

useEffect

Resumen

- Introduciremos `useEffect`
- Lo usaremos para cambiar el título de la página principal

Descripción

Vamos a suponer que queremos cambiar el *título de la pestaña* del navegador.



Sabemos que se hace con:

```
document.title = "Título de ejemplo";
```

Podríamos estar tentados de poner ese código en cualquier lugar del componente React:

```
const PaginaDeEjemplo = () => {

  // ¡MAL!
  document.title = "Título de ejemplo";

  return <div>Lorem ipsum</div>
}
```

Esto estaría... ¡mal!

NO debemos poner código que produce *efectos colaterales*¹ alegremente en cualquier sitio del componente.

¿Por qué?

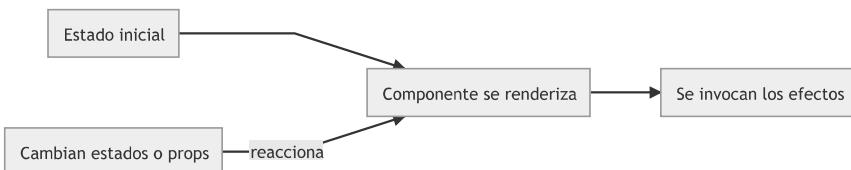
Porque un componente es una *función* que React va a invocar indiscriminadamente *cada vez* que necesite repintarlo.

¡No controlamos cuándo o cuántas veces se va a invocar!

Entonces, ¿qué hago?

Para los *efectos colaterales*, existe en React [`useEffect`](#).

Ahí, podemos poner código que se invoque *después* de que el componente se *renderice* (sea invocado):



La tarea

Añade el siguiente `useEffect` al principio de `Main.js`:

```
// Nuevo import
import { useEffect } from "react"

const Main = () => {
  useEffect(() => {
    });
}
```

// ...

Como ves, a `useEffect` le pasamos una función. En este caso, especificada con la sintaxis *arrow* (`() => {}`).

Ahora, en esa función, **añade** código que cambia el título. Así:

```
const Main = () => {
  useEffect(() => {
    document.title = 'Main'
  });
}

// ...
```

¿Ves el nuevo título en la pestaña del navegador en <http://localhost:3000/>?

🏆 Por último

Sube tus cambios al repositorio en un nuevo *commit*.

1. Cualquier código que afecte a la web más allá del *contexto* del componente puede catalogarse de *efecto colateral*. Por ejemplo: Mandar una petición a otra página web, establecer una conexión (e.g.: chat, vídeo en streaming), invocar ciertas APIs del navegador... 



Rubén Montero @ruben.montero changed milestone to %Sprint 3 3 hours ago