

Directivas para configurar el archivo wp-config.php de WordPress



Directivas wp-config.php

Dentro de los archivos que forman parte de cualquier instalación de WordPress, uno de los más importantes es el fichero “**wp-config.php**”. Se trata del archivo de configuración de este CMS y que se crea de forma automática cuando se realiza su instalación. Su importancia radica en que contiene información básica sobre la configuración de la **página web**, como por ejemplo los datos de conexión a la base de datos. Lo que no todo el mundo conoce es que por medio de ciertas directivas podemos definir unas configuraciones avanzadas, por ejemplo habilitar las funciones de depuración, impedir las actualizaciones automáticas o tener la posibilidad de **reparar la base de datos**.

A lo largo de nuestro White Paper veremos algunas de las directivas más importantes que podemos utilizar en el interior del archivo “**wp-config.php**” para definir funcionalidades avanzadas. Eso sí, antes de seguir, si vais a modificar este archivo sacaros una copia de seguridad, para que en caso de sufrir cualquier problema lo podáis restaurar. Borrarlo o editarlo con errores hará que nuestro portal web deje de funcionar.

¿Dónde colocamos las nuevas directivas?

Como ya hemos comentado anteriormente, al realizar la instalación de este CMS se creará el archivo con los datos básicos para el funcionamiento de la página. Un ejemplo de este tipo de configuración mínima la podemos ver en la siguiente imagen.

```
1 <?php
2 // ** MySQL settings ** //
3 define('DB_NAME', 'nombre_BBDD'); // The name of the database
4 define('DB_USER', 'user_BBDD'); // Your MySQL username
5 define('DB_PASSWORD', 'pass_BBDD'); // ...and password
6 define('DB_HOST', 'localhost'); // 99% chance you won't need to change this value
7 define('DB_CHARSET', 'utf8');
8 define('DB_COLLATE', '');
9
10 // You can have multiple installations in one database if you give each a unique prefix
11 $table_prefix = 'wp_'; // Only numbers, letters, and underscores please!
12
13 // Change this to localize WordPress. A corresponding MO file for the
14 // chosen language must be installed to wp-content/languages.
15 // For example, install de.mo to wp-content/languages and set WPLANG to 'de'
16 // to enable German language support.
17 define ('WPLANG', 'es_ES');
18
19 /* That's all, stop editing! Happy blogging. */
20
21 define('ABSPATH', dirname(__FILE__).'/');
22 require_once(ABSPATH.'wp-settings.php');
23 ?>
24
```

A la hora de añadir nuestras nuevas directivas, las debemos poner antes de línea “**require_once**”.

```
require_once(ABSPATH.'wp-settings.php');
```

Si las añadimos después de ella, nos dará un error y el sitio dejará de funcionar. También es importante saber que el orden de las directivas importa, por lo que cambiar el orden de las directivas ya presentes,

puede provocar que el sitio deje de funcionar. Para evitar esto, recomendamos añadirlas al final, justo antes de la línea que hemos comentado anteriormente.

Configuraciones adicionales en wp-config.php

A continuación pasaremos a ver algunos ejemplos de directivas que podemos utilizar dentro del archivo “wp-config.php” en la instalación de nuestro WordPress.

Activar el modo debug

Cuando se está desarrollando una página web es bueno que en caso de que se produzca algún tipo de error, éste pueda ser mostrado por pantalla o bien en algún archivo log donde poder verlo, para que sea más fácil dar solución al problema. WordPress nos permite activar un modo debug para estos casos por medio de la directiva “WP_DEBUG”. Por defecto, ésta se encuentra desactivada cuando se realiza la instalación. Si lo queremos activar, deberemos añadir la siguiente línea a nuestro archivo.

```
define('WP_DEBUG', true);
```

Si además de que se muestre por pantalla, queremos que los errores se guarden en un archivo llamado “debug.log” dentro de la carpeta “wp-content”, entonces deberíamos añadir también la siguiente línea.

```
define ( 'WP_DEBUG_LOG', true);
```

En el caso de que únicamente queramos que los errores se muestren en el archivo log y no por pantalla, junto a las anteriores directivas, habría que añadir la siguiente.

```
define ( 'WP_DEBUG_DISPLAY', false);
```

Es muy importante que una vez que el sitio esté en producción subido al servidor definitivo, poner a false la directiva “WP_DEBUG”.

Cambiar el prefijo de las tablas de la base de datos

WordPress por defecto asigna el prefijo “wp_” a las tablas que forman parte de su base de datos. Si queréis aumentar la seguridad de vuestra base de datos, es buena idea cambiar este prefijo por otro más complejo formado por letras y números. Para hacer esto, deberíamos modificar la variable “\$table_prefix” que aparece dentro del archivo de configuración. Un ejemplo de prefijo complejo podría ser el siguiente:

```
$table_prefix = '937yjh38_';
```

Una vez cambiado el prefijo en el archivo de configuración, deberíais cambiarlo también en cada una de las tablas que forman parte de la [instalación de WordPress](#), cosa que podéis hacer de formas sencilla desde el phpMyAdmin.

Definir las rutas del sitio

Desde la propia administración de WordPress es posible cambiar las URLs de inicio en el administrador desde el apartado “**Configuración** -> **General**”. Esta información se almacena en base de datos, pero también se puede configurar desde el archivo “**wp-config.php**” mediante el uso de las constantes “**WP_SITEURL**” y “**WP_HOME**”. Definir esto en este archivo nos ofrece dos ventajas:

- Puede ayudarle en el caso de que no pueda acceder a la administración del sitio por cualquier tipo de problema
- Se reduce el número de llamadas a la base de datos mientras se carga la web, ya que los valores almacenados en este archivo tienen preferencia sobre la información almacenada en la base de datos

Mediante la constante WP_SITEURL, se especifica la URL con la que los usuarios pueden visitar nuestro sitio, mientras que con WP_HOME, se define la raíz donde se instaló WordPress. Puede ser en la raíz del dominio o bien en una carpeta.

```
define('WP_SITEURL', 'https://misitio.es');  
define('WP_HOME', 'https://misitio.es/wordpress');
```

Vaciar la papelera transcurrido un tiempo

Es posible hacer que nuestro WordPress se encargue de vaciar automáticamente la papelera después de un cierto número de días que le indiquemos. De esta forma, no tendremos que hacer este trabajo, además de conseguir que la instalación esté limpia de información que no se utiliza. Para ello, es necesario utilizar la constante “**EMPTY_TRASH_DAYS**”.

```
define( 'EMPTY_TRASH_DAYS', 7 );
```

El valor mínimo que se le puede indicar es 0. En ese caso, estaremos desactivando la eliminación automática del vaciado de la papelera.

Cambiar la configuración de las revisiones

Por defecto, WordPress va guardando diferentes revisiones de las entradas y páginas que creamos a medida que las vamos editando. Esto nos ofrece la ventaja de poder volver a una versión anterior en caso

necesario. Pero esto tiene un precio que se traduce en un mayor peso de nuestra base de datos. Si no queremos que WordPress guarde esas revisiones, lo podemos hacer añadiendo la siguiente línea en nuestro archivo de configuración.

```
define( 'WP_POST_REVISIONS', false );
```

Si no queremos desactivar la opción de las revisiones de forma completa, también tenemos la opción de indicar el número de revisiones a guardar.

```
define( 'WP_POST_REVISIONS', 2 );
```

En el caso anterior, únicamente se guardarían dos revisiones de cada entrada o página que editemos.

Aumentar la memoria disponible para PHP

Si tienes experiencia con WordPress, lo más probable es que en alguna ocasión te hayas encontrado con el problema de no tener memoria suficiente. Esto es debido a la memoria que el **servidor** asigna a la ejecución de scripts en PHP. Hay varias formas de solucionar esto, aunque una de las más sencillas es añadiendo la siguiente directiva en nuestro archivo “**wp-config.php**”.

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

En nuestro ejemplo, le estamos asignado una memoria de 256 megas, pero podrías poner el valor que queramos. Eso sí, debemos tener en cuenta la memoria que dispone nuestro servidor, ya que el valor indicado nunca podrá ser superior. Por ejemplo, si nuestro **alojamiento web** dispone de 512 MB no podremos asignarle 600 o 700 megas.

Actualizaciones automáticas en Wordpress

Ya hace algún tiempo que WordPress activó las actualizaciones automáticas. Esto resulta muy útil para aquellas personas que no tienen tiempo para hacerlas por sí mismo. De esta forma, su instalación estará siempre actualizada a la última versión, aunque hay casos en los que nos puede interesar no actualizar por incompatibilidad del tema o con algún plugin que utilicemos. En estos casos, podemos definir si queremos bloquear este tipo de actualizaciones automáticas, o bien si queremos que se ejecuten solo algunos tipos.

Por defecto, WordPress viene configurado para llevar a cabo actualizaciones automáticas menores y archivos de traducción. Pero podemos ver algunos otros casos mediante el uso de las constantes **AUTOMATIC_UPDATER_DISABLED** y **WP_AUTO_UPDATE_CORE**. Veamos qué opciones tenemos.

La primera de ellas es desactivar todo tipo de actualizaciones.

```
define( 'AUTOMATIC_UPDATER_DISABLED', true );
```

En el caso de únicamente querer desactivar las actualizaciones del core de WordPress, deberíamos añadir la siguiente línea.

```
define( 'WP_AUTO_UPDATE_CORE', false );
```

También tenemos la posibilidad de que únicamente se lleven a cabo las actualizaciones del core de WordPress menores, por ejemplo de la 4.8.2 a la 4.8.3. En este caso, deberíamos utilizar lo siguiente.

```
define( 'WP_AUTO_UPDATE_CORE', 'minor' );
```

Desactivar la edición de plugins y temas

Si administras un sitio con varias personas más, te puede interesar que éstos no puedan editar ni los plugins ni los temas para evitar cualquier tipo de problema. Esto lo podemos conseguir añadiendo la siguiente directiva a nuestro archivo de configuración.

```
define( 'DISALLOW_FILE_EDIT', true );
```

Cambiar la localización de directorios de WordPress

Una instalación de WordPress está compuesta de varios directorios importantes, aunque los más destacados son:

- **/wp-content/uploads:** Aquí es donde se ubican todas las imágenes y archivos que subimos desde el administrador del sitio.
- **/wp-content/themes:** Directorio donde se ubica el theme que utiliza nuestra página web. Puede haber más de uno, pero solo uno estará activo.
- **/wp-content/plugins:** Como su nombre bien indica, en su interior se colocarán todos los plugins que vayamos instalando.

Como hemos visto, todos ellos se encuentran dentro del directorio “**wp-content**”. Sin embargo, puede ser que en ciertas ocasiones queramos cambiar esas rutas. En el caso de que queramos que el directorio “wp-content” esté en otro sitio, lo podremos hacer utilizando la directiva **WP_CONTENT_DIR**.

```
define( 'WP_CONTENT_DIR', dirname( __FILE__ ) . '/my-content' );
```

Si únicamente queremos cambiar la ubicación de la carpeta donde se guardan los plugins, también lo podemos hacer, pero en este caso, utilizaremos **WP_PLUGIN_DIR**.

```
define( 'WP_PLUGIN_DIR', dirname(__FILE__) . '/my-plugins' );
```

Por último, podemos cambiar el directorio donde queremos que se guarden los archivos. Para ello, habrá que declarar la constante **UPLOADS**.

```
define( 'UPLOADS', 'wp-content/my-uploas' );
```

Reparar la base de datos

En el caso de tener algún problema con la base de datos y ser necesaria su reparación, WordPress nos permite poder repararla de forma manual añadiendo la siguiente directiva al archivo “**wp-config.php**”.

```
define( 'WP_ALLOW_REPAIR', true );
```

A continuación, habrá que entrar en la siguiente ruta y ejecutar la optimización.

<https://midominio.com/wp-admin/maint/repair.php>

Una vez que hayáis reparado la base de datos, no os olvidéis de eliminar la línea anterior o bien cambiar el “**true**” por “**false**”.

Cambiar el intervalo de guardado automático

Si utilizas habitualmente WordPress, te habrás dado cuenta de que cada cierto tiempo guarda la información de forma automática. Esto viene muy bien en el caso de sufrir algún tipo de problema, ya que nuestro trabajo no se perderá. En el caso de querer cambiar ese intervalo, lo podremos hacer mediante la siguiente directiva:

```
define( 'AUTOSAVE_INTERVAL', 300 );
```

El valor numérico que indicamos representa los segundos que deben pasar entre guardado y guardado.

Como hemos visto a lo largo de nuestro **White Paper**, el archivo de configuración de WordPress nos ofrece muchas opciones diferentes añadiendo algunas directivas. Ahora solo falta que vosotros decidáis cuáles son las que mejor se adaptan a vuestras necesidades.