Escribiendo el componente contador

En esta sección vamos a continuar con el desarrollo de la aplicación de tipo contador: **counterApp**, el objetivo es definir un botón que tenga como finalidad incrementar el contador en 1 cada vez que se le da un clic.

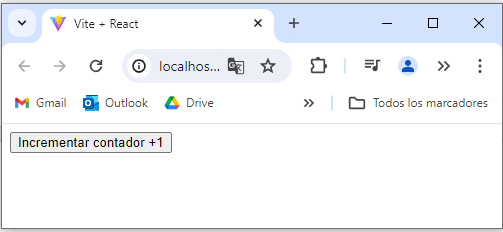
Continuando el proyecto

Sigue estos pasos:

Paso 4: Crear un botón

1. En el componente **CounterApp.jsx**, agrega un botón con las etiquetas **<button></button>**, dentro de las etiquetas se define el texto que tendrá el botón.

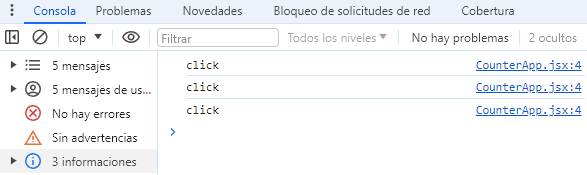
export const CounterApp = () => {
    return <>
        <button>Incrementar contador +1</button>
    </>
}

1. Para hacer que el botón haga algo cuando se hace clic, se va a requerir manejar un evento. En React, al igual que en JavaScript, se maneja eventos. La sintaxis para manejar eventos en React es similar a JavaScript, pero optimizada para React.
2. Se utiliza el evento **onClick** con una función de flecha para ejecutar alguna acción. Por ejemplo, imprimir **"click"** en la consola cada vez que se presiona el botón.

export const CounterApp = () => {
    return <>
        <button onClick={() => {
            console.log("click");
        }}>Incrementar contador +1</button>
    </>
}


1. Pulsa el botón varias veces y verifica en la consola del navegador que se imprima el mensaje **"click"** cada vez que se presiona el botón.



Paso 5: Separar las funciones

1. Primero define una función llamada **counterIncrement** que se encargará de realizar alguna acción cuando se incremente el contador. Por ejemplo, en este caso, la función simplemente imprime **"click"** en la consola (corta y pega la función definida en el evento **onClick**).

export const CounterApp = () => {

    const counterIncrement = () => {
        console.log("click");
    }

    return <>
        <button onClick={ }>Incrementar contador +1</button>
    </>
}

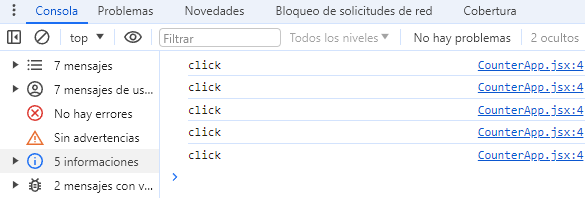

1. Luego asocia la función **counterIncrement** al evento **onClick** del botón, ten en cuenta que se invoca a la función dentro del cuerpo de la función de flecha del evento. Cada vez que se haga clic en el botón, se invocará esta función.

export const CounterApp = () => {

    const counterIncrement = () => {
        console.log("click");
    }

    return <>
        <button onClick={() => {
            counterIncrement()
        }}>Incrementar contador +1</button>
    </>

}

1. Al principio de la función principal del componente CounterApp, define una variable **counter** que representará el valor del contador. Por ahora, se inicializa en **0**. Luego, dentro del **return**, se va a mostrar el valor del contador dentro de un elemento **<h2>**, el cual va a contener el valor de la variable **counter** entre llaves.

export const CounterApp = () => {

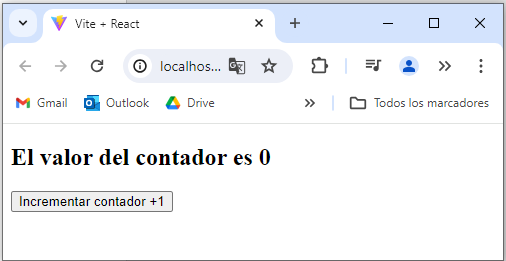
    const counter = 0;

    const counterIncrement = () => {
        console.log("click");
    }

    return <>
        <h2>El valor del contador es {counter}</h2>
        <button onClick={() => {
            counterIncrement()
        }}>Incrementar contador +1</button>
    </>

}


1. Al cargar la página en el navegador, debe mostrar un título con el valor **"El valor del contador es 0".** Cada vez que se haga clic en el botón, se invocará la función **counterIncrement** y se imprimirá **"click"** en la consola.



Paso 6: Implementar el contador en la consola

1. Cambia la declaración de **counter** de **const** a **let**, para que el valor del contador pueda cambiar. También se define la operación para incrementar el valor de la variable **counter** en 1, esto se define dentro de la función **counterIncrement**.

export const CounterApp = () => {

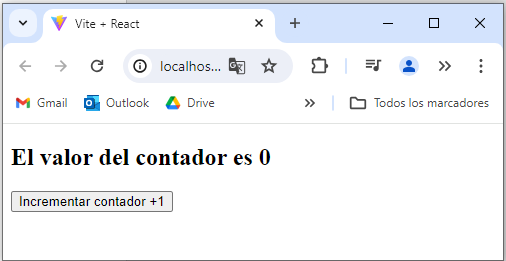
    let counter = 0;

    const counterIncrement = () => {
        counter = counter + 1
        console.log("click");
    }

    return <>
        <h2>El valor del contador es {counter}</h2>
        <button onClick={() => {
            counterIncrement()
        }}>Incrementar contador +1</button>
    </>

}

1. En el navegador, se puede observar que el valor del contador aumenta en la consola cada vez que se hace clic en el botón, pero no se refleja en el título **<h2>**. Esto se debe a que el estado no se está actualizando ni renderizando cuando cambia la información del contador.



1. Para solucionar este problema y actualizar el estado del contador de manera adecuada, se va a utilizar el hook **useState**. Sin embargo, esto se verá en pasos posteriores. Por lo cual puedes comenzar a imprimir el valor de la variable **counter** en la consola.

export const CounterApp = () => {

    let counter = 0;

    const counterIncrement = () => {
        counter = counter + 1
        console.log("click " + counter);
    }

    return <>
        <h2>El valor del contador es {counter}</h2>
        <button onClick={() => {
            counterIncrement()
        }}>Incrementar contador +1</button>
    </>

}


1. Al hacer clic en el botón varias veces en el navegador, se imprimirá el mensaje **"click"** seguido del valor actualizado del contador en la consola, pero el valor **counter** definido en el título **<h2>** no se actualizará porque el estado no se está manejando correctamente.

