

# Guía 01: Instalación de Pila LAMP y Despliegue de CMS

**Objetivo:** Configurar un servidor Linux desde cero para alojar aplicaciones web (WordPress, Moodle, etc.) usando Apache, MariaDB y PHP.

## 1. Preparación del Sistema

Antes de instalar nada, actualiza los repositorios. Esto evita conflictos de versiones.

```
# Actualizar lista de paquetes y sistema  
sudo apt update && sudo apt upgrade -y
```

## 2. Instalación de la Pila LAMP

### A. Apache (Servidor Web)

```
sudo apt install apache2 -y
```

```
# Verificar estado  
sudo systemctl status apache2
```

```
# Habilitar inicio automático  
sudo systemctl enable apache2
```

**Tip:** Si accedes a la IP de tu servidor en el navegador, deberías ver la página “Apache2 Ubuntu Default Page”.

### B. MariaDB (Base de Datos)

Preferimos MariaDB sobre MySQL por ser totalmente open-source y compatible.

```
sudo apt install mariadb-server -y
```

```
# Script de seguridad OBLIGATORIO  
sudo mysql_secure_installation
```

**Respuestas recomendadas para mysql\_secure\_installation:** - Switch to unix\_socket authentication? **Y** - Change the root password? **Y** (Pon una segura) - Remove anonymous users? **Y** - Disallow root login remotely? **Y** - Remove test database? **Y** - Reload privilege tables now? **Y**

### C. PHP y Extensiones Comunes

Los CMS modernos necesitan muchas librerías. Instala este pack estándar para evitar errores después.

```
sudo apt install php libapache2-mod-php php-mysql php-cli php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip -y
```

```
# Reiniciar Apache para cargar PHP  
sudo systemctl restart apache2
```

## 3. Configuración de Base de Datos para CMS

Nunca uses el usuario `root` en la configuración del CMS. Crea uno específico.

```
-- Entrar a la consola  
sudo mysql -u root -p
```

```
-- Comandos SQL  
CREATE DATABASE nombre_db_cms CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE USER 'usuario_cms'@'localhost' IDENTIFIED BY 'password_segura';
```

```
GRANT ALL PRIVILEGES ON nombre_db_cms.* TO 'usuario_cms'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

---

## 4. Configuración de VirtualHost (Apache)

No uses la configuración por defecto (000-default.conf). Crea uno por sitio.

**Archivo:** /etc/apache2/sites-available/mi-sitio.conf

```
<VirtualHost *:80>
    ServerAdmin admin@midominio.com
    ServerName midominio.com
    ServerAlias www.midominio.com
    DocumentRoot /var/www/mi-sitio

    <Directory /var/www/mi-sitio>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

**Activar el sitio:**

```
# Crear directorio
sudo mkdir -p /var/www/mi-sitio

# Habilitar sitio y módulo rewrite (vital para URLs amigables)
sudo a2ensite mi-sitio.conf
sudo a2enmod rewrite
sudo systemctl restart apache2
```

---

## 5. Instalación del CMS (Ejemplo Genérico)

- Descargar:** Ve a /var/www/mi-sitio y descarga el .zip o .tar.gz oficial (usando wget).
- Descomprimir:** unzip archivo.zip O tar -xvf archivo.tar.gz. Asegúrate que los archivos queden en la raíz de /var/www/mi-sitio, no en una subcarpeta.
- Permisos (CRÍTICO):** Apache necesita escribir en ciertos directorios (uploads, cache).

```
# Asignar propietario al usuario de Apache (www-data)
sudo chown -R www-data:www-data /var/www/mi-sitio

# Permisos estándar: Carpetas 755, Archivos 644
sudo find /var/www/mi-sitio -type d -exec chmod 755 {} \;
sudo find /var/www/mi-sitio -type f -exec chmod 644 {} \;
```

## 6. Finalización

Abre tu navegador, ve a tu dominio/IP y sigue el instalador web del CMS. Usa las credenciales de BD creadas en el paso 3.