



**TECNOLÓGICO
NACIONAL DE MÉXICO®**



TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE TEPIC

IW

INVESTIGACIÓN

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Integrantes:

21400702_Hernandez_Samaniega_OscarGabriel

21400701_Hernandez_Ortiz_Luis_Eduardo_Antonio

21400645_Arellano_Garcia_Alessandro_Javier

19400576_Gomez_Lopez_Luis_Antonio

5-6-2025

1. ¿Qué es React?

React (también conocido como React.js o ReactJS) es una biblioteca de JavaScript de código abierto para construir interfaces de usuario, principalmente para aplicaciones web de una sola página (SPA). Fue desarrollada y mantenida por Facebook y una comunidad de desarrolladores independientes.

Su objetivo principal es facilitar la creación de interfaces interactivas, rápidas y reutilizables mediante el uso de componentes.



2. Historia

- **Creación:** React fue lanzado por primera vez en 2013 por Facebook.
- **Motivación:** Fue creado para solucionar problemas de rendimiento y mantenimiento en aplicaciones web complejas, donde el DOM (Document Object Model) se actualizaba frecuentemente.
- **Innovación:** React introdujo el concepto del **DOM virtual** para minimizar las actualizaciones reales al DOM del navegador, optimizando la velocidad y experiencia de usuario.



3. Características principales

3.1 Componentes

React está basado en la creación de **componentes**, que son piezas reutilizables e independientes que representan partes de la interfaz de usuario (botones, formularios, menús, etc.).

- Pueden ser **componentes funcionales** o **componentes de clase**.
- Cada componente puede tener **estado** y **props** (propiedades).

3.2 JSX (JavaScript XML)

React utiliza JSX, una extensión de sintaxis que permite escribir código similar a HTML dentro de JavaScript, facilitando la creación de componentes visuales.

Ejemplo:

```
const saludo = <h1>Hola, mundo!</h1>;
```

3.3 Virtual DOM

React mantiene una copia virtual del DOM y cuando se cambia el estado de un componente, compara el DOM virtual con el real para aplicar solo los cambios necesarios. Esto mejora significativamente el rendimiento.

3.4 Unidireccionalidad de datos

React usa un flujo de datos unidireccional (de padre a hijo), lo que facilita el seguimiento y control de los datos en la aplicación.

3.5 Hooks

Introducidos en React 16.8, los **hooks** permiten usar estado y otras características de React sin escribir una clase.

Ejemplo de hook useState:

```
import React, { useState } from 'react';

function Contador() {
  const [contador, setContador] = useState(0);

  return (
    <div>
      <p>Has clickeado {contador} veces</p>
      <button onClick={() => setContador(contador + 1)}>Click me</button>
    </div>
  );
}
```

4. Ventajas de React

- **Rendimiento:** Gracias al Virtual DOM.
- **Componentización:** Facilita la reutilización y mantenimiento del código.
- **Gran comunidad y ecosistema:** Amplia cantidad de librerías, herramientas y recursos.
- **SEO Friendly:** Con renderizado del lado servidor (SSR) mediante frameworks como Next.js.
- **Desarrollo rápido:** JSX y herramientas modernas mejoran la productividad.
- **Multi-plataforma:** React Native permite usar React para aplicaciones móviles nativas.

5. Desventajas

- **Curva de aprendizaje:** JSX y conceptos como el estado y ciclo de vida pueden ser confusos para principiantes.

- **Solo vista:** React es solo la capa de vista, por lo que se necesitan otras librerías para manejo completo de rutas, estados complejos, etc.
- **Frecuentes actualizaciones:** Algunas veces puede ser difícil seguir las actualizaciones y cambios.

6. Ecosistema y herramientas comunes

- **Redux:** Manejo avanzado del estado global.
- **React Router:** Navegación y rutas para SPA.
- **Next.js:** Framework para renderizado server-side y generación estática.
- **Create React App (CRA):** Herramienta oficial para arrancar proyectos React rápidamente.
- **React DevTools:** Extensión para depurar aplicaciones React.

7. Casos de uso

- Aplicaciones web interactivas y SPA (p.ej. Facebook, Instagram).
- Dashboards y paneles administrativos.
- Aplicaciones móviles con React Native.
- Aplicaciones que requieren actualización frecuente de la UI.

8. Futuro de React

React continúa evolucionando con mejoras en rendimiento, nuevas APIs (como Concurrent Mode y Suspense), y mayor integración con herramientas modernas. La comunidad sigue siendo activa y la adopción sigue creciendo.