

# MÓDULO

# PROGRAMAÇÃO PHP

UNIDADE

## INTRODUÇÃO AO PHP



## ÍNDICE

---

<b>OBJETIVOS.....</b>	<b>3</b>
<b>INTRODUÇÃO.....</b>	<b>4</b>
<b>1. O QUE É O PHP?.....</b>	<b>5</b>
<b>2. HISTÓRIA.....</b>	<b>6</b>
<b>3. APLICAÇÕES DO PHP .....</b>	<b>7</b>
<b>4. FERRAMENTAS NECESSÁRIAS .....</b>	<b>9</b>
<b>5. SERVIDOR WEB.....</b>	<b>10</b>
5.1. CONFIGURAÇÃO DO APACHE.....	11
5.1.1. PARÂMETROS GLOBAIS.....	13
5.1.2. DIRETIVAS OPERACIONAIS.....	13
5.1.3. VIRTUAIS EM APACHE.....	15
<b>6. SINTAXE .....</b>	<b>16</b>
6.1. COMENTÁRIOS EM PHP .....	18
<b>CONCLUSÃO.....</b>	<b>21</b>
<b>AUTOAVALIAÇÃO.....</b>	<b>23</b>
<b>SOLUÇÕES.....</b>	<b>29</b>
<b>PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO.....</b>	<b>30</b>
<b>BIBLIOGRAFIA .....</b>	<b>31</b>



## OBJETIVOS

---

Com esta unidade didática, pretende-se que desenvolva os seguintes objetivos de aprendizagem:

- Descrever o que é PHP.
- Conhecer a história da linguagem.
- Saber o motivo do seu uso tão padronizado.
- Fazer a instalação de um servidor web Apache com PHP e MySQL.
- Reconhecer as possíveis ferramentas para desenvolver páginas web.

## INTRODUÇÃO

---

Nesta unidade, verá o que é o PHP e por que razão é tão amplamente utilizado em todo o mundo, e ainda aprenderá sobre a história desta linguagem de programação web. Abordar-se-á também como instalar um servidor web no computador e as ferramentas necessárias para desenvolver o código. Verá ainda várias ferramentas recomendadas que servem para facilitar o trabalho em PHP.

# 1. O QUE É O PHP?

---

O PHP é uma linguagem de programação chamada "processador de hipertexto". Esta linguagem está embutida diretamente no código HTML e a diferença em relação a linguagens como JavaScript ou VBScript é que o código PHP é executado no servidor e é ele que envia o resultado da execução ao navegador para que este o interprete e o exiba.

Atualmente, é uma das linguagens mais populares entre os programadores web porque é muito fácil de usar, é gratuita (tanto o servidor web quanto a linguagem de programação são de código aberto) e possui inúmeros módulos pré-criados, dada a sua grande popularidade, o que torna a programação muito fácil.

## 2. HISTÓRIA

---

O PHP foi criado em 1994 por Rasmus Lerdorf, que o projetou em Perl, baseando-se num script CGI em C.

Em 1997, dois programadores israelitas reescreveram o analisador de linguagem, criando assim a base para o primeiro PHP (versão 3). Esta versão acabou por se tornar na versão oficial, em 1998–1999, após o código PHP ter sido reescrito.

Em 2000, a nova versão do PHP (PHP4) foi lançada com o novo Zend Engine (mecanismo de processamento de PHP).

Em 2004, foi lançada a versão 5, que apresentava uma nova versão do Zend Engine como um mecanismo de interpretação que trouxe muitas melhorias à linguagem.

No que diz respeito à versão PHP6, esta foi cancelada em 2012, principalmente porque foi decidido utilizar o suporte nativo para caracteres Unicode, o que implicou a implementação de grandes mudanças nas versões 5.2 e posteriores, acabando assim por se descartar a versão anterior e passar-se diretamente para a versão 7.

Em 2015, nasceu o PHP versão 7, que trouxe melhorias de desempenho e incorpora diferenças significativas na utilização de bases de dados.



## 3. APLICAÇÕES DO PHP

---

Com o PHP é possível trabalhar com cookies, processar formulários, criar conteúdo dinâmico (aceder a base de dados), enviar e-mails, etc.

Praticamente tudo o que pode ser feito com um CGI pode ser desenvolvido em PHP.

Uma das vantagens do PHP é a sua compatibilidade para aceder a dados em quase qualquer sistema de armazenamento, desde XML a aplicações de base de dados, como:

- Adabas D.
- dBase.
- Empress.
- FilePro (somente leitura).
- Hyperwave.
- IBM DB2.
- Informix.
- Ingres.
- InterBase.
- FrontBase.
- mSQL.

- MS-SQL direto.
- MySQL.
- ODBC.
- Oracle (OCI7 e OCI8).
- Ovrimos.
- PostgreSQL.
- SQLite.
- Solid.
- Velocis.
- Unix dbm.

O padrão ODBC permite que o PHP se conecte a qualquer base de dados que o suporte.

Além disso, pode ser utilizado em qualquer sistema operativo: Linux, Unix, Windows, Mac-os, etc., e com suporte aos diferentes servidores web: Apache, Nginx, Microsoft IIS, Lighttpd, Litespeed, Zeus, iPlanet, Caudium, etc.

## 4. FERRAMENTAS NECESSÁRIAS

---

O PHP é uma linguagem de texto simples, o que significa que, para programar um website em PHP, pode usar como ferramenta de desenvolvimento qualquer editor de texto simples, como o Notepad ++.

Não obstante, existem ferramentas muito mais poderosas que permitem desenvolver de uma forma mais fácil e intuitiva o código PHP, incluindo, entre outras funções, janelas de visualização, opções de autocompletar, ajuda e marcação de erros.

Seguem-se os três editores de texto mais utilizados:

- Visual Studio Code, da empresa Microsoft. Atualmente, um dos melhores ambientes de programação leves que existe. É gratuito e permite um enorme desenvolvimento, mas, sendo modular, pode tornar-se um pouco lento se se abusar do uso de plug-ins.
- phpDesigner, da empresa MPsoftware. É projetado para edição de PHP, ao contrário do anterior, e facilita muito a navegação no PHP.

Pode descarregá-lo no seguinte link:

<http://www.mpsoftware.dk/phpdesigner.php>.

- SublimeText. É um editor de código-fonte muito leve que suporta várias linguagens e ao qual podem ser adicionadas funcionalidades através de plug-ins.

## 5. SERVIDOR WEB

---

Para testar as atividades práticas que lhe propomos ao longo da formação, ou para poder fazer upload das páginas realizadas, é necessário ter um servidor web. Na Internet existem diversas empresas que oferecem este serviço de alojamento e o disponibilizam a partir de um espaço já configurado para carregar os ficheiros e hospedar a página web em máquinas dedicadas que é possível configurar.

Se não pretender instalar todas as componentes, pode simplesmente instalar um servidor web para esta unidade. Neste caso, é recomendado utilizar o XAMPP ou, em alternativa, o MAMP, que são servidores web simples e gratuitos e que mais tarde, caso pretenda, lhe permitirão fazer ligação com a base de dados.

Poderá fazer o download do XAMPP e do MAMP, respetivamente, nas páginas abaixo:

- <https://www.apachefriends.org/index.html>.
- <https://www.mamp.info/en/downloads/>.

Nesta formação, aconselhamos o uso do XAMPP por incluir já o pacote completo que permite a instalação do Apache (servidor web), MySQL, PHP e um gestor para configurações, chamado phpMyAdmin.

**Vídeo**

Aceda ao Campus Virtual e assista ao vídeo “Introdução ao PHP”, em que mostramos como instalar os programas XAMPP e MAMP.

Em alternativa, para as plataformas Windows existem diversos programas pré-compilados encarregados de fazer a instalação completa neste sistema operativo. São gratuitos e um dos mais conhecidos, pela sua simplicidade de instalação, é o AppServ Open Project que pode encontrar em:

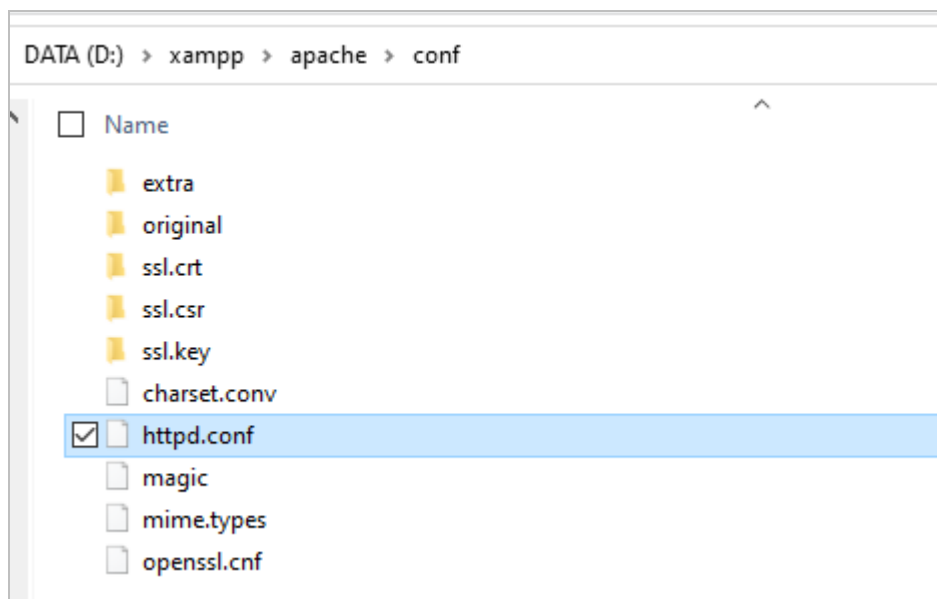
- <http://www.appservnetwork.com/index.php>.

**Vídeo**

Aceda ao Campus Virtual e assista ao vídeo “Instalação do AppServ”.

## 5.1. CONFIGURAÇÃO DO APACHE

A configuração do Apache é realizada a partir do ficheiro `httpd.conf` encontrado na pasta “Conf” dentro da pasta de instalação do Apache. Esta pasta será uma subpasta do servidor web (XAMPP ou MAMP).



Pasta XAMPP.

O ficheiro httpd.conf está dividido em três áreas:

- Parâmetros globais.
- Diretrizes operacionais.
- Hosts virtuais.

Os parâmetros de configuração do ficheiro são encontrados nas secções. As secções mais importantes do ficheiro são:

- **<Directory>**, os parâmetros instalados nesta secção serão aplicados ao diretório especificado e aos seus subdiretórios.
- **<DirectoryMatch>**, o mesmo que o anterior, mas aceita expressões regulares como nome de diretório.
- **<Files>**, parâmetros que dão acesso aos ficheiros por nome.
- **<FilesMatch>**, o mesmo que o anterior, mas o nome aceita expressões regulares.
- **<Local>**, fornece acesso aos ficheiros por URL.
- **<LocationMatch>**, o mesmo que o anterior, mas aceita expressões regulares.

Estas secções podem ser misturadas, por isso deve ter em consideração a ordem de preferência, que é a mesma que definir. Por fim, os comentários no ficheiro são todas as linhas que começam com um cardinal (#) e explicam para que serve cada linha.

Segue-se uma análise das três áreas anteriores em pormenor.

### 5.1.1. PARÂMETROS GLOBAIS

Estes parâmetros não podem ser definidos em nenhuma diretiva e os mais importantes são:

- **ServerRoot**: define o diretório onde o Apache está instalado. Esta diretiva não deve ser alterada, a menos que mova a pasta de instalação.
- **PidFile**: terá um número que identifica o processo quando o servidor Apache inicia; não é modificado.
- **Timeout**: é utilizado para definir tempos de espera para uploads, transmissões, entre outros.
- **KeepAlive**: define se conexões persistentes serão utilizadas, ou seja, se as solicitações de um utilizador serão atendidas com a mesma conexão.
- **Listen**: permite que especifique que porta será utilizada para atender solicitações HTTP por defeito (se não foi modificada na instalação); será utilizada a 80. Também podemos definir que um determinado IP utiliza uma porta, por exemplo:

```
Listen 10.0.0.15:8080
```

- **LoadModule**: utilizado para carregar módulos externos.

### 5.1.2. DIRETIVAS OPERACIONAIS

- **ServerAdmin**: especifica o e-mail do administrador (parâmetro de instalação, que normalmente é utilizado em projetos maiores); este e-mail aparecerá nos erros que o servidor cometer.

- **DocumentRoot:** define a pasta onde o servidor está localizado; todas as solicitações terão esse valor como pasta raiz.
- **ServerName:** nome e porta do servidor; o IP pode ser usado em vez do nome (se houver um problema de DNS).
- **DirectoryIndex:** o valor do ficheiro inicial que é pesquisado ao interpretar uma solicitação; por defeito irá pesquisar "index.html"; se a página inicial se chamasse "home.php", por exemplo, seria neste parâmetro que deveria indicá-lo.
- **AccessFileName:** por padrão é .htaccess e permite configurar o comportamento de cada um dos diretórios individualmente; também é usado para sobrescrever caminhos.
- **HostnameLookups:** se esta diretiva estiver On quando ocorrer um acesso, além de salvar o IP, irá pesquisar o nome, por resolução de nome, e armazená-lo. Ao fazer uma resolução de nome, a velocidade de carregamento diminui, pois é feita uma solicitação para cada resolução de nome.
- **DefaultType:** indica o tipo de dados servidos pelo servidor padrão; após a instalação, este valor indica texto simples.
- **TypesConfig:** indica o nome do ficheiro que determina a lista de tipos MIME permitidos no servidor.
- **ErrorLog:** indica o ficheiro que contém os logs de erros.
- **LogLevel:** especifica através de um valor que tipos de dados de erro serão salvos no log.
- **ServerTokens:** indica as informações que são servidas nos cabeçalhos HTTP.
- **IndexOptions:** esta diretiva controla a aparência da página de listagem de ficheiros de um diretório.
- **AddDefaultCharset:** define a codificação de caracteres dos documentos padrão cujo valor é ISO-8859-1.
- **ErrorDocument:** estabelece o que fazer quando ocorrer um erro. Os valores possíveis são:
  - Obter o texto do erro.
  - Redirecionar para um ficheiro no mesmo diretório.



- ☐ Redirecionar para um ficheiro no servidor.
- ☐ Redirecionar para um ficheiro fora do servidor.
- **CacheSize**: tamanho do cache em kilobytes.

### 5.1.3. VIRTUAIS EM APACHE

Esses tipos de diretivas são importantes, pois permitem criar diretórios virtuais e aceder entre eles.

- **Alias**: permite indicar diretórios virtuais, ou seja, um diretório que não está no caminho e que é indicado ao servidor, mas pretende que seja acessível como se estivesse naquele caminho. Exemplo:

```
Alias /livrarias/traducao "c:/traducao"
```

- **AliasMatch**: igual à diretiva anterior, mas permite o uso de expressões regulares. Exemplo:

```
AliasMatch ^/icons(.*) "c:/icons"
```

## 6. SINTAXE

---

O PHP, por ser uma linguagem do lado do servidor, interpreta o código de uma página antes de a enviar ao navegador. Para isso, interpretará o código-fonte e procurará pelas tags de abertura do código, que em PHP são "<? php" e "?>" (abrir e fechar o código PHP, respetivamente).

Existe a opção de configurar o servidor web para aceitar também a abreviatura "<?" e "? >", mas tal não é recomendado, pois acabará por retardar o tempo de carregamento.

### Exemplo 01

---

Neste primeiro exemplo do PHP, irá criar uma página web que mostrará a frase "Olá, mundo!".

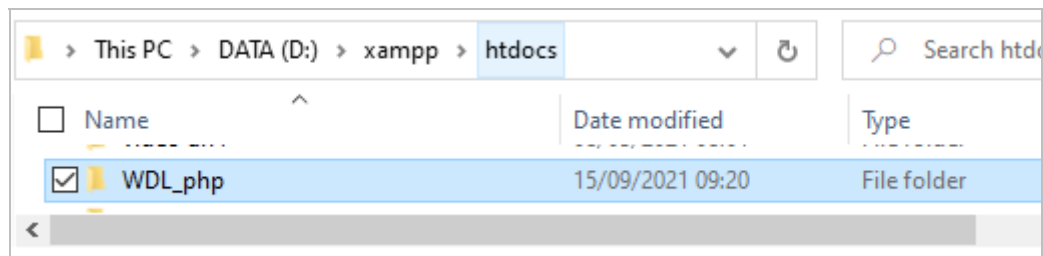
Deverá criar um ficheiro PHP (ou seja, gravado com extensão .php), com a seguinte estrutura:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "Olá Mundo!";
```

```
?>  
  
</body>  
</html>
```

Este ficheiro deverá ser gravado na pasta “htdocs” do seu servidor web. No caso, criou-se uma pasta de trabalho onde serão gravados todos os exemplos da formação e chamada de “WDL\_php”. Dentro desta pasta, gravou-se, então, o ficheiro exemplo1.php.



Pasta de trabalho.

Passamos agora a explicar o código do exemplo.

Foi utilizada uma função própria do PHP, chamada echo, que permite exibir o texto passado como um parâmetro, entre aspas.

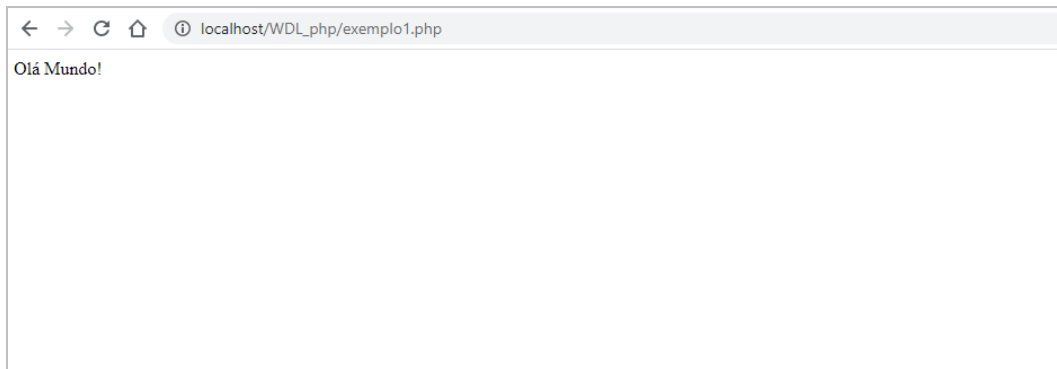
Tal como em outras linguagens de programação, e como pode ver no exemplo, as instruções PHP requerem um ponto e vírgula (;) no final da declaração.

A instrução de fecho do código “?” implica um ponto e vírgula por si só, logo, a linha PHP do exemplo anterior poderia ser escrita da seguinte forma:

```
<?php echo “Olá mundo!” ?>
```

Para visualizar o seu exemplo no browser, deverá escrever o endereço, tendo em conta os nomes que atribuiu à pasta de trabalho e ao ficheiro PHP:

- [http://localhost/WDL\\_php/exemplo1.php](http://localhost/WDL_php/exemplo1.php).



Resultado da visualização.

## 6.1. COMENTÁRIOS EM PHP

Os comentários são uma ferramenta muito importante das linguagens de programação. Graças a eles é possível deixar notas na programação, esclarecer conteúdos, e não só. Para abrir e fechar um comentário utiliza-se `"/*"` e `"*/"`, respetivamente. Todo o conteúdo que estiver entre estes sinais não será interpretado como código, logo não será exibido.

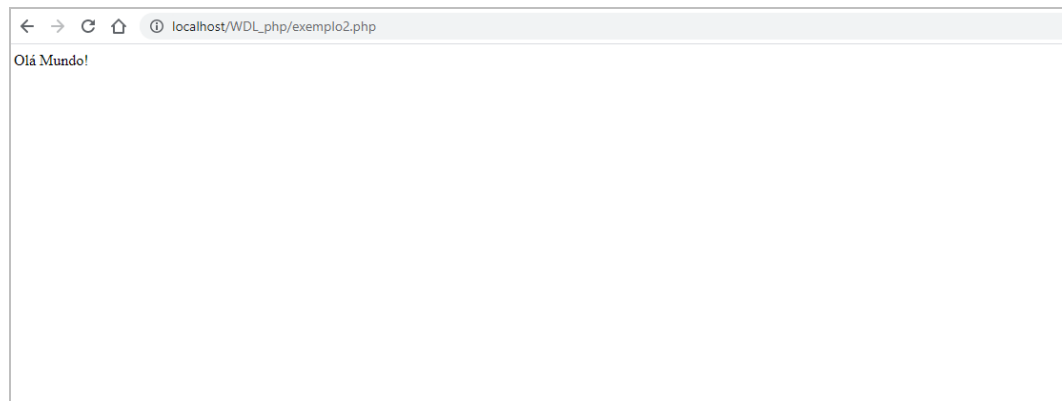
### Exemplo 02

---

Neste exemplo, e aproveitando o exemplo anterior, inseriu-se um comentário: "Este é o meu primeiro php". Como se pode ver, o comentário foi escrito dentro do PHP e entre os sinais `/*` e `*/`. Assim garantimos que o mesmo não será visualizado.

```
<!DOCTYPE html>
<html>
<body>
    <?php
```

```
/* Este é o meu primeiro php */  
echo "Olá Mundo!";  
?  
</body>  
</html>
```



Resultado da visualização.



## CONCLUSÃO

---

O PHP é uma linguagem do lado do servidor, ou seja, é executada no servidor e devolve à web em HTML. Além disso, é uma linguagem gratuita e muito popular com a qual é possível criar qualquer website que se pretender, desde um website de informações até uma loja online. Nesta unidade didática, observou como pode servir-se de um servidor web (Apache) no seu próprio computador para executar testes, fazer desenvolvimentos, etc., além de ter ficado a conhecer ferramentas bastante úteis para poder construir as páginas web de forma mais rápida e fácil.





## AUTOAVALIAÇÃO

---

1. **De que lado está a linguagem PHP?**
  - a) Do lado do cliente.
  - b) Do lado do servidor.
  - c) Da plataforma não web.
  - d) Dentro das bases de dados.
  
2. **Complete a seguinte afirmação: “A linguagem PHP é uma das linguagens...”:**
  - a) ... mais impopulares.
  - b) ... mais velhas.
  - c) ... mais populares.
  - d) ... mais complicadas.
  
3. **Quando é que o PHP foi criado por Rasmus Lerdorf?**
  - a) Em 1997.
  - b) Em 1994.
  - c) Em 2000.
  - d) Em 1980.

- 4. Qual é o mecanismo que interpreta o código PHP desde 2000?**
- a) Zend Engine.
  - b) PHPInterpreter.
  - c) PERL Engine.
  - d) Google.
- 5. Em que ano saiu a versão 6 do PHP?**
- a) Em 2004.
  - b) Em 2007.
  - c) Em 2009.
  - d) Não saiu.
- 6. Qual das seguintes funcionalidades não pode ser feita em PHP?**
- a) Formulários de processo.
  - b) Cookies.
  - c) Enviar e-mails.
  - d) Executar o código no cliente.
- 7. Qual das seguintes bases de dados não é acessível com conexão direta?**
- a) SqlServer.
  - b) MySql.
  - c) Oracle.
  - d) IBM DB2.

**8. Qual das seguintes afirmações é verdadeira?**

- a) O PHP é compatível com qualquer sistema operacional, mas só é compatível com o servidor web Apache.
- b) O PHP é compatível apenas com Linux e Unix, mas é compatível com servidores web Linux, como o Apache.
- c) O PHP é compatível com todos os sistemas operativo e servidores web.
- d) O PHP é compatível com todos os sistemas operativos desde que utilizado o servidor web Apache.

**9. Quais são as tags para abrir e fechar o código PHP?**

- a) `<% %>`.
- b) `<? php ?>`.
- c) `/? ?/`.
- d) `<% php %>`.

**10. Complete a seguinte afirmação: "As instruções PHP requerem ... no final de cada instrução".**

- a) Ponto e vírgula (;).
- b) Chaveta (}).
- c) Parênteses reto (]).
- d) Aspas (").

**11. Que tipo de linguagem é PHP?**

- a) Texto simples.
- b) Pré-compilado.
- c) Que requer um programa de compilação.
- d) Uma linguagem Apache.

**12. Qual é o nome do serviço que algumas empresas oferecem na Internet para poder hospedar as páginas?**

- a) Routing.
- b) Hosting.
- c) Makeing.
- d) Trying.

**13. Qual é a diferença entre o Alias e o AliasMatch?**

- a) Não existe nenhuma diferença.
- b) Não existe nenhum AliasMatch.
- c) O Alias pode utilizar expressões regulares.
- d) O AliasMatch pode utilizar expressões regulares.

**14. No caso de se instalar um servidor Apache no Windows e ter o IIS instalado, o que se deve fazer?**

- a) Alterar o caminho da pasta de instalação padrão.
- b) Alterar a porta padrão.
- c) Alterar o caminho e a porta padrão.
- d) Não necessita de alterar nada.

**15. Qual é o ficheiro de configuração de Apache?**

- a) apache.ini.
- b) config.ini.
- c) server.conf.
- d) httpd.conf.

**16. O que é que o parâmetro global TimeOut permite?**

- a) Definir os tempos de espera de carregamento.
- b) Controlar o tempo até desligar o servidor.
- c) Definir o tempo que passa entre as conexões.
- d) Controlar o tempo que uma página fica hospedada.

**17. Que parâmetro é utilizado para carregar módulos externos?**

- a) ModuleListen.
- b) ModuleLoad.
- c) LoadModule.
- d) Não é possível carregar módulos externos.

**18. O que é que define o parâmetro ServerAdmin?**

- a) Define o e-mail do administrador do sistema.
- b) Define o utilizador de administração do sistema.
- c) Define o caminho do programa do administrador do sistema online.
- d) Define as autorizações do administrador.

**19. Qual é o parâmetro usado para veicular as informações dos cabeçalhos HTTP?**

- a) ServerTokens.
- b) TypesConfig.
- c) HeadHttp.
- d) HTTPHeads.

**20. Qual é o parâmetro usado para criar diretórios virtuais do Apache?**

- a) Directory.
- b) VDirectory.
- c) VirtualDir.
- d) Alias.

## SOLUÇÕES

---

1.	b	2.	c	3.	b	4.	a	5.	d
6.	d	7.	a	8.	c	9.	b	10.	a
11.	a	12.	b	13.	d	14.	b	15.	d
16.	a	17.	c	18.	a	19.	a	20.	d

## PROPOSTAS DE DESENVOLVIMENTO DO ESTUDO

---

Para investigar a história do PHP, pode visitar a seguinte página web:

- <https://pt.wikipedia.org/wiki/PHP>

Visite o seguinte website para saber mais sobre a instalação de um servidor Apache com todas as opções e possíveis melhorias:

- [http://www.php.net/manual/pt\\_BR/install.php](http://www.php.net/manual/pt_BR/install.php)

Experimente instalar um servidor web (Apache) no computador e abrir o código exemplo "Olá, mundo!" desta unidade didática. Para aceder pelo navegador, basta colocar o nome do servidor, que, caso tenha deixado padrão, será:

- <http://localhost>



## BIBLIOGRAFIA

---

- Cabezas, L. M. (2010), *Php* 5. Madrid: Anaya Multimedia.
- The PHP Group (2001), "What Can PHP Do?". Disponível em:  
<https://www.php.net/manual/en/intro-whatcando.php> Consultado a 22 de fevereiro de 2021.

