

# Programación IV

## Profesora Cynthia Estrada

Fecha: 27/08

# "Frameworks"

### Integrantes:

- "Díaz Rossini Juan José",
- "Gallo Genaro"
- "Navarro Victor Leandro"

---

### 1. ¿Qué son los frameworks?

Un framework es una especie de "esqueleto" o "andamiaje" que provee una estructura predefinida para el desarrollo de software. Incluye librerías, herramientas y convenciones que ayudan a programar de forma más organizada, eficiente y segura.

#### En pocas palabras:

**LIBRERÍA** → te da piezas listas para usar.

**FRAMEWORK** → te da el "esqueleto" y reglas de cómo armar todo el proyecto.

### Ejemplo:

**LIBRERÍA:** jQuery (puedes usar una función para hacer un efecto).

**FRAMEWORK:** Angular (te dice cómo estructurar todo el proyecto, dónde va cada cosa).

---

### 2. ¿Qué puede mejorar con el uso de los frameworks?

- **Productividad:** evita repetir código, muchas funcionalidades ya están hechas.
- **Estandarización:** el código sigue un mismo estilo, facilitando el trabajo en equipo.
- **Mantenibilidad:** si otro programador toma el proyecto, entiende la estructura más fácil.
- **Seguridad:** muchos frameworks tienen protección frente a ataques comunes (XSS, inyecciones SQL, etc.).
- **Escalabilidad:** los proyectos pueden crecer sin volverse caóticos.
- **Rendimiento:** algunos frameworks optimizan la ejecución y el uso de recursos.

---

### 3. Tipos de frameworks

#### - Web (Frontend y Backend):

- **Frontend:** React, Angular, Vue.js.
- **Backend:** Django (Python), Laravel (PHP), Spring (Java), Express.js (Node.js).

#### - Móviles:

- React Native
- Flutter
- Ionic
- Xamarin.

#### - Escritorio:

- Electron
- .NET
- Qt.

#### - Testing:

- JUnit
- Jest
- Mocha.

#### - Machine Learning/IA:

- Tensor Flow
- PyTorch.

---

### 4. Ventajas de los frameworks

1. Reducen el tiempo de desarrollo.
2. Fomentan buenas prácticas de programación.
3. Reutilización de código y componentes.
4. Gran documentación y comunidad activa.
5. Seguridad reforzada frente a vulnerabilidades conocidas.
6. Escalabilidad: proyectos grandes y pequeños pueden crecer ordenadamente.

---

### 5. Recomendaciones para elegir un framework

- Analizar el tipo de proyecto (web, móvil, escritorio).
- Evaluar la curva de aprendizaje (¿es fácil de aprender para el equipo?).
- Comprobar si tiene documentación clara y comunidad activa.
- Revisar rendimiento y soporte a largo plazo.
- Analizar la compatibilidad con otras herramientas.
- Ver la popularidad en la industria (asegura más oportunidades laborales).

---

### 6. ¿Es lo mismo un framework que una librería moderna?

No son lo mismo...

- **Librería:** es un conjunto de funciones o componentes que usas cuando lo necesitas. El programador decide cómo y cuándo integrarla.
- **Framework:** marca la estructura de trabajo. Es más rígido, dicta cómo organizar tu código y qué pasos seguir.

**Ejemplo:**

- **Librería:** React (aunque se suele llamar framework, técnicamente es una librería).
- **Framework:** Angular o Vue (definen más reglas y estructura).

---

### 7. ¿Qué es React?

React es una librería de JavaScript creada por Facebook (hoy Meta) en 2013 para construir interfaces de usuario interactivas y dinámicas. Aunque técnicamente es una librería, muchos la consideran un framework por su ecosistema y la forma en que estructura aplicaciones.

---

## 8. ¿Para qué sirve React?

### Sirve para desarrollar:

- Aplicaciones web interactivas.
  - Aplicaciones móviles (con React Native).
  - Interfaces de usuario rápidas, dinámicas y fáciles de mantener.
- 

## 9. Características de React

- **Componentes reutilizables:** la UI se divide en piezas independientes.
  - **Virtual DOM:** React no actualiza todo el HTML, solo los cambios necesarios.
  - **JSX:** permite escribir HTML dentro de JavaScript.
  - **Unidireccionalidad de datos:** los datos fluyen de arriba hacia abajo (padre → hijo).
  - **Gran ecosistema:** compatible con Redux, Next.js, Material UI, Tailwind CSS.
  - **React Native:** permite llevar el mismo concepto a móviles.
- 

## 10. Funcionamiento de React

1. Se crean componentes que representan partes de la interfaz (ejemplo: botón, navbar, tarjeta de usuario).
  2. Los componentes generan un árbol virtual (Virtual DOM).
  3. Cuando cambia un dato, React compara el Virtual DOM con el DOM real y actualiza solo lo que cambió (no recarga toda la página).
  4. Esto hace que las aplicaciones sean rápidas y eficientes
- 

## 11. Ventajas y desventajas de React

### Ventajas:

- Alto rendimiento gracias al Virtual DOM.
- Gran comunidad y ecosistema.
- Fácil de reutilizar componentes.
- Compatible con muchas librerías externas.
- Gran demanda laboral

### Desventajas:

- JSX puede ser confuso al inicio.
- El ecosistema cambia muy rápido (muchas actualizaciones).
- Para proyectos grandes se necesita aprender librerías extras (ej: Redux para manejo de estado).
- No es un framework completo, depende de otras herramientas.

---

## 12. ¿React domina el desarrollo frontend?

**Sí, React domina el desarrollo frontend actualmente por varias razones:**

- Popularidad y respaldo de Meta.
- Comunidad muy grande.
- Gran ecosistema de librerías y frameworks relacionados (Next.js, React Native).
- Fácil integración con backend y otras tecnologías.
- Usado por empresas líderes (Netflix, Instagram, Airbnb, Spotify).

---

## 13. Herramientas necesarias para trabajar con React

- **Node.js + npm o yarn:** para gestionar paquetes.
  - **Editor de código:** Visual Studio Code (el más usado).
  - **Herramientas de inicialización:** Create React App, Vite o Next.js.
  - **Extensiones:** React DevTools (para depurar).  
**Opcionales según proyecto:**
    - **React Router** (ruteo de páginas).
    - **Redux/Zustand** (manejo de estado global).
    - **Tailwind CSS / Material UI** (estilos).
-