

# REACT

**Componentes (Components):** Pieza encapsulada de la Interfaz del Usuario que es reutilizable y contiene su propia lógica. En react se crean usando funciones o clases y las funciones en React se declaran con la nomenclatura PascalCase.

```
function MiComponente() {  
  return <h1>Hola, mundo!</h1>;  
}
```

**Anidación de componentes (Nest Components):** Insertar componentes dentro de otros usando keywords como [export](#), [export default](#) e [import](#).

Declaramos la función

```
// En el archivo MiComponente.js  
  
import React from 'react';  
  
function MiComponente() {  
  return <h1>Hola, mundo!</h1>;  
}  
  
export default MiComponente;
```

La anidamos a otro componente

```
// En el archivo App.js  
  
import React from 'react';  
import MiComponente from './MiComponente';  
  
function App() {  
  return (  
    <div>  
      <MiComponente />  
    </div>  
  );  
}  
  
export default App;
```

**JSX (JavaScript + XML):** Sintaxis usada junto con React, da mayor legibilidad al código. [Ejemplo](#).

Fragment (<Fragment>, </Fragment>): Todo lo que devuelva una función de react debe estar dentro de una etiqueta padre, React no permite devolver varias etiquetas o elementos sin tener otro por encima.

La [versión abreviada](#): (<>, </>).

[Diferencias entre Fragment y Div.](#)

## Hooks

Hooks: Características de React que puedes importar en tus componentes.

[Listo](#) de Hooks

State Hook: Característica de React que permite “recordar” información (estados) y cambiarla según el resultado de una interacción.



The image shows a side-by-side comparison of code and its rendered output. On the left, a code editor displays the following JavaScript code:

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

export default function MyApp() {
  return (
    <div>
      <h1>Counters that update separately</h1>
      <MyButton />
      <MyButton />
    </div>
  );
}

function MyButton() {
  const [count, setCount] = useState(0);

  function handleClick() {
    setCount(count + 1);
  }

  return (
    <button onClick={handleClick}>
      Clicked {count} times
    </button>
  );
}
```

On the right, a web browser displays the rendered application. The title is "Counters that update separately". Below the title, there are two buttons. The first button displays "Clicked 5 times" and the second button displays "Clicked 2 times".