



# Estados base

## clase n° 14

\*REC



recordá  
poner a grabar la clase

# estados en react

El estado en React es un concepto fundamental para **crear aplicaciones interactivas**. Representa cualquier dato que pueda cambiar durante la ejecución de un componente, como valores de formularios, contadores, o datos de usuarios.

En términos simples, el estado le dice a React qué información necesita para renderizar correctamente un componente en un momento dado. Cada vez que el estado cambia, React vuelve a renderizar el componente, mostrando la versión actualizada de la interfaz.

# estados en react

¿Por qué necesitamos el estado?

Antes de profundizar en useState, es importante entender por qué el estado es crucial:

■ **Interactividad:** Permite que la interfaz responda a las acciones del usuario (como clics, escritura, o selección).

■ **Consistencia:** El estado garantiza que los datos visualizados estén siempre sincronizados con los datos internos del componente.

■ **Mantenimiento:** Facilita el desarrollo y depuración de aplicaciones, al mantener el control sobre cómo los datos cambian y afectan la interfaz.

En aplicaciones sin React, se necesitarían muchas líneas de código para actualizar el DOM manualmente cada vez que algo cambia. React simplifica esto con su modelo declarativo, que actualiza automáticamente la interfaz basada en el estado.

# el hook useState

React introduce los hooks como una forma de manejar el estado y otros ciclos de vida dentro de componentes funcionales. El hook useState es el más básico y esencial. Sintaxis de useState:

```
const [estado, setEstado] = useState(valorInicial);
```

- **estado:** Es la variable que almacena el valor actual del estado.
- **setEstado:** Es la función que se utiliza para actualizar el valor del estado.
- **valorInicial:** Es el valor inicial que le damos al estado cuando se carga el componente por primera vez.

# el hook useState

## ¿Cómo funciona?

- Al cargar el componente, **useState** inicializa el estado con el **valorInicial**.
- Cuando se llama a **setEstado** con un nuevo valor, React actualiza el estado y vuelve a renderizar el componente.
- Esto asegura que la interfaz esté sincronizada con el estado más reciente.

## Ciclo de vida del estado

El estado en React pasa por varias etapas durante la ejecución del componente:

**Primer paso.** Inicialización: Se define el estado inicial usando useState.

```
const [contador, setContador] = useState(0);
```

**Segundo paso.** Lectura: Se accede al valor del estado para usarlo en el JSX o lógica del componente.

```
<p>El contador vale: {contador}</p>
```

**Tercer paso.** Actualización: Se usa la función setEstado para actualizar el estado y, como resultado, React vuelve a renderizar el componente.

```
const incrementar = () => setContador(contador + 1);
```

**Cuarto paso.** Renderización: React renderiza el componente con el estado actualizado.

## Ejemplo básico: contador con estado en React

En este ejemplo, vamos a crear un contador que permite incrementar, decrementar y resetear su valor usando estados en React.

```
1 import React, { useState } from "react";
2 function Contador() {
3   const [contador, setContador] = useState (0);
4   const incrementar = () => setContador (contador + 1);
5   const decrementar = () => setContador (contador - 1);
6   const resetear = () => setContador (0);
7   return (
8     <div>
9       <h1>Contador: {contador}</ h1>
10      <button onClick={incrementar}>Incrementar</ button>
11      <button onClick={decrementar}>Decrementar</ button>
12      <button onClick={resetear}>Resetear</ button>
13    </div>
14  );
15 }
16 export default Contador;
```



# buenas prácticas

**Organización:** Mantené el estado lo más simple posible. Evitá incluir datos innecesarios.

**División de Responsabilidades:** Usá múltiples estados cuando sea lógico, en lugar de un único estado grande y complicado.

```
const [nombre, setNombre] = useState('');  
const [edad, setEdad] = useState(0);
```

**Predicción:** Siempre inicializá el estado con un valor que tenga sentido para tu aplicación.



revolución\*  
digital\_