UD2 Introducción a las arquitecturas web y servidores

4. ¿Qué son frontend y backend?

En el desarrollo web moderno, solemos dividir una aplicación en dos grandes partes: el **frontend** y el **backend**. Esta separación permite especializar tareas, distribuir responsabilidades entre diferentes tecnologías y facilitar el mantenimiento y la escalabilidad del software.

4.1. Frontend (lado cliente)

El **frontend** es la parte de la aplicación web que interactúa directamente con el usuario. Todo lo que se muestra en el navegador, lo que el usuario puede ver, tocar o con lo que puede interactuar, forma parte del frontend.

Incluye:

- **Estructura y contenido** (HTML)
- **Estilos y diseño** (CSS)
- **Interactividad y lógica de presentación** (JavaScript)

En aplicaciones modernas, el frontend no solo muestra contenido, sino que también puede realizar peticiones a servidores, validar formularios, gestionar rutas de navegación, y más. Hoy en día se suelen usar frameworks de JavaScript como:

- **React**
- **Vue.js**
- **Angular**
- **Svelte**

Ejemplo: Cuando entras a una red social y ves publicaciones, botones, menús y animaciones, todo eso ha sido renderizado por el frontend.

4.2. Backend (lado servidor)

El **backend** es la parte de la aplicación que funciona en el servidor. Gestiona la lógica de negocio, el acceso a bases de datos, la autenticación, el control de acceso, el envío de correos, la gestión de archivos, y en general, todo lo que no es visible para el usuario pero es imprescindible para el funcionamiento.

Incluye:

- **Aplicaciones del lado servidor** (Node.js, PHP, Java, Python, etc.)
- **Base de datos** (MySQL, PostgreSQL, MongoDB…)
- **Lógica de negocio y seguridad**
- **APIs** para que el frontend se comunique con el backend

Ejemplo: Cuando un usuario inicia sesión, el backend valida sus credenciales, genera una sesión y devuelve los datos del usuario al frontend.

4.3. Comunicación entre frontend y backend

El frontend y el backend se comunican normalmente a través de **peticiones HTTP**. El frontend puede enviar datos (por ejemplo, de un formulario) al backend mediante una petición `POST`, y este responde con los resultados adecuados (por ejemplo, si el login es correcto o no).

Hoy en día, esta comunicación suele hacerse mediante **APIs REST** o **GraphQL**.

4.4. ¿Por qué es importante esta separación?

Separar frontend y backend permite:

- Reutilizar el backend con diferentes frontends (web, móvil, escritorio).
- Desarrollar de forma paralela por equipos diferentes.
- Mejorar la seguridad, ocultando la lógica sensible al usuario final.
- Escalar cada parte de forma independiente.

4.5. Full Stack: ¿qué significa?

Un desarrollador **full stack** es aquel que trabaja tanto en el frontend como en el backend, entendiendo cómo se conectan y qué responsabilidades tiene cada capa. Este perfil es muy valorado en entornos de despliegue y mantenimiento, como los que se abordan en este módulo.