

# php.ini, el archivo de configuración de PHP



El lenguaje de programación PHP es actualmente el más utilizado a escala mundial a la hora de realizar **portales web**. Herramientas open source como **WordPress**, Moodle o tiendas online como Prestashop o Magento, están desarrollada bajo esta tecnología gracias a las muchas opciones que ofrece, opciones que en muchos casos pueden ser configuradas desde su archivo PHP.INI, que será el protagonista a lo largo de este White Paper.

## ¿Qué es el archivo php.ini?

Entre todos los archivos que forma parte de la instalación del lenguaje PHP, hay uno que destaca sobre el resto debido a su importancia a la hora de trabajar con este lenguaje. Se trata del archivo php.ini, un fichero que contiene la configuración de PHP que vamos a utilizar en el **servidor** donde alojemos nuestro portal.

```
451 ;;;;;;;;;;;;;;;;;;;;;;;;;;
452 ; Resource Limits ;
453 ;;;;;;;;;;;;;;;;;;;;;;;;;;
454
455 ; Maximum execution time of each script, in seconds
456 ; http://php.net/max-execution-time
457 ; Note: This directive is hardcoded to 0 for the CLI SAPI
458 max_execution_time=600
459
460 ; Maximum amount of time each script may spend parsing request data. It's a good
461 ; idea to limit this time on productions servers in order to eliminate unexpectedly
462 ; long running scripts.
463 ; Note: This directive is hardcoded to -1 for the CLI SAPI
464 ; Default Value: -1 (Unlimited)
465 ; Development Value: 60 (60 seconds)
466 ; Production Value: 60 (60 seconds)
467 ; http://php.net/max-input-time
468 max_input_time=600
469
470 ; Maximum input variable nesting level
471 ; http://php.net/max-input-nesting-level
472 ;max_input_nesting_level = 64
473
474 ; How many GET/POST/COOKIE input variables may be accepted
475 ; max_input_vars = 1000
476
477 ; Maximum amount of memory a script may consume (128MB)
478 ; http://php.net/memory-limit
479 memory_limit=128M
```

Este archivo lo solemos encontrar dentro del directorio donde hemos realizado la instalación de la distribución PHP. Se trata de un archivo de texto que puede ser editado con cualquier editor de texto simple que tengamos instalado en nuestro equipo y una vez que lo editemos, nos encontraremos con un gran número de líneas de texto. Estas líneas las podemos clasificar de la siguiente forma:

- **Líneas que comienzan por punto y coma (;).** Suelen tratarse de comentarios explicando el funcionamiento de ciertas directivas, siendo todas ellas ignoradas.
- **Texto marcado con corchete.** Por ejemplo [PHP] que indican la cabecera de una determinada sección.
- **Directivas.** Es la parte más importante de este archivo y están formadas por una pareja compuesta por una clave y su determinado valor. Por ejemplo "una\_directiva = off". A la hora de modificar una directiva, hay que tener presente que las mayúsculas y minúsculas no son lo mismo, es decir que "Una\_directiva = on" es diferente a "una\_directiva = on".

Se puede dar el caso de que tengamos directivas con un punto y coma al principio. Esto quiere decir que están comentadas y que no son tenidas en cuenta a la hora de trabajar con PHP. Si las quisiéramos utilizar, deberíamos eliminar el punto y coma del principio.

## ¿Dónde está ubicado mi archivo php.ini?

Anteriormente hemos dicho que el archivo php.ini está ubicado en el directorio donde se realizó la instalación de PHP. En muchos casos, a la hora de contratar un servidor dedicado con alguna **empresa de hosting**, la instalación de PHP viene ya instalada, por lo que si queremos conocer su ubicación y no volvernos loco buscándolo por todo el sistema, lo más sencillo es ejecutar la sentencia "phpinfo()" de php, que nos muestra los valores de configuración de ese archivo, así como su ubicación entre otras muchas cosas.

Para eso, en un fichero con extensión ".php" escribimos el siguiente código.

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

Al ejecutar ese código, obtendremos una pantalla como la que os dejamos a continuación.

PHP Version 5.2.4-2ubuntu5.7	
System	Linux virt101.unelink.net 2.6.32-4-pve #1 SMP Mon Sep 20 11:36:51 CEST 2010 i686
Build Date	Aug 21 2009 19:32:27
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/ssh2.ini, /etc/php5/apache2/conf.d/ksl.ini
PHP API	20041225
PHP Extension	20060813
Zend Extension	220060519
Debug Build	no

En ella podéis ver la ruta donde está ubicado nuestro archivo y el que deberemos editar para cambiar cualquier configuración de nuestro PHP.

## Ejemplos de directivas dentro de php.ini

El archivo php.ini engloba un gran número de directivas orientadas a diferentes aspectos, como puede ser la subida de archivos vía HTTP, otras relacionadas con directorios y archivos o algunas relacionadas con los límites de recursos que pueden utilizar los scripts programados en PHP. A continuación os dejaremos un listado a modo de ejemplo de las que nos podemos encontrar, pero podéis ver muchas más en la **sección de directivas dentro de la página oficial de PHP**.

- **max\_execution\_time.** Mediante esta directiva indicamos el tiempo máximo que un script podrá estar ejecutándose antes de que sea cancelado. El tiempo es indicado en segundos.
- **memory\_limit.** En esta directiva, lo que indicamos es la memoria que se podrá utilizar para la ejecución del código que forma parte de la web.
- **post\_max\_size.** Se trata de la directiva que permite controlar el tamaño de los datos enviados mediante POST, afectando también a los archivos que se subirán al servidor mediante programación. Si el archivo supera el límite indicado, dará un error y no se subirá.
- **max\_input\_time.** Aquí definiremos el tiempo máximo que puede estar PHP procesando la entrada de datos. El valor por defecto es de 60 segundos.
- **upload\_max\_filesize.** Esta directiva lo que controla es el tamaño máximo de ficheros que se puede subir a la web de forma simultánea. Se puede subir uno o más de uno siempre y cuando no se supere el valor indicado aquí.
- **default\_charset.** En esta directiva se le puede indicar el tipo de codificación que se utilizará por defecto, siendo el de por defecto UTF-8.
- **include\_path.** Especifica la lista de directorios donde las funciones require, include, fopen(), file(), readfile() y file\_get\_contents() buscarán ficheros. Permite una lista de directorios separados por punto y coma.
- **file\_uploads.** Si permite o no la subida de archivos vía HTTP.
- **upload\_tmp\_dir.** Aquí indicaremos el directorio temporal utilizado durante el proceso de subida de los archivos. Si no se especifica, PHP utilizará el de por defecto del sistema.
- **max\_file\_uploads.** El máximo número de archivos que se podrán subir de forma simultánea.

## Consejos para hacer más segura la instalación de PHP



Como ocurre con la mayoría de las aplicaciones informáticas que utilizamos, es importante tomar las medidas necesarias para hacerlas lo más **seguras posible**. En el caso de PHP, además de tenerlo siempre actualizado a la última versión estable disponible, podemos hacerla algo más segura desde el propio archivo php.ini.

### 1.- Ocultar la versión de PHP

Lo que se intenta con esta medida es que no se pueda saber la versión de PHP que corre en un determinado servidor haciendo un simple telnet. De esta forma evitaremos que terceras personas puedan conocer la versión que se utiliza para encontrar vulnerabilidades y llevar a cabo algún tipo de ataque. Por tanto conviene cambiar a off la directiva "expose\_php".

## 2.- Deshabilitar sesión ID en la URL

En ocasiones no queremos que en las direcciones aparezcan los IDs de las sesiones. Para ello debemos cambiar el valor de la directiva “session.use\_trans\_sid” y ponerlo a off. De esta forma en vez de tener en la barra de direcciones: `http://miempresaweb.es/?PHPSESSID=4kgj577sgabvnmhjgkdiuy1956if6ska`

Tendríamos: `http://miempresaweb.es`

## 3.- Desactivar acceso a URL remotas desde funciones de manejo de ficheros

Funciones de PHP como “include”, “fopen” o “file\_get\_content”, además de llamar a archivos locales, permiten llamar a archivos remotos. Esto puede ser un grave problema de seguridad, pero lo podemos evitar modificando el valor de “allow\_url\_fopen” y poniéndolo a off.

## 4.- Directiva open\_basedir

Mediante la configuración de esta directiva, indicamos que PHP pueda acceder únicamente a los ficheros contenidos en el directorio indicado (y sus subdirectorios). De esta forma nos garantizamos que no se pueda acceder a sitios que no correspondan. El uso de esta directiva sería de la forma que os indicamos a continuación.

```
open_basedir = /directorio_con_acceso
```

## 5.- Visualización y registros de errores

Es posible configurar PHP para que en caso de que se produzca algún error, no se muestre por pantalla sino que se almacene en un log cuya ruta le indiquemos. Con esto evitamos que se muestre por pantalla información que puede llegar a ser crítica. Para evitar que se muestre esta información, es necesario modificar estas tres directivas que os dejamos a continuación.

```
display_errors = Off
```

```
log_errors = On
```

```
error_log = /ruta/fichero/log
```

Modificando el valor de “display\_errors” a “on” volveríamos a configurarlo para que se mostrase los errores por pantalla. Esto es bueno mientras que se esté depurando la aplicación.

## 6.- Desactivar funciones peligrosas

También es posible desactivar ciertas funciones de PHP por temas de seguridad. Para conseguir esto, es necesario modificar la directiva “disable\_functions”, a la que se le asigna un listado de funciones separadas por comas (,).

```
disable_functions="system,passthru,escapeshellarg,escapeshellcmd,proc_close,proc_open,ini_alter  
,popen,show_source,pcntl_exec"
```

En este WhitePaper hemos hablado del archivo php.ini, archive desde donde configuramos determinados aspectos de la instalación de PHP, y donde deberemos acudir cuando queramos modificar el comportamiento de nuestro intérprete PHP.