



Universidade de Aveiro

Departamento de
Engenharia Mecânica

Informática Industrial 2022/2023

Trabalho Prático Nº 12

PHP + Bases de Dados (MySQL)

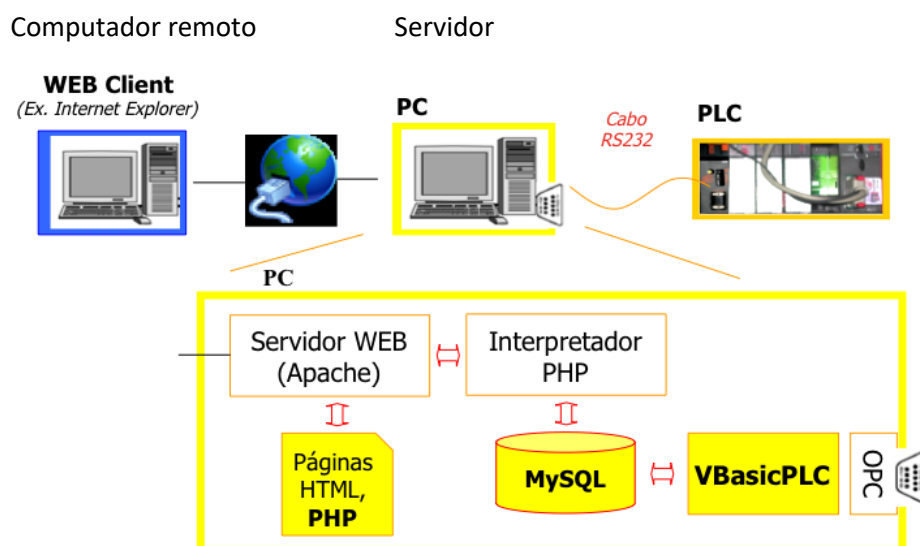
Introdução

Neste trabalho pretende-se, tal como no trabalho anterior, controlar e monitorizar remotamente os recursos fabris, mas desta vez utilizando uma abordagem diferente.

Objetivo

Neste trabalho, pretende-se usar um *Servidor WEB* profissional e desenvolver páginas HTML e PHP para que a partir de um *Browser* remoto seja possível escrever e ler numa base de dados, os dados relativos ao controlo e à supervisão de um reservatório de água.

Usando o *Servidor WEB* (neste caso o Apache) vamos consolidar os conhecimentos adquiridos nas aulas teóricas, relativos ao desenvolvimento de páginas dinâmicas para a WEB, em PHP. Ao contrário das páginas desenvolvidas em HTML que são executadas pelo *Browser* no computador do cliente, as páginas desenvolvidas em PHP são executadas pelo *Servidor WEB + Interpretador PHP* no computador servidor, e permitem, entre outras coisas, gravar no disco duro do servidor as *ordens enviadas pelo cliente a partir do seu Browser*.



No *servidor* residem:

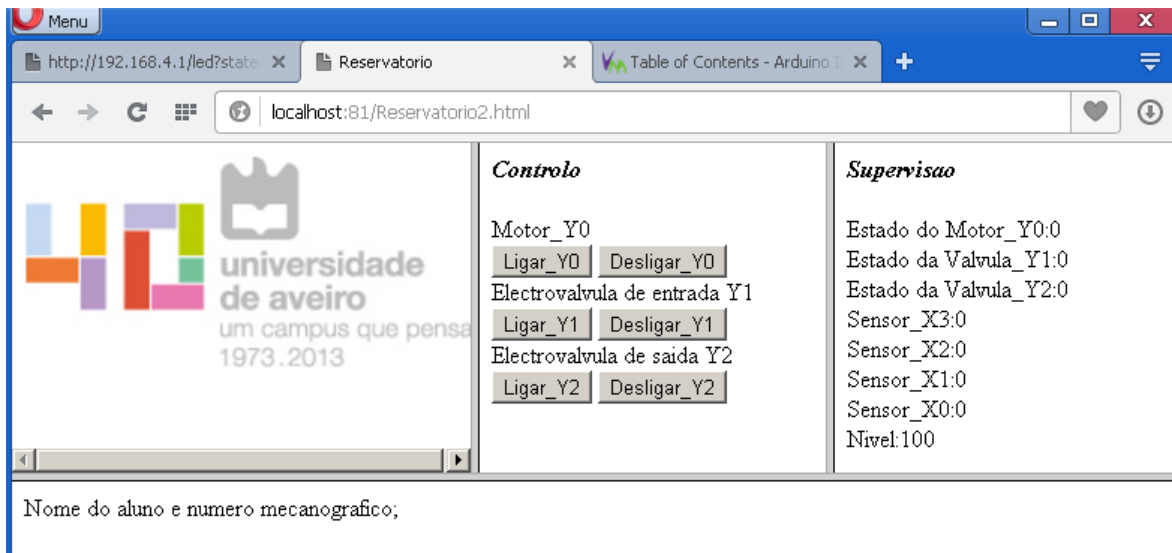
- O servidor WEB Apache, responsável por aceitar as ligações TCPIP e os pedidos do Browser.
- O Interpretador PHP, responsável por traduzir o código PHP em HTML e aceder à BD.
- O Servidor MySQL, responsável por receber as *queries* SQL e gerir a base de dados *reservatorio*
- O Programa desenvolvido em VBasic, responsável por ler a base de dados e aceder ao PLC.

Residem também no computador local:

- As páginas HTML/PHP, no diretório ...\\XAMPP\\htdocs\\
- A base de dados *reservatorio*, no diretório ...\\MySQL\\Data\\

Computador remoto - Browser

A aplicação remota não deve ser alterada, o *Browser* deve ter a mesma interface e as mesmas funcionalidades.



Servidor - Base dados *reservatorio*

Crie uma base de dados *reservatorio* com duas tabelas (Relações): *supervisao* e *controlo*.

controlo(id, data_hora, Y0, Y1, Y2, estado)

supervisao(id, data_hora, Y0, Y1, Y2, X0, X1, X2, X3)

Todos os campos das duas tabelas são do tipo *integer*, exceto os campos “data_hora” do tipo DATETIME. A coluna id deverá ser declarada como chave primária e auto incremental.

Ambas as tabelas irão funcionar do mesmo modo que no trabalho anterior.

Servidor - Servidor WEB + Interpretador de Páginas PHP

Desenvolva uma página PHP (...\\XAMPP\\htdocs\\ControloPLC.php).

Sempre que o cliente, a partir do seu browser, pedir a página *ControloPLC.php* fazendo <http://localhost/ControloPLC.php>, a página PHP deve introduzir os campos Y0, Y1, Y2 e estado na tabela *controlo*. Deve enviar o estado do comando com o valor 0 (comando ainda não enviado para o PLC).

Deve também ler da tabela *supervisao* a linha mais recente e devolver uma pagina HTML ao cliente com o estado de todos os campos.

Antes de desenvolver a página, veja os exemplos no final deste documento.

Servidor - VBasicPLC

É necessário desenvolver um programa em VisualBasic, *VBasicPLC*, que possa aceder à base de dados e por sua vez aceder ao PLC. (pode usar o exemplo do trabalho anterior)

A aplicação *VBasicPLC* deve ler da tabela *controlo* a linha mais recente, se esta linha ainda não tiver sido enviada para o PLC deverá em conformidade ativar ou desativar as saídas Y0, Y1 e Y2 do PLC.

Após envio para o PLC a aplicação deverá atualizar o estado da coluna *estado* para o valor de 1 (comando já enviado para o PCL). A coluna *estado* tem o propósito de impedir a aplicação *VBasicPLC* de enviar comandos continuamente.

Conhecimentos a adquirir

- PHP + MySQL
- Linguagem SQL (Structured Query Language)
 - Tutorial: <http://www.w3schools.com/sql/>

Importante:

- Neste trabalho o PLC não tem programa, todo o controlo e supervisão é efetuado a partir do computador remoto, com a intermediação da aplicação local e do servidor WEB.
- O trabalho será avaliado por questionário individual, na semana seguinte à entrega do mesmo.

Bibliografia:

- Sobre Apache
 - Site <http://www.apache.org>
- Sobre PHP
 - Site <http://www.php.net>
 - Site Manual de PHP
- Carlos Serrão, Joaquim Marques - Programação com PHP 4.3, FCA – Editora de Informática Ida. – Julho 2004
- Carlos Serrão, Joaquim Marques - Programação com PHP 3 e 4, FCA – Editora de Informática Ida. – Setembro 2000. (cota 681.3.06A.214)
- Sobre SQL
- Introduction to SQL http://www.w3schools.com/sql/sql_intro.asp
- Sobre MySQL
 - http://www.w3schools.com/php/php_mysql_intro.asp
 - MySQL Stored Procedure Programming, Steven F., Guy H., 2006. Chapter 13
 - Teach.Yourself.MySQL.in.10.Minutes, By Sams.Sams, 2006. Lesson 24.

Exemplo I: PHP

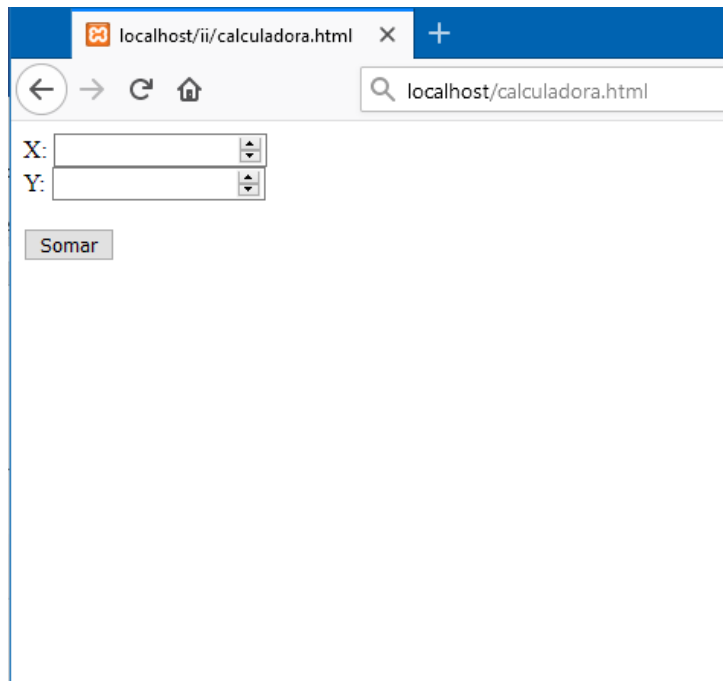
PHP - Calculadora

Este exemplo apresenta uma calculadora desenvolvida em HTML e PHP. Esta calculadora apenas permite adicionar dois números (figura seguinte).

O código em HTML necessário à criação da interface (*<form>*) está gravado no servidor, mais exatamente no *ficheiro Calculadora.html* (canto inferior esquerdo da figura), e o código PHP necessário ao processamento dos dados introduzidos nessa interface está gravado no *ficheiro Calculadora.php* (lado direito da figura).

Para entendermos este exemplo necessitamos de saber utilizar a interface *<form>* disponível em HTML, bem como as entradas *input* do tipo *number*. Podemos ver no exemplo seguinte que uma das entradas de texto tem o nome *name="X"* e a outra o *name="Y"*. Podemos também verificar que o utilizador ao seleccionar o botão *"Somar"* está de facto a executar (*form action*) o ficheiro escrito em PHP que irá processar os valores previamente introduzidos pelo utilizador no seu *BrowserWEB*.

Quando o ficheiro PHP é executado, no lado do servidor, ele pode aceder aos valores introduzidos pelo utilizador na interface *"Form"* usando o array *\$_POST[nome do input da interface HTML]*. Neste caso os elementos *\$_POST['X']* e *\$_POST['Y']* contêm os valores introduzidos pelo utilizador nas entradas *X* e *Y* da interface. A variável *\$Z* recebe o resultado da soma.



//ficheiro calculadora.php

```
<html>
  <body>

    <?php

      $Z = $_POST['X']+$_POST['Y'];
      echo $_POST['X'], "+", $_POST['Y']. "=". $Z;

    ?>

    <br><a href="calculadora.html" > tentar
    outra vez </a>

  </body>
</html>
```

// ficheiro calculadora.html

```
<html>
  <body>
    <form action="calculadora.php" method="post" >
      X: <input type="number" name="X"><br>
      Y: <input type="number" name="Y">
      <br><br>
      <input type="submit" value="Somar">
    </form>
  </body>
</html>
```

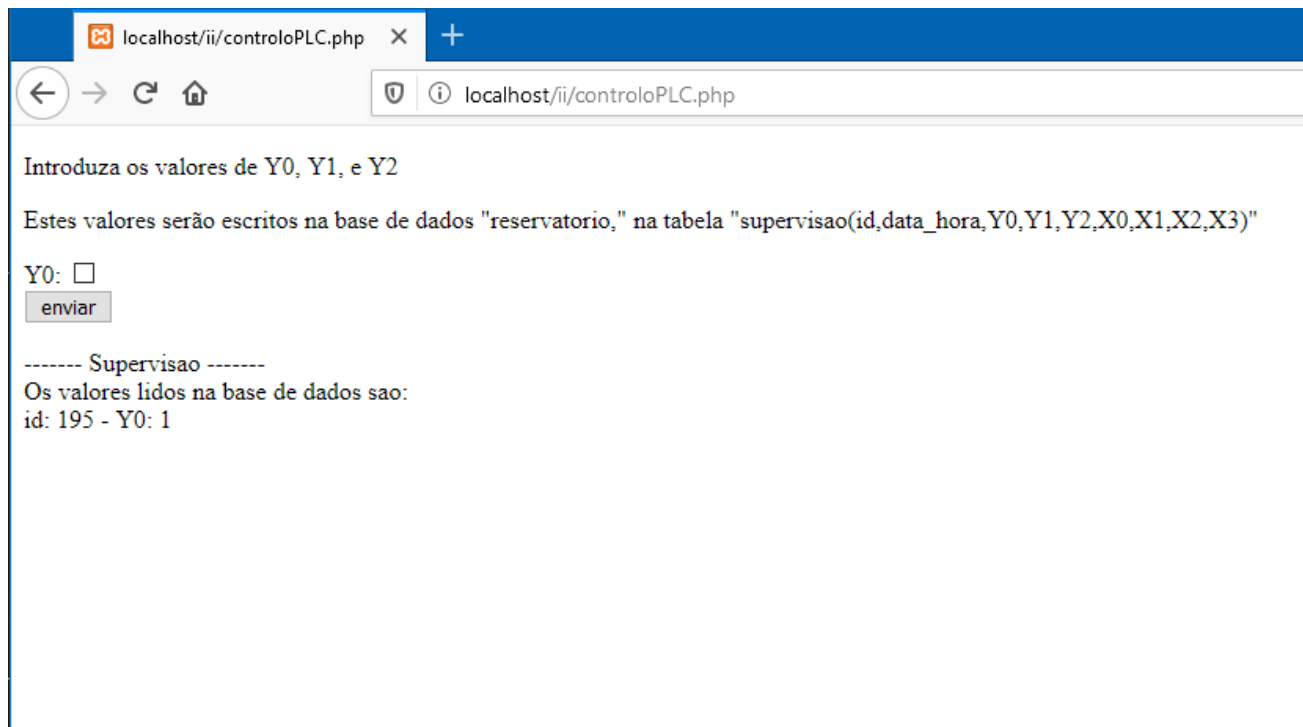
Exemplo II: PHP + MySQL

PHP – Acesso às bases de dados MySQL

A linguagem PHP permite estabelecer ligações TCP/IP com o gestor de bases de dados *MySQL server* e enviar-lhe comandos SQL. Quando o programa *MySQL server* recebe comandos de texto SQL na porta TCP 3306 executa esses comandos. Ao executar esses comandos o servidor *MySQL* pode por exemplo criar novas bases de dados, com tabelas, campos e registos.

Neste exemplo, vamos considerar a utilização de três computadores ligados à Internet. Num deles reside o Browser WEB, no outro reside o Servidor WEB e no terceiro reside o Servidor *MySQL*.

As páginas WEB escritas em PHP são pedidas pelo Browser WEB mas são executadas/interpretadas no servidor WEB. Por sua vez o interpretador de páginas PHP, usado neste exemplo, estabelece uma ligação TCP/IP com o servidor *MySQL* e envia-lhe pedidos/comandos SQL, recebe as respostas do servidor *MySQL* e converte-as para a linguagem HTML. Nessa altura, o servidor WEB envia essas páginas para o Browser WEB onde o utilizador poderá visualizar os dados presentes na base de dados *MySQL*.



localhost/ii/controloPLC.php

Introduza os valores de Y0, Y1, e Y2

Estes valores serão escritos na base de dados "reservatorio," na tabela "supervisao(id,data_hora,Y0,Y1,Y2,X0,X1,X2,X3)"

Y0: ☐

enviar

----- Supervisao -----

Os valores lidos na base de dados sao:
id: 195 - Y0: 1

Desenvolva em PHP o exemplo seguinte

Página HTML/PHP (controloPLC.php)

```
<html>
  <body>
    <form action="controloPLC.php" method="POST">
      <p>
```

Defina o comando a executar.

O comando é inscrito na base de dados "reservatorio," na tabela "controlo"

</p>

<div>

Y0:

<input type="submit" name="Y0" value="Ativar">

<input type="submit" name="Y0" value="Desativar">

</div>

</form>

<?php

/* Conectando, escolhendo a base de dados */

\$link = mysqli_connect("localhost", "reservatorio", "123456", "reservatorio") or die("Não pude conectar: " . mysqli_error());

if(isset(\$_POST['Y0'])){

if(\$_POST['Y0'] == 'Ativar'){

\$Y0 = 1;

}else{

\$Y0 = 0;

}

/* Query SQL de atualização */

\$query = "INSERT INTO controlo SET y0=" . \$Y0;

\$result = mysqli_query(\$link, \$query) or die("A query falhou: " . mysqli_error(\$link));

}

echo "----- Supervisao -----
Os valores lidos na base de dados sao:
";

/* Fazendo a query SQL DE LEITURA DA BASE DE DADOS*/

\$query = "SELECT * FROM controlo order by id desc limit 1";

\$result = mysqli_query(\$link, \$query) or die("A query falhou: " . mysqli_error(\$link));

if (\$result->num_rows > 0) {

while(\$row = \$result->fetch_assoc()) {

echo "id: " . \$row["id"]. " - Y0: " . \$row["y0"]."
";

}

}

/* Liberta o resultado */

```
mysqli_free_result($result);
```

```
/* Fecha a ligação à base de dados*/
```

```
mysqli_close($link);
```

```
?>
```

```
</body>
```

```
</html>
```


Exemplo III: PHP + MySQL

Adapte o exemplo II para funcionar completamente com todas as saídas.