

1. PHP

PHP é uma linguagem poderosa e um interpretador, seja incluído em um servidor web como um módulo (**o que vai ser visto no curso**) ou executado separadamente como binário CGI, é possível acessar arquivos, executar comandos e abrir conexões de rede no servidor.

Como existem diferentes maneiras de utilizar o PHP, existem várias opções de configuração controlando seu comportamento. A flexibilidade de configuração do PHP é comparável com a flexibilidade de código. PHP pode ser usado para montar um servidor de aplicações¹ completo, com todo o poder de um usuário shell², ou pode ser usado para inclusão simples de arquivos no lado do servidor com pouco risco em um ambiente bem controlado. Como montar o ambiente, e o quão seguro ele é, depende muito do desenvolvedor.

O PHP é uma linguagem de programação vocacionada para o desenvolvimento de aplicações orientadas para a www. Existem 2 tipos de linguagens web:

- Baseadas no cliente (browser)
 - (JavaScript, ActiveX, Java Applets, etc...)
- Baseadas no servidor.
 - (PHP, ASP, etc...)

Quando o PHP interpreta um arquivo ele procura pelas tags de abertura e fechamento, **<?php e ?>**, que dizem ao PHP para iniciar ou parar a interpretação do código entre elas. A interpretação desta maneira, permite o PHP ser incluído em vários tipos de documentos, pois tudo que está fora destas tags é ignorado pelo interpretador do PHP.

¹ Um servidor de aplicação é simplesmente um servidor, no sentido da arquitetura cliente-servidor: um processo que atende um ou mais aplicativos clientes que lhe enviam requisições. Coloque para rodar um processo que abre uma porta para atender conexões TCP ou UDP, e você tem um servidor de aplicação.

² O shell é a ligação entre o usuário e o sistema. Na computação, um shell é uma interface de usuário para acessar os serviços de um sistema operacional.

Exemplo:

```
<?php
echo "Hello world";

// ... mais código

echo "última instrução";

// o script termina aqui, sem tag de fechamento PHP
```

Vale a pena ressaltar, que os exemplos de códigos foram baseados na versão 7.0 ou posterior do PHP.

Tudo o que estiver fora das tags PHP é ignorado pelo interpretador, o que permite arquivos PHP de conteúdo misto. Permite que o PHP seja incluído dentro de documentos HTML, para, por exemplo, a criação de templates.

```
<p>Isto vai ser ignorado pelo PHP e exibido pelo navegador.</p>
<?php echo 'Enquanto isto vai ser interpretado.'; ?>
<p>Isto também vai ser ignorado pelo PHP e exibido pelo navegador
.</p>
```

Isso funcionará porque quando o interpretador do PHP encontra a tag de fechamento (?>), ele simplesmente começa a repassar qualquer coisa que encontre (exceto um fim de linha imediato), até que ele encontre outra tag de abertura a não ser que esteja no meio de uma instrução condicional, onde então o interpretador vai determinar o resultado da condicional e assim decidir qual caminho tomar. Veja no próximo exemplo.

Utilizando estruturas avançadas

```
<?php if ($expression == true): ?>
    Isto irá aparecer se a expressão for verdadeira.
<?php else: ?>
    Senão isso que aparecerá.
<?php endif; ?>
```

O PHP 7 removeu o suporte a tags ASP e <script language="php">. Assim, é recomendado utilizar apenas <?php ?> e <?= ?> ao se escrever códigos PHP com maior compatibilidade.

Exemplo 1.

```
<?php echo 'se você quer servir documentos XHTML ou XML,
        escreva assim'; ?>
```

Exemplo 2.

Você pode utilizar também a tag curta de echo para `<?= 'imprimir isso' ?>`.

Ela sempre está disponível do PHP 5.4.0 em diante, e é equivalente a

```
<?php echo 'imprimir isso' ?>.
```

Exemplo 3.

```
<? echo 'esse código entre tags curtas somente funcionará '.
    'se short_open_tag estiver habilitado'; ?>
```

Exemplo 4.

```
<script language="php">
    echo 'alguns editores (como o FrontPage) não
        suportam processar instruções com tags assim';
</script>
Esta sintaxe foi removida no PHP 7.0.0.
```

Exemplo 5.

```
<% echo 'Você também pode utilizar tags no estilo ASP'; %>
<%= $variable; %> é um atalho para <% echo $variable; %>
```

Tags curtas (exemplo três) estão disponíveis apenas quando são **habilitadas pela configuração short_open_tag no arquivo php.ini**, ou se o PHP foi configurado com a opção `--enable-short-tags`.

Tags ASP (exemplo cinco) estão disponíveis somente quando habilitadas através da diretiva de configuração `asp_tags` no arquivo `php.ini`, e **foram removidas no PHP 7.0.0**.

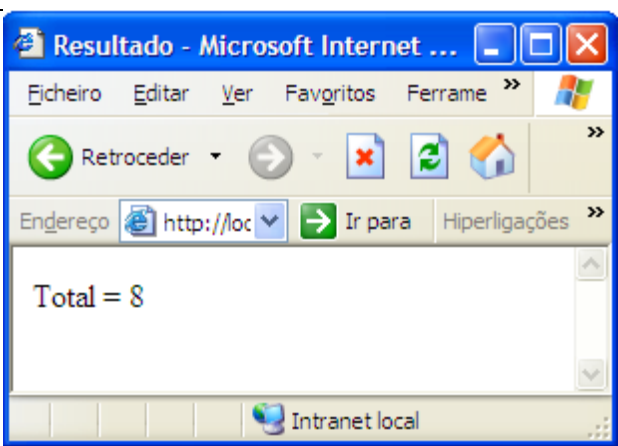
Como no C ou Perl, o PHP requer que as instruções sejam **terminadas com um ponto-e-vírgula ao final de cada comando**. A tag de fechamento de um bloco de código PHP automaticamente implica em um ponto-e-vírgula; você não precisa ter um ponto-e-vírgula terminando a última linha de um bloco PHP. A tag de fechamento do bloco irá incluir uma nova linha logo após, se estiver presente. Veja o próximo exemplo.

```
<?php
    echo 'teste 1';
?>

<?php echo 'teste 2' ?>

<?php echo 'Foi omitida a última tag de fechamento';
```

Como pode ser visto, o código PHP é embebido no código HTML dando origem a um *script* contendo instruções específicas. O servidor lê o código PHP e interpreta os comandos, veja o próximo exemplo.

<pre><?php /*php_1.php */?> <html> <head> <title> Resultado </title> </head> <body> <?php \$X=5; \$Y=3; \$Z=\$X+\$Y; echo "Total "; echo "= \$Z"; ?> </body> </html></pre>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

2. O que é preciso para executar o PHP?

- Um cliente(browser)
- Um servidor Web (ex.: Apache)
- Um servidor de bases de dados (ex.: MySQL)
- Uma ferramenta que permita ligar tudo isto (ex.: PHP)

A figura 1 ilustra o funcionamento de uma aplicação PHP na web.

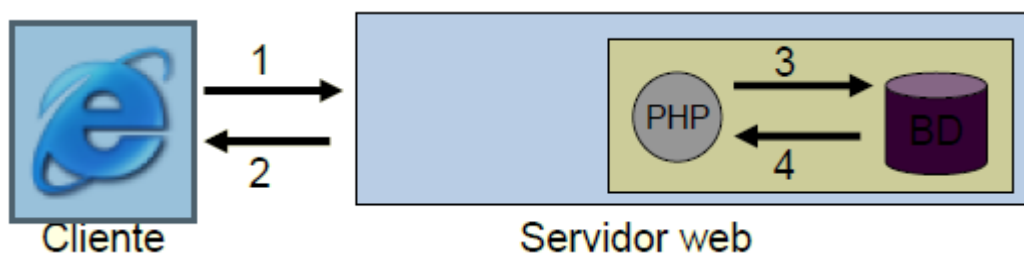


Figura 1 – Esquema de funcionamento de uma aplicação PHP

Onde,

- 1 -O cliente solicita a página ao servidor web. Este, verificando que existe um script PHP, chama o módulo PHP para efetuar o processamento.
- 2 -Se no script existirem comandos que abrem uma ligação a uma base de dados, o PHP trata de efetuar essa ligação.
- 3 -Os dados pretendidos são então extraídos da base de dados, sendo enviados pelo PHP ao servidor web em formato HTML.
- 4 -O servidor web envia os dados ao cliente, sendo estes visualizados numa página HTML devidamente formatada.

No curso vamos trabalhar a linguagem PHP com o banco de dados MySQL. Como o PHP só roda no lado do servidor, vamos então ter que instalar um servidor na máquina, então vamos trabalhar com o servidor http chamado Apache³, um servidor web livre criado em 1995. É disponibilizado em versões para os sistemas operacionais Windows, Novell, OS/2 e outros do padrão POSIX IEEE 1003 (Unix, Linux, FreeBSD, etc.).

A utilização do banco de dados MySQL, é justificada pelos seguintes motivos:

- Baixa exigência de recursos de hardware;
- Simplicidade de administração
- Rapidez
- Foi otimizado para as aplicações típicas da Web, onde ocorrem mais consultas do que atualizações.
- O MySQL é um produto regido pela licença GPL ([General Public License](#))⁴ portanto é [Open Source](#) e não gratuito.

³ Apache Software Foundation trabalha em várias plataformas de S.O é software livre e está na versão 2.4.43 desde 1 de abril de 2020.

⁴ Não é necessário uma licença para incluir o código cliente em programas comerciais

3. Porque usar PHP, Apache e MySQL?

- Porque fornecem tudo o que necessitamos para construir aplicações e web sites sofisticados, e de maneira mais simples do que outras alternativas.
- Principalmente porque são **software livre**!

Preciso utilizar os três em conjunto?

- Não, esta é apenas uma combinação popular na Internet. Outras alternativas seriam:
 - Apache com Perl, Python, ASP, Java, ...
 - PHP com IIS, PWS, Xitami, ...
 - PHP com PostgreSQL, Interbase, Oracle, DB2 ...
 - MySQL com Perl, Python, ASP, Java, ...

Para o desenvolvimento do curso, vamos utilizar o software XAMPP. O XAMPP é um servidor Web multiplataforma constituído por um servidor HTTP Apache. As iniciais do XAMPP são detalhadas a seguir:

X-Sistema operativo (atualmente suportado por windows, Linux, SunSolaris e MacOS.

A-Apache

M-MySQL

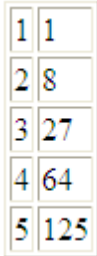
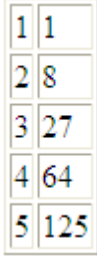
P-PHP

P-PERL

Existe outros softwares similares como:

- **wamp**-Windows, Apache, MySQL e PHP
- **lamp** -Linux, Apache, MySQL e PHP

Mais dois exemplos:

<p>Exemplo 1</p> <pre><table border="1"> <?php for(\$i=1; \$i<6; \$i++) { echo '<tr>'; \$q=\$i*\$i*\$i; echo "<td>\$i </td>"; echo "<td>\$q </td>"; echo '</tr>'; } ?> </table></pre>	<p>Resultado</p> 
<p>Exemplo 2</p> <pre><?php echo '<table border="1">'; for(\$i=1; \$i<6; \$i++) { echo '<tr>'; \$q=\$i*\$i*\$i; echo "<td>\$i </td>"; echo "<td>\$q </td>"; echo '</tr>'; } echo '</table>'; ?></pre>	<p>Resultado</p> 

Repare que a saída é a mesma, mas o primeiro exemplo, você tem a tag <TABLE> e </TABLE> sendo interpretado fora do PHP e no segundo exemplo, todo o código é interpretado pelo PHP. Nesses dois exemplos, fica claro que é possível escrever uma página HTML a partir do PHP, possibilitando assim a criação de páginas dinâmicas, isto é, com conteúdo e formas variáveis, por exemplo.

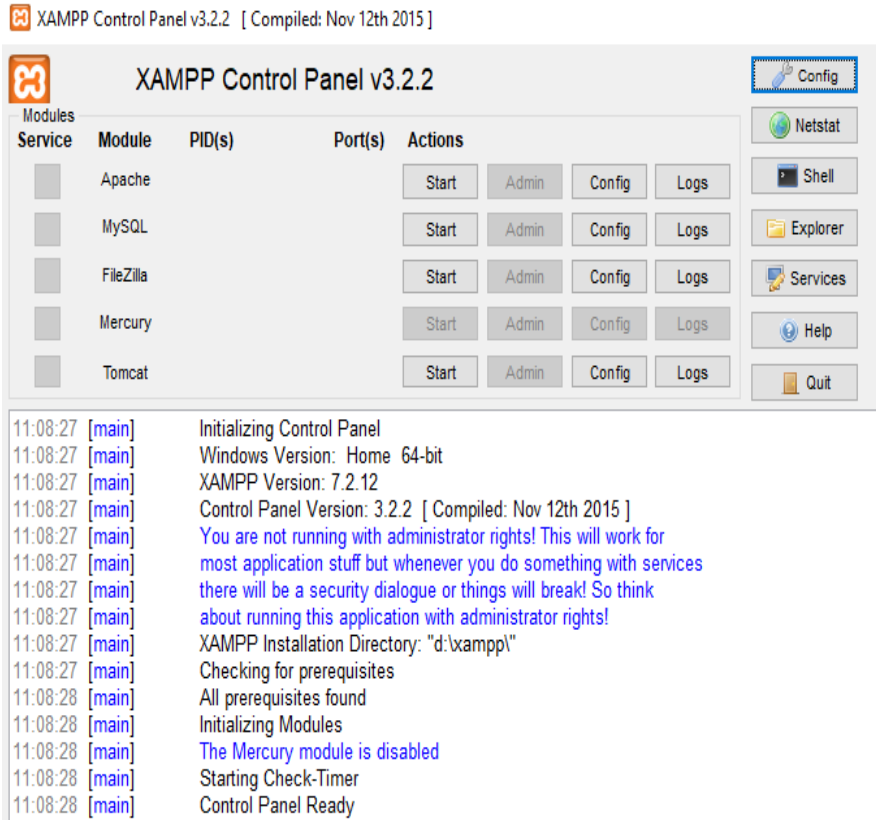
Como exercício, **instale o XAMPP** na máquina (que tenha acesso para uso) e recrie todos os exemplos vistos.

Observações:

Uma vez instalado na máquina, será criado um **diretório na raiz chamado XAMPP**, esse diretório é composto de vários subdiretórios, mas de início, se “preocupem” com o

subdiretório chamado htdocs, vai ser nesse diretório que deve ficar as aplicações desenvolvidas. O quadro 1 mostrará todo o passo a passo para rodar uma aplicação PHP sem banco de dados.

Quadro 1 – Passo a Passo para rodar uma aplicação PHP

Passo 1	<p>Baixar o XAMPP</p> <p>https://www.apachefriends.org/pt_br/download.html</p>
Passo 2	<p>Instalar</p>
Passo 3	<p>Colocar o servidor Apache no Ar. Neste momento a sua máquina será servidora da aplicação cliente que será executada, isto é, quando a aplicação for executada no browse (cliente) uma requisição será feita ao servidor Apache. Então pode –se dizer que a máquina ora é cliente, ora é servidor. Veja a figura 2.</p>  <p>11:08:27 [main] Initializing Control Panel 11:08:27 [main] Windows Version: Home 64-bit 11:08:27 [main] XAMPP Version: 7.2.12 11:08:27 [main] Control Panel Version: 3.2.2 [Compiled: Nov 12th 2015] 11:08:27 [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights! 11:08:27 [main] XAMPP Installation Directory: "d:\xampp" 11:08:27 [main] Checking for prerequisites 11:08:28 [main] All prerequisites found 11:08:28 [main] Initializing Modules 11:08:28 [main] The Mercury module is disabled 11:08:28 [main] Starting Check-Timer 11:08:28 [main] Control Panel Ready</p> <p>Figura 2 – Programa XAMPP Control Panel (Encontrado no menu inicial do S.O)</p> <p>Para colocar o servidor no ar, clique no botão start do Apache, tudo ocorrendo como esperado, o fundo da palavra Apache ficará verde.</p>

Passo 4

Editar o programa e copiar para o diretório `c:/xampp/htdocs`. Esse subdiretório será o diretório do **servidor**, isso quer dizer que, para executar qualquer programa em PHP, o mesmo deve estar nesse diretório. Para uma melhor organização, recomenda-se criar subdiretórios dentro do **htdocs**. A figura 3 mostra um exemplo de uma aplicação rodando no servidor.

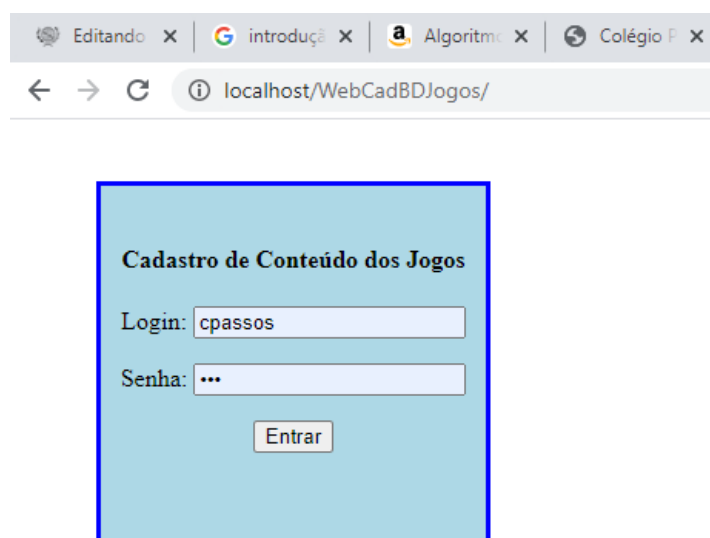


Figura 3 – Exemplo de uma aplicação rodando no servidor

Repare neste exemplo, o link <http://localhost/WebCadBDJogos/>, o nome localhost é o endereço da máquina, seria como se fosse o link do colégio cp2.g12.br. Já o WebCadBDJogos é o diretório, o nome da aplicação foi omitido, porque o arquivo inicial a ser executado chama-se index.php e por padrão se não escreverem o nome do arquivo a ser executado, o Apache procura pelo arquivo index.html e caso não exista procura pelo arquivo index.php.

Lembre-se o Apache tem que está no ar, o arquivo no subdiretório htdocs e no browser usar o localhost.