

## EL ESTADO Y LA MEMORIA DEL COMPONENTE

Los componentes necesitan cambiar lo que se muestra en pantalla como resultado de una interacción con el usuario, como por ejemplo hacer clic en siguiente página, agregar un ítem al carrito de compras entre otros, y básicamente estos componentes como el carrito de compras debe recordar cuántos productos hay en el carrito por ejemplo, esta memoria en react se lo conoce como estado.

```
function App() {
  let index = 0;

  function handleClick () {
    index = index + 1
  }

  return (
    <div>
      {index}
      <button onClick={handleClick}>+</button>
    </div>
  );
}

export default App;
```

Aquí tenemos un ejemplo donde usamos una variable para actualizar el estado pero vemos que esta no funciona y aquí es donde entra el hook `useState()` que es básicamente como declarar una variable en react, actualizamos este ejemplo para que vamos como se haría en React.

```
import { useState } from "react";

function App() {

  const [index, setIndex] = useState(0)

  function handleClick () {
    setIndex(index + 1)
  }

  return (
    <div>
      {index}
      <button onClick={handleClick}>+</button>
    </div>
  );
}
```

```

    </div>
  );
}

export default App;

```

El useState se usa cuando queremos que react recuerde algo. en este caso queremos que recuerde el index.

Cada vez que un componente se renderiza el useState devuelve un array que contiene 2 valores:

- 1- La variable de estado (index) con el valor que almacenamos
- 2- La función que establece el estado (setIndex) que puede actualizar el estado de la a verbal ey a a letras a react para que renderize el componente nuevamente

Básicamente funciona de la siguiente manera:

- 1- Definimos el estado inicial que es 0
- 2- Actualizamos el estado al hacer clic, esto llama a la función setIndex + 1 por lo tanto ahora react recuerda que index es 1 y así sucesivamente.

## COLOCAR MÚLTIPLES VARIABLES DE ESTADO A UN COMPONENTE

```

import { useState } from "react";

function App() {

  const [index, setIndex] = useState(0)
  const [text, setText] = useState(false)

  function handleShow () {
    setText(!text)
  }

  function handleClick () {
    setIndex(index + 1)
  }

  return (
    <div>
      {index}
      <button onClick={handleClick}>+</button>
      <button onClick={handleShow}>{text ? 'Ocultar' :
'Mostrar'}</button>
    </div>
  )
}

```

```

        {text && <p>Hola soy un texto random</p>}
      </div>
    );
  }

export default App;

```

## EL ESTADO ES AISLADO Y PRIVADO

El estado es local para una instancia de un componente en la pantalla m, en otras palabras si renderizamos el mismo componente dos veces cada copia tendrá un estado completamente aislado, cambiar uno de ellos no cambiará al otro:

```

import { useState } from "react";

function Test() {

  const [index, setIndex] = useState(0)
  const [text, setText] = useState(false)

  function handleShow () {
    setText(!text)
  }

  function handleClick () {
    setIndex(index + 1)
  }

  return (
    <div>
      {index}
      <button onClick={handleClick}>+</button>
      <button onClick={handleShow}>{text ? 'Ocultar' :
'Mostrar'}</button>
      {text && <p>Hola soy un texto random</p>}
    </div>
  );
}

export default Test;

```

```

import Test from "../Test";

function App() {

```

```
return (  
  <div>  
    <Test/>  
    <Test/>  
  </div>  
);  
}  
  
export default App;
```