**Custom hooks**

Los customs hooks pueden utilizarse para “limpiar” las declaraciones e importaciones de un componente y “delegar” o “modularizar” las mismas.

Por ejemplo, podríamos considerar un componente Custom.jsx en el cual contengamos un contador (state) y algunos botones que modifiquen el mismo. Una forma de customizar un hook es de la siguiente manera:

Este es el código para el componente:

import { useCustom } from "../hooks/useCustom";

const Custom = () => {

  const [contador, increment, decrement] = useCustom(-64, 8);

  return (

    <div>

      <p>Este es el custom hook : {contador}</p>

      <button *onClick*={decrement}>-</button>

      <button *onClick*={increment}>+</button>

    </div>

  );

};

export default Custom;

notese que se desestructura el retorn de useCustom para utilizar **contador, increment, decrement.**

y en un nuevo archivo useCustom.js tendríamos:

import { useState } from "react";

export const useCustom = (*initialValue*, *range* = 1) => {

    const [contador, setContador] = useState(*initialValue*);

    const increment = () => {

        setContador(contador + *range*);

    };

    const decrement = () => {

        setContador(contador - *range*);

    };

    return [contador, increment, decrement]

}

En este caso, se personaliza useCustom para que reciba el valor inicial y el “rango” de salto del contador. De dicha forma es más reutilizable el hook.

Esto mismo puede realizarse con cualquier componente para “Limpiar” el código y solo renderizar las etiquetas necesarias.