

DESENVOLVIMENTO DE SISTEMAS PHP



Mauricio Saraiva
Jeanine Barreto

Revisão técnica:

Jeferson Faleiro Leon

*Especialista em Formação Pedagógica
Graduado em Desenvolvimento de Sistemas*



S243d Saraiva, Maurício de Oliveira.

Desenvolvimento de sistemas com PHP [recurso eletrônico] / Maurício de Oliveira Saraiva; Jeanine dos Santos Barreto; [revisão técnica: Jeferson Faleiro Leon]. – Porto Alegre : SAGAH, 2018.

ISBN 978-85-9502-322-2

1. Computação. 2. PHP (Linguagem de programação de computador). I. Barreto, Jeanine dos Santos. II. Título.

CDU 004.439PHP

Linguagem PHP – Introdução

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar o servidor de aplicação Apache XAMPP.
- Reconhecer variáveis, constantes, `var _ dump` e operações do PHP.
- Aplicar o comando `phpinfo` da linguagem PHP.

Introdução

A linguagem PHP possibilita a criação de programas confiáveis e de alta complexidade, capazes de satisfazer a qualquer tipo de tarefa envolvida no desenvolvimento de um sistema. O PHP, assim como o Perl, podem ser adquiridos por meio do servidor XAMPP, que forma um pacote de aplicativos que contém, além dessas linguagens de programação, o servidor web Apache e o banco de dados MySQL.

O PHP é uma linguagem completa que oferece ao programador a possibilidade de criar suas próprias variáveis e constantes, utilizar as variáveis e constantes predefinidas ao longo do código e, também, utilizar uma série de funções, como o `var _ dump` e o `phpinfo`, que auxiliam na depuração do código e na identificação do estado atual da instalação do PHP.

Neste capítulo, você vai estudar sobre o servidor de aplicação Apache, as variáveis, constantes e operações do PHP e a aplicação do comando `phpinfo` da linguagem PHP.

Servidor de aplicação Apache XAMPP

O XAMPP é um servidor desenvolvido pela Apache que roda de forma independente de plataforma. A ideia inicial para a criação do XAMPP era

ser usado como uma ferramenta de desenvolvimento que possibilitasse aos programadores testar o seu trabalho sem a necessidade de acesso à internet.

Geralmente, a instalação de um servidor web Apache não é uma tarefa fácil, principalmente se o objetivo for instalar também um banco de dados e linguagens de script. Já o XAMPP se mostrou uma ferramenta de fácil instalação, que possibilitou aos desenvolvedores o trabalho utilizando Apache (APACHE FRIENDS, c2018).

O XAMPP envolve o servidor web Apache, os interpretadores para as linguagens de script PHP e Perl, e o banco de dados MySQL. Logo, o nome XAMPP é resultado da abreviação de:

- X: significa que funciona em diferentes plataformas e sistemas operacionais.
- A: servidor web Apache.
- M: banco de dados MySQL, que será substituído no pacote pelo MariaDB.
- P: linguagem de script PHP.
- P: linguagem de script Perl.



Fique atento

O MariaDB é um banco de dados baseado no MySQL. Seus desenvolvedores mantêm um foco especial na segurança, aspecto muito importante na atualidade.

Ele possui todas as características de segurança do MySQL e, a cada nova versão do MariaDB, os desenvolvedores incluem os elementos de segurança anteriores e as suas melhorias, se existirem. Quando são descobertos problemas de segurança, são preparadas novas versões do MariaDB, que são rapidamente distribuídas para evitar eventuais problemas com seus usuários.

O MariaDB deverá funcionar exatamente como o MySQL, todos os comandos, interfaces, bibliotecas e API próprias do MySQL existirão também no MariaDB, por isso não há necessidade de converter dados do MySQL para que possam ser exibidos no MariaDB.

O MariaDB fará parte do pacote do XAMPP, em substituição ao banco de dados MySQL.

O XAMPP é formado por um conjunto de vários softwares livres, de código aberto e gratuito, que está liberado para ser copiado. A licença do XAMPP é do tipo GNU GPL, ou simplesmente General Public License (GPL),

o que significa que o software é livre e que a licença é pública. Atualmente, o XAMPP funciona em ambiente operacional Microsoft Windows, Linux, Solaris e MacOS X.

Instalação do XAMPP

Para instalar o XAMPP, é preciso primeiro escolher um site para efetuar o download. Recomenda-se o site do Apache Friends (c2018), pois contém informações completas sobre o XAMPP, traduzidas em vários idiomas e conta com links para vários fóruns e comunidades XAMPP, nos quais é possível tirar dúvidas e publicar procedimentos e maneiras que o programador tenha descoberto para executar suas atividades.

Na página inicial do site é possível escolher para qual sistema operacional deverá ser feito o download do XAMPP, como você pode observar na Figura 1.

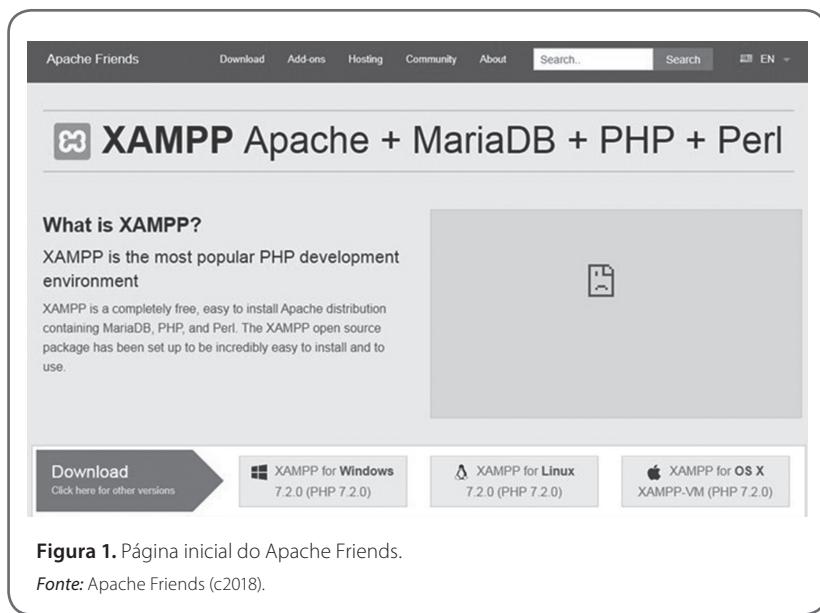
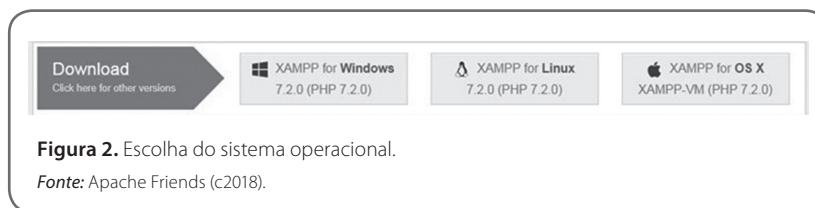


Figura 1. Página inicial do Apache Friends.

Fonte: Apache Friends (c2018).

Na Figura 2, você pode observar o detalhe de seleção para instalação no sistema operacional Microsoft Windows, em que você deve clicar no botão XAMPP FOR WINDOWS.



Em seguida, será apresentada uma janela solicitando o local para a gravação do arquivo de instalação do XAMPP, como demonstra a Figura 3. Basta selecionar a pasta de destino e clicar no botão Salvar.

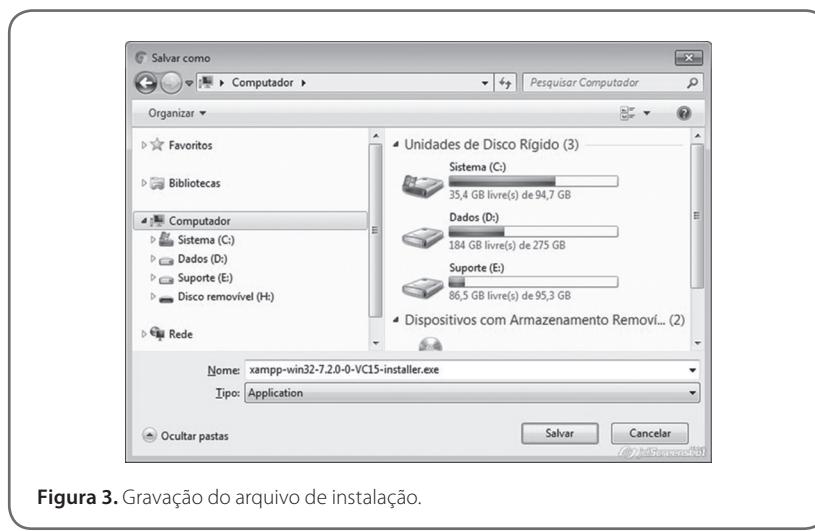


Figura 3. Gravação do arquivo de instalação.

Finalizado o download, basta encontrar o arquivo no local onde foi salvo (ver Figura 4), e executá-lo com duplo clique.



Depois de selecionado o arquivo, será preciso clicar em Executar, como demonstrado na Figura 5.

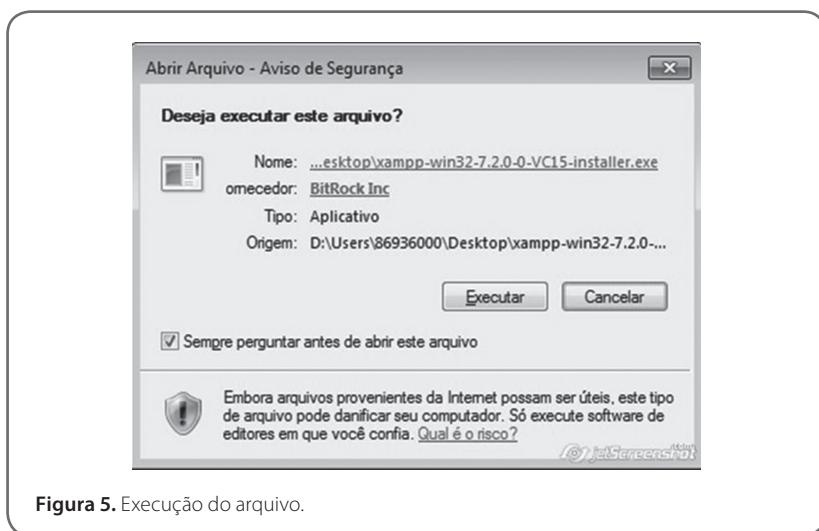


Figura 5. Execução do arquivo.

Na próxima janela, você deve clicar em Next, conforme apresentado na Figura 6.



Figura 6. Next.

Na janela seguinte, será preciso escolher os componentes do XAMPP que deverão ser instalados, como você pode observar na Figura 7. Caso deseje uma instalação padrão, basta clicar em Next.

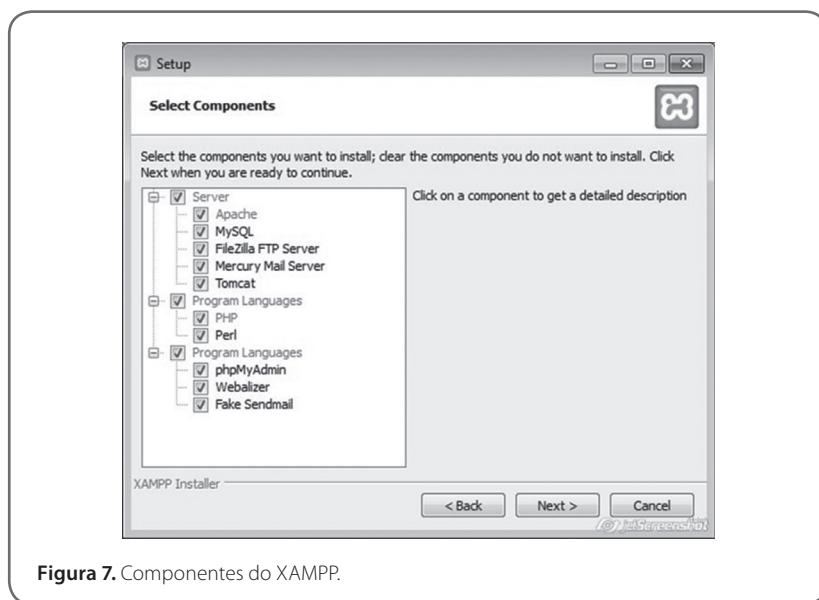


Figura 7. Componentes do XAMPP.

A seguir, você deverá escolher a pasta na qual o XAMPP será instalado (ver Figura 8). Caso deseje que ele seja instalado no local padrão, basta clicar em Next.



Figura 8. Local de instalação do XAMPP.

A partir daí, os componentes do XAMPP serão instalados (ver Figura 9), conforme tenham sido escolhidos na janela anterior.



Figura 9. Instalação do XAMPP.

Será exibida uma janela que indica o término da instalação do XAMPP, como você pode ver na Figura 10.



Figura 10. Finalização da instalação de XAMPP.

O XAMPP possui um painel de controle, apresentado na Figura 11, no qual é possível iniciar, parar, gerenciar, configurar e visualizar logs dos servidores instalados, de forma individual. É sempre interessante visualizar o estado atual do painel de controle durante o desenvolvimento do código do programa.

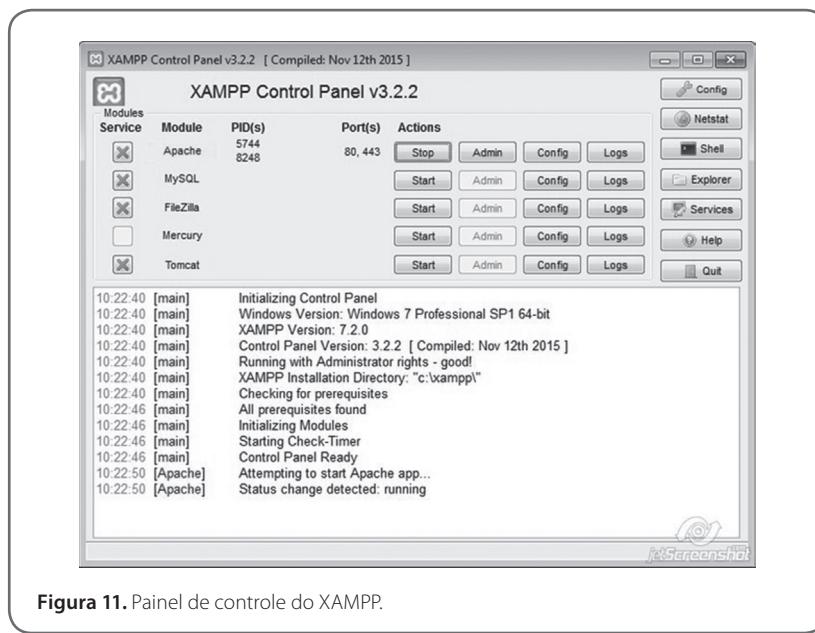


Figura 11. Painel de controle do XAMPP.

Variáveis, constantes, var _ dump e operações do PHP

A linguagem PHP é muito completa e fácil de ser utilizada, bastando que os arquivos com os códigos-fonte dos programas sejam armazenados com a extensão .PHP, em uma pasta conhecida pelo XAMPP, para que possam ser executados.

Caso o XAMPP tenha sido instalado na pasta padrão sugerida no setup (c:\xampp), os arquivos com os códigos dos programas deverão ficar dentro da pasta c:\xampp\htdocs.

Para executar um programa em PHP, você deve, por exemplo, criar um arquivo chamado “teste.php” e colocá-lo em uma subpasta qualquer (nesse caso, foi utilizado c:\xampp\htdocs\ua4).

O próximo passo será editar esse arquivo com um editor de textos ou um IDE do PHP. O conteúdo desse arquivo deve seguir a sintaxe da linguagem PHP, conforme o seguinte exemplo:

```
1 <?php  
2 echo "Hello World!";  
3 ?>
```

Para executar esse pequeno programa, você deve abrir um navegador web e digitar “<http://localhost/ua4/teste.php>” na barra de endereços. O resultado deverá ser conforme a representação da Figura 12.

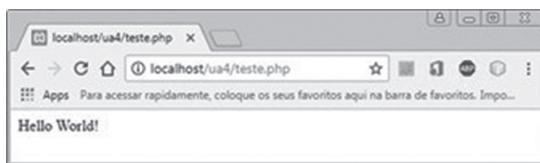


Figura 12. Exemplo de execução de programa em PHP.



Saiba mais

Um Integrated Development Environment (IDE), ou ambiente de desenvolvimento integrado, é um programa de apoio ao desenvolvimento de software que visa agilizar esse processo.

Normalmente, os IDE contêm:

- editor;
- compilador;
- depurador;
- modelador;
- gerador de código;
- distribuidor;
- testador automático.

Para conhecer mais sobre IDE do PHP, acesse o link a seguir (SMITH, 2016).

<https://goo.gl/ywiBDB>



Saiba mais

O termo **localhost** se refere à localização do sistema que está sendo usado. É um dispositivo loopback ao qual é atribuído o endereço IP: 127.0.0.1 no IPv4 ou ::1 no IPv6. Pode ser usado por aplicações TCP/IP para testar a comunicação consigo mesmas.

Variáveis

O PHP é uma linguagem de programação de tipagem fraca, o que significa que é o PHP que se encarrega de alterar os tipos de dados da variável quando for necessário, conforme a operação que estiver sendo realizada no seu valor. Isso torna a tipagem dinâmica, pois não há necessidade de o programador informar, de forma fixa, se o tipo de dados de cada variável é *string*, inteiro, ponto flutuante, objeto, entre outros (PHP, c2001-2018e).

O código PHP a seguir demonstra a tipagem fraca, típica do PHP: primeiro a variável VAR recebe o valor “texto” e, depois, recebe o valor “123”, acompanhe:

```
1 <?php
2 $var = 'texto';
3 echo "Agora o tipo de dados é: " . gettype($var);
4 echo "<br>" ;
5 $var = 123;
6 echo "Agora o tipo de dados é: " . gettype($var);
7 ?>
```

O resultado da execução do código será o seguinte:

```
Agora o tipo de dados é: string
Agora o tipo de dados é: integer
```

Todas as variáveis da linguagem PHP são identificadas por um símbolo de dólar (\$), que precede um nome identificador. O nome identificador deve seguir, obrigatoriamente, às seguintes regras (PHP, c2001-2018e):

- Conter somente letras maiúsculas ou minúsculas (a – z, A - Z), números (1 – 9) e sublinhado (_).
- Iniciar com uma letra ou sublinhado.

- Não conter espaços, por isso, quando um nome tiver mais de uma palavra, elas devem ser separadas por sublinhado (_).
- O nome das variáveis é case-sensitive, o que significa que uma variável nomeada como \$var é diferente de outra com o nome \$VAR.

Apesar da nomenclatura das variáveis precisar seguir essas regras, quando o nome da variável for identificado entre chaves {}, o PHP fará a identificação do que está dentro delas como uma *string* que contém o real nome da variável. No exemplo a seguir, a variável recebe o nome 789, e o valor, abc:

```
1 <?php
2 ${'789'} = 'abc';
3 echo ${'789'};
4 ?>
```

Além das variáveis definidas pelo programador, o PHP oferece variáveis nativas, que são chamadas de superglobais, pois estão disponíveis em qualquer local do código do programa, sem a necessidade de fazer sua inicialização ou definição. As variáveis superglobais do PHP são apresentadas a seguir (PHP, c2001-2018e).

- **\$GLOBALS:** contém todas as variáveis que estão disponíveis no escopo global, na forma de um array que contém as referências das variáveis que são chaves do array.
- **\$_SERVER:** contém informações do servidor e do ambiente de execução.
- **\$_GET:** contém um array associativo de variáveis passadas para o script atual via o método HTTP GET. O que for enviado pelo formulário será mostrado na barra de endereços do navegador.
- **\$_POST:** contém um array associativo de variáveis passados para o script atual via método HTTP POST. O que for enviado pelo formulário não será mostrado na barra de endereço do navegador.
- **\$_FILES:** contém um array associativo de itens com informações relativas aos arquivos enviados via HTTP, que é enviado por meio do script atual pelo método HTTP POST.
- **\$_COOKIE:** contém um array associativo de variáveis passadas para o script atual via HTTP COOKIES.
- **\$_SESSION:** contém um array associativo com variáveis de sessão disponíveis para o atual script.

- **`$_REQUEST`**: contém um array associativo com as variáveis de `$_GET`, `$_POST` e `$_COOKIE`.
- **`$_ENV`**: contém um array associativo com variáveis passadas para o script atual via o método do ambiente. Na internet, equivale ao `$_SERVER`.

Constantes

As constantes são diferentes das variáveis porque conservam seu valor de inicialização durante toda a execução do programa, podendo ser acessadas de qualquer lugar do código. Elas seguem as mesmas regras de nomenclatura das variáveis, mas não possuem o símbolo dólar (\$) precedendo seu identificador. Para uma fácil diferenciação dos nomes das constantes e das variáveis, uma boa prática é nomear as constantes com letras maiúsculas (PHP, c2001-2018a).

A sua definição pode ser feita com o uso da função `define($identificador,$valor)`, na qual são passados dois parâmetros: o nome da constante e o seu valor, como você pode ver a seguir:

```
1 <?php
2 define('SAUDACAO', 'Hello World');
3 echo SAUDACAO;
4 ?>
```

Outra maneira de definir uma constante, a partir da versão 5.3 do PHP, é por meio da expressão `const`. Antes dessa versão, a `const` só podia ser utilizada em escopo de classe, mas atualmente pode ser usada em escopo global.

```
1 <?php
2 const SAUDACAO = 'Hello World';
3 echo SAUDACAO;
4 ?>
```

O resultado dos dois códigos será o seguinte:

Hello World

Além das constantes definidas pelo programador, o PHP oferece constantes predefinidas, que são chamadas de mágicas, pois seu valor se modifica, dependendo do local do código em que ela estiver sendo utilizada. As constantes mágicas são resolvidas em tempo de compilação, ao passo que as demais constantes são resolvidas em tempo de execução. Veja as constantes mágicas do PHP a seguir (PHP, c2001-2018a).

- **`__LINE__`**: retorna o número da linha corrente do código.
- **`__FILE__`**: retorna o caminho completo e o nome do arquivo. Se for utilizado dentro de um `include`, o nome do arquivo incluído também será retornado.
- **`__DIR__`**: retorna o diretório do arquivo. Se for utilizado dentro de um `include`, o diretório do arquivo incluído também será retornado. O nome do diretório não mostrará barra no final, a não ser que seja o diretório raiz. Tem o mesmo resultado que utilizar `dirname(__FILE__)`.
- **`__FUNCTION__`**: retorna o nome da função.
- **`__CLASS__`**: retorna o nome da classe, que inclui o namespace em que foi feita a sua declaração. A partir da versão 5.4 do PHP, essa constante mágica também funciona em traits e, quando utilizada em um método trait, a constante mágica `__CLASS__` retorna o nome da classe que está utilizando a trait.
- **`__TRAIT__`**: retorna o nome do trait, incluindo o namespace em que foi declarado.
- **`__METHOD__`**: retorna o nome do método da classe.
- **`__NAMESPACE__`**: retorna o nome do namespace corrente.

Função `var_dump()`

A função `var_dump` do PHP é muito utilizada para a depuração do código de programação, pois mostra o conteúdo de uma variável, acompanhe:

- se for um `array`, ela exibe todo o array de forma estruturada;
- se for uma variável simples, ela exibe o seu valor;
- se for um objeto, ela exibe todos os campos.

O retorno da função `var_dump` é uma representação estruturada sobre um ou mais elementos, incluindo o tipo de dados e o valor. Sua sintaxe é: `var_dump ($expressao1, $expressao2, ...)`, pois o programador pode usar o `var_dump` para obter informações sobre muitas variáveis ao mesmo tempo (PHP, c2001-2018b).

O exemplo abaixo serve para retornar informações sobre o conteúdo de um `array`, veja:

```
1 <?php
2 $a = array (3, 5, array ("aaa", "bbb", "ccc"));
3 var_dump ($a);
4 ?>
```

O resultado da execução do código será o seguinte:

```
1 array(3) {  
2 [0]=> int(3)  
3 [1]=> int(5)  
4 [2]=> array(3) {  
5 [0]=> string(3) "aaa"  
6 [1]=> string(3) "bbb"  
7 [2]=> string(3) "ccc" } }
```

O próximo exemplo serve para retornar informações sobre o conteúdo de uma variável float e outra do tipo boolean:

```
1 <?php  
2 $preco = 100.75;  
3 $barato = false;  
4 var_dump($preco,$barato);  
5 ?>
```

O resultado da execução do código será o seguinte:

```
1 float(100.75)  
2 bool(false)
```

Operações

Um operador é um elemento que recebe o valor de uma ou mais expressões e que retorna um outro valor. Veja, a seguir, quais são os operadores do PHP (PHP, c2001-2018c).

- **Operadores aritméticos:** servem para efetuar operações de adição (+), subtração (-), multiplicação (*), divisão (/), módulo (%) e exponencial (**).
- **Operadores de atribuição:** servem para atribuir o valor da expressão à direita, para o operando da esquerda (=).
- **Operadores bit a bit (bitwise):** servem para avaliar e modificar bits específicos em um tipo de dados inteiro (AND, OR, XOR, NOT, << e >>).
- **Operadores de comparação:** servem para comparar valores iguais (==), idênticos (===), diferentes (!= ou <>), não idênticos (!==), menor que (<), maior que (>), menor ou igual (<=), maior ou igual (>=). Seu retorno é TRUE ou FALSE.

- **Operadores de controle de erro:** o sinal de arroba (@) serve, quando preceder uma expressão em PHP, para ignorar qualquer mensagem de erro gerada pela sua execução.
- **Operadores de incremento/decremento:** servem para incrementar e decrementar números e strings, por meio de pré-incremento (++ antes da variável), pós-incremento (++ depois da variável), pré-decremento (-- antes da variável) e pós-decremento (-- depois da variável). O decremente de NULL não gera efeitos, mas o incremento resulta em 1.
- **Operadores lógicos:** servem para comparar logicamente dois operandos, por meio de e (and ou &&), ou (OR ou ||), ou exclusivo (XOR), e não (!). Seu retorno é TRUE ou FALSE.
- **Operadores de string:** servem para concatenar strings (), concatenando os argumentos à direita e à esquerda, e também para atribuir a concatenação (.=), acrescentando o argumento do lado direito ao argumento do lado esquerdo.
- **Operadores de array:** servem para fazer operações entre arrays, por meio de união (+), igualdade (==), identidade (====), desigualdade (!= ou <>), e não identidade (!==).



Link

A precedência entre os operadores significa a ordem de prioridade entre eles. Como exemplo, pode-se citar a precedência da multiplicação sobre a soma, na expressão $3 + 4 * 7$, em que a resposta é 31, e não 49.

Para conhecer melhor a precedência de operadores, acesse o link que consta no Manual Oficial do PHP (c2001-2018d).



<https://goo.gl/DpQ7hS>

Comando `phpinfo()` da linguagem PHP

O `phpinfo()` é uma função que serve para mostrar informações sobre o estado atual do PHP. O resultado desse comando inclui informações sobre (PHP, c2001-2018b):

- As opções de compilação do PHP.
- As suas extensões.
- A versão do PHP.
- O servidor PHP.
- O ambiente PHP.
- A versão do sistema operacional.
- Os endereços de arquivos.
- Os valores principais e locais das opções de configuração.
- A licença PHP.

Para criar um arquivo com as informações retornadas pelo `phpinfo()`, deve-se seguir os passos a seguir:

- Abrir um editor de texto.
- Digitar o seguinte código:

```
1 <?php  
2 phpinfo();  
3 ?>
```

- Salvar o arquivo com o nome `phpinfo.php`.
- Fazer o upload desse arquivo para o local em que estão hospedados os arquivos PHP.
- Acessar o arquivo pelo endereço no qual foi feito o upload, por exemplo, <http://localhost/UA4/phpinfo.php>.

Normalmente, a função `phpinfo()` é utilizada para que sejam conhecidas as definições de configuração e as variáveis pré-definidas que estão disponíveis no sistema, pois a configuração em cada sistema, de cada programador, tende a ser diferente.

O retorno da função `phpinfo()` pode ser configurado por meio da passagem de uma ou mais constantes, como parâmetros opcionais, conforme segue (PHP, c2001-2018b).

- **INFO_GENERAL:** exibe informações de configuração, localização do `php.ini`, dados de construção, informações sobre o servidor de internet, sobre o sistema operacional, entre outros (ver Figura 13).
- **INFO_CREDITS:** exibe os créditos do PHP.
- **INFO_CONFIGURATION:** exibe os valores locais e principais para as diretivas de configuração do PHP.

- **INFO_MODULES:** exibe os módulos carregados e, também, as configurações de cada um.
- **INFO_ENVIRONMENT:** exibe as informações a respeito das variáveis de ambiente.
- **INFO_VARIABLES:** exibe todas as variáveis pré-definidas de environment, GET, POST, cookies, server, etc.
- **INFO_LICENSE:** exibe informações sobre a licença do PHP (ver Figura 14).
- **INFO_ALL:** exibe todas as informações que qualquer uma das constantes citadas pode exibir. É o valor padrão e, caso não seja passado nenhum parâmetro, será o INFO_ALL que será retornado.

PHP Version 7.2.0	
System	Windows NT MRS64142885 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Nov 28 2017 23:47:42
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cscript /nologo configure.js --enable-snapshot-build --enable-debug-pack --with-pdo-oci=c:\php-snap-build\deps_au\oracle\v8\instantclient_12_1\ sdk\shared --with-oci8-12=c:\php-snap-build\deps_au\oracle\v8\instantclient_12_1\ sdk\shared --enable-object-out-dir=../obj/ --enable-com-dotnet=shared --without-analyzer --with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.TS.VC15
PHP Extension Build	API20170718.TS.VC15
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert*, consumed, dechunk, zlib*, bzip2,*
This program makes use of the Zend Scripting Language Engine: Zend Engine v3.2.0, Copyright (c) 1999-2017 Zend Technologies	
zendengine	

Figura 13. Exemplo de retorno do PHPINFO(INFO_GENERAL).



Figura 14. Exemplo de retorno do PHPINFO(INFO_LICENSE).



Referências

APACHE FRIENDS. XAMPP Apache + MariaDB + PHP + Perl. [S.I.], c2018. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 20 dez. 2017.

PHP. Constantes. [S.I.]: The PHP Group, c2001-2018a. Disponível em: <http://php.net/manual/pt_BR/language.constants.php>. Acesso em: 22 dez. 2017.

PHP. Funções. [S.I.]: The PHP Group, c2001-2018b. Disponível em: <http://php.net/manual/pt_BR/language.functions.php>. Acesso em: 22 dez. 2017.

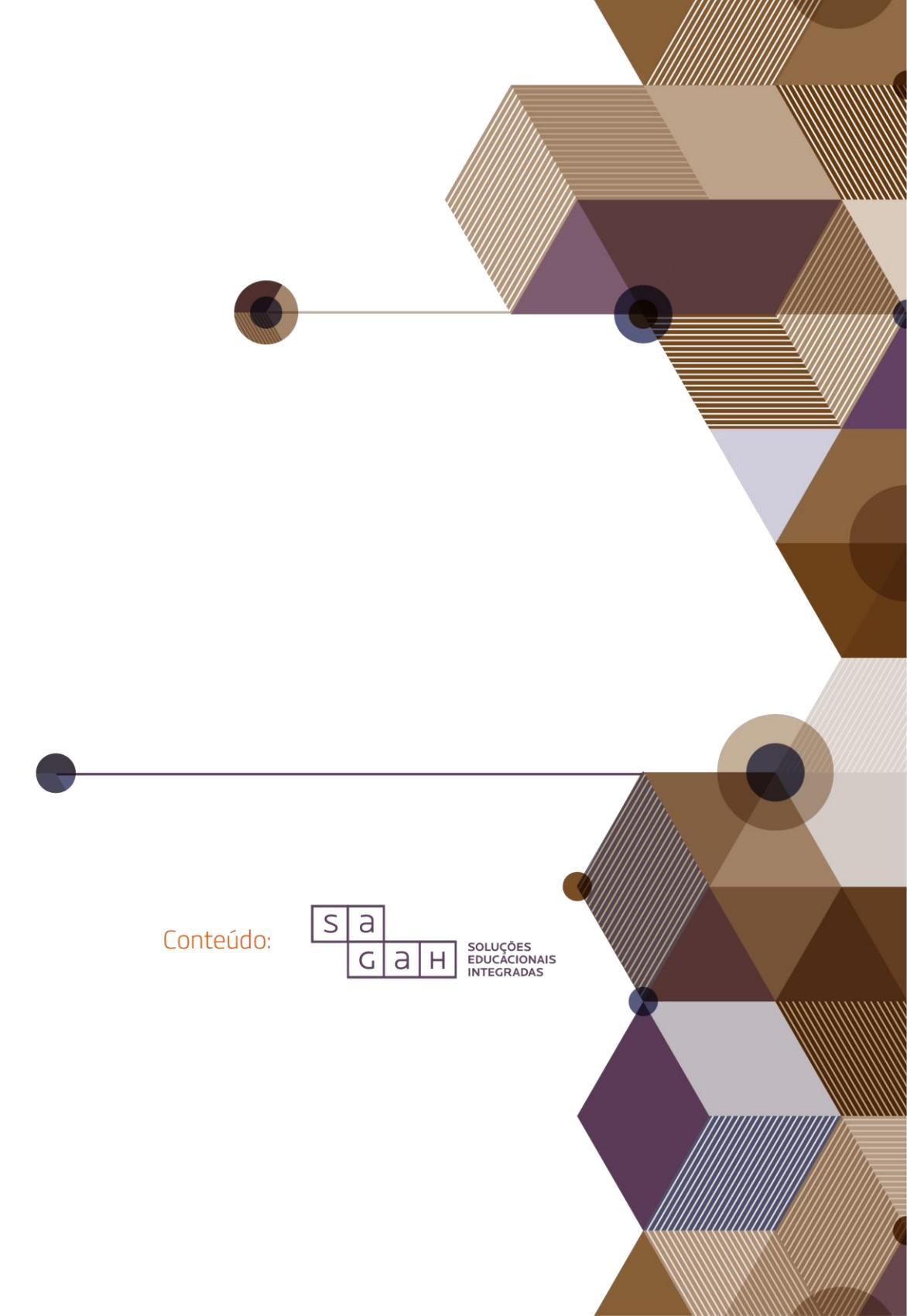
PHP. Operadores. [S.I.]: The PHP Group, c2001-2018c. Disponível em: <http://php.net/manual/pt_BR/language.operators.php>. Acesso em: 22 dez. 2017.

PHP. Precedência de operadores. [S.I.]: The PHP Group, c2001-2018d. Disponível em: <http://php.net/manual/pt_BR/language.operators.precedence.php>. Acesso em: 22 dez. 2017.

PHP. Variáveis. [S.I.]: The PHP Group, c2001-2018e. Disponível em: <http://php.net/manual/pt_BR/language.variables.php>. Acesso em: 22 dez. 2017.

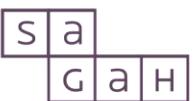
SMITH, D. *Comparação das melhores IDEs de PHP de 2016 e 2017 para Windows, Mac e Linux*. [S.I.]: iMasters, 2016. Disponível em: <<https://imasters.com.br/desenvolvimento/comparacao-das-melhores-ides-de-php-de-2016-e-2017-para-windows-mac-e-linux/?trace=1519021197&source=single>>. Acesso em: 21 jan. 2018.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.



A large, abstract graphic in the background consists of a grid of hexagons in various shades of brown, tan, purple, and blue. Some hexagons have fine diagonal lines running from top-left to bottom-right, while others are solid or have different patterns. Small, dark circular dots are scattered across the white space between the hexagons, some connected by thin lines.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS