicionar ao Carrinho"

function renderProducts() {

productList.innerHTML = '';

products.forEach(product => { ... });

}

**5.3 Carrinho de Compras**

**Adicionar Produto ao Carrinho**

* Se o produto já existe no carrinho, só aumenta a quantidade.
* Caso contrário, cria um novo item.

function addToCart(productId) { ... }

**Atualizar Carrinho**

* Atualiza a listagem dos itens no carrinho.
* Calcula o valor total dos produtos.

function updateCart() { ... }

**Alterar Quantidade de Itens**

* Permite alterar diretamente o valor no campo input dentro do carrinho.

function changeQuantity(productId, newQuantity) { ... }

**Remover Produto do Carrinho**

* Deleta um item específico do carrinho.

function removeFromCart(productId) { ... }

**Deletar Produto do Catálogo**

* Remove permanentemente o produto da lista e também do carrinho.

function deleteProduct(productId) { ... }

**Limpar Todo o Carrinho**

* Remove todos os itens do carrinho de uma vez.

document.getElementById('clear-cart').addEventListener('click', () => {

cart = [];

updateCart();

});

**Finalizar Compra**

* Exibe mensagem de sucesso e esvazia o carrinho.

document.getElementById('checkout').addEventListener('click', () => {

alert('Compra realizada com sucesso!');

cart = [];

updateCart();

});

**Mostrar/Esconder Carrinho**

* Ao clicar nos botões, o carrinho desliza da direita para a tela.

document.getElementById('open-cart').addEventListener('click', () => {

document.getElementById('cart').classList.add('show');

});

javascript

CopiarEditar

document.getElementById('close-cart').addEventListener('click', () => {

document.getElementById('cart').classList.remove('show');

});

**6. Conclusão**

Este projeto mostrou como construir **um mini e-commerce funcional** apenas com HTML, CSS e JavaScript puro.

Aprendemos:

* Criar páginas dinâmicas
* Trabalhar com arrays e objetos
* Manipular eventos DOM
* Fazer um carrinho de compras com CRUD completo

**E o melhor: sem usar frameworks externos!** 🚀