

¿Cuándo se ejecuta?



Resumen

- Entenderemos cuándo se ejecuta `useEffect`
- Comprenderemos que se puede pasar un segundo parámetro (un `array []`) a `useEffect` para controlar cuándo se ejecuta
- Añadiremos un `useEffect` a `SimpleCounter.js`

Descripción

Vamos a ver que podemos poner barreras al *efecto*, y decirle que sólo debe ejecutarse *cuando cambien ciertas propiedades reactivas* (estados o *props*).

Para ello, `useEffect` admite un segundo parámetro de tipo `array`:

```
useEffect(() => {  
  
}, []) // <-- segundo parámetro []
```

Este `array` contendrá aquellos estados o *props* cuyos cambios producirán la ejecución del efecto.

Por el contrario, si un estado o *prop* no está recogido en este `array`, entonces, aunque cambie, el efecto no se invocará.



La tarea

Modifica `SimpleCounter.js` para que contenga un efecto que cambie el título de la página:

```
-import { useState } from "react"  
+import { useState, useEffect } from "react"  
  
const SimpleCounter = () => {  
  const [number, setNumber] = useState(0);  
  
  + useEffect(()=>{  
  +   document.title = "Número: " + number;  
  + }, [])  
  
  return <div>  
    <p>Número: {number}</p>  
    <button data-cy='issue12button' onClick={ () => {setNumber(number+1) }}>Incrementar</button>  
  </div>  
}
```

Como ves, el título de la página ahora reflejará el valor de `number`.

¡Pero hemos añadido un `array` vacío (`[]`) como segundo parámetro! Por lo tanto, el efecto sólo sucederá una vez, inicialmente.

Visita <http://localhost:3000/examples>.

Cuando clicas en el contador, no debe cambiar el título. ¿Es así?

En la siguiente tarea haremos una pequeña modificación 😊



Por último

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



