

# Vamos dar uma rápida passada nas matérias antigas

## Array

```
//Array indexado
$disciplinas = array ("BD", "TPA", "Front-End", "Matemática",
"Português");
echo "<br>Mostrando o conteúdo do array";
var_dump($disciplinas);

//Array associativo
$professor_disciplina = array ("Halley" => "BD", "Mario" =>
"TPA", "Susu" => "Front-End", "Ione" => "Português" );
var_dump($professor_disciplina);
```

## Formulario PHP

```
<form action="processar_notas.php" method="post">
  <div class="form-group">
    <label for="nome">Nome do Aluno:</label>
    <input type="text" class="form-control" id="nome"
name="nome" required>
  </div>
  <div class="form-group">
    <label for="nota1">Nota 1:</label>
    <input type="number" step="0.01" class="form-control"
id="nota1" name="nota1" required>
  </div>
  <div class="form-group">
    <label for="nota2">Nota 2:</label>
```

```
<input type="number" step="0.01" class="form-control"
id="nota2" name="nota2" required>
</div>
```

```
<div class="form-group">
  <label for="nota3">Nota 3:</label>
  <input type="number" step="0.01" class="form-control"
id="nota3" name="nota3" required>
</div>
  <button type="submit" class="btn btn-primary">Calcular
Média</button>
</form>
```

## Quando usamos GET ou POST

\$\_GET e \$\_POST são superglobais em PHP usadas para coletar dados enviados por meio de um formulário HTML com o método GET e POST, respectivamente. A diferença fundamental entre eles reside no modo como os dados são enviados:

1. \$\_GET:
  - Os dados são enviados como parte da URL.
  - Limitado a cerca de 2000 caracteres.
  - Adequado para dados não confidenciais.
  - Os dados são visíveis na barra de endereço do navegador.
  - Usado principalmente para recuperar informações do servidor.
2. Exemplo de URL com dados GET:
3. arduino
4. Copy code

`http://exemplo.com/pagina.php?nome=Joao&idade=25`

5.

6. `$_POST`:

- Os dados são enviados no corpo da solicitação HTTP.
- Não há limite de tamanho.
- Adequado para dados confidenciais.
- Os dados não são visíveis na barra de endereço do navegador.
- Usado principalmente para enviar informações ao servidor.

7. Exemplo de envio de dados POST:

8. html

9. Copy code

```
method "post" action "processar.php"
type "text" name "nome"
type "submit" value "Enviar"
```

10.

Em resumo, use `$_GET` quando estiver obtendo dados do servidor (por exemplo, para pesquisar, filtrar ou classificar) e quando os dados não forem sensíveis ou não confidenciais. Use `$_POST` quando estiver enviando dados para o servidor, especialmente quando os dados contêm informações confidenciais, como senhas ou informações de cartão de crédito.

## Agora vamos para o **CRUD Exemplo**

# CRUD - C

```
function inserirRegistro($pdo, $nome, $email, $senha) {  
    $sql = "INSERT INTO usuarios (nome, email, senha) VALUES  
(:nome, :email, :senha)";  
    $stmt = $pdo->prepare($sql);  
    $stmt->bindParam(':nome', $nome, PDO::PARAM_STR);  
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);  
    $stmt->bindParam(':senha', $senha, PDO::PARAM_STR);  
    return $stmt->execute();  
}
```

# CRUD - R

<?php

```
function listarRegistros($pdo) {  
    $sql = "SELECT * FROM usuarios";  
    $stmt = $pdo->query($sql);  
    return $stmt->fetchAll(PDO::FETCH_ASSOC);  
}
```

// Supondo que \$pdo seja uma instância válida de PDO, criada previamente

```
try {  
    // Crie sua conexão PDO aqui  
    $pdo = new  
PDO('mysql:host=seu_host;dbname=seu_banco_de_dados',  
'seu_usuario', 'sua_senha');  
    // Configure o PDO para relatar erros  
    $pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);  
    // Listar registros
```

```

foreach ($registros as $registro) {
    echo "<tr>";
    echo "<td>" . $registro['id'] . "</td>";
    echo "<td>" . $registro['nome'] . "</td>";
    echo "<td>" . $registro['email'] . "</td>";
    echo "<td>" . $registro['senha'] . "</td>";
    echo "<td>";

    // Faça o que precisa com os registros aqui
    foreach ($registros as $registro) {
        // Por exemplo, imprimir cada registro
        print_r($registro);
    }
} catch (PDOException $e) {
    // Se ocorrer um erro, você pode lidar com isso aqui
    echo "Erro: " . $e->getMessage();
}
?>

```

## CRUD - U

Não pode esquecer de por `<a href='edit.php?id=' . $registro['id'] . '>Editar</a>`

<?php

```

function atualizarRegistro($pdo, $id, $nome, $email, $senha) {
    // Prepare a declaração SQL com marcadores de parâmetros
    $sql = "UPDATE usuarios SET nome = :nome, email = :email,
senha = :senha WHERE id = :id";
    $stmt = $pdo->prepare($sql);
    // Vincule os valores dos parâmetros
    $stmt->bindParam(':nome', $nome);
    $stmt->bindParam(':email', $email);
    $stmt->bindParam(':senha', $senha);
    $stmt->bindParam(':id', $id);

    // Execute a declaração preparada
    return $stmt->execute();
}

```

```
// Supondo que $pdo seja uma instância válida de PDO, criada
previamente
try {
    // Crie sua conexão PDO aqui
    $pdo = new
PDO('mysql:host=seu_host;dbname=seu_banco_de_dados',
'seu_usuario', 'sua_senha');

    // Configure o PDO para relatar erros
    $pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);

    // Exemplo de uso da função para atualizar um registro
    $id = 1; // ID do registro que você quer atualizar
    $nome = "Novo Nome";
    $email = "novo@email.com";
    $senha = "nova_senha";

    // Chamada da função para atualizar o registro
    $resultado = atualizarRegistro($pdo, $id, $nome, $email,
$senha);

    if ($resultado) {
        echo "Registro atualizado com sucesso!";
    } else {
        echo "Erro ao atualizar o registro.";
    }
} catch (PDOException $e) {
    // Se ocorrer um erro, você pode lidar com isso aqui
    echo "Erro: " . $e->getMessage();
}
?>
```

**Aviso :** Certifique-se de substituir 'seu\_host', 'seu\_banco\_de\_dados', 'seu\_usuario' e 'sua\_senha' pelos valores corretos para a sua configuração de banco de dados.

## CRUD - D

Não pode esquecer de por `<a`

```
href='delete.php?id=' . $registro['id'] .
">Excluir</a>
```

```
function excluirRegistro($pdo, $id) {
    try {
        $sql = "DELETE FROM usuarios WHERE id = :id";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(':id', $id, PDO::PARAM_INT);
        return $stmt->execute();
    } catch (PDOException $e) {
        die("Erro ao excluir registro: " . $e->getMessage());
    }
}
```

# Agora vamos fazer um crud que ele passou (Crud-PDO)

## index

```
<!DOCTYPE html>
<html>
<head>
    <title>CRUD de USUARIO</title>
</head>
<body>
    <h1>Lista de USUARIO</h1>
    <a href="create.php">Adicionar Novo Usuario</a>
    <?php
        require ('pdo_con.php');

        // Função para listar todos os registros do banco de dados
        function listarRegistros($pdo) {
            $sql = "SELECT * FROM usuarios";
            $stmt = $pdo->query($sql);
            return $stmt->fetchAll(PDO::FETCH_ASSOC);
        }
    </?php>

```

```

// Listar registros
$registros = listarRegistros($pdo);

// Exibindo os dados em uma tabela
echo "<table border='1'>
    <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Email</th>
        <th>senha</th>
        <th>Ações</th>
    </tr>";

foreach ($registros as $registro) {
    echo "<tr>";
    echo "<td>" . $registro['id'] . "</td>";
    echo "<td>" . $registro['nome'] . "</td>";
    echo "<td>" . $registro['email'] . "</td>";
    echo "<td>" . $registro['senha'] . "</td>";
    echo "<td>
        <a href='edit.php?id=" . $registro['id'] .
        "'>Editar</a>
        <a href='delete.php?id=" . $registro['id'] .
        "'>Excluir</a>
    </td>";
    echo "</tr>";
}

echo "</table>";

?>
</body>
</html>

```

## PDO-Con (conexão com o banco de dados)



```

<?php
$host = 'localhost';
$dbname = 'aula_pdo';
$username = 'root';
$password = '';
try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username,
$password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    #echo "Connected successfully";
} catch (PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
?>

```

## Edit(Para editar o usuário )

```

<?php
require('pdo_con.php');

if (isset($_GET["id"]) && is_numeric($_GET["id"])) {
    $id = $_GET["id"];

    // Função para buscar um registro com o ID específico
    function buscarRegistroPorId($pdo, $id) {
        $sql = "SELECT * FROM usuarios WHERE id = :id";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(':id', $id, PDO::PARAM_INT);
        $stmt->execute();
        return $stmt->fetch(PDO::FETCH_ASSOC);
    }

    // Buscar o registro com o ID específico
    $registro = buscarRegistroPorId($pdo, $id);
}

```

```

?>
<!DOCTYPE html>
<html>
<head>
    <title>Editar Cliente</title>
</head>
<body>
    <h1>Editar Usuario</h1>
    <?php if (isset($registro)) : ?>
        <form action="update.php" method="post">
            <input type="hidden" name="id" value="<?php echo
$registro['id']; ?>">
            <label>Nome:</label>
            <input type="text" name="nome" value="<?php echo
$registro['nome']; ?>" required>
            <br>
            <label>Email:</label>
            <input type="email" name="email" value="<?php echo
$registro['email']; ?>" required>
            <br>
            <label>Senha:</label>
            <input type="text" name="senha" value="<?php echo
$registro['senha']; ?>" required>
            <br>
            <input type="submit" value="Salvar">
        </form>
    <?php else : ?>
        <p>Usuario não encontrado.</p>
    <?php endif; ?>
</body>
</html>

```

## Insert

```

<?php
echo "Inserindo dados abaixo... <br>";
require ('pdo_con.php');

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo $nome = $_POST["nome"] . '<br>';
    echo $email = $_POST["email"];
    $senha = $_POST["senha"];
    echo "<hr>";

    // Função para inserir um novo registro no banco de dados
    function inserirRegistro($pdo, $nome, $email, $senha) {
        $sql = "INSERT INTO usuarios (nome, email, senha) VALUES
(:nome, :email, :senha)";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(':nome', $nome, PDO::PARAM_STR);
        $stmt->bindParam(':email', $email, PDO::PARAM_STR);
        $stmt->bindParam(':senha', $senha, PDO::PARAM_STR);
        return $stmt->execute();
    }
}

if (inserirRegistro($pdo, $nome, $email, $senha)) {
    echo "Registro inserido com sucesso!<br>" . "<a
href='index.php'>HOME</a>";
} else {
    echo 'Erro ao inserir o registro.';
}
?>

```

## Create

```

<!DOCTYPE html>
<html>
<head>
    <title>Adicionar Novo Cliente</title>
</head>
<body>
    <h1>Adicionar Novo Cliente</h1>
    <form action="insert.php" method="post">
        <label>Nome:</label>
        <input type="text" name="nome" required>
        <br>

```

```

        <label>Email:</label>
        <input type="email" name="email" required>
        <br>
        <label>Senha:</label>
        <input type="password" name="senha" required>
        <br>
        <input type="submit" value="Salvar">
    </form>
</body>
</html>

```

## Update

```

<?php
echo "Atualizando dados abaixo... <br>";
require ('pdo_con.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST["id"];
    $nome = $_POST["nome"];
    $email = $_POST["email"];
    $senha = $_POST["senha"];
    echo "<hr>";

    // Função para Atualizar o registro no banco de dados
    function atualizarRegistro($pdo, $id, $nome, $email, $senha) {
        $sql = "UPDATE usuarios SET nome = :nome, email = :email, senha
= :senha WHERE id = :id";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(':nome', $nome);
        $stmt->bindParam(':email', $email);
        $stmt->bindParam(':senha', $senha);
        $stmt->bindParam(':id', $id);
        return $stmt->execute();
    }

    if (atualizarRegistro($pdo, $id, $nome, $email, $senha)) {
        echo "Registro atualizado com sucesso!<br>" . "<a
href='index.php'>HOME</a>";
    } else {
        echo 'Erro ao atualizar o registro.';
    }
} else {
    echo "O formulário não foi submetido via método POST.";
}
?>

```

# Delete

```
<?php
require('pdo_con.php');

if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET["id"]) &&
is_numeric($_GET["id"])) {
    $id = $_GET["id"];

    // Função para deletar o registro no banco de dados
    function excluirRegistro($pdo, $id) {
        $sql = "DELETE FROM usuarios WHERE id = :id";
        $stmt = $pdo->prepare($sql);
        $stmt->bindParam(':id', $id, PDO::PARAM_INT);
        return $stmt->execute();
    }

    if (excluirRegistro($pdo, $id)) {
        echo "Registro excluído com sucesso!<br><a
href='index.php'>HOME</a>";
    } else {
        echo 'Erro ao excluir o registro.';
    }
} else {
    echo "Requisição inválida.";
}
?>
```

# Crud gerado por IA

## 1. Conexão com o Banco de Dados:

php

Copy code

```
<?php
$host = 'localhost';
$dbname = 'seu_banco_de_dados';
$username = 'seu_usuario';
$password = 'sua_senha';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username,
$password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
```

```
die("Erro na conexão com o banco de dados: " . $e->getMessage());
}
?>
```

## 2. Listar Usuários (Read - R):

php

Copy code

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

$stmt = $pdo->query("SELECT * FROM usuarios");
$usuarios = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html>
<head>
<title>Listar Usuários</title>
</head>
<body>
<h2>Usuários</h2>
<table border="1">
<tr>
<th>ID</th>
<th>Nome</th>
<th>Email</th>
<th>Ações</th>
</tr>
<?php foreach ($usuarios as $usuario): ?>
<tr>
<td><?php echo $usuario['id']; ?></td>
<td><?php echo $usuario['nome']; ?></td>
<td><?php echo $usuario['email']; ?></td>
<td>
<a href="editar.php?id=<?php echo $usuario['id']; ?>">Editar</a>
<a href="excluir.php?id=<?php echo $usuario['id']; ?>">Excluir</a>
</td>
</tr>
<?php endforeach; ?>
</table>
<br>
<a href="adicionar.php">Adicionar Novo Usuário</a>
</body>
</html>
```

## 3. Adicionar Usuário (Create - C):

php

Copy code

```
<!DOCTYPE html>
<html>
<head>
<title>Adicionar Usuário</title>
```

```

</head>
<body>
  <h2>Adicionar Usuário</h2>
  <form action="adicionar_processar.php" method="post">
    Nome: <input type="text" name="nome"><br><br>
    Email: <input type="email" name="email"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <input type="submit" value="Adicionar">
  </form>
</body>
</html>

```

#### 4. Processamento para Adicionar Usuário:

php

Copy code

```

<?php
include 'conexao.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $senha = $_POST['senha'];

    $stmt = $pdo->prepare("INSERT INTO usuarios (nome, email, senha) VALUES
(:nome, :email, :senha)");
    $stmt->bindParam(':nome', $nome, PDO::PARAM_STR);
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);
    $stmt->bindParam(':senha', $senha, PDO::PARAM_STR);
    $stmt->execute();

    header('Location: listar.php');
    exit();
}
?>

```

#### 5. Editar Usuário (Update - U):

php

Copy code

```

<?php
include 'conexao.php';

$id = $_GET['id'];
$stmt = $pdo->prepare("SELECT * FROM usuarios WHERE id = :id");
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
$usuario = $stmt->fetch(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html>
<head>
  <title>Editar Usuário</title>

```

```

</head>
<body>
  <h2>Editar Usuário</h2>
  <form action="editar_processar.php" method="post">
    <input type="hidden" name="id" value="<?php echo $usuario['id']; ?>">
    Nome: <input type="text" name="nome" value="<?php echo $usuario['nome'];
?>"><br><br>
    Email: <input type="email" name="email" value="<?php echo $usuario['email'];
?>"><br><br>
    Senha: <input type="password" name="senha" value="<?php echo
$usuario['senha']; ?>"><br><br>
    <input type="submit" value="Salvar">
  </form>
</body>
</html>

```

## 6. Processamento para Editar Usuário:

php

Copy code

```

<?php
include 'conexao.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id = $_POST['id'];
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $senha = $_POST['senha'];

    $stmt = $pdo->prepare("UPDATE usuarios SET nome = :nome, email = :email, senha
= :senha WHERE id = :id");
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);
    $stmt->bindParam(':nome', $nome, PDO::PARAM_STR);
    $stmt->bindParam(':email', $email, PDO::PARAM_STR);
    $stmt->bindParam(':senha', $senha, PDO::PARAM_STR);
    $stmt->execute();

    header('Location: listar.php');
    exit();
}
?>

```

## 7. Excluir Usuário (Delete - D):

php

Copy code

```

<?php
include 'conexao.php';

$id = $_GET['id'];
$stmt = $pdo->prepare("DELETE FROM usuarios WHERE id = :id");
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();

```



```
header('Location: listar.php');  
exit();  
?>
```