

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 <div>Status do Pedido</div>
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/utlis.js"></script>
29 <script src="js/pedido.js"></script>
30 </script>
31 document.getElementById('year').textContent = new Date().getFullYear();
32 </html>
```

- 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: #f8f8f8;
3 --cartão: #ffffff;
4 --texto: #c1c1c1;
5 --silenciado: #7a7a7a;
6 --sotaque: #ded7c6;
7 --fronteira: #e6e5e3;
8 --raio: 12px;
9 --sombra: rgba(0,0,0,0.05);
10 família de fontes: "Inter",interface do sistema,sans 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 margem: 0auto;
20 acolchoamento: 20px 16px;
21 }
22 .silenciado{cor: var(--silenciado); }
23
24 /* CABEÇALHO */
25 .cabeçalho do site{
26 fundo: var(--cartão);
27 borda inferior: 1px solid 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.

```
principal
Q. Acesse o arquivo
download do projeto/ Chef Expre...
CSS
style.css
imagem
js
admin-dashboards
carinho.js
checkout.js
login.js
pedidos.js
produtos.js
utils.js
admin.html
carinho.html
checkout.html
index.html
login.html
1 documento.adicionarEvento de eventos('DOMContentLoaded',function(){
2 documento.getElementById('menu').Conteúdo do texto=menuData().obterDadosCompleto();
3 constante pedidos=JSON.amalgamar(armazenamento local.obterItens('pedidos'))['IT']; constante Tabela=document.getElementById('tabelaPedidos');
4 se(pedidos.comprimento){ Tabela.HTML interno='<tr><td colspan="5" class="metad">Nenhum pedido encontrado.</td></tr>'; retornar; }
5 constante Despertador=pedidos.filtro(p=p.status==='Cancelado').reduzir((s,p)=>s+(p.total||0),0);
6 documento.getElementById('faturamentoTotal').Conteúdo do texto=RS '<Despertador.parafixado(2).substituir('.',',')>';
7 documento.getElementById('totalPedidos').Conteúdo do texto=pedidos.comprimento;
8 documento.getElementById('totalCancelado').Conteúdo do texto=pedidos.filtro(p=p.status==='Cancelado').comprimento;
9 Tabela.HTML interno='';
10 pedidos.para cada(p)=>{
11 constante tr=document.createElement('tr');
12 tr.HTML interno=
13 <td>${p.id}</td>
14 <td>${p.ordenar}</td>
15 <td>${(numero(p.total||0).parafixado(2).substituir('.',',')).}</td>
16 <td>span class="status">${p.status}</td></td>
17 <td>${(new Date(p.criado_em).parafixadoString('pt-BR'))}</td>
18 <td>
19 <select data-id="${p.id}" class="statusSelect">
20 <option value="In" selected="selected">Iniciado
21 </select>
22 </td>
23 }
24 Tabela.adicionarFilho(tr);
25 }
26 Tabela.adicionarEvento de eventos('click',function(){
27 se(e.alvo.lista de classes contém 'statusSeletor')
28 constante eu ia=e.alvo.conjunto de dados.eu ia; constante novo=e.alvo.valor;
29 constante id=pedido.id de busca(=>x.eu ia==eu ia); se(id===-1) retornar;
30 pedidos[id].status=novo; armazenamento local.definirItens('pedidos',JSON.stringify(car(pedidos))); showToast('Status!'); setTimeout(()=>localização.recarregar(
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>