
AGENDA 6

PHP: PERSISTÊNCIA DE DADOS PARTE 1 - MYSQLI



GEEaD - Grupo de Estudos de Educação a Distância
Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
PROGRAMAÇÃO MOBILE I

Expediente

Autor:

Paulo Eduardo Cardoso Andrade

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação: Flávio Biazim



Mysqli

Para ser possível o acesso e manipulação de uma base de dados através da linguagem de programação PHP, é necessário um Driver (MYSQLI), para realizar a ponte entre linguagem e base de dados. Formalizando a extensão MySQLi é um driver de banco de dados relacional usado na linguagem PHP, fornecendo uma interface com os bancos de dados MySQL. Então como o nome indica, o uso dessa extensão é possível apenas para bancos MySQL.

O PHP costumava utilizar orinalmento o driver homônimo do banco MySQL, que suporta as versões mais antigas do MySQL. No entanto, essa extensão foi descontinuada e substituída por MySQLi.

Obs.1: As funções MySQLi funcionam apenas com **PHP 5** (ou superior) e **MySQL 4.1.3** (ou superior).

Obs.2: MySQLi - o "i" vem da palavra em língua inglesa *improved* ou melhorado.

Vamos começar! Antes de qualquer programação, será necessário criar uma base de dados e uma tabela. Para exemplo vamos utilizar o banco de dados MySQL disponível no USBWebServer e em conjunto vamos utilizar MySQL Workbench que é uma ferramenta visual para design de banco de dados, dentro dela podemos desenvolver, administrar, criar e manter, tudo em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL. Download pode ser realizado através do link: <https://dev.mysql.com/downloads/workbench/>.

Obs.1: Para a criação do banco de dados vocês está livre para usar a ferramenta que for mais amigável para você. Um outro exemplo é o phpmyadmin, que para acessá-lo, basta clicar no botão de mesmo nome no painel do USBWebserver



Imagem 3. Destaque do botão phpmyAdmin.

Esta ação encaminhará pra uma página para realizar um login, com o usuário é “root” e senha usbw” (utilizando o MySql do USBWebserver.

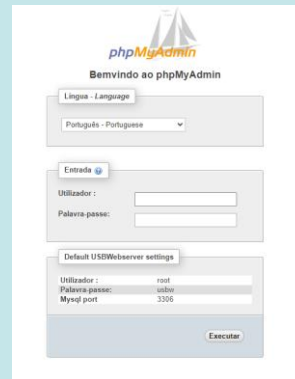


Imagem 4. Tela de login.

Após a realização do login, você será redirecionado para a tela que oferece todos recursos necessários para criar e gerenciar sua base de dados.

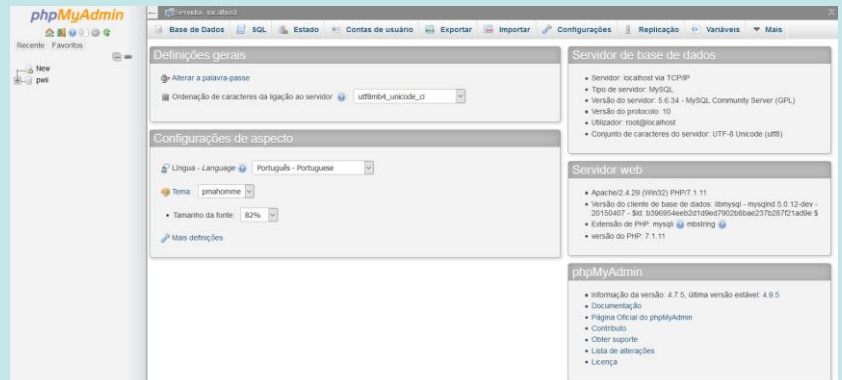


Imagem 5. Tela principal do phpmyadmin.

Para começarmos vamos precisar de uma base de dados, caso queira utilizar o USBWebServer, o usuário é “root” e senha usbw”, então crie uma base de dados denominada “pwii”.

```
CREATE DATABASE `pwii`;
```

Ou

```
CREATE SCHEMA `pwii` ;
```

Também crie uma tabela nesta base denominada amigos, que tenha os campos:

- id (chave primária e auto incremento).
- Nome.
- Apelido.
- Email.

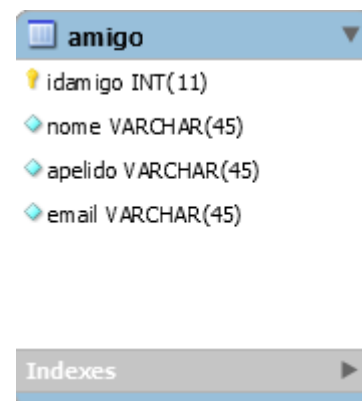


Imagem 6. Diagrama Banco de dados.

ou utilize o script a seguir.

```
CREATE TABLE `pwii`.`amigo` (
  `idamigo` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `apelido` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idamigo`));
```

Para este exemplo, será necessário criar diversos arquivos para a abordagem das quatro operações básicas (inserir, excluir, atualizar e deletar). Vamos começar, no visual studio Code, dentro da pasta Agenda6 (ou outra pasta de sua preferência), crie o primeiro arquivo com o nome de "index.php" - Neste arquivo vamos criar dois links, que irão nos direcionar para a realização das ações relacionadas ao banco de dados. Então codifique.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <title>Projeto - MYSQLI</title>
</head>
<body>
<div class="w3-padding w3-text-grey w3-half w3-display-middle w3-center">
  <h1 class="w3-center w3-teal w3-round-large w3-
margin">Projeto Lista de Amigos</h1>
  <div class="w3-row">
    <div class="w3-col w3-button w3-teal w3-cell w3-round-
large" style="width:45%;">
      <a href="cadastro.php" style="text-decoration: none;">
        <i class=" fa fa-user-plus" style="font-size: 10.5em"></i>
        <p style="font-size: 2em">Adicionar </p>
      </a>
    </div>
    <div class="w3-col w3-button w3-teal w3-cell w3-round-large w3-
right" style="width:45%;">
      <a href="listar.php" style="text-decoration: none;">
        <i class="fa fa-vcard-o" style="font-size: 10.5em"></i>
        <p style="font-size: 2em">Listar</p>
      </a>
    </div>
  </div>
</div>
</body>
</html>
```

Como é possível perceber pelo código anterior, os dois links criados nos encaminham para dois arquivos diferentes: cadastro.php e listar.php.

O resultado no Navegador:

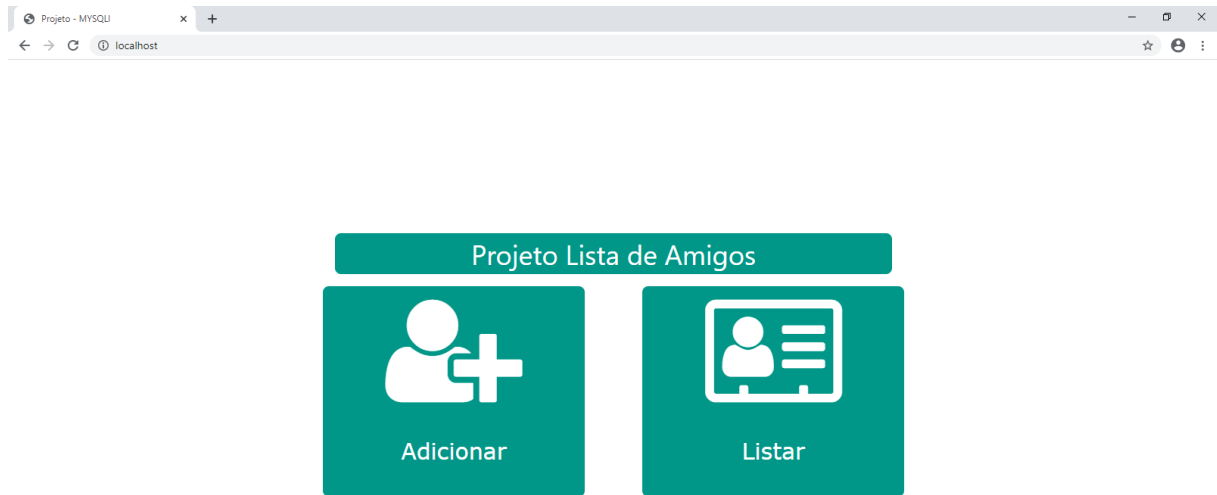


Imagem 7. Resultado no Navegador.

Para continuidade, criaremos agora o arquivo “cadastro.php”, na mesma pasta do projeto. Neste arquivo vamos criar um formulário com quatro campos, sendo eles repectivos para: ID, Nome, Apelido e e-mail, ao final, também será criado um botão para adicionar o amigo, a sua base de dados. Código a seguir:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <title>Cadastro - MYSQLI</title>
</head>
<body>
<a href="index.php" class="w3-display-topleft">
  <i class="fa fa-arrow-circle-left w3-large w3-teal w3-button w3-
xxlarge"></i>
</a>
<div class="w3-padding w3-content w3-text-grey w3-third w3-margin w3-display-
middle">
  <h1 class="w3-center w3-teal w3-round-large w3-
margin">Cadastro de Amigos</h1>
  <form action="cadastroAction.php" class="w3-container" method='post'>
```

```

        <label class="w3-text-teal" style="font-
weight: bold;">Código</label>
        <input name="txtID" class="w3-input w3-grey w3-
border" disabled><br>
        <label class="w3-text-teal" style="font-
weight: bold;">Nome</label>
        <input name="txtNome" class="w3-input w3-light-grey w3-
border"><br>
        <label class="w3-text-teal" style="font-
weight: bold;">Apelido</label>
        <input name="txtApelido" class="w3-input w3-light-grey w3-
border"><br>
        <label class="w3-text-teal" style="font-
weight: bold;">Email</label>
        <input name="txtEmail" class="w3-input w3-light-grey w3-
border"><br>
        <button name="btnAdicionar" class="w3-button w3-teal w3-cell w3-
round-large w3-right w3-margin-right">
            <i class="w3-xxlarge fa fa-user-plus"></i> Adicionar
        </button>
    </form>
</div>

```

Obs.: Logo no início do corpo da página foi criado um link, este tem a função de retorno para a página inicial, fazendo com que a navegação em nosso site fique mais fluída e completa.

Resultado no Navegador:

The screenshot shows a web browser window with the title 'Cadastro - MYSQLI'. The address bar shows 'localhost/cadastro.php'. The page content includes a form titled 'Cadastro de Amigos' with the following fields:

- Código**: A disabled input field.
- Nome**: An input field.
- Apelido**: An input field.
- Email**: An input field.

At the bottom right of the form is a green button with a plus icon and the text 'Adicionar'.

Imagem 8. Resultado no Navegador.

A action do formulário do código anterior, está definida com o arquivo “cadastroAction.php”. Arquivo que devemos criar. Então para a continuidade, crie o arquivo “cadastroAction”, também dentro da mesma pasta do projeto. Neste arquivo, vamos interagir pela primeira vez com a nossa base de dados criada anteriormente, realizando a conexão com o banco, criando e executando a sentença sql de insert, e para finalizar enviar mensagens de sucesso ou fracasso, a depender do resultado da nossa inserção de dados na base.

Para criar a conexão, há uma boa prática de criar variáveis para armazenar o nome do servidor, nome do usuário, senha e por fim o nome da base de dados. Após a definição dos valores para cada uma das variáveis, criamos uma instância do mysqli, que nos fornecerá recursos para realizar a inserção no banco de dados.

```
$servername = "localhost";
$username = "root";
$password = "usbw";
$dbname = "pwii";
$conexao = new mysqli($servername, $username, $password, $dbname);
if ($conexao->connect_error) {
    die("Connection failed: " . $conexao->connect_error);
}
```

Obs.: Ao observar o código anterior, percebemos que há um if, para verificar se a conexão foi realizada com sucesso.

O próximo passo será a criação da sentença sql, para a inserção de um registro na tabela amigo, utilizando os dados oriundos do formulário do arquivo “cadastro.php”.

```
$sql = "INSERT INTO amigo (nome, apelido, email)
VALUES ('".$_POST['txtNome']."' , '".$_POST['txtApelido']."' , '".$_POST['txtEmail']."' )";

if ($conexao->query($sql) === TRUE) {
    echo '
    <a href="index.php">
        <h1 class="w3-button w3-teal">Amigo Salvo com sucesso! </h1>
    </a>
    ';
}
```



```

    } else {
        echo '
        <a href="index.php">
            <h1 class="w3-button w3-teal">ERRO! </h1>
        </a>
        ';
    }

```

A execução da query sql, está sendo realizada dentro de uma estrutura condicional “if”, verificando obtenção de sucesso em sua ação, fornecendo assim uma mensagem de sucesso em caso verdadeiro ou uma mensagem de falha em caso falso.

O código completo deste arquivo, com os recursos em html e css deve ficar:

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <title>Cadastro - MYSQLI</title>
</head>
<body>
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-middle">
    <?php
        $servername = "localhost";
        $username = "root";
        $password = "usbw";
        $dbname = "pwii";
        $conexao = new mysqli($servername, $username, $password, $dbname);
        if ($conexao->connect_error) {
            die("Connection failed: " . $conexao->connect_error);
        }
        $sql = "INSERT INTO amigo (nome, apelido, email)
        VALUES ('".$_POST['txtNome']. "', '".$_POST['txtApelido']. "', '".$_POST['txtEmail']. "')";

        if ($conexao->query($sql) === TRUE) {
            echo '
            <a href="index.php">
                <h1 class="w3-button w3-teal">Amigo Salvo com sucesso! </h1>
            </a>
            ';
        } else {
            echo '
            <a href="index.php">

```

```

        <h1 class="w3-button w3-teal">ERRO! </h1>
    </a>
    ';
}
$conexao->close();
?>
</div>
</body>
</html>

```

Claro não podemos esquecer de fechar a conexão com o banco.

```
$conexao->close();
```

Obs.: Tanto para a mensagem de sucesso, quanto para mensagem de falha, terão links direcionando para a página inicial “index.php”.

O resultado no navegador para a inserção no banco com sucesso será:

Amigo Salvo com sucesso!

Imagem 9. Resultado no Navegador para mensagem de sucesso.

Obs: A mensagem de falha segue o mesmo design.

Em continuidade ao projeto, a tarefa de agora é criar o arquivo “listar.php” (arquivo do segundo link da página index), então crie o arquivo “listar”, na mesma pasta do projeto.

Neste arquivo vamos criar uma tabela com seis campos, sendo eles repectivos para ID, Nome, Apelido, e-mail, remover amigo e atualizar amigo, estes dois últimos serão links que redirecionarão para páginas respectivas de exclusão e atualização de amigos. Para a montagem desta tabela, vamos utilizar os dados salvos no banco de dados em vamos precisar novamente:

- Criar instância Mysqli
- Verificar conexão.
- Criar sentença SQL

```
$servername = "localhost";
$username = "root";
$password = "usbw";
$dbname = "pwii";
$conexao = new mysqli($servername, $username, $password, $dbname);
if ($conexao->connect_error) {
    die("Connection failed: " . $conexao->connect_error);
}
$sql = "SELECT * FROM amigo" ;
```

Tudo seguindo o mesmo padrão da inserção de dados na base. No entanto, para a setença select, será retornado uma matriz de dados (array), ou seja, precisamos atribuir o resultado em uma variável, também sem esquecer a confirmação para a certeza se foi retornado o resultado esperado.

```
$resultado = $conexao->query($sql);
if($resultado != null)
```

Em caso de retorno, uma a estrutura de repetição foreach (como desenvolvido na AGENDA 5), será utilizada para montar a tabela, listando todos os amigos.

```
foreach($resultado as $linha) {
    echo '<tr>';
    echo '<td>'.$linha['idamigo'].'</td>';
    echo '<td>'.$linha['nome'].'</td>';
    echo '<td>'.$linha['apelido'].'</td>';
    echo '<td>'.$linha['email'].'</td>';
    echo '<td><a href="excluir.php?id='.$linha['idamigo'].'&nome='.$linha['nome'].'&apelido='.$linha['apelido'].'&email='.$linha['email'].'"><i class="fa fa-user-times w3-large w3-text-teal"></i> </a></td></td>';
    echo '<td><a href="atualizar.php?id='.$linha['idamigo'].'&nome='.$linha['nome'].'&apelido='.$linha['apelido'].'&email='.$linha['email'].'"><i class="fa fa-refresh w3-large w3-text-teal"></i></a></td></td>';
    echo '</tr>';
}
```

Neste código, perceba que também já criamos os links para as páginas de exclusão e atualização de amigos, e que enviaremos os dados através da url, utilizando os conceitos do método get, desenvolvido nas primeiras agendas.

O código completo desta página deve ficar.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <title>Listagem de Amigos - MYSQLI</title>
</head>
<body>
<a href="index.php" class="w3-display-topleft">
    <i class="fa fa-arrow-circle-left w3-large w3-teal w3-button w3-xxlarge"></i>
</a>

    <?php
        $servername = "localhost";
        $username = "root";
        $password = "usbw";
        $dbname = "pwii";
        $conexao = new mysqli($servername, $username, $password, $dbname);
        if ($conexao->connect_error) {
            die("Connection failed: " . $conexao->connect_error);
        }

        echo '
            <div class="w3-paddingw3-content w3-half w3-display-topmiddle w3-margin">
                <h1 class="w3-center w3-teal w3-round-large w3-margin">Listagem de Amigos</h1>
                <table class="w3-table-all w3-centered">
                    <thead>
                        <tr class="w3-center w3-teal">
                            <th>Código</th>
                            <th>Nome</th>
                            <th>Apelido</th>
                            <th>Email</th>
                            <th>Excluir</th>
                            <th>Atualizar</th>
                        </tr>
                    </thead>
                </div>
                '
        $sql = "SELECT * FROM amigo" ;
```

```

$resultado = $conexao->query($sql);
if($resultado != null)
foreach($resultado as $linha) {
    echo '<tr>';
    echo '<td>'.$linha['idamigo'].'</td>';
    echo '<td>'.$linha['nome'].'</td>';
    echo '<td>'.$linha['apelido'].'</td>';
    echo '<td>'.$linha['email'].'</td>';
    echo '<td><a href="excluir.php?id='.$linha['idamigo'].'&nome='.$linha['nome'].'&apelido='.$linha['apelido'].'&email='.$linha['email'].'"><i class="fa fa-user-times w3-large w3-text-teal"></i> </a></td></td>';
    echo '<td><a href="atualizar.php?id='.$linha['idamigo'].'&nome='.$linha['nome'].'&apelido='.$linha['apelido'].'&email='.$linha['email'].'"><i class="fa fa-refresh w3-large w3-text-teal"></i></a></td></td>';
    echo '</tr>';
}
echo '
    </table>
</div>';
$conexao->close();
?>
</div>
</body>
</html>

```

Obs.: Logo no início do corpo da página foi criado um link, este tem a função de retorno para a página, inicial, fazendo com que a navegação em nosso site fique mais fluída.

O resultado no navegador para será como representado na imagem a seguir.



Imagem 10. Resultado no Navegador para lista de amigos com apenas o contato da gabriela.

Agora é o momento para a criação de mais dois arquivos: excluir e excluir action, para o primeiro criaremos um formulário com os campos: nome, apelido e email (todos desabilitados), um botão para confirmar a exclusão e um link para cancelar a exclusão. O resultado deve ser como o apresentado na imagem a seguir.

Imagem 11. Resultado no Navegador para exclusão de um item da lista de amigos.

Para a construção deste arquivo, codifique.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <title>Excluir - MYSQLI</title>
</head>
<body>
<a href="index.php" class="w3-display-topmiddle w3-red w3-center w3-
padding w3-button" style="text-decoration:none; ">
  <i class="fa fa-ban" style="font-size:5em"></i>
  <p style="font-weight:bold;">CANCELAR EXCLUSÃO</p>
</a>
<div class="w3-padding w3-content w3-text-grey w3-third w3-margin w3-display-
middle">
  <h1 class="w3-center w3-teal w3-round-large w3-
margin">EXLUIR - ID: <?php echo " ".$_GET['id']?> </h1>
  <form action="excluirAction.php" class="w3-container w" method='post'>
    <input name="txtID" class="w3-input w3-grey w3-
border" type="hidden" value="<?php echo $_GET['id']?>">
```

```

        <br>
        <label class="w3-text-teal" style="font-weight: bold;">Nome</label>
        <input name="txtNome" class="w3-input w3-border w3-
grey" disabled value="<?php echo $_GET['nome']?>">
        <br>
        <label class="w3-text-teal" style="font-weight: bold;">Apelido</label>
        <input name="txtApelido" class="w3-input w3-border w3-
grey" disabled value="<?php echo $_GET['apelido']?>">
        <br>
        <label class="w3-text-teal" style="font-weight: bold;">Email</label>
        <input name="txtEmail" class="w3-input w3-border w3-
grey" disabled value="<?php echo $_GET['email']?>">
        <br>
        <button name="btnExcluir" class="w3-button w3-teal w3-cell w3-round-
large w3-right">
            <i class="w3-xxlarge fa fa-check"></i> Confirmar Exclusão.
        </button>
    </form>
</div>
</body>
</html>

```

Perceba com o código, que preenchemos os dados do formulário com os dados obtidos via método get, enviados do arquivo “lista.php”.

Como o atributo action deste form está indicando o arquivo “excluirAction”, precisamos desenvolvê-lo. Então devemos:

- Criar instância Mysqli.
- Verificar conexão.
- Criar sentença DELETE.
- Executar a sentença verificando se mesma obteve sucesso.
- Gerar mensagens de sucesso e falha com link para o arquivo listar.php.
- Fechar a conexão.

Então codifique.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

```

```

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<title>Exclusão - MYSQLI</title>
</head>
<body>
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-
middle" id="eProfissional">
    <?php
        $servername = "localhost";
        $username = "root";
        $password = "usbw";
        $dbname = "pwii";
        $conexao = new mysqli($servername, $username, $password, $dbname);
        if ($conexao->connect_error) {
            die("Connection failed: " . $conexao->connect_error);
        }
        $sql = "DELETE FROM amigo WHERE idamigo = '$_POST['txtID'] .''";
        if ($conexao->query($sql) === TRUE) {
            echo '
                <a href="listar.php">
                    <h1 class="w3-button w3-
teal">Amigo Excluido com sucesso! </h1>
                </a>
            ';
        } else {
            echo '
                <a href="listar.php">
                    <h1 class="w3-button w3-teal">ERRO! </h1>
                </a>
            ';
        }
        $conexao->close();
    ?>
</div>

```

O resultado de um amigo excluído será como demonstrado na imagem a seguir.



Amigo Excluido com sucesso!

Imagem 12. Resultado no Navegador para mensagem de sucesso.

Obs: A mensagem de falha segue o mesmo design.

Para a última das quatro operações básicas, sobrou a atualização dos dados. Para isso vamos criar mais dois arquivos: “atualizar.php” e “atualizarAction.php”. Para o primeiro vamos criar um formulário com os campos (inputs): o nome, apelido e email, que também receberá os dados da url através da utilização do método get, direto do arquivo listar.php, oferecendo a possibilidade do usuário realizar alterações nos dados e assim executar o update, através do clique no botão “atualizar”, confirmando a alteração dos dados

O resultado no navegador deve ser como o apresentado na imagem a seguir.

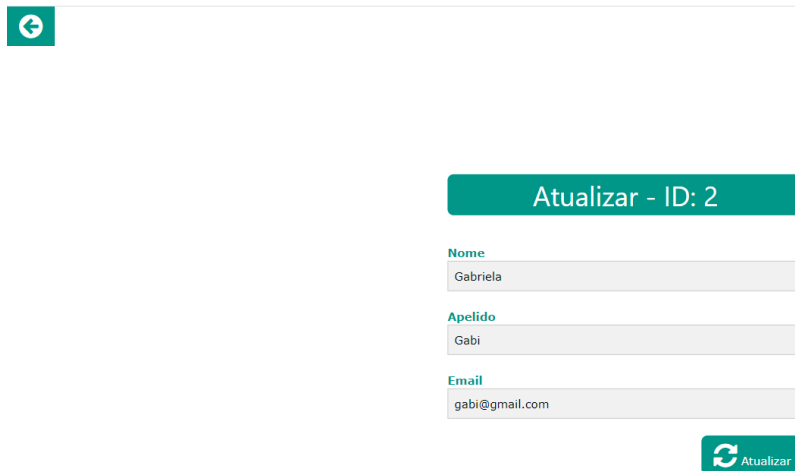


Imagem 13. Resultado no Navegador para mensagem de sucesso.

A codificação para a construção dessa página, será.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <title>Atualizar - MYSQLI</title>
</head>
<body>
<a href="index.php" class="w3-display-topleft">
  <i class="fa fa-arrow-circle-left w3-large w3-teal w3-button w3-
xxlarge"></i>
</a>
<div class="w3-padding w3-content w3-text-grey w3-third w3-margin w3-display-
middle">
  <h1 class="w3-center w3-teal w3-round-large w3-
margin">Atualizar - ID: <?php echo " ".$_GET['id']?> </h1>
```

```

<form action="atualizarAction.php" class="w3-container" method='post'>

    <input name="txtID" class="w3-input w3-grey w3-
border" type="hidden" value="<?php echo $_GET['id']?>">
    <br>
    <label class="w3-text-teal" style="font-weight: bold;">Nome</label>
    <input name="txtNome" class="w3-input w3-light-grey w3-
border" value="<?php echo $_GET['nome']?>">
    <br>
    <label class="w3-text-teal" style="font-weight: bold;">Apelido</label>
    <input name="txtApelido" class="w3-input w3-light-grey w3-
border" value="<?php echo $_GET['apelido']?>">
    <br>
    <label class="w3-text-teal" style="font-weight: bold;">Email</label>
    <input name="txtEmail" class="w3-input w3-light-grey w3-
border" value="<?php echo $_GET['email']?>">
    <br>
    <button name="btnAtualizar" class="w3-button w3-teal w3-cell w3-round-
large w3-right">
        <i class="w3-xxlarge fa fa-refresh"></i> Atualizar
    </button>
</form>
</div>
</body>
</html>

```

Como o atributo action deste form, indica o arquivo “atualizarAction.php”, precisamos desenvolvê-lo. Para isso devemos:

- Criar instância Mysqli
- Verificar conexão.
- Criar sentença Update
- Executar a sentença verificando se mesma obteve sucesso
- Gerar mensagens de sucesso e falha com link para o arquivo listar.php
- Fechar conexão

Então codifique.

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">

```

```

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<title>Atualização - MYSQLI</title>
</head>
<body>
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-middle" >
  <?php
    $servername = "localhost";
    $username = "root";
    $password = "usbw";
    $dbname = "pwii";
    $conexao = new mysqli($servername, $username, $password, $dbname);
    if ($conexao->connect_error) {
        die("Connection failed: " . $conexao->connect_error);
    }
    $sql = "UPDATE amigo SET nome = '". $_POST['txtNome']."' , apelido = '".
$_POST['txtApelido']."' ,
    email='". $_POST['txtEmail']."' WHERE idamigo =". $_POST['txtID']. " ";

    if ($conexao->query($sql) === TRUE) {
        echo '
        <a href="listar.php">
            <h1 class="w3-button w3-
teal">Amigo Atualizado com sucesso! </h1>
        </a>
        ' ;
        $id = mysqli_insert_id($conexao);

    } else {
        echo '
        <a href="listar.php">
            <h1 class="w3-button w3-teal">ERRO! </h1>
        </a>
        ' ;
    }
    $conexao->close();
  ?>
</div>
</body>
</html>

```

O resultado de um amigo atualizado será como demonstrado na imagem a seguir.

Amigo Atualizado com sucesso!

Imagem 14. Resultado no Navegador para mensagem de sucesso.

Obs: O formato dessa agenda abordou o conceito utilizando apenas o conteúdo desenvolvido até o momento nas agendas e módulo anteriores, há outros métodos e conceitos para fazer todas as operações contidas nessa agenda.



ATENÇÃO - Será disponibilizado o arquivo **usbwebserver - Agenda6**, nele terá todos os arquivos php e tabelas no banco de dados do mergulhando no tema, você no comando e itens descritos na atividade online disponíveis para uso e consulta.



Utilizando o que foi visto até agora, foi criada uma tabela no **banco de dados** com o nome: **jogos**, com os atributos **idjogo** (auto incremento), **nome** e **fabricante**, conforme diagrama a seguir.

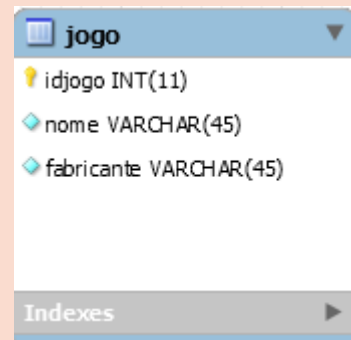


Imagem 15. Diagrama Banco de dados.

1. Obtenha os dados da tabela, utilizando o driver Mysqli
2. Crie uma tabela utilizando os dados obtidos através da consulta ao banco de dados e exiba no navegador.

Dicas:

- Utilize os conceitos e exemplos de estrutura de repetição da Agenda 5.
- De preferência ao uso da estrutura de repetição Foreach.

Caso esteja com dificuldade a seguir script sql para criação da tabela jogo.

```
CREATE TABLE `pwii`.`jogo` (
  `idjogo` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `fabricante` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idjogo`));
```

Caso esteja com dificuldade, a seguir script sql inserção de alguns registros de jogos.

```
INSERT INTO `pwii`.`jogo` (`nome`, `fabricante`) VALUES ('FIFA 2020', 'EA');
INSERT INTO `pwii`.`jogo` (`nome`, `fabricante`) VALUES ('FINAL FANTASY', 'Square Enix');
INSERT INTO `pwii`.`jogo` (`nome`, `fabricante`) VALUES ('GTA', 'Rockstar Games');
INSERT INTO `pwii`.`jogo` (`nome`, `fabricante`) VALUES ('CS: GO', 'Valve');
```

O resultado deverá ser exibido ao usuário em forma de tabela conforme demonstra a imagem a seguir.



The screenshot shows a web browser window with the address bar displaying 'localhost/VoceNoComando/tabela/jogos.php'. The page content features a table titled 'Listagem de Jogos' with the following data:

Código	Nome	Fabricante
1	FIFA 2020	EA
2	FINAL FANTASY	Square Enix
3	GTA	Rockstar Games
4	CS: GO	Valve

Imagem 16. Imagem tabela de lista de jogos.

Confira abaixo se você conseguiu resolver o desafio propostos!

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```

<title>Tabela Jogos - MYSQLI</title>
</head>
<body class="w3-black">
    <div class="w3-padding w3-content w3-half w3-display-topmiddle w3-
margin">
        <h1 class="w3-center w3-orange w3-round-large w3-
margin">Listagem de Jogos</h1>
        <table class="w3-table-all w3-centered w3-text-black">
            <thead>
                <tr class="w3-center w3-orange ">
                    <th>Código</th>
                    <th>Nome</th>
                    <th>Fabricante</th>
                </tr>
            </thead>
            <?php
                $servername = "localhost";
                $username = "root";
                $password = "usbw";
                $dbname = "pwii";
                $conexao = new mysqli($servername, $username, $password, $dbname);
                if ($conexao->connect_error) {
                    die("Connection failed: " . $conexao->connect_error);
                }
                $sql = "SELECT * FROM jogo" ;
                $resultado = $conexao->query($sql);
                if($resultado != null)
                    foreach($resultado as $linha) {
                        echo '<tr>';
                        echo '<td>'.$linha['idjogo'].'</td>';
                        echo '<td>'.$linha['nome'].'</td>';
                        echo '<td>'.$linha['fabricante'].'</td>';
                        echo '</tr>';
                    }
                $conexao->close();
            ?>
        </table>
    </div>
</body>
</html>

```



ATENÇÃO - Será disponibilizado o arquivo **usbwebserver - Agenda6**, nele terá todos os arquivos php e tabelas no banco de dados do mergulhando no tema, você no comando e itens descritos na atividade online disponíveis para uso e consulta.



Envio de Atividade



Imagem 17. Gabriela comemorando a oportunidade.
(STUDIOGSTOCK / FREEPIK, 2020).

Inês, professora de Inglês, ficou sabendo que Gabriela estudava programação, ficou interessada e pediu para ela desenvolver uma página para exibir as notas de cada um dos 4 módulos de inglês, de cada aluno concluinte da escola, oferecendo para Gabriela desconto em sua mensalidade.

Gabriela ficou muito interessada e começou a buscar informações para desenvolver o projeto. Por sorte a escola, possui um banco de dados, que contém uma tabela, com o código do aluno, seu nome e as notas dos quatros módulos de cada um dos alunos concluintes.

Utilizando o que foi visto até agora, vamos criar a página solicitada, com a seguinte tabela.

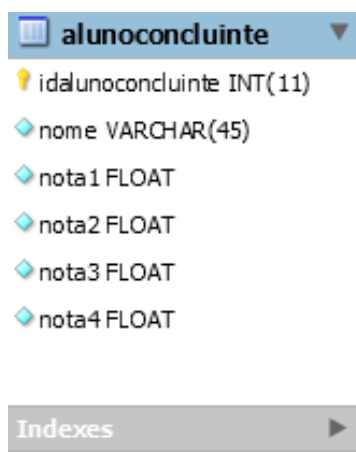


Imagem 18. Diagrama da tabela alunoconcluinte.

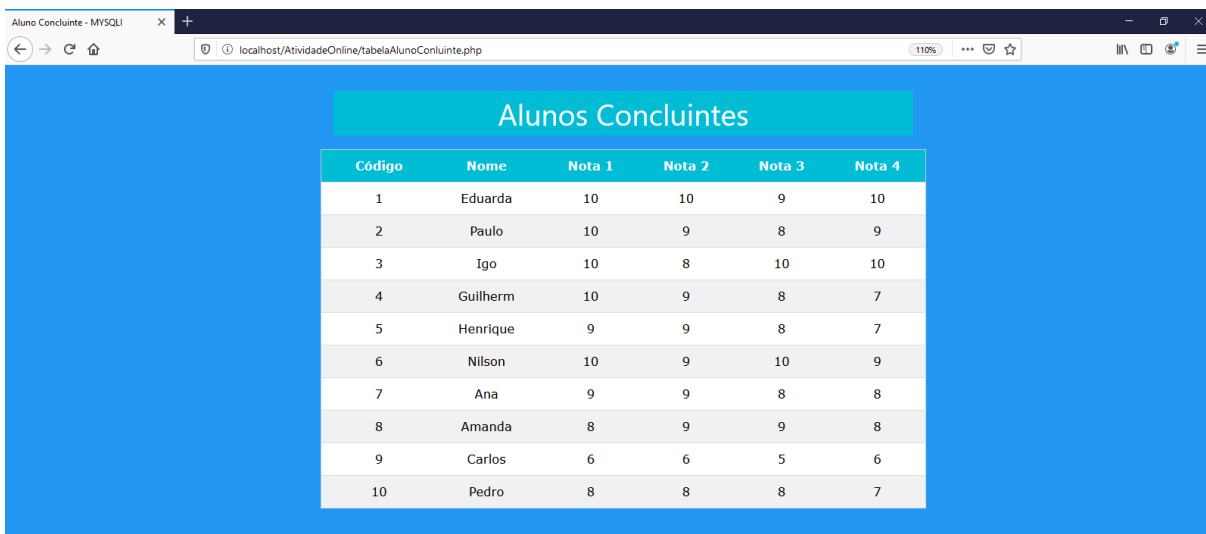
Caso esteja com dificuldade a seguir script sql para criação da tabela AlunoConcluinte

```
CREATE TABLE `pwii`.`alunoconcluinte` (
  `idalunoconcluinte` INT(11) NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  `nota1` FLOAT NOT NULL,
  `nota2` FLOAT NOT NULL,
  `nota3` FLOAT NOT NULL,
  `nota4` FLOAT NOT NULL,
  PRIMARY KEY (`idalunoconcluinte`));
```

Caso esteja com dificuldade, a seguir script sql inserção de alguns registros de alunos.

```
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Eduarda', '10', '10', '9', '10');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Paulo', '10', '9', '8', '9');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Igo', '10', '8', '10', '10');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Guilherm', '10', '9', '8', '7');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Henrique', '9', '9', '8', '7');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Nilson', '10', '9', '10', '9');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Ana', '9', '9', '8', '8');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Amanda', '8', '9', '9', '8');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Carlos', '6', '6', '5', '6');
INSERT INTO `pwii`.`alunoconcluinte` (`nome`, `nota1`, `nota2`, `nota3`, `nota4`) VALUES ('Pedro', '8', '8', '8', '7');
```

O resultado deverá ser exibido ao usuário em forma de tabela conforme demonstra a imagem a seguir.



A screenshot of a web browser window displaying a table titled "Alunos Concluintes". The browser's address bar shows the URL "localhost/AtividadeOnline/tabelaAlunoConcluinte.php". The table has 6 columns: "Código", "Nome", "Nota 1", "Nota 2", "Nota 3", and "Nota 4". It contains 10 rows of student data.

Código	Nome	Nota 1	Nota 2	Nota 3	Nota 4
1	Eduarda	10	10	9	10
2	Paulo	10	9	8	9
3	Igo	10	8	10	10
4	Guilherm	10	9	8	7
5	Henrique	9	9	8	7
6	Nilson	10	9	10	9
7	Ana	9	9	8	8
8	Amanda	8	9	9	8
9	Carlos	6	6	5	6
10	Pedro	8	8	8	7

Imagem 19. Exemplo de tabela solicitada para Gabriela.

Vamos ajudar Gabriela a Desenvolver esta atividade!

Dicas:

- Você pode utilizar html e css para deixar a página mais agradável
- Utilize o exercício você no comando como base para a resolução desse exercício.



ATENÇÃO - Será disponibilizado o arquivo **usbwebserver - Agenda6**, nele terá todos os arquivos php e tabelas no banco de dados do mergulhando no tema, você no comando e itens descritos na atividade online disponíveis para uso e consulta.



Vídeos:

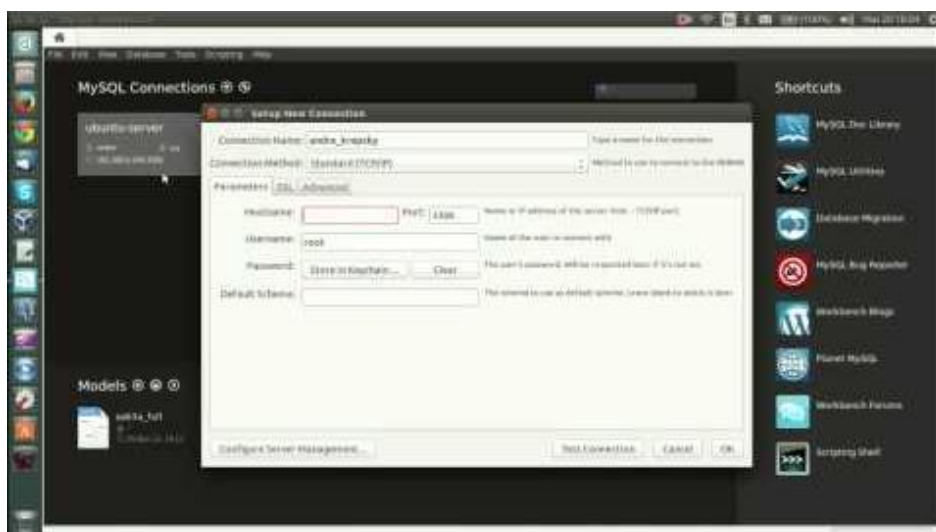
Não deixe de assistir aos vídeos:

phpMyAdmin: curso rápido em 17 min

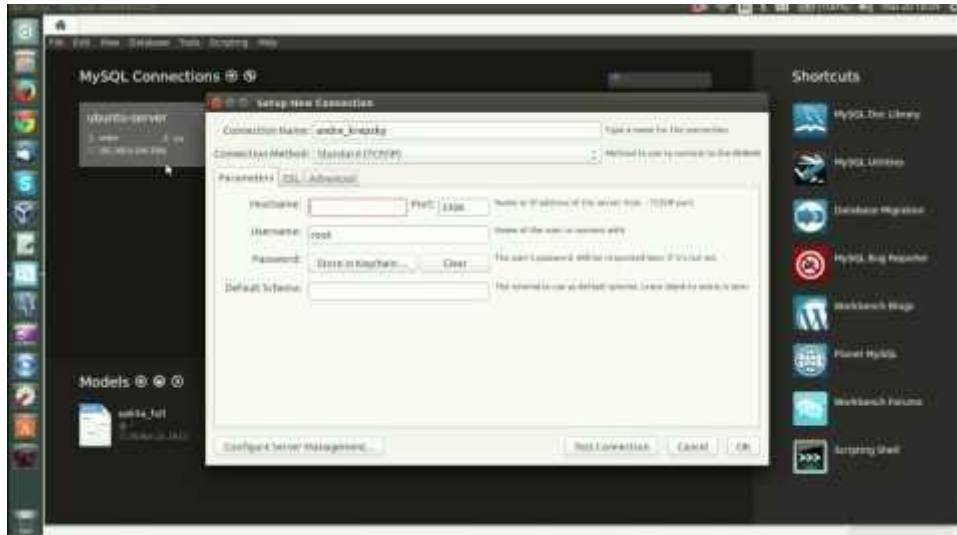


HomeHost - phpMyAdmin: curso rápido em 17 min | MySQL | Insert, Select, Update, Delete, Create table e Backup. Disponível em <https://www.youtube.com/watch?v=kviT7G14gqk>. Acessado em 30/06/2020.

Criando conexão no MySQL Workbench 6.



Professor DB - 01 - Criando conexão no MySQL Workbench 6. Disponível em <https://www.youtube.com/watch?v=xzAdW7fW0p8>. Acessado em 30/06/2020.

CREATE TABLE: Construção da tabela Titulos

Professor DB - REATE TABLE: Construção da tabela Titulos. Disponível em <https://www.youtube.com/watch?v=0Uldzp4UI0w>. Acessado em 30/06/2020.

Links:

Thiago Belem – Guia prático de MySQLi no PHP. Acessado em 21/03/2025

<http://blog.thiagobelem.net/guia-pratico-de-mysqli-no-php>.

Livro:

OGGIO, Pablo D. **PHP: Programando Com Orientacao a Objetos**. 4ª Edição. São Paulo. Novatec. 2018. ISBN: 978-8575226919