

# DÍA 17 – EVENTOS Y FORMULARIOS EN REACT

## Objetivo del día

- Comprender cómo funcionan los **eventos** en React (`onClick`, `onChange`).
- Crear **inputs controlados** con `useState`.
- Validar datos de formularios (condicionales).
- Renderizar mensajes dinámicos (éxito o error).

## ¿Qué es un evento en React?

Analogía: Un evento es como un interruptor.  
Haces clic → pasa algo. Tecleas algo → se actualiza.

Ejemplos comunes:

- `onClick` → cuando haces clic en botón.
- `onChange` → cuando escribes en un input.

## Input controlado

Concepto clave: El valor de tu input depende del estado (`useState`).

Analogía: Imagina que **tu input** es un vaso de agua. El `useState` decide **cuánta** agua hay.  
Cada `onChange` es **una** gota que cae **y** actualiza **el** vaso.

```
const [nombre, setNombre] = useState("");  
<input value={nombre} onChange={(e) => setNombre(e.target.value)} />
```

## Validar datos

Usa condicionales para verificar campos:

- Email debe contener @
- Checkbox aceptado = `true`
- Si falta algo, muestra mensaje de `error`.

## Mensajes dinámicos

Renderiza mensajes según estado:

```
{error && <p style={{color: "red"}}>Email inválido</p>  
{success && <p style={{color: "green"}}>Formulario enviado!</p>}
```

## Resumen de patrón básico

Concepto	Qué hace	Analogía
useState	Guarda el valor del <b>input</b>	El vaso <b>que</b> guarda el agua
onChange	Actualiza el estado cuando escribes	Gotas llenando el vaso
Validación lógica	Comprueba condiciones antes <b>de</b> enviar	Puerta <b>que se</b> abre solo si ok
Mensaje condicional	Renderiza si algo es true	Luz <b>que se</b> enciende

## Frase para recordar

"Un formulario controlado es como un tablero **de** mandos:  
cada **input** tiene un cable conectado a **su** estado."

## EJERCICIO GUIADO FINAL

### Crear FormularioPerfil.jsx completo

#### Objetivo:

Un formulario **que**:

- Pida nombre y email.
- Valide **que** nombre **no esté** vacío y email incluya @.
- Tenga checkbox **de** "Acepto términos".
- Bloquee botón si algo falla.
- Muestre mensaje **de** éxito si todo **está** correcto.

#### Entrada:

- 2 inputs (nombre, email).
- 1 checkbox (aceptado).
- 1 botón enviar.
- 1 mensaje condicional.

#### Pasos Lógicos:

1. Crear useState para nombre, email, aceptado, mensaje.
2. Hacer inputs controlados con value y onChange.
3. Botón Enviar:
  - Si nombre vacío o email sin @ o no aceptado → error.
  - Si todo ok → mensaje éxito.
4. Botón Reset para limpiar todo.

## Código paso a paso:

```
// src/components/FormularioPerfil.jsx
import { useState } from 'react';

export default function FormularioPerfil() {
  // Vasos de agua para cada dato
  const [nombre, setNombre] = useState("");
  const [email, setEmail] = useState("");
  const [aceptado, setAceptado] = useState(false);
  const [mensaje, setMensaje] = useState("");

  // Acción al enviar
  const manejarEnvio = (e) => {
    e.preventDefault();
    if (nombre === "" || !email.includes("@") || !aceptado) {
      setMensaje("    Completa todos los campos correctamente.");
    } else {
      setMensaje("    Formulario enviado con éxito.");
    }
  };

  // Acción al resetear
  const manejarReset = () => {
    setNombre("");
    setEmail("");
    setAceptado(false);
    setMensaje("");
  };

  return (
    <form onSubmit={manejarEnvio}>
      <h2>Formulario Perfil</h2>

      { /* Nombre */ }
      <input
        type="text"
        placeholder="Tu nombre"
        value={nombre}
        onChange={(e) => setNombre(e.target.value)}
      />

      { /* Email */ }
      <input
        type="email"
        placeholder="Tu email"
        value={email}
        onChange={(e) => setEmail(e.target.value)}
      />

      { /* Checkbox */ }
      <div>
        <input
          type="checkbox"
          checked={aceptado}
          onChange={(e) => setAceptado(e.target.checked)}
        />
        <label>Acepto términos</label>
      </div>
    </form>
  );
}
```

```
    {/* Botón enviar */}
    <button type="submit" disabled={!aceptado}>
      Enviar
    </button>

    {/* Botón reset */}
    <button type="button" onClick={manejarReset}>
      Limpiar
    </button>

    {/* Mensaje */}
    {mensaje && <p>{mensaje}</p>}
  </form>
);
}
```

### Analogía general:

Cada campo es **una** tubería. Los useState guardan **lo** que entra, los onChange mantienen **el** flujo limpio. El botón solo abre **la** puerta si todas las tuberías están bien conectadas.