

Introducción a PHP

PHP es un lenguaje de programación imperativo, interpretado, dinámico, de tipado débil.

Breve historia de PHP

La invención de php se debe a la necesidad de incrustar código en un documento web de manera que, al ejecutarse, cambiase el contenido de dicho documento. Para ello, se desarrollo un mecanismo que permite procesar "código embebido", que un servidor recibiera junto con las demás etiquetas html, obviándolas e interpretando sólo el código, que podrá producir distintos efectos y modificaciones sobre la apariencia y comportamiento del documento web.

Al inicio, su creador, **Rasmus Lerdorf**, lo denominó *Personal Home Page*, pero actualmente sus siglas corresponden a **PHP Hypertext Preprocessor** (su acrónimo es recursivo a sí mismo).

¿Cómo es PHP en la actualidad?

A pesar de que se inspiraron en C y Perl para elaborar su sintaxis, actualmente se asemeja más a Java, aunque por sus características funcionales tiene más parecido a JavaScript (ya que ambos son lenguajes interpretados, dinámicos y de tipado débil).

Instalación

Dado que Ubuntu 21.04 no incluye en sus repositorio los paquetes referentes a php8.0, vamos a añadir un repositorio externo que si contenga dichos paquetes.

Importante destacar que agregar repositorios externos puede suponer una brecha de seguridad en nuestro sistema y sólo debemos agregar repositorios de fuentes de confianza.

1. Añadimos el repositorio de [ondrej/php](https://ppa.launchpad.net/ondrej/php) mediante la siguiente orden:

```
sudo apt-add-repository ppa:ondrej/php
```

2. A continuación, instalaremos una serie de paquetes:

```
sudo apt install php8.0 php8.0-cli php8.0-pgsql php8.0-mbstring
```

- php8.0 - Servidor de PHP versión 8.
- php8.0-cli - SAPI para la interfaz en línea de comandos (Command Line Interpreter) .
- php8.0-pgsql - Módulo controlador de PDO para la integración y soporte con bases de datos PostgreSQL.
- php8.0-mbstring - Módulo para la gestión de cadenas multi-byte.

Existen otros paquetes que pueden ser útiles:

- php8.0-fpm - SAPI alternativo útil para sitios web con mucho tráfico. (FastCGI Process Manager)

- php8.0-sqlite3 - Módulo controlador de PDO para la integración y soporte con bases de dato SQLite, versión 3.

SAPI's

Los SAPI's son distintos entornos donde se puede ejecutar php (Contexto de ejecución)

- SAPI-CLI: Entorno en linea de comando (Comand Line Interpreter).
- SAPI-Apache (web): Entorno con interfaz web, o que corre sobre un servidor web. Este es quizás el que más nos interesa.
- FPM: Una manera de ejecutar php a traves de un server web, la forma en que se monta php en producción. Este irá asociado a un servidor web.

PDO (PHP Data Object)

Un PDO es un Objeto de datos de PHP. Define una interfaz ligera para poder acceder a bases de datos en PHP, proporcionando una capa de abstracción de acceso a datos.

A partir de ella, y con el uso de un **controlador de PDO** específico al tipo de base de de datos con el que queremos trabajar (Postgre, MySQL, SQLite), tendremos acceso a un servidor de bases de datos, con características específicas y funciones habituales. Ejemplos son los módulos `php8.0-pgsql` y `php8.0-sqlite3`.

Configuración

En la caperta `/etc/php` estan los ficheros de configuracion de PHP. Habra carpetas separadas por version. Aqui podremos ver directorios por los distintos SAPI's. De los 3 que disponemos, vamos a usar cli. Con el podremos incluso simular un servidor web.

Configuración principal

`php.ini` es el fichero de configuracion principal*. Está ubicado en `/etc/php/9.0/cli*/php.ini`.

*: Cada SAPI posee su propio fichero php.ini.

`-i` (info) ofrece información sobre php. Sobre, por ejemplo, cómo está configurado(recomendable acompañar con `less`). También aparecerán los distintos módulos cargados, indicando detalles sobre el mismo. Sale primero el valor por defecto y despues el valor asignado.

Revisando la configuración principal de PHP

Vamos a echar un vistazo al fihcero php.ini No se recomienda un editor de código demasiado avanzado.

Editores en linea de comandos son las opciones más interesantes (vi, nano) aunque en mi caso voy a utilizar por el momento el editor gráfico gedit.

El fichero `php.ini` dispone de una gran cantidad de documentación en su interior, con incluso recomendaciones sobre su configuración, dependiendo de si se pretende usar para desarrollar, o en modo de "producción".

Se denomina *modo de producción* cuando el servidor está desplegado y suministrando sus servicios sobre un producto finalizado, de consumo para los usuarios del sitio web en el que funciona.

Recomendaciones de configuración inicial

A continuación mencionamos algunos de los elementos principales a configurar:

- **display_errors:** Decide si, cuando hay un error en el código, los mensajes serán mostrados en pantalla o no. Cuando vamos a desplegar nuestra aplicación, lo lógico es no mostrar ese error, ya que puede mostrar información sensible.

Su valor por defecto es "Off". En modo de desarrollo se recomienda "On".

- **error_reporting:** Si se van a generar y/o mostrar mensajes de error, filtraremos qué errores se van a generar, y cuáles no. Esto es, en definitiva, el nivel de traza a partir del cual los errores se mostrarán en pantalla o no.

Su valor por defecto es: "E_ALL & ~E_DEPRECATED & ~E_STRICT". En modo de desarrollo se recomienda "E_ALL".

- **log_errors:** Decidiremos si los errores se reportarán a un fichero de registro de errores.

Su valor por defecto es "On". En modo de desarrollo se recomienda "On".

- **default_charset:** Es interesante comprobar que usamos el mapa de caracteres apropiado, que depende también del objetivo de nuestro sitio web.

Su valor por defecto es "UTF-8".

- **date.timezone:** Nos interesa trabajar con la zona horaria que nos corresponde, en nuestro caso se trata de **UTC**.

Por defecto esta opción viene comentada, sin incluir ningún valor. Nosotros vamos a descomentarla y a indicar como valor 'UTC'.

Hay una opción que funciona como interruptor general para activar o desactivar el motor php. Se trata de `engine`. Debe estar activo si queremos que el motor php funcione bajo el servidor web (cli-apache?).

Resumen de elementos a configurar en el fichero `php.ini`:

```
display_errors: On
display_startup_errors: On
```

```
error_reporting: E_ALL
log_errors: 0n
default_charset: "UTF-8"
date.timezone = 'UTC'
```

Extensiones (Módulos)

Hay un directorio (`mod-available`) donde se ubican los módulos disponibles. En él habrá una serie de ficheros `.ini` que componen los distintos módulos, como una extensión que puedes acoplar a PHP para agregarle más funcionalidades pero OJO estos no están activados. Los mods que se activarán son los que se encuentran dentro del directorio del SAPI que vamos a usar.

Dentro del SAPI hay un directorio llamado `conf.d`, donde encontramos una serie de ficheros, que son en realidad enlaces simbólicos que se encuentra dentro de la carpeta de módulos disponibles. Por lo tanto, Cuantos más enlaces haya en este directorio, más módulos habrá activos.

Agregar y remover módulos

Se puede usar `phpenmod` para añadir módulos de una forma más fácil.

Sintaxis:

```
phpenmod [-v ALL\php_version] [-s ALL\sapi_name] module_name [module_name_2]
```

- -v: Version de php.
- -s: SAPI al que añadir el módulo.
- module_name(+): Nombre del módulo para añadir (+: 1 o más).

Del mismo modo, existe también `phpdismod` para desactivar módulos.

Ejecutar código PHP

Existen tres modos principales para ejecutar código php:

- Intérprete por lotes: Recibe ficheros con instrucciones php y lo ejecuta. Lo usaremos un poco, para hacer pruebas y testear nuestros primeros programas. Pero el funcionamiento normal es incrustar PHP en documentos HTML.
Se ejecuta mediante la orden `php`, seguida del fichero a procesar..
- Intérprete interactivo (REPL): Como su nombre indica, vamos introduciendo instrucciones a medida que estás van ejecutandose. Usaremos un intérprete diferente del integrado en PHP. Se llama **psysh** (psyshell)
- Servidor web integrado. En este modo se interpreta código PHP incrustado en documentos HTML.

El intérprete por lotes

Para que nuestro código php se detecte, es necesario introducirlo entre las etiqueta de apertura `<?php` y la etiqueta de cierre `?>`.

Tiene dos modos de operación:

- Modo PHP: Se entra cada vez que se encuentra la etiqueta de apertura. En el modo PHP el intérprete ejecuta el código interpretado. Cuando encuentra la etiqueta de cierre detiene este modo y pasa al modo HTML.
- Modo HTML: Es el modo por defecto al comenzar la ejecución. Este modo funciona como un método de pasar directamente todo lo que vaya procesando por la salida. Si detecta una etiqueta de apertura, cambiará al modo PHP hasta que encontrar una tiqueta de cierre.

Esto significa que podemos tener varios bloques de código PHP. Si nuestro fichero php no va a contener ningun otro lenguaje, se acostumbra a no incluir la etiqueta de cerradura, ya que no es necesario volver al modo HTML.

El intérprete interactivo (REPL)

Si llamamos a la óden php y le acompañamos del parámetro `-a` accederemos a la sesión interactiva de php. Dado que es muy básico y *endeble*, no es el método más adecuado para aprender y practicar php. Este intérprete espera sentencias en lugar de expresiones, por lo que nos conviene mejor un interptrete que sea capaz de evaluar expresiones.

Psysh

Se trata de un intérprete de php que incluye mejoras sustanciales con respecto a su versión nativa.

Instalación:

1. Usaremos wget <https://psysh.org/psysh> para descargar el fichero 'psysh'.
2. Le daremos permisos de ejecución:

```
chmod +x psysh
```
3. A partir de ahí ya nos funcionaría, pero lo vamos a mover a un directorio dedicado del path para que siempre podamos llamarlo.

```
sudo mv psysh /usr/local/bin
```

Ahora podemos ejecutarlo en cualquier lugar desde la terminal.

Detalle interesante de psyshell: Ofrece documentación sobre elementos del lenguaje php mediante la orden `doc`.