

Desenvolvimento Web I

PHP

Tiago Lopes Telecken
telecken@gmail.com

Referências – Sites e tutoriais

- <http://www.php.net/> (Página oficial PHP)
- http://www.php.net/manual/pt_BR/ (documentação em português)
- <http://www.phpbrasil.com/> (comunidade php)
- <http://pt.wikipedia.org/wiki/PHP> (wiki)

Referências – Editores e IDEs

- <https://code.visualstudio.com/> VS Code - Editor de PHP, HTML ,JS ,CSS entre outros
- <https://www.eclipse.org/pdt/> Eclipse PHP Development Tools
- <https://www.jetbrains.com/phpstorm/features/> PHPstorm
- <https://netbeans.org/features/php/> Netbeans

Instalações

Instalar pacotes

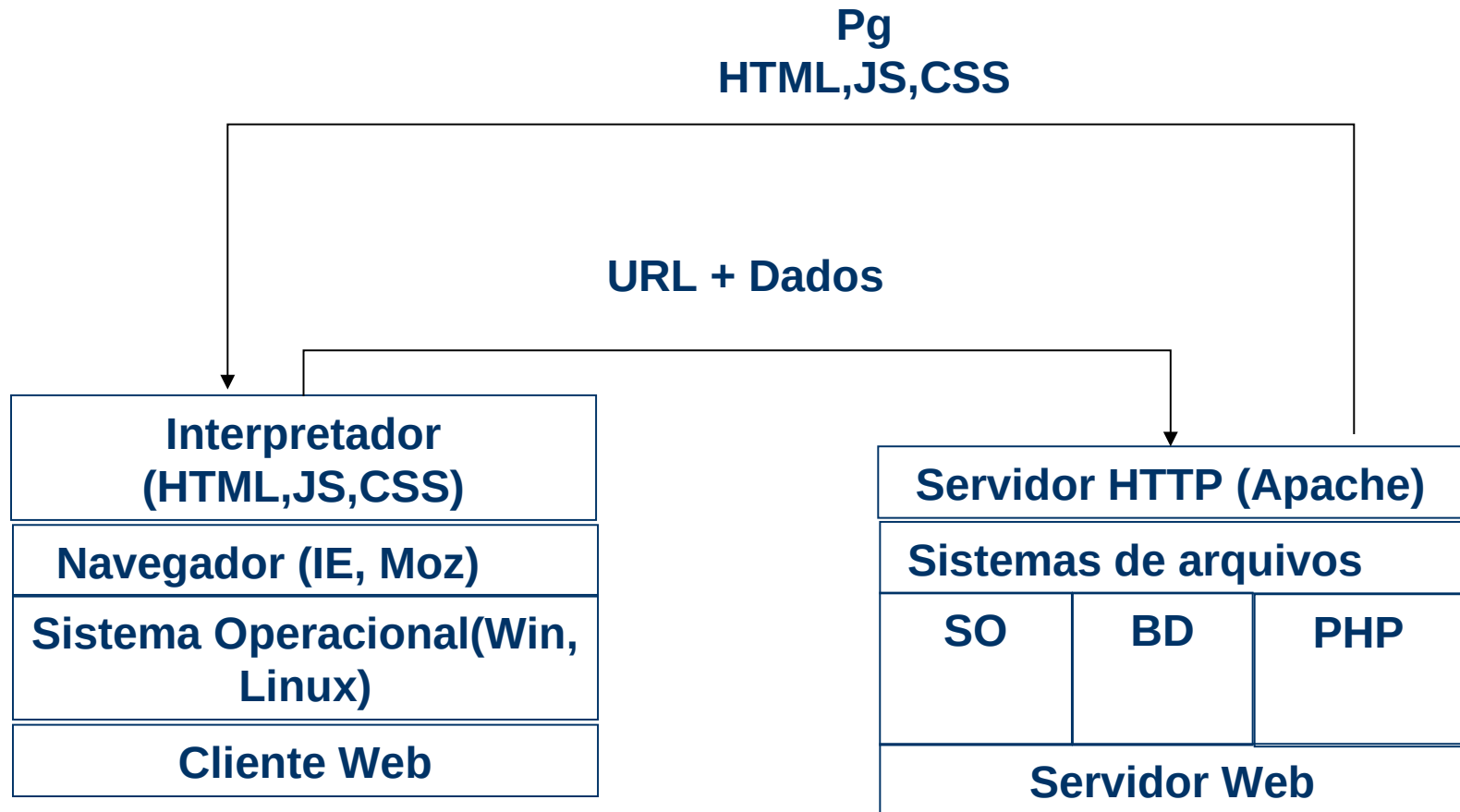
- <https://bitnami.com/stacks> (Instalar pacote com Postgres (ou Mysql), Apache e Php no Linux ou Windows. Escolher/clicar uma das seguintes infraestruturas WAMP, LAMP, WAPP ou LAPP)
- <https://www.easyphp.org/download.php/> (instalar pacote no Windows com php,apache, mysql)
- <https://www.wampserver.com/en/> (Wamp - instalar pacote com php, apache e postgres no windows)

A maioria destes pacotes também instala junto o phppgadmin ou o phpmyadmin

Instalar os componentes separadamente

- <https://www.php.net/downloads> (instalar PHP)
- <http://www.apache.org/> (Apache)
- <https://www.postgresql.org/download/> (instalar postgres)
- <https://www.mysql.com/> (instalar mysql)

Arquitetura web



Arquitetura web

- **Páginas estáticas**

- O cliente envia para a máquina servidora uma requisição que aponta par uma pg HTML que está no sistema de arquivos do servidor
- O servidor HTTP recebe a requisição, acessa a pg solicitada e envia de volta ao cliente
- Se a pg HTML conter outros códigos, como JavaScript, o código vai junto com a página enviada
- Se a pg HTML referencia arquivos externos (arquivos JS, CSS, etc) os arquivos referenciados também são enviados.

Arquitetura web

- **Páginas dinâmicas**

- O cliente envia para a máquina servidora uma requisição que aponta para uma pg PHP que está no sistema de arquivos do servidor. Eventualmente dados do cliente também são enviados (normalmente vindos de um formulário HTML)
- O servidor HTTP recebe a requisição, e acessa a pg solicitada
- A pg php solicitada conterá comandos e estes comandos serão executados. Os comandos são muito variados, como por exemplo:
 - Construir uma pg html dinamicamente (a pg pode conter JS, css, etc)
 - Acessar um BD (alterar e consultar)
 - Acessar recursos do SO, Sistemas de arquivos e outros programas do servidor
- Ao final da execução dos comandos, normalmente uma pg HTML é gerada e enviada de volta ao cliente

O PHP

- PHP - Hypertext Preprocessor
- Linguagem de programação para sistemas Web (*servidor*). A mais difundida de sua categoria
- Versões:
 - PHP/FI: (*Personal Home Page/Forms Interpreter*) Criada em 1994 por Rasmus Lerdorf. Conjunto de scripts voltados a criação de páginas dinâmicas. Em 1995 o código fonte do php foi disponibilizado na web.
 - PHP/FI 2: lançada em 1997. Neste ano aproximadamente 1% dos domínios da internet utilizavam php.
 - PHP3: (*Hypertext Preprocessor*) disponibilizado em 1998. Extensibilidade (API), conexão com BD e outros sistemas, orientação a objetos. Ao final de 1998 o php já estava presente em 10% dos domínios da internet.
 - PHP4: Lançado em 2000. Melhoria em recursos, performance, modularidade. Uso: 20% dos domínios.
 - PHP5: Lançado em 2004. Mudança no modo de orientação a objetos.

O PHP – O programa

- Um programa PHP roda no servidor, pode interagir com BD, SO e outros sistemas do Servidor.
- É uma página (um arquivo texto interpretado pela *Engine* PHP) que além de interagir com sistemas do servidor pode montar pgs HTML e envia-las para clientes web
- As páginas PHP devem ser salvas com a extensão .php . Entretanto também pode-se salvar páginas com as extensões: .php3, php4, phtml

PHP a linguagem

- Interpretada (não precisa compilar o código)
- Fracamente tipada (definição de tipos de dados flexível)
- Sintaxe semelhante ao Java, JavaScript, C
- PHP é case sensitive
- O código PHP pode ser misturado com outros textos/códigos

Delimitadores

- Um código PHP deve estar entre delimitadores
- Todo comando PHP deve ser separado por ;
<?php
// código;
// código;
// código;
?>
- Outros delimitadores: <? ?>

Comentários

- Uma linha
 - `// ola`
 - `# ola`
- Várias linhas
 - `/* ola`
 - `ola */`

Escrevendo na página do cliente

- O comando “echo” escreve um texto na página que será montada e enviada ao cliente

```
<?php  
    echo 'Ola Mundo!';  
?>
```

Escrevendo

- Pode-se escrever na página do cliente simplesmente colocando o texto fora dos delimitadores

Ola Mundo fora do PHP!

```
<?php
```

```
    echo " \n Ola Mundo dentro do PHP! ";
```

```
?>
```

Escrevendo

- **Pode-se intercalar vários trechos de texto e código php**

Ola Mundo fora do PHP1!

```
<?php
```

```
    echo ' Ola Mundo dentro do PHP1! ';
```

```
?>
```

Ola Mundo fora do PHP2!

```
<?php
```

```
    echo ' Ola Mundo dentro do PHP2! ';
```

```
?>
```

Escrevendo

- Pode-se escrever código HTML no arquivo que será enviado para o cliente. E quando o navegador do cliente recebe um código HTML ele é interpretado

```
<html>
  <head></head>
  <body>
    <?php
      echo "<H1> Ola Mundo</H1>";
    ?>
    <b>Ola mundo fora do php</b>
  </body>
<?php echo "</html>"; ?>
```


Escrevendo

```
<html>  
  <head></head>  
  <body>  
    <H1> Ola Mundo</H1>  
    <b>Ola mundo fora do php</b>  
  </body>  
</html>
```

Ola Mundo

Ola mundo fora do php

Escrevendo

- Pode-se escrever qualquer texto que possa ser colocado em uma página HTML. Depois, no cliente, este texto será interpretado (Ex: JavaScript)

```
<html>
  <head></head>
  <body>
    Ola mundo com HTML<br>
    <script language="JavaScript">
      document.write('<B>Ola mundo JS!</B><BR>');

    <?php
      echo "document.write('<B>Ola mundo JS!Escrito com php</B><BR>');";
    ?>
  </script>
  <?php
    echo " <b>Ola mundo com HTML!Escrito com PHP</b>";?>
  </body>
</html>
```

Escrevendo

```
<html>
  <head></head>
  <body>
    Ola mundo com HTML<br>
    <script language="JavaScript">
      document.write('<B>Ola mundo JS!</B><BR>');
      document.write('<B>Ola mundo JS!Escrito com
php</B><BR>');
    </script>
    <b>Ola mundo com HTML!Escrito com PHP</b>
  </body>
</html>
```

Escrevendo

- O comando echo e os textos fora dos delimitadores escrevem qualquer texto em um arquivo que será enviado para o cliente (Um navegador web).
- Se o código que chegar for HTML ele será interpretado como HTML, se for Java Script será interpretado como JavaScript. Se chegar com erros de sintaxe a página irá gerar erros ou outros efeitos indesejáveis. (O interpretador php não detecta erros do HTML, JS, etc)

Variáveis

- Para criar variáveis utiliza-se o operador de atribuição “=”
- O nome das variáveis devem começar por “\$”

```
$nome = 'João ';
```

```
$sobrenome= 'Silva';
```

```
echo "$nome $sobrenome"; // mostra: João Silva
```

```
echo '$nome $sobrenome'; //mostra: $nome  
$sobrenome
```

```
// “ ” substitui as variáveis por seu conteúdo
```

```
// ' ' escreve sem substituir
```

Variáveis

- Variáveis podem receber o conteúdo de outras variáveis

```
$a=5;
```

```
$b=$a;
```

```
echo $b; //mostra 5
```

```
$b=10;
```

```
echo $a; //mostra 5
```

```
echo $b; //mostra 10
```

Variáveis

- Variáveis podem apontar para uma mesma área na memória

```
$a=5;
```

```
$b=&$a; //$b apontará para o mesmo end. de  
$a
```

```
$b=10;
```

```
echo $a; //mostra 10
```

```
echo $b; //mostra 10
```

Tipos de variáveis

- **Booleano**
 - TRUE ou FALSE

```
$mostrar=TRUE;  
If ($mostrar)  
{ echo 'Ola'; }
```

```
$a=10;  
$b=(5 > $a);  
echo $b; //imprime FALSE
```

- **Também são considerados FALSE**
 - 0 ; 0.0 ; “0”; NULL ; string vazia; array vazio

Tipos de variáveis

- Tipo numérico
 - `$a= 1022; // inteiro`
 - `$a= -1022; // negativo`
 - `$a= 10.22; // ponto flutuante`
 - `$a= 0x1A; // hexadecimal`
 - `$a= 4e23; // notação científica`

Tipos de variáveis

- **String**
 - `$a='ola';`
 - `$a="Ola";`

Convertendo tipos - settype

- ```
<?php
$foo = "5bar"; // string
$bar = true; // boolean

settype($foo, "integer"); // $foo é agora 5
(integer)
settype($bar, "string"); // $bar é agora "1"
(string)
?>
```

# Convertendo tipos – type casting

**<?**

```
$foo = 10; // $foo é um inteiro
$bar = (boolean) $foo; // $bar é um booleano
```

**?>**

**As moldagens permitidas são:**

**(int), (integer) - converte para inteiro**

**(bool), (boolean) - converte para booleano**

**(float), (double), (real) - converte para ponto flutuante**

**(string) - converte para string**

**(binary) - converte para string binária (PHP 6)**

**(array) - converte para array**

**(object) - converte para objeto**

# Constantes

- Os valores só podem ser boolean, inteiro, ponto flutuante ou string
- Utiliza-se o comando “define”

<?

```
define(“MAXIMO_REGISTROS”,100);
```

```
Echo MAXIMO_REGISTROS;
```

?>



# Operadores - Atribuição

- `$a = 5;`
- `$a++; //$a= $a+1;`
- `$a--;`
- `$a+=5; //$a=$a+5;`

# Operadores Concatenação

- .

```
$b="ola1";
```

```
$c="ola2";
```

```
a$=$b."texto1"." texto2 ".$b.$c;
```

```
echo $a;
```



# Operadores Aritiméticos

| Operador     | Ação                                     |
|--------------|------------------------------------------|
| $\$a + \$b$  | // Adição soma \$a mais \$b              |
| $\$a - \$b$  | // Subtração subtrai \$a de \$b          |
| $\$a * \$b$  | // multiplicação de \$a por \$b          |
| $\$a / \$b$  | // divisão de \$a por \$b                |
| $\$a \% \$b$ | // Resto resto da divisão de \$a por \$b |

# Operadores de comparação

| Ex.                         | Nome                      | Resultado                                              |
|-----------------------------|---------------------------|--------------------------------------------------------|
|                             |                           |                                                        |
| <b>\$a != \$b</b>           | <b>Diferente</b>          | <b>Verdadeiro se \$a não for igual a \$b.</b>          |
| <b>\$a &lt;&gt;<br/>\$b</b> | <b>Diferente</b>          | <b>Verdadeiro se \$a não for igual a \$b.</b>          |
| <b>\$a &lt; \$b</b>         | <b>Menor<br/>que</b>      | <b>Verdadeiro se \$a é estritamente menor que \$b.</b> |
| <b>\$a &gt; \$b</b>         | <b>Maior que</b>          | <b>Verdadeiro se \$a é estritamente maior que \$b.</b> |
|                             |                           |                                                        |
| <b>\$a &lt;=<br/>\$b</b>    | <b>Menor ou<br/>igual</b> | <b>Verdadeiro se \$a é menor ou igual a \$b.</b>       |
| <b>\$a &gt;=</b>            | <b>Maior ou</b>           |                                                        |

# Operadores lógicos

| Exemplo                   | Nome       | Resultado                                                         |
|---------------------------|------------|-------------------------------------------------------------------|
| <b>\$a and \$b</b>        | <b>E</b>   | <b>Verdadeiro (TRUE) se tanto \$a quanto \$b são verdadeiros.</b> |
| <b>\$a &amp;&amp; \$b</b> | <b>E</b>   | <b>Verdadeiro se tanto \$a quanto \$b são verdadeiros.</b>        |
| <b>\$a or \$b</b>         | <b>OU</b>  | <b>Verdadeiro se \$a ou \$b são verdadeiros.</b>                  |
| <b>\$a    \$b</b>         | <b>OU</b>  | <b>Verdadeiro se \$a ou \$b são verdadeiros.</b>                  |
| <b>\$a xor \$b</b>        | <b>XOR</b> | <b>Verdadeiro se \$a ou \$b são verdadeiros, mas não ambos.</b>   |
| <b>! \$a</b>              | <b>NÃO</b> | <b>Verdadeiro se \$a não é verdadeiro.</b>                        |

# Precedência dos operadores

| 1 | Operador                                                       |
|---|----------------------------------------------------------------|
|   | <b>new</b>                                                     |
|   | <b>[</b>                                                       |
|   | <b>! ~ ++ -- (int) (float) (string) (array)<br/>(object) @</b> |
|   | <b>* / %</b>                                                   |
|   | <b>+ - .</b>                                                   |
|   | <b>&lt;&lt; &gt;&gt;</b>                                       |
|   | <b>&lt; &lt;= &gt; &gt;=</b>                                   |

| 2 | Operador                                                        |
|---|-----------------------------------------------------------------|
|   | <b>^</b>                                                        |
|   | <b> </b>                                                        |
|   | <b>&amp;&amp;</b>                                               |
|   | <b>  </b>                                                       |
|   | <b>? :</b>                                                      |
|   | <b>= += -= *= /= .= %= &amp;=  = ^=<br/>&lt;&lt;= &gt;&gt;=</b> |
|   | <b>print</b>                                                    |

# Estruturas de controle

- IF, ELSE, ELSEIF

```
<?php
 if ($a > $b) {
 echo "a é maior que b";
 } elseif ($a == $b) {
 echo "a é igual a b";
 } else {
 echo "a é menor que b b";
 }
?>
```

# Estruturas de controle

- Switch

```
<?php
switch ($i) {
 case 0:
 echo "i igual a 0";
 break;
 case 1:
 echo "i igual a 1";
 break;
 case 2:
 echo "i igual a 2";
 break;
 default:
 echo "i não é igual a 0, 1 ou 2";
}
?>
```

# Estruturas de controle

- While

```
<?php
 $i = 1;
 while ($i <= 10) {
 echo $i++;
 }
?>
```

- Do...While

```
<?php
 $i = 0;
 do {
 echo $i;
 } while ($i > 0);
?>
```

# Estruturas de controle

- For

```
for ($i = 1; $i <= 10; $i++) {
 echo $i;
}
```



# Estruturas de controle

- Continue

```
<?php
 for ($i = 0; $i < 5; ++$i) {
 if ($i == 2)
 continue
 print "$i\n";
 } // imprime 0,1,3,4
?>
```

- Break

```
<?php
 for ($i = 0; $i < 5; ++$i) {
 if ($i == 2)
 break
 print "$i\n";
 } // imprime 0,1
?>
```

# **Desenvolvimento Web I**

## **PHP**

**Tiago Lopes Telecken**  
**telecken@gmail.com**