

# Wordpress

Trucos y utilidades para Wordpress

- [Crear un Usuario administrador de WordPress desde phpMyAdmin](#)
- [Modo DEBUG en Wordpress](#)
- [Agregar Elemento a la lista de medios](#)
- [Comandos wp cli](#)
- [Bloquear el registro de usuarios en WordPress](#)
- [Plugin de Redis en Wordpress](#)

# Crear un Usuario administrador de WordPress desde phpMyAdmin

Ingresamos al **Panel** de nuestra cuenta, buscamos la sección «**BASE DE DATOS**» y damos clic en la pestaña **phpMyadmin**.

Escogemos la base de datos que usamos en nuestro **WordPress** y hacemos clic en nuestra base de datos.

En nuestro **phpMyadmin** buscamos la pestaña «**SQL**» para copiar la sentencia que vamos a ejecutar.

Pegamos el siguiente código y cambiando los datos:

Este es el código que debemos pegar en la pestaña **SQL** y esta es la forma de **como crear usuario WordPress desde phpMyAdmin**.

```
INSERT INTO `wp_usermeta` (`umeta_id`, `user_id`, `meta_key`, `meta_value`)
VALUES (NULL, (Select max(id) FROM wp_users), 'wp_capabilities',
'a:1:{s:13:"administrator";s:1:"1";}');
```

```
INSERT INTO `wp_users` (`user_login`, `user_pass`, `user_nicename`, `user_email`,
`user_status`)
VALUES ('username', MD5('Mipassword1234'), 'Nombre Apellido', 'admin@example.com', '0');
```

Todas las **bases de datos de WordPress** vienen por default el prefijo «**wp\_**», recomendamos que en las bases de datos por seguridad, el prefijo o el nombre suele estar cambiado, entonces debemos poner el prefijo por ejemplo "**34jx**", entonces las consultas serían sobre **34jxwp\_usermeta** y **34jxwp\_users**

# Modo DEBUG en Wordpress

Para activar el modo DEBUG en Wordpress debemos añadir algunos parámetros al archivo de configuración wp-config.php de Wordpress.

Al final del archivo añadimos (si no existe) lo siguiente:

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

En el siguiente ejemplo vemos como como quedaría:

```
/**
 * Para desarrolladores: modo debug de WordPress.
 *
 * Cambia esto a true para activar la muestra de avisos durante el desarrollo.
 * Se recomienda encarecidamente a los desarrolladores de temas y plugins que usen WP_DEBUG
 * en sus entornos de desarrollo.
 */
define(' WP_DEBUG', false);

/* ¡Eso es todo, deja de editar! Feliz blogging */
```

Si además queremos tener el modo DEBUG de Wordpress activo pero que no se muestren los mensajes de debug o de error en la ventana al cargar WordPress, añadimos a continuación en otro línea:

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

Si queremos que todos registros producidos por el modo DEBUG se guarden en un archivo de registro en la instalación de Wordpress, debemos añadir el siguiente parámetro:

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

## Ejemplos:

Mode debug activo, sin visualizar y guardando en el log:

```
/**
 * Para desarrolladores: modo debug de WordPress.
 *
 * Cambia esto a true para activar la muestra de avisos durante el desarrollo.
 * Se recomienda encarecidamente a los desarrolladores de temas y plugins que usen WP_DEBUG
 * en sus entornos de desarrollo.
 */
define(' WP_DEBUG', true);
define(' WP_DEBUG_DISPLAY', false);
define(' WP_DEBUG_LOG', true);

/* ¡Eso es todo, deja de editar! Feliz blogging */
```

Mode debug activo, visualizando por pantalla y guardando en el log:

```
/**
 * Para desarrolladores: modo debug de WordPress.
 *
 * Cambia esto a true para activar la muestra de avisos durante el desarrollo.
 * Se recomienda encarecidamente a los desarrolladores de temas y plugins que usen WP_DEBUG
 * en sus entornos de desarrollo.
 */
define(' WP_DEBUG', true);
define(' WP_DEBUG_DISPLAY', true);
define(' WP_DEBUG_LOG', true);

/* ¡Eso es todo, deja de editar! Feliz blogging */
```

Desactivar el modo DEBUG

```
/**
 * Para desarrolladores: modo debug de WordPress.
 *
 * Cambia esto a true para activar la muestra de avisos durante el desarrollo.
 * Se recomienda encarecidamente a los desarrolladores de temas y plugins que usen WP_DEBUG
 * en sus entornos de desarrollo.
 */
define(' WP_DEBUG', false);
```

```
define(' WP_DEBUG_DISPLAY', true);  
define(' WP_DEBUG_LOG', true);  
  
/* ¡Eso es todo, deja de editar! Feliz blogging */
```

Como vemos al desactivar el debug, el resto de modificadores no afecta.

## Con el comando wp cli

El comando wp cli es muy útil, como podemos [comprobar en este artículo al respecto](#)

Este comando nos permite modificar el fichero de configuración de Wordpress (El wp-config.php), pero al contrario de lo que podemos hacer (si no somos cuidadosos), este nos deja todo en el sitio que tiene que estar.

Por ejemplo si queremos **habilitar el modo debug en Wordpress**

```
wp config set WP_DEBUG true --raw
```

Si queremos habilitar todas las opciones de depuración

```
wp config set --raw WP_DEBUG true  
wp config set --raw WP_DEBUG_LOG true  
wp config set --raw WP_DEBUG_DISPLAY true
```

# Agregar Elemento a la lista de medios

Algunas veces podemos necesitar subir un archivo por FTP o SSH a la carpeta Uploads de Wordpress. El problema, es que ese archivo no aparecerá en el visor de medios de Wordpress.

Para arreglar esto podemos usar el wp-cli

```
sudo -u usuario_del_hosting /opt/php-7.4/bin/php wp-cli.phar media import /wp-content/uploads/2023/05/07_archivo_enorme.jpg
```

# Comandos wp cli

## Comprobar instalación Wordpress

```
wp core verify-checksums
```

Si la versión que tenemos instalada es en otro idioma

```
wp core verify-checksums --locale=es_ES
```

## Comprobar los Plugins

```
wp plugin verify-checksums
```

## Deshabilitar comentarios

```
wp disable-comments settings --types=post --add  
wp post list --format=ids | xargs wp post update --comment_status=closed  
wp post list --post_type=page --format=ids | xargs wp post update --ping_status=closed
```

## Actualizar plugins

```
wp plugin update --all
```

## Actualizar temas

```
wp theme update --all
```

## Comprobar actualizaciones de plugins

```
wp plugin update --all --dry-run
```

## Comprobar y mostrar la salida en JSON

```
wp plugin update --all --dry-run --format=json
```

## Instalar find para buscar instalaciones de Wordpress

```
wp package install wp-cli/find-command
```

## wp find (encuentra instalaciones de Wordpress)

```
wp find ./ --format=json
```

## Instalar Wordpress

```
wp core download --path=wpclidemo.dev
Creating directory '/srv/www/wpclidemo.dev/'.
Downloading WordPress 4.6.1 (en_US)...
Using cached file '/home/vagrant/.wp-cli/cache/core/wordpress-4.6.1-en_US.tar.gz'...
Success: WordPress downloaded
cd wpclidemo.dev
wp core config --dbhost=host.db --dbname=prefix_db --dbuser=username --dbpass=password
chmod 644 wp-config.php
```

```
wp core install --url=yourwebsite.com --title="Your Blog Title" --admin_name=wordpress_admin
--admin_password=4Long&Strong1 --admin_email=you@example.com
cd wp-content
mkdir uploads
chgrp web uploads/
```

## Migrar Wordpress

```
/opt/php-7.0/bin/php wp-cli.phar search-replace 'dominioviejo.com' 'dominionuevo.es' --
recurse-objects --network --skip-columns=guid --skip-tables=wp_users
```

## Reinstalar Wordpress

```
wp core download --force --skip-content
```

## Otros ejemplos

```
wp post list --format=ids | xargs wp post update --comment_status=closed

wp post list --comment_status=open --format=ids | xargs wp post update --comment_status=closed
wp post list --ping_status=open --format=ids | xargs wp post update --ping_status=closed
```



```
wp plugin update --all --dry-run --format=json
wp theme update --all --dry-run --format=json
wp find ./ --format=json

wp core download --path=wpclidemo.dev
```

# Otros comandos de wp cli

## Crear un fichero de configuración de Wordpress

Este comando crea un fichero wp-config.php

```
wp config create
```

Más información [en este enlace](#)

Y la joya de la corona

```
wp config set
```

Este comando nos permite modificar el fichero de configuración de Wordpress (El wp-config.php), pero al contrario de lo que podemos hacer (si no somos cuidadosos), este nos deja todo en el sitio que tiene que estar.

Por ejemplo si queremos **habilitar el modo debug en Wordpress**

```
wp config set WP_DEBUG true --raw
```

Si queremos habilitar todas las opciones de depuración

```
wp config set --raw WP_DEBUG true
wp config set --raw WP_DEBUG_LOG true
wp config set --raw WP_DEBUG_DISPLAY true
```



# Bloquear el registro de usuarios en WordPress

Cuando se instala un Wordpress, muchas veces se deja habilitada la opción de que los usuarios puedan registrarse, esto no se debería hacer, salvo contadas excepciones, ya que es una fuente de intentos de ataque de bots y spammers.

Por ello es muy importante desactivar esta opción desde el backend (parte de administración de Wordpress) en “Ajustes” >> “Generales” y en Miembros desactivar la opción de cualquiera puede registrarse

**Miembros**

☐

Cualquiera puede registrarse

# Plugin de Redis en Wordpress

Como podemos ver en este artículo de [configuración de Redis](#), podemos usar el caché de objetos de Redis, para mejorar el rendimiento de nuestras instancias de Wordpress.

Para ello, procederemos a instalar el plugin de Redis para Wordpress que se encuentra en este [enlace](#)

Después de instalar y activar el complemento, vete a WordPress -> Configuración -> Redis

Allí, habilita el caché y verifica si el complemento se puede conectar automáticamente.

De forma predeterminada, la memoria caché de objetos se conectará al servidor Redis a través de TCP en 127.0.0.1:6379 y usará la base de datos 0, si ves Estado: no conectado, comprueba que el servidor Redis está arrancado y que la configuración es correcta.

Existen unos parámetros de configuración de Redis

```
// adjust Redis host and port if necessary
define( 'WP_REDIS_HOST', '127.0.0.1' );
define( 'WP_REDIS_PORT', 6379 );

// change the prefix and database for each site to avoid cache data collisions
define( 'WP_REDIS_PREFIX', 'mi-wordpress' );
define( 'WP_REDIS_DATABASE', 0 ); // 0-15

// reasonable connection and read+write timeouts
define( 'WP_REDIS_TIMEOUT', 1 );
define( 'WP_REDIS_READ_TIMEOUT', 1 );
```

El complemento Redis Object Cache viene con un amplio conjunto de opciones de configuración. Si no estás seguro de cómo usarlos, lee las instrucciones de instalación.

Básicamente los parámetros de configuración disponibles son:

Parámetro	Default	Descripción
-----------	---------	-------------

<code>WP_REDIS_HOST</code>	<code>127. 0. 0. 1</code>	Nombre o IP del Servidor Redis
<code>WP_REDIS_PORT</code>	<code>6379</code>	Puerto del Servidor Redis
<code>WP_REDIS_PATH</code>		Ruta al socket de Unix del del Servidor Redis
<code>WP_REDIS_SCHEME</code>	<code>tcp</code>	El esquema utilizado para conectar <code>tcp</code> or <code>unix</code>
<code>WP_REDIS_DATABASE</code>	<code>0</code>	La base de datos para el cache: <code>0-15</code>
<code>WP_REDIS_PREFIX</code>		El prefijo utilizado para todas las claves de caché para evitar colisiones de datos, reemplaza <code>WP_CACHE_KEY_SALT</code> . Debe ser una cadena legible, no un "salt" al uso
<code>WP_REDIS_PASSWORD</code>		La contraseña del servidor Redis. Admite arrays de ACL de Redis: <code>[ 'user', 'password' ]</code>
<code>WP_REDIS_MAXTTL</code>	<code>0</code>	El tiempo de vida máximo de las claves de caché
<code>WP_REDIS_CLIENT</code>		EL cliente para comunicarse con Redis: <code>redis</code> , <code>phpredis</code> or <code>relay</code>
<code>WP_REDIS_TIMEOUT</code>	<code>1</code>	El tiempo de espera de la conexión en segundos
<code>WP_REDIS_READ_TIMEOUT</code>	<code>1</code>	El tiempo de espera de la conexión en segundos para leer/escribir
<code>WP_REDIS_IGNORED_GROUPS</code>	<code>[]</code>	Grupos que no deben almacenarse en caché entre solicitudes en Redis