

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

17-04-2025

SERVIDORES DE APLICACIONES

ACTIVIDAD 2

Several thin, curved lines in shades of blue and grey sweep upwards from the bottom left corner of the page.

Francisco J. Otero Herrero
GRUPO ATU

SERVIDORES DE APLICACIONES

SERVIDORES DE APLICACIONES

- **Contestar a las siguientes preguntas que se plantean de la forma más detallada posible:**

1. **Exponer las diferentes opciones más destacadas y que se te ocurran para instalar un servidor Apache, PHP y MySQL y comparar algunas de sus características.**

✓ Instalación Manual

- Implica descargar cada componente individualmente (**Apache, PHP, MySQL/MariaDB**) y configurarlos para que funcionen juntos.
- Descargar los archivos de instalación o binarios de cada componente desde sus respectivos sitios web oficiales.
- Instalar cada componente siguiendo las instrucciones específicas para el sistema operativo.
- Configurar **Apache** para que procese archivos **PHP** (esto generalmente implica modificar el archivo de configuración de **Apache, httpd.conf**).
- Configurar PHP para que se conecte a la base de datos **MySQL/MariaDB**.
- Ajustar la configuración de **MySQL/MariaDB** según sea necesario.

Ventajas:

- Mayor control sobre la configuración de cada componente.
- Posibilidad de instalar versiones específicas de cada componente.
- Mejor comprensión del funcionamiento interno de la pila.

Desventajas:

- Más complejo y requiere más conocimientos técnicos.
- Más tiempo y esfuerzo para la instalación y configuración.
- Mayor riesgo de errores de configuración.

SERVIDORES DE APLICACIONES

✓ **Paquetes Preconfigurados (XAMPP, WampServer, etc.)**

- Son paquetes que incluyen **Apache, PHP y MySQL/MariaDB** ya configurados para funcionar juntos. Facilitan la instalación de la pila.
- **XAMPP**: Multiplataforma (Windows, Linux, macOS).
- **WampServer**: Específico para Windows.
- **MAMP**: Específico para macOS.
- Descargar el paquete desde el sitio web oficial.
- Ejecutar el instalador y seguir las instrucciones.
- Los componentes (**Apache, PHP, MySQL/MariaDB**) se instalan y configuran automáticamente.

Ventajas:

- Instalación rápida y sencilla.
- Fácil de usar para principiantes.
- Incluye herramientas útiles como **phpMyAdmin** para administrar **MySQL/MariaDB**.

Desventajas:

- Menos control sobre la configuración.
- Puede incluir versiones de los componentes que no son las últimas.
- No se recomienda para entornos de producción en muchos casos.

✓ **Gestores de Paquetes (Linux)**

- **En sistemas Linux**, los gestores de paquetes (como **apt en Ubuntu/Debian, yum en CentOS/RHEL, dnf en Fedora**) facilitan la instalación de software.
- Utilizar la línea de comandos para instalar **Apache, PHP y MySQL/MariaDB** desde los repositorios del sistema operativo.
- Configurar los componentes según sea necesario.

Ventajas:

- Instalación sencilla y rápida.
- Mantenimiento fácil (actualizaciones, etc.).
- Alta compatibilidad y estabilidad.

Desventajas:

- Menos control que la instalación manual.
- Las versiones de los componentes pueden depender de los repositorios del sistema operativo.

SERVIDORES DE APLICACIONES

✓ **Contenedores (Docker)**

- Utilizar **contenedores Docker** para empaquetar y ejecutar **Apache, PHP y MySQL/MariaDB** en entornos aislados.
- Instalar **Docker**.
- Obtener o crear imágenes de **Docker para Apache, PHP y MySQL/MariaDB**.
- Configurar los contenedores para que se comuniquen entre sí.
- Ejecutar los contenedores.

Ventajas:

- Portabilidad (funciona igual en diferentes entornos).
- Aislamiento (evita conflictos entre aplicaciones).
- Escalabilidad.

Desventajas:

- Requiere conocimientos de Docker.
- Puede ser más complejo para configuraciones iniciales simples.

2. Describe las funciones de un servidor de aplicaciones. ¿Cuáles son los parámetros de configuración de Apache2?

Un servidor de aplicaciones es un **framework de software** que proporciona servicios de aplicación a las aplicaciones. Actúa como un intermediario entre los clientes (como navegadores web o aplicaciones móviles) y los servidores de bases de datos. **Los servidores de aplicaciones** simplifican el desarrollo de aplicaciones empresariales al manejar tareas comunes, permitiendo a los desarrolladores centrarse en la lógica de negocio específica de la aplicación.

✓ **Funciones Clave de un Servidor de Aplicaciones:**

- **Ejecución de Lógica de Negocio:** Los servidores de aplicaciones proporcionan un entorno para ejecutar la lógica de negocio de la aplicación, a menudo escrita en lenguajes como **Java, Python, o .NET**.
- **Conectividad con Bases de Datos:** Facilitan la conexión y el acceso a bases de datos, gestionando pools de conexiones, transacciones y otros aspectos relacionados con la persistencia de datos.
- **Gestión de Transacciones:** Aseguran la integridad de las transacciones, coordinando múltiples operaciones para que se completen con éxito o se reviertan en caso de fallo.

SERVIDORES DE APLICACIONES

- **Seguridad:** Implementan mecanismos de seguridad como autenticación, autorización y encriptación para proteger la aplicación y los datos.
- **Escalabilidad:** Permiten escalar la aplicación para manejar un mayor número de usuarios y solicitudes, mediante técnicas como **el balanceo de carga y el clustering**.
- **Servicios Web:** Muchos servidores de aplicaciones soportan la creación y el consumo de servicios web, facilitando la integración con otras aplicaciones.
- **Gestión de Recursos:** Optimizan el uso de recursos del sistema, como memoria y CPU, para mejorar el rendimiento de la aplicación.

✓ **Parámetros de Configuración de Apache2**

Apache2 se configura principalmente a través de archivos de texto. El archivo de configuración principal es **httpd.conf**, y puede incluir otros archivos de configuración. Estos archivos contienen directivas que controlan el comportamiento del servidor.

Algunos de los parámetros de configuración más comunes e importantes de **Apache2** incluyen:

- Listen:** Especifica el puerto (y opcionalmente la dirección IP) en el que Apache2 escuchará las conexiones entrantes.
- ServerRoot:** Define el directorio donde reside el software del servidor Apache2.
- DocumentRoot:** Establece el directorio raíz desde donde Apache2 servirá los archivos web.
- <Directory>:** Encierra un grupo de directivas que se aplicarán solo al directorio especificado y sus subdirectorios.
- Options:** Controla qué características están disponibles en un directorio, como la ejecución de **scripts CGI** o la visualización de índices de directorio.
- AllowOverride:** Especifica qué directivas pueden ser anuladas por los archivos **.htaccess** en los directorios.
- Require / Allow / Deny:** Controlan el acceso a los directorios según la dirección IP, el nombre de host u otros criterios.
- LoadModule:** Carga módulos en el servidor Apache2, extendiendo su funcionalidad (por ejemplo, para manejar **PHP o SSL**).
- VirtualHost:** Define hosts virtuales, permitiendo que un solo servidor Apache2 aloje múltiples sitios web.
- ServerName:** Establece el nombre de dominio del servidor.
- ErrorLog / CustomLog:** Especifican dónde Apache2 debe registrar los errores y los accesos, respectivamente.
- KeepAlive:** Controla si se permiten las conexiones persistentes (**múltiples solicitudes a través de la misma conexión TCP**).
- Timeout:** Establece el tiempo que Apache2 esperará ciertas acciones, como la recepción de una solicitud.