

# MÓDULO

# PROGRAMAÇÃO PHP

## PRÁTICA 30

CONEXÃO, ACESSO E LISTAGEM DE UMA  
BASE DE DADOS UTILIZANDO O MYSQL, COM  
AJAX E PHP

## CONEXÃO, ACESSO E LISTAGEM DE UMA BASE DE DADOS UTILIZANDO O MYSQL, COM AJAX E PHP

---

### 1.1. DESCRIÇÃO DA PRÁTICA

Tempo estimado: 1 h.

REQUISITOS
■ Ter concluído a unidade didática “PHP e JavaScript/Ajax”.
OBJETIVOS
■ Aprender a carregar dados de forma assíncrona com Ajax e PHP.
■ Conhecer um pouco da linguagem SQL para fazer consultas simples.
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
■ Editor de código.
■ Servidor web (XAMPP, MAMP, ou similar).

## 1.2. DESENVOLVIMENTO DA PRÁTICA

Nesta prática, verá uma das utilidades mais importantes da programação de páginas web com PHP: aceder e listar dados de uma base de dados MySQL.

É bem possível que tenha uma grande quantidade de dados em tabelas na base de dados, mas, para exibi-los, para interagir com eles e para aceder às informações de que necessita no momento, é necessária uma forma de conectar a essa base de dados, que é possível graças aos processos que são gerados em tempo de execução.

Para esta prática, é necessário conhecer a linguagem utilizada nas bases de dados, neste caso MySQL, para poder utilizar as consultas necessárias no código.

Neste exemplo específico vão ser criadas oito consultas que vão devolver resultados diferentes, mas a estrutura da lista de dados geralmente será semelhante. Assim, cada consulta será explicada de forma independente e veremos o que cada uma faz e o resultado que devolve.

Para que não seja uma prática em que são vistas apenas algumas consultas numa página estática, será incluído um pouco de jQuery/Ajax no código HTML para realizar consultas numa parte do index.php sem recarregar a página inteira.

A ideia desta prática é ter uma div com um formulário a partir do qual serão carregadas as consultas selecionadas, juntamente com outra div, onde as informações da consulta que selecionar serão mostradas.

Esta será a janela inicial, com o tradicional design que tem sido utilizado nas práticas:



The screenshot shows a web application titled 'master.D' in a dark blue header. Below the header, the text 'Listagens MySQL apartir do PHP com jQuery/Ajax' is displayed. A dropdown menu labeled 'Selecionar uma consulta' is positioned above a large, empty text input field. A 'Consultar' button is located below the input field.

Ao selecionar uma consulta, o valor da seleção será mostrado na caixa de texto e será lançado quando clicar no botão Consultar. Isto mostrará a div inferior, onde aparecerá a lista selecionada.



The screenshot shows the same web application, but the dropdown menu now displays 'Consulta 1'. The text input field below it contains the SQL query 'SELECT \* FROM tbl\_alunos'. The 'Consultar' button remains at the bottom.

Ao escolher a primeira consulta no select do formulário, o valor da consulta selecionada será carregado automaticamente no campo de texto, que será lançado ao premir o botão Consultar. É aqui que o Ajax entra em ação, inserindo o código PHP na div que estava oculta com o jQuery e o seu método fadeOut. Irá recolher o valor do select selecionado e processá-lo dentro do código PHP para mostrar o resultado na div inferior, como pode ser visto na imagem a seguir.

# master.D

Listagens MySQL apartir do PHP com jQuery/Ajax

Consulta 1 ▼

```
SELECT * FROM tbl_alunos
```

Consultar

ID	Nome	Apelidos	Idade
1	João	Santos Teste	22
2	Maria	Silva Teste	25
3	André	Rodrigues Teste	20
4	José	Fernandes Teste	21
5	Margarida	Cruz Teste	26
6	Pedro	Pereira Teste	22
7	João	Guilherme Teste	25
8	Ana	Reis Teste	20
9	Soraia	Dias Teste	23
10	Díogo	Pires Teste	24

Logicamente, é necessário mostrar a div correspondente com o método jQuery fadeIn, caso contrário, a lista que acaba de ser invocada não será visível. Juntamente com a div, verá as diferentes mensagens do servidor, indicando se foi ou não possível conectar-se à base de dados e se a conexão foi encerrada corretamente. Também poderá ver as informações dessa mesma conexão. No caso que pode ver na imagem, a conexão foi feita através do localhost:8889 via TCP/IP.

## A estrutura dos ficheiros

---

Nesta prática, irá fazer várias conexões à base de dados com uma consulta por conexão, selecionada na forma superior. Para carregar com Ajax, o código PHP é dividido em três ficheiros: dois para a conexão e encerramento da base de dados e o terceiro, e mais importante, que recolhe as consultas selecionadas e as executa no MySQL por meio de queries ou consultas, chamando a conexão e fechando os ficheiros PHP de acordo com a demanda de cada operação.

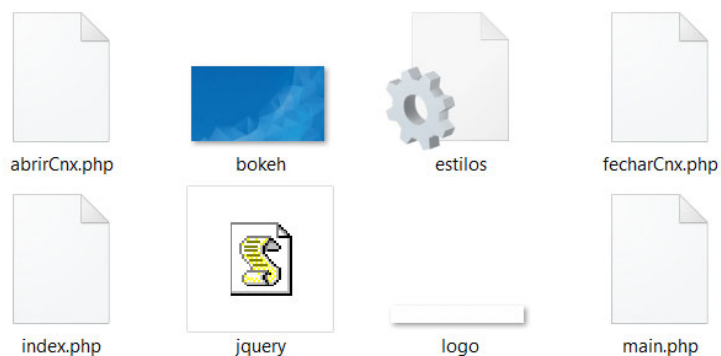
Haverá, ainda, o ficheiro index.php, que será a base de trabalho de toda a prática e onde serão incluídas as referências aos outros ficheiros. Estes, por sua vez, farão o mesmo com o resto.

De seguida, terá o ficheiro de estilos CSS e um ficheiro Javascript, onde é armazenado o código jQuery que foi utilizado no index.php.

Por fim, terá as imagens utilizadas como fundo e logótipo, que são as mesmas que foram utilizadas nas práticas anteriores.

Esta será a estrutura completa dos ficheiros utilizados nesta prática:

## Base de dados MySQL



A base de dados nesta prática é muito importante, sendo que vai continuar a ser usado o exemplo de uma escola. Neste caso, será uma tabela simples, chamada `tbl_alunos`, com quatro colunas e dez linhas, mais precisamente as seguintes:

Query 1 x

Limit to 1000 rows

```
1 • SELECT * FROM tbl_alunos;
```

Result Grid

IDaluno	Nome	Apelidos	Idade
1	João	Santos Teste	22
2	Maria	Silva Teste	25
3	André	Rodrigues Teste	20
4	José	Fernandes Teste	21
5	Margarida	Cruz Teste	26
6	Pedro	Pereira Teste	22
7	João	Guilherme Teste	25
8	Ana	Reis Teste	20
9	Soraia	Dias Teste	23
10	Diogo	Pires Teste	24
*	NULL	NULL	NULL

A coluna *id*, auto increment, será o campo-chave da tabela. As restantes colunas servem apenas para mostrar os dados associados ao campo-chave e criar uma pequena estrutura de dados. Assim, não teremos muito interesse em completá-las com dados reais – os que podem ser vistos na imagem serão suficientes.

## Consultas SQL

---

Nesta prática, vão ser utilizadas oito consultas predefinidas sobre os valores das opções de seleção existentes no formulário principal. São as seguintes:

1. Mostrar todos os registos no sentido crescente do *id*.  
`$query = "SELECT * FROM alunos";`
2. Mostrar cinco registos, começando com o primeiro, que é zero.  
`$query = "SELECT * FROM alunos LIMIT 0,5";`
3. Mostrar dois registos a partir do registo número 3.  
`$query = "SELECT * FROM alunos LIMIT 3,2";`
4. Mostrar 20 registos no sentido decrescente do *id*.  
`$query = "SELECT * FROM alunos ORDER BY id DESC LIMIT 10";`
5. Mostrar dois registos a partir do registo número 3, mas no sentido decrescente do *id*.  
`$query = "SELECT * FROM alunos ORDER BY id DESC LIMIT 3,2";`
6. Mostrar os resultados onde o nome é Pedro.  
`$query = "SELECT * FROM alunos WHERE nome = 'Pedro'";`
7. Mostrar os resultados onde o *id* é 3 e o nome é Pedro.  
`$query = "SELECT * FROM alunos WHERE id = 3 AND nome = 'Pedro'";`
8. Mostrar todos os resultados onde a idade é superior ou igual a 24.  
`$query = "SELECT * FROM alunos WHERE idade >= 24";`



## O código

---

Nesta prática, serão utilizados o HTML/CSS, o PHP e o jQuery/Ajax. Já foi referida a estrutura de ficheiros a utilizar, por isso, explicar-se-á cada um dos ficheiros para ver como tudo funcionará, de modo a obter os resultados que se pretende. Começamos com o ficheiro index.php, onde é inserido o código HTML. Este ficheiro deve ser gravado na pasta htdocs do servidor web (XAMPP ou similar).

```
<html>

<head>

<meta charset="UTF-8">

<title>Prática 10</title>

<link href="estilos.css" rel="stylesheet">

<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>

<script language="javascript" type="text/javascript"
src="jquery.js"></script>

</head>
```

Este excerto de código será o cabeçalho do ficheiro, de onde serão invocados três ficheiros. O primeiro é o já conhecido ficheiro de estilos, fornecido na pasta de apoio à prática no Campus Virtual. De seguida, é invocado o ficheiro de biblioteca Ajax e, finalmente, o ficheiro da função jQuery (também fornecido na pasta de apoio).

```
<body>

<div class="caixa0">

<span id="logo"></span>

</div>
```

É claro que há o fundo e o container div, que abriga o logótipo Master.D, com o seu "cabeçalho" de classe CSS que aplica os estilos necessários.

```
<div class="caixa1" id="selector">

<p>Listagens MySQL apartir do PHP com jQuery/Ajax</p>

<form id="form1">

<select id="querys" onChange="mostrarVal()">

<option value="" selected>Selecionar uma consulta</option>

<option value="SELECT * FROM alunos">Consulta 1</option>

<option value="SELECT * FROM alunos LIMIT 0,5">Consulta 2</option>

<option value="SELECT * FROM alunos LIMIT 3,2">Consulta 3</option>

<option value="SELECT * FROM alunos ORDER BY id DESC LIMIT
10">Consulta 4</option>

<option value="SELECT * FROM alunos ORDER BY id DESC LIMIT
3,2">Consulta 5</option>

<option value="SELECT * FROM alunos WHERE nome = 'Pedro'">Consulta
6</option>

<option value="SELECT * FROM alunos WHERE id = 3 AND nome =
'Pedro'">Consulta 7</option>

<option value="SELECT * FROM alunos WHERE idade >= 24">Consulta
8</option>

</select>

<br/><br/>

<input class="textbox" id="texto" name="text" maxlength="70"
size="65" disabled />

<br/><br/>

<button type="submit" class="botao azul"
id="botao">Consultar</button>

</form>

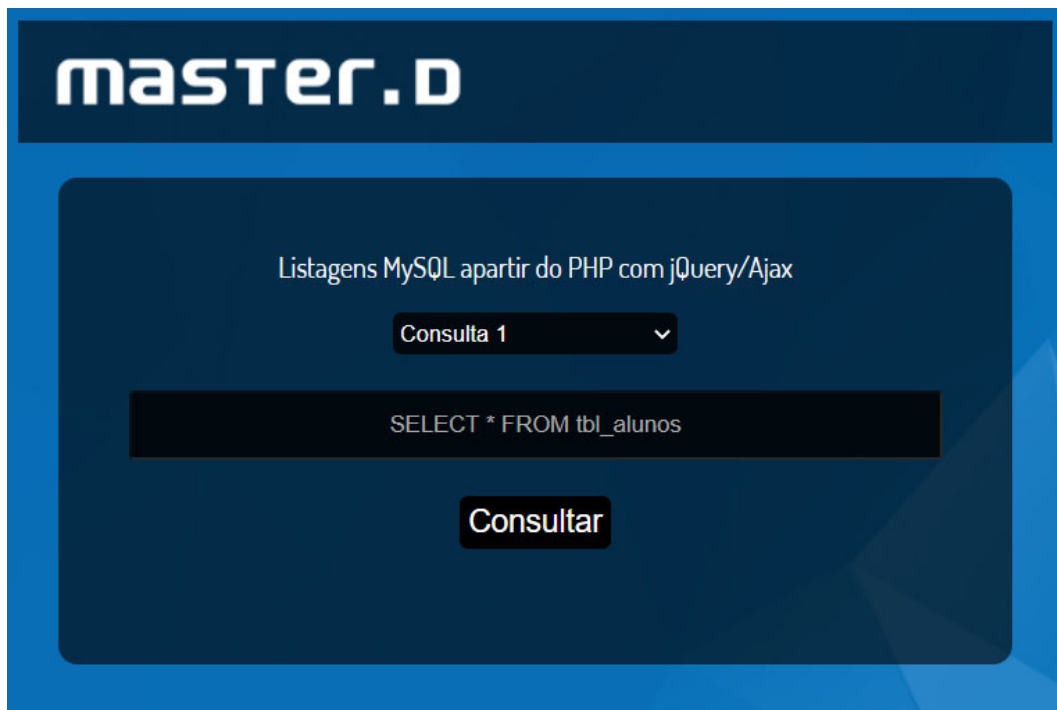
</div>
```

Esta div, com um estilo chamado caixa1 e um id selector (que será utilizado mais tarde nas funções jQuery), será utilizada para exibir as consultas escolhidas, juntamente com uma caixa de texto onde exibirá o valor de cada consulta selecionada e o botão de envio de consulta.

Quando o select muda, no ficheiro .js, a função jQuery mostrarVal() será executada.

```
function mostrarVal() {  
    $( "#texto" ).val($("#querys option:selected" ).val());  
};
```

Esta função faz com que a caixa de texto com id texto assuma o valor da opção do select com o id queries. Ao clicar na consulta que pretende escolher, o valor muda, pelo que a função jQuery é executada e o valor é modificado pelo da opção escolhida, ou seja, se seleccionar Query1, o valor da caixa de texto será SELECT \* FROM tbl\_alunos, que é o valor da opção do select escolhido.



Cada vez que é escolhida uma consulta diferente, o valor da caixa de texto com o id texto mudará.

De seguida, é necessária uma div onde seja possível visualizar a consulta, uma vez iniciada.

```
<br>

<div class="caixa1" id="conteudo"></div>

</body>

</html>
```

Com este excerto de código é, então, fechado o ficheiro index.php, que já tem o que é necessário e que estará nos restantes ficheiros onde tudo vai acontecer. É importante referir que o id da div conteudo será essencial para indicar, através do jQuery, onde a lista deverá ser carregada.

O próximo passo será tratar dos ficheiros para abrir e fechar a conexão com a base de dados MySQL.

```
<?php

$conexao = mysqli_connect('localhost:8889', 'root', 'senha',
'escola');

// host, username, password, database

if (!$conexao) {

echo "<p style='color:red;'>Erro: Não foi possível conectar ao
MySQL.</p>" . PHP_EOL;

echo "<p>Erro: </p>" . mysqli_connect_errno() . "<br/><br/>" .
PHP_EOL ;

exit;

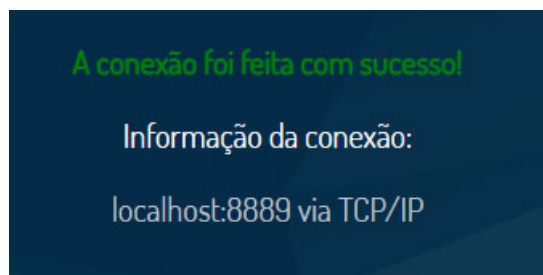
}

echo "<p style='color:green;'>A conexão foi feita com sucesso!</p>"
. PHP_EOL;
```

```
echo "<p>Informação da conexão: </p>" .  
  
mysqli_get_host_info($conexao) . "<br/><br/>" . PHP_EOL;  
  
?>
```

Desta vez, é utilizado o método `mysqli_connect()`. Este método permite conectar diretamente à base de dados selecionada, neste caso chamada de escola. A conexão é guardada numa variável de sessão chamada `$conexao`, para que seja possível ser utilizada quando necessário.

Se a conexão com a base de dados não estiver correta, uma mensagem de erro é escrita no ecrã a vermelho e o número do erro é mostrado com o método `mysqli_connect_errno()`. Quando a conexão for bem sucedida, é mostrada uma mensagem a verde, indicando o status da conexão e as suas informações, através do método `mysqli_get_host_info()`.



```
<?PHP  
  
mysqli_close($conexao);  
  
echo "<p style='color:brown;'>Conexão à base de dados  
encerrada.</p>";  
  
?>
```

O código de encerramento da conexão é muito mais simples. É utilizado o método `mysqli_close()`, seguido pela variável onde foi armazenada a conexão, para que esta seja fechada. Para terminar, mostra-se um texto a castanho que indica que a conexão foi encerrada.

Já estão três ficheiros analisados, falta apenas terminar o ficheiro JavaScript.

```
$(document).ready(function() {  
    $('#conteudo').fadeOut(); // ocultar a div  
    $('#botao').click(function(){ // quando é clicado  
        $('#conteudo').fadeIn(); // mostrar a div e mostrar a  
        consulta  
        $('#conteudo').load("main.php",{consulta:$("#texto").val()});  
        return false; // em caso de erro, não mostrar nada  
    });  
});
```

Aqui é criada uma função jQuery, utilizada para ocultar a div com o id conteudo ao carregar o documento, através do método fadeOut(). Quando o botão é pressionado, é executado o método fadeIn() do conteudo e é carregado o valor da caixa de texto na variável querys, que é passada para o ficheiro main.php. O URL ficará assim:

```
main.php?consulta="SELECT * FROM tbl_alunos"
```

Esta seria a consulta que seria carregada na div com o id conteudo. A consulta selecionada seria, então, iniciada a partir do ficheiro main.php e exibida na div correspondente no index.php.

ID	Nome	Apelidos	Idade
1	João	Santos Teste	22
2	Maria	Silva Teste	25
3	André	Rodrigues Teste	20
4	José	Fernandes Teste	21
5	Margarida	Cruz Teste	26
6	Pedro	Pereira Teste	22
7	João	Guilherme Teste	25
8	Ana	Reis Teste	20
9	Soraia	Dias Teste	23
10	Diogo	Pires Teste	24

Para terminar esta prática, falta apenas obter a lista do ficheiro main.php assim que a consulta for iniciada.

```
<table border='0' cellpadding='4' cellspacing='2'>
<tr>
<td><b>ID</b></td>
<td><b>Nome</b></td>
<td><b>Apelidos</b></td>
<td><b>Idade</b></td>
</tr>
<?php
if(!empty($_REQUEST['consulta'])){
include('abrirCnx.php');

$consulta = $_REQUEST['consulta']; // Esta linha armazena a
consulta

$resultado = mysqli_query($conexao, $consulta);
while($registro = mysqli_fetch_assoc($resultado)) {
echo "
```

```
<tr>

<td>".$registro['id']. "</td>

<td>".$registro['nome']. "</td>

<td>".$registro['apelidos']. "</td>

<td>".$registro['idade']. "</td>

</tr>

";

};

mysqli_free_result($resultado);

echo "</table>";

include('fecharCnx.php');

}else{

echo "<p>A consulta não devolveu nenhum resultado. <br/> Deve
selecionar uma consulta.</p>";

};

?>
```

Começa-se por estabelecer a tabela que irá conter os campos necessários para preencher com os dados devolvidos da consulta SQL da tabela `tbl_alunos`.

É utilizada diretamente a linguagem HTML, pois este código será inserido na div destinada para este fim dentro do `index.php`, onde já existe a estrutura HTML necessária para a tabela ser desenhada. O Ajax faz com que o resultado do `main.php` seja desenhado dentro da div destinada a tal uso no ficheiro `index.php`.

O que se segue é o código PHP que mais interessa. Existe uma condição que faz com que a variável `consulta` seja obtida por referência e avaliada com um `REQUEST`, para ver se vem vazia do `index.php`. Se estiver vazia, surge uma mensagem a informar que a consulta não obteve nenhum resultado, portanto, deverá selecionar uma das disponíveis. Se a variável tiver uma consulta, o ficheiro que abre a conexão é invocado e o valor é obtido por referência, armazenando-o numa variável chamada `$consulta`.



De seguida, o resultado da consulta é armazenado na variável `$resultado`, utilizando o método `mysqli_query()` em conjunto com a variável `$conexao` e com a variável `$querys`, cujos valores já estão à disposição.

Para preencher a tabela com os valores obtidos na consulta, é utilizado um loop `while` seguido pelo método `mysqli_fetch_assoc()` e pela variável `$resultado`.

Graças a isto, é obtido um array com todos os valores da consulta, que serão percorridos pelo loop `while`, inserindo os valores do `$registo` no seu campo correspondente dentro da tabela, como pode ser verificado no código.

Assim que o loop terminar, o resultado da variável é libertado para ser utilizado na próxima consulta, a tabela é fechada com o código HTML dentro de um `echo` PHP e é chamado o ficheiro que fecha a conexão com a base de dados.

Estas são todas as etapas que precisam de ser seguidas para que esta simples página web funcione corretamente.

