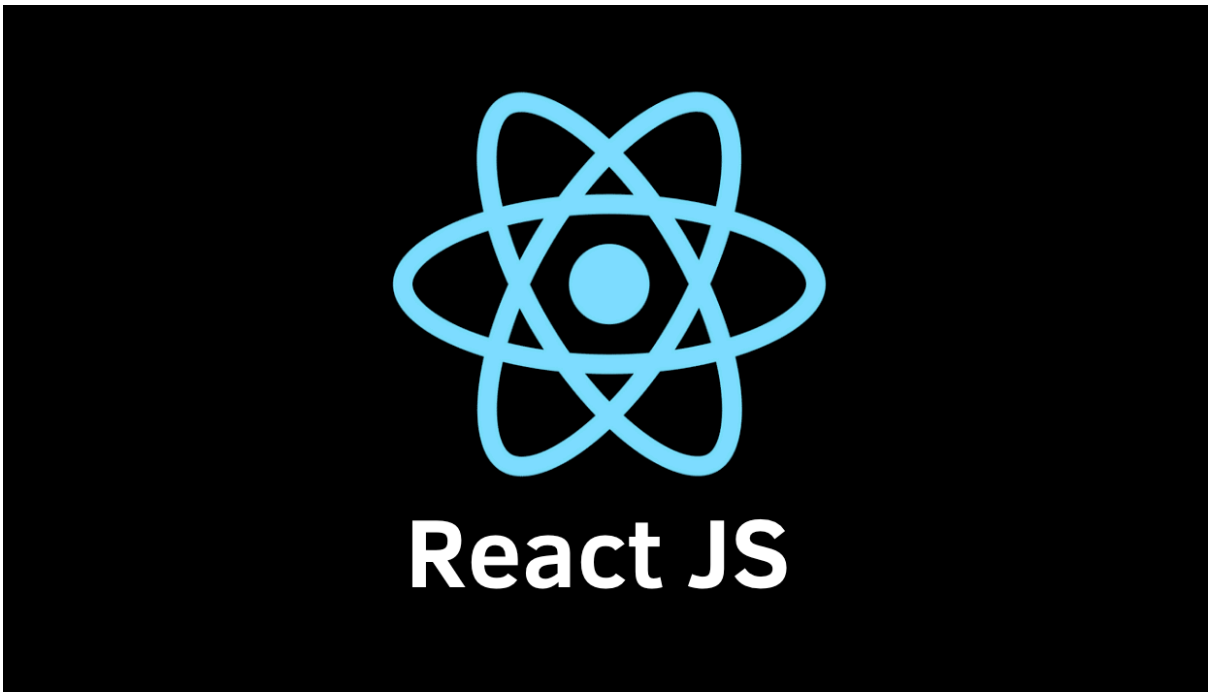
Desarrollo web en entorno cliente - Nivel 6 React

Pablo Valero López *22/01/2026*

2ºDAW



Índice:

Objetivo

En esta práctica vas a aprender a reaccionar a interacciones del usuario (click, cambio de texto y envío de formularios) usando manejadores de eventos en React. Verás cómo React te entrega un evento sintético y cómo combinar eventos con estado para actualizar la interfaz de manera dinámica.

Parte A: onClick y el objeto event

En la siguiente imagen se muestra el código de onClick.

App.jsx M × main.jsx

Desarrollo_Web_Entorno_Cliente > nivel1_react > my-react-app >

```jsx
function App(){
    const handleClick = (event) => {
        console.log('Hola desde React!')
        console.log(event)
    }

    return (
        <button onClick={handleClick}>
            Click me
        </button>
    )
}

export default App
```

Imagen 1: Código de onClick. Elaboración propia.

En las siguientes imágenes se muestran los resultados de onClick.

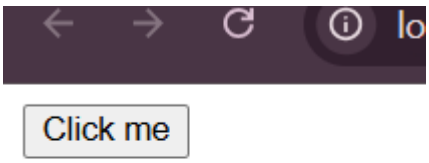


Imagen 2: Resultado de onClick. Elaboración propia.

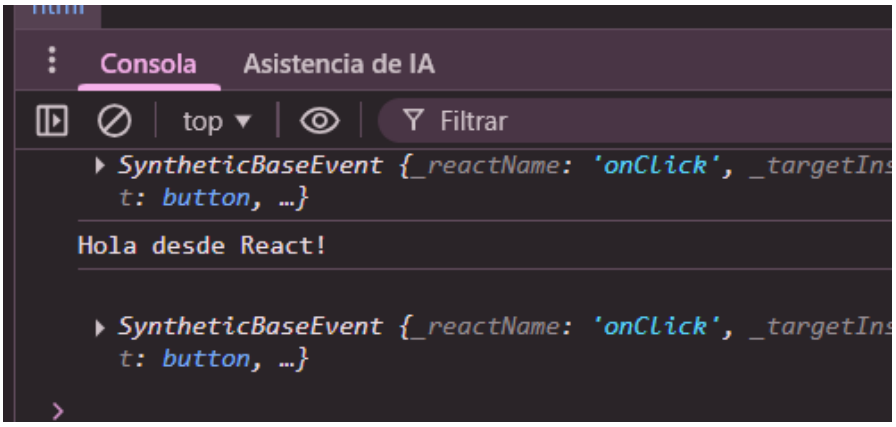


Imagen 3: Resultado de onClick. Elaboración propia.

Parte B: Cambiar la UI con estado + evento

En la siguiente imagen se muestra el código para cambiar el estado.

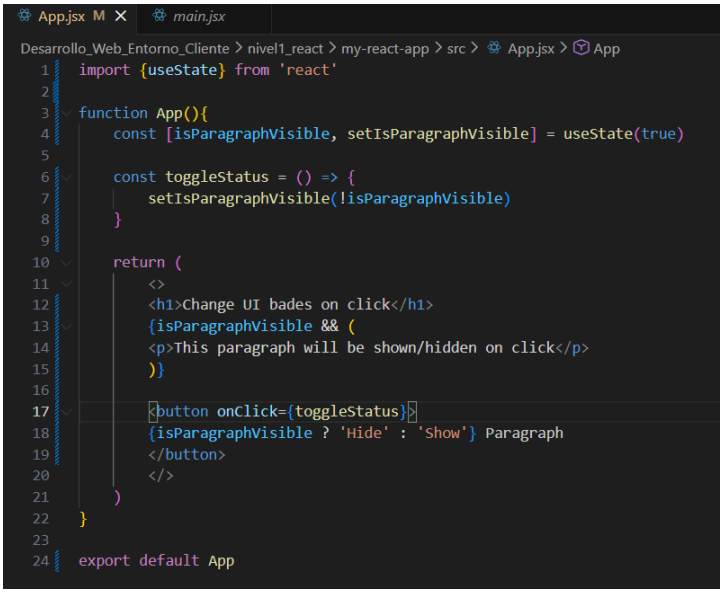


Imagen 4: Código de evento. Elaboración propia.

En la siguientes imagenes se muestran el resultado del evento.

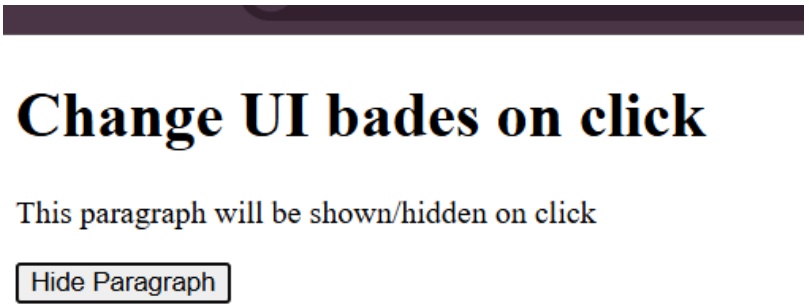


Imagen 5: Código de evento. Elaboración propia.

Imagen 6: Código de evento. Elaboración propia.

Parte C: onChange (entrada controlada)

En la siguiente imagen se muestra el código de una entrada controlada.

App.jsx M main.jsx

Desarrollo_Web_Entorno_Cliente > nivel1_react > my-react-app > src > App.jsx > default

```jsx
import {useState} from 'react'

function App(){
    const [nickname, setNickname] = useState('')

    const toggleStatus = () => {
        setIsParagraphVisible(!isParagraphVisible)
    }

    return (
        <>
        <h1>Alias del jugador</h1>
        <input type="text"
        placeholder='Escribe tu alias'
        value={nickname}
        onChange={(e=>setNickname(e.target.value))} />
        <p>Tu alias es: <strong>{nickname || '...'}</strong></p>
        </>
    )
}

export default App
```

Imagen 6: Código de entrada controlada. Elaboración propia.

En las siguientes imágenes se muestran los resultados de entrada controlada

Imagen 6: Resultado de entrada controlada. Elaboración propia.

Imagen 7: Resultado de entrada controlada. Elaboración propia.

Parte D: onSubmit (formulario)

En la siguiente imagen se muestra el código con el formulario.

Imagen 8: Código de formulario. Elaboración propia.

En la siguiente imagen se muestra el resultado del formulario.

Imagen 9: Resultado de formulario. Elaboración propia.

Miniretillo (pero esta vez sin ayuda)

En las siguientes imágenes se muestra el código de minireto

Imagen 10: Código minireto. Elaboración propia.

```
    return (
        <>
        <h1>Miniretillo</h1>
        {isParagraphVisible && (
        <p>Este es un texto oculto.</p>
        )}
        <button onClick={toggleStatus}>
        {isParagraphVisible ? 'Esconder' : 'Mostrar'} texto
        </button>
        <p>Contador {newContador}</p>
        <button onClick={() => setNewContador((n) => n + 1)}>+1
        numero</button>
        <button onClick={() => setNewContador(0)}>Reset</button>
        <br/>
        <br/>
        <input type="text" value={modo} placeholder='Añade algo (papafrita)' onChange={(e)=>setModo(e.target.value)}/>
        {modo && <p>{modo}</p>}
        <br/>
        <br/>
        {estadoBoton && (
            <p>Este botón ya estaba al inicio.</p>
        )}
        <button onClick={cambiarBoton}>{estadoBoton ? 'Ocultar' : 'Mostrar'}</button>
        </>
    )
}

export default App
```

Imagen 11: Código minireto. Elaboración propia.

En la siguiente imagen se muestra el resultado del minireto.

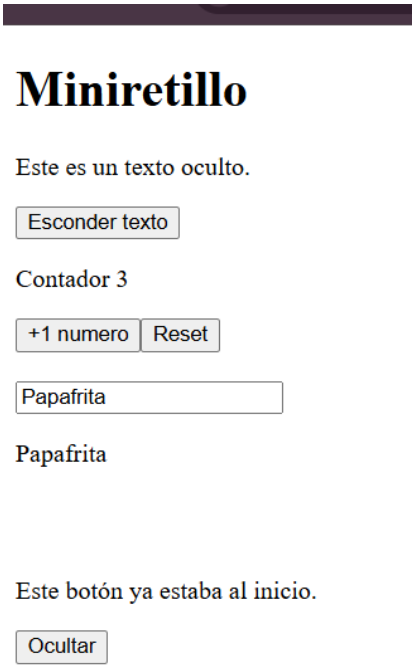


Imagen 12: Resultado miniretro. Elaboración propia.