

Tarefa 2: Instalación e Configuración dun Servidor LAMP con phpMyAdmin

Carlos Martínez López

Obxectivos da Práctica

- Comprender o concepto de contorno de servidor web e a arquitectura LAMP (Linux, Apache, MariaDB, PHP).
- Adquirir destreza na instalación de paquetes e xestión de servizos en Ubuntu a través da terminal.
- Configurar correctamente un servidor web Apache, incluíndo a creación de Virtual Hosts.
- Instalar e asegurar un servidor de bases de datos MariaDB.
- Instalar PHP e os módulos necesarios para a súa integración con Apache e MariaDB.
- Instalar e configurar a ferramenta de xestión de bases de datos phpMyAdmin.
- Verificar o correcto funcionamento de todos os componentes do contorno.

Contexto

Como futuros desenvolvedores de aplicacións web, é fundamental que sexades capaces de montar, configurar e xestionar o voso propio contorno de desenvolvemento e producción. Esta práctica guiaravos a través do proceso de posta en marcha dun dos *stacks* más populares da industria, preparándovos para o despregamento de aplicacións web dinámicas.

Requisitos Previos

- Unha máquina virtual con Ubuntu 24.04 con acceso a internet.
- Acceso á terminal cun usuario con privilexios de `sudo`.

Fases da Práctica

Fase 1: Preparación do Sistema e Instalación de Apache

Nesta primeira fase, actualizaremos o sistema e instalaremos o servidor web Apache.

- 1. Actualización do Sistema:**

Abre unha terminal e executa os seguintes comandos para asegurar que todos os paquetes

do sistema están actualizados:

```
sudo apt update  
sudo apt upgrade -y
```

2. Instalación de Apache:

Instala o servidor web Apache2.

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

3. Configuración do Firewall:

Ubuntu trae por defecto o firewall UFW. Debemos permitir o tráfico web a través del. Apache rexistra varios perfís en UFW.

- Apache: Abre o porto 80 (tráfico non cifrado).
- Apache Secure: Abre o porto 443 (tráfico cifrado con TLS).
- Apache Full: Abre ambos os portos.

Permite o perfil Apache Full e activa o firewall:

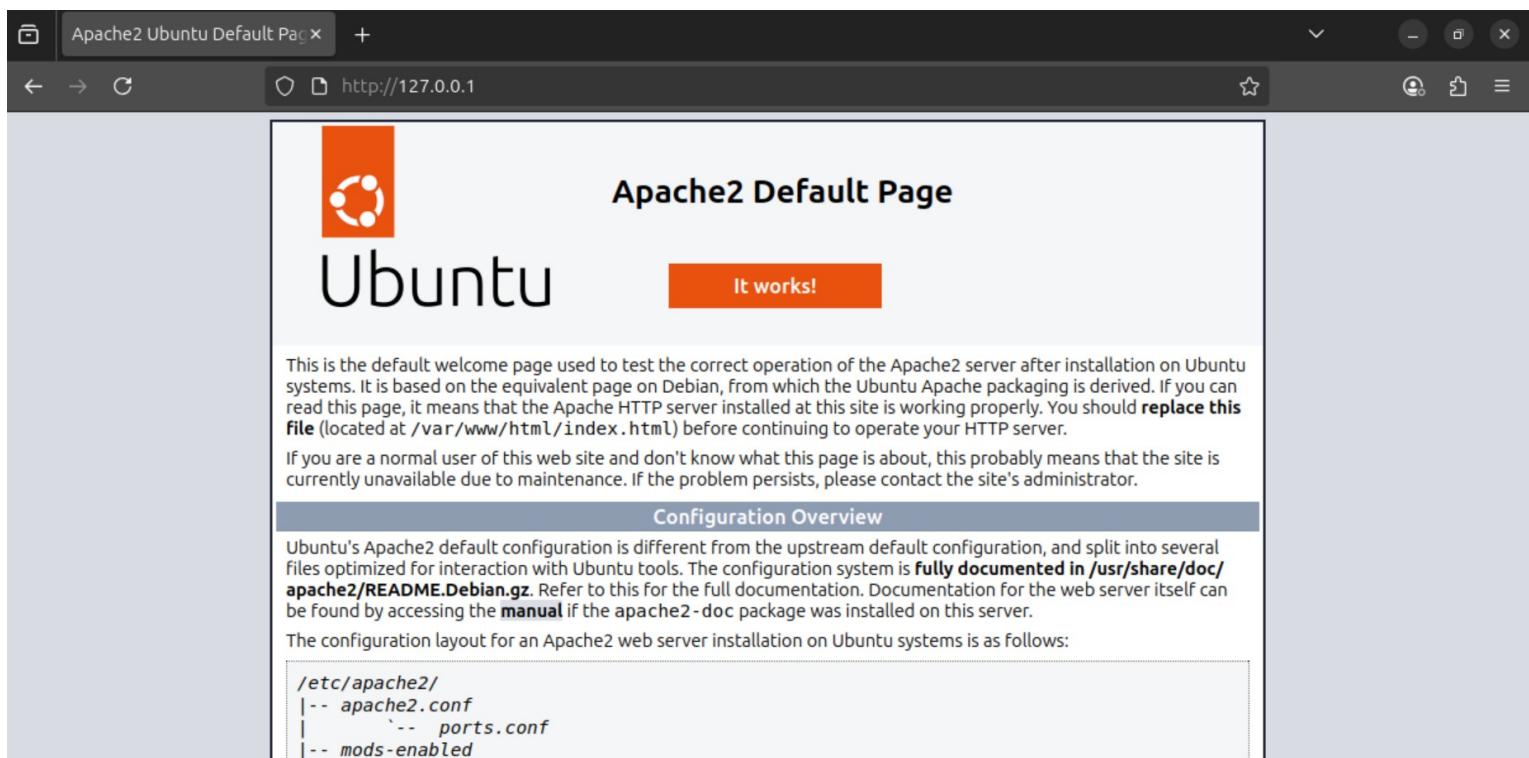
```
sudo ufw allow 'Apache Full'  
sudo ufw enable
```

4. Verificación do Servidor Web:

Comproba que o servizo de Apache está activo.

```
sudo systemctl status apache2
```

Abre un navegador web e introduce o endereço IP da túa máquina virtual. Deberías ver a páxina de benvinda por defecto de Apache2.



Comandos importantes de esta fase:

```
sudo apt install apache2 -y: para instalar el servidor  
sudo ufw allow 'Apache Full'  
sudo ufw enable                                <- para habilitar los permisos
```

Problemas atopados:

Ningun problema atopado.

Fase 2: Instalación do Servidor de Bases de Datos MariaDB

Agora instalaremos MariaDB, un sistema de xestión de bases de datos relacional de código aberto, compatible con MySQL.

1. Instalación de MariaDB:

Executa o seguinte comando para instalar o servidor e o cliente de MariaDB:

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

2. Asegurando a Instalación de MariaDB:

MariaDB inclúe un script para mellorar a seguridade da instalación por defecto.

```
sudo mysql_secure_installation
```

Responde ás preguntas que che fará o script. Recoméndase:

- Establecer un contrasinal para o usuario `root` da base de datos.
- Eliminar usuarios anónimos.
- Desactivar o inicio de sesión remoto do usuario `root`.
- Eliminar a base de datos de proba.
- Recargar os privilexios.

3. Verificación da Base de Datos:

Comproba que podes acceder á consola de MariaDB co usuario `root` e o contrasinal que acabas de establecer.

```
mysql -u root -p
```

Unha vez dentro, podes saír escribindo `exit;.`

Cleaning up...

```
All done! If you've completed all of the above steps, your MariaDB installation should now be secure.
```

```
Thanks for using MariaDB!
```

```
administrador@daw:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \
g.
Your MariaDB connection id is 47
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> exit;
Bye
administrador@daw:~$
```

Comandos importantes de esta fase:

```
sudo apt install mariadb-server mariadb-client -y: instalar a base de datos
sudo mysql_secure_installation: configurar instalación
```

Problemas atopados:

Tiven un problema ao introducir o contrasinal pero foi solucionado volvendo a introducir o comando da instalación e volvendo a comenzala.

Fase 3: Instalación de PHP

PHP é a linguaxe de programación do lado do servidor que usaremos para crear contido dinámico. Instalaremos a versión máis recente disponible nos repositorios de Ubuntu 24.04 xunto con algúns módulos esenciais.

1. Instalación de PHP e Módulos:

Instala PHP e os módulos para a integración con Apache e MariaDB.

```
sudo apt install php libapache2-mod-php php-mysql -y
```

Que paquetes instala a maiores xenerados polas dependencias dos indicados para instalar na liña de comandos. Que uso teñen dentro do noso setup?

libaio1t64 libevent-core-2.1-7t64, permite utilizar llamadas al sistema e/s del kernel de linux al espacio de usuario que es importante para bases de datos y otras aplicaciones.

libevent-pthreads-2.1-7t64 libmecab2, proporciona un mecanismo para ejecutar una función de devolución de llamada cuando se produce un evento específico en un descriptor de archivo o después de que se haya alcanzado un tiempo de espera.

libprotobuf-lite32t64 mecab-ipadic mecab-ipadic-utf8, contiene liberias y headers necesarias para aplicaciones escritas en c++

mecab-utils, una librería de código abierto para la segmentación del texto en japones.

2. Verificación da Instalación de PHP:

Para confirmar que Apache está a procesar correctamente os ficheiros PHP, crea un ficheiro de prueba no directorio raíz do servidor web.

```
sudo nano /var/www/html/info.php
```

Engade o seguinte contido ao ficheiro:

```
<?php  
phpinfo();  
?>
```

Garda e pecha o ficheiro. Agora, no teu navegador, visita http://<A_TÚA_IP>/info.php. Deberías ver unha páxina con toda a información sobre a túa instalación de PHP.

IMPORTANTE: Por motivos de seguridade, é crucial eliminar este ficheiro despois de verificalo.

```
sudo rm /var/www/html/info.php
```

PHP Version 8.3.6	
System	Linux daw 6.14.0-29-generic #29~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Aug 14 16:52:50 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlind.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/15-xml.ini, /etc/php/8.3/apache2/conf.d/20-bz2.ini

Comandos importantes de esta fase:

```
sudo apt install php libapache2-mod-php php-mysql -y: instalación  
sudo nano /var/www/html/info.php: crear pagina de info  
sudo rm /var/www/html/info.php: eliminar por motivos de seguridad
```

Problemas atopados:

Sen problemas atopados.

Fase 4: Instalación e Configuración de phpMyAdmin

Para facilitar a xestión das bases de datos, instalaremos phpMyAdmin, unha popular ferramenta web.

1. Instalación de phpMyAdmin:

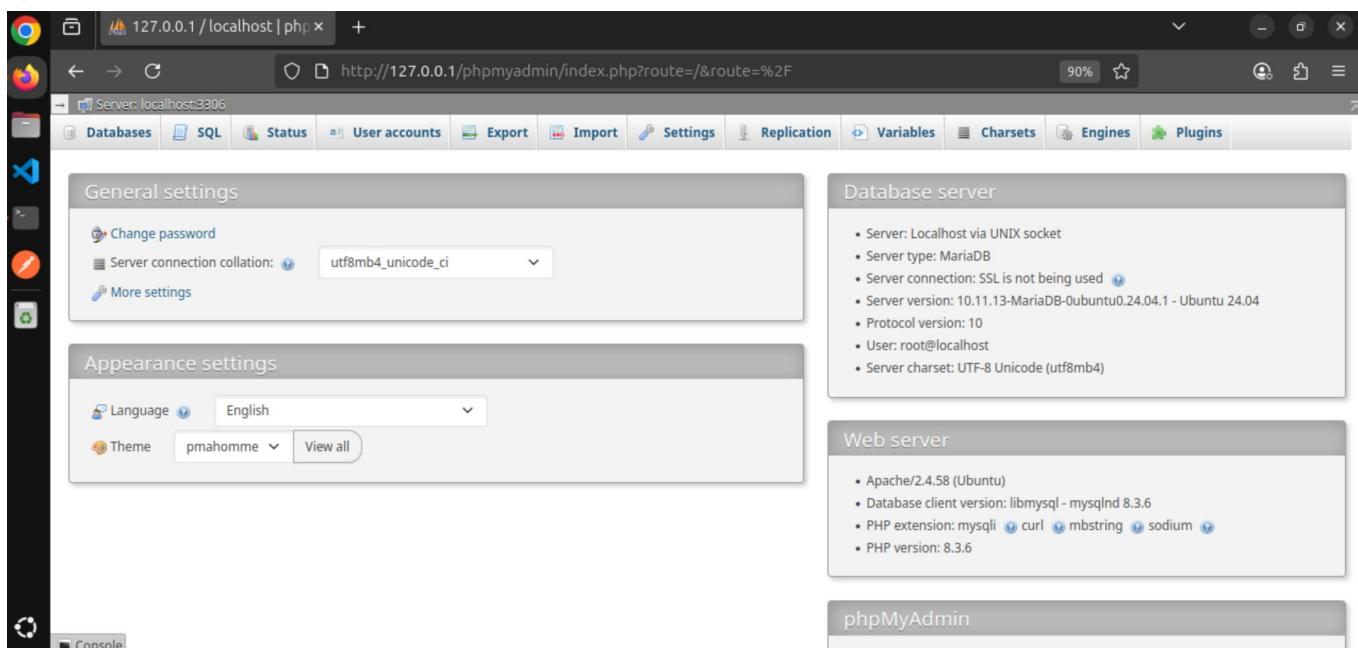
O paquete está disponible nos repositorios oficiais.

```
sudo apt install phpmyadmin -y  
``` Durante a instalación, aparecerán varias pantallas de configuración:  
* **Selección do servidor web:** Elixe `apache2` premendo a barra espaciadora e logo `Intro`.
* **Configurar base de datos para phpmyadmin:** Selecciona `Si` para permitir que o instalador configure a súa propia base de datos.
* **Contrasinal para o usuario `phpmyadmin`:** Establece un contrasinal para o usuario que phpMyAdmin usará para conectarse a MariaDB.
```

### **2. Acceso a phpMyAdmin:**

Unha vez rematada a instalación, poderás acceder a phpMyAdmin a través do teu navegador na seguinte URL: [http://<A\\_TÚA\\_IP>/phpmyadmin](http://<A_TÚA_IP>/phpmyadmin).

Poderás iniciar sesión co usuario `root` de MariaDB e o contrasinal que definiches na Fase 2.



### **Comandos importantes de esta fase:**

```
sudo apt install phpmyadmin -y: instalacion de la configuración de la base de datos
```

### **Problemas atopados:**

Nesta fase como xa estaba instalado tivemos que facer un “purge” ao comando anterior para borrar o programa e volvelo a instalar con la configuración que queremos.

### **\*\*Fase 5: Creación dun Virtual Host**

Para aloxar múltiples sitios web nun mesmo servidor, utilizanse os "Virtual Hosts". Nesta fase, crearás un para un dominio de proba chamado proxecto.daw.

#### **1. Crear a Estrutura de Directrios:**

```
sudo mkdir -p /var/www/proxecto.daw
```

#### **2. Asignar Permisos:**

```
sudo chown -R $USER:$USER /var/www/proxecto.daw
```

#### **3. Crear unha Páxina de Exemplo:**

```
nano /var/www/proxecto.daw/index.html
```

Engade o seguinte contido:

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="UTF-8">
 <title>Benvido a Proxecto DAW!</title>
</head>
<body>
 <h1>Éxito! O Virtual Host para proxecto.daw funciona correctamente!
</h1>
</body>
</html>
```

#### **4. Crear o Ficheiro de Configuración do Virtual Host:**

```
sudo nano /etc/apache2/sites-available/proxecto.daw.conf
```

Pega a seguinte configuración:

```
<VirtualHost *:80>
 ServerAdmin admin@proxecto.daw
 ServerName proxecto.daw
 ServerAlias www.proxecto.daw
 DocumentRoot /var/www/proxecto.daw
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

#### **5. Activar o Novo Sitio e Desactivar o Predeterminado:**

```
sudo a2ensite proxecto.daw.conf
```

```
sudo a2dissite 000-default.conf
```

## 6. Verificar a Configuración e Reiniciar Apache:

```
sudo apache2ctl configtest
sudo systemctl restart apache2
```

## 7. Modificar o Ficheiro hosts Local:

Para que o teu sistema local poida resolver o dominio proxecto.daw, edita o ficheiro hosts na túa máquina virtual e engade a seguinte liña:

```
<A_TÚA_IP_DA_VM> proxecto.daw
```

Agora, ao visitar <http://proxecto.daw> no navegador, deberías ver a páxina de exemplo que creaches.



### Comandos importantes de esta fase:

```
sudo chown -R $USER:$USER /var/www/proxecto.daw : asignar los permisos
adecuados
sudo nano /etc/apache2/sites-available/proxecto.daw.conf: crear paxina
exemplo
sudo nano /etc/apache2/sites-available/proxecto.daw.conf: crear
ficheiro configuración
sudo a2ensite proxecto.daw.conf: activar novo sitio
sudo a2dissite 000-default.conf : desactivar sitio predeterminado
```

### Problemas atopados:

Sen problemas atopados.

### Ampliación: Un Módulo para Cada Necesidade - Instalando Extensións de PHP

**Oxectivo:** Comprender que PHP é modular e que moitas veces é necesario instalar extensións adicionais para engadir funcionalidades (procesamento de imaxes, conexións a APIs, etc.).

## Descripción:

1. Imos simular que a túa aplicación necesita xerar imaxes dinamicamente. Para iso, PHP necesita a extensión **GD**.
2. Primeiro, verifica que non a tes instalada. Revisa a túa páxina `phpinfo()` (recórdate de creala de novo temporalmente) e busca a sección "gd". Probablemente non a atoparás.
3. Investiga cal é o nome do paquete de Ubuntu para instalar a extensión GD de PHP.
4. Instala o paquete usando `apt`.
5. Despois de instalar unha extensión de PHP, que servizo debes reiniciar para que os cambios teñan efecto?  
El de apache con “`service apache2 restart`”, para que surga efecto.
6. Verifica de novo coa páxina `phpinfo()` que a sección "gd" agora aparece, o que confirma que a extensión está activa e lista para ser usada.

## Pistas para a investigación:

- Usa o comando `apt search` para buscar paquetes relacionados con PHP. Proba con `apt search php-gd`.
- O nome do paquete adoita incluír a versión de PHP, por exemplo `php8.3-gd`. Podes verificar a túa versión de PHP co comando `php -v`.

## Entregable da ampliación:

- Captura de pantalla da páxina `phpinfo()` amosando claramente a sección "gd" activada despois da instalación.

FTP support	enabled
FTPS support	enabled
<b>gd</b>	
GD Support	enabled
GD headers Version	2.3.3
GD library Version	2.3.3
FreeType Support	enabled
FreeType Linkage	with freetype
GIF Read Support	enabled
GIF Create Support	enabled
JPEG Support	enabled
PNG Support	enabled
WBMP Support	enabled
XPM Support	enabled
XBM Support	enabled
WebP Support	enabled
BMP Support	enabled
TGA Read Support	enabled

- O GD ya estaba instalado ,es un acrónimo que significa “Graphic Draw”. La biblioteca

GD en PHP le permite crear datos de imágenes y es la herramienta de procesamiento de imágenes más utilizada en PHP.

### ***Entrega da Práctica***

Para avaliar a correcta realización da práctica debes entregar un documento que inclúa:

1. Capturas de pantalla que demostren o éxito de cada fase:
  - A páxina de benvida de Apache.
  - A conexión exitosa á consola de MariaDB.
  - A páxina xerada por `phpinfo()`.
  - A páxina de inicio de sesión de phpMyAdmin.
  - A páxina de benvida do Virtual Host `proxecto.daw`.
2. Un breve resumo dos comandos más importantes utilizados en cada fase e unha explicación da súa función.
3. Unha reflexión sobre os problemas atopados e como foron solucionados.

Ten en conta que as capturas de pantalla de terminais, código, etc deben ter un fondo branco e as imaxes deben ter un tamaño que facilite a lectura do texto cando sexa necesario. Con tal fin, pode que precises facer algunha captura da parte na que está o texto de interés.

Entrega un documento pdf co seguinte nome: U1\_T2\_OTeuNomeDeUsuario.pdf