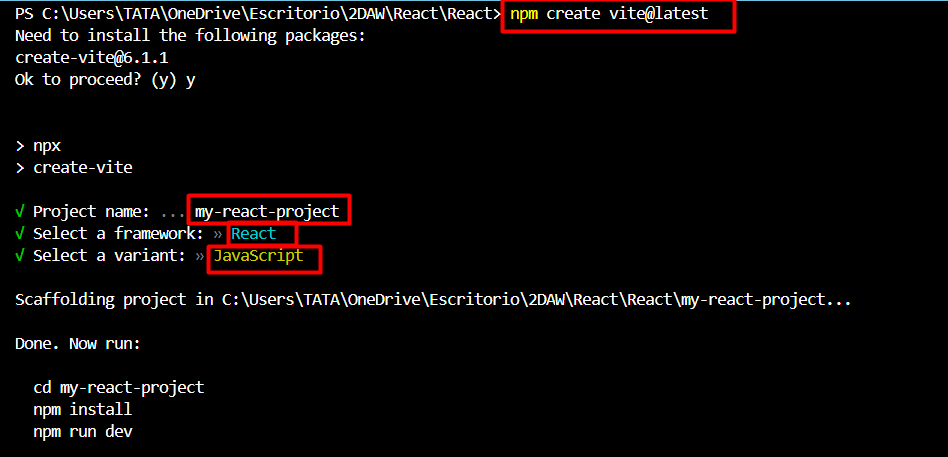
CREAR PROYECTO

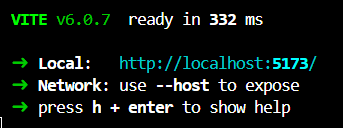


INSTALAR DEPENDENCIAS

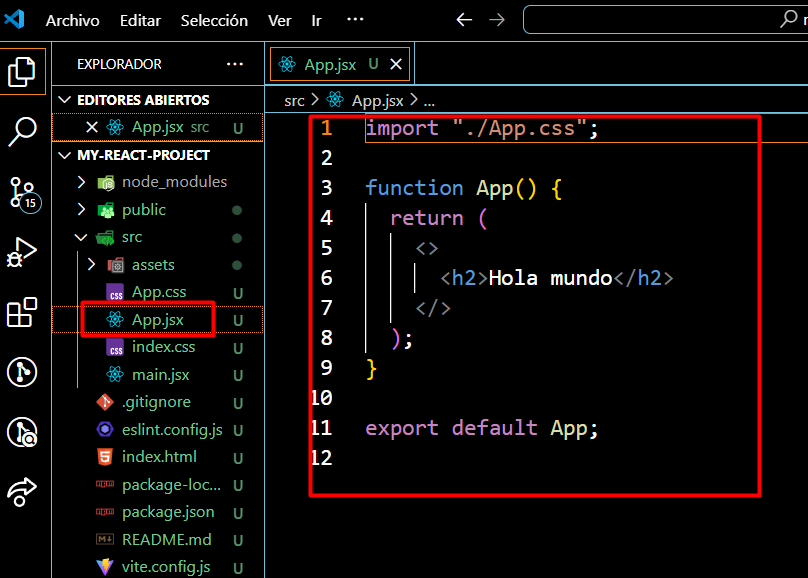
Npm install

EJECUTAR PROYECTO

NPM RUN DEV



VISTA BÁSICA



VACIAMOS TAMBIEN INDEX.CSS Y APP.CSS

VITE.CONFIG NO SE TOCA

README DOCUMENTACION

PACKAJE.JSON (DEPENDENCIAS)

NODE\_MODULES(LIBRERIAS)

PACKAJE-LOCK (DEPENDENCIAS)

INDEX (UNICA PÁGINA DE INICIO) NO SE TOCA

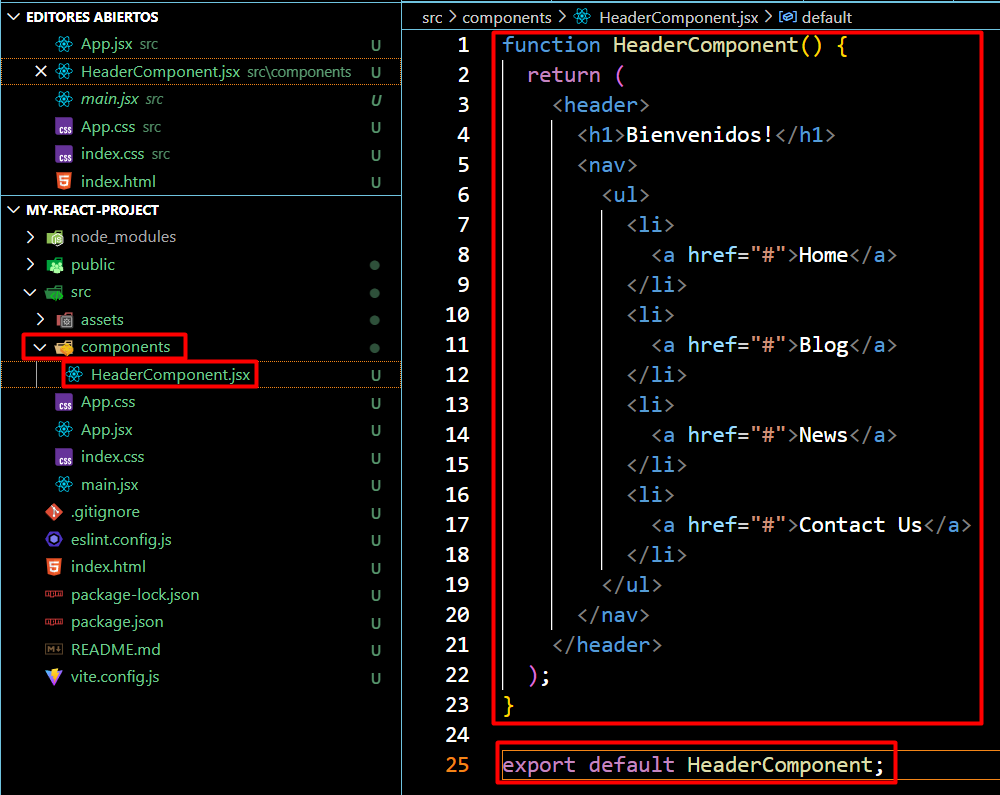
PUBLIC- ASSETS

MAIN.JSX (COMO SE RELACIONAN LOS COMPONENTES DE MI APP)

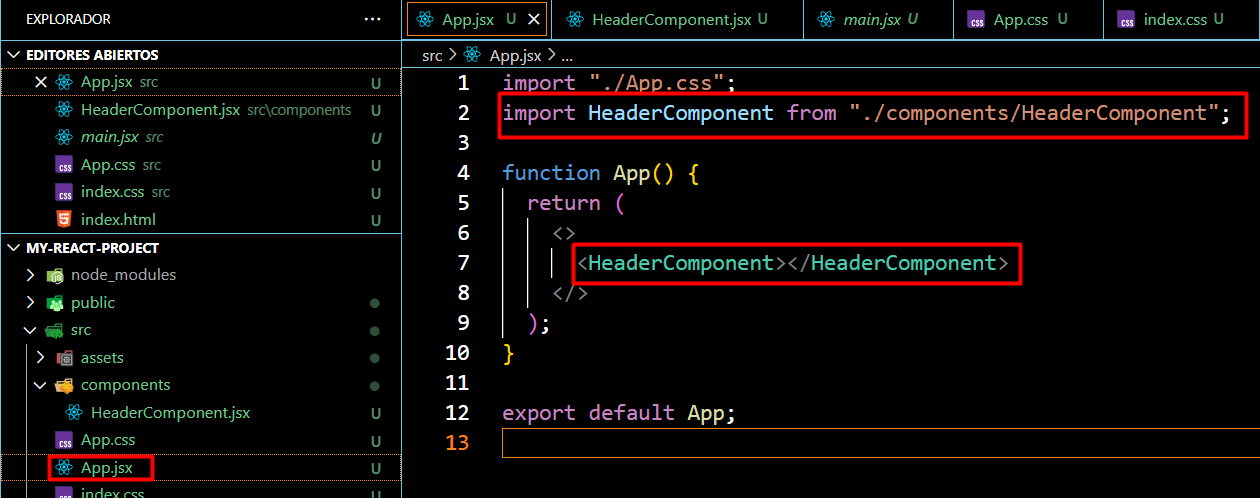
INDEX.CSS (FICHERO DE ESTILOS GLOBAL)

APP.JSX –APP.CSS

Crear componente

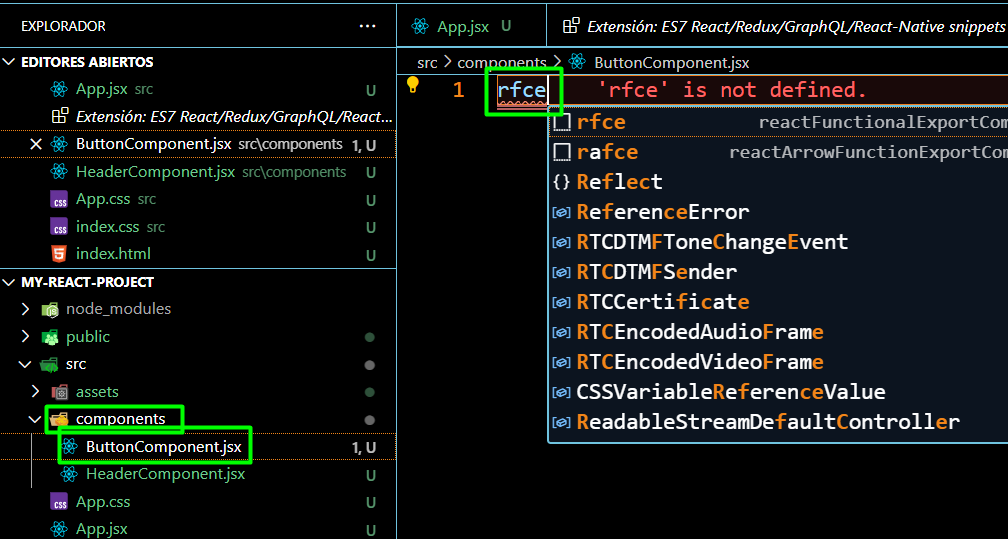


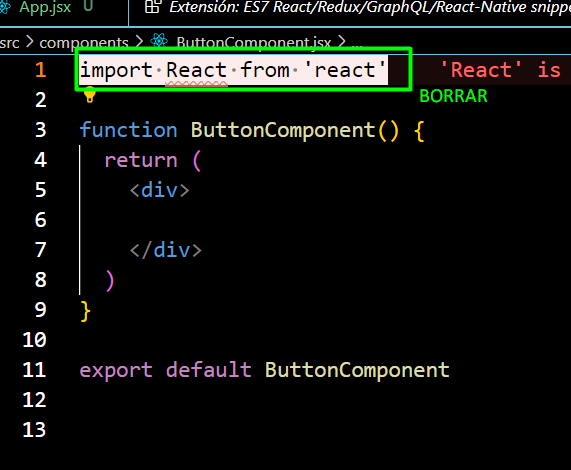
Usar componente



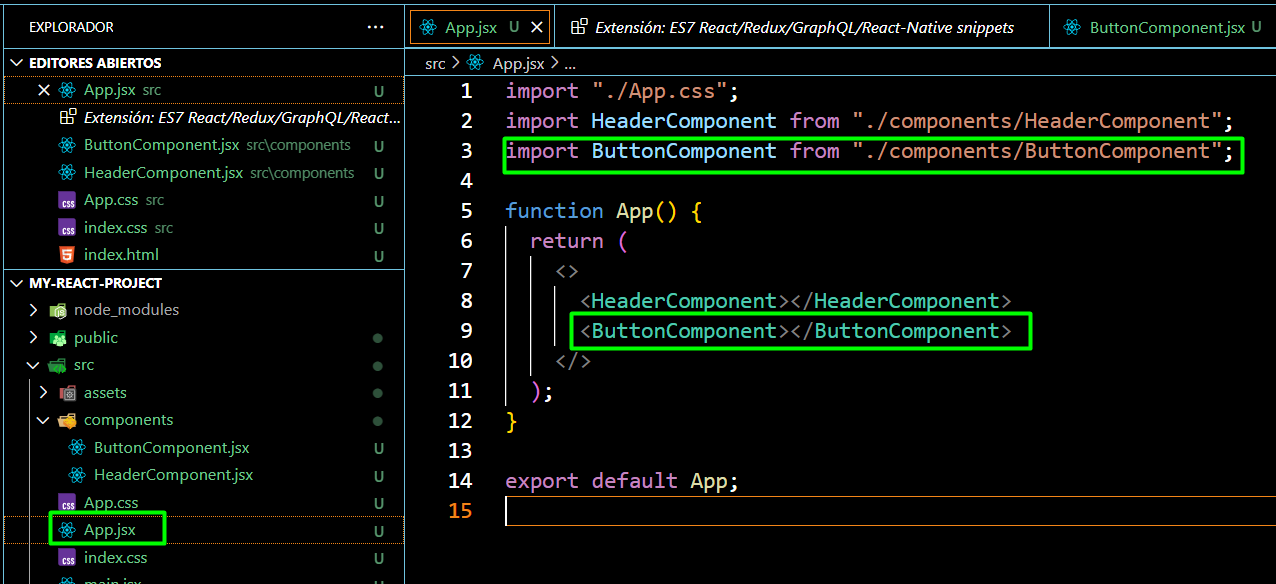
Crear componente con extensión



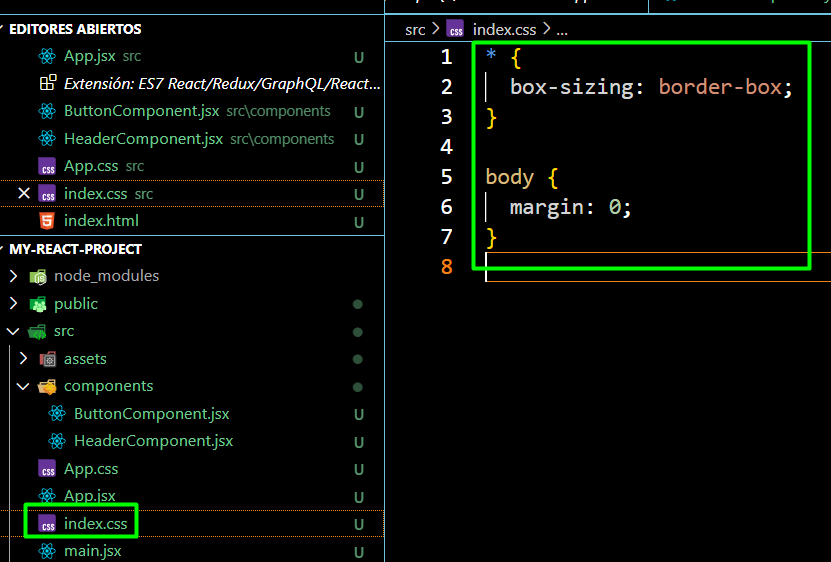




Usar componente

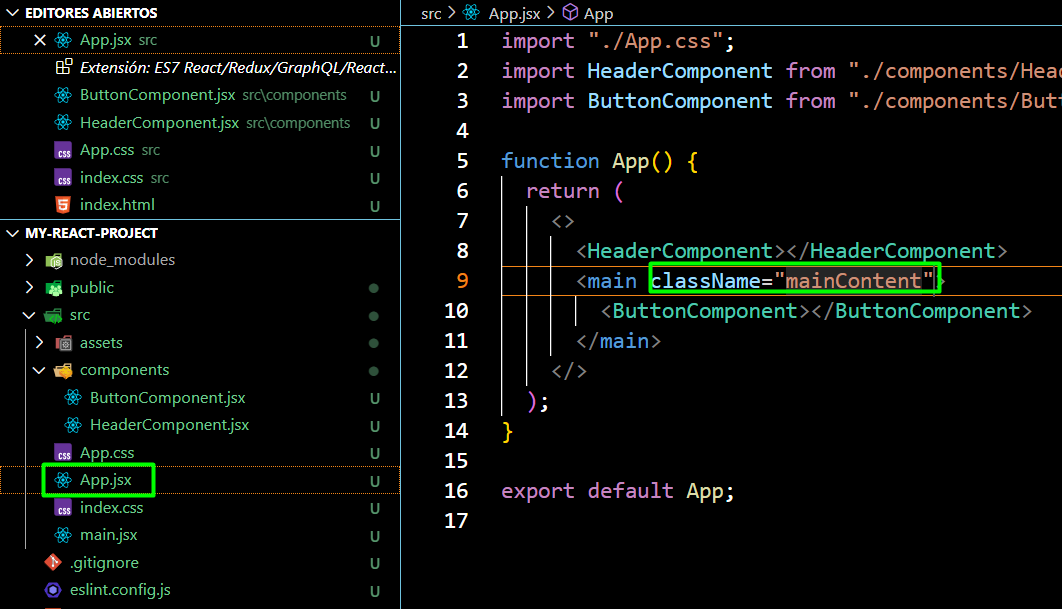


Estilos globales



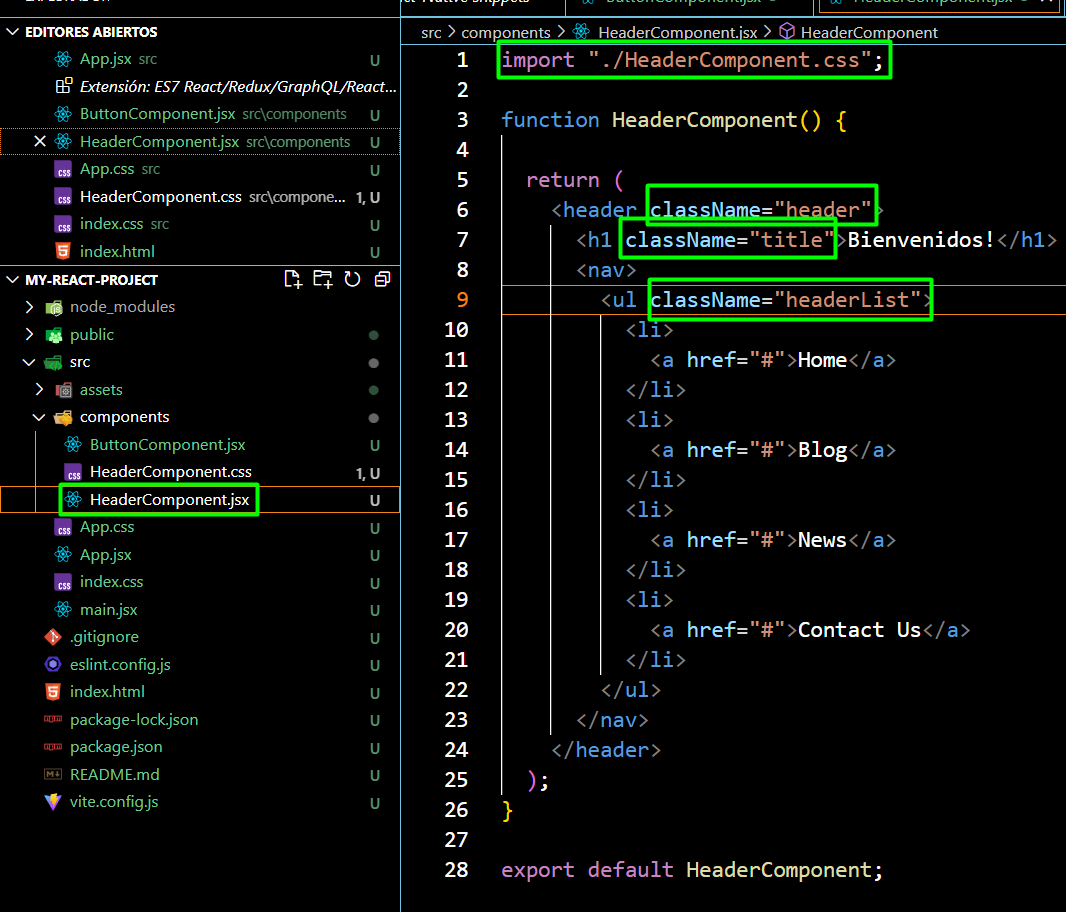
Estilos para componentes

Crear clases en app.jsx e implementarla en app.css

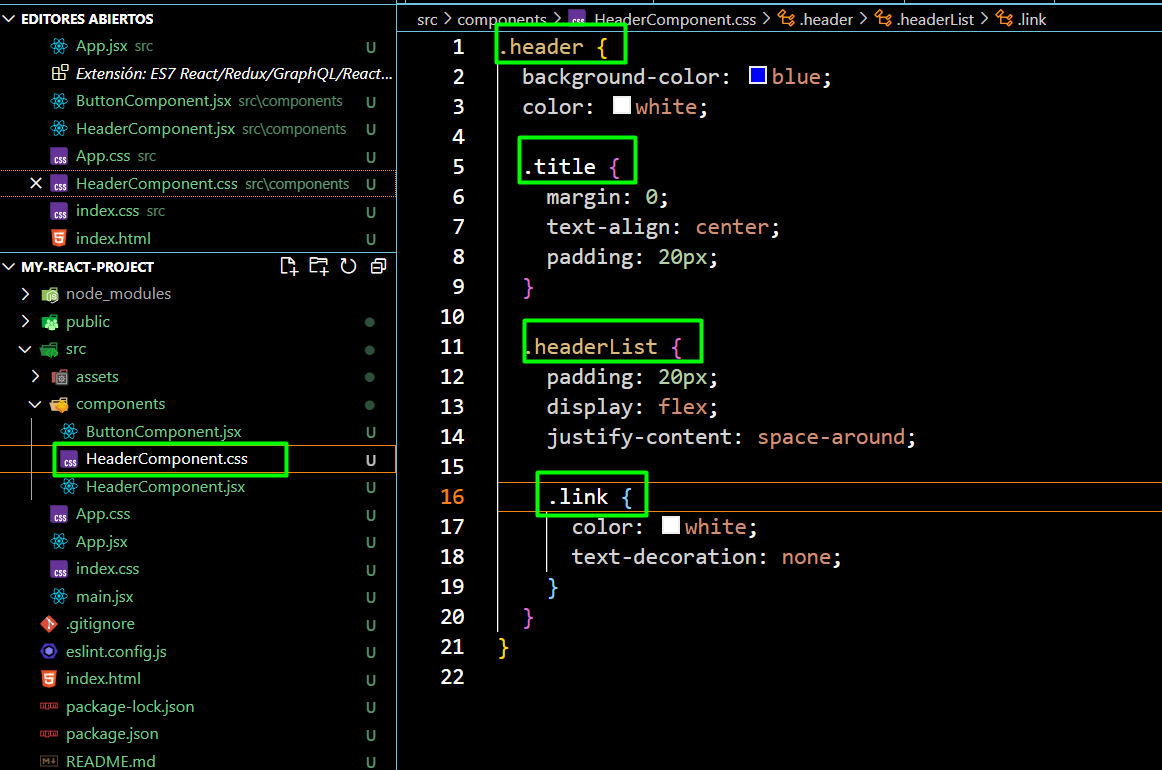


Estilo a un componente concreto

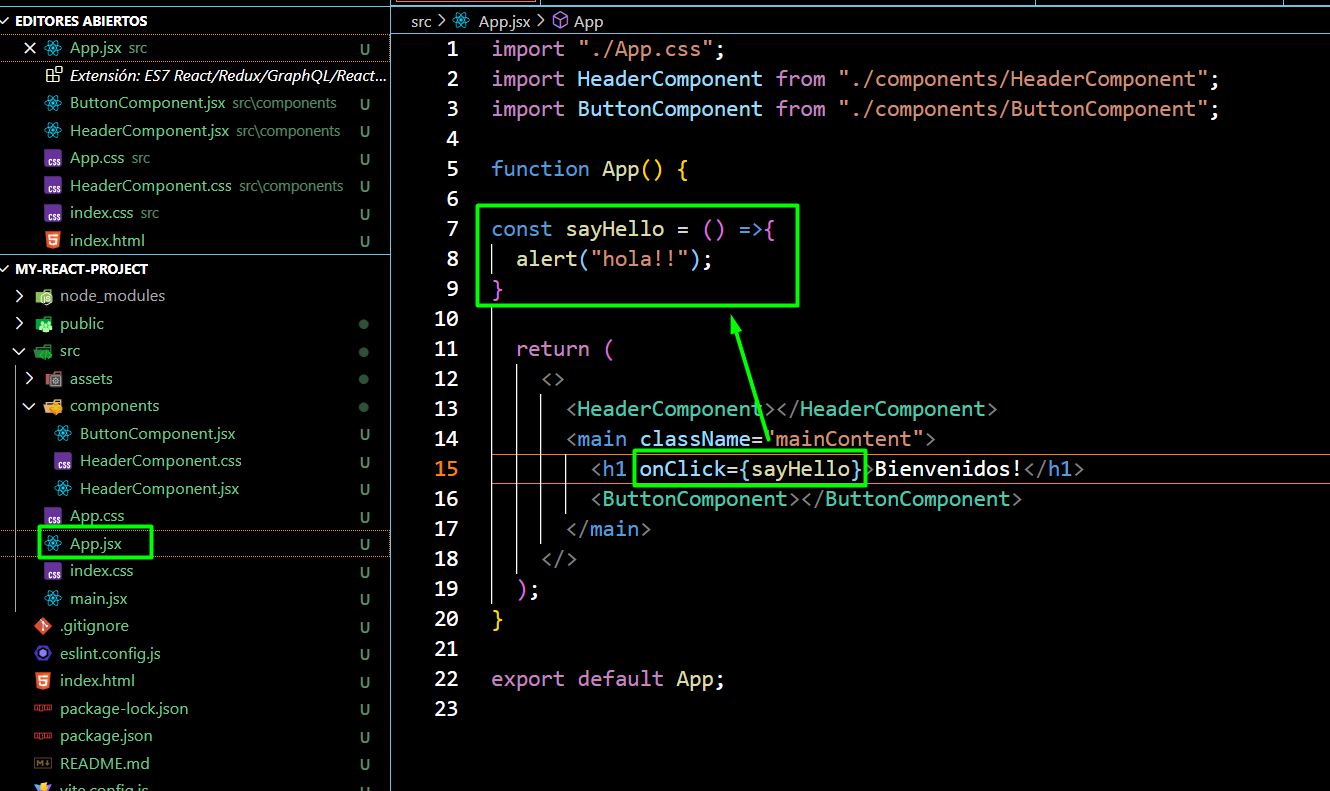
En jsx



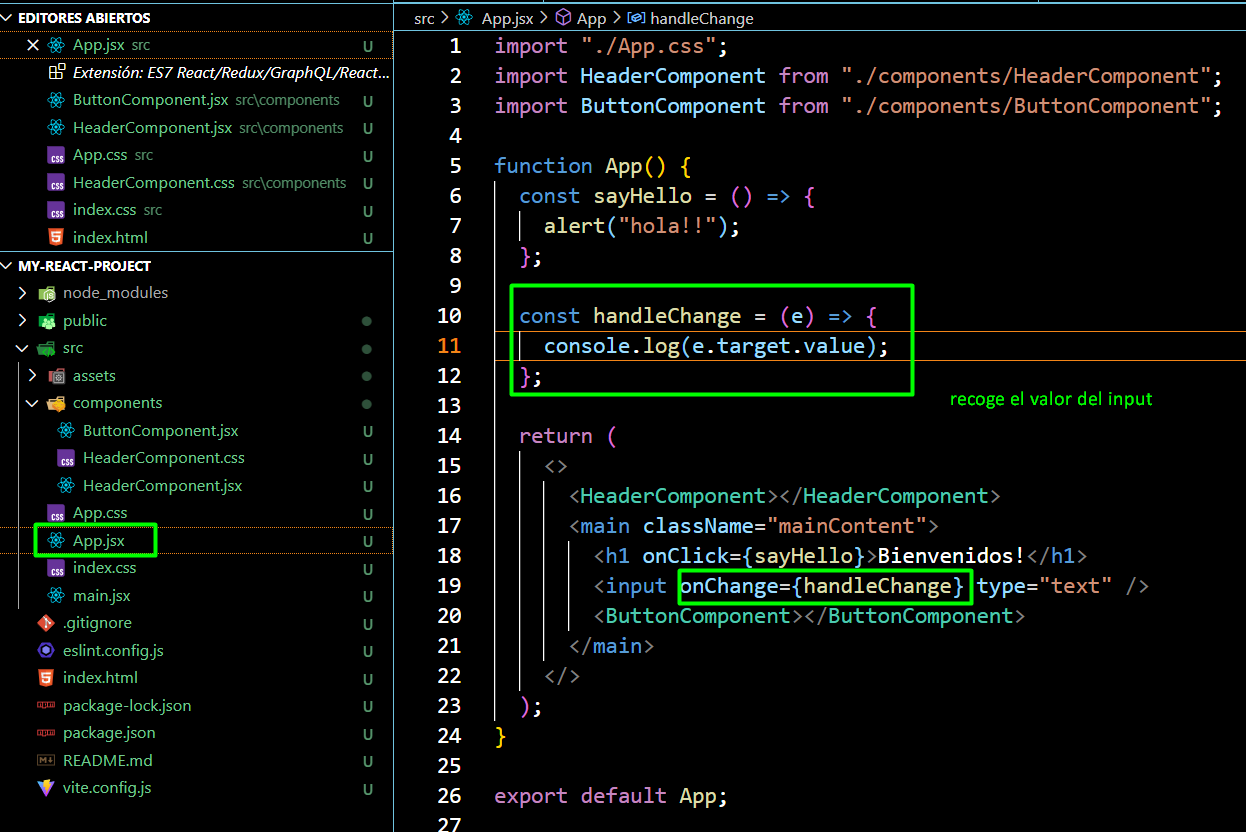
En css



Eventos



Evento onchange en input



Variables

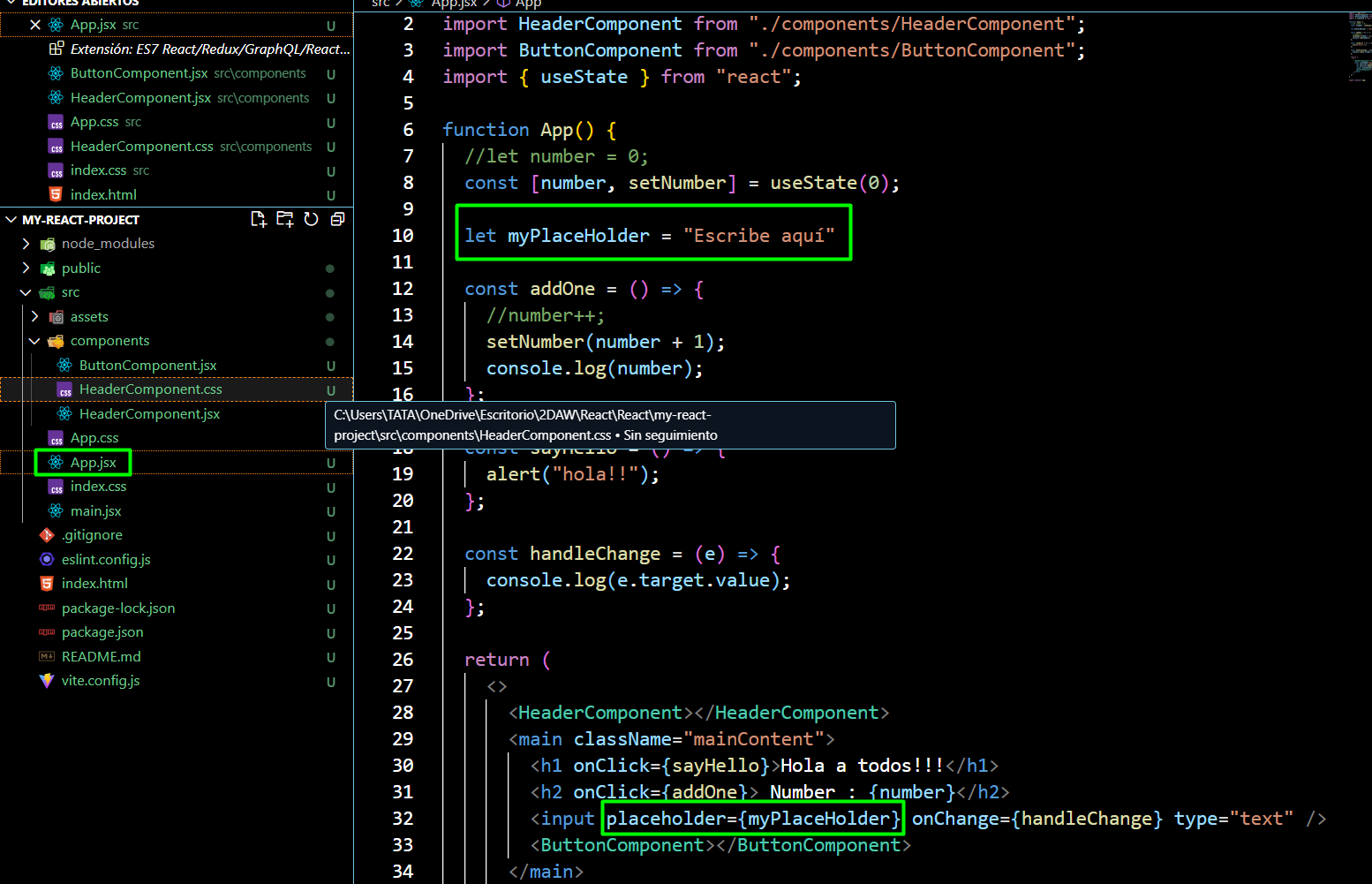
Con let, no se actualiza

Variables reactivas con hooks (funciones especiales de react)



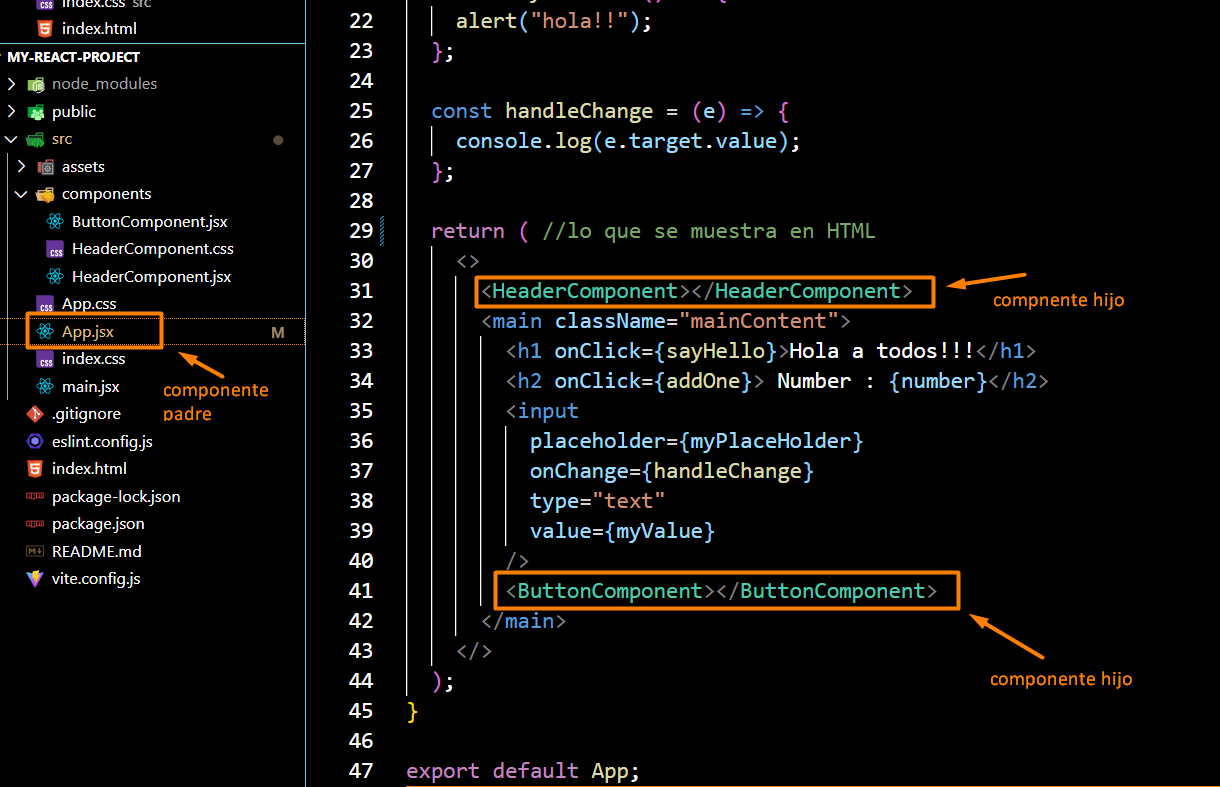
Las variables del state son asíncronas, no coincide con el valor en consola. CUIDADO

Variables en atribs html



Props pasar info de componente padre a componente hijo (variable)

Herencia entre componentes

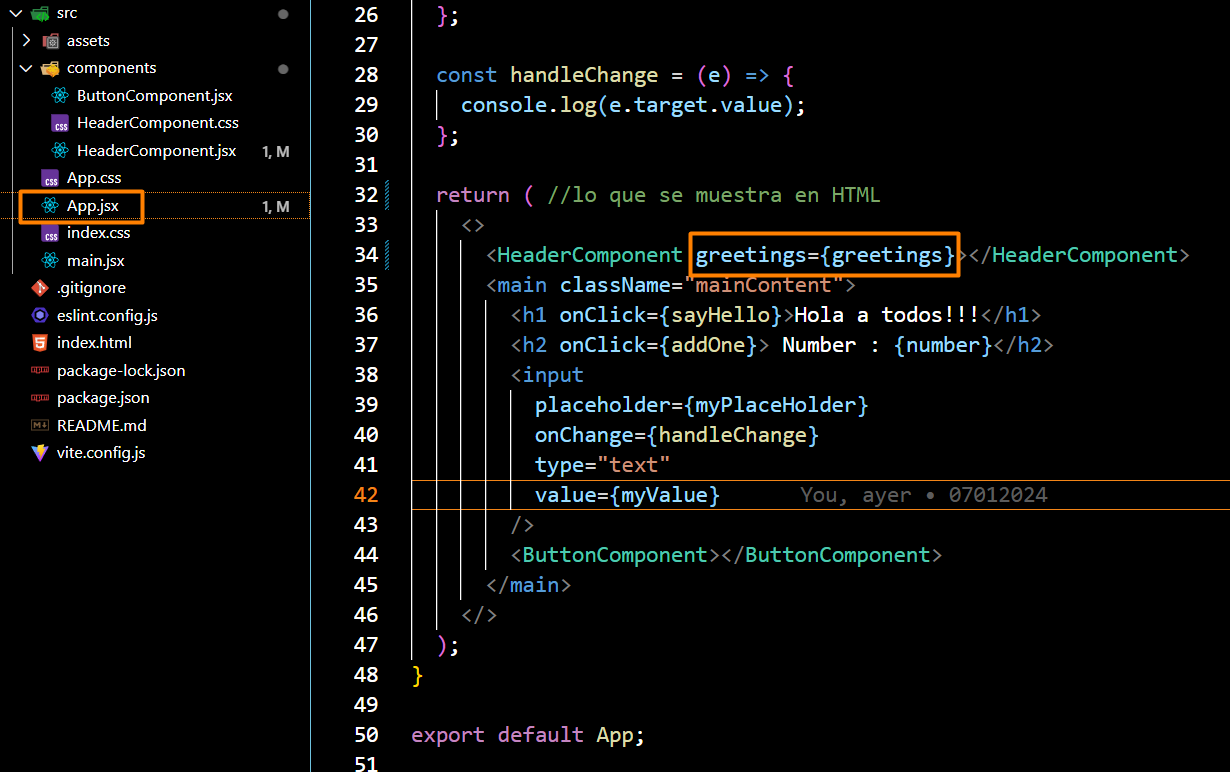


En el componente hijo le decimos en la función que va a recibir props

En app.jsx donde están las variables creamos una variable reactiva



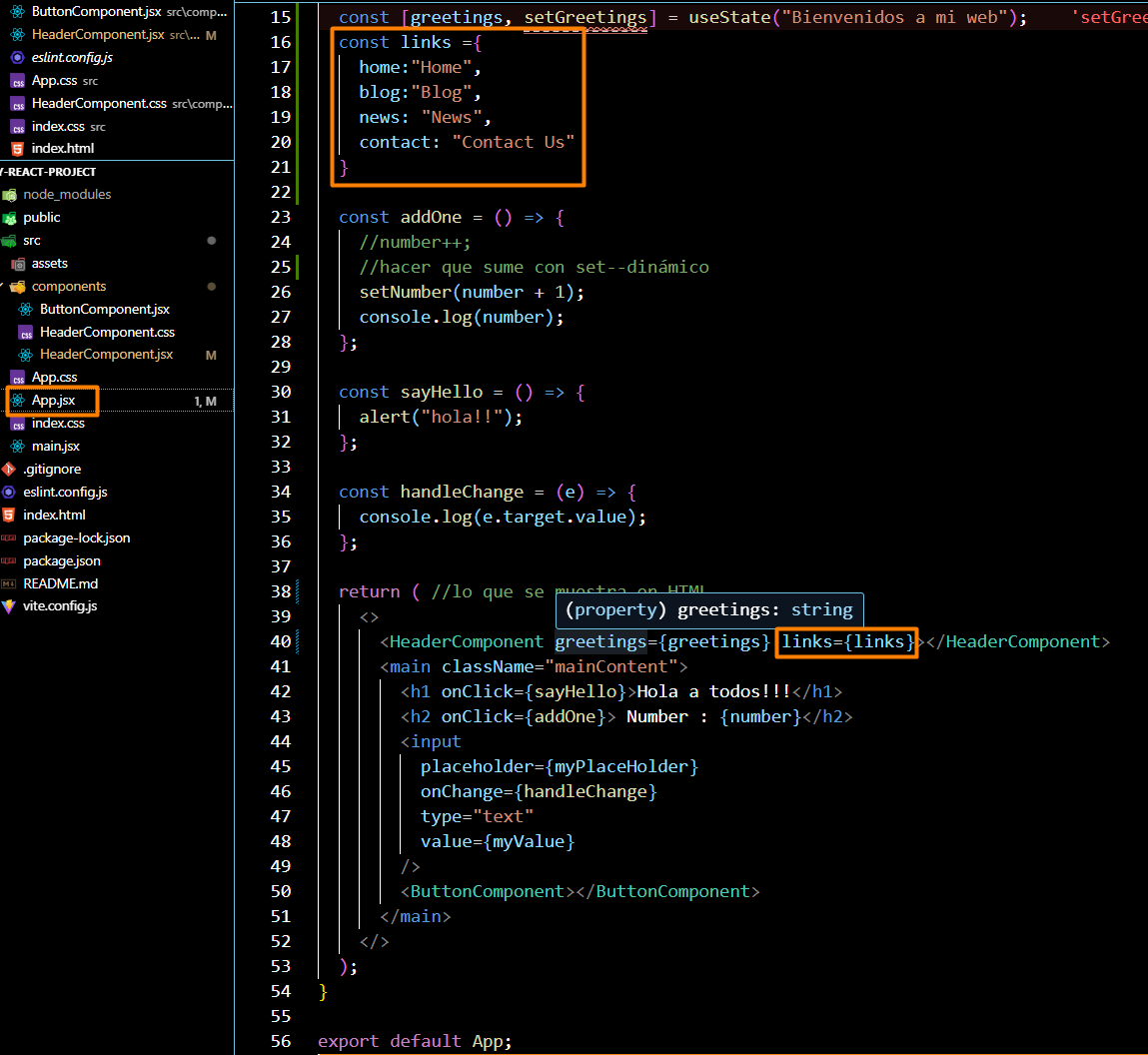
En app.jsx en el tag del componente hijo mencionamos la variable



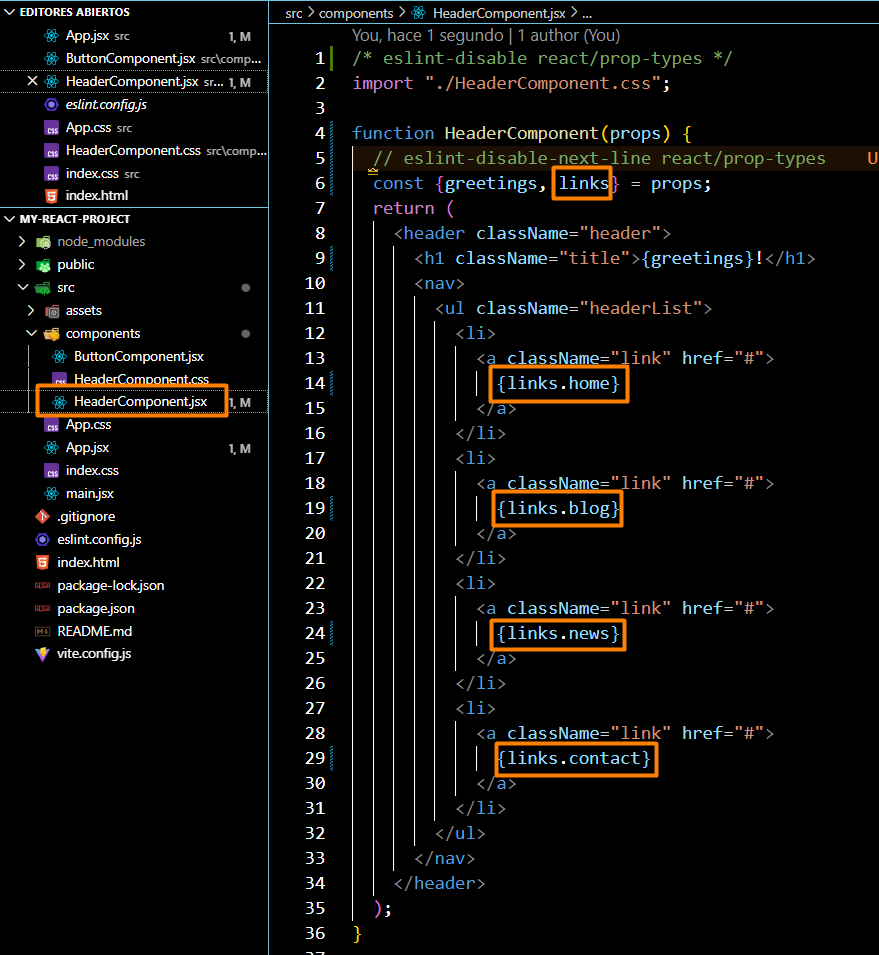
En componente hijo definimos props



Creamos otra variable en app.jsx y la mencionamos en los tags de return

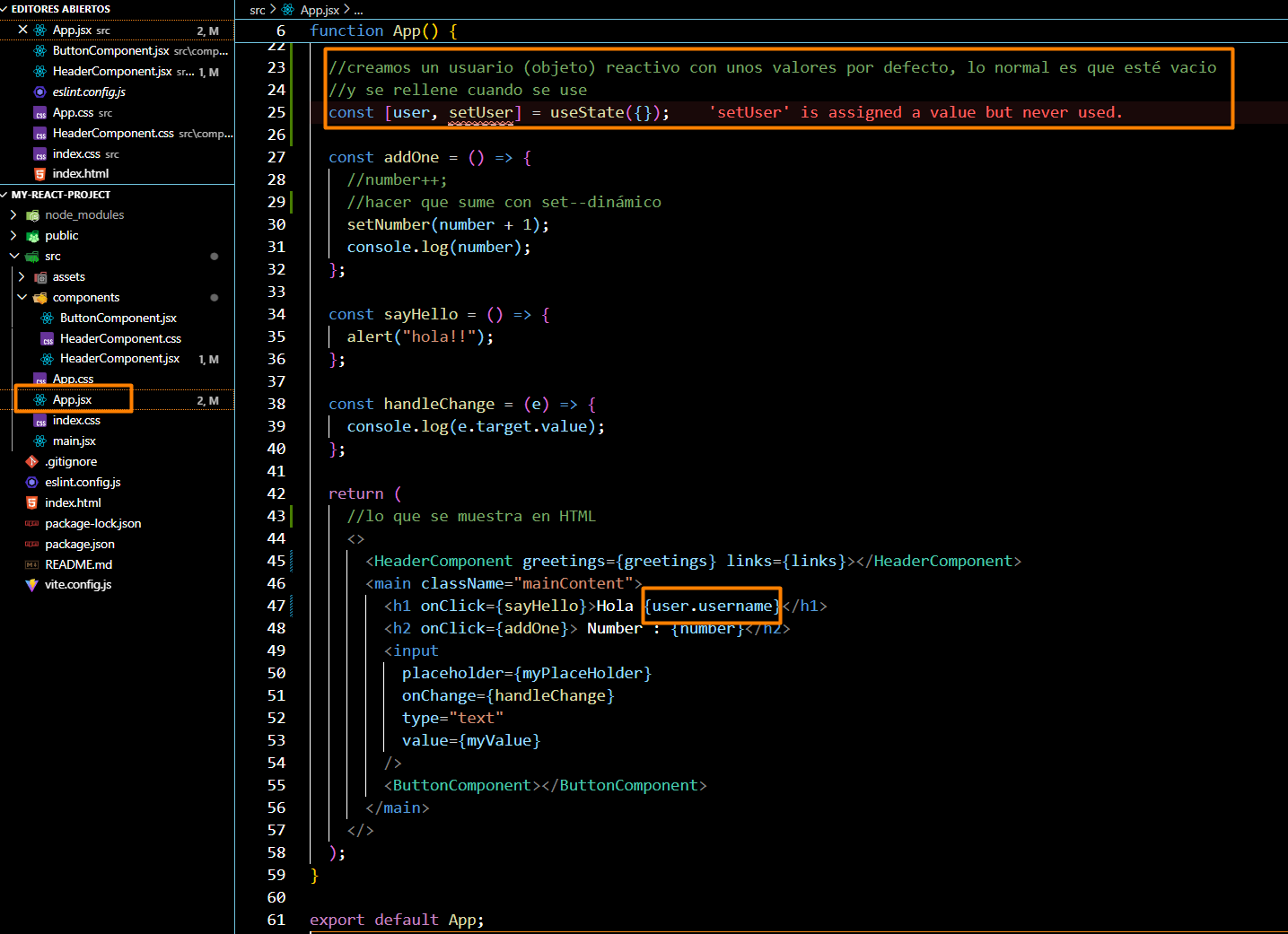


En componente



Pasar información de componente hijo a componente padre

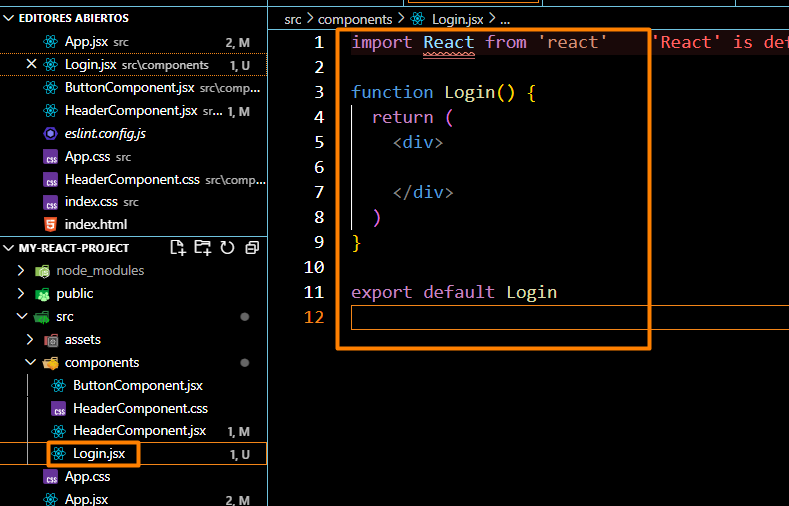
En app.jsx querremos saludar al usuario que ha hecho logiin, para ello en componente padre (app.jsx)



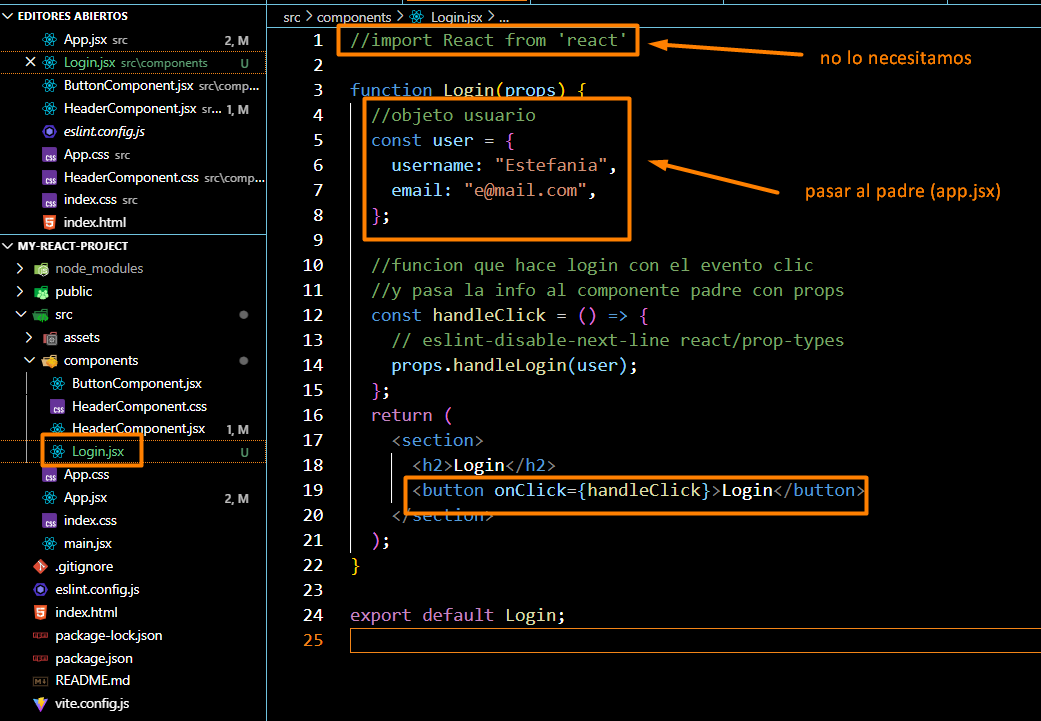
Creamos un cmp login para pasar esos datos al padre (app.jsx)

Login.jsx

Rfce +enter (creamos plantilla)

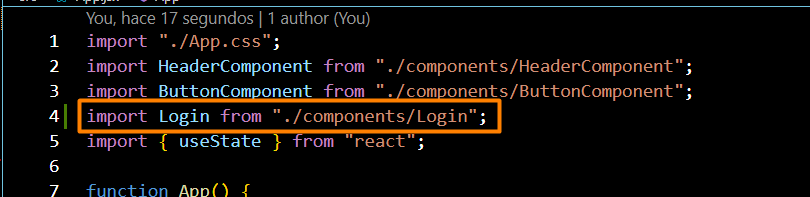


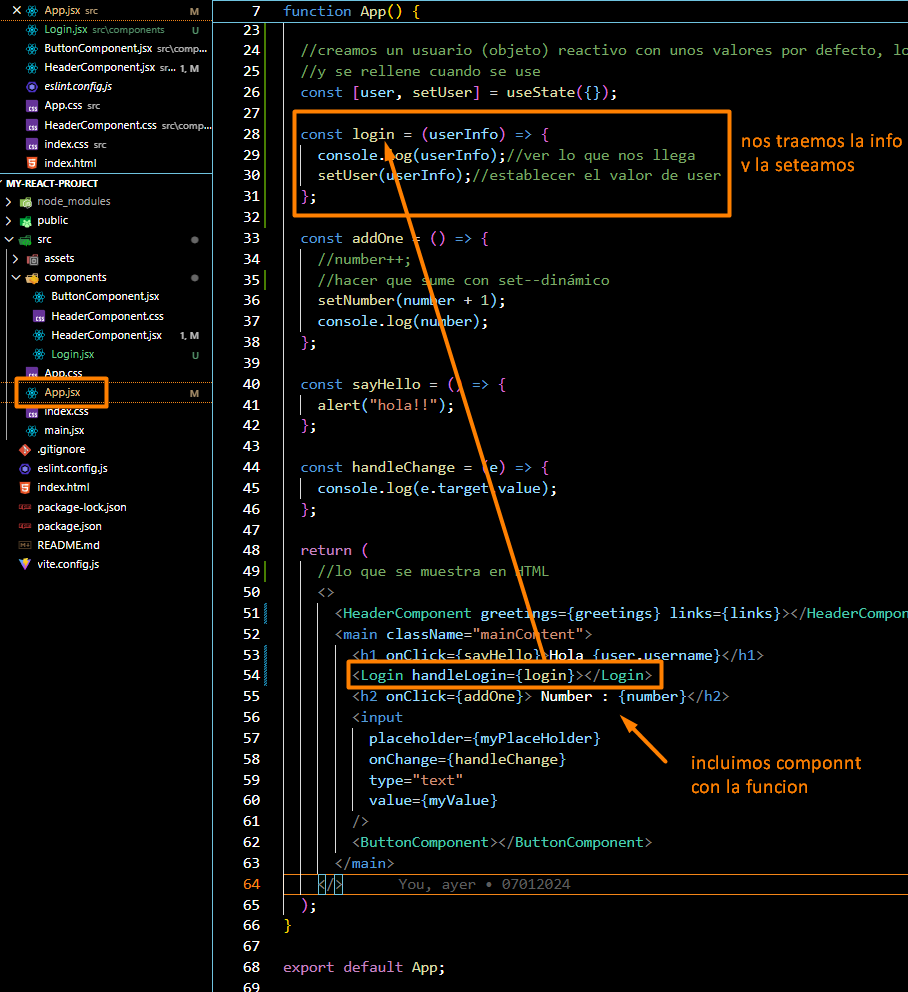
En login.jsx (componente hijo)



Con props.habdleLogin(user) en la función pasamos al padre

IMPORTAR NUEVO COMPONENTE (IMPORTANTE)





Mostrar condicionalmente algunos elmts en nuestro proyecto react (renderizado condicional)

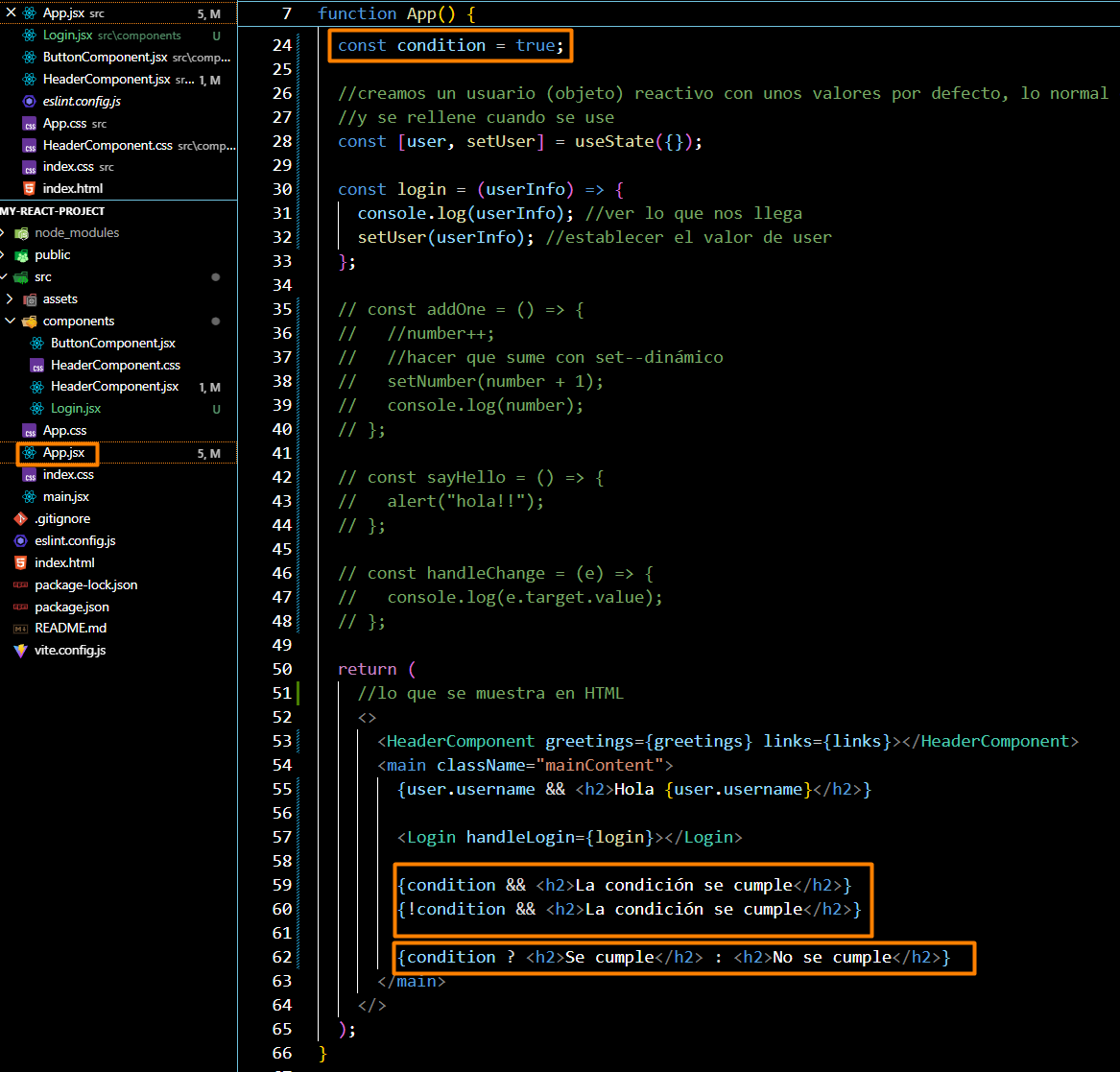
MOSTRAR HOLA USUARIO, SOLO SI TENEMOS USUARIO



Meter entre llaves el fragmento de código que queremos mostrar u ocultar

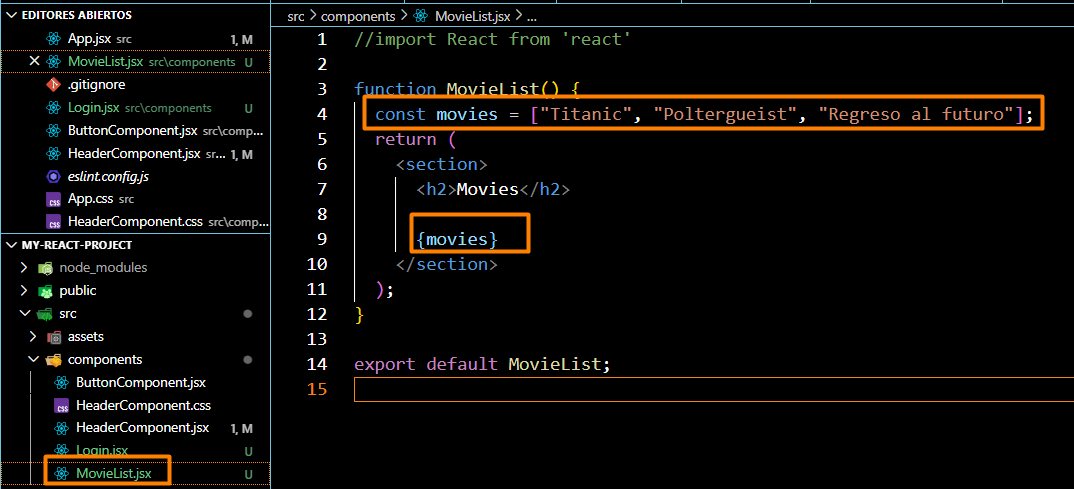
Condición && si es true se muestra lo siguiente, si no no se muestra nada

Otra forma, ternarios

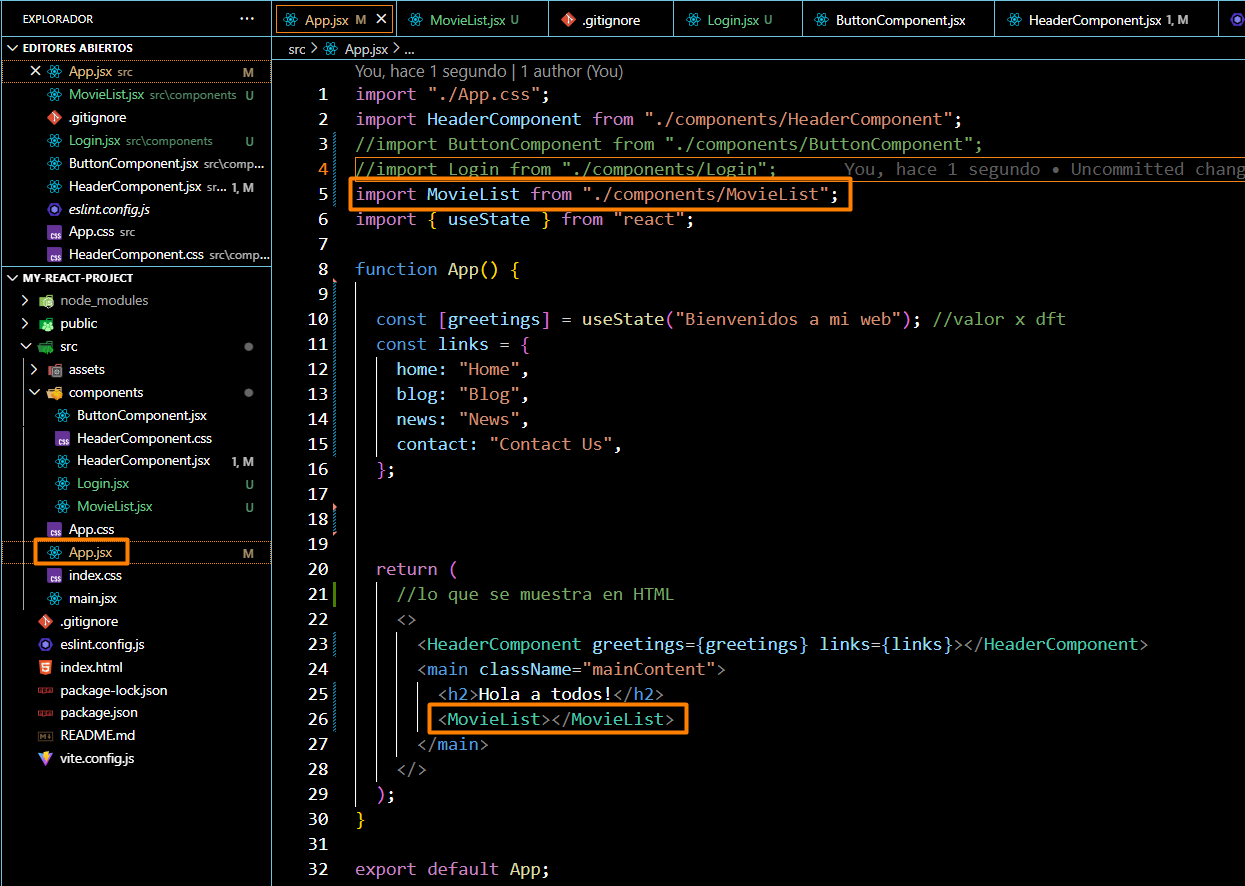


Renderizado de listas

Creamos nuevo comp

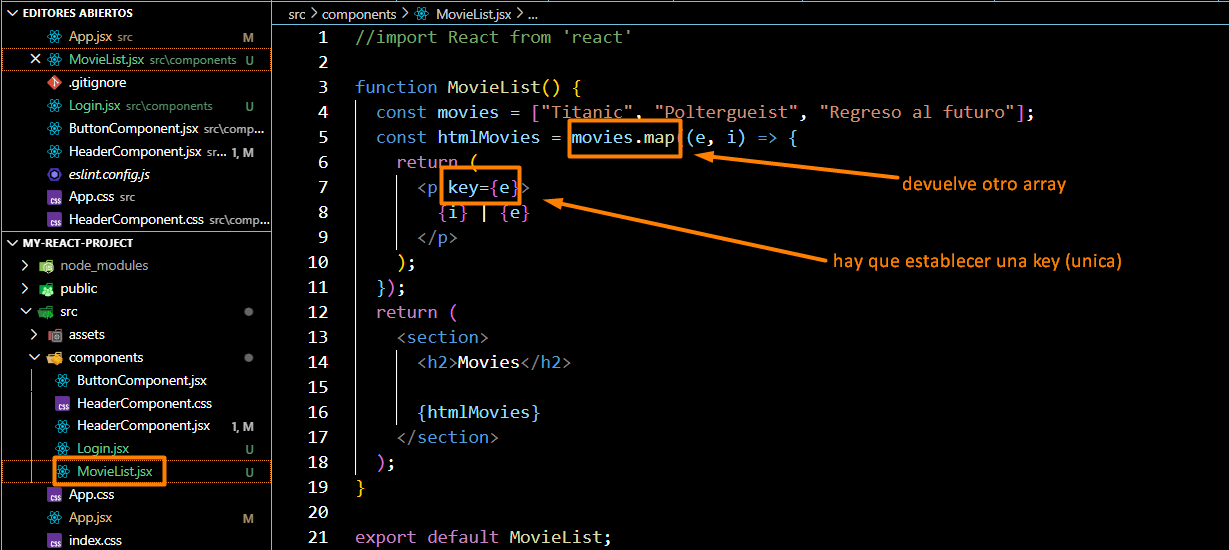


En padre (app.jsx)



PERO EN WEB NOS SALE LA LISTA SIN FORMATEAR

En componente



Mostrar una lista de OBJETOS

Crear comp

Crear array de objs

Formatear

