
**Programación de Internet**

**Estudiante:**

José Ignacio Bermúdez Fernández

Sede Virtual, Universidad Central

Prof. Fabián Chinchilla Mayorga

Investigación: Fundamentos, Estructura y Usos de React en el Desarrollo Web Moderno

III Cuatrimestre 2025

# Fundamentos, Estructura y Usos de React en el Desarrollo Web Moderno

## Objetivo

Analizar las características, estructura y ventajas de React, con énfasis en ejemplos prácticos de código que permitan comprender su funcionamiento y aplicaciones habituales.

## ¿Qué es React? Origen y evolución

Tal y como menciona Monteserín, P. (2022). el React es “una librería Javascript de código abierto, desarrollada por Facebook, con la que podremos crear interfaces de usuario de tipo SPA (single page applications)”. Donde de paso se acota que para dominar el React, se debe de tener conocimiento esencial de JavaScript y Node.js.

En las páginas web modernas, es necesario manipular el DOM constantemente. El problema es que hacerlo a menudo puede afectar seriamente al rendimiento de la aplicación. React utiliza un DOM virtual, lo que significa que todas las actualizaciones ocurren en la memoria (esta operación es más rápida que manipular el DOM real directamente).

De hecho, grandes y reconocidas empresas internacionales como Airbnb, Microsoft, Netflix, Disney, Dropbox, Twitter, PayPal, Salesforce, Tesla y Uber utilizan React de forma generalizada en sus proyectos.

## Características y ventajas principales de React frente a otros enfoques

Según Santana, C. (2020) menciona las siguientes características y ventajas principales de React:

**(i)** **let y const.** La nueva forma de declarar variables es mediante let o const. Puede utilizar let para declarar variables que pueden cambiar sus valores, pero solo en el ámbito de un bloque del programa. La diferencia entre let y var es que let es una variable en el ámbito de bloque, por lo que no puede ser global, y con var puede declarar una variable global. Por ejemplo:

```
var name = ‘Carlos Santana’;
let age = 30;
console.log(window.name); // Carlos Santana
console.log(window.age); // No definido
```

**(ii) const:** También, al utilizar objetos, puede añadir, eliminar o modificar nodos:

```
const person = {
  name: 'Carlos Santana',
  age:
  30,
  email: 'carlos@milkzoft.com'
};
// Adición de un nuevo nodo...
person.website = 'https://www.codejobs.com';

// Eliminación de un nodo...
delete person.email;

// Actualización de un nodo...
person.age = 29;
```

**(iii) rest:** El parámetro rest es una matriz que contendrá el resto de los parámetros de una función cuando el número de argumentos es superior al número de parámetros especificado:

```
function setNumbers(param1, param2, ...args) {
  // param1 = 1
  // param2 = 2
  // args = [3, 4, 5, 6];
  console.log(param1, param2, ...args); // Log: 1, 2, 3, 4, 5, 6
}
setNumbers(1, 2, 3, 4, 5, 6);
```

**Concepto de componentes y clasificación**

Dentro de lo mencionado por Santana, C. (2020) se pueden mencionar los siguientes componentes:

**1. Componentes Funcionales:** Son simples funciones de JavaScript que aceptan un objeto de propiedades (props) como argumento y devuelven elementos de React (JSX) que describen lo que debe aparecer en la pantalla. Uso Moderno: Son la forma preferida y moderna de escribir componentes en React, ya que el uso de Hooks (como useState y useEffect) les permite manejar el estado y el ciclo de vida sin necesidad de usar clases.

**2. Componentes de Clase:** Son clases de ES6 que extienden de React.Component y requieren un método render() para devolver el JSX.

**Ciclo de Vida y Hooks más usados (useState, useEffect, useMemo)**

Acerca del Ciclo de Vida y Hooks más usados, Monteserín, P. (2022). los resume de la siguiente manera:

**userState**: Añade estado local a los componentes funcionales. Permite que el componente almacene y gestione datos que pueden cambiar con el tiempo.

Ejemplo: const [variable, setVariable] = useState(valorInicial);

**useEffect**: Permite realizar efectos secundarios (tareas que interactúan con el mundo exterior de React), como llamadas a API, manipulación manual del DOM.

Ejemplo: useEffect(() => { /* Código del efecto */ }, [dependencia1, dependencia2]);

**useMemo:** Se utiliza para memorizar (almacenar en caché) el resultado de una función costosa y solo recalcularlo cuando las variables de sus dependencias cambian.

Ejemplo: const resultadoMemoizado = useMemo(() => calculoCostoso(a, b), [a, b]);

**Context API o Redux**

Las diferencias entre Context API y Redux según Monteserín, P. (2022). son los siguientes:

- Redux es una librería totalmente externa a React. Context API, sin embargo, forma parte del core de React.

Redux permite ejecutar funciones cuando una variable ha cambiado. Con Context API, al actualizarse una propiedad de un objeto, se actualizarán todos los componentes; mientras que con Redux solo se actualizarán los componentes que utilicen dicha propiedad. Por tanto, usaremos Context API para cambios poco frecuentes en los estados globales o en aplicaciones con pocos estados globales, y Redux para grandes aplicaciones con cambios constantes en los estados globales. Context API es más sencillo de usar que Redux.

**Angular (Breve Mención)**

Según Santana, C. (2020) Angular es “un framework de código abierto, basado en TypeScript, desarrollado y mantenido por Google. A diferencia de React, que es una librería enfocada solo en la capa de la vista.”

**Conclusión (perspectiva personal)**

Según lo aprendido el react y tomando tanto las referencias de Monteserín Fernández, P. (2022) como de Santana Roldán, C. (2020) recomiendo usar React tal y como lo hacen muchas empresas para desarrollo de aplicaciones de escritorio multiplataforma (como el VS Code) y servicios de web complejos, en creación de interfaces altamente interactivas y personalizadas, la rápida actualización de datos, almacenamiento de información.

**Referencias Bibliográficas:**

Monteserín Fernández, P. (2022). *Aprender React con 100 ejercicios prácticos:* (1 ed.). Marcombo. https://elibro.net/es/lc/ucentral/titulos/281872

Santana Roldán, C. (2020). *Ejercicios prácticos con React:* (1 ed.). Marcombo. https://elibro.net/es/lc/ucentral/titulos/281193