

# Instalación y Configuración de un Servidor de Aplicaciones Web en Windows y Linux



**AUTOR:** Pedro Antonio Giménez Meroño

## Índice

<b>1. Introducción.....</b>	<b>3</b>
1.2 Objetivos.....	3
1.3 Elección del Servidor de Aplicaciones Web y configuración.....	3
1.3.1 Resumen archivos de configuración Apache2 en WAMP.....	3
1.3.2 Resumen archivos de configuración MySQL en WAMP.....	4
1.3.3 Resumen archivos de configuración PHP en WAMP.....	4
1.3.4 Resumen archivos de configuración Apache2 en LAMP.....	4
1.3.5 Resumen archivos de configuración MySQL en LAMP.....	4
1.3.6 Resumen archivos de configuración PHP en LAMP.....	5
<b>2. WAMP.....</b>	<b>5</b>
2.1 Descarga e instalación.....	5
2.2 Configuración básica.....	6
2.3 Prueba del entorno.....	12
<b>3. LAMP.....</b>	<b>12</b>
3.1 Descarga e instalación.....	12
3.2 Configuración básica para probar el entorno.....	13
3.3 Prueba del entorno.....	19
<b>4. Comparación y Reflexión.....</b>	<b>20</b>
4.1 Comparación.....	20
4.2 Reflexión.....	21

## 1. Introducción

### 1.2 Objetivos

- Repasar la instalación de máquinas virtuales con distintos sistemas operativos.
- Familiarizarse con la instalación y configuración de un servidor de aplicaciones web en distintos sistemas operativos.
- Comparar el proceso de instalación y funcionamiento en Windows y Linux.
- Entender el uso de herramientas de desarrollo local para pruebas de aplicaciones web.
- Realizar pruebas de conectividad y funcionamiento del servidor web en ambos entornos.

### 1.3 Elección del Servidor de Aplicaciones Web y configuración

Elegimos LAMP y WAMP porque ofrecen un equilibrio ideal entre facilidad de uso, control, y compatibilidad con entornos de producción. LAMP es el estándar en servidores de producción debido a su robustez, flexibilidad y rendimiento en entornos empresariales. WAMP proporciona un entorno de desarrollo fácil de manejar en Windows, que permite a los desarrolladores replicar entornos de producción basados en LAMP, pero con la facilidad de uso y las herramientas gráficas propias de Windows.

Esto asegura que los proyectos puedan desarrollarse y probarse en Windows con WAMP, y luego ser fácilmente migrados y ejecutados en LAMP en servidores de producción.

#### 1.3.1 Resumen archivos de configuración Apache2 en WAMP

- **httpd.conf**: Archivo de configuración principal de Apache.
- **httpd-vhosts.conf**: Configuración de sitios virtuales.
- **php.ini**: Archivo de configuración de PHP.
- **httpd-ssl.conf**: Configuración para HTTPS/SSL.
- **C:/wamp64/www/**: Directorio raíz de archivos del sitio web.
- **C:\wamp64\logs\**: Archivos de registro de Apache.
- **hosts**: Archivo del sistema para asignar dominios locales.
- **phpMyAdmin**: Herramienta para administrar bases de datos MySQL.

### 1.3.2 Resumen archivos de configuración MySQL en WAMP

- **C:\wamp64\bin\mysql\mysqlX.X.X\my.ini:** El archivo principal de configuración de MySQL permite ajustar los parámetros del servidor de base de datos, como la gestión de memoria, el tamaño de los logs, la codificación de caracteres y las opciones de red.

### 1.3.3 Resumen archivos de configuración PHP en WAMP

- **C:\wamp64\bin\php\phpX.X.X\php.ini:** El archivo **php.ini** permite ajustar cómo PHP maneja las solicitudes, qué módulos están cargados, y los límites de ejecución de scripts, entre otras configuraciones esenciales.

### 1.3.4 Resumen archivos de configuración Apache2 en LAMP

- **/etc/apache2/apache2.conf:** Configuración principal de Apache.
- **/etc/apache2/sites-available/** y **/etc/apache2/sites-enabled/:** Configuraciones de sitios virtuales.
- **/etc/apache2/ports.conf:** Configuración de los puertos en los que Apache escucha.
- **/etc/apache2/mods-available/** y **/etc/apache2/mods-enabled/:** Módulos de Apache disponibles y habilitados.
- **/etc/apache2/conf-available/** y **/etc/apache2/conf-enabled/:** Configuraciones adicionales para Apache.
- **/etc/apache2/envvars:** Variables de entorno para Apache.
- **/var/log/apache2/:** Archivos de registro de Apache.
- **/etc/apache2/ssl/:** Almacena certificados SSL/TLS.
- **/var/www/html/:** Directorio raíz de los archivos web.

### 1.3.5 Resumen archivos de configuración MySQL en LAMP

- **/etc/mysql/my.cnf:** El archivo principal de configuración de MySQL permite ajustar los parámetros del servidor de base de datos, como la gestión de memoria, el tamaño de los logs, la codificación de caracteres y las opciones de red.

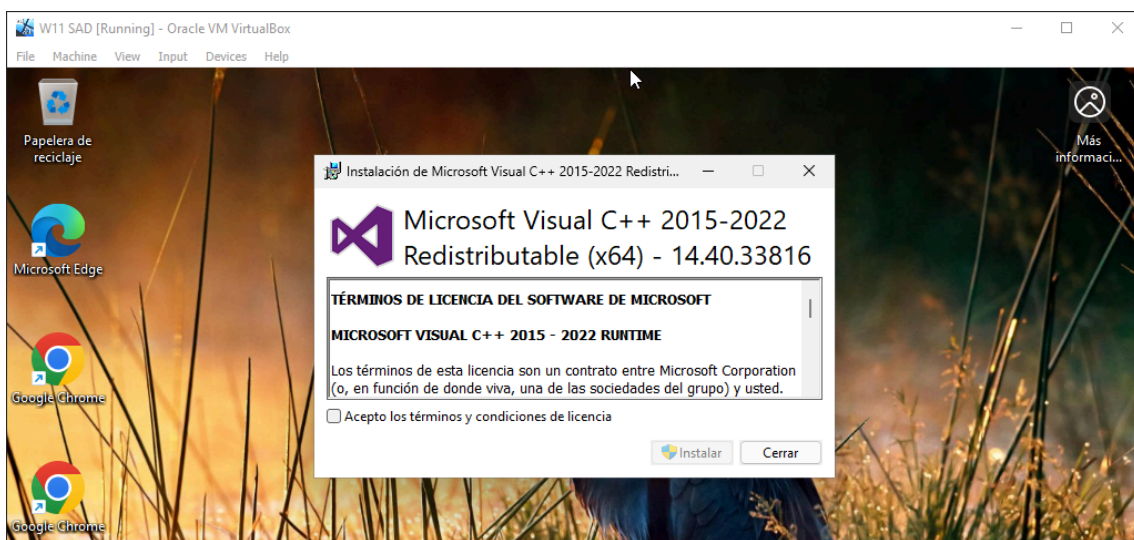
### 1.3.6 Resumen archivos de configuración PHP en LAMP

- **/etc/php/X.X/apache2/php.ini:** El archivo php.ini permite ajustar cómo PHP maneja las solicitudes, qué módulos están cargados, y los límites de ejecución de scripts, entre otras configuraciones esenciales.

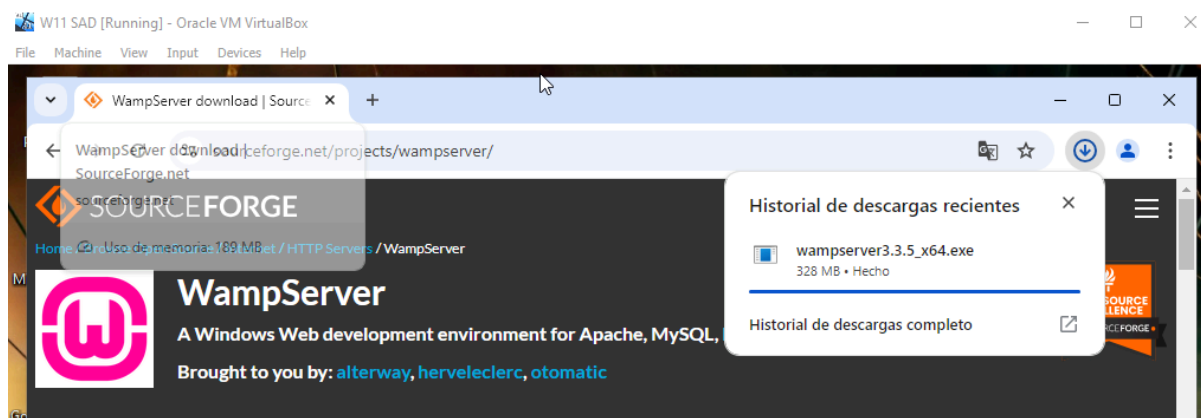
## 2. WAMP

### 2.1 Descarga e instalación

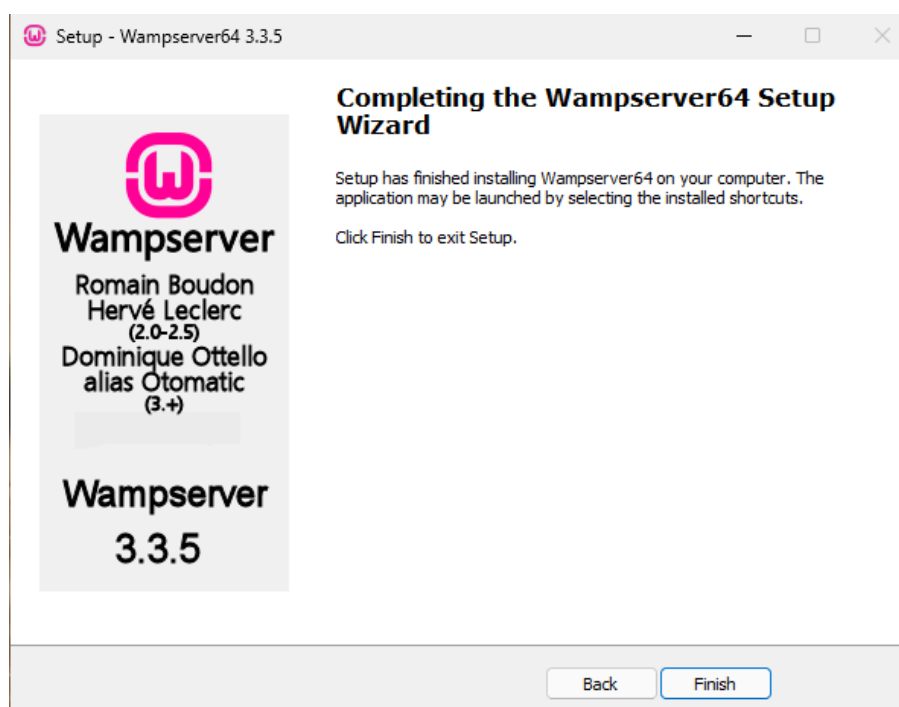
ANTES de proceder con la instalación de Wampserver, debe asegurarse de que ciertos elementos están instalados en su sistema, de lo contrario Wampserver no funcionará en absoluto, y además, la instalación será defectuosa y tiene que quitar Wampserver ANTES de instalar los elementos que faltaban. Asegúrese de estar "a la fecha" en los paquetes redistribuibles VC9, VC10, VC11, VC13 y VC14.



Visita el sitio oficial de WAMP en <https://www.wampserver.com/> y descarga la última versión compatible con tu arquitectura de Windows (64 bits o 32 bits).



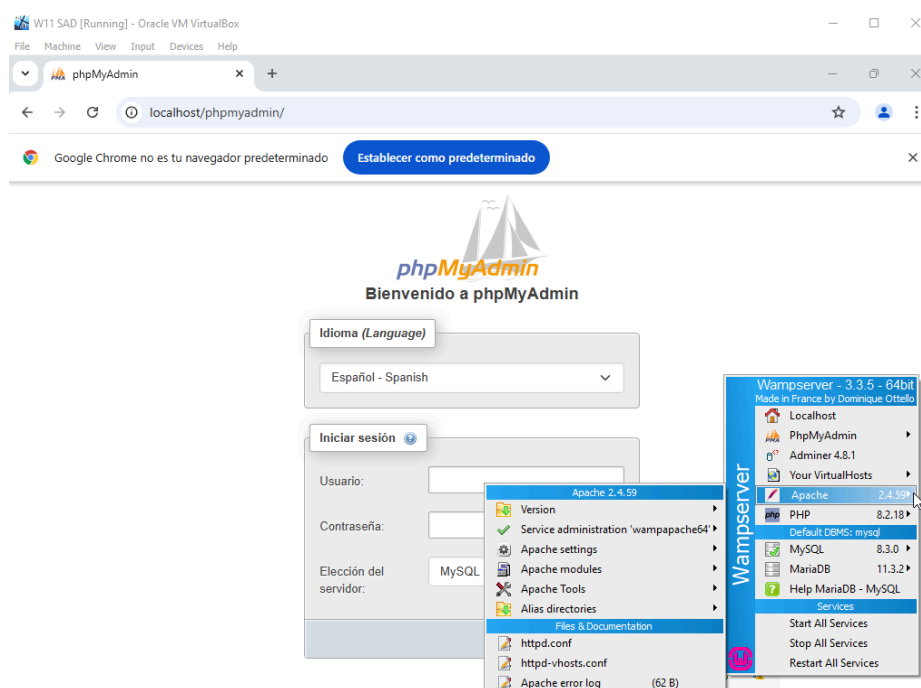
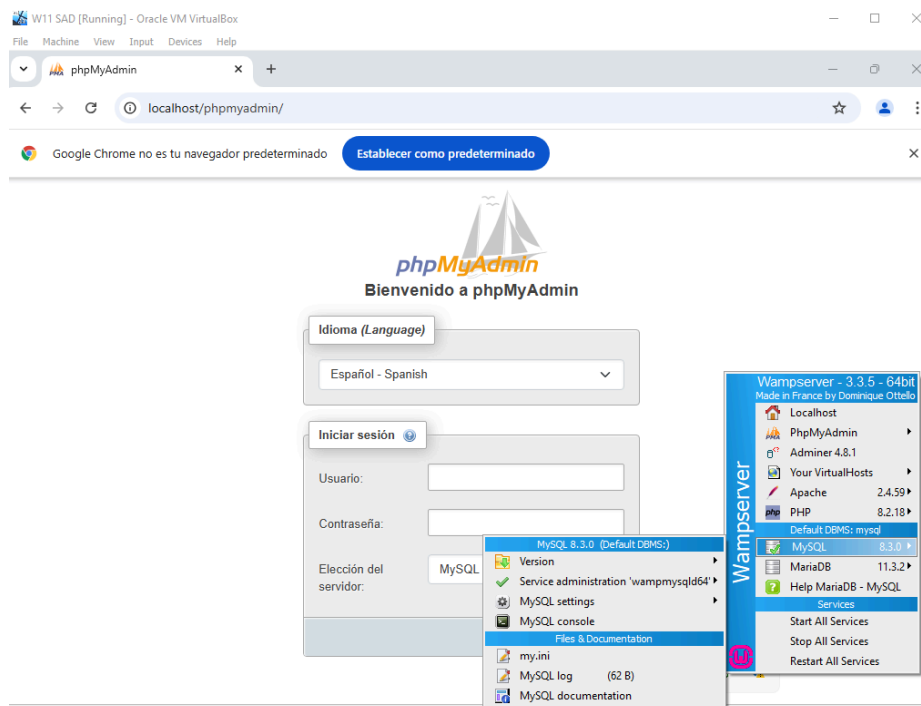
Ejecuta el instalador de WAMP como administrador. Sigue los pasos de la instalación. Acepta las rutas predeterminadas para los directorios. Una vez completada la instalación, ejecuta WAMP Server.



## 2.2 Configuración básica

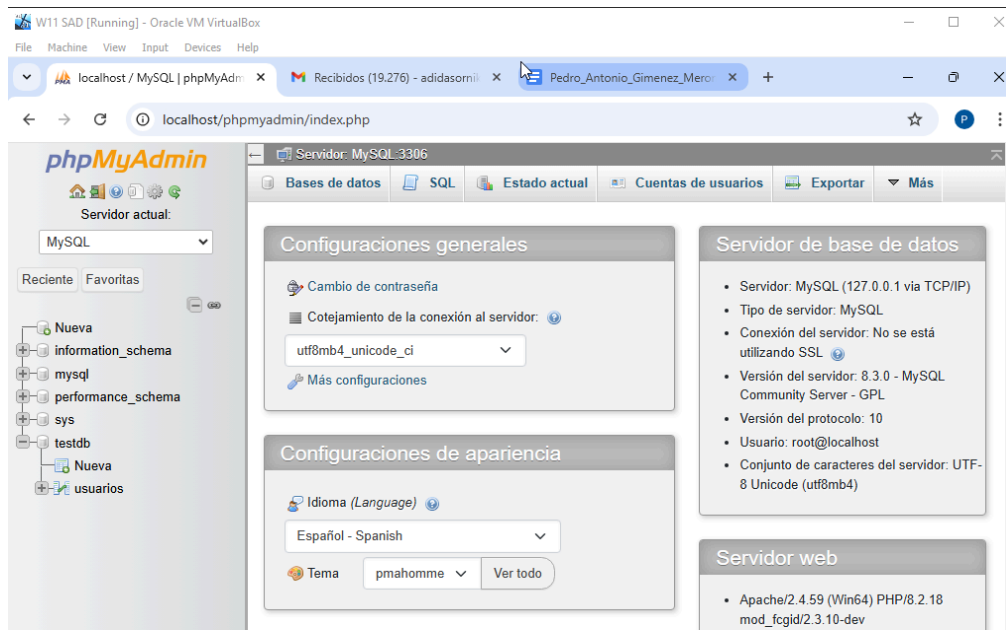
Asegúrate de que los servicios de Apache y MySQL están activos. Haz clic en el ícono de WAMP en la bandeja del sistema y verifica que ambos estén en verde. Si no están activos, puedes iniciarlos manualmente haciendo clic en el ícono y seleccionando "Iniciar todos los servicios".

## Instalación y Configuración LAMP y WAMP



Ahora accedemos a MySQL a través de la interfaz de phpMyAdmin desde el menú de WAMP (en el navegador dirígete a <http://localhost/phpmyadmin/>). Inicia sesión con el usuario **root** (sin contraseña por defecto). Y ya en el panel izquierdo puedes crear bases de datos y tablas:

## Instalación y Configuración LAMP y WAMP



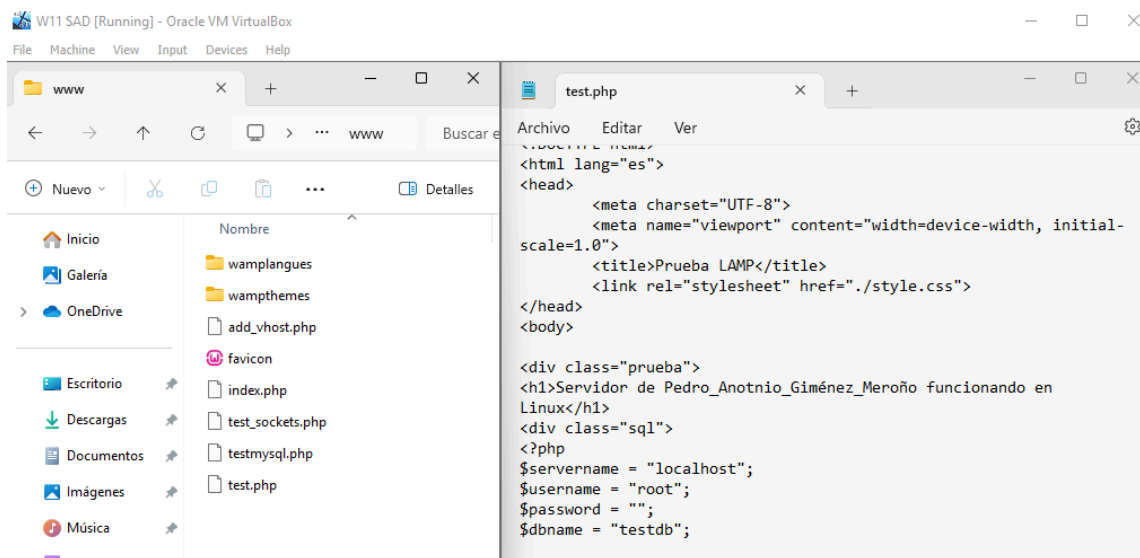
Una vez en MySQL yo he creado una base de datos llamada testdb. Después la he seleccionado y dentro he creado una tabla llamada usuarios. He Insertado algunos registros en la tabla usuarios para probar la base de datos y salimos de mysql:



```
INSERT INTO usuarios (nombre, correo) VALUES ('Juan Pérez',  
'juan@example.com'), ('María López', 'maria@example.com');
```



Crear un archivo PHP para conectarse a la base de datos. Navega a la carpeta donde se encuentran los archivos web en WAMP, que por defecto es C:\wamp64\www\. Crea un archivo llamado test.php con el siguiente contenido:



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Prueba LAMP</title>
  <link rel="stylesheet" href="./style.css">
</head>
<body>

<div class="prueba">
<h1>Servidor de Pedro_Anotnio_Giménez_Meroño funcionando en Windows</h1>
<div class="sql">
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "testdb";

// Crear conexión
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}

// Obtener datos de la tabla usuarios
$sql = "SELECT id, nombre, correo FROM usuarios";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Mostrar datos en una tabla
    echo "<table
border='1'><tr><th>ID</th><th>Nombre</th><th>Correo</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["id"]. "</td><td>" . $row["nombre"].
"</td><td>" . $row["correo"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "No se encontraron registros.";
}

$conn->close();
?>
</div>
</div>
</body>
</html>
```

Puedes añadir un archivo de css para dar estilo a tu página de prueba:

```
body {
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
```

```
    font-family: Arial, sans-serif;
    background-color: #1c1c1c;
}

.prueba {
    text-align: center;
}

.sql table {
    margin: 20px auto;
    border-spacing: 0;
    border-radius: 10px;
    overflow: hidden;
    width: 50%;
    background-color: #333;
    border: 1px solid #f4f4f4;
}

.sql th, .sql td {
    padding: 10px;
    border: 1px solid #f4f4f4;
    text-align: left;
}

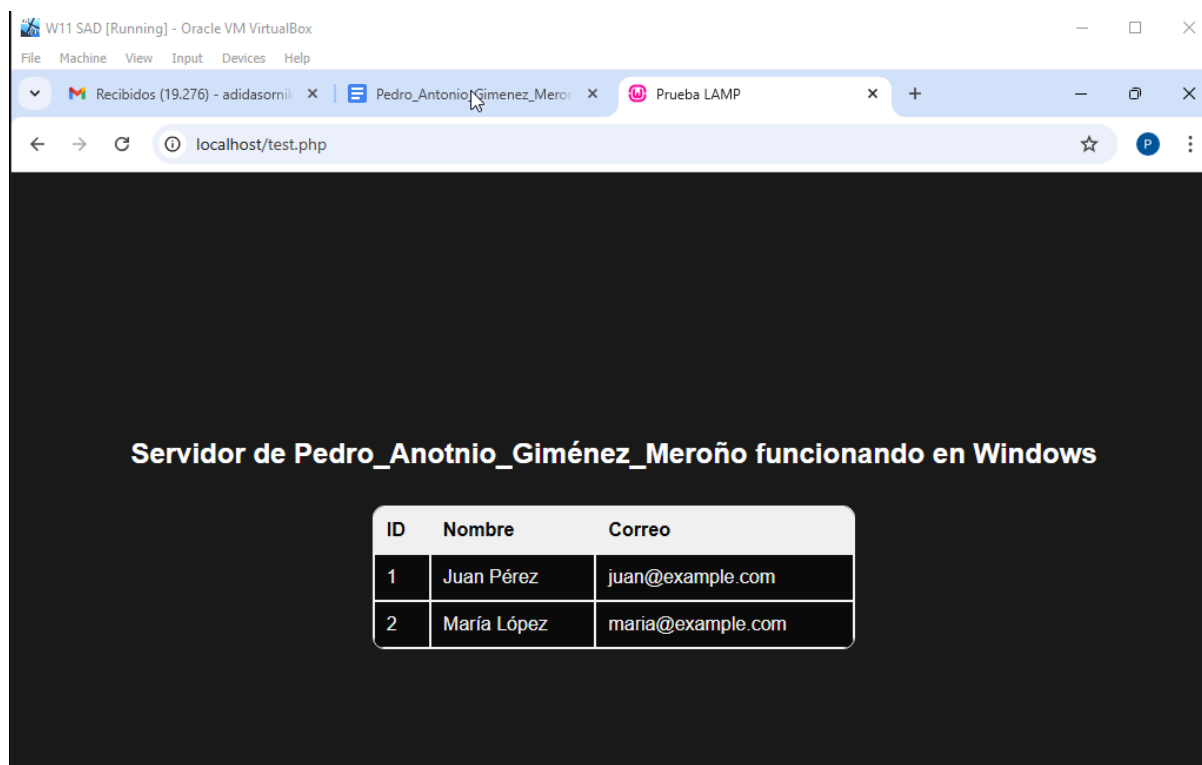
.sql th {
    background-color: #f4f4f4;
    color: #0d0d0d;
    border-bottom: 1px solid #f4f4f4;
}

.sql td {
    background-color: #0d0d0d;
    color: #f4f4f4;
    border-bottom: 1px solid #f4f4f4;
}

h1 {
    margin-bottom: 30px;
    font-size: 24px;
    color: #fff;
}
```

### 2.3 Prueba del entorno

Una vez ya configurado todo si lo hemos hecho bien deberíamos poder ver nuestra página de prueba a través del navegador en el link → <http://localhost:80/test.php> :



## 3. LAMP

### 3.1 Descarga e instalación

Abre un terminal y ejecuta:

```
sudo apt update && sudo apt upgrade  
sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql
```

```
pedro@server: /var/www/html
pedro@server:/var/www/html$ sudo apt update && sudo apt upgrade
Obj:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los siguientes paquetes se han retenido:
  ubuntu-drivers-common
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
pedro@server:/var/www/html$ sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
libapache2-mod-php ya está en su versión más reciente (2:8.1+92ubuntu1).
php ya está en su versión más reciente (2:8.1+92ubuntu1).
php-mysql ya está en su versión más reciente (2:8.1+92ubuntu1).
apache2 ya está en su versión más reciente (2.4.52-1ubuntu4.12).
mysql-server ya está en su versión más reciente (8.0.39-0ubuntu0.22.04.1).
```

### 3.2 Configuración básica para probar el entorno

Para la prueba de entorno vamos a intentar ver cómo funcionan todas las características de LAMP juntas haremos uso de una base de datos en una página .php en local a través del puerto 80 por defecto en Apache2.

Una vez instalados los paquetes vamos a comprobar que los servicios Apache2 y MySQL están activos con systemctl y si no lo están los activaremos (systemctl start <servicio>).

```
pedro@server: ~
pedro@server:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-10-24 16:15:05 UTC; 3h 49min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 1608 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 1613 (apache2)
    Tasks: 7 (limit: 3432)
   Memory: 13.9M
      CPU: 4.653s
   CGroup: /system.slice/apache2.service
           └─1613 /usr/sbin/apache2 -k start
             1614 /usr/sbin/apache2 -k start
             1615 /usr/sbin/apache2 -k start
             1616 /usr/sbin/apache2 -k start
             1617 /usr/sbin/apache2 -k start
             1618 /usr/sbin/apache2 -k start
             1622 /usr/sbin/apache2 -k start

oct 24 16:15:05 server systemd[1]: Starting The Apache HTTP Server...
oct 24 16:15:05 server apachectl[1612]: AH00558: apache2: Could not reliably determine the server's fully qualified d
oct 24 16:15:05 server systemd[1]: Started The Apache HTTP Server.
lines 1-21/21 (END)
pedro@server:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-10-24 16:17:52 UTC; 3h 46min ago
 Main PID: 783 (mysqld)
   Status: "Server is operational"
    Tasks: 38 (limit: 3432)
   Memory: 438.0M
      CPU: 20min 35.099s
   CGroup: /system.slice/mysql.service
           └─783 /usr/sbin/mysqld

oct 24 16:17:49 server systemd[1]: Starting MySQL Community Server...
oct 24 16:17:52 server systemd[1]: Started MySQL Community Server.
pedro@server:~$
```

Después continuaremos configurando MySQL. Para asegurar tu instalación, ejecuta el script de seguridad:

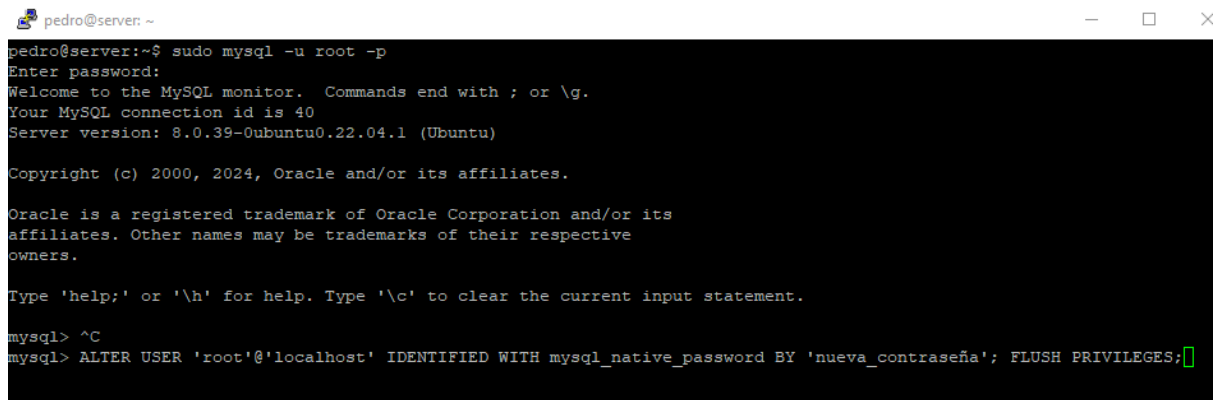
```
sudo mysql_secure_installation
```

Cuando termine el script vamos a verificar el tipo de autenticación de MySQL. En algunas distribuciones de Linux, MySQL utiliza auth\_socket para la autenticación del usuario root en lugar de una contraseña. Si este es el caso y deseas usar una contraseña para la conexión desde PHP, deberás cambiar el plugin de autenticación del usuario root a mysql\_native\_password. Ejecutamos mysql:

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

Cambia el método de autenticación del usuario root:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'nueva_contraseña';  
FLUSH PRIVILEGES;
```



```
pedro@server: ~  
pedro@server:~$ sudo mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 40  
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> ^C  
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'nueva_contraseña'; FLUSH PRIVILEGES;
```

Ahora vamos a crear una base de datos y una tabla para la prueba del entorno. Abre una terminal y accede al cliente de MySQL:

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

Una vez en MySQL yo he creado una base de datos llamada testdb. Después la he seleccionado y dentro he creado una tabla llamada usuarios:

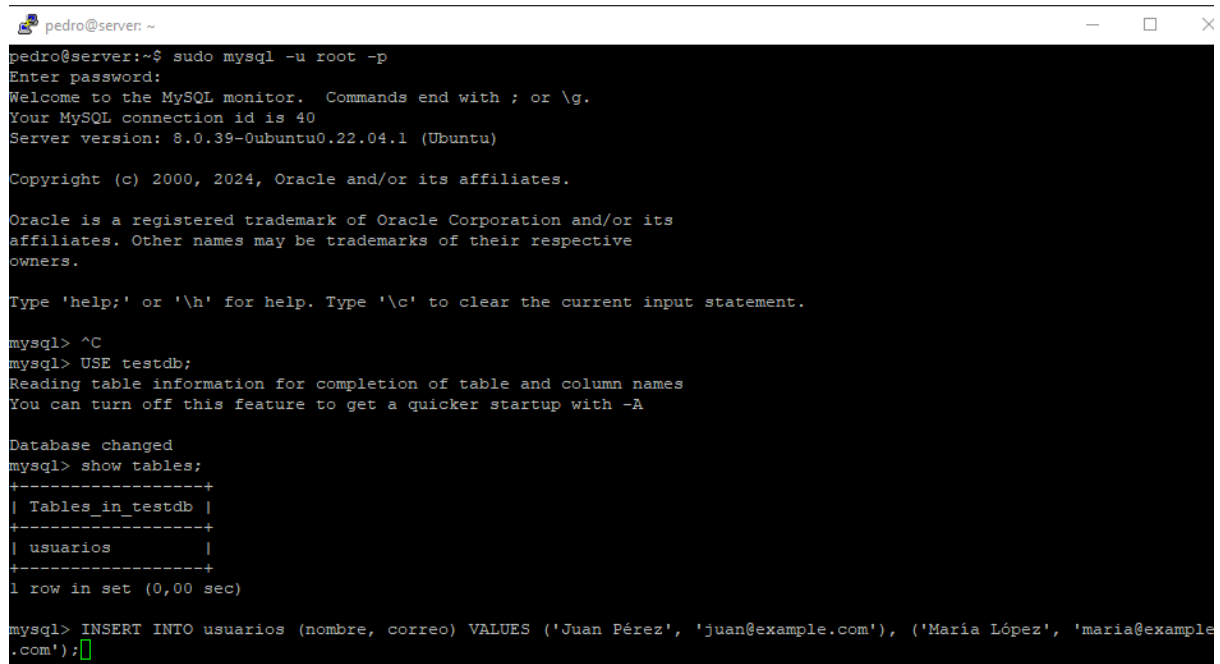
```
CREATE DATABASE testdb;
```

```
USE testdb;

CREATE TABLE usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50),
  correo VARCHAR(50)
);
```

He Insertado algunos registros en la tabla usuarios para probar la base de datos y salimos de mysql:

```
INSERT INTO usuarios (nombre, correo) VALUES ('Juan Pérez',
'juan@example.com'), ('María López', 'maria@example.com');
exit;
```



```
pedro@server: ~
pedro@server:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> ^C
mysql> USE testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables in testdb |
+-----+
| usuarios          |
+-----+
1 row in set (0,00 sec)

mysql> INSERT INTO usuarios (nombre, correo) VALUES ('Juan Pérez', 'juan@example.com'), ('María López', 'maria@example.com');

```

Ahora vamos al directorio donde subiremos los archivos web (/var/www/html) y creamos un .php nuevo con el siguiente código donde usamos HTML para dar estructura y nos conectamos gracias a PHP a la base de datos de MySQL para arrojar en la página web la tabla que hemos creado anteriormente y ver sus datos:

```
pedro@server: ~  
pedro@server:~$ sudo cat /var/www/html/test.php  
[sudo] password for pedro:  
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Prueba LAMP</title>  
  <link rel="stylesheet" href="./style.css">  
</head>  
<body>  
  
<div class="prueba">
```

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
  <title>Prueba LAMP</title>  
  <link rel="stylesheet" href="./style.css">  
</head>  
<body>  
  
<div class="prueba">  
<h1>Servidor de Pedro_Anotnio_Giménez_Meroño funcionando en Linux</h1>  
<div class="sql">  
<?php  
$servername = "localhost";  
$username = "root";  
$password = "11223344";  
$dbname = "testdb";  
  
// Crear conexión  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Verificar la conexión  
if ($conn->connect_error) {  
    die("Conexión fallida: " . $conn->connect_error);  
}  
  
// Obtener datos de la tabla usuarios  
$sql = "SELECT id, nombre, correo FROM usuarios";  
$result = $conn->query($sql);
```



```

if ($result->num_rows > 0) {
    // Mostrar datos en una tabla
    echo "<table
border='1'><tr><th>ID</th><th>Nombre</th><th>Correo</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["id"]. "</td><td>" . $row["nombre"].
"</td><td>" . $row["correo"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "No se encontraron registros.";
}

$conn->close();
?>
</div>
</div>
</body>
</html>

```

Puedes añadir un archivo de css para dar estilo a tu página de prueba:

```

body {
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    font-family: Arial, sans-serif;
    background-color: #1c1c1c;
}

.prueba {
    text-align: center;
}

.sql table {
    margin: 20px auto;
    border-spacing: 0;
}

```

```
        border-radius: 10px;
        overflow: hidden;
        width: 50%;
        background-color: #333;
        border: 1px solid #f4f4f4;
    }

    .sql th, .sql td {
        padding: 10px;
        border: 1px solid #f4f4f4;
        text-align: left;
    }

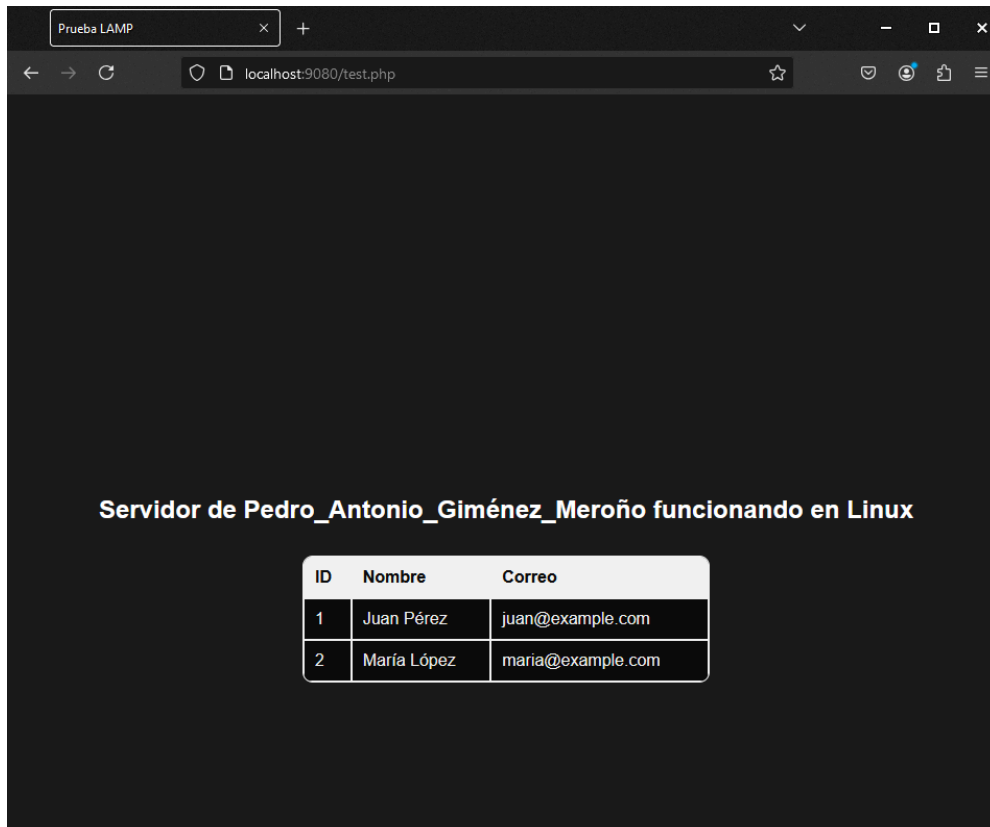
    .sql th {
        background-color: #f4f4f4;
        color: #0d0d0d;
        border-bottom: 1px solid #f4f4f4;
    }

    .sql td {
        background-color: #0d0d0d;
        color: #f4f4f4;
        border-bottom: 1px solid #f4f4f4;
    }

    h1 {
        margin-bottom: 30px;
        font-size: 24px;
        color: #fff;
    }
```

### 3.3 Prueba del entorno

Una vez ya configurado todo si lo hemos hecho bien deberíamos poder ver nuestra página de prueba a través del navegador en el link → <http://localhost:80/test.php> :



## 4. Comparación y Reflexión

### 4.1 Comparación

Característica	WAMP (Windows)	LAMP (Linux)
<b>Sistema Operativo</b>	Funciona en <b>Windows</b>	Funciona en <b>Linux</b>
<b>Facilidad de Uso</b>	Fácil de instalar y configurar con herramientas gráficas. Ideal para desarrolladores en Windows que desean un entorno local de desarrollo rápido.	Requiere más conocimiento técnico, pero ofrece mayor control y flexibilidad, ideal para servidores de producción.
<b>Estabilidad</b>	Adecuado para entornos de desarrollo local, no recomendado para producción.	Muy estable y ampliamente utilizado en servidores de producción.
<b>Rendimiento</b>	Rendimiento adecuado para pruebas locales.	Mejor rendimiento y optimización para servidores, especialmente en entornos de producción.
<b>Control y Personalización</b>	Menos control sobre configuraciones profundas del sistema operativo.	Ofrece más control y personalización sobre el sistema operativo y los servicios.

<b>Soporte para Producción</b>	Generalmente no se usa en producción, más bien para desarrollo local.	Se utiliza en servidores de producción, desde proyectos pequeños hasta grandes aplicaciones empresariales.
--------------------------------	---	--

## 4.2 Reflexión

### ¿Qué diferencias encontraste entre instalar el servidor en Windows y en Linux?

En Windows (WAMP), la instalación es más rápida y sencilla gracias a la interfaz gráfica, ideal para desarrollo local. En Linux (LAMP), la instalación es más manual a través de la terminal, lo que ofrece mayor control y personalización, siendo mejor para entornos de producción.

### ¿Cuál de los dos sistemas te resultó más cómodo para la gestión del servidor web? ¿Por qué?

WAMP en Windows es más cómodo para desarrollo debido a su facilidad y herramientas gráficas. Sin embargo, LAMP en Linux es preferido para producción por el control, seguridad y eficiencia que ofrece.

### ¿Qué ajustes realizarías para poner este servidor en producción de forma segura?

- Habilitar HTTPS con un certificado SSL/TLS.
- Restringir el acceso remoto a MySQL y usar contraseñas fuertes.
- Configurar un firewall para permitir solo los puertos necesarios.
- Deshabilitar funciones peligrosas en PHP (`exec`, `shell_exec`).
- Mantener actualizado el servidor y monitorizar los logs de seguridad.