

Parte 2: Despliegue y flujo de aplicaciones web con LAMP

Pregunta 1: Entorno del servidor

Imagina que acabas de desplegar tu aplicación web en un servidor Linux.

a) ¿Qué componentes del stack LAMP instalaste para que la aplicación funcione correctamente?

Linux – Sistema operativo base.

Apache – Servidor web.

MySQL – Sistema de gestión de bases de datos.

PHP – Lenguaje de programación del lado del servidor.

b) Explica brevemente la función de cada componente dentro del funcionamiento de tu aplicación.

Linux:

Es el sistema operativo en el que corre todo el servidor. Se encarga de manejar los recursos del sistema, las conexiones de red y mantener la seguridad. Básicamente, es la base sobre la que funcionan los demás componentes.

Apache:

Es el servidor web que recibe las peticiones HTTP de los usuarios cuando entran a la página. Apache se encarga de responder enviando los archivos solicitados, como páginas HTML. Si el archivo es PHP, lo pasa al intérprete de PHP para que lo ejecute. Es como el “puente” entre el navegador y el código del servidor.

MySQL:

Es el sistema de base de datos donde se guarda toda la información de la aplicación, como usuarios, productos o pedidos. PHP se conecta a MySQL para consultar, agregar o modificar esos datos según lo que el usuario haga en la web.

PHP:

Es el lenguaje de programación que se ejecuta del lado del servidor. Se encarga de

procesar la lógica de la aplicación, comunicarse con la base de datos MySQL y generar páginas HTML dinámicas que luego Apache envía al navegador.

Pregunta 2: Flujo de una petición

Cuando un usuario escribe la URL de tu aplicación en el navegador, describe qué ocurre desde que se hace la petición HTTP hasta que el usuario ve la página web en su navegador.

Cuando un usuario escribe la dirección de mi aplicación en el navegador (por ejemplo, www.bryanapp.com), ocurre lo siguiente paso a paso:

1. El navegador hace la petición HTTP

Cuando el usuario presiona *Enter* o realiza alguna acción (como hacer clic en un enlace), el navegador envía una solicitud al servidor para pedir una página, por ejemplo *index.php*.

Esa solicitud incluye información como la URL, el tipo de navegador, las cookies y algunos parámetros si son necesarios (por ejemplo, *?validarEdad=edad*).

2. El servidor Linux recibe la solicitud

La petición llega primero al servidor, que usa Linux como sistema operativo.

Linux se encarga de manejar la red, repartir los recursos del sistema y mantener todo funcionando de forma segura y estable para que la aplicación responda bien.

3. Apache recibe y analiza la petición

Después, el servidor web Apache toma la solicitud.

Si el archivo que se pidió es estático (como una imagen, un archivo CSS o un HTML), Apache lo entrega directamente al navegador.

Pero si el archivo tiene código PHP, Apache lo envía al intérprete de PHP para que lo ejecute.

4. PHP ejecuta el código del servidor

El intérprete de PHP lee y ejecuta el código del archivo (por ejemplo, *index.php*).

En este paso, PHP puede hacer varias cosas, como:

- Revisar si el usuario ha iniciado sesión.

- Procesar formularios.
 - Conectarse a la base de datos MySQL para obtener o guardar información.
-

5. MySQL maneja las consultas de datos

Si el archivo PHP necesita información, envía una consulta SQL a MySQL.
Por ejemplo:

```
SELECT * FROM empleados WHERE edad >= 18;
```

MySQL busca esos datos en las tablas y devuelve los resultados a PHP.

6. PHP genera la página HTML

Con los datos que recibió de MySQL, PHP construye una página completa en HTML, que puede incluir estilos (CSS), scripts (*JavaScript*) o que entienda y pueda mostrar.

7. Apache envía la respuesta al navegador

Cuando PHP termina de generar la página, Apache la envía de vuelta al navegador del usuario como una respuesta HTTP.

8. El navegador muestra la página al usuario

Finalmente, el navegador recibe el HTML y muestra la página en pantalla con todo su contenido: textos, imágenes, botones y demás elementos visuales.

Si la página necesita más archivos (como imágenes o estilos CSS), el navegador hace nuevas peticiones para cargarlos.

En resumen, así entiendo yo el flujo:

Usuario → HTTP Request → Apache → PHP → MySQL → PHP → Apache → HTTP Response → Usuario

