

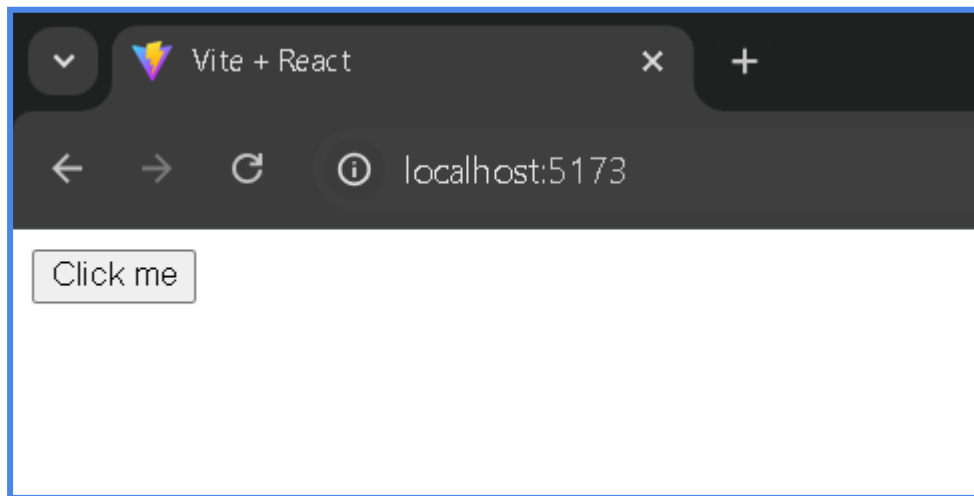
Nivel 6 React

DAW2-DWEC

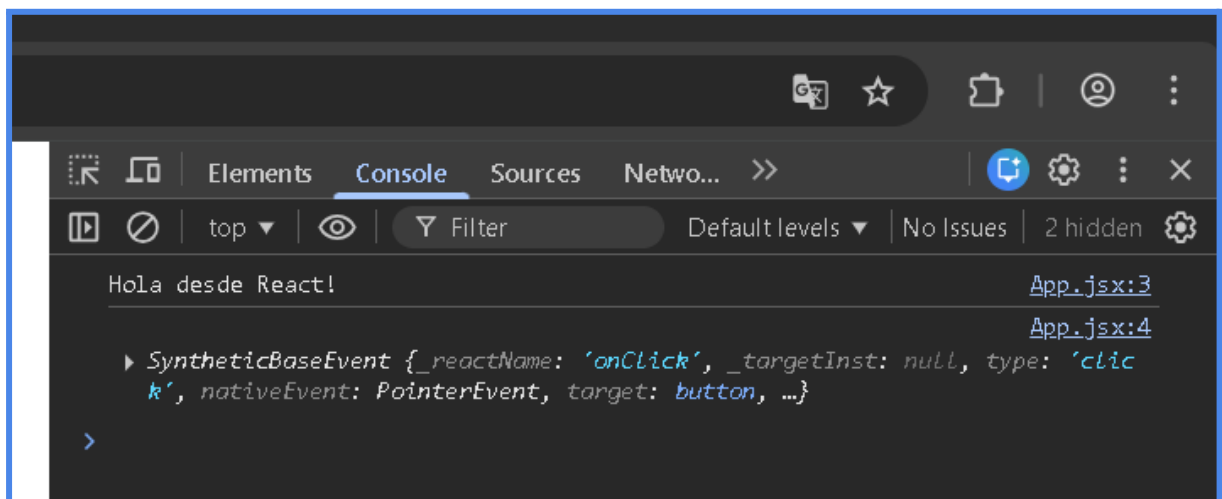
Índice

Parte A: onClick y el objeto event.....	2
Parte B: Cambiar la UI con estado + evento.....	3
Parte C: onChange (entrada controlada).....	3
Parte D: onSubmit (formulario).....	4
Miniretillo (pero esta vez sin ayuda).....	4
Preguntas extra para el PDF.....	10

Parte A: onClick y el objeto event

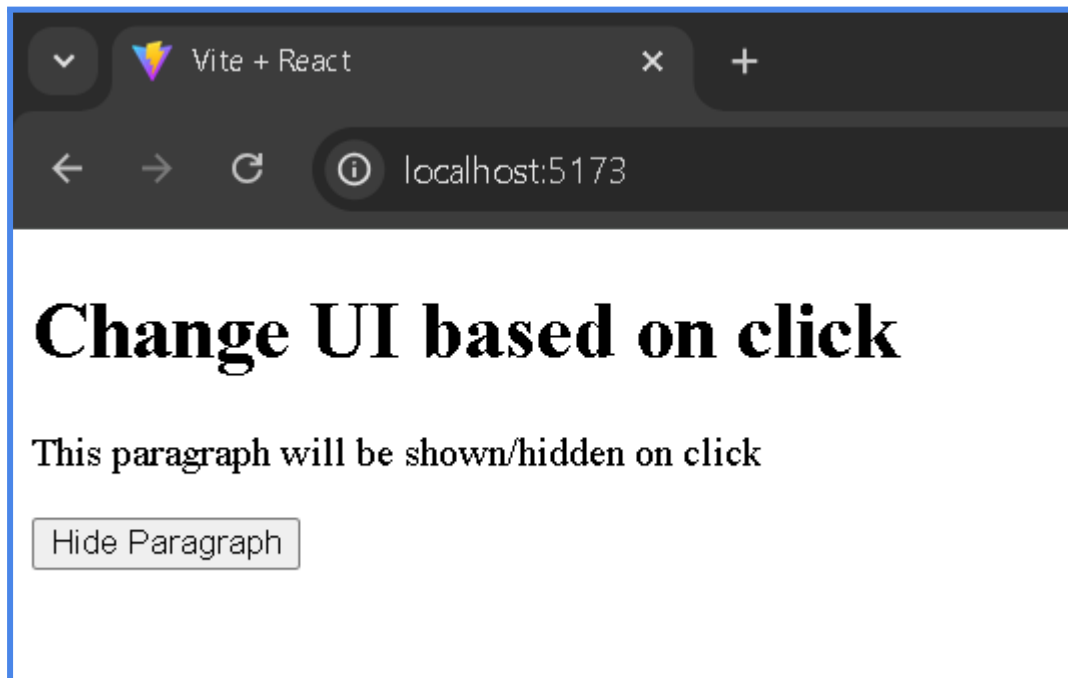


“Figura 01, Botón que llama una función handleClick, elaboración propia”



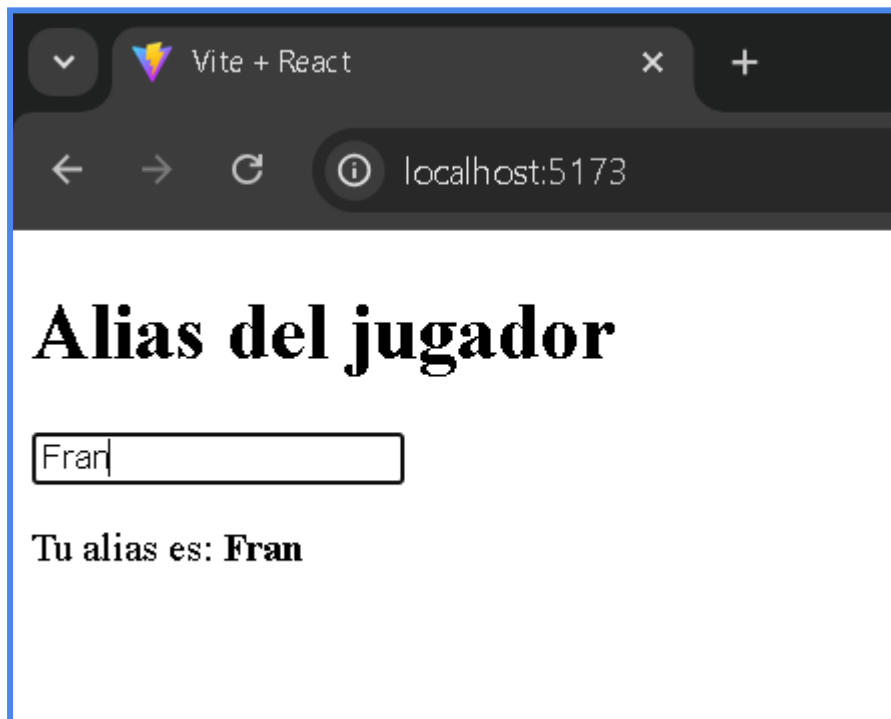
“Figura 02, comprobación en la consola, elaboración propia”

Parte B: Cambiar la UI con estado + evento



“Figura 03, botón que implementa un boton que muestra y oculte un párrafo, elaboración propia”

Parte C: onChange (entrada controlada)



“Figura 04, Alias de jugador con input controlado, elaboración propia”

Parte D: onSubmit (formulario)



“Figura 05, Registro con onSubmit, elaboración propia”

Miniretillo (pero esta vez sin ayuda)

1. Un botón que muestre/oculte una sección de texto (toggle).

```

ly-react-app > src > Panel.jsx > default
1  import { useState } from 'react';
2
3  function Panel() {
4      const [parrafo, setParrafo] = useState(true);
5      const [modo, setModo] = useState('');
6
7      const toggleStatus = () => {
8          setParrafo(!parrafo);
9      };
10
11     return (
12         <div>
13             <button onClick={toggleStatus}>
14                 {parrafo ? 'Esconder' : 'Mostrar'} Párrafo
15             </button>
16
17             {parrafo && (
18                 <p>Párrafo que desaparece al hacer click</p>
19             )}
20
21             <hr />

```

“Figura 06, botón de frase oculto, elaboración propia”

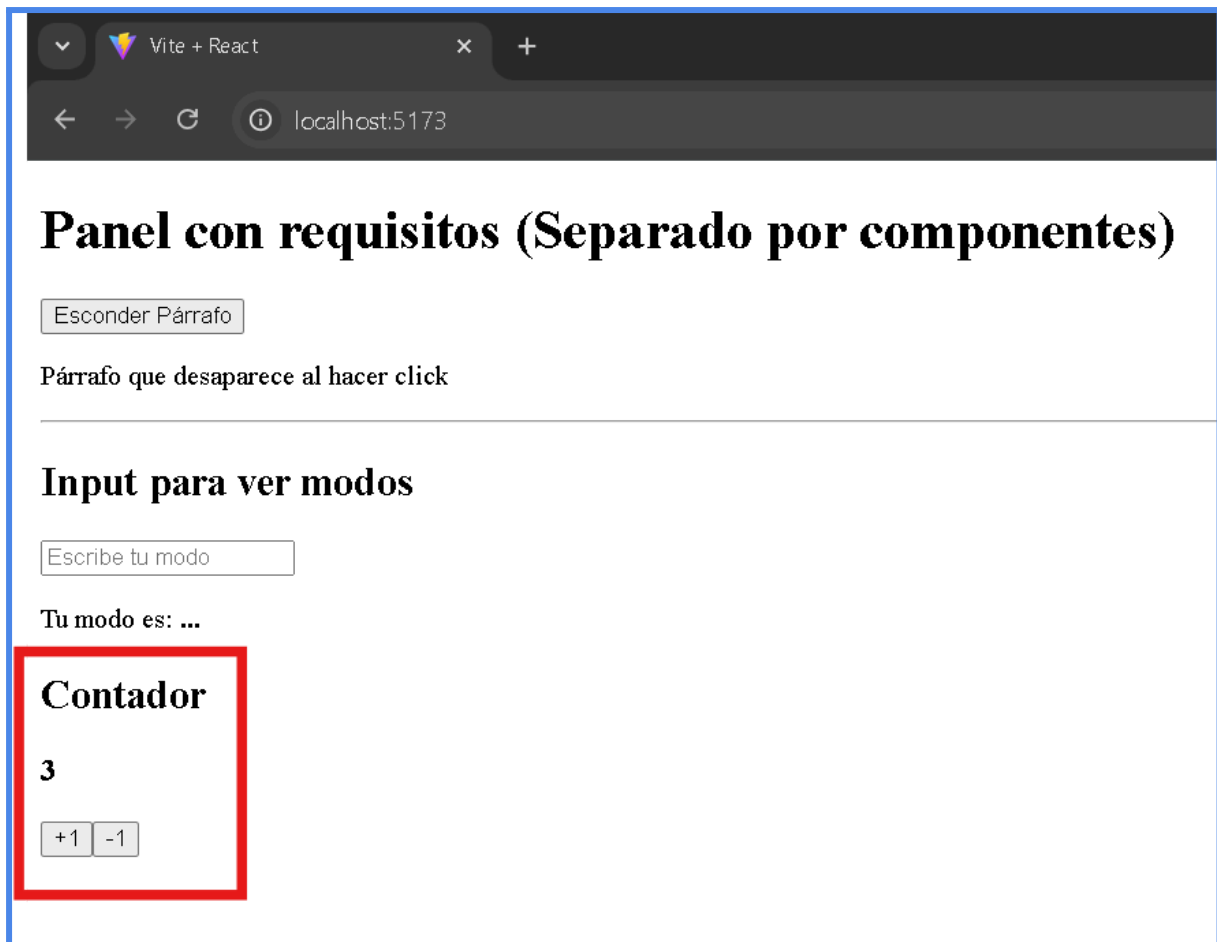


“Figura 07, muestra en pantalla, elaboración propia”

2. Un contador con botón +1 y botón -1 (mínimo 0).

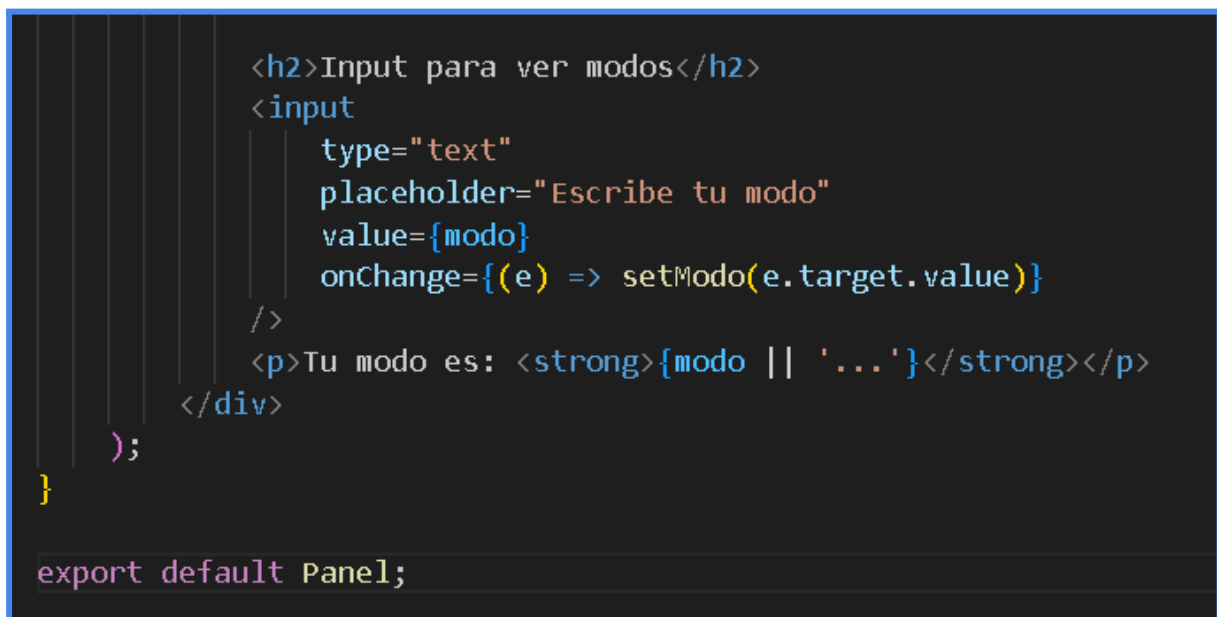
```
App.jsx Contador.jsx X Panel.jsx main.jsx
my-react-app > src > Contador.jsx > Contador
1  import { useState } from 'react';
2
3  function Contador() {
4    const [contador, setContador] = useState(0);
5
6    const incrementarContador = () => {
7      setContador(contador + 1);
8    };
9
10   const decrementarContador = () => {
11     if (contador > 0) {
12       setContador(contador - 1);
13     }
14   };
15
16   return (
17     <div>
18       <h2>Contador</h2>
19       <h3>{contador}</h3>
20       <button onClick={incrementarContador}>+1</button>
21       <button onClick={decrementarContador}>-1</button>
22     </div>
23   );
24 }
25
26 export default Contador;
```

“Figura 08, botones para sumar y restar números, elaboración propia”

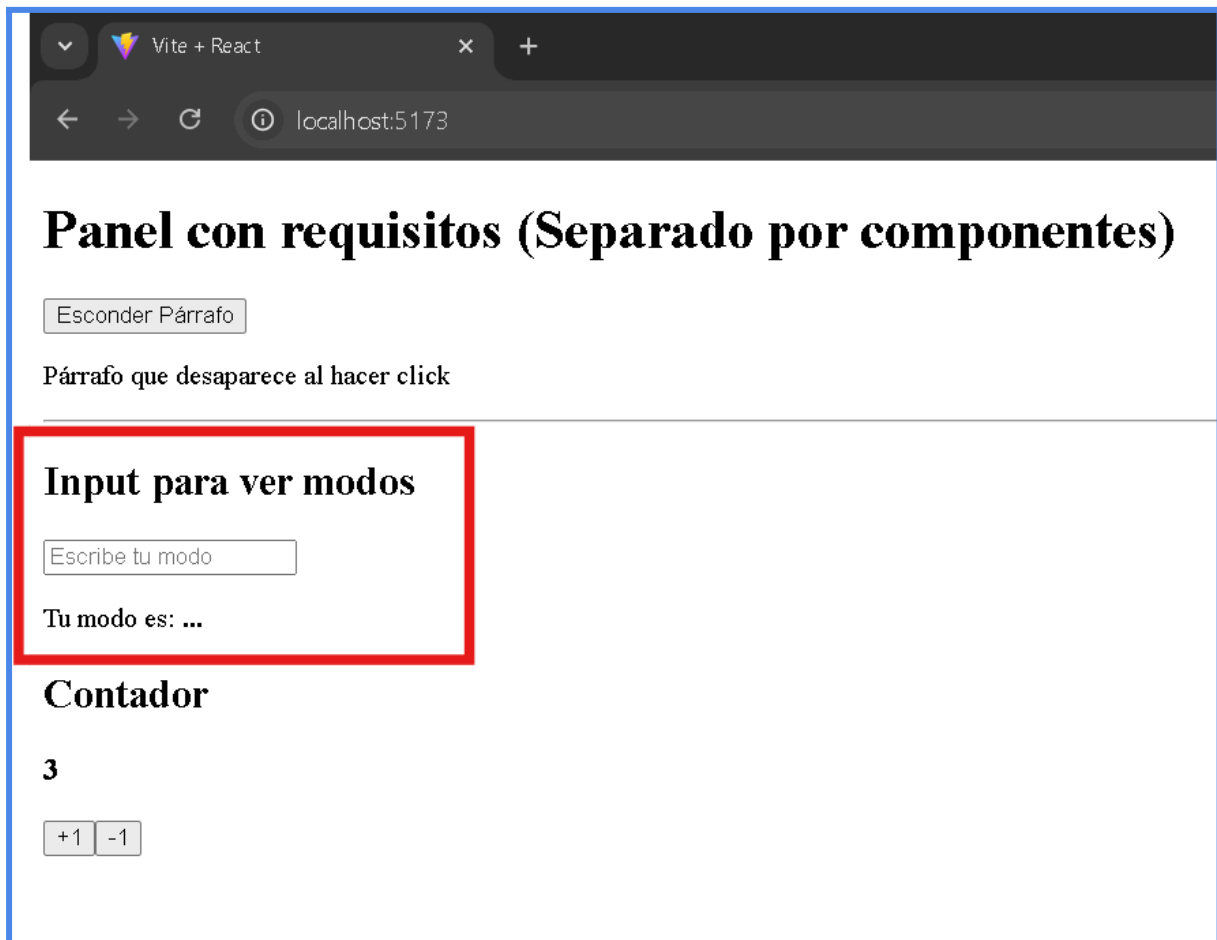


“Figura 09, muestra en pantalla, elaboración propia”

3. Un input para escribir un "modo" (por ejemplo: stealth, turbo, chill) y que se vea en pantalla en tiempo real.



“Figura 10, input de modo, elaboración propia”



“Figura 11, muestra en pantalla, elaboración propia”

4. Opcional (venga no): cambia el texto del botón según el estado (mostrar/ocultar).

```
<button onClick={toggleStatus}>
  {parrafo ? 'Esconder' : 'Mostrar'} Párrafo
</button>
```

“Figura 12, cambio de texto, elaboración propia”

Preguntas extra para el PDF

1. ¿Qué diferencia hay entre onClick y onSubmit?

- El onClick detecta cuando le doy click con el ratón a cualquier cosa, mientras que onSubmit se puede hacer click tanto con el ratón o con la tecla enter.

2. ¿Por qué usamos e.preventDefault() en un formulario?

- Porque si no lo uso la página empieza a parpadear porque está cargando la página entera, si lo uso se carga de forma más suave y que no se puede ver

3. ¿Qué es una "entrada controlada" y por qué usamos value + onChange?

- Es cuando el que controla el input es React y no el navegador
- uso value para hacer que el input muestre lo que dice el estado si o si, y uso onChange para actualizar el estado mientras que escribo

4. En tu mini-reto, que estado(s) manejas y que evento(s) los actualizan?

- Manejo 3 estados:
 - booleano para el párrafo que se muestra y se oculta
 - número para el contador que suma y resta numeros
 - texto para mostrar el modo en el que estoy
- Los actualizo con onClick para los botones y el contador y onChange para poder capturar lo que pongo en el input.