

SOFTWARE DE PEDIDOS PARA RESTAURANTE EM HTML, CSS, JAVASCRIPT E SQL

Orientador: Prof. Ranyere Lima

Docente do curso de Análise e Desenvolvimento de Sistemas.

Daylson Nascimento de Souza¹, Josué Nascimento Lucena¹, Leonardo Del'Aguila Rabelo¹, Gideon¹,
Graduandos em Análise e Desenvolvimento de Sistemas, Faculdade Martha Falcão Wyden, Adrianópolis, Manaus/AM, Brasil.

Introdução

O presente projeto apresenta o desenvolvimento de um software voltado para otimizar o processo de pedidos em restaurantes, utilizando as linguagens **HTML, CSS e JavaScript**, integradas a um banco de dados **SQL**. A proposta surgiu a partir da necessidade de automatizar e agilizar o atendimento ao cliente, permitindo que os pedidos sejam registrados, armazenados e acompanhados de forma digital e eficiente.

Objetivos

O objetivo geral do projeto é **criar um sistema funcional de pedidos online**, permitindo que clientes realizem seus pedidos e acompanhem o status em tempo real. Os objetivos específicos incluem:

- Desenvolver a interface web utilizando HTML e CSS para estrutura e design;
- Implementar funcionalidades interativas com JavaScript;
- Integrar o banco de dados SQL para armazenar pedidos, produtos e informações dos clientes;
- Oferecer uma experiência intuitiva tanto para o cliente quanto para o atendente do restaurante.

Metodologia

O sistema foi desenvolvido com base em uma arquitetura web composta por:

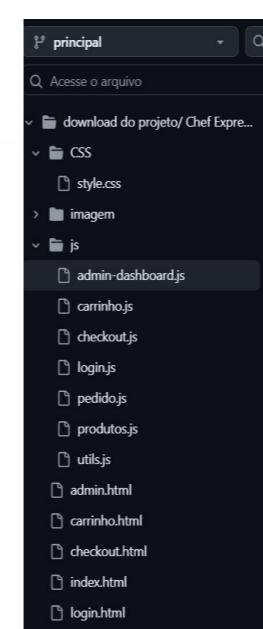
- HTML (HyperText Markup Language): responsável pela estrutura das páginas e organização dos elementos.

```
Code Blame 34 lines (31 loc) · 928 Bytes
1 <!doctype html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width,initial-scale=1" />
6   <title>Pedido - Chef Express</title>
7   <link rel="stylesheet" href="css/style.css" />
8 </head>
9 <body>
10  <header class="site-header">
11    <div class="container header-inner">
12      <a class="brand" href="index.html">Chef Express</a>
13      <a href="index.html" class="btn-link">Novo Pedido</a>
14    </div>
15  </header>
16
17  <main class="container main-content">
18    <h1>Status do Pedido:</h1>
19    <div id="pedidoBox" class="pedido-box"></div>
20  </main>
21
22  <footer class="site-footer">
23    <div class="container">
24      <small>© <span id="year"></span> Chef Express</small>
25    </div>
26  </footer>
27
28  <script src="js/utils.js"></script>
29  <script src="js/pedido.js"></script>
30  <script>
31    document.getElementById('year').textContent = new Date().getFullYear();
32  </script>
```

- CSS (Cascading Style Sheets): utilizado para a estilização e layout, garantindo boa experiência visual.

```
Código Culpa 138 linhas (124 locais) · 5,13 KB
1 .raiz{
2   --bg: #FFB6F6;
3   --cartao: #FFFFFF;
4   --texto: #1c1c1c;
5   --silenciado: #7a7a7a;
6   --sotaque: #d6d7e6;
7   --fronteira: #e6e5e3;
8   --raio: 12px;
9   --sombra: rgba(0,0,0,0.05);
10  familia de fontes: "Inter", interface do sistema, serif;
11 }
12
13 *(dimensionamento de caixa:caixa de borda;margem: 0;acolchoamento: 0; }
14 corpo{fundo: var(--bg);cor: var(--texto);altura da linha: 1.5; }
15
16 .recipiente{
17   largura: 100%;
18   largura-máxima: 1100px;
19   margin: auto;
20   acolchoamento: 20px 16px;
21 }
22 .silenciado{cor: var(--silenciado); }
23
24 /* CABEÇALHO */
25 .cabecalho do site{
26   fundo: var(--cartao);
27   borda inferior: 1px sólida var(--fronteira);
28   posição: pegajoso;
29   principal: 0;
30   índice z: 100;
31   sombra de caixa: 0 1px 2px var(--sombra);
32 }
```

JavaScript: linguagem responsável pela interatividade e manipulação dinâmica dos elementos da interface.



```
1 documento.adicionar ouvirte de eventos('DOMContentLoaded', função(){
2   documento.obterElementoPorID('ano').Conteudo do texto-novo Date().obterDiaCompleto();
3   constante pedidos = JSON.parse(localStorage.getItem('pedidos')) || [];
4   constante Tabela=documento.obterElementoPorID('tabelapedidos');
5   se (pedidos.length){Tabela.innerHTML = amazenamento local.obterItem('pedidos')[0].map(pedido=>`|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Menor pedido encontrado. | | | | | |
`)}
6   constante Desperdício=pedidos.filter(p=>p.status==='Cancelado').reduir((p, p)=>(p.total)[0]);
7   documento.obterElementoPorID('faturamentoTotal').Conteudo do texto="R$ " +Desperdício.paraFormatar();
8   documento.obterElementoPorID('totalPedidos').Conteudo do texto=pedidos.length;
9   Tabela.innerHTML =`;
10  pedidos para cada(p=>
11    constante tr=documento.createElement('tr');
12    tr.innerHTML="";
13    <td>p.eu.id</td>
14    <td>p.usuarioId||'-'</td>
15    <td>${p.nome} ${p.total}</td>
16    <td>${p.status}</td>
17    <td>${p.data}</td>
18    <td>
19      select data-id="p.eu.id" class="statusSelect">
20        ${['Recebido', 'Em', 'Saiu para entrega', 'Entrada', 'Cancelado'].map(s=>`<option value="${s}">${s}`).juntar('')}
21    </select>
22  </td>;
23  Tabela.adicionarFilho(tr);
24 });
25 Tabela.adicionar ouvirte de eventos('blur', função(e){
26   se(e.alvo.lista de classes.contains('statusSelect')){
27     constante eu =e.alvo.conjunto de dados.eu.id; constante novo=e.alvo.valor;
28     constante idx=pedidos.indice de busca(x=>x.eu.id==eu.id); set(idx==1) retornar;
29     pedidos[idx].status=novo; amazenamento local.definirItem('pedidos', JSON.stringify(pedidos)); showToast('Status');
30   }
31 });
32 
```

- Banco de dados SQL: utilizado para o armazenamento das informações de pedidos, produtos e usuários. O desenvolvimento seguiu o modelo incremental, permitindo aprimoramentos contínuos durante os testes e validações.

Resultados

O resultado final foi um sistema de pedidos online totalmente funcional, capaz de registrar pedidos, exibir o cardápio, gerenciar produtos e acompanhar o status das solicitações. O software apresentou ótimo desempenho, com interface amigável, design responsivo e baixo tempo de resposta, podendo ser facilmente adaptado para diferentes estabelecimentos gastronômicos.

Discussão/Conclusão

O projeto demonstrou a importância da integração entre **front-end** e **back-end** para soluções voltadas ao comércio local. O uso de tecnologias livres como HTML, CSS, JavaScript e SQL possibilitou o desenvolvimento de um sistema **econômico, eficiente e escalável**, atendendo às demandas de pequenos restaurantes. O aprendizado prático permitiu aos alunos compreender o processo completo de **criação, modelagem e implementação** de um software funcional voltado ao mercado real.

Referências

1. SILVA, Ricardo. *Desenvolvimento Web com HTML, CSS e JavaScript*. São Paulo: Novatec, 2022.
2. OLIVEIRA, João. *Banco de Dados SQL: Fundamentos e Práticas*. Rio de Janeiro: LTC, 2021.
3. <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide>
4. <https://www.alura.com.br/apostila-html-css-javascript>