

# Programação para Web I

Aula - 16

# Formulários HTML e PHP

- Definição : Formulários são usados para coletar entradas de usuários em uma aplicação web.
- Integração com PHP: O PHP processa os dados enviados pelos formulários HTML.
- Métodos de Envio: Dois principais métodos - **GET** e **POST**.

# Formulários HTML e PHP

- Estrutura de um Formulário HTML

```
<form action="processa.php" method="post">  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" name="nome">  
  <input type="submit" value="Enviar">  
</form>
```

- Atributos:

- **action:** Especifica o arquivo PHP que irá processar o formulário.
- **method:** Especifica o método (GET ou POST).

# Formulários HTML e PHP

- Métodos GET e POST
  - **GET:**
    - Envia dados do formulário como parâmetros na URL.
      - Exemplo: `processa.php?nome=Alex`
    - Adequado para dados não sensíveis.
  - **POST:**
    - Envia dados no corpo da requisição.
    - Usado para dados sensíveis ou grandes.

# Formulários HTML e PHP

- Processamento de Formulários em PHP
  - Acessando os Dados: Superglobais do PHP
    - `$_GET['nome']`: Recupera dados enviados via GET.
    - `$_POST['nome']`: Recupera dados enviados via POST.

- Exemplo:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $nome = $_POST['nome'];  
    echo "Olá, $nome!";  
}
```

# Formulários HTML e PHP

- Validação de Formulário
  - Importância: Garante que os dados enviados pelos usuários são corretos e seguros.
  - Exemplo: Validando se um campo está vazio

```
if (empty($_POST['nome'])) {  
    echo "O nome é obrigatório";  
} else {  
    $nome = $_POST['nome'];  
}
```

# Formulários HTML e PHP

- Sanitização de Formulários
  - Propósito: Previne a entrada maliciosa, removendo caracteres indesejados, por exemplo:
    - " e ': usados para delimitar atributos HTML.
    - < e >: usados para abrir e fechar tags HTML.
  - **htmlspecialchars()**: converte caracteres especiais em entidades HTML, prevenindo a execução de código malicioso ou a quebra da estrutura da página.
    - Exemplo de uso:
      - `$nome = htmlspecialchars($_POST['nome']);`

# Formulários HTML e PHP

- Exemplo: Formulário Simples

- Formulário HTML:**

```
<form method="post" action="contato.php">  
    <label for="nome">nome:</label>  
    <input type="text" id="nome" name="nome">  
    <input type="submit" value="Enviar">  
</form>
```

- Script PHP (contato.php):**

```
<?php  
    if ($_SERVER["REQUEST_METHOD"] == "POST") {  
        if (empty($_POST['nome'])) {  
            echo "O campo de nome é obrigatório.";  
        } else {  
            $nome = htmlspecialchars($_POST['nome']);  
            echo "nome recebido: " . $nome;  
        }  
    }  
?>
```



# Banco de dados com PHP

- Estrutura do Banco de Dados (MySQL).

```
CREATE DATABASE produtosDB;
```

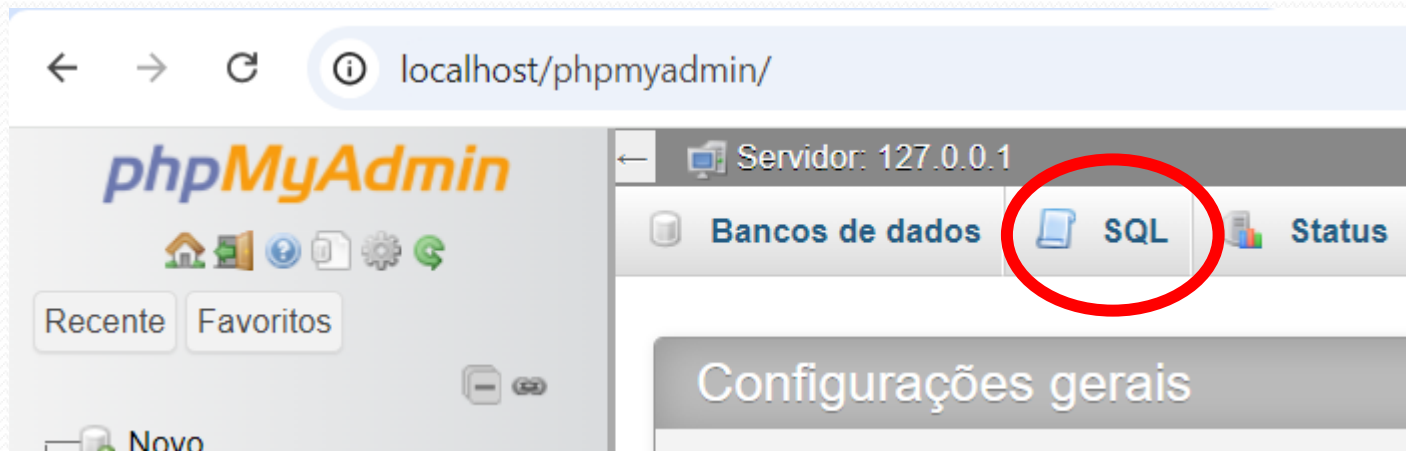
```
USE produtosDB;
```

```
CREATE TABLE produtos (  
    id INT(11) AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(255) NOT NULL,  
    preco DECIMAL(10, 2) NOT NULL,  
    quantidade INT NOT NULL  
);
```

- Execute esse script no gerenciador de banco de dados (phpMyAdmin).

# Banco de dados com PHP

- Para executar o script do slide anterior faça:
  - No XAMPP Control Panel, inicie o **Apache** e o **MySQL**.
  - Acesse: <http://localhost/phpmyadmin/>
  - Clique na aba SQL, cole e execute o script.



# Banco de dados com PHP

- Conexão com o Banco de Dados (db.php).

```
<?php
```

```
$host = 'localhost';
```

```
$db = 'produtosDB';
```

```
$user = 'root'; // altere conforme necessário
```

```
$pass = ""; // altere conforme necessário
```

```
try {
```

```
    $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);
```

```
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
} catch (PDOException $e) {
```

```
    die("Erro na conexão: " . $e->getMessage());
```

```
}
```

```
?>
```

# Banco de dados com PHP

- **Insere produto no banco de dados.**

```
<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nome = htmlspecialchars(trim($_POST['nome'])); // Sanitizar nome
    $preco = filter_var(trim($_POST['preco']), FILTER_VALIDATE_FLOAT);
    $quantidade = filter_var(trim($_POST['quantidade']), FILTER_VALIDATE_INT);

    // Verificar por erros de validação
    if ($preco === false || $quantidade === false) {
        // Tratar o erro (ex: redirecionar, exibir mensagem de erro)
        die('Preço ou quantidade inválidos.');// Exemplo de tratamento de erro
    }

    $sql = "INSERT INTO produtos (nome, preco, quantidade) VALUES (?, ?, ?)";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$nome, $preco, $quantidade]);
}
?>
```

# Banco de dados com PHP

- **Lista produtos.**

```
<?php
    include 'db.php';
    $sql = "SELECT * FROM produtos";
    $stmt = $pdo->query($sql);
    $produtos = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
<html>
<body>
<table>
    <?php foreach ($produtos as $produto): ?>
        <tr>
            <td><?= htmlspecialchars($produto['id']) ?></td>
            <td><?= htmlspecialchars($produto['nome']) ?></td>
            <td><?= htmlspecialchars($produto['preco']) ?></td>
            <td><?= htmlspecialchars($produto['quantidade']) ?></td>
        </tr>
    <?php endforeach; ?>
</table>
</body>
</html>
```

# Banco de dados com PHP

- **Edita dados produto.**

```
<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id = filter_var(trim($_POST['id']), FILTER_VALIDATE_INT);
    $nome = htmlspecialchars(trim($_POST['nome'])); // Sanitizar nome
    $preco = filter_var(trim($_POST['preco']), FILTER_VALIDATE_FLOAT);
    $quantidade = filter_var(trim($_POST['quantidade']), FILTER_VALIDATE_INT);

    // Verificar por erros de validação
    if ($id === false || $preco === false || $quantidade === false) {
        // Tratar o erro (ex: redirecionar, exibir mensagem de erro)
        die('Id, preço ou quantidade inválidos.');// Exemplo de tratamento de erro
    }

    $sql = "UPDATE produtos SET nome = ?, preco = ?, quantidade = ? WHERE id = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$nome, $preco, $quantidade, $id]);
}
?>
```

# Banco de dados com PHP

- **Exclui produto.**

```
<?php
    include 'db.php';

    if (isset($_GET['id'])) {

        $id = filter_var($_GET['id'], FILTER_VALIDATE_INT);

        if ($id === false) {
            die('ID inválido.');
```

```
        }

        $sql = "DELETE FROM produtos WHERE id = ?";
        $stmt = $pdo->prepare($sql);
        $stmt->execute([$id])

    }
?>
```

# Exercício

- Desenvolva uma aplicação web que permita a manipulação (criar, ler, atualizar e deletar) de informações sobre colaboradores de uma empresa, utilizando HTML, PHP e MySQL. A página deve gerenciar os dados dos colaboradores, que incluem: nome completo, número de telefone, endereço residencial, anos de experiência em programação e salário mensal.
- Elabore uma interface do usuário com um formulário que permita ao usuário inserir as informações de um novo colaborador e exiba uma lista de colaboradores cadastrados em uma tabela, com opções para editar e excluir cada colaborador.
- Inclua validações simples para garantir que os dados inseridos no formulário estejam corretos e seguros.