

# Hooks

React estado + ciclo de vida



# ¿ Qué son los React Hooks?

React Hooks son **funciones** que nos permiten añadir estado y ciclo de vida a componentes funcionales.

*"Dotan de super poderes a un componente funcional"*

STATE

LIFECYCLE LIKE

*"Los components Funcionales pasan a ser ciudadanos de primera categoría"*

Fuente: <https://hackernoon.com/react-hooks-usestate-using-the-state-hook-89ec55b84f8c>

# Basic hooks

## useState

**useState** es un hook que te permite añadir estado a tus components funcionales.

```
const [state, setState] = useState(initialState);
```

# Basic hooks

## useEffect

**UseEffect** es un hook que te permite encapsular Código que tiene 'side effects'. Te permite ejecutar cierto código cuando un componente se crea, cuando se actualiza, cuando una variable ha cambiado, o cuando se destruye.

```
import React, { useState, useEffect } from "react";

const App = () => {
  const [name, setName] = useState("Ceci");

  useEffect(() => {
    document.title = name;
  }, [name]);

  return (
    <input
      value={name}
      onChange={event => setName(event.target.value)}
    />
  );
}
```

# Reglas básicas usando Hooks



Invoca a los hooks desde el nivel superior de tu función

**No los invoques dentro de bucles, condiciones, o funciones anidadas.**

Es importante que los hooks se llamen en el mismo orden en cada render



Los hooks sólo pueden ser invocados desde funciones React

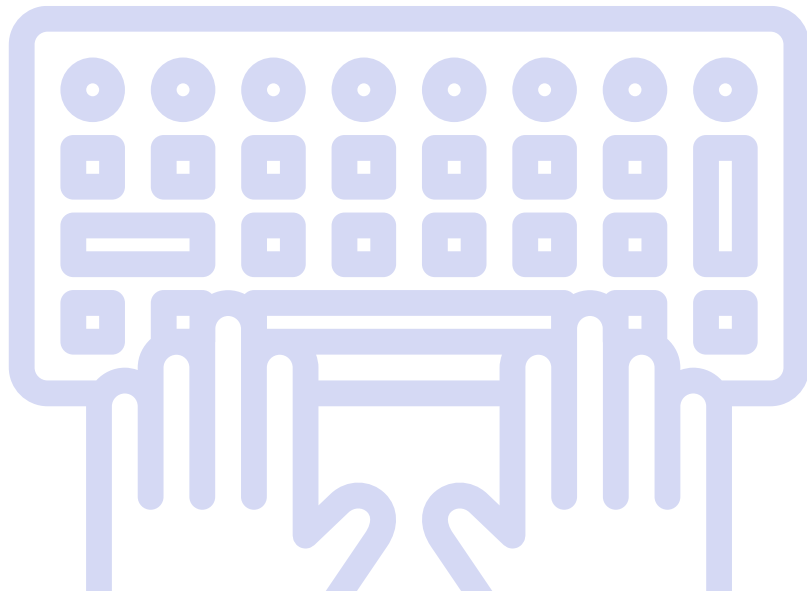
**No llames a un hook desde una función Javascript plana.** Sólo puedes:

- ✓ Invocar a un Hook desde un componente funcional de React
- ✓ Invocar a un Hooks desde otro Hook

# ¿Cómo aprender?


La mejor forma de aprender Hooks es a base de ejemplos

## Demo Time



# ¡ Muchas gracias !



 @lemoncoders



<https://github.com/lemoncode>



 @basefactorteam