

---

# Administración de Wordpress con la utilidad WP-CLI

Implantación de Aplicaciones Web

José Juan Sánchez Hernández

Curso 2024/2025

# Índice

<b>1</b>	<b>Práctica: Administración de Wordpress con la utilidad WP-CLI</b>	<b>1</b>
1.1	Requisitos . . . . .	1
1.2	Instalación de WP-CLI en el servidor LAMP . . . . .	1
1.3	Ejemplo de uso: Instalación de WordPress con WP-CLI . . . . .	2
1.3.1	Paso 1. Descarga del código fuente de WordPress ( <code>wp core download</code> ) . . . . .	2
1.3.2	Paso 2. Creación de la base datos, usuario y contraseña en MySQL Server . . . . .	3
1.3.3	Paso 3. Creación del archivo de configuración ( <code>wp config create</code> ) . . . . .	3
1.3.4	Paso 4. Comprobación de los valores del archivo de configuración ( <code>wp config get</code> ) . . . . .	3
1.3.5	Paso 5. Instalación de WordPress ( <code>wp core install</code> ) . . . . .	4
1.3.6	Paso 6. Modificación del propietario y grupo del directorio <code>/var/www/html</code> . . . . .	5
1.3.7	Paso 7. Configuración de los enlaces permanentes o permalinks . . . . .	5
1.4	Comandos útiles . . . . .	6
1.4.1	Actualización de los plugins a la última versión . . . . .	6
1.4.2	Actualización de los themes a la última versión . . . . .	6
1.4.3	Comprobar la versión del core de WordPress . . . . .	6
1.4.4	Actualizar el core de WordPress a la última versión . . . . .	7
1.4.5	Cambiar el idioma de WordPress . . . . .	7
1.4.6	Listar los plugins instalados . . . . .	7
1.4.7	Instalar y activar un plugin . . . . .	7
1.4.8	Configurar un plugin . . . . .	7
1.4.9	Desactivar un plugin . . . . .	7
1.4.10	Eliminar un plugin . . . . .	8
1.4.11	Eliminar los plugins que están inactivos . . . . .	8
1.4.12	Listar los themes instalados . . . . .	8
1.4.13	Instalar y activar un theme . . . . .	8
1.4.14	Activar un theme previamente instalado . . . . .	8
1.4.15	Eliminar un theme . . . . .	9
1.4.16	Eliminar los themes que están inactivos . . . . .	9
1.5	Entregables . . . . .	9
1.5.1	Documento técnico . . . . .	9
1.5.2	Scripts de Bash . . . . .	10
<b>2</b>	<b>Referencias</b>	<b>11</b>
<b>3</b>	<b>Licencia</b>	<b>12</b>

# Índice de figuras

# Índice de cuadros

# 1 Práctica: Administración de Wordpress con la utilidad WP-CLI

En esta práctica vamos a realizar la administración de un sitio [WordPress](#) desde el terminal con la utilidad [WP-CLI](#).

Con la utilidad [WP-CLI](#) podemos realizar las mismas tareas que se pueden hacer desde el panel de administración web de [WordPress](#), pero desde la línea de comandos.

En la web oficial de [WordPress](#) podemos encontrar:

- [El manual de uso de WP-CLI](#)
- [Los comandos de WP-CLI](#).

## 1.1 Requisitos

La máquina donde vamos a instalar la utilidad [WP-CLI](#) será la misma máquina donde queremos realizar la instalación de [WordPress](#) y tendrá que tener instalado todo el software necesario de la pila LAMP.

## 1.2 Instalación de WP-CLI en el servidor LAMP

Descargamos el archivo `wp-cli.phar` del repositorio oficial de [WP-CLI](#). Los archivos `.phar` son unos archivos similares a los archivos `.jar` de Java. Estos archivos se utilizan para distribuir aplicaciones o librerías de PHP en un único archivo. Puede encontrar [más información sobre los archivos .phar en la documentación oficial de PHP](#).

```
1 curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

Le asignamos permisos de ejecución al archivo `wp-cli.phar`.

```
1 chmod +x wp-cli.phar
```

Movemos el archivo `wp-cli.phar` al directorio `/usr/local/bin/` con el nombre `wp` para poder utilizarlo sin necesidad de escribir la ruta completa donde se encuentra.

```
1 sudo mv wp-cli.phar /usr/local/bin/wp
```

Una vez que hemos realizado el paso anterior podemos hacer uso de la utilidad desde la línea de comandos. Si escribimos `wp` en el terminal nos deberá mostrar la página de ayuda del comando.

```
1 wp
```

En la [documentación oficial](#) puede encontrar información más detallada sobre el proceso de instalación.

## 1.3 Ejemplo de uso: Instalación de WordPress con WP-CLI

Vamos a realizar un **ejemplo de uso sobre una máquina** que ya tiene instalado todo el software necesario de la pila LAMP.

### 1.3.1 Paso 1. Descarga del código fuente de WordPress (`wp core download`)

Para descargar el código fuente de WordPress con el comando `wp core download` podemos hacerlo de dos formas:

#### Opción 1

(1) Nos situamos en el directorio donde queremos realizar la instalación de WordPress.

```
1 cd /var/www/html
```

Descargamos el código fuente de WordPress con el comando `wp core download`.

```
1 wp core download
```

Al ejecutar el comando anterior descargaremos el código fuente en el directorio actual.

#### Opción 2

(2) Utilizamos el parámetro `--path` para indicar el directorio donde queremos descargar el código fuente de WordPress. Por ejemplo:

```
1 wp core download \  
2 --path=/var/www/html
```

También es posible seleccionar el idioma de WordPress, para que sólo se descargue el paquete del idioma que hemos elegido.

```
1 wp core download \  
2 --locale=es_ES \  
3 --path=/var/www/html
```

Si va a ejecutar el comando `wp` como `sudo` deberá añadir el parámetro `--allow-root`.

```
1 wp core download \  
2 --locale=es_ES \  
3 --path=/var/www/html \  
4 --allow-root
```

Tenga en cuenta que si ejecuta el comando anterior como `sudo`, el código fuente de WordPress que ha descargado en el directorio `/var/www/html` tendrá como propietario al usuario `root` y el grupo también será `root`. Por lo tanto, al final de todo el proceso de instalación tendrá que modificar el propietario y el grupo del directorio `/var/www/html` al usuario `www-data` y al grupo `www-data`.

Puede encontrar más información sobre el comando `wp core download` en la [documentación oficial](#).

### 1.3.2 Paso 2. Creación de la base datos, usuario y contraseña en MySQL Server

Antes de crear el archivo de configuración debemos crear en MySQL Server la base de datos, el usuario y la contraseña donde vamos a instalar WordPress.

### 1.3.3 Paso 3. Creación del archivo de configuración (`wp config create`)

Una vez que hemos creado la base de datos, usuario y contraseña en MySQL Server, podemos crear el archivo de configuración `wp-config.php` para WordPress con el siguiente comando:

```
1 wp config create \  
2 --dbname=wordpress_db \  
3 --dbuser=wordpress_user \  
4 --dbpass=wordpress_password \  
5 --dbhost=localhost \  
6 --path=/var/www/html \  
7 --allow-root
```

En este paso le estamos indicando los siguientes parámetros:

- `--dbname`: Nombre de la base de datos. En el ejemplo anterior se utiliza `wordpress_db`.
- `--dbuser`: Usuario de la base de datos. En el ejemplo anterior se utiliza `wordpress_user`.
- `--dbpass`: Contraseña del usuario de la base de datos. En el ejemplo anterior se utiliza `wordpress_password`.
- `--dbhost`: Host de la base de datos. En el ejemplo anterior se utiliza `localhost`.

Tenga en cuenta que en su caso tendrá que reemplazar los valores: `wordpress_db`, `wordpress_user` y `wordpress_password`, por los valores que quiera utilizar para su base de datos, usuario y contraseña.

Al crear el archivo de configuración se crearán automáticamente los valores aleatorios para las **security keys**.

Puede encontrar más información sobre el comando `wp config create` en la [documentación oficial](#).

### 1.3.4 Paso 4. Comprobación de los valores del archivo de configuración (`wp config get`)

Podemos comprobar los valores del archivo de configuración del archivo `wp-config.php` con el comando `wp config get`:

```
1 wp config get \
2 --path=/var/www/html \
3 --allow-root
```

El comando anterior nos devolverá la siguiente salida:

```
1 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 | name                | value                                                                                               | type      |
3 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 | table_prefix        | wp_                                                                                               | variable  |
5 | DB_NAME              | wordpress_demo                                           | constant  |
6 | DB_USER              | wordpress_user                                           | constant  |
7 | DB_PASSWORD         | wordpress_password                                       | constant  |
8 | DB_HOST              | localhost                                                | constant  |
9 | DB_CHARSET          | utf8                                                     | constant  |
10 | DB_COLLATE           |                                                           | constant  |
11 | AUTH_KEY             | 4T-<=<=4ye0(9^Y/1-Cak7Y7w;I(hl*0C%(^Y4}j1((J.Bjy^F^.p3RuI
12 | SECURE_AUTH_KEY     | IaT>N:]LUq3ghCgI<8Kd!{;CSx<Dj0P;S|As5R cJDfDF-<.U}v_{:TL
13 | LOGGED_IN_KEY       | %Yu8YeNBZi8 d7Dw|}m6_*?B&I3Cv_Hg]#:d-%7E;J8hmPsHpbTjRH<
14 | NONCE_KEY           | d)6jmy+R^g`- ]4Q8R-CEP:$l-@:4(Y=( _r9(T4WpFzU!$mTR5&o?q{
15 | AUTH_SALT           | _?(_k0f5V{a)*YSbb4_E=g)`P[z(Va4/x@xS]M]If &b1oS`+ RsLSy-
16 | SECURE_AUTH_SALT   | 0CsKv6U9ko0m*g74V9-Z(cCZMn89?JQ0w:NO6N1m1->qV(wg.0G!05d}6
17 | LOGGED_IN_SALT     | oy%3|A9!dWt=q+{+r$s;3*k1&dD02WN27J0[_TBt,E1Nr* uuf|,7
18 | NONCE_SALT         | y#rfX-<Vz[X!mHwIg(?X)]P$t(#~F#C6f({}&0^?.@9g;7@es_W/|RtDWV^
19 | WP_CACHE_KEY_SALT | S_xmq*@Xp%p+sw8i.bn**5?*&0td.|&C#At_h;5H^n,%#9T4.`
20 | WP_CACHE_KEY_SALT | s785siMz3$@QKb | constant |
```

Puede encontrar más información sobre el comando `wp config get` en la [documentación oficial](#).

### 1.3.5 Paso 5. Instalación de WordPress (`wp core install`)

Una vez que tenemos la base de datos creada y el archivo de configuración preparado podemos realizar la instalación de WordPress con el comando:

```
1 wp core install \  
2   --url=practica-wordpress.ddns.net \  
3   --title="IAW" \  
4   --admin_user=admin \  
5   --admin_password=admin_password \  
6   --admin_email=test@test.com \  
7   --path=/var/www/html \  
8   --allow-root
```

En este paso le estamos indicando los siguientes parámetros:

- `--url`: Dominio del sitio WordPress. En el ejemplo anterior hemos utilizado el dominio `practica-wordpress.ddns.net`, que es el nombre del dominio que hemos configurado en el servidor DNS. En una instalación en una máquina local también podríamos poner `localhost` o la dirección IP del servidor.
- `--title`: Título del sitio WordPress.
- `--admin_user`: Nombre del usuario administrador.
- `--admin_password`: Contraseña del usuario administrador.
- `--admin_email`: Email del usuario administrador.

Tenga en cuenta que en su caso tendrá que reemplazar los valores: `practica-wordpress.ddns.net`, `IAW`, `admin`, `admin_password` y `test@test.com`, por los valores que quiera utilizar para el dominio del sitio, título, nombre, contraseña y dirección de correo del administrador.

Puede encontrar más información sobre el comando `wp config install` en la [documentación oficial](#).

### 1.3.6 Paso 6. Modificación del propietario y grupo del directorio `/var/www/html`

Como hemos estado ejecutando el comando `wp` como `sudo`, el contenido del directorio `/var/www/html` tiene como propietario al usuario `root` y el grupo `root`. Por lo tanto, tenemos que modificar esta configuración para que el propietario sea el usuario `www-data` y el grupo `www-data`.

```
1 sudo chown -R www-data:www-data /var/www/html
```

### 1.3.7 Paso 7. Configuración de los enlaces permanentes o permalinks

Para configurar los enlaces permanentes con el nombre de las entradas podemos utilizar el siguiente comando.

```
1 wp rewrite structure '/%postname%' \  
2   --path=/var/www/html \  
3   --allow-root
```

El parámetro `/%postname%/` indica el patrón de reescritura que queremos utilizar. Otros patrones de reescritura que podemos utilizar son:

- `/%year%//%monthnum%//%day%//%postname%/`: Año, mes, día y nombre de la entrada.

- `/%year%/monthnum%/postname%/:` Año, mes y nombre de la entrada.
- `/%year%/postname%/:` Año y nombre de la entrada.
- `/%postname%/:` Nombre de la entrada.

Tenga en cuenta que para que los enlaces permanentes funcionen correctamente es necesario que el módulo `mod_rewrite` de Apache esté activado y que la directiva `AllowOverride All` esté configurada en su sitio virtual de Apache.

También es necesario un archivo `.htaccess` en el directorio raíz de WordPress con las reglas de reescritura apropiadas. Este archivo se crea automáticamente cuando se configuran los enlaces permanentes desde el panel de administración web de WordPress o también puede crearlo de forma manual y copiarlo en el directorio raíz de WordPress.

Si queremos que el comando `wp rewrite structure` cree el archivo `.htaccess` de forma automática es necesario configurar correctamente el archivo de configuración de WP-CLI como se indica en la [documentación oficial](#).

## 1.4 Comandos útiles

En la [documentación oficial](#) puede encontrar un listado de todos los comandos que se pueden utilizar.

A continuación, se muestra un pequeño resumen de algunos de los comandos que podemos utilizar con [WP-CLI](#).

### 1.4.1 Actualización de los plugins a la última versión

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

Puede encontrar más información sobre el comando `wp plugin update` en la [documentación oficial](#).

### 1.4.2 Actualización de los themes a la última versión

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

Puede encontrar más información sobre el comando `wp plugin update` en la [documentación oficial](#).

### 1.4.3 Comprobar la versión del core de WordPress

```
1 wp core check-update
```

Puede encontrar más información sobre el comando `wp core check-update` en la [documentación oficial](#).

### 1.4.4 Actualizar el core de WordPress a la última versión

```
1 wp core update
```

Puede encontrar más información sobre el comando `wp core update` en la [documentación oficial](#).

### 1.4.5 Cambiar el idioma de WordPress

Podemos listar los idiomas disponibles para WordPress.

```
1 wp language core list
```

Una vez que hemos encontrado el nombre del idioma podemos cambiar el idioma con el siguiente comando.

```
1 wp language core install en_US --activate
```

Puede encontrar más información sobre el comando `wp language core` en la [documentación oficial](#).

### 1.4.6 Listar los plugins instalados

```
1 wp plugin list
```

Puede encontrar más información sobre el comando `wp plugin list` en la [documentación oficial](#).

### 1.4.7 Instalar y activar un plugin

Para instalar el plugin `wps-hide-login` ejecutaríamos el siguiente comando:

```
1 wp plugin install wps-hide-login --activate
```

Puede encontrar más información sobre el comando `wp plugin install` en la [documentación oficial](#).

### 1.4.8 Configurar un plugin

Para configurar el nombre de la nueva URL que vamos a utilizar para acceder al panel de administración con el plugin `wps-hide-login` ejecutaríamos el siguiente comando:

```
1 wp option update whl_page "nombre_secreto" --path=/var/www/html --allow-root
```

Puede encontrar más información sobre el comando `wp option update` en la [documentación oficial](#).

### 1.4.9 Desactivar un plugin

Para desactivar el plugin `wps-hide-login` ejecutaríamos el siguiente comando:

```
1 wp plugin deactivate wps-hide-login
```

Puede encontrar más información sobre el comando `wp plugin deactivate` en la [documentación oficial](#).

#### 1.4.10 Eliminar un plugin

Para eliminar el plugin `wps-hide-login` ejecutaríamos el siguiente comando:

```
1 wp plugin delete wps-hide-login
```

Puede encontrar más información sobre el comando `wp plugin delete` en la [documentación oficial](#).

#### 1.4.11 Eliminar los plugins que están inactivos

```
1 wp plugin delete $(wp plugin list --status=inactive --field=name)
```

Puede encontrar más información sobre el comando `wp plugin delete` en la [documentación oficial](#).

#### 1.4.12 Listar los themes instalados

```
1 wp theme list
```

Puede encontrar más información sobre el comando `wp theme list` en la [documentación oficial](#).

#### 1.4.13 Instalar y activar un theme

Para instalar y activar el theme `mindscape` ejecutaríamos el siguiente comando:

```
1 wp theme install mindscape --activate
```

Puede encontrar más información sobre el comando `wp theme install` en la [documentación oficial](#).

#### 1.4.14 Activar un theme previamente instalado

Para activar el theme `twentytwenty` ejecutaríamos el siguiente comando:

```
1 wp theme activate twentytwenty
```

Puede encontrar más información sobre el comando `wp theme activate` en la [documentación oficial](#).

### 1.4.15 Eliminar un theme

Para eliminar el theme `mindscape` ejecutaríamos el siguiente comando:

```
1 wp theme delete mindscape
```

Puede encontrar más información sobre el comando `wp theme delete` en la [documentación oficial](#).

### 1.4.16 Eliminar los themes que están inactivos

```
1 wp theme delete $(wp theme list --status=inactive --field=name)
```

Puede encontrar más información sobre el comando `wp theme delete` en la [documentación oficial](#).

## 1.5 Entregables

Deberá crear un repositorio en [GitHub](#) con el nombre de la práctica y añadir al profesor como colaborador.

El repositorio debe tener el siguiente contenido:

- Un **documento técnico** con la descripción de todos los pasos que se han llevado a cabo.
- Los **scripts de Bash** que se han utilizado para automatizar la instalación y configuración de [WordPress](#) con la utilidad [WP-CLI](#).

Además del contenido anterior puede ser necesario crear otros archivos de configuración. A continuación se muestra un ejemplo de cómo puede ser la estructura del repositorio:

```
1 .|—
2  README.md|—
3  conf|
4    └─ 000-default.conf|—
5  htaccess|
6    └─ .htaccess|—
7  scripts|—
8    .env|—
9    install_lamp.sh|—
10   setup_letsencrypt_https.sh|—
11   deploy_wordpress_with_wpcli.sh
```

### 1.5.1 Documento técnico

El documento técnico `README.md` tiene que estar escrito en [Markdown](#) y debe incluir **como mínimo** los siguientes contenidos:

- Descripción del proceso de instalación de la instalación de [WordPress](#) con la utilidad [WP-CLI](#).

## 1.5.2 Scripts de Bash

El directorio `scripts` debe incluir los siguientes archivos:

- `.env`: Este archivo contiene todas las variables de configuración que se utilizarán en los scripts de Bash.
- `install_lamp.sh`: Script de Bash con la automatización del proceso de instalación de la pila LAMP.
- `setup_letsencrypt_https.sh`: Script de Bash con la automatización del proceso de solicitar un certificado SSL/TLS de Let's Encrypt y configurarlo en el servidor web Apache.
- `deploy_wordpress_with_wpcli.sh`: Script de Bash con la automatización del proceso de instalación de WordPress sobre el directorio raíz `/var/www/html` con la utilidad `wp-cli`.

## 2 Referencias

- [WordPress](#).
- [WP-CLI](#).
- [WP-CLI de 0 a 100. Fernando García \(video\)](#). WorldCamp Online 2020.
- [WP-CLI de 0 a 100. Fernando García \(pdf\)](#). WorldCamp Online 2020.

## **3 Licencia**

Esta página forma parte del curso Implantación de Aplicaciones Web de José Juan Sánchez y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.