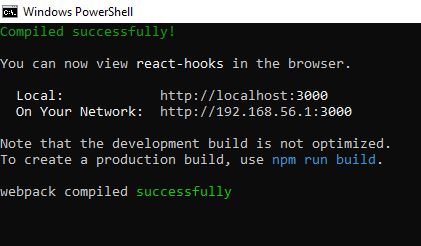
[Hooks video](https://www.youtube.com/watch?v=rgnFto7zVck&t=851s) 8’07’’

abrir cmd

comando para crear el proyecto: **npx create-react-app react-hooks**

**cd a carpeta proyecto**

**npm start**

****

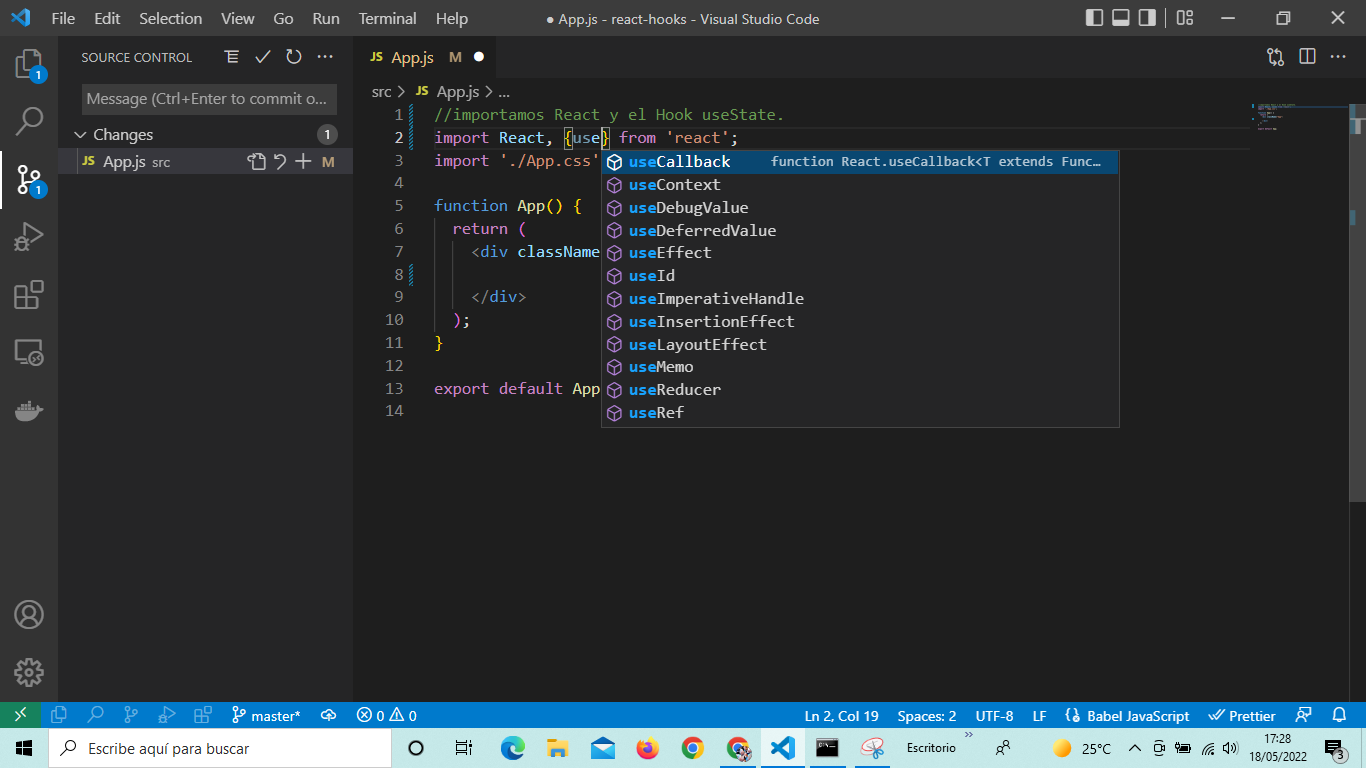
En este caso copiamos y pegamos: <http://localhost>: 3000 (Se suele abrir la página por defecto).

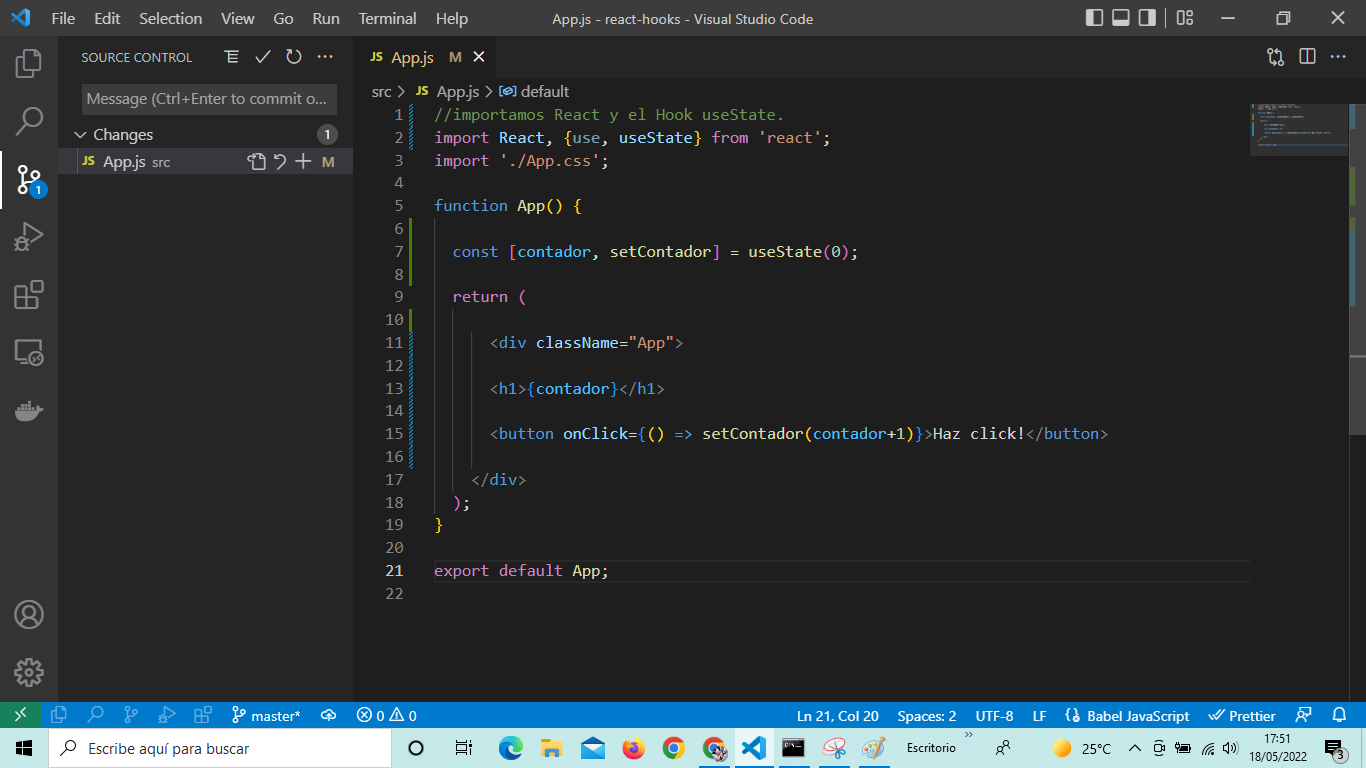
**Control +c** **(ctrl+c)** para detener el proceso.

**code .** sirve para abrir el proyecto en Visual Studio Code.

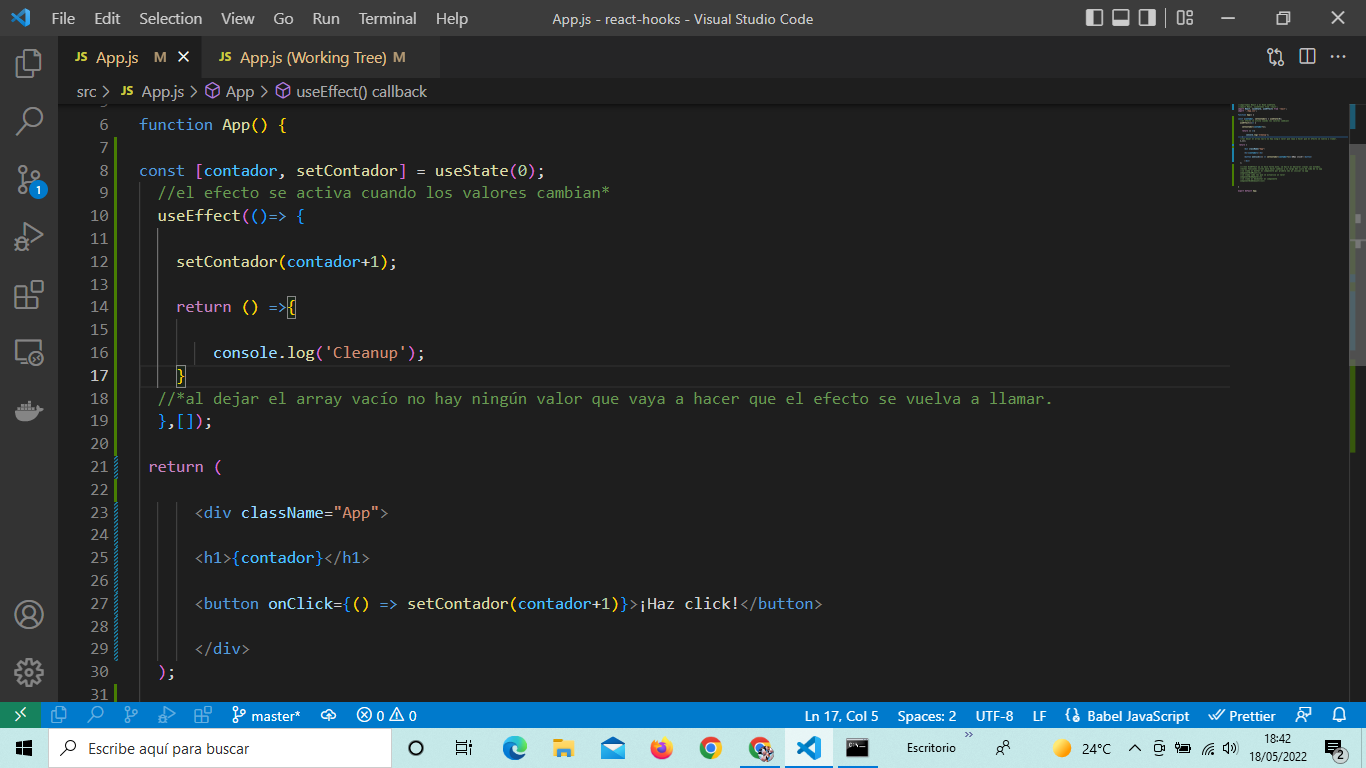
vamos a cmd y de nuevo escribimos: **npm start**

importamos react y el hook **useState: import React, {useState} from ‘react’;**

****



Uso básico de **useState**, recibe como parámetro el valor con el que inicializamos la variable useState{0} la función devolverá un vector con dos valores, [contador, setContador], el primer valor lo utilizaremos para actualizar el componente, el segundo es un setter: podemos llamarla cuando queramos actualizar el estado del componente. Si bajamos hasta la línea 15 vemos que setContador recibe un parámetro: contador+1, así cada vez que presionamos sobre el botón (button) actualizamos el componente y le pasamos el valor contador+1.



En la imagen falta la línea de importación:  **import React, {useState, useEffect} from ‘react’;** y después del </div> : la de exportación: **export default App;**

**useContext** const value = useContext(MyContext);

Permite acceder a los valores de un componente desde cualquier otro componente sin tener que acceder a los props. Primero creamos un contexto, después un hook que se ocupará de extraer los datos del contexto para después poderlos utilizar en cualquier componente.

[useContext](https://www.youtube.com/watch?v=mnKHJDkpZos)