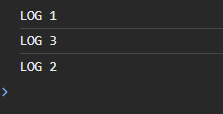
Efectos 🡪 nos ayudan encapsular la parte pesada de la lógica de nuestros componentes y podamos protegerlos para que únicamente se ejecuten cuando se requiera.

Se usa el hook useEffect



A computer screen with white and blue text

Description automatically generated

Se renderizo de ultimas el LOG 2

A computer screen shot of a number

Description automatically generated Se puede agregar un segundo parámetro en el useEffect, como [] esto causa que el Log 2 se cargue solo una primera vez, pero si después se porvoca un render en algún componente ya no se va a renderizar el log 2

A screenshot of a computer

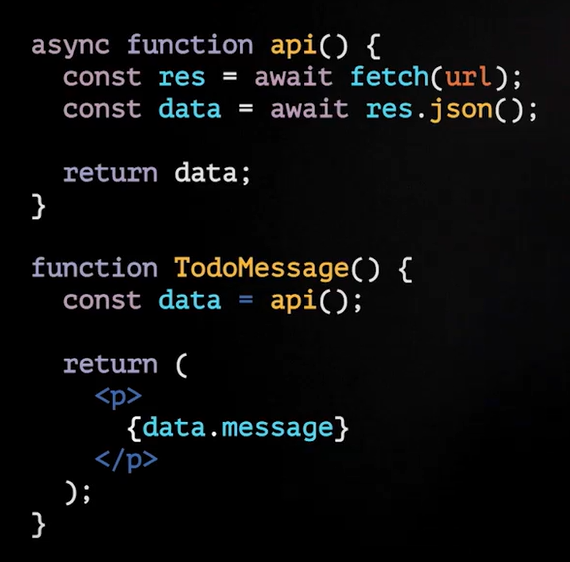
Description automatically generated

A white rectangular object with black text

Description automatically generatedA computer screen shot of a program code

Description automatically generated se puede ingresar al array, un estado el cual si cambia, lo de adentro de effect se va a ejecutar

Si se guarda algo en un efecto se va a ejecutar después, y segundo que si se envia un segundo argumento al efecto como un array vacio, el contenido de va a ejecutar una vez, y si se le envía un estado particular ese contenido interno se va a ejecutar todas las veces que se ejecutan ese estado



Es un Await por lo tanto ese api no va a entregar información de forma inmediata, lo cual puede tardar un rato en contestar, ¿qué va a pasar entonces? El usuario va a ver un loader, una página en blanco, y que pasa si algún componente causa un render en el componente, ¿vamos a esperar de nuevo el api? No. Hay bloques pesados en la aplicación que toca agregarle estados y efectos.



Se cambia a esto, aquí se usa un estado inicializado con un array vacío, por lo que en la condición del elemento <p> no se va a cumplir y va a mostrar el mensaje de Cargando.

En el useEffect se le setea un valor al estado, además ese useEffect como recibe un segundo argumento de [] esta lógica solo se va a ejecutar una vez, con eso ya después esa api se olvide para los renders

 hay un guardian de efectos que no deja ejecutar los efectos hasta después de rendizar todo lo demás.

Cuando carga el api 

Ya el efecto se va a ejecutar pero por el [], solo una vez.

A person in a blue shirt

Description automatically generatedPor ultimo el guardian de estados al detectar un cambio en el estado, obliga al elemento a rendizarse nuevamente

A person in a blue shirt

Description automatically generated

Y al ejecutarse de nuevo el guardian de los efectos se da cuenta que no debe ejecutar de nuevo ese efecto por el segundo parámetro

  React.useEffect(() => {

    console.log('LOOOOOOOOOOOOOG 2')

  });

//Solo se ejecuta una vez

  React.useEffect(() => {

    console.log('LOOOOOOOOOOOOOG 2')

  }, []);

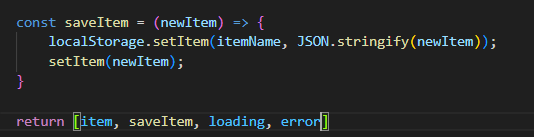
//Atado a un estado

  React.useEffect(() => {

    console.log('LOOOOOOOOOOOOOG 2')

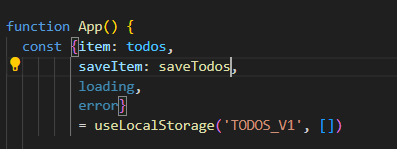
  }, [totalTodos]);

Si hay mas de dos elementos que retonar en un custom hook, es mejor retornar un objeto que un array

 Esto NO

A screen shot of a computer program

Description automatically generatedEsto SI

 para pasarla ya no se puede usar el nombre que uno le asigna en array sabiendo la posición del elemento, lo que puede ocasionar conflictos de nombres, para ello se pueda renombrar el atributo del objeto con dos puntos



Este siginfica que al actualizar el estado por el efecto este estado de nuevo se renderiza y nuevamente trata de ejecutar el efecto, en si se vuelve un bucle…