Bases de Dados e Páginas da Rede — Como?

Apache	servidor de páginas capaz de processar o HTML e o PHP (entre outras).
HTML (Hypertext Meta-Language)	linguagem para a construção de páginas da rede.
PHP	linguagem de programação genérica embutida em ficheiros HTML e capaz de comunicar com SGBDs.
MySQL	um SGBD capaz de comunicar com várias linguagens de programação (PHP, ...).

- Solução possível em: Linux; MacOS; MS-Windows; ...
- Solução servidor, isto é, não necessita de nenhuma funcionalidade especial nos navegadores dos "clientes".

Apache

O programa *Apache* é o servidor de páginas "Web".

- Responde aos pedidos dos clientes.
- Trata das questões de segurança inerentes a um serviço "público".
- Tem como clientes os Navegadores com os quais comunica.
- Interpreta (se configurado para tal) a linguagem PHP, assim como outras linguagens "externas".

Apache — Convenções

- Num sistema Linux/Unix cada utilizador pode criar uma página. Por omissão o directório a usar deve-se designar por public_html:
  - `~nomeUtilizador/public_html/index.html` ← Sistema de ficheiros Unix.
  - `http://nomeDoDominio/~nomeUtilizador/index.html` ← URL
- Por omissão o ficheiro inicial designa-se por index.html
  - `~nomeUtilizador/public_html/index.html` ← Sistema de ficheiros Unix.
  - `http://nomeDoDominio/~nomeUtilizador/` ← URL
- Para que num ficheiro contendo código PHP este seja interpretado de forma correcta, a extensão do ficheiro tem de ser php.

HTML — HyperText Markup Language

Uma colecção de "meta-marcas" ("markup tags") usadas para definir as várias componentes de um texto da rede.

- "HTML Tutorial" — `http://www.w3schools.com/html/default.asp`.
- "Getting started with HTML" Dave Raggett — `http://www.w3.org/MarkUp/Guide/`.
- Formulários I — `http://www.w3.org/TR/html4/interact/forms.html`.
- Formulários II — `http://www.mat.uc.pt/~pedro/manuais/overview.html`.

HTML, (muito) Breve Introdução

O HTML (HyperText Markup Language) é uma meta-linguagem com capacidade de referenciação. Isto é, é uma linguagem que descreve uma outra linguagem (a linguagem dos textos da rede) e que é capaz de incluir, num dado texto, referências a outros textos.

- Cabeçalho ("head") - contém dados gerais sobre o texto que se segue.
- Corpo ("body") - contém a descrição do texto através de uma série de comandos que lidam com as várias estruturas de um texto.
  - O espaçamento entre palavras é definido de forma dinâmica. Um espaço no texto fonte, vale tanto como vinte espaços, o resultado final é o mesmo.
  - Não tem a noção de linha, ou melhor de quebra de linha. As linhas podem ter uma largura variável.
  - Não tem a noção de página de texto. O comprimento dos textos é, em teoria, infinito.

HTML — Um exemplo

```
<html>
<head>
<TITLE>Um exemplo simples de um texto em HTML</TITLE>
</head>

<body>
<H1>HTML é simples de Aprender</H1>
<P>
Exemplo de um parágrafo em HTML.
Como podem ver tem a marca ‘‘tag’’, ‘‘P’’!
</P>
<P>
Todos (quase todos) os comandos HTML tem uma marca inicial e uma marca

final, sendo que essa só difere da inicial pelo prefixo ‘‘/’’.
</P>
</body>
</html>
```

HTML, (muito) Breve Introdução

- Como numa linguagem de programação "normal" o uso da indentação é opcional.
- Para textos normais (sem PHP) a utilização de editores especializados permite "esquecer" os pormenores da linguagem.
  `(x)emacs, Quanta plus, ....`
- O HTML é uma linguagem sem a noção de estado ("stateless"), isto é não permite, entre outras, a comunicação entre textos através de parâmetros.
- A comunicação entre texto é feita através do mecanismos de formulários

HTML, Marcas

HTML	Marca o início (e o fim) do texto HTML. Informa o navegador que o texto contém código HTML (a extensão `.html` tem o mesmo efeito.
HEAD	Define o cabeçalho - informação genérica sobre o documento.
TITLE	Contém o título "exterior" do documento, isto é identifica o texto no contexto global (é uma das informações que o "google" procura).
BODY	Define o corpo do documento, isto é, o documento que vai ser visível através do navegador. É aqui que se pode definir o texto através de um conjunto de marcas para as várias estruturas em que um texto é constituído.

HTML, Marcas (continuação)

Headings Cabeçalhos, o HTML tem seis níveis de cabeçalhos: <Hx>, com $x = 1, 2, \ldots, 6$.

Parágrafos Dado que o HTML não tem a noção de linha (nem de espaçamento fixo), sempre que se quer começar um novo parágrafo é necessário usar a marca <P>.

Listas O HTML suporta: listas não numeradas; listas numeradas; e listas de definições.

- **Listas Não Numeradas:** Marca de início/fim de lista não numerada <UL>; marca de início/fim de um elemento ("item") da lista <LI>.
- **Listas Numeradas:** Marca de início/fim de lista não numerada <OL>; marca de início/fim de um elemento ("item") da lista <LI>.
- **Listas de Definições:** Marca de início/fim de lista não numerada <DL>; marca de início/fim do título para o elemento a definir <DT>; marca de início/fim da definição <DD>;

HTML, Marcas: Tabelas

A marca <TABLE> delimita o espaço de construção de uma tabela, isto é, um texto formatado em linhas, em que cada linha está dividida em colunas. Dentro desse espaço temos acesso às seguintes marcas:

Linhas cada linha da tabela é definida através da marca <TR>.

Cabeçalhos (das colunas) podemos definir uma linha de cabeçalhos (em geral a linha de topo) usando a marca <TH>, uma por coluna.

Colunas dentro de cada linha podemos definir as colunas através da marca <TD>.

HTML, Marcas (continuação)

Texto pre-formatado <PRE> secção de texto em que os espaços e as mudanças de linha são significativas e em que o tipo de letra usado é de largura fixa. Óptimo para incorporar descrições de programas escritos numa dada linguagem de programação.

Mudanças de linha pode-se forçar a mudança de linha utilizando a marca
. Note-se que neste caso não se está perante um par de marcas, só existe a marca descrita.

Linhas Horizontais pode-se introduzir uma linha horizontal (a separar duas secções de texto), através da marca <HR/>. Esta é também uma marca isolada.

HTML, Atributos

Muitas das marcas no HTML podem ter a sua acção modificada através da utilização de atributos.

Por exemplo:

- <P ALIGN="center"> permite definir um parágrafo centrado.

- <TABLE border="1">, define uma tabela com uma bordadura de tipo 1 (espessura mínima).

HTML, Formulários

Sendo que o HTML é uma linguagem sem a noção de estado... como é que é possível comunicar entre páginas, e entre uma página e um programa?

A resposta é dada pelas "Forms".

```
<FORM ACTION="url" METHOD="POST"> ... </FORM>
```

Os formulários em HTML podem ser de dois tipos (`Method=...`):

- GET — os dados são passados através do URL.
- POST — os dados são passados através da construção de uma página própria para o efeito.

Temos então a marca `<Form>` a marcar o início/fim de um formulário. Neste caso a especificação dos atributos é essencial.

HTML, Formulários

Atributos da marca "Form"

- ACTION — define o "URL" do programa que é suposto processar a informação recolhida pelo formulário.
- METHOD — "GET", ou "POST", define a forma como os valores são passados.
  Qual dos métodos a utilizar vai depender da aplicação pretendida, em geral o primeiro reserva-se para quando a quantidade de informação a passar é pequena e/ou pública, ficando o outro método reservado para os outros casos.

No âmbito dos formulários temos acesso às seguintes marcas:

- INPUT — um elemento simples (uma só linha de texto).
- SELECT — a escolha de um elemento entre várias opções.
- TEXTAREA — um elemento multi-linhas.

HTML, Formulários, INPUT

```
<input type='text' name='nomeVar' size='80' maxlength='80'>
```

Os atributos desta marca são:

- TYPE — `text`; `password`; `checkbox`; `radio`; `submit`; `reset`.
- NAME — define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.
- VALUE — pode ser usado para se definir o valor do elemento.
- CHECKED — especifica se um elemento do tipo `checkbox` ou `radio` está activado, por omissão.
- SIZE — tamanho físico (em caracteres) do campo de entrada tal como ele vai ser formatado.
- MAXLENGTH — tamanho máximo (em caracteres) do elemento de entrada.

HTML, Formulários, SELECT

```
<SELECT NAME="lista-de-opções">
<OPTION> Primeira opção</OPTION>
...
<OPTION> opção N</OPTION>
</SELECT>
```

Os atributos desta marca são:

- NAME — define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.
- SIZE — número de elementos visíveis da lista de opções, para uma dada posição da barra de corrimento.
- MULTIPLE — se presente pode-se escolher vários elementos

Os atributos da marca "OPTION" são:

- SELECTED — especifica que a opção em causa é seleccionada por omissão.

HTML, Formulários, TEXTAREA

```
<TEXTAREA NAME="nomeVar" ROWS=4 COLS=40>
Este é o conteúdo por omissão
</TEXTAREA>
```

Os atributos desta marca são:

- NAME define (a exemplo de uma variável) um identificador para o elemento que se está a especificar.
- ROWS o número de linhas da janela de entrada.
- COLS o número de colunas (caracteres) da janela de entrada.

PHP

Linguagem de programação genérica embutida em ficheiros HTML.

- Um ficheiro de extensão `php` pode conter: HTML; marcas PHP; código PHP (entre um par de marcas PHP);
- O Apache (ou outro programa que faça a gestão do serviço) tem de estar configurado para interpretar o código PHP.
- Os ficheiros que contêm código PHP têm de ter a extensão `php`

Um exemplo de programa (`exemplo.php`).

```
<?php
echo "Olá mundo";
?>
```

O PHP é uma das mais populares linguagens para construção de páginas dinâmicas numa perspectiva de uma solução servidor (todo o processamento é feito no servidor).
O PHP providência uma muito fácil ligação a bases de dados.

PHP — Bibliografia

- Manual do PHP ("on-line")
  `http://pt.php.net/manual/pt_BR/index.php`
- Luke Welling & Laura Thomson, *PHP and MySQL Web Development*, 4rd Edition, Sams Publishing, 2008.
- Andy Harris, *PHP 6/MySQL Programming*, Premier Press, 2008.

PHP — Introdução

A sintaxe, do PHP é semelhante à linguagem C (assim como as potencialidades).

- **Variáveis**
  - sem declaração explícita, uma dada instrução de atribuição cria a variável, de um dado tipo, com um dado valor;
  - o valor, assim como o tipo de uma variável pode ser mudado por uma outra atribuição posterior.
  - identificam-se pelo carácter $ inicial, por exemplo: `$nome`
- **Tipos**
  - Inteiros;
  - Reais;
  - Sequências de caracteres;
  - Booleanas;
  - Tabelas;

A linguagem é "case sensitive", ou seja maiúsculas e minúsculas são consideradas diferentes.

PHP — Âmbito das Variáveis

O PHP não tem mecanismos (explícitos) de comunicação entre ficheiros, como tal o âmbito de uma variável está sempre ligado ao ficheiro aonde foi definida.

Entre ficheiros (sessões Apache):

- Variáveis **Super Globais** – sempre visíveis.

Num dado ficheiro:

- **Constantes** – sempre visíveis.
- Variáveis **Globais** – em todo o ficheiro excepto dentro das funções.
- Variáveis declaradas dentro de uma função como estáticas, referem-se às variáveis com o mesmo nome, mas mantêm o valor entre chamadas sucessivas.
- Variáveis declaradas dentro de uma função são locais a essa função.

PHP

Algumas variáveis Super Globais

- `$_SERVER` - tabela contendo as variáveis de ambiente do servidor;
- `$_SESSION` - tabela contendo as variáveis de sessão;
- `$_GET` - tabela contendo as variáveis de um formulário HTML em modo "get";
- `$_POST` - tabela contendo as variáveis de um formulário HTML em modo "post";

- Os operadores são os usuais da linguagem `C`
- As precedência dos operadores são também as usuais;
- Os comentários são também os usuais.

PHP — Estruturas de Controlo

Estruturas de Controlo

- composição sequencial;
- atribuição.
- condicionais:
  - `if`
  - `if else`
  - `elseif`
  - `switch`
- Ciclos:
  - `while`
  - `for`
  - `for each`
  - `do ...while`
- Funções.

PHP — Programação Modular

Programação modular - O PHP tem uma dupla personalidade: programação imperativa clássica; programação orientada aos objectos.

Nestas breves notas vou tratar só da primeira aproximação.

Funções a sintaxe e semântica da declaração de funções é similar à da linguagem C;

```
function
nome_da_funcao($arg1,$arg2,...,$argN);
```

Comunicação entre funções a chamada de uma função é também idêntica à da linguagem C;

```
nome_da_funcao(val1,val2,...,valN);
```

Além dos argumentos a comunicação pode também ser feita através da utilização de variáveis globais.

Comunicação entre módulos por módulos entenda-se ficheiros. Esta é fácil de responder: não é possível (a menos das variáveis super-globais).

PHP — Comunicação Entre Ficheiros

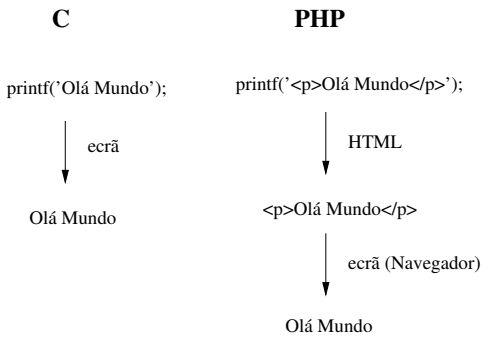
Comunicação entre ficheiros: como foi dito anteriormente não há mecanismos de comunicação entre ficheiros, isto dado que o protocolo HTML não tem a noção de estado.
Soluções?

- Formulários (POST ou GET).
- Sessões (com as respectivas variáveis de sessão).
- Ficheiros auxiliares: é possível ler e escrever ficheiros

PHP & HTML

A *entradas* e *saídas* na linguagem PHP são sempre feitas através da linguagem HTML:

Saídas Os comandos usuais de visualização de uma linguagem de programação estão presentes no PHP (sintaxe similar ao C), no entanto elas têm de ser pensadas como tendo um passo intermédio que é o HTML



PHP & HTML

Entradas a leitura de valores é feita somente através dos formulários HTML

- especificação do ficheiro PHP que vai processar os dados do formulário (no formulário HTML):

```
<form action='processa.php' method='post'>
```

- Especificação dos nomes dos campos que vão receber os valores do formulário (no formulário HTML):

```
<input type='text' name='nomedocampo'>
```

- a "leitura" dos valores no PHP é feita através da matriz associativa `$_POST` (ou `$_GET`) (no ficheiro PHP):

```
$nomeVariavel = $_POST['nomedocampo'];
```

MySQL

Referências — MySQL, páginas oficiais.

- `http://www.mysql.com/` — Página geral
- `http://dev.mysql.com/doc/` — "MySQL Documentation"
- `http://dev.mysql.com/doc/refman/5.5/en/` — Manual de Referência.

"MySQL GUI Tools" — ferramenta gráficas de administração/utilização.

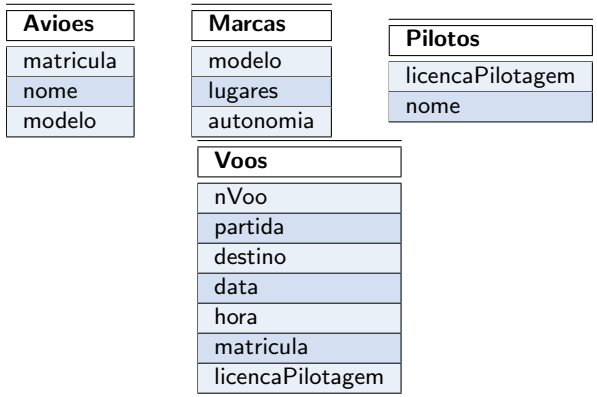
- `http://dev.mysql.com/downloads/gui-tools/`

Alguns dos comandos mais úteis na linha de comando:

- `use` — seleccionar uma base de dados;
- `show databases/tables` — mostrar as bases de dados/tabelas acessíveis;
- `describe <nome_da_tabela>` — mostra a informação respeitante aos campos de uma dada tabela.

MySQL

Uma base de dados de exemplo:



PHP & MySQL

A ligação entre o PHP e o MySQL é feita através de um conjunto de funções própria do PHP.

- **Ligação ao Servidor MySQL** função `mysql_connect`, argumentos: nomes do servidor, utilizador e senha de acesso.

```
@ $ligacao = mysql_connect($servidor,$utilizador,$senha);

if (mysql_errno()) { // verifica a ligação ao servidor
  echo "<p>Erro: ligação aos servidor não possível</p>";
  exit;
}
```

- **Ligação à Base de Dados do Grupo:** função `mysql_select_db`, argumentos: nomes da base de dados.

```
if (!mysql_select_db($nomeBD)) { // verifica a ligação à
BD
  $msg=mysql_error();
  echo $msg;
  echo '<p>Erro: ligação à Base de Dados não possível</p>';
  exit;
}
```

PHP & MySQL

- **Comandos SQL:** coloca-se a pesquisa que se quer efectuar numa variável do tipo "string", e após isso usa-se a função `mysql_query`, tendo como argumentos os comandos SQL e a ligação.
  - Obter elementos

```
$sql = "SELECT correio_electronico,utilizador,senha FROM pessoa";
$resultado = mysql_query($sql,$ligacao);
```

  - Inserir elementos

```
$sql = "INSERT INTO pessoa (correio_electronico,utilizador,senha)
    VALUES ('ana@ana.pt','Ana','xpto')";
$resultado = mysql_query($sql,$ligacao);
```

- **Obtenção dos Resultados:** os resultados são obtidos através de várias funções, entre elas temos `mysql_fetch_assoc` que permite recolher os resultados obtidos numa tabela associativa;

```
$linha = mysql_fetch_assoc($resultado);
$utilizador = $linha['utilizador'];
$senha = $linha['senha'];
```

Exemplo Prático 1

Dada a base de dados `bdGrupoX` (com $x = 1, 2, \ldots$) construa:

- formulários de entrada de valores para as várias tabelas.
- uma página aonde se visualize o conteúdo da tabela `Voos`.

Exemplo Prático — resolução

1. Apache — convenções — criação da página pessoal (Linux):
   1.1 criar o directório **public_html**
   1.2 colocar as permissões correctamente nesse directório:
   - o directório deve estar acessível para leitura para todos;
   - o caminho até ao directório também deve estar acessível (para leitura) para todos.
   - todos os ficheiros no directório `public_html` devem ser acessíveis para leitura para todos.

   1.3 criar o ficheiro **index.html**, a "porta de entrada" para a página. Por exemplo:

```
<HTML>
<HEAD>
  <TITLE>Página pessoal do Grupo N</TITLE>
</HEAD>
<BODY>
  <H2>Página do Grupo N</H2>
  <UL>
  <LI><A HREF="trabalho1.html">Trabalho 1 de Bases de Dados</A></LI>

  </UL>
</BODY>
</HTML>
```

HTML & PHP

No âmbito deste exemplo (de Base de Dados) os ficheiros HTML serão o mais simples possível.
Ficheiro de entrada do Exeplo, **exemplo.html** contendo uma lista não numerada referenciando os dois pedidos:

- um formulário de entrada de valores para a tabela `pessoa`.
- uma página aonde se visualize o conteúdo da tabela `pessoa`.

Por exemplo:

```
<BODY>
  <H2>Exemplo de Bases de Dados</H2>
  <UL>
  <LI><A HREF="visualizar.php">Visualize o conteúdo da tabela</A></LI>
  <LI><A HREF="inserir.html">Introduzir novos valores na tabela</A></LI>
  </UL>
</BODY>
```

HTML & PHP & MySQL

A ligação com a Base de Dados (MySQL) é feita através do PHP.

1. Fazer a ligação à Bases de Dados. Como este é uma tarefa que terá de ser repetida de todas as vezes que se queira obter/colocar informação na base de dados, o melhor será criar um ficheiro PHP separado, só para essa tarefa, **ligacao.php**.

```
<?php
$utilizadorBD = $_SESSION['utilizadorBD'];
$senhaBD = $_SESSION['senhaBD'];
$servidor = $_SESSION['servidor'];
$nomeBD = "bd01exemplo";

@ $ligacao = mysql_connect($servidor,$utilizador,$senha);

if (mysql_errno()) { // verifica a ligação ao servidor
  echo "<p>Erro: ligação aos servidor não possível</p>";
  exit; // termina de imediato
}

if (!mysql_select_db($nomeBD)) { // verifica a ligação à BD
  $msg=mysql_error();
  echo $msg;
  echo '<p>Erro: ligação à Base de Dados não possível</p>';
  exit; // termina de imediato
}
?>
```

Notar a utilização das variáveis de sessão.

Ligação com a Base de Dados

Por questões de segurança a senha de ligação à base de dados (entre outros) não deve ficar exposta num ficheiro com acesso público.
Solução:

1. Colocar a informação sensível num ficheiro à parte (**.dados.php**);
2. colocar esse ficheiro numa zona exterior ao directório (público) `public_html`, e protegê-lo (~**/DadosBD**);
3. Ir buscar esses valores através de variáveis de sessão.

```
<?php
$_SESSION['utilizadorBD']="bdXX";
$_SESSION['senhaBD']="senhaBDXX";
$_SESSION['servidor']="rena2.mat.uc.pt";
?>
```

Depois é só uma questão de utilizar sessões e, sempre que se queira efectuar a ligação à base de dados, carregar (`require`) os ficheiros apropriados:

```
<?php
session_start();
.
.
.
// Faz a ligação à  base de dados
require_once('../../DadosBD/.dados.php');
require('liga.php');
.
.
.
```

HTML & PHP & MySQL

Para o caso da visualização dos conteúdos da base de dados.

2. Enviar as pesquisas e receber os resultados (`visualizar.php`):
   2.1 fazer a ligação;
   2.2 construir o comando ("query") SQL;
   2.3 enviar o comando ao SGBD;
   2.4 receber os resultados;
   2.5 processar/vizualizar os resultados.

Para o caso da introdução de novos dados na base de dados.

2. Construir um Formulário HTML (`inserir.html`):
   2.1 especificar qual é o ficheiro PHP que vai processar os dados;
   2.2 especificar os campos que vão receber os dados.
3. Enviar as pesquisas e receber os resultados (`processaInserir.php`):
   3.1 fazer a ligação;
   3.2 receber os dados do formulário;
   3.3 construir o comando ("query") SQL;
   3.4 enviar o comando ao SGBD;
   3.5 mostrar os resultados da consulta.

HTML & PHP & MySQL — **visualizar.php**

```
<?php
session_start();
echo "<body><H2>Exemplo de Bases de Dados - Visualizar Dados</H2>";
// Faz a ligação à Bases de Dados
require("../../DadosBD/.dados.php");
require("ligacao.php");
// Constroí o comando SQL
$sql = "SELECT correio_electronico,utilizador,senha
        FROM pessoa";

// Envia o comando ao SGBD (a variável $ligacao é definida em ligacao.php)
$resultado = mysql_query($sql,$ligacao);
/*
O resultado é obtido através de uma função apropriada numa matriz associativa, a cada chamada dessa
função
tem-se como efeito colateral o avançar duma posição na matriz. Vai-se construir uma tabela em HTML
*/
echo "<CENTER> \n";
echo "<TABLE BORDER='2'>\n";
echo "<TR><TH>Utilizador</TH><TH>Senha</TH></TR>\n";
// Os identificadores dos campos da matriz são os nomes dos campos da tabela da base de dados
while ($linha = mysql_fetch_assoc($resultado)) {
    $utilizador = $linha['utilizador'];
    $senha = $linha['senha'];
    $correioE = $linha['correio_electronico'];
    echo "<TR> \n";
    echo "<TD>$utilizador</TD><TD>$senha</TD><TD>$correioE</TD>\n";
    echo "</TR>\n";
  }
echo "</TABLE> \n";
echo "</CENTER> \n";
?>
</body>
```

HTML & PHP & MySQL — **inserir.html**

```
<body>
<h2>Exemplo de Bases de Dados - Introduzir Dados</h2>

<form action='processaInserir.php' method='post'>
<center>
<table width='90%' bgcolor='skyblue' border='2'>
<tr>
  <th width='100%' colspan='4' align='center'><b>Introduzir Dados na Tabela "pessoa"</b></th>
</tr>
<tr>
  <td align='center'><b>Utilizador</b></td>
  <td align='center'><input type='text' name='Utilizador' size='15' maxlength='15'></td>
  <td align='center'><b>Senha</b></td>
  <td align='left'><input type='text' name='Senha' size='40' maxlength='40'></td>
</tr>
<tr>
  <td align='center'><b>Correio Electrónico</b></td>
  <td align='center' colspan='3'><input type='text' name='CorreioE' size='100' maxlength='100'></td>
</tr>
<tr>
  <td width=100% colspan='4' align='center'><input type='submit' value='Insere a Nova Informação'></td>
</tr>
</table>
</center>
</form>

</body>
```

HTML & PHP & MySQL — **processaInserir.php**

```
<?php
session_start();
echo "<body><H2>Exemplo de Bases de Dados - Visualizar Dados</H2>";
// Faz a ligação à Bases de Dados
require("../../DadosBD/.dados.php");
require("ligacao.php");

// Obtêm os valores do formulário. Os nomes na matriz POST são os nomes do formulário.
$correioE = $_POST['CorreioE'];
$utilizador = $_POST['Utilizador'];
$senha = $_POST['Senha'];

// Constroí o comando SQL
$sql = "INSERT INTO pessoa (correio_electronico,utilizador,senha)
        VALUES ('$correioE','$utilizador','$senha')";

// Envia o comando ao SGBD (a variável $ligacao é definida em ligacao.php)
$resultado = mysql_query($sql,$ligacao);

if ($resultado) {
    echo "Introdução de novos dados feita com sucesso.\n";
    echo "<meta http-equiv='refresh' content='2; url=exemplo.html' />";
    }
  else {
    echo "Erro - a introdução de novos dados falhou\n";
    echo "<meta http-equiv='refresh' content='2; url=exemplo.html' />";
  }
echo "</body>"
?>
```