

Tema Integrador III

Atividade 1: Estudo de Caso

Pesquisa de Empresa

Para o estudo de caso, selecionaremos uma pequena mercearia local fictícia chamada "Zézinho". O público-alvo são pequenas empresas locais que desejam oferecer um serviço de entrega a domicílio para seus clientes.

Levantamento de Requisitos com o Cliente

Durante a entrevista com o dono do "Zézinho", foram identificados os seguintes requisitos e usuários do sistema:

- **Usuários:**
 - Clientes (primeira vez e retornantes)
 - Administrador (dono do mercado)

Estudo de Caso

Área do Negócio

O "Zézinho" é uma pequena mercearia que atende os moradores da vizinhança. O mercado oferece uma variedade de produtos alimentícios e itens de uso diário.

Objetivo do Projeto

- **Objetivo Geral:** Criar um site que facilite as compras online e a entrega de produtos do mercado aos clientes.
- **Objetivos Específicos:**
 - a. Permitir o cadastro de novos clientes.
 - b. Facilitar o login de clientes já cadastrados.
 - c. Disponibilizar um catálogo de produtos para seleção e compra.
 - d. Implementar um sistema de carrinho de compras.
 - e. Facilitar a confirmação de compras e envio de informações via WhatsApp para o dono do mercado.

Justificativa

A criação do site é necessária para modernizar o serviço do "Zézinho", permitindo que os clientes façam suas compras de maneira prática e rápida sem sair de casa. Isso ajudará a aumentar as vendas e a satisfação dos clientes, oferecendo conveniência e melhorando o atendimento.

Cronograma

Etapa	Duração
Levantamento de Requisitos	1 semana
Análise e Planejamento	1 semana
Desenvolvimento	3 semanas
Testes	1 semana
Implementação	1 semana
Total	7 semanas

Metodologia do Projeto

Para este projeto, utilizaremos a metodologia Ágil (Scrum), pois permite maior flexibilidade e interação constante com o cliente, ajustando o desenvolvimento conforme as necessidades.

Descrição do Projeto

- **Elaboração:** O projeto será elaborado em ciclos de sprints de 1 semana, onde cada sprint focará em uma parte específica do sistema (cadastro, login, catálogo de produtos, etc.).
- **Controle:** O progresso será monitorado por meio de reuniões diárias de stand-up, revisões de sprint e retrospectivas para avaliar o que foi feito e como melhorar.

Atividade 2: Levantamento dos Requisitos

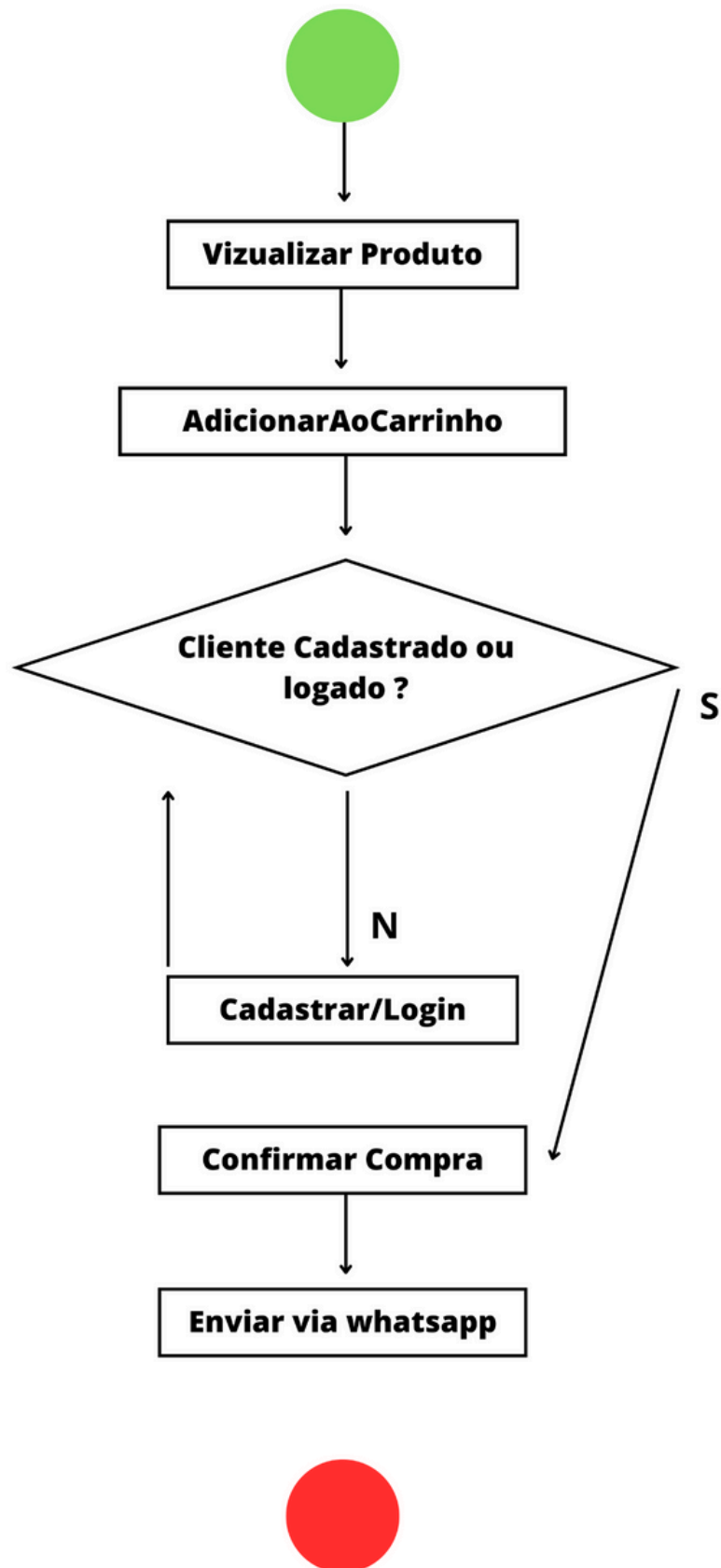
Funcionais

1. Cadastro de novos clientes.
2. Login de clientes cadastrados.
3. Visualização de catálogo de produtos.
4. Adição de produtos ao carrinho.
5. Confirmação de compra.
6. Envio de informações via WhatsApp.

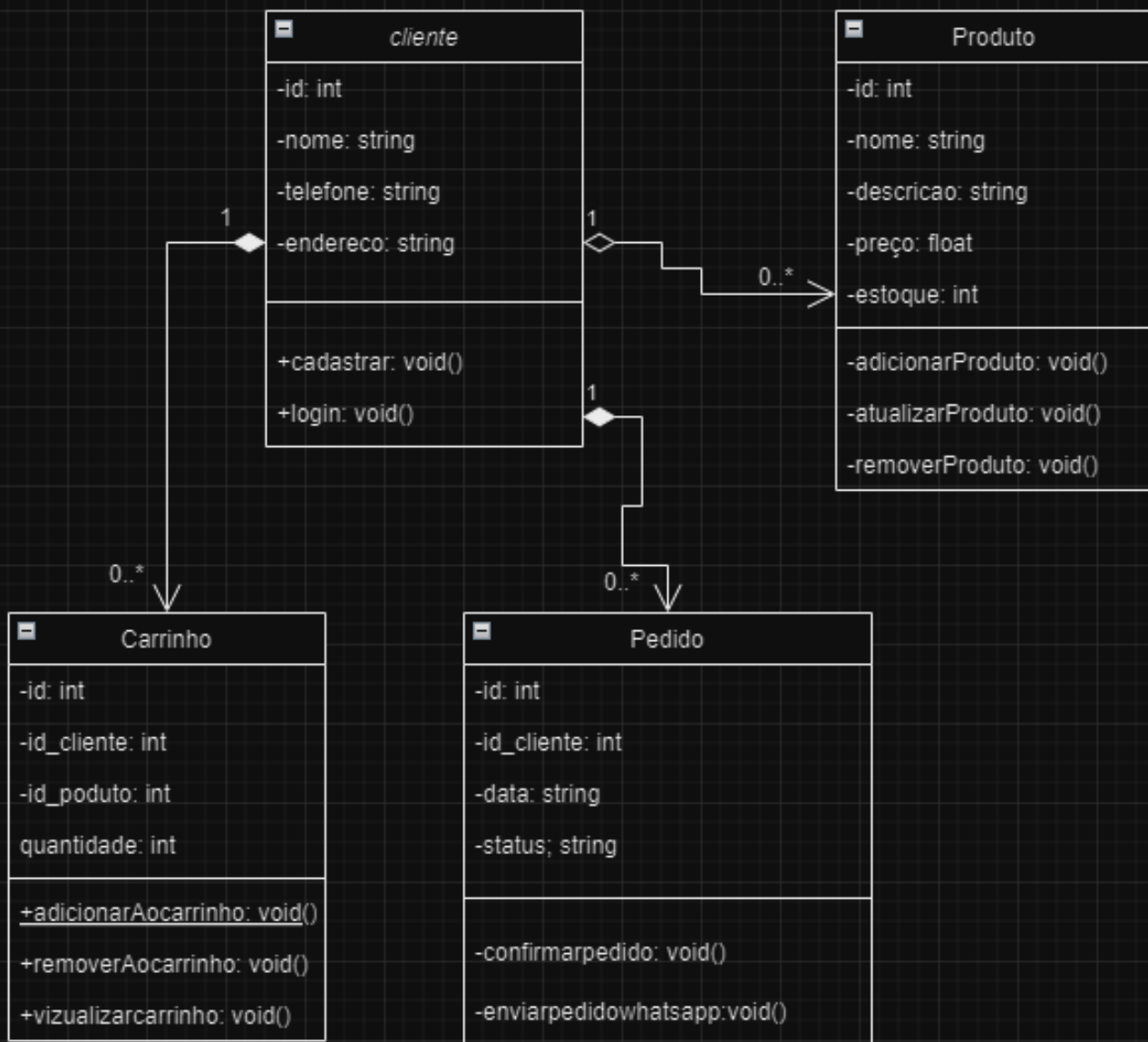
Não Funcionais

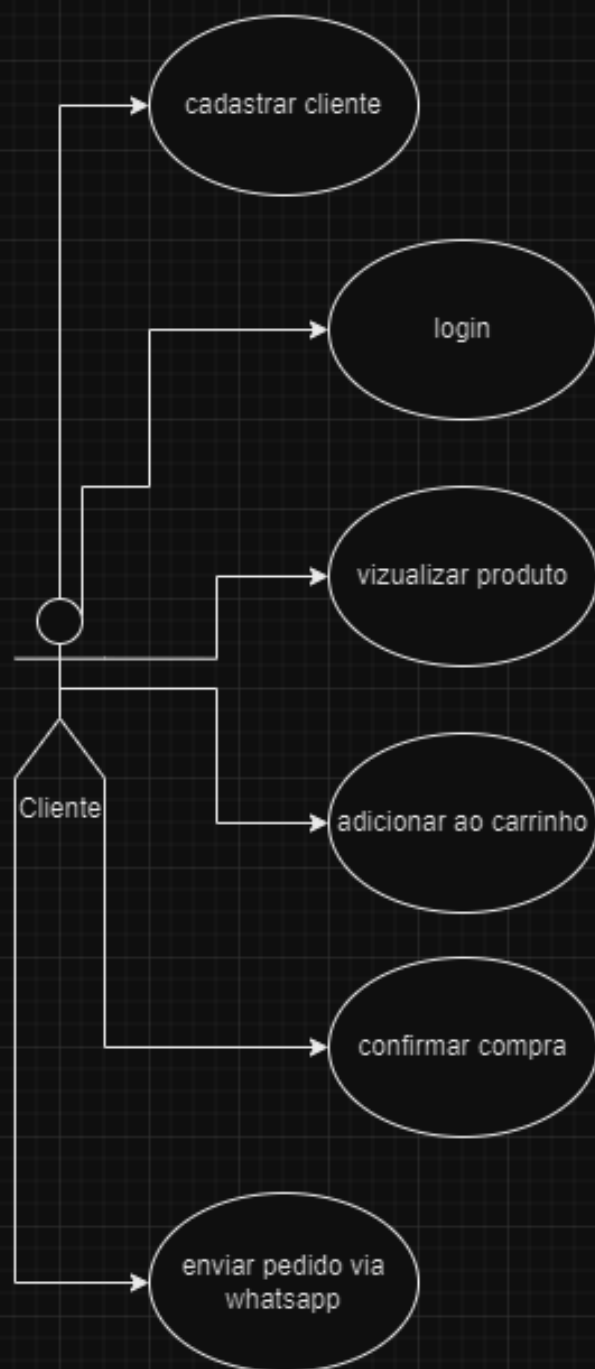
1. Interface amigável e intuitiva.
2. Responsividade para dispositivos móveis.
3. Segurança dos dados dos clientes.
4. Desempenho rápido e eficiente.

Atividade 3: Mapeamento dos Processos



Atividade 4: Diagrama de Classes e Diagrama de Caso de Uso





Atividade 6: Criação do Banco de Dados

Dicionário de Dados

Tabelas principais:

1. Clientes: id, nome, telefone, endereço, senha.
2. Produtos: id, nome, descrição, preço, estoque.
3. Carrinho: id, id_cliente, id_produto, quantidade.
4. Pedidos: id, id_cliente, data, status.

mercado.zézinho/mercado/cliente - HeidiSQL 12.6.0.6765

Arquivo Editar Pesquisar Consulta Ferramentas Ir para Ajuda

Filtro de banco de dados Filtro de tabela Servidor: 127.0.0.1 Banco de dados: mercado Tabela: cliente Dados Consulta

mercado.zézinho

- information_schema
- mercado 112,0 KiB
 - carrinho 48,0 KiB
 - cliente 16,0 KiB
 - pedido 32,0 KiB
 - produto 16,0 KiB
- mysql
- performance_schema
- reservabicicletas
- sys

mercado.cliente: 2 registros totais (exact)

#	id	nome	telefone	endereco	senha
1	1	João Silva	123456789	Rua A, 123	senha123
2	2	Maria Oliveira	987654321	Rua B, 456	senha456

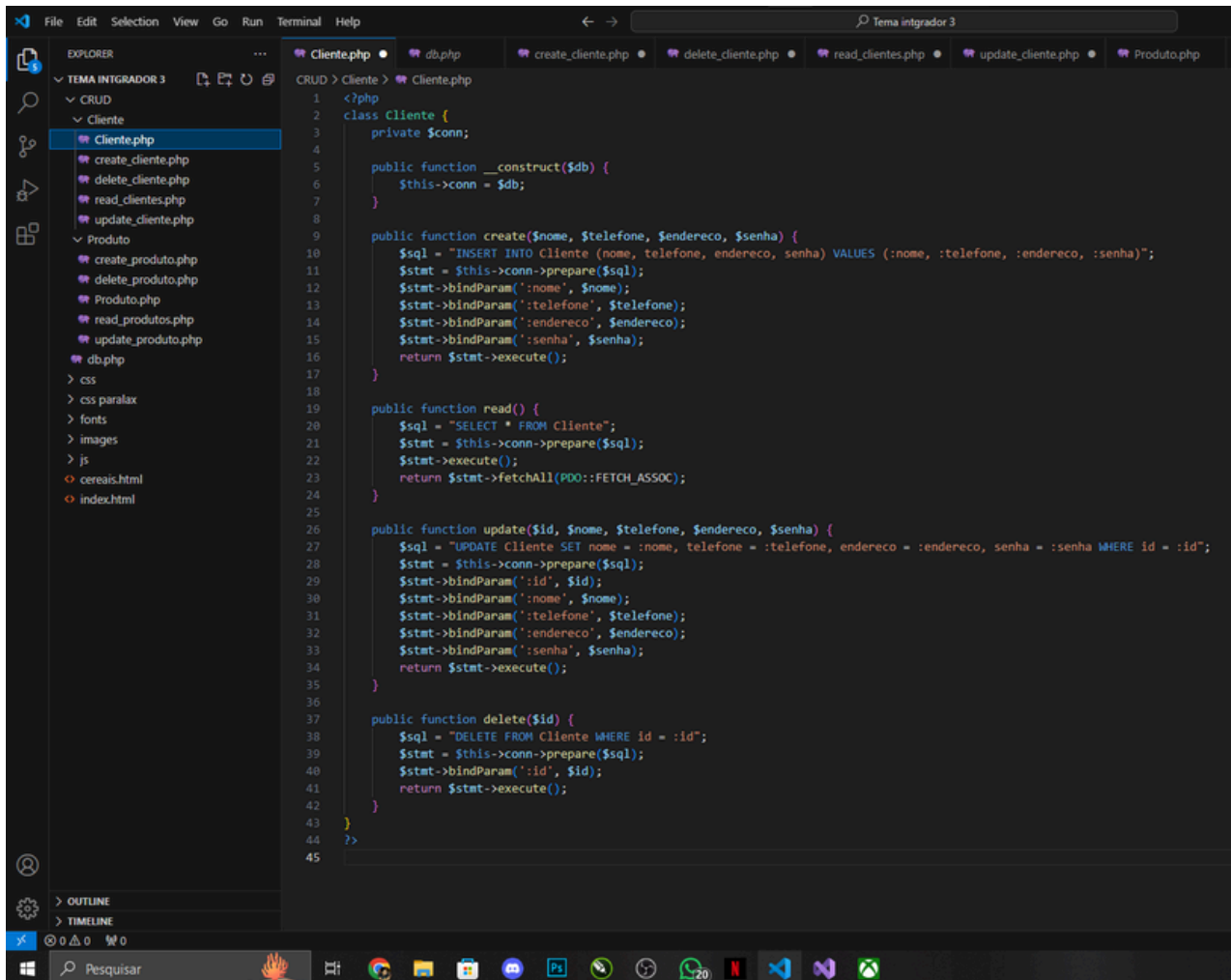
X Filtros: Expressão regular

```
18 SHOW PROCEDURE STATUS WHERE `Db`='mercado';
19 SHOW TRIGGERS FROM `mercado`;
20 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.EVENTS WHERE `EVENT_SCHEMA`='mercado';
21 /* Corregendo arquivos "C:\Users\ital\AppData\Roaming\HeidiSQL\Backups\query-tab-.sql" (1,9 KiB) na aba de consulta #1 */
22 /* Reset height of query tab #1 to give result area vertical space. */
23 /* Redimensionando controles para tela DPI: 100% */
24 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='carrinho' ORDER BY ORDINAL_POSITION;
25 SHOW INDEXES FROM `carrinho` FROM `mercado`;
26 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='carrinho' AND REFERENCED_TABLE_NAME IS NOT NULL;
27 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='carrinho' AND REFERENCED_TABLE_NAME IS NOT NULL;
28 SHOW ENGINES;
29 SHOW COLLATION;
30 SHOW CREATE TABLE `mercado`.`carrinho`;
31 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='carrinho';
32 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='cliente' ORDER BY ORDINAL_POSITION;
33 SHOW INDEXES FROM `cliente` FROM `mercado`;
34 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='cliente' AND REFERENCED_TABLE_NAME IS NOT NULL;
35 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='cliente' AND REFERENCED_TABLE_NAME IS NOT NULL;
36 SHOW CREATE TABLE `mercado`.`cliente`;
37 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='cliente';
38 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='pedido' ORDER BY ORDINAL_POSITION;
39 SHOW INDEXES FROM `pedido` FROM `mercado`;
40 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='pedido' AND REFERENCED_TABLE_NAME IS NOT NULL;
41 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='pedido' AND REFERENCED_TABLE_NAME IS NOT NULL;
42 SHOW CREATE TABLE `mercado`.`pedido`;
43 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='pedido';
44 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='produto' ORDER BY ORDINAL_POSITION;
45 SHOW INDEXES FROM `produto` FROM `mercado`;
46 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='produto' AND REFERENCED_TABLE_NAME IS NOT NULL;
47 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='mercado' AND TABLE_NAME='produto' AND REFERENCED_TABLE_NAME IS NOT NULL;
48 SHOW CREATE TABLE `mercado`.`produto`;
49 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='mercado' AND TABLE_NAME='produto';
50 SELECT `id`, `nome`, LEFT(`descricao`, 256), `preco`, `estoque` FROM `mercado`.`produto` LIMIT 1000;
51 SHOW CREATE TABLE `mercado`.`carrinho`;
52 SELECT * FROM `mercado`.`carrinho` LIMIT 1000;
53 SHOW CREATE TABLE `mercado`.`cliente`;
54 SELECT * FROM `mercado`.`cliente` LIMIT 1000;
```

r1: c1 Conectado: 00:02 h MariaDB 11.3.2

Atividade 7: Implementação do CRUD

Exemplo de CRUD para a Classe Cliente e Produto



```
1 <?php
2 class Cliente {
3     private $conn;
4
5     public function __construct($db) {
6         $this->conn = $db;
7     }
8
9     public function create($nome, $telefone, $endereco, $senha) {
10         $sql = "INSERT INTO Cliente (nome, telefone, endereco, senha) VALUES (:nome, :telefone, :endereco, :senha)";
11         $stmt = $this->conn->prepare($sql);
12         $stmt->bindParam(':nome', $nome);
13         $stmt->bindParam(':telefone', $telefone);
14         $stmt->bindParam(':endereco', $endereco);
15         $stmt->bindParam(':senha', $senha);
16         return $stmt->execute();
17     }
18
19     public function read() {
20         $sql = "SELECT * FROM Cliente";
21         $stmt = $this->conn->prepare($sql);
22         $stmt->execute();
23         return $stmt->fetchAll(PDO::FETCH_ASSOC);
24     }
25
26     public function update($id, $nome, $telefone, $endereco, $senha) {
27         $sql = "UPDATE Cliente SET nome = :nome, telefone = :telefone, endereco = :endereco, senha = :senha WHERE id = :id";
28         $stmt = $this->conn->prepare($sql);
29         $stmt->bindParam(':id', $id);
30         $stmt->bindParam(':nome', $nome);
31         $stmt->bindParam(':telefone', $telefone);
32         $stmt->bindParam(':endereco', $endereco);
33         $stmt->bindParam(':senha', $senha);
34         return $stmt->execute();
35     }
36
37     public function delete($id) {
38         $sql = "DELETE FROM Cliente WHERE id = :id";
39         $stmt = $this->conn->prepare($sql);
40         $stmt->bindParam(':id', $id);
41         return $stmt->execute();
42     }
43 }
44 ?>
45
```


File Edit Selection View Go Run Terminal Help

EXPLORER

- TEMA INTGRADOR 3
 - CRUD
 - Cliente
 - Cliente.php
 - create_cliente.php
 - delete_cliente.php
 - read_clientes.php
 - update_cliente.php
 - Produto
 - create_produto.php
 - delete_produto.php
 - Produto.php
 - read_produtos.php
 - update_produto.php
 - db.php
 - css
 - css paralax
 - fonts
 - images
 - js
 - cereais.html
 - index.html

CRUD > Cliente > create_cliente.php

```
1 <?php
2 include 'db.php';
3 include 'Cliente.php';
4
5 $cliente = new Cliente($conn);
6
7 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8     $nome = $_POST['nome'];
9     $telefone = $_POST['telefone'];
10    $endereco = $_POST['endereco'];
11    $senha = $_POST['senha'];
12
13    if ($cliente->create($nome, $telefone, $endereco, $senha)) {
14        echo "Cliente criado com sucesso!";
15    } else {
16        echo "Erro ao criar cliente.";
17    }
18 }
19 ?>
20
21 <form method="post" action="create_cliente.php">
22     Nome: <input type="text" name="nome" required><br>
23     Telefone: <input type="text" name="telefone" required><br>
24     Endereço: <input type="text" name="endereco" required><br>
25     Senha: <input type="password" name="senha" required><br>
26     <input type="submit" value="Criar Cliente">
27 </form>
28
```

File Edit Selection View Go Run Terminal Help

EXPLORER

- TEMA INTGRADOR 3
 - CRUD
 - Cliente
 - Cliente.php
 - create_cliente.php
 - delete_cliente.php
 - read_clientes.php
 - update_cliente.php
 - Produto
 - create_produto.php
 - delete_produto.php
 - Produto.php
 - read_produtos.php
 - update_produto.php
 - db.php
 - css
 - css paralax
 - fonts
 - images
 - js
 - cereais.html
 - index.html

CRUD > Cliente > read_clientes.php

```
1 <?php
2 include 'db.php';
3 include 'Cliente.php';
4
5 $cliente = new Cliente($conn);
6 $clientes = $cliente->read();
7 ?>
8
9 <table border="1">
10     <tr>
11         <th>ID</th>
12         <th>Nome</th>
13         <th>Telefone</th>
14         <th>Endereço</th>
15     </tr>
16     <?php foreach ($clientes as $cli) { ?>
17         <tr>
18             <td><?php echo $cli['id']; ?></td>
19             <td><?php echo $cli['nome']; ?></td>
20             <td><?php echo $cli['telefone']; ?></td>
21             <td><?php echo $cli['endereco']; ?></td>
22         </tr>
23     <?php } ?>
24 </table>
25
```

The screenshot shows the Visual Studio Code editor with a project named "TEMA INTGRADOR 3". The Explorer sidebar on the left shows a directory structure with folders "CRUD" and "css", and subfolders "Cliente" and "Produto". The "update_cliente.php" file is selected in the Explorer and in the main editor. The code in the editor is as follows:

```
1 <?php
2 include 'db.php';
3 include 'Cliente.php';
4
5 $cliente = new Cliente($conn);
6
7 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8     $id = $_POST['id'];
9     $nome = $_POST['nome'];
10    $telefone = $_POST['telefone'];
11    $endereco = $_POST['endereco'];
12    $senha = $_POST['senha'];
13
14    if ($cliente->update($id, $nome, $telefone, $endereco, $senha)) {
15        echo "Cliente atualizado com sucesso!";
16    } else {
17        echo "Erro ao atualizar cliente.";
18    }
19 } else {
20     $id = $_GET['id'];
21     $clientes = $cliente->read();
22     $cli = null;
23     foreach ($clientes as $c) {
24         if ($c['id'] == $id) {
25             $cli = $c;
26             break;
27         }
28     }
29 }
30 ?>
31
32 <?php if ($cli) { ?>
33     <form method="post" action="update_cliente.php">
34         <input type="hidden" name="id" value="<?php echo $cli['id']; ?>">
35         Nome: <input type="text" name="nome" value="<?php echo $cli['nome']; ?>" required><br>
36         Telefone: <input type="text" name="telefone" value="<?php echo $cli['telefone']; ?>" required><br>
37         Endereço: <input type="text" name="endereco" value="<?php echo $cli['endereco']; ?>" required><br>
38         Senha: <input type="password" name="senha" value="<?php echo $cli['senha']; ?>" required><br>
39         <input type="submit" value="Atualizar Cliente">
40     </form>
41 <?php } else { ?>
42     Cliente não encontrado.
43 <?php } ?>
44
```

The screenshot shows the Visual Studio Code editor with the same project "TEMA INTGRADOR 3". The Explorer sidebar on the left shows the same directory structure. The "delete_cliente.php" file is selected in the Explorer and in the main editor. The code in the editor is as follows:

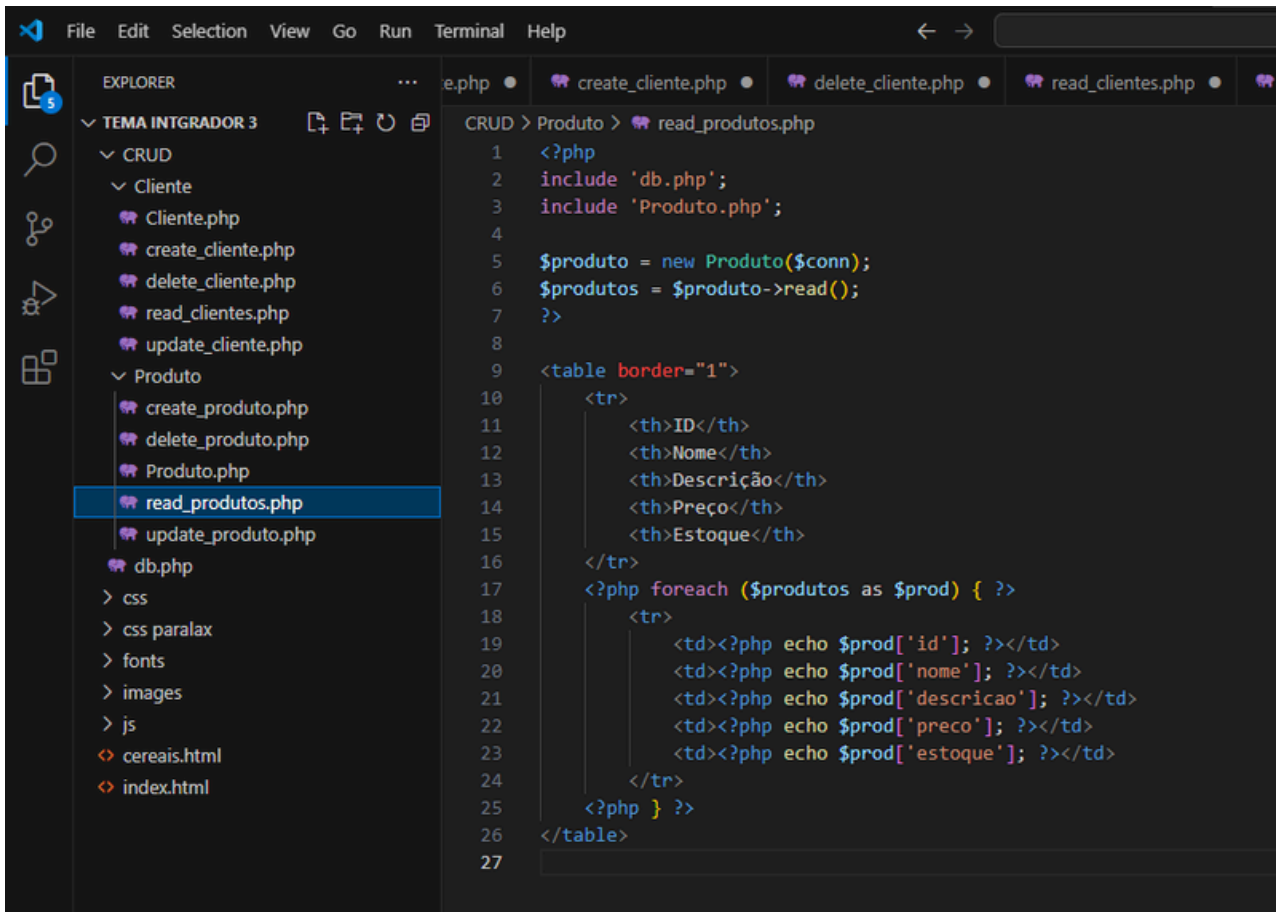
```
1 <?php
2 include 'db.php';
3 include 'Cliente.php';
4
5 $cliente = new Cliente($conn);
6
7 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8     $id = $_POST['id'];
9
10    if ($cliente->delete($id)) {
11        echo "Cliente deletado com sucesso!";
12    } else {
13        echo "Erro ao deletar cliente.";
14    }
15 } else {
16     $id = $_GET['id'];
17 }
18 ?>
19
20 <form
21
```

This screenshot shows the Visual Studio Code editor with the 'Produto.php' file open. The Explorer sidebar on the left shows a project structure for 'TEMA INTEGRADOR 3' with folders for 'CRUD' and 'Cliente'. The 'Produto.php' file contains a PHP class with the following methods:

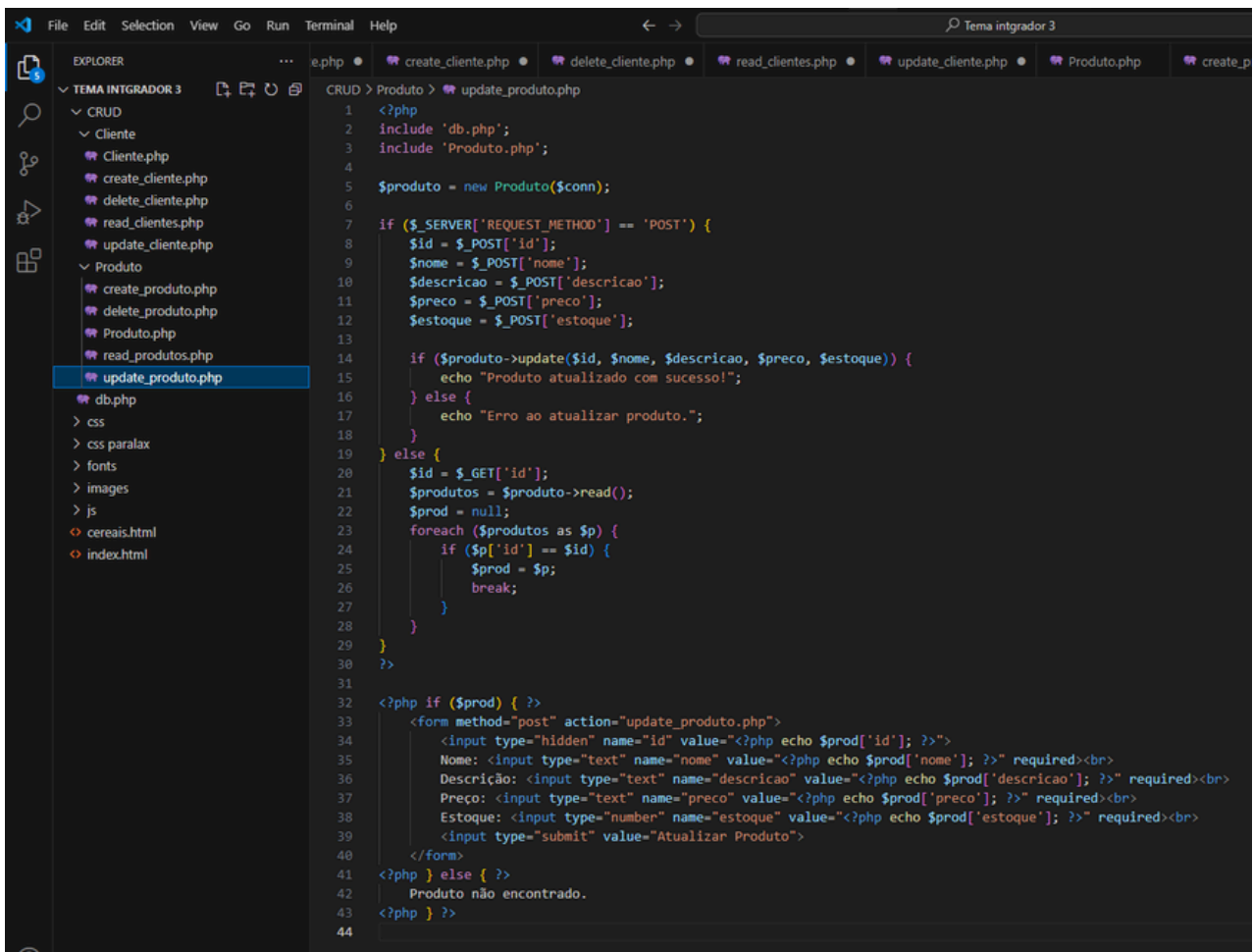
```
1 <?php
2 class Produto {
3     private $conn;
4
5     public function __construct($db) {
6         $this->conn = $db;
7     }
8
9     public function create($nome, $descricao, $preco, $estoque) {
10        $sql = "INSERT INTO Produto (nome, descricao, preco, estoque) VALUES (:nome, :descricao, :preco, :estoque)";
11        $stmt = $this->conn->prepare($sql);
12        $stmt->bindParam(':nome', $nome);
13        $stmt->bindParam(':descricao', $descricao);
14        $stmt->bindParam(':preco', $preco);
15        $stmt->bindParam(':estoque', $estoque);
16        return $stmt->execute();
17    }
18
19    public function read() {
20        $sql = "SELECT * FROM Produto";
21        $stmt = $this->conn->prepare($sql);
22        $stmt->execute();
23        return $stmt->fetchAll(PDO::FETCH_ASSOC);
24    }
25
26    public function update($id, $nome, $descricao, $preco, $estoque) {
27        $sql = "UPDATE Produto SET nome = :nome, descricao = :descricao, preco = :preco, estoque = :estoque WHERE id = :id";
28        $stmt = $this->conn->prepare($sql);
29        $stmt->bindParam(':id', $id);
30        $stmt->bindParam(':nome', $nome);
31        $stmt->bindParam(':descricao', $descricao);
32        $stmt->bindParam(':preco', $preco);
33        $stmt->bindParam(':estoque', $estoque);
34        return $stmt->execute();
35    }
36
37    public function delete($id) {
38        $sql = "DELETE FROM Produto WHERE id = :id";
39        $stmt = $this->conn->prepare($sql);
40        $stmt->bindParam(':id', $id);
41        return $stmt->execute();
42    }
43 }
44 ?>
```

This screenshot shows the Visual Studio Code editor with the 'create_produto.php' file open. The Explorer sidebar on the left shows the same project structure, with 'create_produto.php' selected under the 'Produto' folder. The file contains the following PHP code and HTML form:

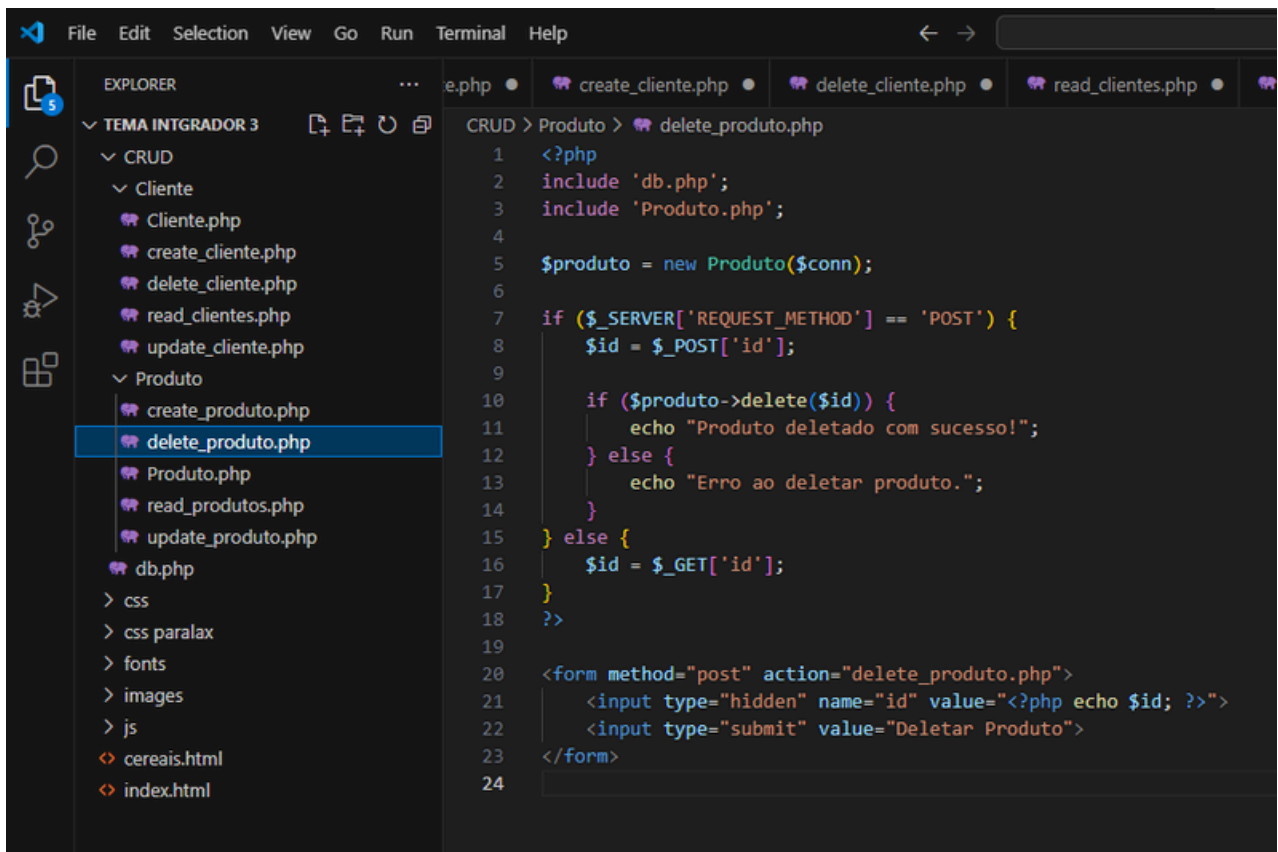
```
1 <?php
2 include 'db.php';
3 include 'Produto.php';
4
5 $produto = new Produto($conn);
6
7 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8     $nome = $_POST['nome'];
9     $descricao = $_POST['descricao'];
10    $preco = $_POST['preco'];
11    $estoque = $_POST['estoque'];
12
13    if ($produto->create($nome, $descricao, $preco, $estoque)) {
14        echo "Produto criado com sucesso!";
15    } else {
16        echo "Erro ao criar produto.";
17    }
18 }
19 ?>
20
21 <form method="post" action="create_produto.php">
22     Nome: <input type="text" name="nome" required><br>
23     Descrição: <input type="text" name="descricao" required><br>
24     Preço: <input type="text" name="preco" required><br>
25     Estoque: <input type="number" name="estoque" required><br>
26     <input type="submit" value="Criar Produto">
27 </form>
28
```



```
1 <?php
2 include 'db.php';
3 include 'Produto.php';
4
5 $produto = new Produto($conn);
6 $produtos = $produto->read();
7 ?>
8
9 <table border="1">
10     <tr>
11         <th>ID</th>
12         <th>Nome</th>
13         <th>Descrição</th>
14         <th>Preço</th>
15         <th>Estoque</th>
16     </tr>
17     <?php foreach ($produtos as $prod) { ?>
18         <tr>
19             <td><?php echo $prod['id']; ?></td>
20             <td><?php echo $prod['nome']; ?></td>
21             <td><?php echo $prod['descricao']; ?></td>
22             <td><?php echo $prod['preco']; ?></td>
23             <td><?php echo $prod['estoque']; ?></td>
24         </tr>
25     <?php } ?>
26 </table>
27
```



```
1 <?php
2 include 'db.php';
3 include 'Produto.php';
4
5 $produto = new Produto($conn);
6
7 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
8     $id = $_POST['id'];
9     $nome = $_POST['nome'];
10    $descricao = $_POST['descricao'];
11    $preco = $_POST['preco'];
12    $estoque = $_POST['estoque'];
13
14    if ($produto->update($id, $nome, $descricao, $preco, $estoque)) {
15        echo "Produto atualizado com sucesso!";
16    } else {
17        echo "Erro ao atualizar produto.";
18    }
19 } else {
20     $id = $_GET['id'];
21     $produtos = $produto->read();
22     $prod = null;
23     foreach ($produtos as $p) {
24         if ($p['id'] == $id) {
25             $prod = $p;
26             break;
27         }
28     }
29 }
30 ?>
31
32 <?php if ($prod) { ?>
33     <form method="post" action="update_produto.php">
34         <input type="hidden" name="id" value="<?php echo $prod['id']; ?>">
35         Nome: <input type="text" name="nome" value="<?php echo $prod['nome']; ?>" required><br>
36         Descrição: <input type="text" name="descricao" value="<?php echo $prod['descricao']; ?>" required><br>
37         Preço: <input type="text" name="preco" value="<?php echo $prod['preco']; ?>" required><br>
38         Estoque: <input type="number" name="estoque" value="<?php echo $prod['estoque']; ?>" required><br>
39         <input type="submit" value="Atualizar Produto">
40     </form>
41 <?php } else { ?>
42     Produto não encontrado.
43 <?php } ?>
44
```



Atividade 8: Publique esse site em uma hospedagem gratuita

Site: <https://italowebdesigner.github.io/Tema-Integrador-III/>

GitHub: <https://github.com/ItaloWebdesigner/Tema-Integrador-III>

Apresentação do site para o cliente – relatório do feedback

O design do site está atrativo e funcional. A navegação é intuitiva e isso facilita para os clientes. Agradeço pelo empenho e dedicação demonstrados ao longo do projeto. Estou muito ansioso para ver o site em pleno funcionamento.