

# Introducción al desarrollo front-end

 2026-01 —  15 min.

**Que es "front-end"?**

**Front-end es aquello que el  
usuario ve y manipula.**

# Front-end

Se considera "front-end" aquello que manipula el usuario, o el lado de la aplicación que está bajo el control del usuario.

# Back-end

Se considera "back-end" aquello que **NO** manipula el usuario, o el lado de la aplicación que **NO** está bajo el control del usuario.

El front-end interactúa con el back-end para obtener y enviar datos.

# Front-end

Para las aplicaciones web,

El front-end es la página web. El back-end es el servidor que maneja los datos.

# Front-end

Para cualquier aplicación,

El front-end es la interfaz de usuario (y el programa que lo contiene). El back-end es el servidor y la lógica de negocio.

# Front-end y Back-end

El front-end también es llamado "el cliente", o "client-side".

El back-end también es llamado "el servidor", o "server-side".

El front-end se ejecuta en el navegador del usuario.

# Front-end

En términos del modelo MVC:

Vista: Frontend

Modelo: Backend

Controlador: Puede estar en ambos lados

## REST API Model



# Que es una aplicación web?

Una aplicación a la que se accede usando un navegador.

Adicionalmente, puede ser una aplicación desarrollada usando tecnologías típicamente usadas en desarrollo web.

# Front-end moderno

HTML para la estructura.

CSS para el estilo.

JavaScript para la interactividad.

# Librerías y Frameworks

Para facilitar el desarrollo, existen librerías y frameworks que proporcionan herramientas y estructuras predefinidas.

Ambos ayudan a escribir menos código y desarrollar más rápido.

# Que es una Libreria?

Una **librería** front-end tiene las herramientas para construir una pagina web sencilla con minima duplicación de código.

No incluye soluciones para routing, estado global, o build tools.

Tú eliges las herramientas adicionales que necesites.

Ejemplos: React, Vue, Preact, Lit, Svelte

# Que es un Framework?

Un **framework** es como una librería pero trae todo.

Incluye routing, manejo de estado, build tools, y más.

Provee convenciones y estructura predefinida.

Ejemplos: Angular, Next.js, Nuxt, SvelteKit

# Diferencias Clave

## Librería:

- Solo UI, solo routing, solo estado, etc.
- Tú eliges el resto de herramientas
- Mayor flexibilidad

## Framework:

- Incluye todo lo necesario
- Estructura y herramientas predefinidas
- Más opinado sobre arquitectura

# React: Librería o Framework?

**React es una librería**, no un framework.

Solo maneja la UI. No incluye routing, manejo de estado global, etc.

Debes agregar herramientas como React Router, Redux/Zustand, etc.

**Next.js** es un framework popular basado en React que sí incluye todo.

# Frameworks Front-end Populares

**Angular:** Framework completo de Google

**Next.js:** Framework basado en React

**Nuxt:** Framework basado en Vue

**SvelteKit:** Framework basado en Svelte

Cada uno tiene sus ventajas y casos de uso específicos.

En esta clase veremos React y Next.js.