

Librerías, Frameworks, SPA y Cursor

Autor: L. Falcon

Librería vs Framework

Librería

Conjunto de funciones/utilidades que usás cuando querés.

Control: Vos llamás a la librería cuando necesitás.

Instalación:

```
npm install nombre-libreria
```

```
Ejemplo de uso: axios.get(...)
```

No impone estructura, mejor para proyectos pequeños.

Framework

Estructura completa que dicta cómo se organiza y ejecuta tu app.

Control: El framework controla el flujo de ejecución.

Instalación:

```
npm install nombre-framework
```

Ejemplo de uso: Crear componentes, rutas, estados en React o Angular

Define carpetas, componentes, servicios y flujo.

Pasos para iniciar un proyecto con NPM

1- Crear un directorio

```
mkdir mi-proyecto cd mi-proyecto
```

2- Inicializar NPM

```
npm init -y
```

3- Estructura básica de package.json

```
{"name": "mi-proyecto", "version": "1.0.0", ...}
```

4- Instalar dependencias

```
npm install express
```

5- Usar scripts en package.json

```
"scripts": {"start": "node index.js"} npm start
```

Librerías y Frameworks: React, Angular y Vue

React

Librería de JavaScript (Facebook). Ideal para SPA con interfaces interactivas. Ventajas: modularidad, ecosistema grande, Virtual DOM. Desventajas: solo maneja la vista, requiere librerías extra.

Angular

Framework completo (Google). Ideal para proyectos grandes con TypeScript. Ventajas: herramientas integradas, modularidad. Desventajas: curva de aprendizaje alta.

Vue

Framework progresivo (Evan You). Ideal para apps pequeñas que escalan. Ventajas: curva suave, equilibrio entre simplicidad y estructura. Desventajas: menor adopción empresarial.

Paradigmas de Programación (diseño)

Modelo-Vista-Controlador (MVC)

Modelo: gestiona los datos. Vista: interfaz de usuario. Controlador: intermediario que procesa entradas y actualiza el modelo.

Modelo-Vista-ViewModel (MVVM)

Modelo: datos y lógica de negocio. Vista: interfaz mostrada. ViewModel: lógica de presentación que vincula Modelo y Vista automáticamente.

SPA (Single Page Application)

Una SPA carga un único HTML y actualiza contenido dinámicamente con JS.

Ejemplos: Gmail, Google Drive, Trello.

1- Tener un HTML base

2- Crear un solo archivo JS

```
app.js controlará todo.
```

3- Dividir contenido en funciones

```
function cargarHeader() {...}
```

4- Cargar dinámicamente

```
async function cargarSeccion(url,id){...}
```

5- Simular navegación interna

```
btn.addEventListener('click',()=>cargarSeccion('home.html','main'))
```

Cursor

Cursor es un fork de Visual Studio Code con integración de IA.

1- Descargar Cursor

cursor.so

2- Conexión con VSC

Puedes importar configuraciones, extensiones y temas.

3- Configuración inicial

Conectar con GitHub/repos y crear cuenta en Cursor.