

## 2. Introducció a PHP

**PHP** és un dels llenguatges script de servidor més utilitzat i el que veurem en aquest mòdul.

PHP (acrònim recursiu de PHP: *Hypertext Preprocessor*) és un llenguatge de script de propòsit general i codi obert molt escaient pel desenvolupament web i que pot ser incrustat en HTML. Vegem-ho amb el següent exemple:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemple</title>
5   </head>
6   <body>
7
8     <?php
9       echo "Hola, sóc un script de PHP!";
10      ?>
11
12   </body>
13 </html>
```

Aquest exemple correspon a una pàgina HTML amb codi PHP incrustat. Aquest codi incrustat fa alguna cosa, en aquest cas, mostrar per pantalla “**Hola, sóc un script de PHP!**”. El codi de PHP està tancat entre les etiquetes especials de començament i final **<?php** y **?>** com a contingut de l’element **<body>** d’HTML.

El millor d’utilitzar PHP és la seva extrema simplicitat per als programadors nous, però alhora ofereix moltes característiques avançades per als programadors professionals.

Encara que el desenvolupament de PHP està centrat en la programació de scripts del costat del servidor, es pot utilitzar per a moltes altres coses. Existeixen principalment tres camps on s’utilitzen scripts de PHP:

- **Scripts del costat del servidor.** Aquest és el camp més tradicional i el focus principal del llenguatge. Són necessàries tres coses perquè això funcioni: l’intèrpret de PHP, un servidor web i un navegador web. Aquests tres components poden estar en la mateixa màquina, que és com treballarem nosaltres. És necessari executar el servidor amb una instal·lació de PHP connectada. Es pot accedir al resultat del programa de PHP amb un navegador, veient la pàgina de PHP que ens proporciona el servidor.
- **Scripts des de la línia de comandes.** Es pot crear un script de PHP i executar-ho sense necessitat d’un servidor o navegador. Solament és necessari l’intèrpret de PHP per a utilitzar-lo d’aquesta manera. Aquest tipus d’ús és ideal per a scripts que s’executin amb regularitat emprant cron (en Linux) o el Planificador de tasques (en Windows). Aquests scripts també poden usar-se per a tasques simples de processament de text.

- **Escriure aplicacions d'escriptori.** Probablement PHP no és el llenguatge més apropiat per a crear aplicacions d'escriptori amb una interfície gràfica d'usuari, però si es coneix bé PHP, es pot utilitzar per a escriure aquests programes mitjançant funcionalitats avançades.

PHP és multiplataforma i admet la majoria de servidors que existeixen avui en dia. Així doncs, podem triar el sistema operatiu i el servidor web que volem utilitzar. A més, podem utilitzar-lo com a llenguatge de programació per procediments o com a llenguatge de programació orientada a objectes (POO), o una barreja d'ambdues.

Una de les característiques més potents i destacables de PHP és el seu suport per a un ampli ventall de bases de dades. Escriure una pàgina web amb accés a una base de dades és molt simple utilitzant una de les extensions del llenguatge específiques per bases de dades com per exemple per MySql.

Va néixer el 1994 de la mà de Rasmus Lerdorf, però actualment el desenvolupament de referència de PHP el realitza “The PHP Group”.

L’última versió d’aquest llenguatge és la 8 i és la que farem servir nosaltres.

#### **Lloc oficial de PHP**

El següent enllaç [www.php.net](http://www.php.net) és el lloc oficial de PHP, on trobarem les llibreries de PHP categoritzades alfabèticament i per categoria, la documentació, descarregues, novetats...

## **2.1 Instal·lació de PHP8**

Els paquets que necessitem per instal·lar PHP8 en el sistema operatiu Ubuntu, es troben en el repositori de programari predeterminat d’Ubuntu. Així doncs, el primer que farem és actualitzar aquest repositori mitjançant l’ordre amb permisos de superusuari (sudo):

```
1 # sudo apt update
```

Un cop actualitzat, ja tindrem disponible l’última versió dels paquets que necessitem i els instal·larem amb la següent ordre també amb permisos de superusuari:

```
1 # sudo apt install php8.0 libapache2-mod-php8.0 php8.0-common php8.0-cli
```

- **php8** és el nom del paquet bàsic que necessitem per instal·lar l’intèrpret de PHP8.
- **libapache2-mod-php8.0** és el nom del paquet que necessitem per treballar PHP8 amb el servidor Apache.

Amb aquests dos paquets tenim prou per treballar amb PHP8, però les funcionalitats queden molt limitades, per aquest motiu instal·lem la resta de paquets, per ampliar aquestes funcionalitats:

- **php8.0-common** és el nom del paquet que proporciona la documentació, exemples i mòduls comuns per a PHP8.

- **php8.0-cli** és el nom del paquet que en proporciona l'intèpret de línia de comandes per a PHP8.

Un cop instal·lats els paquets de PHP8, només ens queda reiniciar el servidor Apache perquè la instal·lació sigui efectiva. Ho podem fer mitjançant la següent ordre també amb permisos de superusuari:

```
1 # sudo service apache2 restart
```

Per comprovar que la instal·lació ha estat exitosa, ho podem fer mitjançant la consola comprovant quina versió hi ha instal·lada de PHP, si no surt res, és que no està instal·lat i si surt alguna versió i aquesta és la que volíem instal·lar, és que la instal·lació ha estat un èxit. Per fer aquesta comprovació escriurem la següent ordre:

```
1 # php -v
```

I en el nostre cas, el resultat serà el de la figura 8.1.

**FIGURA 2.1.** Versió d'Apache

```
PHP 8.0.2 (cli) (built: Feb 7 2021 12:21:54) ( NTS )
Copyright (c) The PHP Group
Zend Engine v4.0.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.0.2, Copyright (c), by Zend Technologies
```

Finalment, comprovarem que PHP8 està operatiu. Per fer aquesta comprovació escriurem el nostre primer script. Per escriure el nostre primer script obrirem qualsevol editor de text pla com per exemple Gedit d'Ubuntu i escriurem el següent:

```
1 <?php phpinfo() ?>
```

Guardarem l'script amb el nom que vulguem i amb extensió **php**, per exemple **primerScript.php**. On el guardarem?, doncs en el servidor Apache, és a dir, si estem treballant en Ubuntu, per defecte, a **var/www/html**.

A continuació obrim un navegador com Firefox i escrivim l'URL **localhost/primerScript.php**, on localhost és el nom del nostre servidor i primerScript.php el nom de l'script que hem creat. Iniciem la navegació de l'URL escrit i si PHP8 està operatiu el que veurem en el navegador és una taula amb informació sobre PHP8.

## 2.2 Configuració de **php.ini**

**php.ini** és un dels arxius que formen part de la instal·lació de PHP i es tracta del fitxer que conté la configuració de PHP per treballar amb el servidor Apache.

L'ubicació d'aquest fitxer la podem trobar en la taula d'informació de PHP que ens retorna l'script:

```
1 <?php phpinfo() ?>
```

La figura 5.2 mostra on trobarem la ubicació en la taula:

**FIGURA 2.2.** Ubicació php.ini

PHP Version 8.0.2	
<b>System</b>	Linux fta-Lenovo-G50-80 4.15.0-135-generic #139-Ubuntu SMP Mon Jan 18 17:38:24 UTC 2021 x86_64
<b>Build Date</b>	Feb 7 2021 12:21:54
<b>Build System</b>	Linux
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/8.0/apache2
<b>Loaded Configuration File</b>	/etc/php/8.0/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/8.0/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php/8.0/apache2/conf.d/10-opcache.ini, /etc/php/8.0/apache2/conf.d/10-pdo.ini, /etc/php/8.0/apache2/conf.d/20-calendar.ini, /etc/php/8.0/apache2/conf.d/20-ctype.ini, /etc/php/8.0/apache2/conf.d/20-exif.ini, /etc/php/8.0/apache2/conf.d/20-fil.ini, /etc/php/8.0/apache2/conf.d/20-fileinfo.ini, /etc/php/8.0/apache2/conf.d/20-ftp.ini, /etc/php/8.0/apache2/conf.d/20-gettext.ini, /etc/php/8.0/apache2/conf.d/20-iconv.ini, /etc/php/8.0/apache2/conf.d/20-phar.ini, /etc/php/8.0/apache2/conf.d/20-posix.ini, /etc/php/8.0/apache2/conf.d/20-readline.ini, /etc/php/8.0/apache2/conf.d/20-shmop.ini, /etc/php/8.0/apache2/conf.d/20-sockets.ini, /etc/php/8.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.0/apache2/conf.d/20-sysvsem.ini, /etc/php/8.0/apache2/conf.d/20-sysvshm.ini, /etc/php/8.0/apache2/conf.d/20-tokenizer.ini
<b>PHP API</b>	20200930
<b>PHP Extension</b>	20200930

En el cas d'Ubuntu aquest fitxer només té permisos d'escriptura per superusuari, per tant, si volem modificar el seu contingut, tenim dues opcions:

- Obrir-lo com a superusuari.
- Canviar-li els permisos perquè altres usuaris que no siguin superusuari els puguin modificar. Aquesta segona opció és més pràctica, ja que si ho fem així, podrem modificar el fitxer obrint-lo amb qualsevol editor de text pla com Gedit, però també és menys segura, ja que si estem treballant en un entorn compartit, on per exemple, l'ordinador que es fa servir per programar amb PHP el fan servir diferents usuaris, aleshores aquest fitxer el pot modificar qualsevol usuari que utilitzi aquest entorn. Així doncs, aquesta segona opció només és recomanable si tenim clar que l'ordinador només el faran servir els programadors que treballaran amb PHP i poden modificar aquest fitxer.

Si escollim la primera opció, l'obrirem mitjançant el terminal d'Ubuntu amb permisos de superusuari, utilitzant un editor com Nano amb la següent ordre:

```
1 # sudo nano /etc/php/8.0/apache2/php.ini
```

On **/etc/php/8.0/apache2/php.ini** és la ubicació on trobem el fitxer php.ini que apareix en la taula d'informació de PHP.

Si decidim optar per la segona opció, per canviar els permisos del fitxer php.ini, escriurem la següent comanda amb permisos de superusuari:

```
1 # sudo chmod 777 /etc/php/8.0/apache2/php.ini
```

Un cop obrim el fitxer, ens trobarem amb un gran nombre de línies de text. La figura 5.3 mostra part del contingut del fitxer php.ini:

**FIGURA 2.3.** Contingut php.ini

```
[Pdo]
; Whether to pool ODBC connections. Can be one of "strict", "relaxed" or "off"
; http://php.net/pdo-odbc.connection-pooling
;pdo_odbc.connection_pooling=strict

[Pdo_mysql]
; Default socket name for local MySQL connects. If empty, uses the built-in
; MySQL defaults.
pdo_mysql.default_socket=

[Phar]
; http://php.net/phar.readonly
;phar.readonly = On

; http://php.net/phar.require-hash
;phar.require_hash = On

;phar.cache_list =

[mail function]
; For Win32 only.
; http://php.net/smtp
SMTP = localhost
; http://php.net/smtp-port
smtp_port = 25

; For Win32 only.
; http://php.net/sendmail-from
;sendmail_from = me@example.com

; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
;sendmail_path =
```

Aquestes línies les podem classificar de la següent forma:

- **Línies que comencen per punt i coma (;**). Solen tractar-se de comentaris explicant el funcionament d'unes certes directives, com per exemple “;MySQL defaults.”.
- **Text marcat amb claudàtor.** Per exemple [Pdo] que indiquen la capçalera d'una determinada secció.
- **Directives.** És la part més important d'aquest arxiu, ja que mitjançant elles, establim la configuració de PHP i estan formades per una parella composta per una clau i valor associat a aquesta clau amb el següent format:

<sup>1</sup> nom\_directiva=valor\_directiva

On **nom\_directiva** serà el nom de la directiva (clau) i **valor\_directiva**, el valor que li assignarem a la directiva (valor associat a la clau), com per exemple:

<sup>1</sup> SMTP = localhost

Es pot donar el cas que tinguem directives amb un punt i coma al principi. Això vol dir que estan comentades i que no són tingudes en compte a l'hora de treballar amb PHP. Si les volguéssim utilitzar, hauríem d'eliminar el punt i coma del principi.

Algunes de les modificacions que podem fer en el fitxer php.ini que ens permetran configurar-lo per adaptar-lo a la nostra manera de treballar i aportar-li seguretat a PHP són:

---

Cal tenir present que les majúscules i minúscules en el fitxer php.ini no són el mateix, és a dir que “Directiva = on” és diferent a “directiva = on”, ja que en el primer cas la “D” és majúscula i en el segon cas minúscula.

---

- Visualització i registres d'errors
- Ocultar la versió de PHP
- Desactivar accés a URL remotes des de funcions de maneig de fitxers
- Directiva open\_basedir

### 2.2.1 Visualització i registres d'errors

És possible configurar PHP perquè en cas que es produueixi algun error es generin notificacions. Per aconseguir-ho hem de modificar la directiva **error\_reporting** de la següent manera:

---

```
1 error_reporting = E_ALL
```

---

En aquest cas, aquesta configuració és aconsellable quan estem en fase de desenvolupament; és a dir, creant scripts, perquè es generaran les següents notificacions:

- Els errors de programació que es puguin produir.
- Si fem servir alguna funcionalitat que està obsoleta (*deprecated*).
- Avisos sobre el nostre script que no hem de confondre amb els errors. Un avís simplement ens notifica algun aspecte de l'script que hem de decidir si modificar o no, però no produeix cap mal funcionament en l'execució de l'script. Un exemple d'avís seria que no hem inicialitzat una variable, en aquest cas, hem de decidir si és necessari inicialitzar-la o no.

Quan l'script ja està en fase de producció (en ús), l'única informació que ens interessarà són els errors que es produueixin, per tant, el valor que li assignarem a la directiva **error\_reporting** serà:

---

```
1 error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
```

---

D'aquesta manera només es generaran les notificacions dels errors, però no les funcionalitats obsoletes o els avisos.

És important que el programador visualitzi aquesta informació i la pot visualitzar de dues maneres:

- **Imprimint-la per pantalla.** Aquesta opció no és recomanable quan l'script està en producció, ja que l'usuari pot visualitzar informació que pot arribar a ser crítica, però és molt pràctica pel programador quan l'script està en fase de desenvolupament, ja que en el moment que executa l'script, en la mateixa pantalla visualitzarà els possibles errors.

- **Emmagatzemant-la en un fitxer log (fitxer de registre).** En aquest fitxer de text pla es guardarà tota la informació per línies indicant la data i hora en què s'ha produït. Aquesta opció és la recomanable quan l'script està en producció perquè d'aquesta manera l'usuari no pot veure la informació.

Si seleccionem la primera opció, hem de modificar la directiva **display\_errors** de la següent manera:

```
1 display_errors = On
```

Si seleccionem la segona opció, és necessari modificar tres directives: **display\_errors**, **log\_errors** i **error\_log**. Ho fem de la següent manera:

```
1 display_errors = Off
2 log_errors = On
3 error_log = /ruta/log
```

On **/ruta/log** és la ruta on guardarem els logs i que hem de definir nosaltres.

## 2.2.2 Ocultar la versió de PHP

El que s'intenta amb aquesta mesura és que no es pugui saber la versió de PHP que corre en el servidor web. Així evitarem que terceres persones puguin conèixer la versió que s'utilitza per a trobar vulnerabilitats i dur a terme algun tipus d'atac. Per tant, modificarem la directiva **expose\_php** de la següent manera:

```
1 expose_php = Off
```

## 2.2.3 Desactivar accés a URL remotes des de funcions de maneig de fitxers

Funcions de PHP com **include**, a més de cridar a arxius locals, permeten cridar a arxius remots. Això pot ser un greu problema de seguretat, però ho podem evitar modificant la directiva **allow\_url\_fopen** de la següent manera:

```
1 allow_url_fopen = Off
```

Estudiarem la funció "include" a l'apartat "Referència del llenguatge I" d'aquesta unitat (UF1).

## 2.2.4 Directiva open\_basedir

Mitjançant la configuració d'aquesta directiva, indiquem que PHP pugui accedir únicament als fitxers continguts en el directori indicat (i als seus subdirectoris).

D'aquesta manera, ens garantim que no es pugui accedir a llocs que no corresponguin. L'ús d'aquesta directiva seria el següent:

```
1 open_basedir = /directori_amb_accés
```

On **/directori\_amb\_accés** serà l'únic directori on es podrà accedir.