

REACT

NICOLAS OBREGON ROJAS

Que es un componente en react??

- Un **componente** en React es una pieza reutilizable de la interfaz de usuario. Esencialmente, es una función o clase en JavaScript que devuelve elementos JSX para renderizar en la pantalla. Los componentes pueden ser **funcionales** (basados en funciones) o **de clase** (basados en clases, aunque están en desuso con la introducción de hooks).

¿Para qué sirve?

Los componentes permiten:

- **Reutilización de código:** Se pueden usar en diferentes partes de la aplicación sin duplicar código.
- **Modularidad:** Separa la lógica en pequeñas piezas independientes, facilitando el mantenimiento.
- **Manejo del estado:** Mediante `useState` y otros hooks, los componentes pueden gestionar datos y responder a eventos de usuario.

¿Cómo se usa en React?

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

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

  return (
    <div>
      <p>Contador: {contador}</p>
      <button onClick={() => setContador(contador + 1)}>Incrementar</button>
    </div>
  );
};

export default Contador;
```

Este componente Contador:

- Usa el **estado** (useState) para almacenar el valor del contador.
- Renderiza un botón que al hacer clic **actualiza el estado**.
- Es independiente y puede reutilizarse en cualquier parte de la aplicación.

¿Cuándo no es recomendable usar un componente en React?

- **Cuando la lógica es demasiado simple y no se reutilizará:** Si solo necesitas un pequeño bloque de JSX en un solo lugar, usar un componente puede ser innecesario.
- **Cuando el rendimiento se ve afectado:** Excesivos componentes pueden generar renderizados innecesarios. Se debe optimizar con `React.memo()` o `useCallback()`.
- **Cuando se necesita modificar el DOM directamente:** React maneja el Virtual DOM, por lo que manipulaciones directas con `document.querySelector()` pueden ser problemáticas. En su lugar, usa `useRef()`.
- **Cuando es más fácil manejarlo con una solución global:** Si se necesita compartir estado en múltiples componentes, usar Context API o Redux puede ser mejor que pasar demasiadas props.