

ReactJS Developer

Módulo 5

Hooks personalizados

Hooks personalizados

Construir tus propios *Hooks* te permitirá **extraer la lógica del componente en funciones reutilizables**. Cuando queremos compartir lógica entre dos funciones de JavaScript, lo extraemos en una tercera función. Ambos componentes y *Hooks* son funciones, ¡así que esto funciona para ellos también!

Un *Hook* personalizado es una función de JavaScript cuyo nombre comienza con **use** y que puede llamar a otros *Hooks*.

A diferencia de un componente de React, un *Hook* personalizado no necesita tener una firma específica. Podemos decidir lo que adopta como argumentos y lo que, si lo hace, debería devolver. En otras palabras, es como una función normal.



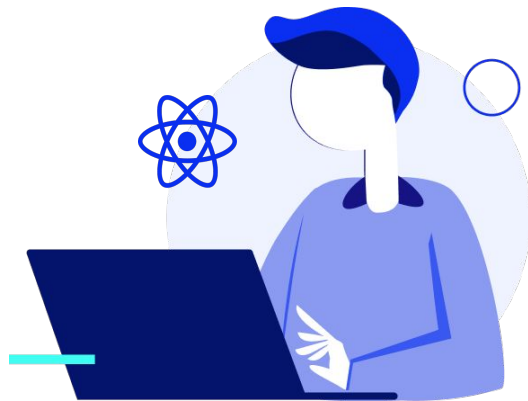
Creando el *Hook* personalizado

- Un *Hook* es, simplemente, una función que lleva el prefijo *use* y usa otros *hooks* por defecto de React.
- Recuerda usar los *hooks* en el **nivel superior** de la función.
- Puedes usar el *hook* de estado **useState**. Este *hook* puede ser un buen punto de comunicación entre tu *hook* personalizado y el componente que lo consuma.
- Recuerda que puedes usar **cualquier otro *hook* y cualquier recurso del lenguaje.**

```
1  const { useState, useEffect } = require("react");
2
3  function useApi(resource)
4  {
5      const [data, setData] = useState({});
6
7      useEffect(() => {
8          fetch(resource)
9              .then(r => r.json())
10             .then(d => setData(d))
11         });
12
13     return data;
14 }
15
16 export default useApi;
```

Usando el *Hook* personalizado

- Para usar el *hook* simplemente debemos **importarlo**.
- Una vez importado se usa normalmente **como cualquier otro *hook***.
- Recuerda que, como otros *hooks*, **debe ser usado en el nivel superior del componente**.



```
1  import useApi from "../useApi";
2
3  function PostsList(props) {
4      const posts = useApi("/api/posts");
5
6      return posts.map((p, i) => <div key={i}>{p.name}</div>);
7  }
8
9  export default PostsList;
10
```

**¡Sigamos
trabajando!**