



Desarrollo de Aplicaciones Informáticas

Tema: Introducción a **React** y **Next.js**

Docente: Ing. Pablo Morandi

Ayudante: Matías Marchesi

Curso: 5to año

Especialidad: Informática

Redactado por: Matías Marchesi

Versión: v1.1

- [1. ¿Qué es React?](#)
 - [2. Arquitectura MVC y React](#)
 - [3. DOM Virtual](#)
 - [4. Next.js - El Framework](#)
-

1. ¿Qué es React?

React es una biblioteca de JavaScript que nos permite la creación de interfaces más copadas mediante un elemento muy útil como lo son los **componentes**.

Historia y desarrollo:

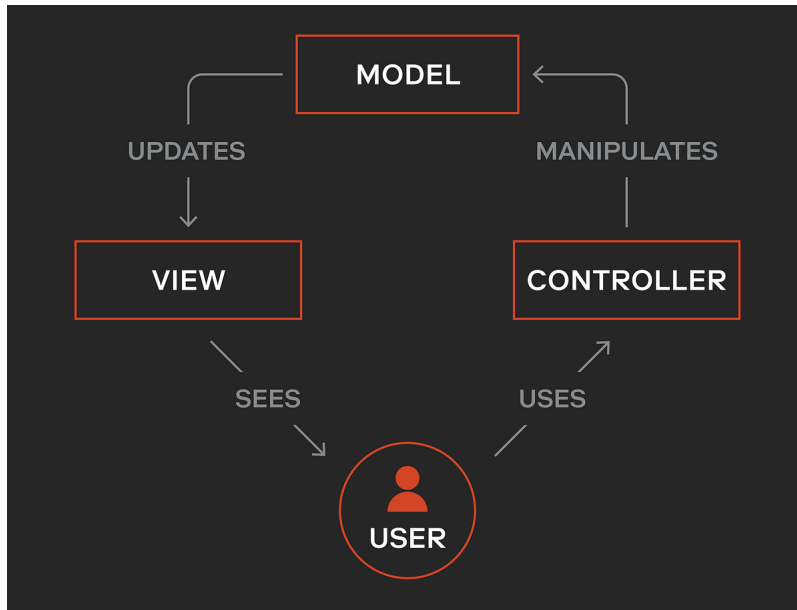
- Se desarrolló en **2013 por Facebook**
- Es mantenido por la comunidad ya que es de **código abierto**
- Actualmente es una de las tecnologías más populares para desarrollo web

¿Por qué React? React revoluciona la forma en que construimos interfaces de usuario al permitirnos crear aplicaciones más dinámicas, reutilizables y fáciles de mantener.

2. Arquitectura MVC y React

Para el desarrollo de aplicaciones seguimos el modelo de arquitectura **MVC** y React justamente se encarga de la **V (View)**:

- **Model:** Representa la información que puede y no puede ver el usuario
- **Controller:** Gestiona las peticiones de ver y no ver información
- **View:** Visualiza la información (aquí entra React)

Flujo de la aplicación:

3. DOM Virtual

Lo que tenemos con React es la **simplificación del DOM** como lo conocemos hasta ahora.

¿Esto significa que todo lo que hicimos fue para nada? ¡No!

Diferencias clave:

- **DOM tradicional:** Se mantiene directamente en el navegador (Chrome, Edge, etc.)
- **DOM Virtual:** React lo tiene virtualizado en su propio entorno

Ventajas del DOM Virtual:

- **Mayor performance:** Solo actualiza lo que realmente cambió
- **Control de estado:** Representa el momento exacto de la aplicación
- **Gestión de propiedades:** Similar al DOM común pero más eficiente

4. Next.js - El Framework

Next.js es el framework que nosotros vamos a utilizar, hecho por **Vercel**, que se encarga de:

- **Renderización** (o muestra) de páginas
- **Routing** (cómo acceder a las mismas)
- **Organización** de la aplicación

"React nos permite hacer componentes. Pero Next.js nos organiza la aplicación: cómo accedemos a las páginas, cómo cargamos datos, cómo se muestra lo que hicimos. En vez de preocuparnos por cómo conectar todo, Next ya lo hace por nosotros."

¿Por qué Next.js?

- Configuración automática
 - Routing basado en archivos
 - Optimización de rendimiento
 - Fácil deploy y producción
-