

# Apresentação de **REDES DE COMUNICAÇÃO** Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos

MÓDULO VII

Acesso a Bases de  
Dados via Web

---

Duração: 50 tempos

# Conteúdos (1)

- Conceitos sobre programação para Web
- Distinguir Processamento Server-Side de Client-Side
- Desenvolvimento de conteúdo dinâmico no servidor
- Desenvolvimento de conteúdo activo no cliente
- Linguagem de scripting – PHP
  - Operadores e variáveis
  - Instruções condicionais, ciclos

## Conteúdos (2)

- Linguagem de scripting – PHP (cont.)
  - Envio de formulários
  - Arrays
  - Cookies
- Bases de dados
  - Utilização de ferramentas de acesso a BDs
  - Utilização do SQL em consultas; inserção, modificação, procura, listagem e eliminação de dados
  - Formatação de saída de B.D.'s num *Browser*

# Critérios e tipo de avaliação

- Critérios:
  - 30% - parte sócio-afectiva;
  - 70% - parte cognitiva.
- Tipo de avaliação:
  - Teste sumativo prático.

# Notas introdutórias ao PHP (1)

- PHP é uma linguagem de programação de scripts que são interpretados no lado dos servidores.
- Em 1995 PHP surgiu com o significado de Personal Home Pages. Actualmente a sigla significa Hypertext Preprocessor

# Notas introdutórias ao PHP (2)

- PHP é uma tecnologia que teve grande sucesso porque é distribuída gratuitamente e tem uma excelente capacidade de articulação com:
  - os servidores web mais divulgados: o Apache (de distribuição gratuita) e o IIS (Internet Information Server), comercializado pela Microsoft.
  - sistemas de gestão de bases de dados de distribuição gratuita (por exemplo o MySQL) e outros de natureza comercial.

# Notas introdutórias ao PHP (3)

- PHP é uma linguagem interpretada, tal como o é o JavaScript.
- O JavaScript é interpretado no próprio browser que recebe a página HTML, enquanto o PHP é interpretado no servidor que envia a página.
- Podemos ver o código JavaScript no nosso browser mas quando recebemos uma página que teve como origem um script PHP só podemos ver o código HTML.

# Notas introdutórias ao PHP (4)

- Vamos utilizar o
  - XAMPP para Windows Version 1.6.8 Beta2
- O XAMPP inclui:
  - Servidor Apache;
  - Servidor MySQL;
  - E outros.

# Notas introdutórias ao PHP (5)

- Os sites quando utilizam PHP devem ser alojados na directoria do servidor:

`C:\xampp\htdocs` ou `D:\xampp\htdocs`

- Ou numa subdirectoria:

`C:\xampp\htdocs\exemplos` ou

`D:\xampp\htdocs\exemplos`

- Neste caso para acedermos a um exemplo escrevemos na barra de endereços do nosso browser o seguinte:

`http://localhost/exemplos/test1.php`

# Notas introdutórias ao PHP (6)

- Para administrar as bases de dados em MySQL:  
<http://localhost/phpmyadmin>
- Abre a página de informação do PHP

info.php

```
<?phpinfo();?>
```

# Notas introdutórias ao PHP (7)

- As tags(marcas) que identificam os scripts podem ter quatro formatos diferentes:
  - 1 – Estilo XML: `<?php ... ?>`
  - 2 – Estilo abreviado: `<? ... ?>`
  - 3 – Estilo ASP: `<% ... %>`
  - 4 – Estilo script: `<script language='php'> ... </script>`

# PHP intercalado com HTML

if1.php

```
<?php
    if ($x==0) {
?>
        <b> É verdade que x=0. </b>
<?php
    } else {
?>
        <b> É falso que x=0. </b>
<?php
    }
?>
```

# Chamada de ficheiro externo

require.php

```
<html>
<body>
  <?php
  echo "<b>Um script PHP </b> <br>";
  echo "chama outro script <br> <br>";
  $x=1;
  require ('if1.php');
  ?>
</body>
</html>
```

Chama o ficheiro externo  
if1.php

# Elementos básicos da linguagem PHP

- PHP é case sensitive, há diferença entre maiúsculas e minúsculas
- os nomes das variáveis começam sempre por \$
- as instruções terminam por ;
- um bloco de instruções é incluído dentro de chavetas:  
    {...}
- os comentários inserem-se entre /\* e \*/ ou numa linha após // ou #

# Operadores

## Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão inteira

## Operadores de incremento e decremento

Operador	Significado
++	incremento
--	decremento

## Operadores lógicos

Operador	Significado
! / not	Not – Negação lógica
&& / and	And – Conjunção lógica
/ or	Or – Disjunção lógica

**FIG. 3.103.** Os principais operadores usados em PHP são os mesmos das linguagens C, C++, Java e JavaScript. Os operadores de igualdade restrita (===) e desigualdade restrita (!==) existem em PHP e não em C/C++. A igualdade restrita (===) significa que as expressões comparadas devem ser iguais em valor e tipo de dados. A desigualdade restrita (!==) significa que as expressões devem ser diferentes em valor e/ou no tipo de dados.

## Operadores aritméticos de atribuição

Operador	Significado
+=	adição seguida de atribuição
-=	subtração seguida de atribuição
*=	multiplicação seguida de atribuição
/=	divisão seguida de atribuição
%=	resto da divisão seguida de atribuição

## Operadores relacionais ou comparativos

Operador	Significado
==	igual a
!=	diferente de
<	menor que
>	maior que
<=	menor ou igual que
>=	maior ou igual que
===	igualdade restrita
!==	desigualdade restrita

## Operador de junção de strings

Operador	Significado
.	junção de strings

# Variáveis

- Não existe declaração prévia de variáveis;
- Uma variável passa a existir quando o seu nome é escrito pela primeira vez, por exemplo numa instrução de atribuição: `$x=10;`
- O tipo de dados de uma variável é determinado pelo último valor que lhe for atribuído;
- Tipos de dados primitivos:
  - booleanos ou lógicos;
  - números inteiros;
  - números reais ou do tipo floating point;
  - strings ou cadeias de caracteres.

# Variáveis e constantes

- Outros tipos de dados são:
  - array;
  - object;
  - resource;
  - NULL.
- Em PHP podemos definir constantes com a função `define(constante, valor)`. Exemplos:
  - `define(TAXA,0.19);`
  - `define (N1,"Ana Dias");`

# Tipo de dados de cada variável

tipodados.php

```
<?php
$b= ($x==10);
echo gettype($b). '<br>';
$x=10;
echo gettype($x). '<br>';
$x=1.5;
echo gettype($x). '<br>';
$x="Ana Dias";
echo gettype($x). '<br>';

echo gettype($z). '<br>';
?>
```



gettype indica-nos o tipo de variável

# If (1) – Maior de dois números

lf2.php

```
<?php
$a=15;
$b=20;
    if ($a>$b) {
        echo "<b> $a é maior que $b </b>";
    } else {
        echo "<b> $b é maior ou igual a $a </b>";
    }
?>
```

Condição

## If (2) - Exercício ?

If3.php

**Crie uma função que devolva o maior de 3 números inteiros.**

# Switch (1)

```
<html><body>
<?php
$x=2;
switch ($x) {
case 1:
    echo "Número um";
    break;
case 2:
    echo "Número dois";
    break;
case 3:
    echo "Número três";
    break;
default:
    echo "Números fora do intervalo
        entre um e três";
}
?>
</body></html>
```

\$x=2

Switch1.php

# Switch (2) - Exercício ?

Switch2.php

- **Apresentar numa página o resultado do totobola com 13 jogos de futebol:**
  - 1º jogo (1 ou x ou 2).
  - 2º jogo (1 ou x ou 2).
  - 
  - 
  - 13º jogo (1 ou x ou 2).

# O ciclo For (1)

ForCubos.php

```
<html> <body>
<table border="1">
<tr> <td><b>Números</b></td>
      <td><b>Cubos</b></td>
</tr>
<?php
for ($i=1;$i<=5;$i++) {
    $q=$i*$i*$i;
    echo "<tr> <td> $i </td>";
    echo "<td> $q </td></tr>";
}
?>
</table>
</body></html>
```

Números	Cubos
1	1
2	8
3	27
4	64
5	125

Cálculo do **cubo** do número **i**

## O ciclo For (2) - Exercícios ?

For1.php

- **Escrever numa página os números de 1 a 22 de 3 em 3.**

For2.php

- **Escrever numa página os números por ordem decrescente de 20 até 10.**

# O ciclo For (3) - Exercício ?

For3.php

- Apresentar numa página a tabuada do 5.

# O ciclo While (1)

## Este exercício tem um erro ?

While1.php

```
<html>
<body>

<?php
echo "Tabuada do 7 <br>";
$i=1;
while($i<=10)
{
    echo "7 X " . $i . " = " . 7 * $i . "<br>";
    $i++;
}
?>
</body>
</html>
```

Cálculo da tabuada

## O ciclo While (2) - Exercícios ?

While2.php

**Calcule 20 múltiplos do número 5 e apresente-os numa página.**

While3.php

**Calcule a média dos números do 1 ao 50 e apresente-a numa página.**

# Envio de dados de uma página para outra – Métodos GET e POST (1)

- **Método GET:**
  - Os valores e respetivos indicadores são transmitidos pela URL;
  - **Exemplo:**
    - <http://www.meusite.com/enviodados.php?id=32&descricao=CPU>
  - O nº de bytes máximo enviado é 1024;
  - Os dados são visíveis na URL – é preciso ter cuidado com a informação confidencial.

## Envio de dados de uma página para outra – Métodos GET e POST (2)

- **Método POST:**
  - Os dados não são transmitidos pela URL e portanto não estão visíveis ao utilizador;
  - Não existe limite para a quantidade de dados;
  - É necessário utilizar um formulário para enviar os dados;
  - No formulário na propriedade action colocamos a página para onde enviamos os dados;
  - Quando premimos o botão submit todos os dados do formulário são enviados.

# Envio de dados de uma página para outra – Métodos GET e POST (3)

- **Como utilizar os dados enviados:**
  - O PHP armazena as variáveis vindas por GET e por POST em vetores associativos (vetores cujos índices são strings).
  - **No método GET:**
    - `$meuid = $_GET["id"]` e `$minhadescricao = $_GET["descricao"]`;
  - **No método POST:**
    - `$meunome = $_POST["nome"]` e `$minhaidade = $_POST["idade"]`;
    - Os índices do vetor serão os nomes dados aos inputs no formulário.

# Envio dos dados de um formulário(1)

EnvioForm2.htm

```
<html>
<head><title> Envio de dados </title></head>
<body> <h3> Envio de dados </h3>
<form method ="post" action="process2.php">
<table border=0>
  <tr> <td> Título de um livro: </td>
    <td> <input type="text" name="title">
  <tr> <td> Endereço Web: </td>
    <td> <input type="text" name="pweb">
  <tr> <td> Data de envio: </td>
    <td> <input type="text" name="data">
</table> <p>
<input type="submit" value="Enviar dados">
</form></body></html>
```

Ao ser premido o botão **submit** os dados do formulário são enviados para o ficheiro **process2.php**

## Envio dos dados de um formulário(2)

Process2.php

```
<?php
//variáveis que recebem os dados
$title=$_POST['title'];
$pweb=$_POST['pweb'];
$data=$_POST['data'];
?>
<html><body>
<h3> Dados recebidos </h3>
<?php
echo "Título do livro: $title <br>";
echo "Endereço Web: " . $pweb. "<br>";
echo "Data de envio: " . $data. "<br>";
echo "Data actual: ".date("y-m-d");
?>
</body></html>
```

Atribuição das variáveis enviadas pelo formulário a partir do array `$_POST`

Escrita na página das variáveis e da data actual

# Envio dos dados de um formulário com teste dos campos enviados (1)

EnvioForm.htm

```
<html>
<head> <title> Formulário com dados </title>
<script language="javascript">
  function avaliar(frm) {
    if (frm.prod.value==" " || frm.quant.value==""){
      window.alert("É necessário preencher os campos.");
      return(false); }
    else if (isNaN(frm.quant.value)){
      window.alert("A quantidade tem que ser um número.");
      return(false); }
    else return (true);
  }
</script> </head>
```

Função que testa se campos estão bem preenchidos

Se os campos não estão bem preenchidos retorna **false** senão retorna **true**

## Envio dos dados de um formulário com teste dos campos enviados (2)

EnvioForm.htm

```
<body> <h2> Dados a enviar </h2>
<form name="formd" method="post" action="process.php"
onsubmit="return avaliar(formd);">
  <table border=0>
    <tr> <td> Nome do produto:
      <td> <input type="text" name="prod">
    <tr> <td> Quantidade desejada:
      <td> <input type="text" name="quant">
  </table>
  <input type="reset" value="Apagar dados">
  <input type="submit" value="Enviar dados">
</form> </body> </html>
```

Se função avaliar retornar true abre ficheiro process.php

Caixas de texto do formulário

Botões do formulário

# Envio dos dados de um formulário com teste dos campos enviados (3)

process.php

```
<?php
//variáveis para receber os dados
$produto= $_POST['prod'];
$quantid= $_POST['quant'];
?>

<html> <head> <title> Dados Recebidos </title> </head>
<body>
<h3> Dados Recebidos </h3>

<?php
echo 'Artigo: '.$produto.<br>';
echo 'Quantidade: '.$quantid.<br>';
?>

</body> </html>
```

Atribuição das variáveis enviadas pelo formulário

Escrita na página das variáveis

# Envio de Dados - Exercício ?

EnvioForm3.htm

**Crie um formulário com Utilizador e Password.**

**Valide utilizando javascript o nome de utilizador que não pode estar vazio. Valide a password que não pode estar vazia e que tem que ter mais de 4 caracteres.**

**Envie os dados para o ficheiro process3.php e neste ficheiro apresente-os na página.**

# Caracteres especiais em PHP (1)

Caracteres	Significado
<code>\"</code>	carácter " – aspas duplas
<code>\\$</code>	carácter \$ – cifrão
<code>\\</code>	carácter \ – barra invertida
<code>\n</code>	new line – nova linha
<code>\t</code>	tabulação horizontal

# Caracteres especiais em PHP (2)

caracteres1.php

```
<html><body>
<h3>Manipular Strings </h3>
<?php
echo "http://www.oi.pt/"; echo "<br>";
echo 'http://www.oi.pt/'; echo "<br>";

// echo 'Developer's Guide'; //Não funciona
// echo "C:\Programas\"; //Não funciona

$t1= "Developer's Guide";
echo "Titulo do livro: $t1 <br>";
echo 'Titulo do livro: '. $t1. '<br>';
echo "c:\\Programas\\<br>";
echo "\\Manual de PHP\\<br>";
echo "Valor em dolars:\\$25.00<br>";
?>
```

Echo's não funcionam

Exemplos que funcionam

# Caracteres especiais em PHP (3)

```
<html> <body>
<table border="1">
<tr><td><b>Números </td>
      <td><b>Strings </td>
```

caracteres2.php

```
<?php
$st="It's PHP";
for ($i=2; $i<6; $i++)
{
echo "<tr><td align='center'> $i </td>";
echo      "<td> <font color='0000FF' size='\$i'> $st </font> </td> </tr>";
}
?>
</table>
</body></html>
```

Números	Strings
2	It's PHP
3	It's PHP
4	It's PHP
5	It's PHP

Aspas são precedidas pelo caracter \

# Arrays indexados

array1.php

```
<?php
    $a= array ('Ana', 'Eva', 'Rui');
    for($i=0; $i<3; $i++)
        echo $a[$i]. '<br>';
    echo '<br>';
    $a[]=3.14;
    foreach ($a as $valor)
        echo $valor. '<br>';
?>
```

Índice do array

# Arrays associativos (1)

array2.php

```
<?php
$N= array ('Ana'=>10, 'Eva'=>12, 'Rui'=>14);
echo $N['Ana']. '<br>';
echo '<br>';
$N['Ana'] = 11;

foreach ($N as $valor)
    echo $valor. '<br>';

$N['Leo']=15;
echo '<br>';

foreach ($N as $chave => $valor)
    echo $chave. ' = ' . $valor. '<br>';
?>
```

Valor

Chave

# Arrays associativos (2) EnvioForm

array3.php

```
<html>
<body>
<h3> Dados recebidos </h3>
<?php
    foreach ($_POST as $chave => $valor)
        echo $chave. ' = ' . $valor. '<br>';
?>
</body>
</html>
```

# Arrays associativos (3) EnvioForm

Envioform4.htm

```
<html>
<head><title> Envio de dados </title></head>
<body> <h3> Envio de dados </h3>
<form method ="post" action="array3.php">
<table border=0>
  <tr> <td> Título de um livro: </td>
    <td> <input type="text" name="title">
  <tr> <td> Endereço Web: </td>
    <td> <input type="text" name="pweb">
  <tr> <td> Data de envio: </td>
    <td> <input type="text" name="data">
</table> <p>
<input type="submit" value="Enviar dados">
</form></body></html>
```

- Copiar exercício de envioform2 (alterando action=...)

# Arrays – Exercícios ?

foreach1.php

**Guarde num array os números de 1 a 4 e utilizando o ciclo foreach apresente esses números na página.**

foreach2.php

**Guarde num array os seguintes dados utilizando a instrução seguinte:**

```
$lista = array("um"=>1, "dois"=>2, "tres"=>3, "quatro"=>4);
```

**e utilizando o ciclo foreach apresente os dados numa página da seguinte forma:**  
`$lista[um] => 1, $lista[dois] => 2 ...etc`

# Cookies(1)

A função `setcookie()` é usada para criar uma cookie.

A função `setcookie()` deve aparecer antes da tag `<html>`.

## **Sintaxe:**

```
setcookie(nome, valor, expira, caminho, domínio);
```

## Cookies(2)

### Cookies1.php

```
<?php
// Cria cookie que expira passado uma hora
setcookie("utilizador", "Alexandre", time()+3600);
?>
```

### Cookies2.php

```
<?php
// Cria cookie que expira passado um mês
$expira=time()+60*60*24*30;
setcookie("utilizador", "Alexandre", $expira);
?>
```

# Cookies(3)

Cookies3.php

```
<?php
// Retorna o valor da cookie e mostra-a na página
echo $_COOKIE["utilizador"];

// Mostra todas as cookies na página
print_r($_COOKIE);
?>
```

# Cookies(4)

Cookies4.php

```
<html><body>
<?php
// Testa se cookie existe e está activa
if (isset($_COOKIE["utilizador"]))
    echo "Bem-vindo " . $_COOKIE["utilizador"] . "!<br />";
else
    echo " Bem-vindo convidado!<br />";
?>
</body></html>
```

## Cookies(5)

Cookies5.php

```
<?php  
/* Apagar uma cookie colocando uma data de expiração passada */  
setcookie("utilizador", "", time()-3600);  
?>
```

# Abre página (1)

Abrepagina.php

```
<?php
$acesso=true;
if ($acesso==true) {
    header("Location: acessomeusite.php");}
else {
    header("Location: acessonegadomeusite.php");}
?>
```

## Abre página (2)

acessomeusite.php

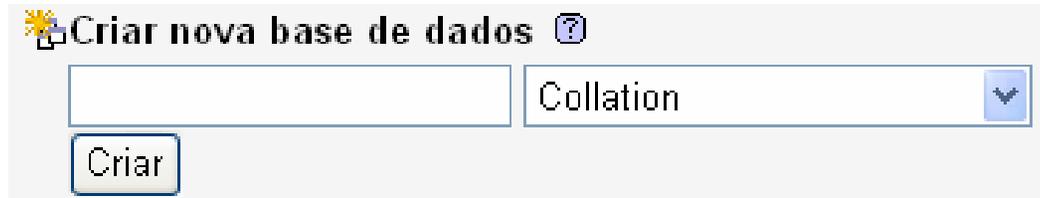
```
<?php  
echo "Sucesso! Você teve acesso à página!";  
?>
```

acessonegadomeusite.php

```
<?php  
echo "Acesso recusado! Você não teve acesso à página!";  
?>
```

# Vendas - Acesso a Base de Dados (1) – criação da BD em MySQL

Criar a base de dados **vendas** em:  
<http://localhost/phpmyadmin/>



The screenshot shows the 'Criar nova base de dados' (Create new database) form in phpMyAdmin. It features a text input field for the database name, a 'Collation' dropdown menu, and a 'Criar' (Create) button. The form is titled 'Criar nova base de dados' with a question mark icon.

# Vendas - Acesso a Base de Dados (2) – criação da BD em MySQL

Criar a tabela clientes com três campos em:

 Criar nova tabela na base de dados **vendas**

Nome:  Number of fields:

	Campo	Tipo	Collation	Atributos	Nulo	Defeito	Extra
<input type="checkbox"/>	<u>CodCli</u>	varchar(10)	latin1_swedish_ci		Não		
<input type="checkbox"/>	Nome	varchar(50)	latin1_swedish_ci		Não		
<input type="checkbox"/>	Morada	varchar(100)	latin1_swedish_ci		Não		

## Índices:

Nome do Índice	Tipo	Quantidade	Acções	Campo
PRIMARY	PRIMARY	2	 	CodCli

# Vendas - Acesso a Base de Dados

## (3) - Inserir

```
<html><head> <title> Entrar </title> </head>
<body> <h2> Entrar e Inserir Dados </h2>
<form action="inserir.php" method="post">
<table border="1">
<tr> <td> Código do cliente: </td>
<td> <input type="text" name="cod"> </td></tr>
<tr> <td> Nome do cliente: </td>
<td> <input type="text" name="nome"> </td></tr>
<tr> <td> Morada do cliente: </td>
<td> <input type="text" name="morada"> </td></tr>
<tr> <td colspan="2" align="center">
<input type="submit" value="Registar">
</td></tr>
</table>
</form>
<a href="listar.php"> Listar registos </a> <br>
<a href="procurar.htm"> Procurar registos </a> <br>
</body></html>
```

entrar.htm

Formulário  
dentro de uma  
tabela

Hiperligações

# Vendas - Acesso a Base de Dados

## (4) - Inserir

```
<html>
<head> <title> Inserir </title> </head>
<body> <h2> Inserir Registos </h2>
<?php
$cod=$_POST['cod'];
$nome=$_POST['nome'];
$morada=$_POST['morada'];
if (!$cod || !$nome || !$morada) {
    echo 'Campos em falta. Volte atrás e tente de novo.';
    exit;}
echo 'Dados recebidos: <br />';
echo 'Código: '.$cod.'<br />';
echo 'Nome: '.$nome.'<br />';
echo 'Morada: '.$morada.'<br />';
```

inserir.php

Testa se algum campo está por preencher

Imprime na página os dados recebidos do formulário

# Vendas - Acesso a Base de Dados

## (5) - Inserir

inserir.php

```
$ligax=mysqli_connect('localhost','root');
if (!$ligax) {
    echo '<p> Erro: Falha na ligação.';
    exit;}
mysqli_select_db($ligax, 'vendas');
$insere="insert into clientes values ('".$cod."','".$nome."','".$morada."')";
$result = mysqli_query($ligax, $insere);
if ($result==1) echo "<p> Dados inseridos.<br>";
else echo "<p> Dados não inseridos.<br>";
?>
<a href="entrar.htm"> Voltar à entrada </a> <br>
<a href="listar.php"> Listar registos </a> <br>
</body>
</html>
```

Inserir dados na base de dados **vendas**

hiperligações

# Vendas - Acesso a Base de Dados (6) - Inserir

- Exemplo de como fica a string \$insere:

**\$insere=**

**"insert into clientes values ('3','Pedro','R. Alegria')";**

# Vendas - Acesso a Base de Dados

## (7) - Listar

listar.php

```
<html>
<head> <title> Listar</title> </head>
<body> <h2> Listar Registos </h2>
<a href="entrar.htm"> Voltar à entrada </a> <br>

<?php
$ligax=mysqli_connect('localhost','root');
if (!$ligax) {
    echo '<p> Falha na ligação.';
    exit;}
mysqli_select_db($ligax, 'vendas');
$consulta="select * from clientes";
$result = mysqli_query($ligax, $consulta);
$nregistos=mysqli_num_rows($result);
echo 'Nº de registos encontrados: ' . $nregistos
?>
```

Ligação ao  
mysql

Seleccção da  
base de dados

Execução de  
consulta

Obtenção do  
número de  
registos

# Vendas - Acesso a Base de Dados

## (8) - Listar

listar.php

```
<table border="1">
<tr><td> Código: <td> Nome: <td> Morada: <td> Editar: </tr>
<?php
for ($i=0;$i<$nregistos;$i++)
{
    $registo=mysqli_fetch_assoc($result);
    echo '<tr>';
    echo '<td>'.$registo['CodCli'].'</td>';
    echo '<td>'.$registo['Nome'].'</td>';
    echo '<td>'.$registo['Morada'].'</td>';
    echo '<td><a href="editar.php?CodCli='.$registo['CodCli'].'"> Editar
    </a></td>';
    echo '</tr>';
    echo '</p>';
}
?>
</table></body></html>
```

Devolve um registo

Imprime os campos do registo em colunas de uma tabela na página e uma hiperligação para editar

# Vendas - Acesso a Base de Dados (9) - Procurar

procurar.htm

```
<html>
<head> <title> Procurar </title> </head>
<body> <h2> Escreva o nome a procurar </h2>
  <form action="mostrar.php" method="post">
  <table border="1">
  <tr> <td> Nome a procurar: </td>
    <td> <input type="text" name="nome"> </td>
  </tr>
  <tr> <td colspan="2" align="center">
    <input type="submit" value="Procurar Registos">
  </td>
  </tr>
  </table>
  </form>
  <a href="entrar.htm"> Voltar à entrada </a> <br>
  <a href="listar.php"> Listar registos </a> <br>
</body>
</html>
```

Formulário de procura de nomes

hiperligações

# Vendas - Acesso a Base de Dados (10) - Procurar

mostrar.php

```
<html>
<head> <title> Mostrar</title> </head>
<body> <h2> Mostrar nome procurado </h2>

<?php
$nomeproc=$_POST['nome'];
if (!$nomeproc){
    echo 'Volte atrás e escreva o nome.';
    exit;
}
echo '<p>Nome procurado: '.$nomeproc.'<p>';
$ligax=mysqli_connect('localhost','root');
if (!$ligax) {
    echo '<p> Falha na ligação.';
    exit;}
mysqli_select_db($ligax, 'vendas');
$procura="select * from clientes where nome like '%" . $nomeproc . "%'";
$result = mysqli_query($ligax, $procura);
$nregistos=mysqli_num_rows($result);
echo 'Nº de registos encontrados: ' . $nregistos
?>
```

Se **\$nomeproc** vazio manda escrever novamente e sai da rotina.

Procura clientes que tenham **\$nomeproc** no nome

# Vendas - Acesso a Base de Dados (11) - Procurar

```
<table border="1">
<tr><td>Código: <td> Nome: <td> Morada: </tr>
<?php
for ($i=0;$i<$nregistos;$i++)
{
    $registo=mysqli_fetch_assoc($result);
    echo '<tr>';
    echo '<td>'.$registo['CodCli'].'</td>';
    echo '<td>'.$registo['Nome'].'</td>';
    echo '<td>'.$registo['Morada'].'</td>';
    echo '</tr>';
    echo '</p>';
}
?>
</table>
<form action="remover.php" method="post">
<p>Para apagar um registo, escreva aqui o nome completo.</p>
<input type="text" name="nome"> </td>
<input type="submit" value="Apagar registo">
</form>
<a href="listar.php"> Listar registos </a> <br>
</body></html>
```

mostrar.php

Mostra registos encontrados numa tabela.

Formulário para apagar registo

# Vendas - Acesso a Base de Dados (12) - Eliminar

```
<html>
<head> <title> Remover</title> </head>
<body> <h2> Remover Registo </h2>

<?php
$nomerem=$_POST['nome'];
if (!$nomerem){
    echo 'Volte atrás e escreva o nome.';
    exit;
}

echo '<p>Nome a remover: '.$nomerem.'<p>';
$ligax=mysqli_connect('localhost','root');
if (!$ligax) {
    echo '<p> Falha na ligação.';
    exit;}

mysqli_select_db($ligax, 'vendas');
$consulta="select * from clientes";
$result = mysqli_query($ligax, $consulta);
$registos_antes =mysqli_num_rows($result);
```

remove.php

Se **\$nomerem** vazio  
manda escrever  
novamente e sai da  
rotina..

Obtém número de  
clientes antes de  
apagar registos

# Vendas - Acesso a Base de Dados (13) - Eliminar

remover.php

```
$remove="delete from clientes where nome='".$nnumerem.'";  
$result = mysqli_query($ligax, $remove);  
if ($result == 0) echo "<p> Não removido<br>";  
echo 'Resultado:'. $result;  
$consulta="select * from clientes";  
$result = mysqli_query($ligax, $consulta);  
$nregistos_depois=mysqli_num_rows($result);  
  
$nregistos_removidos=$nregistos_antes-$nregistos_depois;  
echo '<p>Nº de registos removidos: ' . $nregistos_removidos;  
?>  
  
<a href="listar.php"> <p>Listar registos</p> </a> <br>  
</body>  
</html>
```

Remove registos com o nome igual a **\$nnumerem**

Obtém número de clientes depois de apagar registos

# Vendas - Acesso a Base de Dados (14) - Editar

editar.php

```
<?php
$CodCli=$_GET['CodCli'];
$ligax=mysqli_connect('localhost','root');
if (!$ligax) {
    echo '<p> Falha na ligação.';
    exit;}
mysqli_select_db($ligax, 'vendas');
$consulta="select * from clientes where CodCli='".$CodCli."'";
$result = mysqli_query($ligax, $consulta);
$registo=mysqli_fetch_assoc($result);
?>
```

Obtém registo  
com o código  
igual a **\$CodCli**

# Vendas - Acesso a Base de Dados (15) - Editar

editar.php

Coloca na caixa de texto o **Nome** encontrado na base de dados

```
<form action="actualizar.php?cod=?php echo $registo['CodCli'];?" method=post>
<table border="1">
  <tr> <td> Código do cliente: </td>
    <td>?php echo $registo['CodCli'];?> </td>
  </tr>
  <tr><td> Nome do cliente: </td>
    <td><input type="text" name="nome" value="<?php echo $registo['Nome']; ?>"></td>
  </tr>
  <tr><td>Morada do cliente: </td>
    <td><input type="text" name="morada" value="<?php echo $registo['Morada']; ?>"></td>
  </tr>
  <tr> <td colspan="2" align="center">
    <input type="submit" value="Actualizar" >
  </td></tr>
</table></form></body></html>
```

# Vendas - Acesso a Base de Dados (16) - Editar

actualizar.php

```
<html>
<head> <title> Inserir </title> </head>
<body> <h2> Inserir Registos </h2>
<?php
$cod=$_GET['cod'];
$nome=$_POST['nome'];
$morada=$_POST['morada'];
if (!$cod || !$nome || !$morada) {
    echo 'Campos em falta. Volte atrás e tente de novo.';
    exit;}
echo 'Dados recebidos: <br />';
echo 'Código: '.$cod.'<br />';
echo 'Nome: '.$nome.'<br />';
echo 'Morada: '.$morada.'<br />';
```

# Vendas - Acesso a Base de Dados (17) - Editar

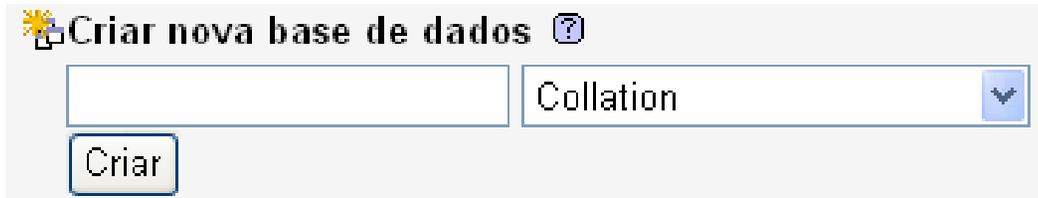
actualizar.php

```
$ligax=mysqli_connect('localhost','root');  
if (!$ligax) {  
    echo '<p> Erro: Falha na ligação.';  
    exit;}  
mysqli_select_db($ligax, 'vendas');  
$actualiza="update clientes set Nome="".$nome."" , Morada="".$morada.""  
where CodCli="".$cod.""";  
$result = mysqli_query($ligax, $actualiza);  
if ($result==1) echo "<p> Dados actualizados.<br>";  
else echo "<p> Dados não actualizados.<br>";  
?>  
<a href="entrar.htm"> Voltar à entrada </a> <br>  
<a href="listar.php"> Listar registos </a> <br>  
</body></html>
```

Actualiza registo  
com o código igual  
a \$cod

# Login (1) – criação da Base de Dados em MySQL

Criar a base de dados **login** em:  
<http://localhost/phpmyadmin/>



The screenshot shows the 'Criar nova base de dados' (Create new database) form in phpMyAdmin. It features a text input field for the database name, a 'Collation' dropdown menu, and a 'Criar' (Create) button.

# Login (2) – criação da Base de Dados em MySQL

Criar a tabela **acesso** com dois campos em:

 Criar nova tabela na base de dados **login**

Nome:  Number of fields:

	Campo	Tipo	Collation	Atributos	Nulo	Defeito	Extra
<input type="checkbox"/>	<u>utilizador</u>	varchar(20)	latin1_swedish_ci		Não		
<input type="checkbox"/>	password	varchar(8)	latin1_swedish_ci		Não		

## Índices:

Nome do Índice	Tipo	Quantidade	Acções	Campo
PRIMARY	PRIMARY	2	 	utilizador

# Login (3) - Exercício de Acesso

Login.htm

**Crie um formulário com Utilizador e Password.**

**Valide utilizando javascript o nome de utilizador que não pode estar vazio. Valide a password que não pode estar vazia e que tem que ter mais de 4 caracteres.**

**(Copiar de envioform3.htm)**

Redes de Comunicação - Pedro Matos Gonçalves

## Login (4) - Exercício de Acesso

Login.php

**Envie os dados para o ficheiro login.php e neste ficheiro procure o utilizador e a password na base de dados login, na tabela acesso.**

**Se encontrar o utilizador e a password na base de dados abre a página:  
[www.google.pt](http://www.google.pt).**

**Se não encontrar o utilizador e a password volta à página login.htm.**

## Login (5) - Inserir

introduzir.htm

**Crie um formulário com as caixas de texto **utilizador e password****

**Valide utilizando javascript o nome de utilizador que não pode estar vazio. Valide a password que não pode estar vazia e que tem que ter mais de 4 caracteres.**

**E envie os dados para o ficheiro **inserir.php**.**

## Login (6) – Inserir e Listar

inserir.php

Receba os dados do ficheiro **introduzir.htm** e guarde os dados na base de dados.

listar.php

Coloque uma hiperligação no ficheiro **introduzir.htm** para este ficheiro e apresente, numa tabela, os nomes dos vários utilizadores constantes da base de dados.

## Login (7) - Exercício de Elimina Utilizador

apagar.htm

Coloque uma hiperligação no ficheiro **introduzir.htm** para este ficheiro e crie um formulário com uma caixa de texto **utilizador**.

apaga.php

Envie os dados para o ficheiro **apaga.php** e neste ficheiro:

- **Elimine** o utilizador que foi introduzido no formulário da base de dados;
- **Devolva o número de utilizadores restantes.**

## Login (8) - Editar

editar.php

Coloque uma hiperligação junto a cada registo de utilizador no ficheiro **listar.php** que envie o nome do utilizador para este ficheiro. Neste ficheiro permita a edição da password.

Envie os dados alterados para o ficheiro **actualizar.php** que vai guardar os dados alterados na base de dados.

## Login (9) - Editar

actualizar.php

Receba os dados do utilizador e password alterados no ficheiro **editar.php** e guarde as alterações da password no registo correspondente da base de dados.

## Login (10) – Eliminar (alterações)

Coloque uma hiperligação junto a cada registo de utilizador no ficheiro **listar.php** que envie o nome do utilizador a eliminar para o ficheiro **apaga.php**. Faça as correspondentes alterações aos ficheiros **apaga.php** e **apagar.htm**.

# Bibliografia

- SÉRGIO, Ricardo. – **Redes de Comunicação 2.**  
Porto: Areal Editores, 2010. ISBN 9789896471866