

## 1. PHP

Resumidamente, PHP (Hypertext preprocessor - Personal Home Page Tools) é uma linguagem de script executada no lado do servidor, também conhecida como *server side* ou *back end*.

Assim como outras linguagens disponíveis no mercado, o PHP executa o programa no lado do servidor e devolve para o cliente via protocolo de rede http, uma página codificada de internet (HTML, CSS, Javascript,...).

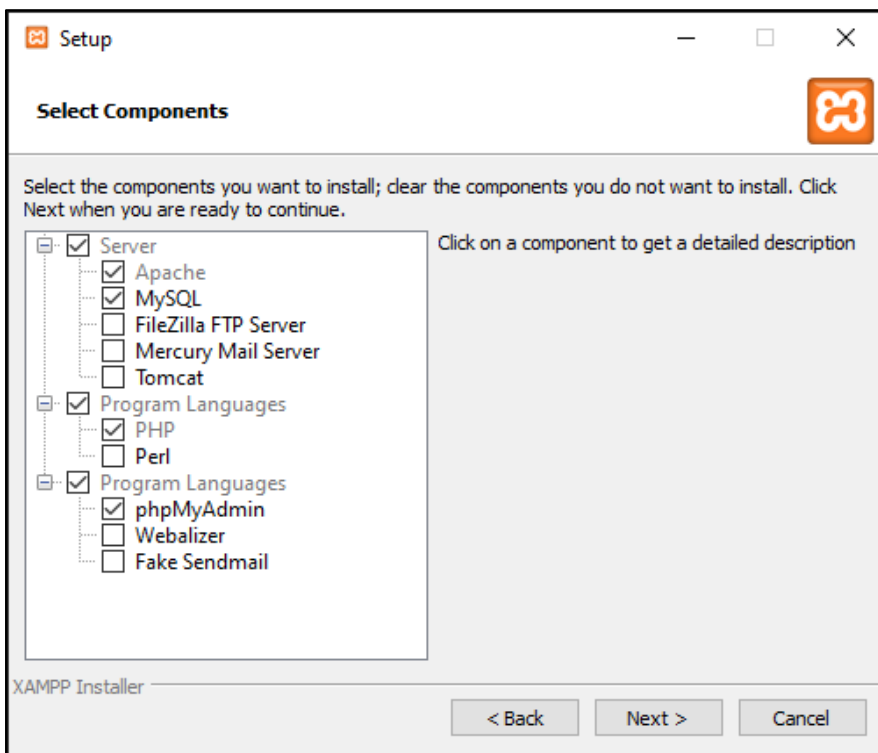
A intenção em rodar um programa no lado do servidor é o de realizar processamento rápido e entregar somente o que o usuário precisa ter acesso. Acesso direto à base de dados que está no servidor para fins de autenticação, leitura e manipulação de dados é uma das principais funções do PHP.

## 2. Instalando um servidor local de internet incluindo o PHP

Existem diversos servidores de internet, um deles é o APACHE. Neste tutorial utilizaremos o software XAMPP para instalar na máquina local o APACHE e o SGBD MariaDB.

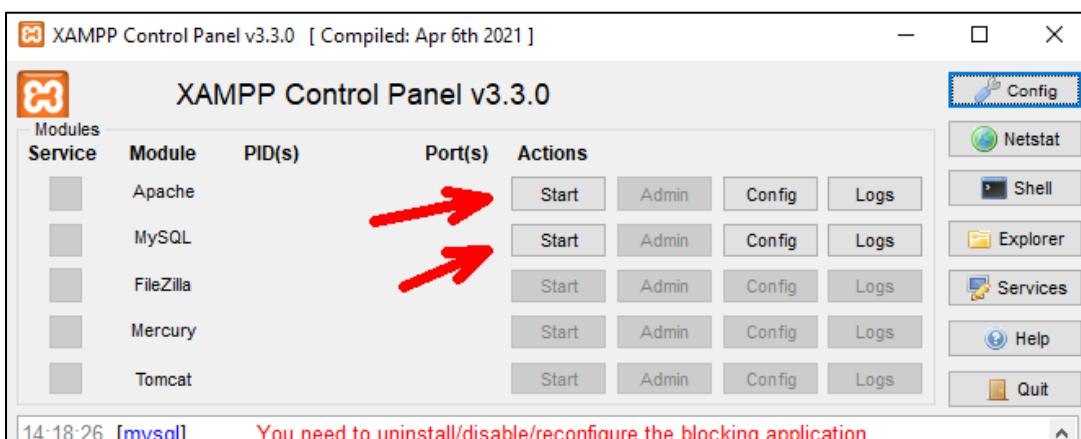
Acesse <https://www.apachefriends.org/>, baixe e instale a última versão.

Não é necessário instalar todos os módulos, marque apenas a indicação abaixo.



Instale no local padrão: "C:\xampp".

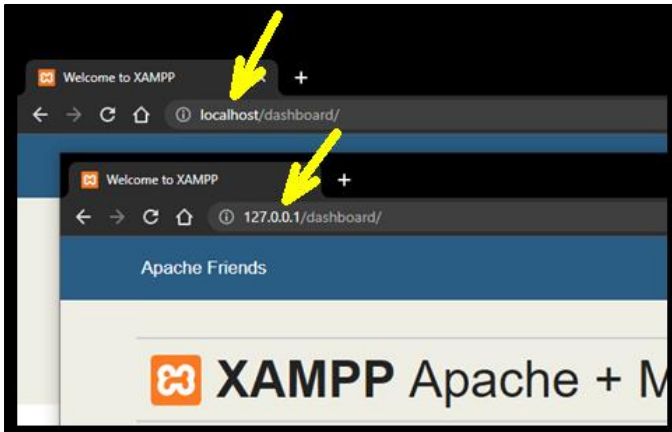
Após a instalação abra o painel do Xampp e "Start" o servidor Apache. O MySql é opcional por hora.



## 2.1. Testando o servidor de internet 1

Abra o navegador de internet (Chrome) e digite na barra de endereços: localhost ou 127.0.0.1

Aparecerá a tela do dashboard do Xampp:

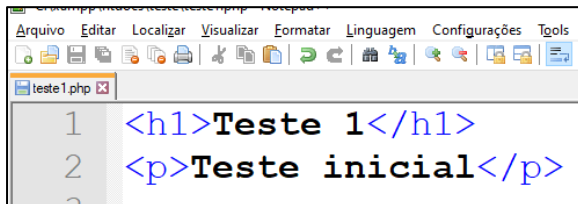


## 2.2. Testando o servidor de internet 2

Vamos agora criar uma página de teste.

Antes, localize a pasta especial “htdocs”, é nela onde deverão estar os scripts html, css, php, Javascript, etc para que o apache (servidor de internet) possa disponibilizar publicamente via protocolo de rede de internet.

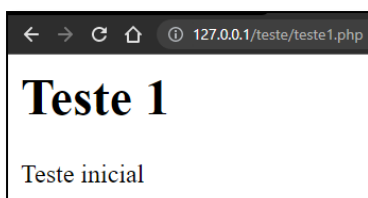
1. Localize a pasta “htdocs” em c:/xampp.
2. Dentro de “htdocs” crie a pasta “teste”
3. Nela, crie o arquivo “teste1.php” (<https://youtu.be/Q7Gn3U4wf10>)
4. Edite este arquivo com o seu editor preferido. Adotaremos aqui o Notepad++
5. Digite apenas as duas linhas abaixo.



- 6.
7. O script acima está em HTML, tags Header (<h1>) e paragraph (<p>).
8. Sugestão para curso de HTML: <https://www.w3schools.com/html/default.asp>
9. Salve o seu arquivo, abra o navegador e digite o endereço:
10. Localhost/teste ou <http://127.0.0.1/teste>



- 11.
12. Clique em teste1.php para executar o script.



- 13.

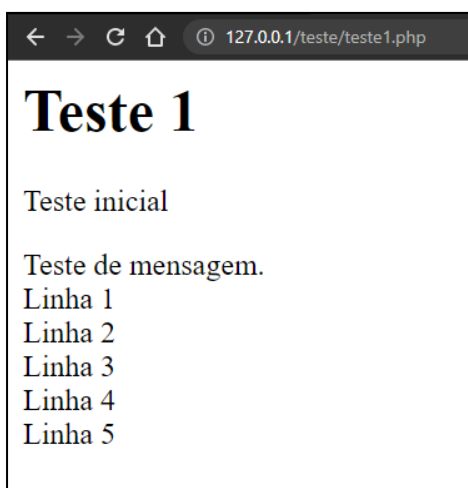
14. Repare que salvamos o arquivo com a extensão “.php”, poderíamos ter salvo com a extensão “.html” ou “.htm”. Contudo deixaremos com a extensão “.php” por conta da configuração padrão do servidor Apache para que possa rodar scripts da linguagem PHP.
15. Inclua as seguintes linhas em seu teste1.php:

```

1 <h1>Teste 1</h1>
2 <p>Teste inicial</p>
3
4 <div><?php echo "Teste de mensagem.";?></div>
5
6 <div>
7     <?php
8     for ($i=1;$i<=5;$i++){
9         echo "Linha " . $i . "<br>";
10    }
11    ?>
12 </div>
13

```

- 16.
17. Sugestão de tutorial da linguagem PHP: <https://www.w3schools.com/php/>
18. Todos os comandos em PHP devem ser precedidos da tag de abertura “<?php” e finalizada com “?>”, assim o Apache sabe que os códigos internos devem ser executados pelo PHP.
19. Salve e veja o resultado no navegador. Atualize a página para carregar a alteração.



- 20.
21. Repare que o cliente que chama o programa “teste1.php” só visualiza o resultado da programação em PHP. Mesmo se o usuário der o comando “CTRL + u” para ver o código do navegador, não verá a programação em PHP. Está é a ideia da arquitetura CLIENTE-SERVIDOR. Experimente, no Chrome, pressionar CTRL + u.

## COMANDOS/FUNÇÕES:

echo	Exibir na tela algum texto ou conteúdo de uma variável. <a href="https://www.w3schools.com/php/php_echo_print.asp">https://www.w3schools.com/php/php_echo_print.asp</a> echo "Teste 123..."; \$teste = "exemplo de teste"; echo \$teste; echo "Valor da variável teste: " . \$teste;
for	Função de loop <a href="https://www.w3schools.com/php/php_looping_for.asp">https://www.w3schools.com/php/php_looping_for.asp</a> for ( atribui valor inicial ; limita o valor final ; incremente o valor) for(\$i=1;\$i<=5;\$i++){ echo \$i; }
Concatenar	Concatenar é o mesmo que “juntar”, no código de teste1.php utilizamos o “PONTO” para juntar texto e variável: echo "Linha " . \$i . " ";

	Acima a palavra "Linha " está concatenada com o valor da variável \$i e com a tag " ".
; (ponto e vírgula)	Praticamente toda linha de comando PHP precisa estar finalizada com o ";"
TAG  	É a tag HTML para quebra de linha (pular uma linha) – line break. <a href="https://www.w3schools.com/tags/tag_br.asp">https://www.w3schools.com/tags/tag_br.asp</a>
Variáveis PHP	Toda variável PHP deve ser precedida do "\$". <a href="https://www.w3schools.com/php/php_variables.asp">https://www.w3schools.com/php/php_variables.asp</a>

## 2.3. Formulário HTML

### FORMULÁRIO HTML (sem processamento com PHP)

Crie o arquivo "teste2.php", nele vamos criar um formulário em HTML para solicitar o nome do usuário e, ao enviar o formulário, o PHP deverá carregar o que o usuário digitou e exibir uma mensagem na tela.

1. Digite o código HTML abaixo:

```

1 <h3>Formulário</h3>
2
3 <form method="POST" action="teste2.php">
4   <input type="text" name="nome_user" placeholder="Digite seu nome">
5   <button>Enviar</button>
6 </form>
7

```

- 2.
3. Salve e rode-o no navegador (teste2.php)
4. Teste digitando e clicando em enviar.

- 5.
6. A TAG <FORM> do HTML é especial, com ela o usuário (front end) pode enviar informações (texto e arquivos) diretamente para o servidor processá-los.
7. Por hora nosso formulário não faz nada além de colher o nome e enviar para ele mesmo.
8. Altere o atributo "method" da tag FORM para "GET" ao invés de "POST" e veja a diferença ao enviar seu formulário.

```

1 <h3>Formulário</h3>
2
3 <form method="GET" action="teste2.php">
4   <input type="text" name="nome_user" placeholder="Digite seu nome">
5   <button>Enviar</button>
6 </form>
7

```

- 9.
10. Salve e atualize o navegador. Repare que o que você digitar e enviar aparecerá como uma variável de URL na barra de endereço do navegador. Com o método POST as variáveis não ficam exibidas na barra de endereço.

- 11.

12. Qual método utilizar GET ou POST em formulários HTML? Use sempre o POST, raros os casos utilizamos o GET. De qualquer forma é bom saber que os dois existem.

### COMANDOS/FUNÇÕES:

<code>&lt;form method="POST" action="teste2.php"&gt;</code>	TAG HTML para formulário. "method" indica qual método os dados serão enviados "GET" ou "POST". "action" indica para qual script os dados serão enviados. <a href="https://www.w3schools.com/html/html_forms.asp">https://www.w3schools.com/html/html_forms.asp</a>
<code>&lt;input type="text" name="nome_user" placeholder="Digite seu nome"&gt;</code>	TAG <b>&lt;input&gt;</b> é uma tag HTML para receber uma informação do usuário. <b>Type="text"</b> : campo para digitação de texto em geral. <b>name="nome_user"</b> : indica que o que o usuário digitar ficará armazenado na variável "nome_user" (você especifica qualquer nome para esta variável). <b>Placeholder</b> : apenas a mensagem que aparecerá dentro do campo.
<code>&lt;button&gt;Enviar&lt;/button&gt;</code>	TAG <b>&lt;button&gt;</b> : elemento do tipo "botão". Também serve para enviar os dados do formulário para o script indicado na tag <b>&lt;FORM&gt;</b> .

### FORMULÁRIO HTML com processamento em PHP

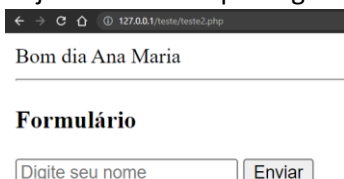
Agora vamos "interceptar" o que o usuário do lado do "cliente" enviou via formulário.

Antes deixe o formulário com o método "POST"

1. Inclua o código abaixo

```
1 <?php
2 if( isset($_POST["nome_user"]) ) {
3
4     $nome_digitado = $_POST["nome_user"];
5
6     echo "Bom dia " . $nome_digitado;
7     echo "<hr>";
8 }
9 ?>
10
11 <h3>Formulário</h3>
12
13 <form method="POST" action="teste2.php">
14     <input type="text" name="nome_user" placeholder="Digite seu nome">
15     <button>Enviar</button>
16 </form>
```

- 2.
3. Ao rodar pela primeira vez o arquivo acima, o comando "if ( )" verifica se existe o "post" da variável "nome\_user" (se ela está "setada", ou seja, possui algum valor – linha 2). Se sim, é atribuído à variável "\$nome\_digitado" o valor do que foi postado (linha 4).
4. O comando `$_GET["nome_user"]` serve para "carregar" a variável "nome\_user" que foi postada do formulário.
5. Em seguida, na linha 6 é exibida a frase "Bom dia " concatenada com a variável \$nome\_digitado.
6. Na linha 7 o comando "echo" exibe a tag html "<hr>" (horizontal line).
7. Veja o resultado após digitar um nome:



- 8.

# ATIVIDADE

Experimente incluir no seu formulário a data de nascimento.

<https://youtu.be/Vg0ZF71FMgU>