

# Teoría React I

## ¿Qué es el ciclo de vida de un componente?

En React, cada componente pasa por una serie de "etapas" durante su existencia en la aplicación, comúnmente conocidas como el "ciclo de vida" del componente. Estas etapas incluyen la creación, actualización y eliminación del componente. Entender estas etapas es crucial para manejar correctamente cómo se comporta el componente en diferentes momentos.

En versiones recientes de React, especialmente con la introducción de hooks como `useState` y `useEffect`, el manejo del ciclo de vida en componentes funcionales se ha simplificado, permitiendo una mayor flexibilidad y un código más limpio comparado con los componentes de clase. Los hooks `useEffect` y `useState` cubren la mayoría de las necesidades del ciclo de vida en componentes funcionales.

## Repasemos: ¿Qué son los estados de React?

Un "estado" en React es una forma de mantener datos que pueden cambiar durante la vida útil de un componente. Por ejemplo, el texto en un campo de formulario, si un menú está abierto o cerrado, o los artículos en un carrito de compras, son ejemplos de datos que podrían manejarse a través del estado.

React nos ofrece una función para poder manejar estos cambios de estados. La función `useState()` ofrece:

- Una variable de estado para mantener los datos entre renderizados.
- Una función que actualiza el estado para actualizar la variable y provocar que React renderice el componente nuevamente.
- Configurar el valor inicial del estado.

Unset

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

import DisplayMenu from './...';

export default function Menu() {
```

```
const [show, setShow] = useState(false);

const showMenu = () => setShow(!show);

return (<>

  <button onClick={showMenu}>Menu</button>

  {show && <DisplayMenu />}

</>);
}
```

En este ejemplo, la función devuelve un array con el estado y la función seteadora del estado y se inicializa con un booleano. El estado se inicia en false, por lo que no se mostrará el componente desplegable del menú DisplayMenu. Cada click, cambiará el estado al booleano opuesto, permitiendo mostrar/ocultar el desplegable.

## Recordemos: ¿Qué son los Hooks?

Los Hooks son funciones que empieza con "use" y que sólo están disponibles mientras React está renderizando. Los Hooks permiten «engancharnos» a diferentes características de los componentes de React.

El hook `useState()` es una función que se engancha al estado del componente para modificarlo, pero el estado es solo una de esas características. Existen muchos más Hooks .



Te invitamos a investigar a tu ritmo sobre los distintos hooks existentes, sus usos y buenas prácticas en la documentación oficial :👉 [React - Hooks](#)