
AGENDA 7

PHP: PERSISTÊNCIA DE DADOS PARTE 2 - PDO



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

PDO

PDO conceitualmente trata-se de uma interface PHP Data Objects (PDO), que ao ser implementada, fornece uma camada de abstração em relação a conexão com o banco de dados, trazendo o grande benefício de efetuar a conexão com diversos bancos de dados da mesma forma, alterando basicamente sua string de conexão.

Podemos observar através da tabela a seguir, as semelhanças e diferenças entre PDO e MySQLi.

	PDO	MySQLi
Suporte a Bancos de Dados	12 drivers diferentes	Somente MySQL
API	Orientada a Objetos	Orientada a Objetos + Procedural
Conexão	Fácil	Fácil
Parâmetros Nomeados	Sim	Não
Mapeamento de Objetos	Sim	Sim
Sentenças Preparadas (lado do cliente)	Sim	Não
Performance	Rápido	mais Rápido
Procedimentos Armazenados	Sim	Sim

Obs.: Em alguns casos, o PDO deve ser habilitado no servidor para ter seu funcionamento adequado. No caso do USBWebServer, é possível ser habilitado a partir do arquivo php.ini, localizado na pasta settings, removendo o comentário (;) da `extension=php_pdo_mysql.dll`.

```
908 extension=php_openssl.dll
909 ;extension=php_pdo_firebird.dll
910 extension=php_pdo_mysql.dll
911 ;extension=php_pdo_oci.dll
912 ;extension=php_pdo_odbc.dll
913 extension=php_pdo_pgsql.dll
914 extension=php_pdo_sqlite.dll
```

Imagem 3. `extension=php_pdo_mysql.dll`, sem comentário

Em outros servidores talvez também seja necessário, a ativação do: `extension=php_pdo.dll`, que também é realizada removendo o (;) .

Vamos começar! Seguindo mesmo padrão da agenda anterior, será necessário criar uma base de dados e uma tabela. Continuaremos a utilizar o banco de dados MySQL.

Para começarmos a base de dados denominada “pwii” ou você pode usar a mesma base da agenda anterior.

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

Ou

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

Também crie uma tabela nesta base denominada produto, que contenha os campos:

- id (chave primária e auto incremento).
- Nome.
- Preço.
- quantidade.

Conforme mostra o diagrama a seguir.

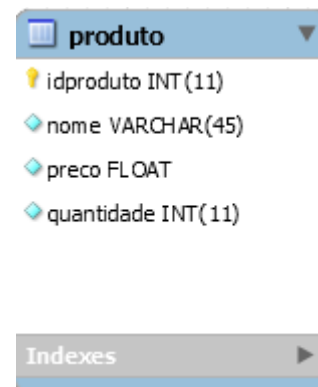


Imagem 6.
Diagrama
Banco de dados.

ou utilize o script a seguir:

```
CREATE TABLE `pwii`.`produto` (  
  `idproduto` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `preco` FLOAT NOT NULL,  
  `quantidade` INT NOT NULL,  
  PRIMARY KEY (`idproduto`));
```

Neste exemplo, também será necessário criar diversos arquivos.

No visual studio Code, dentro da pasta Agenda7 (ou outra pasta de sua preferência), crie o primeiro arquivo com o nome “index.php” - Neste arquivo vamos criar quatro links, que terá o propósito de direcionar a navegação 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 - PDO</title>
</head>
<body class="w3-black">
<h1 class=" w3-center w3-purple w3-margin w3-padding w3-round-
large">Projeto Lista de Produtos</h1>
<div class="w3-container w3-col w3-row w3-center  ">
  <div class="w3-col w3-button w3-blue w3-cell w3-round-large w3-margin-
right w3-margin-left" style="width:23%;">
    <a href="cadastro.php" style="text-decoration: none;">
      <i class="fa fa-cube" style="font-size: 10.5em"></i>
      <p style="font-size: 2em">Adicionar</p>
    </a>
  </div>
  <div class="w3-col w3-button w3-red w3-cell w3-round-large w3-margin-
right w3-text-white" style="width:23%;">
    <a href="listar.php" style="text-decoration: none;">
      <i class=" fa fa-trash" style="font-size: 10.5em"></i>
      <p style="font-size: 2em">Remover</p>
    </a>
  </div>
  <div class="w3-col w3-button w3-deep-purple w3-cell w3-round-large w3-
margin-right" style="width:23%;">
    <a href="listar.php" style="text-decoration: none;">
      <i class=" fa fa-refresh" style="font-size: 10.5em"></i>
      <p style="font-size: 2em">Atualizar</p>
    </a>
  </div>
  <div class="w3-col w3-button w3-indigo w3-cell w3-round-large w3-margin-
right" style="width:23%;">
    <a href="listar.php" style="text-decoration: none;">
      <i class=" fa fa-list-alt" 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 quatros links criados nos encaminham para 2 arquivos diferentes: cadastro.php (link adicionar) e listar.php (links Listar, Atualizar e Exlcuir).

O resultado no Navegador será como o representado na imagem a seguir.

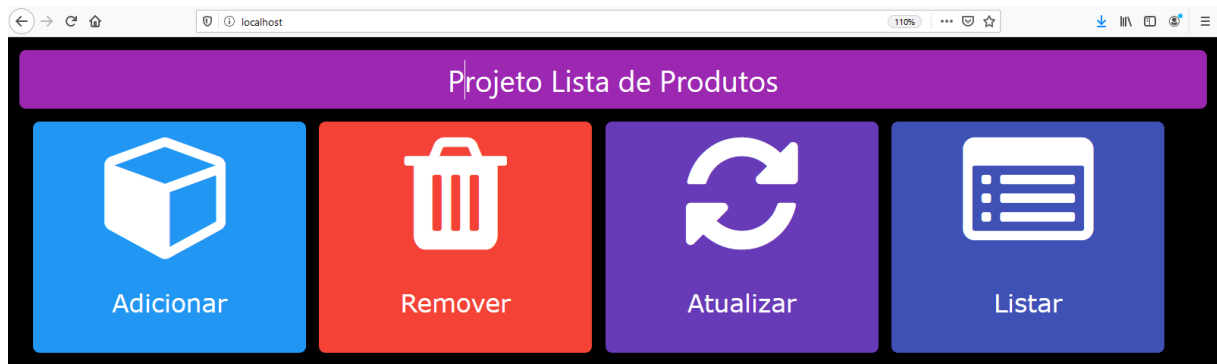


Imagem 7. Resultado no Navegador.

Continuando, crie o arquivo “cadastro.php”, na mesma pasta do projeto. Este arquivo terá um formulário com quatro campos: ID, Nome, Preço e quantidade, ao final, também será criado um botão para adicionar o produto.

```
<!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 - PDO</title>
</head>
<body class="w3-black">
<a href="index.php" class="w3-display-topleft ">
  <i class="fa fa-arrow-circle-left w3-large w3-blue 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-blue w3-round-large w3-
margin">Cadastro de Produtos</h1>
  <form action="cadastroAction.php" class="w3-container" method='post'>
    <label class="w3-text-blue" style="font-
weight: bold;">Código</label>
    <input name="txtID" class="w3-input w3-grey w3-
border" disabled><br>
    <label class="w3-text-blue" style="font-
weight: bold;">Nome</label>
```

```

        <input name="txtNome" class="w3-input w3-light-grey w3-
border"><br>
        <label class="w3-text-blue" style="font-
weight: bold;">Preço</label>
        <input name="txtPreco" class="w3-input w3-light-grey w3-
border"><br>
        <label class="w3-text-blue" style="font-
weight: bold;">Quantidade</label>
        <input name="txtQtd" class="w3-input w3-light-grey w3-border"><br>
        <button name="btnAdd" class="w3-button w3-blue w3-cell w3-round-
large w3-right w3-margin-right">
            <i class="w3-xxlarge fa fa-plus-square"></i> Adicionar
        </button>
    </form>
</div>
</body>
</html>

```

Resultado no navegador:



The image shows a web browser window with a dark background. At the top, there is a blue header bar with the text "Cadastro de Produtos" in white. Below the header, there are four input fields, each with a label above it: "Código", "Nome", "Preço", and "Quantidade". The labels are in a light blue color. The input fields are white with a light grey border. At the bottom right of the form, there is a blue button with a white plus icon and the text "Adicionar" in white.

Imagem 8. Resultado no Navegador.

Neste código, a action do formulário tem como valor atribuído o arquivo “cadastroAction.php”, Arquivo de vamos criar na mesma pasta do nosso projeto. Neste arquivo, terá a conexão com o banco de dados através do PDO, a criação e execução da sentença sql de insert, e por fim uma mensagem de sucesso ou fracasso.

Para criar a conexão, continuamos com a boa prática criar variáveis para armazenar o nome do servidor, nome do usuário, senha e por fim o nome da base de dados. Neste momento criamos uma instância do driver PDO.

```
$host = "localhost";
$usuario = "root";
$senha = "usbw";
$bd = "pwii";
try{
    $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306; charset=utf8", $usuario, $senha);
    $conecta->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}catch(PDOException $e){
    echo "falha ao conectar: ". $e->getMessage();
}
```

Ao observar o código anterior, percebemos que dentro do método construtor, há uma string com a indicação “mysql:”, essa indicação faz com que a instância seja feita para uma base de dados que utilize o SGBD Mysql. Perceba também que foi determinada uma porta e charset utilizado para salvar os dados na base.

Perceba também que neste exemplo, foi acionada um método setAttribute, sua principal função é alterar o modo que o PDO apresenta seus erros, fazendo com que seja lançada uma exceção, para quando eventuais problemas surgirem, assim, já justificamos o uso do try...catch, que pegará a exceção lançada, e oferta recursos para manipularmos essas informações sobre o erro em questão.

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

```
$sql = "INSERT INTO produto (nome, preco, quantidade)
VALUES ('".$_POST['txtNome']. "', '".$_POST['txtPreco']. "', '".$_POST['txtQtd']. "')";

try{
    $resultado = $conecta->query($sql);
    echo '
    <a href="index.php">
        <h1 class="w3-button w3-blue">Produto Salvo com sucesso! </h1>
    </a>
    ';
```



```

    }catch(PDOException $e){
        echo '
            <a href="index.php">
                <h1 class="w3-button w3-blue">ERRO! </h1>
            </a>
        ';
    }

```

A execução da query sql, está sendo realizada dentro de uma estrutura try, verificando se mesma obteve sucesso em sua ação, gerando assim uma mensagem de sucesso em caso verdadeiro, e uma mensagem de cado se falha (alguma exceção aconteça).

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 - PDO</title>
</head>
<body class="w3-black">
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-middle">
    <?php
        $host = "localhost";
        $usuario = "root";
        $senha = "usbw";
        $bd = "pwii";
        try{
            $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306; charset=utf8", $usuario, $senha);
            $conecta->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch(PDOException $e){
            echo "falha ao conectar: ". $e->getMessage();
        }

        $sql = "INSERT INTO produto (nome, preco, quantidade)
        VALUES ('".$_POST['txtNome']. "', '".$_POST['txtPreco']. "', '".$_POST['txtQtd']. "')";

        try{
            $resultado = $conecta->query($sql);
            echo '
                <a href="index.php">
                    <h1 class="w3-button w3-blue">Produto Salvo com sucesso! </h1>
            '

```

```

        </a>
        ';
    }catch(PDOException $e){
        echo '
            <a href="index.php">
                <h1 class="w3-button w3-blue">ERRO! </h1>
            </a>
        ';
    }
?>
</div>
</body>
</html>

```

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:

Produto Salvo com sucesso!

Imagem 9. Resultado no Navegador para mensagem de sucesso.

Obs: A mensagem de falha segue o mesmo design.

Dando continuidade, nossa tarefa agora é criar o arquivo “listar.php” (arquivo do quarto 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, Preço, Quantidade, remover e atualizar produto, 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

Tudo seguindo o mesmo padrão da inserção de dados na base. No entanto, para a setença select, será retornado uma matriz de dados, ou seja, precisamos atribuir o resultado em uma variável.

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

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>Lista de Produtos - PDO</title>
</head>
<body class="w3-black">
<a href="index.php" class="w3-display-topleft">
  <i class="fa fa-arrow-circle-left w3-large w3-indigo w3-button w3-xxlarge"></i>
</a>

  <?php
    $host = "localhost";
    $usuario = "root";
    $senha = "usbw";
    $bd = "pwii";
    try{
      $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306; charset=utf8", $usuario, $senha);
      $conecta->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```

        echo '
        <div class="w3-padding w3-content w3-half w3-display-
topmiddle w3-margin">
        <h1 class="w3-center w3-indigo w3-round-large w3-
margin">Lista de Produtos</h1>
        <table class="w3-table-all w3-centered w3-text-black">
        <thead>
            <tr class="w3-center w3-indigo">
                <th>Código</th>
                <th>Nome</th>
                <th>Preço</th>
                <th>Quantidade</th>
                <th>Excluir</th>
                <th>Atualizar</th>
            </tr>
        </thead>
        ';
        $sql = "SELECT * FROM produto" ;
        $resultado = $conecta->query($sql);
        if($resultado != null)
        foreach($resultado as $linha) {
            echo '<tr>';
            echo '<td>'.$linha['idproduto'].'</td>';
            echo '<td>'.$linha['nome'].'</td>';
            echo '<td>'.$linha['preco'].'</td>';
            echo '<td>'.$linha['quantidade'].'</td>';
            echo '<td><a href="excluir.php?id='.$linha['idproduto'].'&
nome='.$linha['nome'].'&preco='.$linha['preco'].'&quantidade='.$linha['quantid
ade'].'><i class="fa fa-user-times w3-large w3-text-indigo"></i> </a></td>';
            echo '<td><a href="atualizar.php?id='.$linha['idproduto'].'
&nome='.$linha['nome'].'&preco='.$linha['preco'].'&quantidade='.$linha['quant
idade'].'><i class="fa fa-refresh w3-large w3-text-indigo"></i> </a></td>';
            echo '</tr>';
        }
        echo '
        </table>
        </div>';
    }catch(PDOException $e){
        echo "falha ao conectar: ". $e->getMessage();
    }

    ?>
</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 fluído.

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



Imagem 10. Resultado no Navegador para lista de produto.

Agora é o momento para a criação de mais dois arquivos: excluir e excluir action, para o primeiro vamos criar um formulário com campos nome, preco e quantidade 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 produtos.

Para a construção dessa página, 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 - PDO</title>
</head>
<body class="w3-black">
<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-indigo 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-indigo" style="font-weight: bold;">Nome</label>
    <input name="txtNome" class="w3-input w3-grey w3-
border" disabled value="<?php echo $_GET['nome']?>">
    <br>
    <label class="w3-text-indigo" style="font-weight: bold;">Preco</label>
    <input name="txtPreco" class="w3-input w3-gray w3-
border" disabled value="<?php echo $_GET['preco']?>">
    <br>
    <label class="w3-text-indigo" style="font-
weight: bold;">Quantidade</label>
    <input name="txtQtd" class="w3-input w3-gray w3-
border" disabled value="<?php echo $_GET['quantidade']?>">
    <br>
    <button name="btnExcluir" class="w3-button w3-indigo w3-cell w3-round-
large w3-right">
      <i class="w3-xxlarge fa fa-check"></i> Confirmar Exclusão.
    </button>
  </form>
</div>
</body>
</html>
```

Como o atributo action deste form, indica o arquivo “excluirAction”, precisamos desenvolvê-lo. Para isso vamos:

- 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.

Neste exemplo, vamos modificar um pouco a forma como montamos a sentença SQL, através de um recurso de preparação de parâmetro.

```
$sql = $conecta->prepare("DELETE FROM produto WHERE idproduto = ?;");  
$sql->bindParam(1, $_POST['txtID']);  
$sql->execute();
```

Conforme podemos observar no código anterior, começamos preparando a string no SQL utilizando o método prepare, que tem a função apenas de iniciar uma sentença SQL, percebe a presença de um caracter de interrogação (?), este caracter será substituído pelos valores adicionados, através do uso método bindParam.

O método bindParam neste exemplo utiliza duas passagens de parâmetros. A primeira (1) trata-se da posição da interrogação que será substituído (caso existam mais de uma interrogação, para cada uma delas é dada um número em ordem crescente da direita para esquerda) e o segundo parâmetro (\$_POST['txtID']) destina-se ao valor que será inserido na sentença sql.

Obs.: Pode existir um terceiro parâmetro, referente ao tipo de valor que será enviado.

Ficou alguma dúvida? Complemente seus estudos e assista ao vídeo:

Prepared Statements no PHP Orientado a Objetos com PDO.



UpInside - Prepared Statements no PHP Orientado a Objetos com PDO (aula #33)- Disponível

em: <https://www.youtube.com/watch?v=I0Xf4hyWWqw>. Acessado em 12/09/2025.

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 - PDO</title>
</head>
<body class="w3-black">
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-middle">
  <?php
    $host = "localhost";
    $usuario = "root";
    $senha = "usbw";
    $bd = "pwii";
    try{
      $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306; chars
et=utf8", $usuario, $senha);
      $conecta->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }catch(PDOException $e){
      echo "falha ao conectar: ". $e->getMessage();
    }
    try{
      $sql = $conecta-
>prepare("DELETE FROM produto WHERE idproduto = ?");
      $sql->bindParam(1,$_POST['txtID']);
```



```

        $sql->execute();
        echo '
        <a href="listar.php">
            <h1 class="w3-button w3-
indigo">Produto Excluido com sucesso! </h1>
        </a>
        ';
    }catch(PDOException $e){
        echo '
        <a href="listar.php">
            <h1 class="w3-button w3-indigo">ERRO! </h1>
        </a>
        ';
    }
    ?>
</div>
</body>
</html>

```

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

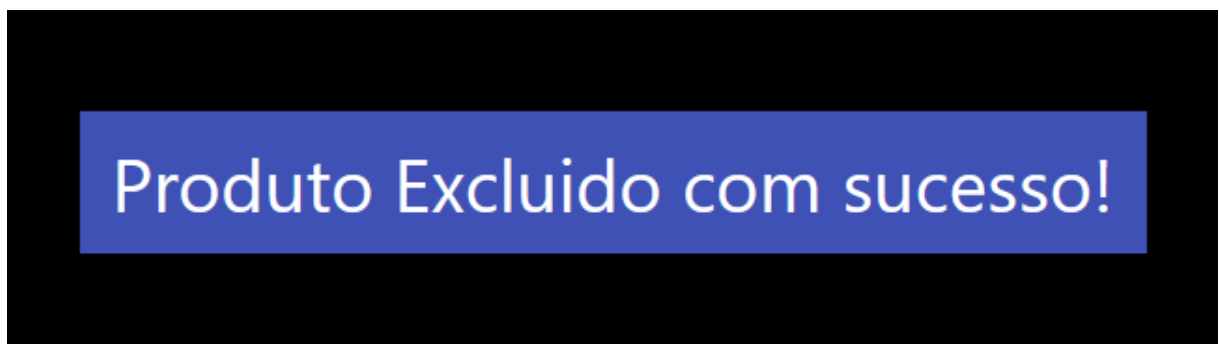


Imagem 12. Resultado no Navegador para mensagem de sucesso.

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” e “atualizarAction”. Primeiro vamos criar um formulário com inputs para o nome, preço e quantidade, que receberá os dados oriundos do método get do arquivo “listar.php”, ofertando a possibilidade do usuário realizar alterações e o update através de botão “atualizar”, o resultado deve ser como o apresentado na imagem a seguir.


```

<h1 class="w3-center w3-deep-purple 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-deep-purple " 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-deep-purple " style="font-
weight: bold;">Preço</label>
    <input name="txtPreco" class="w3-input w3-light-grey w3-
border" value="<?php echo $_GET['preco']?>">
    <br>
    <label class="w3-text-deep-purple " style="font-
weight: bold;">Quantidade</label>
    <input name="txtQtd" class="w3-input w3-light-grey w3-
border" value="<?php echo $_GET['quantidade']?>">
    <br>
    <button name="btnAtualizar" class="w3-button w3-deep-purple 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”, precisamos desenvolvê-lo. Para isso vamos:

- Criar instância PDO
- 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

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 - PDO</title>
</head>
<body class="w3-black">
<div class="w3-padding w3-content w3-text-grey w3-third w3-display-middle">
    <?php
        $host = "localhost";
        $usuario = "root";
        $senha = "usbw";
        $bd = "pwii";
        try{
            $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306; charset=utf8", $usuario, $senha);
            $conecta->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch(PDOException $e){
            echo "falha ao conectar: ". $e->getMessage();
        }
        try{
            $sql = $conecta->prepare("UPDATE produto SET nome = ?, preco = ?,
            quantidade=? WHERE idproduto = ?");
            $sql->bindParam(1,$_POST['txtNome']);
            $sql->bindParam(2,$_POST['txtPreco']);
            $sql->bindParam(3,$_POST['txtQtd']);
            $sql->bindParam(4,$_POST['txtID']);
            $sql->execute();
            echo '
            <a href="listar.php">
                <h1 class="w3-button w3-deep-purple ">Produto Atualizado com sucesso! </h1>
            </a>
            ';
        }catch(PDOException $e){
            echo '
            <a href="listar.php">
                <h1 class="w3-button w3-deep-purple ">ERRO! </h1>
            </a>
            ';
        }
    ?>
</div>
</body>
</html>

```

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

Produto 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 disciplinas, há outros métodos para fazer todas as operações contidas nessa agenda.



ATENÇÃO - Será disponibilizado o arquivo – **Agenda7**, nele terá todos os arquivos php e tabelas no banco de dados do mergulhando no tema e você no comando.



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

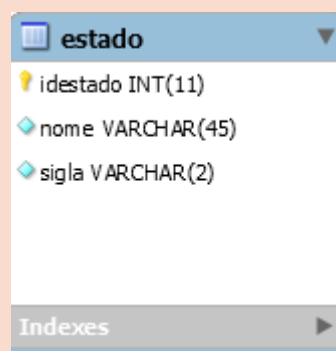


Imagem 16.
Diagrama
Banco de dados.

1. Obtenha os dados da tabela, utilizando o driver PDO
2. Crie uma tabela no navegador 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 estado

```
CREATE TABLE `pwii`.`estado` (  
  `idestado` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  `sigla` VARCHAR(2) NOT NULL,  
  PRIMARY KEY (`idestado`));
```

Caso esteja com dificuldade, a seguir script sql inserção de todos registros de estados.

```
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Acre','AC');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Alagoas','AL');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Amapá','AP');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Amazonas','AM');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Bahia','BA');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Ceará','CE');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Espírito Santo','ES');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Goiás','GO');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Maranhão','MA');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Mato Grosso','MT');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Mato Grosso do Sul','MS');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Minas Gerais','MG');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Pará','PA');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Paraíba','PB');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Paraná','PR');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Pernambuco','PE');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Piauí','PI');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Rio de Janeiro','RJ');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Rio Grande do Norte','RN');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Rio Grande do Sul','RS');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Rondônia','RO');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Roraima','RR');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Santa Catarina','SC');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('São Paulo','SP');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Sergipe','SE');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Tocantins','TO');  
INSERT INTO `pwii`.`estado` (`nome`, `sigla`) VALUES ('Distrito Federal','DF');  
;
```

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



Código	Nome	Sigla
1	Acre	AC
2	Alagoas	AL
3	Amapá	AP
4	Amazonas	AM
5	Bahia	BA
6	Ceará	CE
7	Espírito Santo	ES
8	Goiás	GO
9	Maranhão	MA
10	Mato Grosso	MT
11	Mato Grosso do Sul	MS
12	Minas Gerais	MG
13	Pará	PA
14	Paraíba	PB

Imagem 16. Imagem tabela de lista de todos os Estados.

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 de Estados - PDO</title>
</head>
<body style="background-color: blue;">
  <div class="w3-paddingw3-content w3-half w3-display-topmiddle w3-margin">
    <h1 class="w3-center w3-yellow w3-round-large w3-margin">Lista de Estados</h1>
    <table class="w3-table-all w3-centered w3-text-black">
      <thead>
        <tr class="w3-center w3-green ">
          <th>Código</th>
          <th>Nome</th>
          <th>Sigla</th>
        </tr>
      <thead>
        <?php
```

```

        $host = "localhost";
        $usuario = "root";
        $senha = "usbw";
        $bd = "pwii";
        try{
            $conecta = new PDO("mysql:dbname=$bd; host=$host; port=3306;
charset=utf8", $usuario, $senha);
            $conecta-
>setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch(PDOException $e){
            echo "falha ao conectar: ". $e->getMessage();
        }
        try{
            $sql = "SELECT * FROM estado";
            $resultado = $conecta->query($sql);

            foreach($resultado as $linha) {
                echo '<tr>';
                echo '<td>'.$linha['idestado'].'</td>';
                echo '<td>'.$linha['nome'].'</td>';
                echo '<td>'.$linha['sigla'].'</td>';
                echo '</tr>';
            }
        }catch(PDOException $e){
            echo '
                <h1 class="w3-button w3-indigo">ERRO! </h1>
            ';
        }
    ?>
</table>
</div>
</body>
</html>

```



ATENÇÃO - Será disponibilizado o arquivo – **Agenda7.rar**, nele terá todos os arquivos php e tabelas no banco de dados do mergulhando no tema, você no comando disponíveis para uso e consulta.