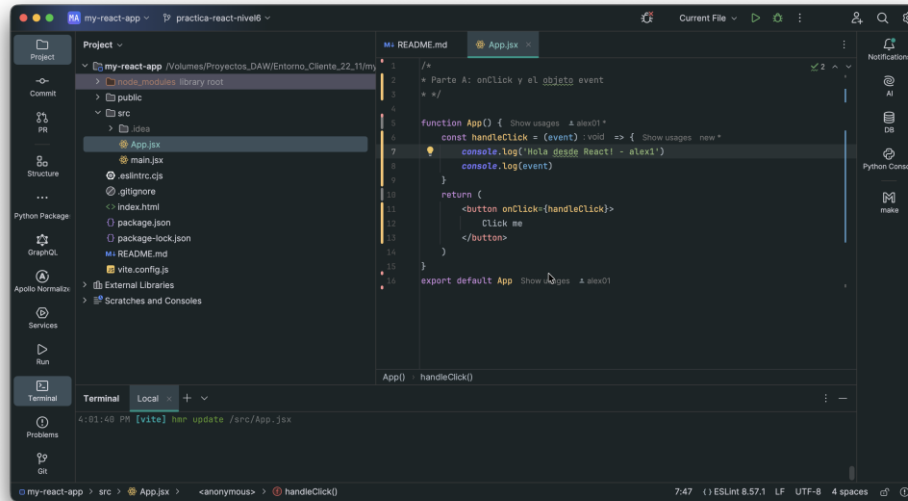


## 1- Parte A: onClick y el objeto event

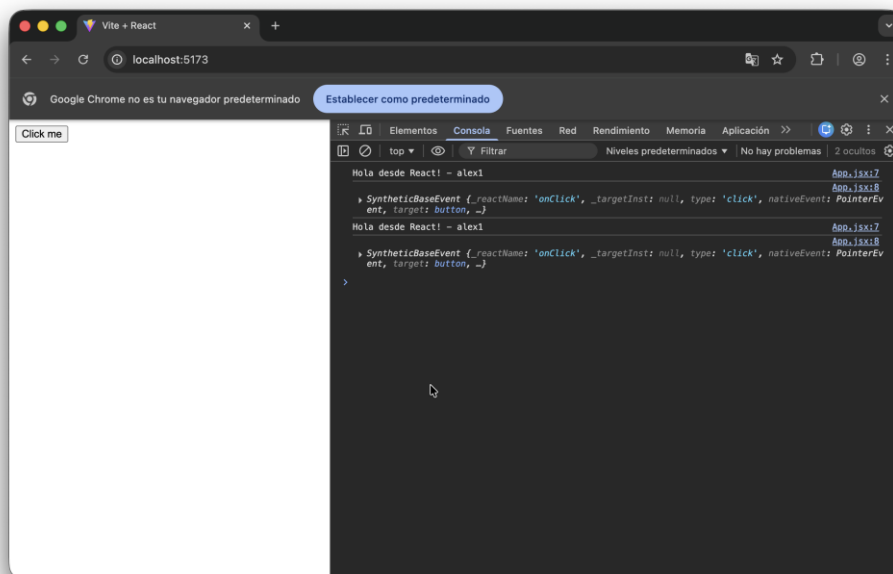
- Código



```
1  /*
2  * Parte A: onClick y el objeto event
3  */
4
5  function App() { Show usages & alex01 *
6    const handleClick = (event) => { Show usages new *
7      console.log('Hola desde React! - alex1')
8      console.log(event)
9    }
10
11    return (
12      <button onClick={handleClick}>
13        Click me
14      </button>
15    )
16  }
17
18  export default App Show usages & alex01
```

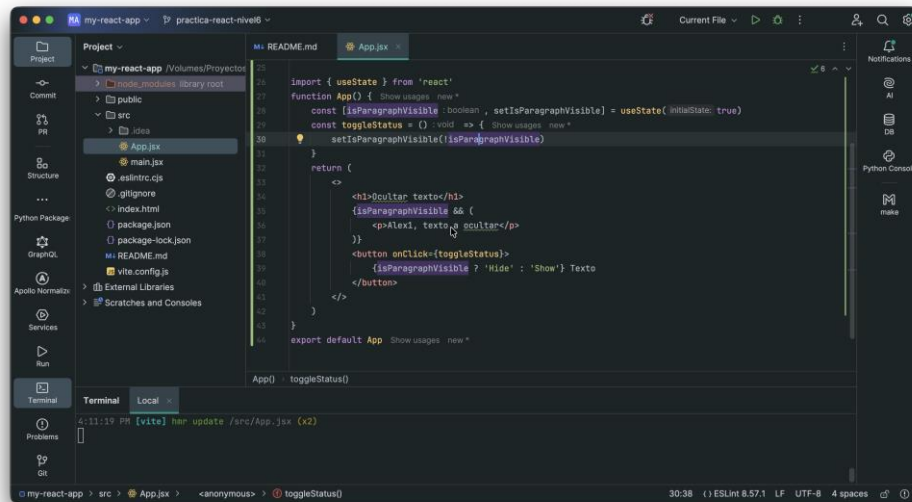
Terminal: 4:01:48 PM [vite] dev update /src/App.jsx

- Resultado

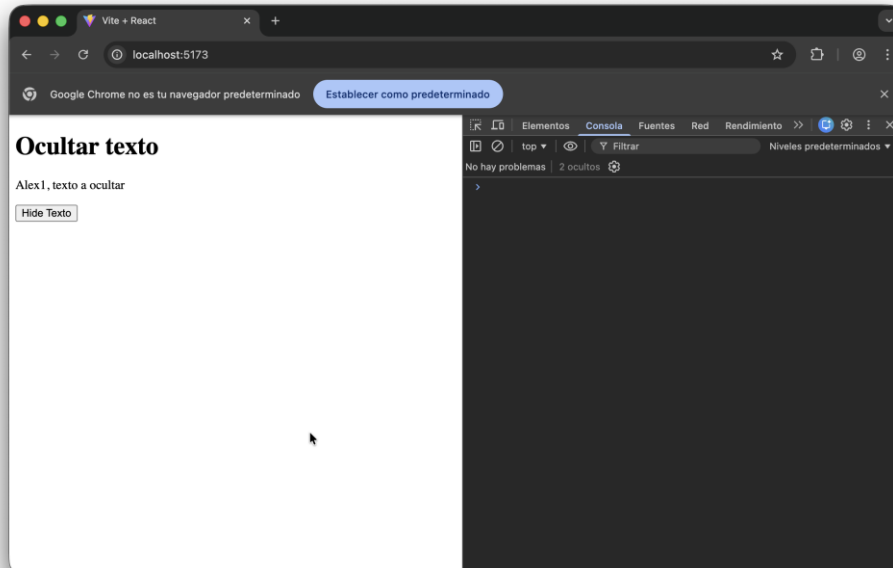


## 2- Parte B: Cambiar la UI con estado + evento

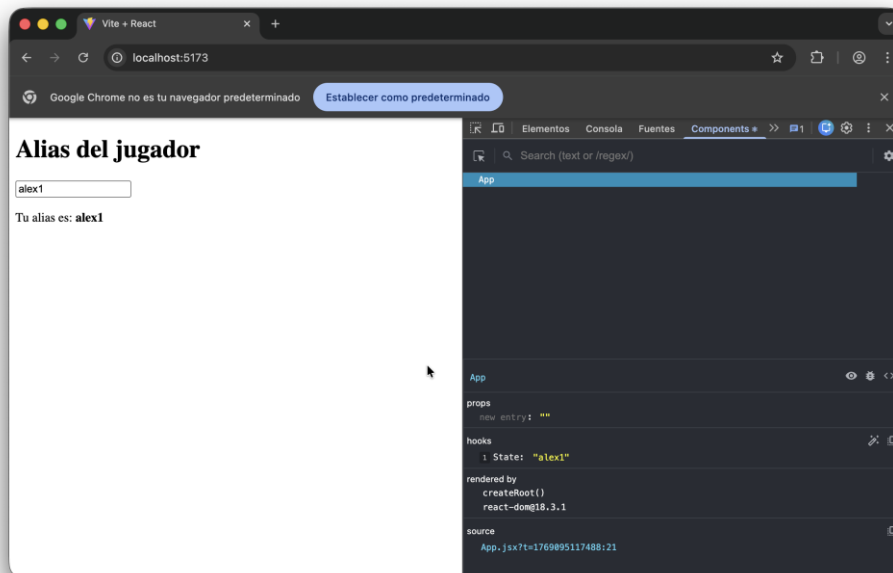
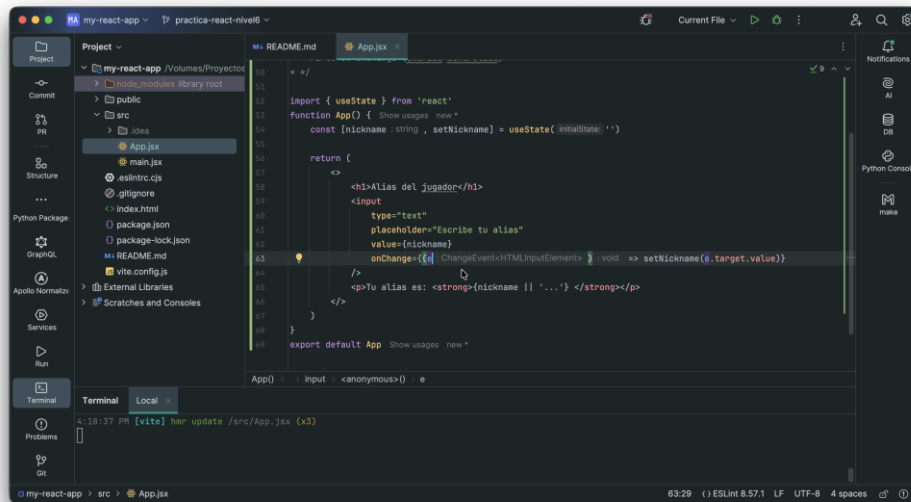
- La función que tiene el botón funciona en base al estado, de inicio empieza en true, con el texto en pantalla, al pulsar el botón cambia, ya que la función para a ser false



```
import { useState } from 'react'
function App() {
  const [showImages, setshowImages] = useState(true)
  const [isParagraphVisible, setisParagraphVisible] = useState(true)
  const toggleStatus = () => {
    setisParagraphVisible(!isParagraphVisible)
  }
  return (
    <div>
      <h1>Ocultar texto</h1>
      <p>{isParagraphVisible ? 'Alex1, texto a ocultar' : ''}</p>
      <button onClick={toggleStatus}>
        {isParagraphVisible ? 'Hide' : 'Show'} Texto
      </button>
    </div>
  )
}
export default App
```



### 3- Parte C: onChange (entrada controlada)



#### 4- Parte D: onSubmit (formulario)

- He agregado una nueva variable, de **msj** para que cuando mande el correo, también reciba el **msj** que se le ha escrito.

```

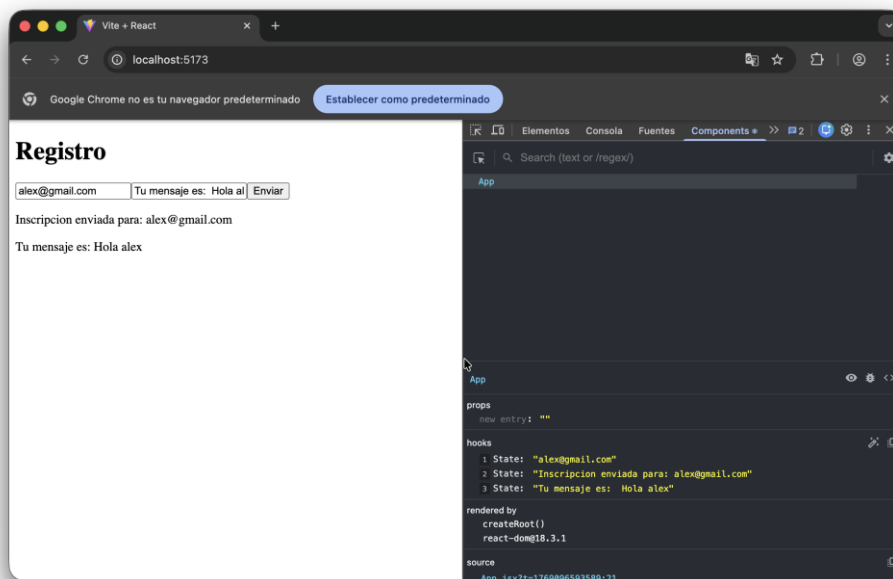
function App() {
  const [email, setEmail] = useState('')
  const [message, setMessage] = useState('')
  const [registered, setRegistered] = useState(false)

  const handleSubmit = (e) => {
    e.preventDefault()
    setRegistered(true)
    setEmail('')
    setMessage('')
  }

  return (
    <div>
      <h1>Registro</h1>
      <form onSubmit={handleSubmit}>
        <input
          type="email"
          placeholder="tu@gmail.com"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
        <input
          type="text"
          placeholder="Escribe tu msg"
          value={message}
          onChange={(e) => setMessage(e.target.value)}
        />
        <button type="submit">Enviar</button>
      </form>
      <div>
        <p>Inscripcion enviada para: {email}</p>
        <p>Tu mensaje es: {message}</p>
      </div>
    </div>
  )
}

```

## - Resultado



## 5- Mini reto

- He usado tres variables **hideText**, **contador** y **modo**
- **HideText**: Este es simple, método que variable en función al botón de true o false muestra el texto oculto.

- **Contador:** En el botón, he añadido llamado a `setContador` y que vaya suman de 1 en 1 para que sea incrementable.
- **Modo:** He creado el método `msj`, para para luego llamarlo en el input y he llamado la variable al final del Form `{modo} && <p>{modo}</p>`
- Para que cuando escribas en el input, se vea el texto cambiando.

```

function App() {
  const [hideText, setHideText] = useState(false);
  const [contador, setContador] = useState(0);
  const [modo, setModo] = useState('');

  const toggleStatus = () => setHideText(!hideText);

  const msj = () => {
    if (hideText) {
      return 'Soy el texto oculto';
    } else {
      return 'Show';
    }
  };

  const resetContador = () => setContador(0);

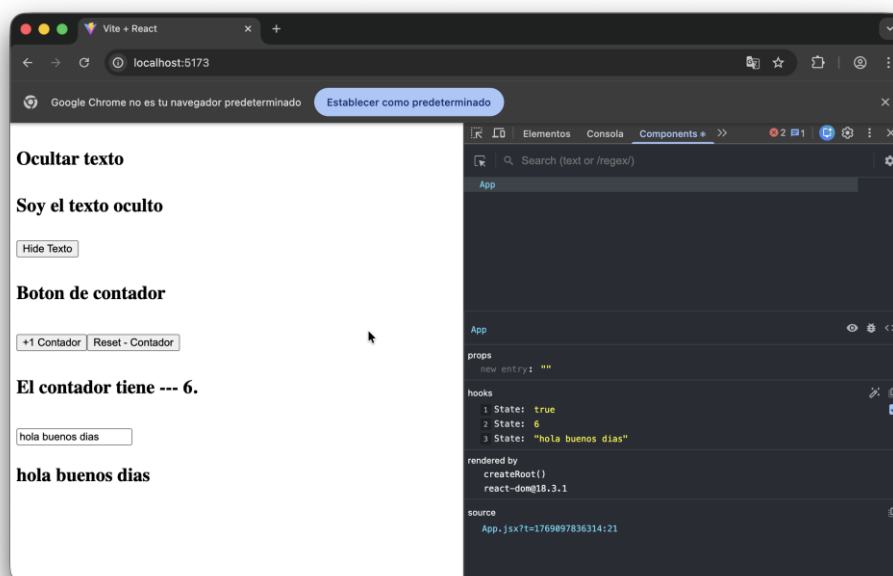
  const incrementContador = () => setContador((n) => n + 1);

  const onChangeModo = (e) => setModo(e.target.value);

  return (
    <div>
      <div>
        <h2>Ocultar texto</h2>
        <p>{msj()}</p>
        <button onClick={toggleStatus}>{hideText ? 'Hide' : 'Show'} Texto</button>
      </div>
      <div>
        <h2>Boton de contador</h2>
        <button onClick={incrementContador}>Contador</button>
        <button onClick={resetContador}>Reset - Contador</button>
      </div>
      <div>
        <p>{contador} > 0 && {contador} < 6 ? 'El contador tiene --- {contador}' : 'El contador tiene --- {contador}'</p>
      </div>
      <div>
        <h2>Formulario</h2>
        <form onSubmit={msj}>
          <input
            type="text"
            placeholder="modo"
            value={modo}
            onChange={onChangeModo}
            required
          />
          <button type="submit">Enviar</button>
        </form>
        <p>{modo} && {modo}</p>
      </div>
    </div>
  );
}

```

- Resultado



PREGUNTAS EXTRA:

1. ¿Qué diferencia hay entre onClick y onSubmit?

- onClick: Al ejecutarlo llama a un elemento ya sea un div, o alguna acción
- onSubmit: Se utiliza cuando se envía un formulario, este es recomendable para formularios.

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

- Por defecto, a la hora de hacer una acción de mandar un formulario la página se recarga, con esto se evita que pase eso

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

- Por ejemplo, un input, ya que es controlada por React mediante su estado, esto hace que nosotros tengamos el control.

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

- Los estados que utilizo son, hideText, Contador y modo
- Los eventos:
- OnClick para mostrar u ocultar el texto.
- OnChange actualiza el modo mientras escribes.