

# DÍA 17 – EJERCICIOS INTERACTIVOS:

## EVENTOS Y FORMULARIOS

---

### 1. Validar email con “@”

#### OBJETIVO

Crear un **input** que solo permita enviar si incluye “@”.

#### ENTRADA

- 1 input tipo email
- Botón enviar

#### PASOS LÓGICOS

1. ``useState`` para guardar email.
2. ``onChange`` actualiza email.
3. Validar con ``.includes("@")``.

#### ANALOGÍA

“Verifica que tu carta tiene **dirección**.”

#### Código ejemplo:

```
const [email, setEmail] = useState("");

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

<button disabled={!email.includes("@")}>Enviar</button>
```

## 2. Checkbox “Acepto términos”

### OBJETIVO

No permitir envío **sin** aceptar **términos**.

### ENTRADA

[x] Checkbox **booleano**

### PASOS LÓGICOS

1. `useState` para checkbox.
2. `onChange` actualiza.
3. `disabled` en botón si no aceptado.

### ANALOGÍA

“**Sin** contrato, **no** hay trato.”

## 3. Mensaje de éxito

### OBJETIVO

Mostrar mensaje solo si campos están bien.

### ENTRADA

Nombre + email + checkbox

### PASOS LÓGICOS

1. Validar todos.
2. Mostrar `<p>` si OK.

### ANALOGÍA

“Como un **sello** de aprobado.”

## 4. Botón reset

### OBJETIVO

Limpiar todo **con** botón.

### ENTRADA

Resetea nombre, email, checkbox, mensaje.

### PASOS LÓGICOS

1. Crear función reset.
2. Asignar onClick.

### ANALOGÍA

"Botón mágico de volver a empezar."

## 5. Push a GitHub

### OBJETIVO

Subir proyecto limpio.

### ENTRADA

- README.**md** **con**:
- Qué hace
- Cómo instalar
- Cómo correr npm run dev

### ANALOGÍA

"Tu trabajo **en la** nube, listo para enseñar."