**React**

**Hook - useState.**

Documentación: <https://es.reactjs.org/>

<https://es.reactjs.org/docs/getting-started.html>

<https://es.reactjs.org/docs/hooks-state.html>

Al utilizar el Hook useState este devuelve un valor con estado y una función para actualizarlo y se realiza de la siguiente manera. Ejemplo:

import { useState } from "react"; // Importamos el Hook - useState

import PropTypes from "prop-types"

const Counter = ({ value }) => {

    // Desestructuramos el Hook useState

    /\*\*

     \* Notas del Hook useState

     \* 1.- Cuando trabajamos con el useState necesitamos mandar cual va a ser

     \*     ese valor inicial de ese estado: ( value ).

     \*

     \* 2.- Al realizar la desestructuración del retorno de la función useState,

     \*     - El primer valor "counter" durante la representación inicial del estado devuelto ()

     \*       es el mismo que el valor pasado como primer argumento ()

     \*     - El segundo valor "setCounter" será una función que usaremos para actualizar

     \*       el valor y estado de "counter".

     \*

     \* 3.- Cuando mandamos a llamar setCounter, le decimos a React, oye el estado cambio.

     \*     Por consecuencia tienes que volver a ejecutar la renderización del componente.

     \*/

    const [ counter , setCounter ] = useState( value )

    const Increment = () => {

        // Forma 1 de incrementar el valor

        setCounter( counter + value )

        // Forma 2 de incrementar el valor, utilizar un nombre de una variable internamente...

        setCounter( (c) => c + value )

    }

    return (

        <>

            <h1>App Counter: { counter }</h1>

            <button onClick={ Increment } >¡Click {value}++!</button>

        </>

    );

}