

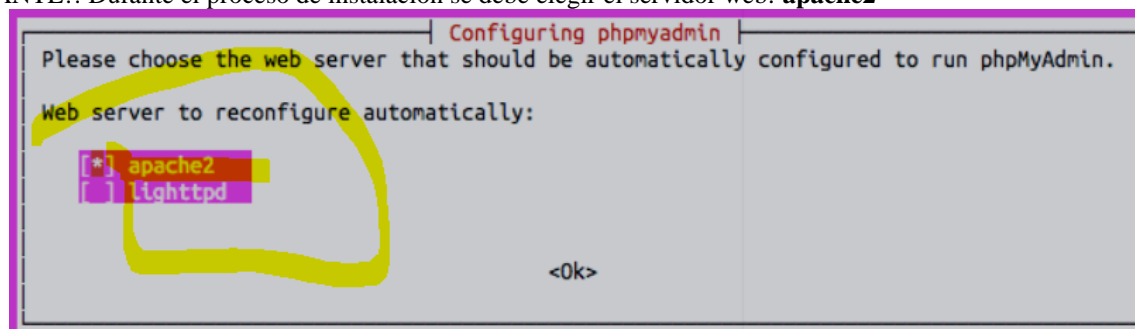
## LAMP Y LEMP

BLOQUE	APACHE 2.4	NGINX 1.23
<b>Instalación de PHP (LAMP/LEMP)</b>	<p style="text-align: center;"><b>LAMP STACK EN UBUNTU/DEBIAN</b></p> <p>-Ubuntu 22.04/Debian 12 (Sistema operativo)          -Apache v2.4 (Servidor web)          -PHPv8.X (Lenguaje de programación)          -MariaDBv10.X (Sistema gestor de base de datos)</p> <p style="text-align: center;"><b><u>APACHE2 Y PHP</u></b></p> <p><b>Paso 1: Actualización repositorios:</b>          apt update</p> <p><b>Paso 2: Instalación Apache2:</b>          #apt install apache2 -y</p> <p><b>Paso 3: Instalación PHP</b>          #apt install php libapache2-mod-php php-mysql -y</p> <p><b>Paso 4: Editamos sitio web por defecto (000-default.conf):</b>  <b>Nota:</b> Añadimos la directiva <b>DirectoryIndex</b> indicando la página PHP por defecto:  <b>El sitio web (host virtual) quedaría de la siguiente manera:</b>  <pre>&lt;VirtualHost *:80&gt; #ServerName www.example.com ServerAdmin webmaster@localhost DocumentRoot /var/www/html DirectoryIndex index.html index.php (Nota: Se debe añadir esta línea) ErrorLog \${APACHE_LOG_DIR}/error.log CustomLog \${APACHE_LOG_DIR}/access.log combined &lt;/VirtualHost&gt;</pre> <p><b>Paso 5: Reiniciamos servicio apache2</b>          #systemctl restart apache2</p> <p><b>Paso 6:- Comprobación de LAMP stack:</b>          Creamos en el “documentroot” del sitio web por defecto (/var/www/html) una página de prueba: <b>info.php</b> para testear php:  <b>echo "&lt;?php phpinfo(); ?&gt;" &gt; info.php</b>          Desde el navegador e incluimos la siguiente URL:  <b>http://ip_servidor/info.php</b></p> <p style="text-align: center;"><b><u>ANEXOS</u></b></p> <p><b>Instalación Extensiones PHP (No necesario, de momento)</b>          sudo apt install php-cli php-mbstring php-xml php-common php-curl</p> </p>	<p style="text-align: center;"><b>LEMP STACK EN UBUNTU/DEBIAN</b></p> <p>-Ubuntu 22.04/Debian 12 (Sistema operativo)          -Apache v2.4 (Servidor web)          -PHPv8.X (Lenguaje de programación)          -MariaDBv10.X (Sistema gestor de base de datos)</p> <p style="text-align: center;"><b><u>NGINX Y PHP</u></b></p> <p><b>Paso 1: Actualización repositorios:</b>          apt update</p> <p><b>Paso 2: Instalación nginx</b>          apt install nginx -y</p> <p><b>Paso 3: Instalación PHP:</b>          sudo apt install php-fpm php-mysql</p> <p><b>Paso 4: Editamos sitio web por defecto (default):</b>          /etc/nginx/sites-available/default y realizamos los siguientes cambios:          1.- En la sección index añadimos el valor index.php en primer lugar para que darle prioridad respecto a los archivos index.html.  <b>index index.php index.html index.htm index.nginx-debian.html;</b>          2.- Descomentamos, y en su caso modificamos, las siguientes directivas:  <pre>location ~ \.php\$ { include snippets/fastcgi-php.conf; fastcgi_pass unix:/var/run/php/php8.2-fpm.sock; } location ~ /\.ht { deny all; }</pre> <p><b>Nota:</b> Hay que asegurarse de la versión de <b>phpX.X</b> instalada en el sistema y modificarla en la directiva fastcgi_pass sino coincidiera.  <b>El sitio web (host virtual) quedaría de la siguiente manera:</b>  <pre>server { listen 80 default_server; root /var/www/html; index index.php index.html index.htm index.nginx-debian.html; server_name _; location / {</pre> </p></p>

	<p><b>Instalación Otras versiones PHP (No necesario)</b></p> <pre>apt install php8.2 apt install php7.4 apt install php5.6</pre> <p><b>Nota:</b> En LAMP apache2 hara el papel de servidor web y servidor de aplicaciones PHP</p> <p><b>Fichero configuración módulo php:</b> /etc/apache2/mods-available/phpX.X.conf y phpX.X.load</p> <p><b>Verificación instalación php:</b> php -v</p> <p><b>Activación y desactivación módulo PHP:</b></p> <pre>a2enmod php a2dismod php a2enmod php7.4 a2dismod php7.4</pre> <p><b>LAMP se puede instalar también de manera AUTOMÁTICA (NO RECOMENDABLE):</b></p> <pre>apt update apt install -y lamp-server mysql_secure_installation</pre>	<pre>try_files \$uri \$uri/ =404; } location ~ /\.php\$ {     include snippets/fastcgi-php.conf;     fastcgi_pass unix:/var/run/php/php8.2-fpm.sock; } location ~ /\.ht {     deny all; } }</pre> <p><b>Paso 5: Reiniciamos servicio nginx</b> #systemctl restart nginx</p> <p><b>Paso 6: Test/ comprobación de LEMP stack:</b> Creamos en el “documentroot” del sitio web por defecto (/var/www/html) una página de prueba: info.php para testear php: <b>echo "&lt;?php phpinfo(); ?&gt;" &gt; info.php</b> Desde el navegador e incluimos la siguiente URL: <b>http://ip_servidor/info.php</b></p> <p style="text-align: center;"><u><b>ANEXOS:</b></u></p> <p><b>Versión PHP:</b> php -v</p>
<b>Instalación y configuración Mysql Server y Mariadb</b>	<p style="text-align: center;"><b>MARIADB</b></p> <p><b>Paso 1: Actualización repositorios:</b> apt update</p> <p><b>Paso 2: Instalación servidor de base de datos y cliente</b> apt install -y mariadb-server mariadb-client apt install -y mariadb-server mariadb (alternativo)</p> <p><b>Paso 3: Acceso a MariaDB desde consola servidor (como root)</b> #mariadb #mariadb -u root</p> <p><b>Paso 4: Cambiar la contraseña de root</b> MariaDB&gt; <b>ALTER USER 'root'@'localhost' identified BY 'nueva_contraseña';</b> MariaDB&gt; flush privileges; MariaDB&gt; exit:</p> <p style="text-align: center;"><u><b>ANEXOS</b></u></p> <p><b>Iniciar, parar y consultar el estado del servicio</b> systemctl status mariadb</p>	<p style="text-align: center;"><b>MYSQL SERVER</b></p> <p><b>Paso 1: Actualización repositorios:</b> apt update</p> <p><b>Paso 2: Instalación servidor de base de datos y cliente</b> apt install mysql-server mysql-client -y</p> <p><b>Paso 3: Acceso a MySQL Server desde consola servidor (como root)</b> #mysql #mysql -u root</p> <p><b>Paso 4: Cambiar la contraseña de root</b> MySQL&gt; <b>ALTER USER 'root'@'localhost' identified BY 'nueva_contraseña';</b> MySQL&gt; flush privileges; MySQL &gt; exit:</p> <p style="text-align: center;"><u><b>ANEXOS</b></u></p> <p><b>Iniciar, parar y consultar el estado del servicio</b> sudo systemctl restart mysql sudo systemctl status mysql</p>

	<pre>systemctl restart mariadb</pre> <p><b>Archivos de configuración:</b></p> <pre>/etc/mysql/mariadb.cnf /etc/mysql/mariadb.conf.d/ /etc/mysql/mariadb.conf.d/50-server.cnf [mysqld] bind-address = 127.0.0.1</pre> <p><b>Conexión remota:</b></p> <pre>\$mariadb -h host -u root -p \$mariadb -h host -u instalador -p</pre> <p><b>Cambiar la contraseña de root (otras alternativas):</b></p> <p><b>Alternativa 1 (Antigua):</b></p> <pre>MariaDB&gt;ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'nueva_contraseña';</pre> <p><b>Alternativa 2 (Nueva mas segura):</b></p> <pre>MariaDB&gt;ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'nueva_contraseña';</pre> <p><b>Otra alternativa:</b></p> <pre>#mysql_secure_installation y seguir asistente (si no funciona ejecuta el siguiente comando) #mariadb_secure_installation y seguir el asistente</pre> <p><b>Crear un usuario administrador nuevo:</b></p> <pre>MariaDB&gt;CREATE USER 'instalador'@'localhost' IDENTIFIED BY 'contraseña';</pre> <p><b>Crear un usuario administrador nuevo y asignar permisos:</b></p> <pre>MariaDB&gt; GRANT ALL PRIVILEGES ON *.* to instalador identified by 'contraseña' with grant option; MariaDB&gt; flush privileges; MariaDB&gt; exit;</pre> <p><b>Crear una nueva BD y un usuario que tenga acceso a la misma:</b></p> <pre>MariaDB &gt;CREATE DATABASE newdb; MariaDB &gt;CREATE USER 'username'@'localhost' IDENTIFIED BY 'userpassword'; MariaDB &gt;GRANT ALL PRIVILEGES ON newdb.* to 'username'@'localhost'; MariaDB &gt;flush privileges; MariaDB &gt;quit;</pre>	<p><b>Archivos de configuración</b></p> <pre>/etc/mysql/mysql.cnf /etc/mysql/mysql.conf.d/ /etc/mysql/conf.d/ /etc/mysql/mysql.conf.d/mysqld.cnf [mysqld] bind-address = 127.0.0.1 ----- [mysqld] bind-address = 0.0.0.0</pre> <p><b>Archivos de log:</b></p> <pre>/var/log/mysql/error.log</pre> <p><b>Conexión remota:</b></p> <pre>\$mysql -h host -u root -p \$mysql -h host -u instalador -p</pre> <p><b>Cambiar la contraseña de root (otras maneras):</b></p> <p><b>Alternativa 1 (Antigua):</b></p> <pre>MySQL&gt; ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'nueva_contraseña';</pre> <p><b>Alternativa 2 (Nueva mas segura):</b></p> <pre>MySQL&gt; ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'nueva_contraseña';</pre> <p><b>Otra alternativa:</b></p> <pre>#mysql_secure_installation y seguir asistente (si no funciona ejecuta el siguiente comando)</pre> <p><b>Crear un usuario administrador nuevo:</b></p> <pre>MySQL&gt;CREATE USER 'instalador'@'localhost' IDENTIFIED BY 'contraseña';</pre> <p><b>Crear un usuario administrador nuevo y asignar permisos:</b></p> <pre>MySQL&gt; GRANT ALL PRIVILEGES ON *.* to instalador identified by 'contraseña' with grant option; MySQL&gt; flush privileges; MySQL&gt; exit;</pre> <p><b>Crear una nueva BD y un usuario que tenga acceso a la misma:</b></p> <pre>MySQL &gt;CREATE DATABASE newdb; MySQL &gt;CREATE USER 'username'@'localhost' IDENTIFIED BY 'userpassword'; MySQL &gt;GRANT ALL PRIVILEGES ON newdb.* to 'username'@'localhost'; MySQL &gt;flush privileges; MySQL &gt;quit;</pre>
<p><b>Instalación de PhpMyAdmin en LAMP (Administración de la BD)</b></p>	<p><b>MÉTODO 1: (OPCIÓN RECOMENDABLE)</b></p> <p>PhpMyadmin es una aplicación web escrita en PHP que nos posibilita la gestión de una base de datos mysql/mariadb. Normalmente vamos a instalar las aplicaciones web descargando directamente el código de la aplicación al servidor.</p> <p><b>Paso 1: Instalación PHPMyAdmin:</b></p> <pre>apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl -y</pre>	

**Paso 2: IMPORTANTE!!** Durante el proceso de instalación se debe elegir el servidor web: **apache2**



**Paso 3:** Confirma que desea utilizar **dbconfig-common** para configurar la base de datos

**Paso 4:** Finalmente se solicitará una contraseña para phpMyadmin, por ejemplo, usuario@1

**Paso 5:** Acceso a PHPMyAdmin: **http://ip\_host/phpmyadmin/**

## ANEXOS

A continuación se describe brevemente qué es lo que contiene cada uno de ellos:

- php-mbstring: Incluye el módulo mbstring (multi-byte string) que permite administrar cadenas no-ASCII y convertir cadenas a diferentes codificaciones.
- php-zip: Permite la carga de archivos .zip a phpMyAdmin.
- php-gd: Incluye la librería GD Graphics que permite crear y modificar imágenes.
- php-json: Añade soporte para trabajar con el formato JSON desde PHP.
- php-curl: Permite interactuar con servidores haciendo uso de diferentes protocolos desde PHP.

## ¿CÓMO PODRÍAMOS AUTOMATIZAR ESTA INSTALACIÓN DESDE UN SCRIPT DE BASH?

Si quisiéramos automatizar el proceso de instalación de phpMyAdmin con apt tendríamos que hacer uso de la herramienta **debconf-set-selections**.

Las distribuciones Linux basadas en Debian utilizan la utilidad **debconf** para hacer preguntas al usuario durante el proceso de instalación de un paquete. En el caso de phpmyadmin las respuestas que queremos automatizar son las siguientes:

Seleccionar el servidor web que queremos configurar para ejecutar.

```
echo "phpmyadmin phpmyadmin/reconfigure-webserver multiselect apache2" | debconf-set-selections
```

Confirmar que desea utilizar dbconfig-common para configurar la base de datos.

```
echo "phpmyadmin phpmyadmin/dbconfig-install boolean true" | debconf-set-selections
```

Seleccionar la contraseña para phpMyAdmin.

```
echo "phpmyadmin phpmyadmin/mysql/app-pass password $PHPMYADMIN_APP_PASSWORD" | debconf-set-selections
```

```
echo "phpmyadmin phpmyadmin/app-password-confirm password $PHPMYADMIN_APP_PASSWORD" | debconf-set-selections
```

En este ejemplo estamos guardando la contraseña en la variable de entorno \$PHPMYADMIN\_APP\_PASSWORD que tendremos que configurar previamente asignándole un valor inicial.

**Nota:** Necesitamos definir un archivo, por ejemplo, **.env**, donde definimos las variables:

```
PHPMYADMIN_APP_PASSWORD=usuario@1
```

**Instalación de  
PhpMyAdmin en**

## MÉTODO 2: Instalación MANUAL de PHPMyAdmin (OPCIÓN NO RECOMENDABLE)

En esta sección se explica los pasos que hay que llevar a cabo para instalar WordPress en directorio raíz de Apache. Por ejemplo:

<b>LAMP (Administración de la BD)</b>	<p>/var/www/html.que permiten añadir nuevas sus funcionalidades y personalizar su diseño de una forma muy sencilla.</p> <p><b>Paso 1: CREACIÓN Y CONFIGURACIÓN DE LA BD</b></p> <pre>\$mysql -u root -p MySQL&gt;create database <b>phpmyadmin</b>; MySQL&gt;use <b>phpmyadmin</b>; MySQL&gt;create user '<b>phpmyadmin</b>'@'localhost' identified by '<b>usuario@1</b>'; MySQL&gt;drop user '<b>phpmyadmin</b>'@'localhost'; (si tuvieras que borrar el usuario) MySQL&gt;grant all privileges on <b>phpmyadmin</b>.* to '<b>phpmyadmin</b>'@'localhost' with grant option; MySQL&gt;flush privileges; MySQL&gt;quit;</pre> <p><b>Paso 2: DESCARGA E INSTALACIÓN PHPMYADMIN</b></p> <ul style="list-style-type: none"> <li>-Descarga PHPMyAdmin desde la página oficial a <b>/var/www/html</b>:  <pre>wget https://files.phpmyadmin.net/phpMyAdmin-5.2.0/phpMyAdmin-5.2.0-all-languages.zip</pre></li> <li>-Descomprimos archivo:  <pre>unzip ....zip.</pre></li> </ul> <p><b>Nota:</b> Si no lo tienes, puedes instalar unzip en tu equipo</p> <ul style="list-style-type: none"> <li>-Renombramos carpeta:  <pre>mv phpMyAdmin-5.2.0-all-languages phpmyadmin</pre></li> <li>-Asignamos propietario.  <pre>chown -R www-data:www-data /var/www/phpmyadmin/</pre></li> <li>-En el sitio web (host virtual) por defecto u otro sitio creado “ex profeso” para la ocasión definir la siguiente directiva:  <pre>DirectoryIndex index.php index.html.....</pre></li> <li>-Iniciamos asistente para completar la configuración de PHPMyAdmin desde la instancia de Windows (host bastión):  <b><a href="http://ip_servidor1/phpmyadmin/">http://ip_servidor1/phpmyadmin/</a></b></li> </ul>
<b>Instalación de Adminer (Administración de la BD)</b>	<p><b>Adminer</b> es una utilidad de administración de BD mas sencilla que PHPMyAdmin que se caracteriza por estar formada por un único fichero PHP. Se puede descargar desde su página oficial, <a href="https://www.adminer.org/">https://www.adminer.org/</a>, crear un directorio en el sitio web, por ejemplo /var/www/html/adminer y copiar el archivo descargado. Si lo consideras oportuno, se puede renombrar el archivo a <b>index.php</b></p>
<b>Instalación de WordPress</b>	<p><b>Instalación MANUAL del CMS WORDPRESS:</b></p> <p>Un Sistema de Gestión de Contenidos (CMS) es un software que permite a los usuarios crear, editar y gestionar de una forma sencilla el contenido de un sitio web. Algunos ejemplos de CMS son: WordPress, Joomla y Drupal.</p> <p>WordPress es un sistema de gestión de contenidos (CMS) muy utilizado para crear sitios webs y blogs. Es software libre y gratuito, y está desarrollado en PHP. Además cuenta con una gran cantidad de plugins y temas que permiten añadir nuevas sus funcionalidades y personalizar su diseño de una forma muy sencilla.</p> <p>En esta sección se explica los pasos que hay que llevar a cabo para instalar WordPress en directorio raíz de Apache. Por ejemplo: /var/www/html.que permiten añadir nuevas sus funcionalidades y personalizar su diseño de una forma muy sencilla.</p> <p><b>Paso 1: Instalación otros paquetes php necesarios</b>  <pre>apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip</pre></p> <p><b>Paso 2: Creación y configuración de la base de datos:</b></p>

```
$mysql -u root -p
MySQL >create database wordpress;
MySQL >use wordpress;
MySQL >create user 'wordpress'@'localhost' identified by 'usuario@1';
MySQL >grant all privileges on wordpress.* to 'wordpress'@'localhost' with grant option;
MySQL >flush privileges;
MySQL>quit;
```

**Paso 2: (OTRA ALTERNATIVA):**

```
MySQL > create database wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci; (otra alternativa)
MySQL >create user 'wordpress@'%' identified by 'usuario@1'; (otra manera)
MySQL >grant all on wordpress.* TO 'wordpress'@'%' ; (otra manera)
MySQL >GRANT ALL ON wordpress.* TO ' wordpress'@'localhost' IDENTIFIED BY 'usuario@1'; (otra manera: se crea el usuario
y se le asigna privilegios: todo en uno)
```

## ¿CÓMO PODRÍAMOS AUTOMATIZAR ESTA INSTALACIÓN DESDE UN SCRIPT DE BASH?

```
mysql -u root <<< "DROP DATABASE IF EXISTS $WORDPRESS_DB_NAME"
mysql -u root <<< "CREATE DATABASE $WORDPRESS_DB_NAME"
mysql -u root <<< "DROP USER IF EXISTS $WORDPRESS_DB_USER@$IP_CLIENTE_MYSQL"
mysql -u root <<< "CREATE USER $WORDPRESS_DB_USER@$IP_CLIENTE_MYSQL IDENTIFIED BY
'$WORDPRESS_DB_PASSWORD'"
#mysql -u root <<< "GRANT ALL PRIVILEGES ON $WORDPRESS_DB_NAME.* TO
$WORDPRESS_DB_USER@$IP_CLIENTE_MYSQL"
```

**Nota:** Necesitamos definir un archivo, por ejemplo, **.env**, donde definimos las variables:

```
PHPMYADMIN_APP_PASSWORD=usuario@1
WORDPRESS_DB_NAME=wordpress
WORDPRESS_DB_USER=wordpress
WORDPRESS_DB_PASSWORD=usuario@1
IP_CLIENTE_MYSQL=localhost
```

**Paso 3: Configuración APACHE 2**

- Vamos a utilizar el sitio por defecto: 000-default.conf

-Editamos **000-default.conf** y añadimos las siguientes directivas:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    DirectoryIndex index.html index.php
    <Directory /var/www/html/>
        AllowOverride All
    </Directory>
</VirtualHost>
```

-Ahora, se activa el mod\_rewrite para utilizar la función de permalink o enlace permanente de WordPress:

### **a2enmod rewrite**

- Tendrás que reiniciar el servidor web Apache utilizando el siguiente comando:  
systemctl restart apache2

## **¿CÓMO PODRÍAMOS AUTOMATIZAR ESTA INSTALACIÓN DESDE UN SCRIPT DE BASH?**

Se tendría que añadir al archivo **000-default.conf** las siguientes líneas.

```
DirectoryIndex index.html index.php
<Directory /var/www/html/>
    AllowOverride All
</Directory>
```

**Nota:** Recuerda utilizar el comando sed. Ejemplo: **sed -i '3 i texto a insertar' archivo.sh**

A través del comando anterior insertamos en la 3ª línea de archivo.txt un texto

### **Paso 4: Descarga e instalación WORDPRESS**

Una vez completados todos los preparativos, es el momento de instalar WordPress. Hay dos métodos: configurar WordPress a través de una interfaz web o editar manualmente el archivo wp-config.php. **Nosotros lo vamos a hacer a través de la interfaz web.**

- Instala el paquete **wget** en tu MV. Esto será útil para descargar los archivos de WordPress. Ejecuta este comando en la línea de comandos:

```
#apt install wget -y
```

- A continuación, utiliza el comando wget seguido del enlace de descarga de WordPress:

```
wget https://es.wordpress.org/latest-es_ES.zip -P /tmp
```

**Nota:** La opción -P /tmp descarga el comprimido en/tmp

- Una vez que hayas descargado el archivo comprimido, instala la utilidad de descompresión **unzip** utilizando estos comandos:

```
# apt install unzip -y
```

- El contenido se ha descomprimido en una carpeta que se llama wordpress. Ahora, movemos el contenido de /tmp/wordpress a /var/www/html.

```
mv -f /tmp/wordpress/* /var/www/html
```

- Cambio el usuario propietario del documentRoot:

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

- Reinicia el servicio Apache2:

```
sudo systemctl restart apache2
```

- Termina de configurar WordPress a través de un navegador web. Abre un navegador web y escribe la dirección IP del servidor.

**http://ip\_servidor1/**

- Una vez que hayas iniciado la sesión, accederás al panel de administración de WordPress. Ahora puedes empezar a personalizar el sitio web instalando plugins y temas de WordPress. (ver figuras)



A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos

El nombre de la base de datos que quieres usar con WordPress.

Nombre de usuario

El nombre de usuario de tu base de datos.

Contraseña

[Ocultar](#)

La contraseña de tu base de datos.

Servidor de la base de datos

Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

Prefijo de tabla

Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)





¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

Nombre de  
usuario

usuario

Contraseña

La contraseña que has elegido.

Acceder



Nombre de usuario o correo electrónico

usuario

Contraseña

•••••

☐ Recuérdame

Acceder

¿Has olvidado tu contraseña?

← Ir a Sitio WordPress LAMP



Español

Cambiar

Hola

¡Este es el famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

### Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	<input type="text" value="Sitio WordPress LAMP"/>
Nombre de usuario	<input type="text" value="usuario"/> <small>Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.</small>
Contraseña	<input type="password" value="usuario@1"/> <a href="#">Ocultar</a> <div>Muy débil</div> <p><b>Importante:</b> Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.</p>
Confirma la contraseña	<input checked="" type="checkbox"/> Confirma el uso de una contraseña débil.
Tu correo electrónico	<input type="text" value="usuario@gmail.com"/> <small>Comprueba bien tu dirección de correo electrónico antes de continuar.</small>
Visibilidad en los motores de búsqueda	<input type="checkbox"/> Pedir a los motores de búsqueda que no indexen este sitio <small>Depende de los motores de búsqueda atender esta petición o no.</small>
<input type="button" value="Instalar WordPress"/>	

## ¿CÓMO PODRÍAMOS AUTOMATIZAR ESTA INSTALACIÓN DESDE UN SCRIPT DE BASH?

1.- Preparamos la configuración para los enlaces permanentes de WordPress. En este paso tendremos que crear un archivo **.htaccess** en el directorio `/var/www/html` con un contenido similar a este.

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule . /index.php [L]  
</IfModule>  
# END WordPress
```

2.- Creamos un archivo de configuración **wp-config.php** a partir del archivo de ejemplo wp-config-sample.php.

```
cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
```

En este paso tenemos que configurar las variables de configuración del archivo de configuración de WordPress. El contenido original del archivo wp-config.php será similar a este:

```
define( 'DB_NAME', 'database_name_here' );  
define( 'DB_USER', 'username_here' );  
define( 'DB_PASSWORD', 'password_here' );  
define( 'DB_HOST', 'localhost' );
```

Por lo tanto, lo que haremos será reemplazar el texto database\_name\_here, username\_here, password\_here y localhost por los valores de las variables \$WORDPRESS\_DB\_NAME, \$WORDPRESS\_DB\_USER, \$WORDPRESS\_DB\_PASSWORD y \$WORDPRESS\_DB\_HOST respectivamente. Para realizar este paso utilizaremos el comando sed:

```
sed -i "s/database_name_here/$WORDPRESS_DB_NAME/" /var/www/html/wp-config.php  
sed -i "s/username_here/$WORDPRESS_DB_USER/" /var/www/html/wp-config.php  
sed -i "s/password_here/$WORDPRESS_DB_PASSWORD/" /var/www/html/wp-config.php  
sed -i "s/localhost/$WORDPRESS_DB_HOST/" /var/www/html/wp-config.php
```

Tenga en cuenta que las variables \$WORDPRESS\_DB\_NAME, \$WORDPRESS\_DB\_USER, \$WORDPRESS\_DB\_PASSWORD y \$WORDPRESS\_DB\_HOST estarán definidas en el archivo .env, por ejemplo:

```
PHPMYADMIN_APP_PASSWORD=usuario@1  
WORDPRESS_DB_NAME=wordpress  
WORDPRESS_DB_USER=wordpress  
WORDPRESS_DB_PASSWORD=usuario@1  
IP_CLIENTE_MYSQL=localhost
```