**https://medium.com/@josianebarros/criando-sistema-de-cadastro-de-produtos-com-php-e-mysql-462f3e99be66**

**Criando sistema de cadastro de produtos com PHP e MySQL**

[[](https://medium.com/@josianebarros?source=post_page-----462f3e99be66--------------------------------)](https://medium.com/@josianebarros?source=post_page-----462f3e99be66--------------------------------)

[Josiane Barros](https://medium.com/@josianebarros?source=post_page-----462f3e99be66--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fb1fcdcaca9bc&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40josianebarros%2Fcriando-sistema-de-cadastro-de-produtos-com-php-e-mysql-462f3e99be66&user=Josiane+Barros&userId=b1fcdcaca9bc&source=post_page-b1fcdcaca9bc----462f3e99be66---------------------post_header-----------)

6 min read

·

Sep 25, 2023

Neste artigo eu irei ensinar como criar um sistema de cadastro de produtos usando HTML, CSS, JavaScript, PHP e o MySQL.

Eu usei o XAMPP neste tutorial para executar o sistema, vou considerar que você já tenha instalado no seu computador ele ou algo que cumpra a sua função.

**Estrutura do banco de dados**

Crie um banco de dados com o nome **cadastroProdutos**, e dentro dele crie a tabela **produtos**.

Para criar a tabela use o seguinte código SQL:

CREATE TABLE produtos (  
 id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 nome VARCHAR(255) NOT NULL,  
 quantidade INT NOT NULL,  
 valor DECIMAL(10,2) NOT NULL  
)

**Explicação do código:**

**CREATE TABLE produtos** — Cria uma tabela com o nome **produtos**.

**id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY** — Cria uma coluna chamada **id** dentro da tabela, que irá armazenar números inteiros (**INT**) de até 6 dígitos. O **UNSIGNED** específica que a coluna só pode conter números positivos. O **AUTO\_INCREMENT** significa que o valor do **id** será incrementado automaticamente toda vez que uma nova linha for adicionada na tabela. Já o **PRIMARY KEY** define que a coluna (**id**) será a chave primária da tabela, isso significa que cada valor na coluna deve ser único e não nulo.

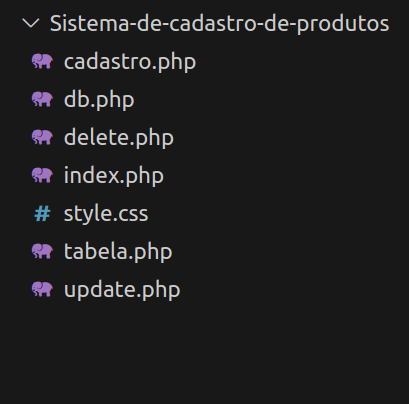
**nome VARCHAR(255) NOT NULL** — Cria uma coluna chamada **nome**, que pode armazenar até 255 caracteres. O **NOT NULL** significa que a coluna deve ter algum valor, ou seja, ela não pode ficar em branco.

**quantidade INT NOT NULL** — Cria uma coluna chamada **quantidade**, que irá armazenar números inteiros (**INT**). Assim como a coluna **nome**, ela não pode ficar em branco.

**valor DECIMAL(10,2) NOT NULL** — Cria a coluna **valor**, que pode armazenar números decimais, o número 10 representa a quantidade máxima de números que podem ser armazenados, já o número 2 representa a quantidade de números que podem ser armazenados à direita do ponto decimal. Assim como as colunas anteriores, essa coluna não pode ter um valor nulo.

**Estrutura dos arquivos**

Todos os arquivos vão ficar dentro de uma só pasta:



estrutura dos arquivos

**Conexão com o banco de dados**

O arquivo **db.php** será responsável por fazer a conexão com o banco de dados.

<?php  
  
// Informações de login  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "cadastroProdutos";  
  
// Conexão com o banco de dados  
$connection = new mysqli($servername, $username, $password, $dbname);  
  
if ($connection->connect\_error){  
 die($connection->connect\_error);  
}  
  
?>

Modifique as informações de login, caso necessário.

**Criando o arquivo index.html**

Agora, o próximo passo é criar um formulário com os campos: **Nome**, **Quantidade** e **Valor**.

<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Sistema de cadastro de produtos</title>  
</head>  
<body>  
 <h1>Sistema de cadastro de produtos</h1>  
 <form action="cadastro.php" method="post">  
 <label>Nome do produto:</label>  
 <input type="text" name="nome">  
  
 <label>Quantidade:</label>  
 <input type="number" min="0" name="quantidade">  
  
 <label>Valor por unidade:</label>  
 <input type="number" min="0" name="valor">  
  
 <input type="submit" value="Cadastrar produto">  
 </form>  
  
 <p>Produtos cadastrados:</p>  
  
</body>  
</html>

**Cadastro de produtos**

Quando o botão “Cadastrar produto” for clicado os dados adicionados no formulário serão enviados para o arquivo **cadastro.php**, que irá pegar os valores dos campos **nome**, **quantidade** e **valor** e inserir no banco de dados.

Salve o seguinte código dentro do arquivo **cadastro.php**

<?php  
include "db.php";  
  
// Adicionar as informações do produto no banco de dados  
if ($\_SERVER["REQUEST\_METHOD"] == "POST"){  
 $nome = $\_POST["nome"];  
 $quantidade = $\_POST["quantidade"];  
 $valor = $\_POST["valor"];  
 $insertDados = "INSERT INTO produtos(nome, quantidade, valor) VALUES ('$nome', '$quantidade', '$valor')";  
 $connection->query($insertDados);  
}  
  
$url = "index.php";  
  
header('Location: '.$url);  
  
$connection->close();  
?>

**Exibindo os produtos**

O arquivo **tabela.php** irá criar uma tabela para exibir o conteúdo do banco de dados na página.

<?php  
include "db.php";  
  
// Exibi os produtos  
$selectProdutos = "SELECT \* FROM produtos";  
$queryProdutos = $connection->query($selectProdutos);  
  
function exibirProdutos(){  
 global $queryProdutos;  
 if ($queryProdutos->num\_rows > 0){  
 echo "  
 <table>  
 <tr>  
 <th>Nome</th>  
 <th>Quantidade</th>  
 <th>Valor por unidade (R$)</th>  
 </tr>  
 ";  
 while($row = $queryProdutos->fetch\_assoc()){  
 echo "<tr>";  
 echo "<td>" . "<input type='text' class='$row[id]' value='$row[nome]'>" . "</td>";  
 echo "<td>" . "<input type='text' class='$row[id]' value='$row[quantidade]'>" . "</td>";  
 echo "<td>" . "<input type='text' class='$row[id]' value='$row[valor]'>" . "</td>";  
 echo "<td>" . "<input type='submit' value='Salvar alterações'>" . "</td>";  
 echo "<td>" . "<input type='submit' value='Apagar'>" . "</td>";  
 echo "</tr>";  
 }  
 echo "</table>";  
 } else{  
 echo "<p>Sem produtos cadastrados</p>";  
 }  
}  
  
$connection->close();  
?>

Agora, é necessário adicionar o seguinte trecho de código no final do arquivo **index.php**

<?php  
 include "tabela.php";  
 exibirProdutos();  
?>

Esse código irá simplesmente chamar a função **exibirProdutos()** que é responsável por construir a tabela com os dados dentro.

**Atualizando os dados**

Adicione no botão “Salvar alterações” do arquivo **tabela.php** o evento **onclick** que deve chamar a função **atualizarDados()** quando o botão for clicado.

echo "<td>" . "<input type='submit' onclick='atualizarDados($row[id])' value='Salvar alterações'>" . "</td>";

A função **atualizarDados()** irá pegar os valores dos input **nome**, **quantidade** e **valor** com base no seu índice, que é determinado pelo valor do **id**, logo em seguida é criado um objeto **XMLHttpRequest** que será usado para enviar os valores das variáveis **nome**, **quantidade** e **valor** para o arquivo **update.php**.

Adicione o seguinte código no arquivo **index.php**

<script>  
 function atualizarDados(id){  
 let classValor = document.getElementsByClassName(id);  
 let nome = classValor[0].value;  
 let quantidade = classValor[1].value;  
 let valor = classValor[2].value;  
  
 const xhttp = new XMLHttpRequest();  
 xhttp.open("POST", "update.php");  
 xhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");  
 xhttp.send("id=" + id + "&nome=" + nome + "&quantidade=" + quantidade + "&valor=" + valor);  
 }  
</script>

O arquivo **update.php** é responsável por pegar os valores enviados usando o método **POST** é realizar a atualização das informações no banco de dados, como mostrado no seguinte código:

<?php  
include "db.php";  
  
// Atualiza os produtos  
if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["id"])){  
 $id = $\_POST["id"];  
 $nome = $\_POST["nome"];  
 $quantidade = $\_POST["quantidade"];  
 $valor = $\_POST["valor"];  
 $updateDados = "UPDATE produtos SET nome='$nome', quantidade='$quantidade', valor='$valor' WHERE id=$id";  
 $connection->query($updateDados);  
}  
  
$connection->close();  
?>

**Apagando produtos**

No botão “Apagar” do arquivo **tabela.php** também deve ser adicionado um evento **onclick**, que deve chamar a função **apagarDados()** quando o botão for pressionado.

echo "<td>" . "<input type='submit' onclick='apagarDados($row[id])' value='Apagar'>" . "</td>";

A função **apagarDados()** irá simplesmente criar um objeto **XMLHttpRequest**, logo em seguida será enviado uma solicitação do tipo **POST** para o arquivo **delete.php** contendo o **id** do produto que deve ser apagado. Se a solicitação for bem sucedida a página será recarregada.

Adicione o seguinte código entre as tags **<script></script>** dentro do arquivo **index.php**

function apagarDados(id){  
 const xh = new XMLHttpRequest();  
 xh.open("POST", "delete.php");  
 xh.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");  
 xh.send("id=" + id);  
 xh.onload = function() {  
 if (xh.status == 200) {  
 location.reload();  
 }  
 }  
 }

Quando a solicitação chegar no arquivo **delete.php** será verificado se a solicitação é do tipo **POST** e se o parâmetro **id** está definido, se ambas as condições forem verdadeiras a consulta SQL armazenada na variável **$deleteDados** será executada.

Crie o arquivo **delete.php** é adicione o seguinte código:

<?php  
include "db.php";  
  
// Apaga o produto do banco de dados  
if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["id"])){  
 $id = $\_POST["id"];  
 $deleteDados = "DELETE FROM produtos WHERE id=$id";  
 $connection->query($deleteDados);  
}  
  
$connection->close();  
?>

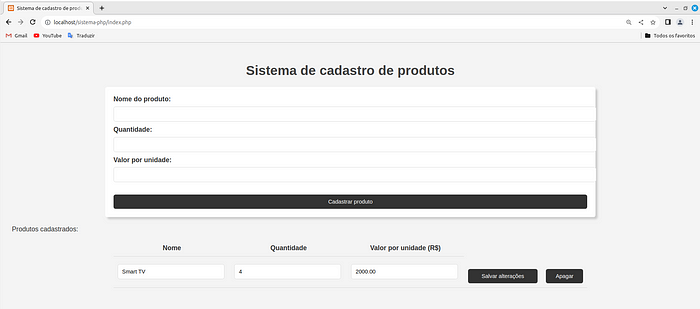
**Criando o arquivo style.css**

O seguinte código irá estilizar os elementos HTML.

body {  
 font-family: 'Segoe UI', Tahoma, Verdana, sans-serif;  
 background-color: #f4f4f4;  
 padding: 20px;  
 color: #333;  
}  
  
h1 {  
 color: #444;  
 text-align: center;  
}  
  
form {  
 background-color: #fff;  
 border-radius: 5px;  
 padding: 20px;  
 width: 70%;  
 margin: 20px auto;  
 box-shadow: 5px 5px 5px #ccc;  
}  
  
label {  
 font-weight: bold;  
}  
  
input[type="text"], input[type="number"] {  
 width: 100%;  
 padding: 10px;  
 border-radius: 5px;  
 border: 1px solid #ddd;  
 margin-top: 8px;  
 margin-bottom: 10px;  
}  
  
input[type="submit"] {  
 display: block;  
 width: 100%;  
 padding: 10px;  
 border-radius: 5px;  
 border: none;  
 color: #fff;  
 background-color: #333;  
 margin-top: 20px;  
}  
  
input[type="submit"]:hover {  
 background-color: #444;  
}  
  
table {  
 width: 70%;  
 margin: auto;  
 border-collapse: collapse;  
}  
  
th, td {  
 border-bottom: 1px solid #ddd;  
 padding: 10px;  
}  
  
td input[type="text"] {  
 width: 90%;  
}

**Imagens do sistema de cadastro de produtos**





**GitHub**

Todo o código apresentado durante o artigo está disponível no seguinte link: <https://github.com/JosianeBarros/Sistema-de-cadastro-de-produtos>

Se você tiver alguma dúvida fique a vontade para comentar. Até o próximo artigo!!