

**PÓS-GRADUAÇÃO EM
DESENVOLVIMENTO DE
APLICAÇÕES PARA
DISPOSITIVOS MÓVEIS E
CLOUD COMPUTING**

DM107

Desenvolvimento de Web Services com Autenticação sob plataforma Java e PHP

Profa. Daniela E. C. de Almeida

E-mail: daniela.edvana@inatel.br

Prof. Elton Barbosa

E-mail: elton.barbosa@inatel.br

Agenda

- Introdução;
- Por que escolher PHP;
- Instalação;
- Sintaxe básica.

Introdução

- *Hypertext Preprocessor* (PHP), é uma linguagem de programação interpretada cujo o principal objetivo é permitir a escrita de páginas *web* dinâmicas, no entanto é possível explorar outras possibilidades.
- Foi criada em 1994 por Rasmus Lerdorf.
- Primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua *home-page* apenas para que ele pudesse ter informações sobre as visitas realizadas em seu *site*.

Introdução

- Primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “*Personal Home Page Tools*”.
- Em meados de 1995 o interpretador foi reescrito, e ganhou o nome de PHP/FI.
- Em meados de 1997 deixou de ser um projeto de Rasmus com contribuições de outras pessoas para ter uma equipe de desenvolvimento mais organizada.

Por que escolher PHP

- PHP nasceu para a *web* e sua integração com servidores *web* é simples.
- PHP tem uma curva de aprendizado suave, comparada a outras linguagens.
- PHP é uma tecnologia livre.
- É fácil encontrar serviços de hospedagem que oferecem PHP.
- Serviços de hospedagem PHP é mais barato que serviços semelhantes para outras tecnologias.

Instalação

- Uma das melhores opções é a utilização do XAMPP que além de gratuito disponibiliza tudo que é preciso para desenvolver em PHP.

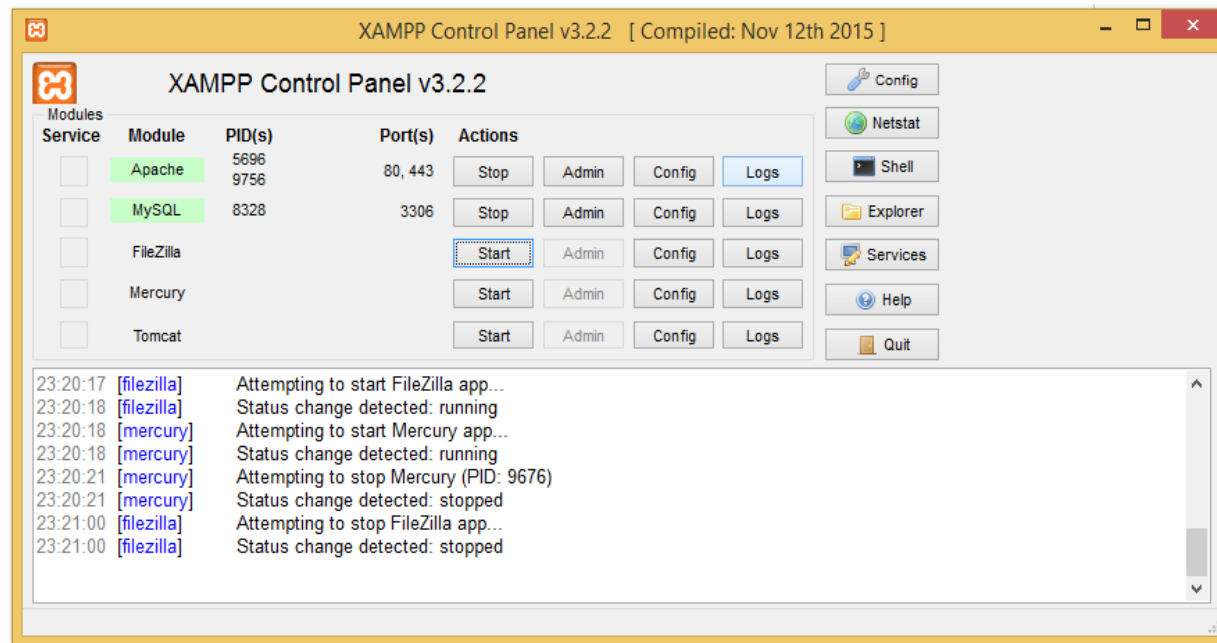


Figura 1 – Painel de controle do XAMPP

Sintaxe Básica

- É possível delimitar código PHP de quatro formas:

Forma 1:

```
<?php Comandos; ?>
```

Forma 2:

```
<? Comandos; ?>
```

Forma 3:

```
<script language="php"> Comandos;</script>
```

Forma 4:

```
<% Commandos; %>
```


PHP

- **Prática 1 - Criando o primeiro projeto usando PHP**

Sintaxe Básica - Variáveis

- As variáveis em PHP são declaradas com \$ por exemplo:

```
<?php
```

```
$a = 10;
```

```
$b = "10";
```

```
$c = 10.2;
```

```
echo "Número Inteiro:" . $a . "<br>" . "String:" . $b . "<br>" . "Ponto Flutuante:" . " " . $c;
```

```
?>
```

Sintaxe Básica - Comentários

- Comentário de uma linha
 - Pode ser delimitado pelo caracter “#” ou por duas barras //.
- ```
<? echo “teste”; #isto é um teste ?>
```
- ```
<? echo “teste”; //este teste é similar ao anterior ?>
```

- Comentário de bloco
 - Tem como delimitadores os caracteres “/*” para o início do bloco e “*/” para o final do comentário.

```
<?
echo “teste”; /* Isto é um comentário com mais de uma linha que funciona corretamente
*/
?>
```

Sintaxe Básica - Operadores

- Aritméticos

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão

Sintaxe Básica - Operadores

- Comparação

==	Igual a
!=	Diferente de
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a

Sintaxe Básica - Operadores

- Lógicos

and	“e” lógico
or	“ou” lógico
xor	ou exclusivo
!	não (inversão)
&&	“e” lógico
 	“ou” lógico

Sintaxe Básica - Operadores

- Para Strings

.

Concatenação

Sintaxe Básica - Operadores

- Atribuição

=	Atribuição simples
-=	Atribuição com subtração
+=	Atribuição com adição
*=	Atribuição com multiplicação
/=	Atribuição com multiplicação
%=	Atribuição com módulo
.=	Atribuição com concatenação

Sintaxe Básica - Operadores

- Incremento e decremento

++	Incremento
--	Decremento

Sintaxe Básica – Estruturas de controle

- if/else

```
<?php
    if ($a > $b) {
        echo "a is bigger than b";
        $b = $a;
    }
?>
```

Sintaxe Básica – Estruturas de controle

- switch

```
<?php
    switch ($i) {
        case 0:
            echo "i equals 0";
            break;
        case 1:
            echo "i equals 1";
            break;
        case 2:
            echo "i equals 2";
            break;
    }
?>
```

Sintaxe Básica – Estruturas de repetição

- while

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

Sintaxe Básica – Estruturas de repetição

- Do-while

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

Sintaxe Básica – Estruturas de repetição

- for

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

Sintaxe Básica – Estruturas de repetição

- foreach
 - Fornece uma maneira fácil de iterar sobre *arrays*. O *foreach* funciona somente em *arrays* e objetos, e emitirá um erro ao tentar usá-lo em uma variável com um tipo de dado diferente ou em uma variável não inicializada.

```
<?php
    $arr = array(1, 2, 3, 4);
    foreach ($arr as &$value) {
        $value = $value * 2;
    }
?>
```

Sintaxe Básica – Quebra de Fluxo

- `break`
 - Pode ser utilizado em laços de *do*, *for* e *while*, além do uso já visto no comando *switch*. Ao encontrar um *break* dentro de um desses laços, o interpretador PHP para imediatamente a execução do laço, seguindo normalmente o fluxo do *script*.

```
while ($x > 0) {  
    if ($x == 20) {  
        echo "erro! x = 20";  
    }  
    break;  
}
```


Sintaxe Básica – Quebra de Fluxo

- continue
 - O comando *continue* também deve ser utilizado no interior de laços, e funciona de maneira semelhante ao *break*, com a diferença que o fluxo ao invés de sair do laço volta para o início dele.

```
for ($i = 0; $i < 100; $i++) {  
    if ($i % 2) continue;  
    echo " $i ";  
}
```

Sintaxe Básica – Arrays

```
<?php
    $array[0] = "valor1";
    $array[1] = "valor2";
    $array[2] = "valor3";
    print_r($array);
    echo "<br>";
    echo "Array na posição 1:" . $array[1];
?>
```

Sintaxe Básica – Arrays

- Outras maneiras de inicializar um *array*.

```
<?php
$array = array(0=>"valor1", 1=>"valor2", 2=>"valor3");
print_r($array);
echo "<br>";
echo "Array na posição 1:" . $array[1];
?>
```

PHP

- **Prática 2 – Exercícios sobre a sintaxe básica PHP**

Referências

- Bento, Evaldo J. **Desenvolvimento web com MySQL e PHP**. Casa do Código.
- Lopez, Antonio. **Learning PHP 7**. Birmingham: Packt Publishing, 2016.
- **PHP Hypertext Preprocessor**. Disponível em <<https://secure.php.net/>>. Acesso em 15 Ago. 2017