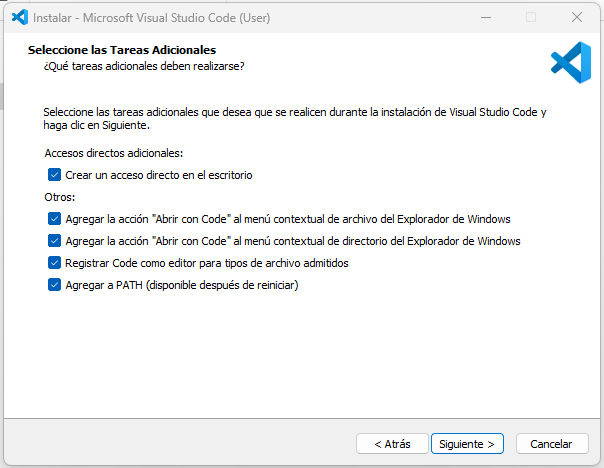
**Login con React**

https://www.youtube.com/watch?v=ZMCALHPeXCQ&list=PLAXdMOxCmqPv-AtG8yfYaqKvBTX20LI\_a&index=1

Requerimientos

1. Visual Studio Code (en esta parte es recomendable chulear todo)



1. Node JS
2. Git

Complementos

Vite (<https://vitejs.dev/>), nos permite crear proyectos base

Picoscss (<https://picocss.com/>), me proporciona código para mejorar el stylo

netlify (https://app.netlify.com/drop) , me publica mi archivo en la web

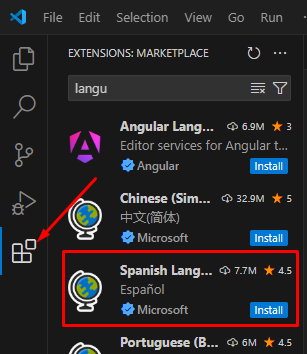
Paso a paso

Creo una carpeta donde se va a alojar todo mi código

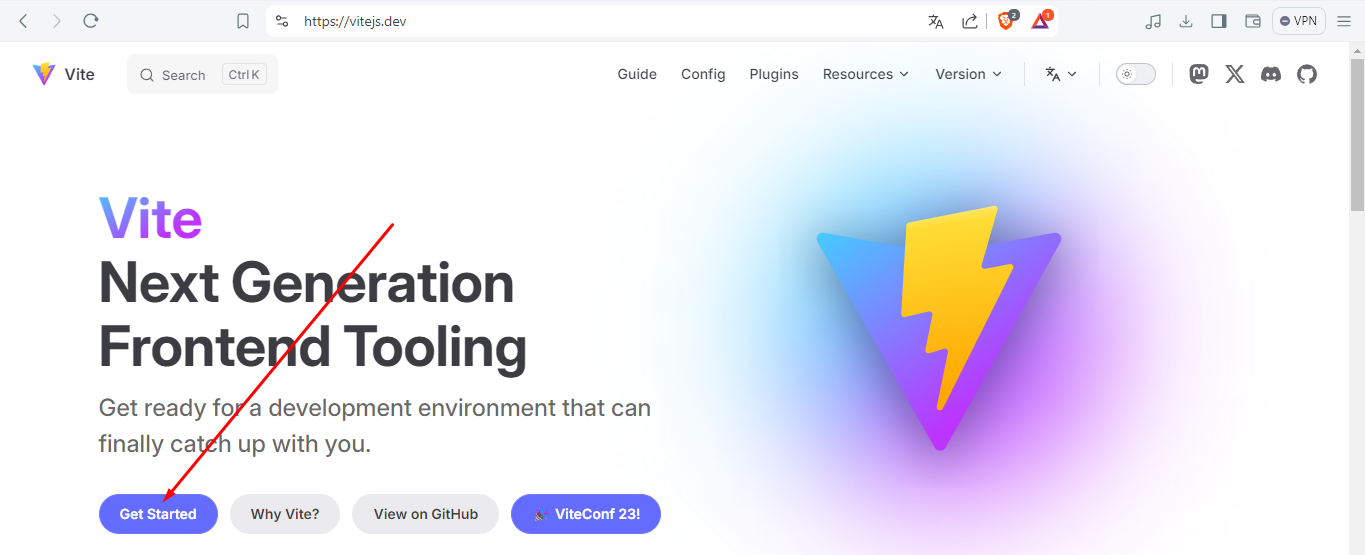


Doy click derecho en ella y doy en abrir con visual studio code

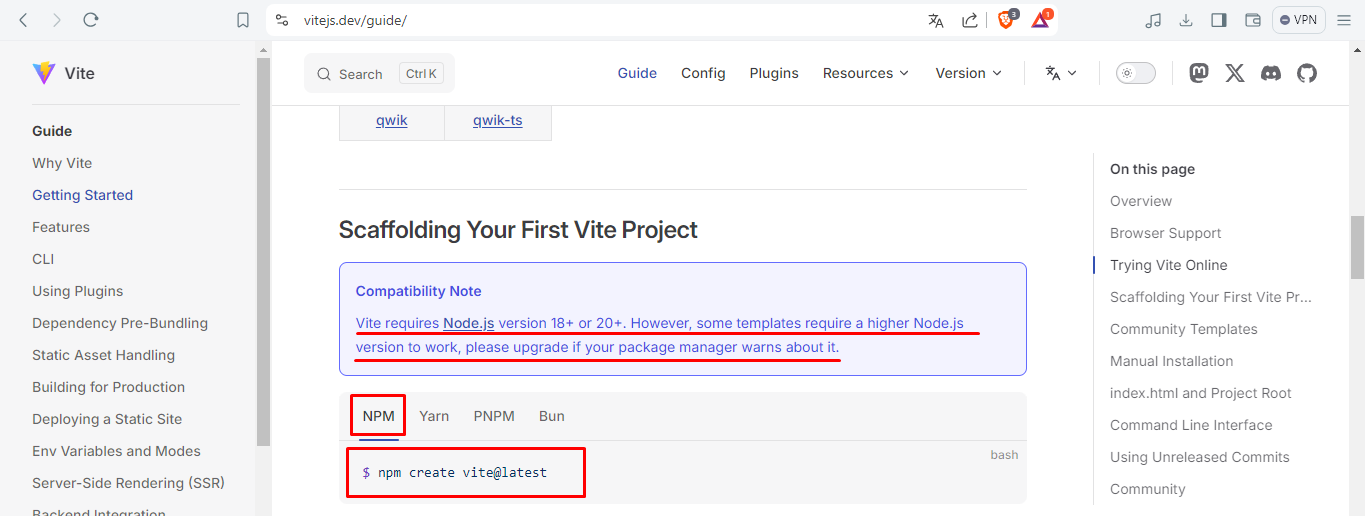
Ahora voy a cambiar el idioma de mi VSC, para lo cual instalo el paquete de español.



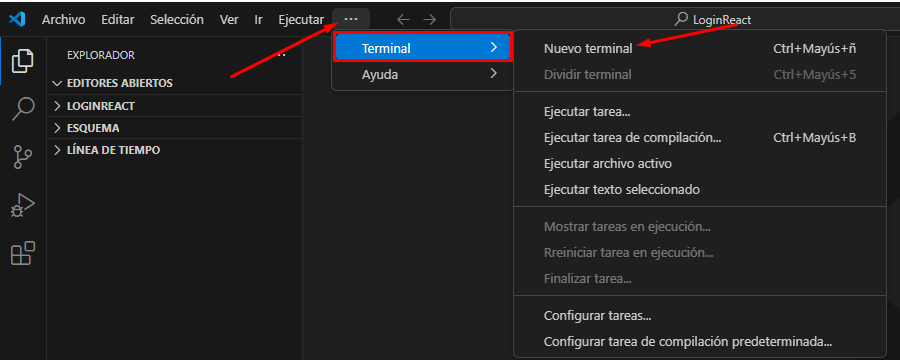
Voy a vite y doy en get started



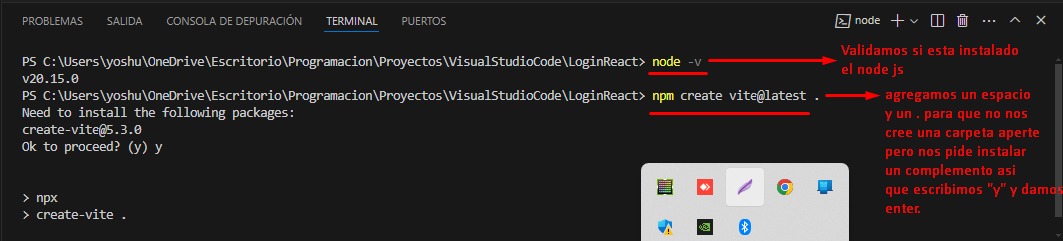
Bajo y voy a buscar el siguiente código y lo copeo



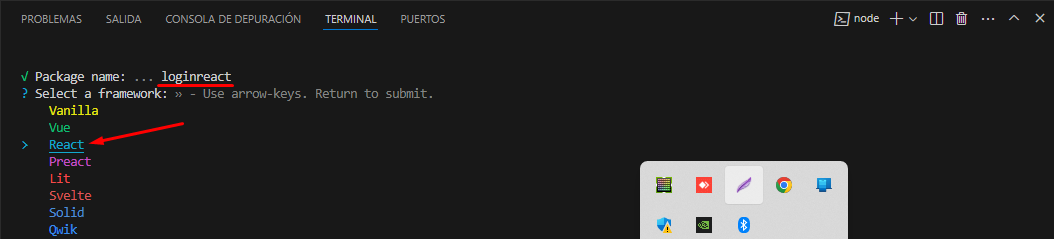
Voy a VSC y abro una nueva terminal

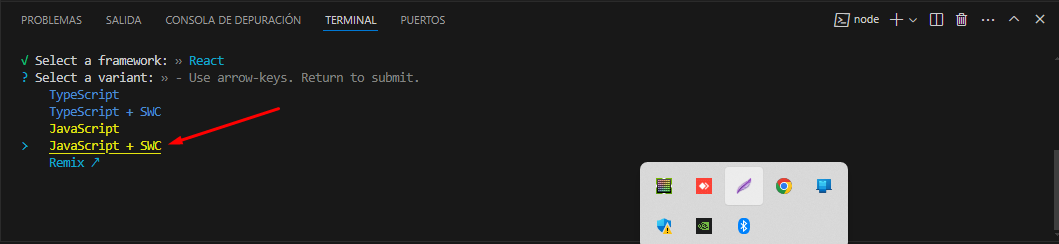


Y pegamos el código que copiamos de vite, pero para que no me cree una carpeta por fuera de mi carpeta, doy un espacio y coloco un punto, y luego doy enter.

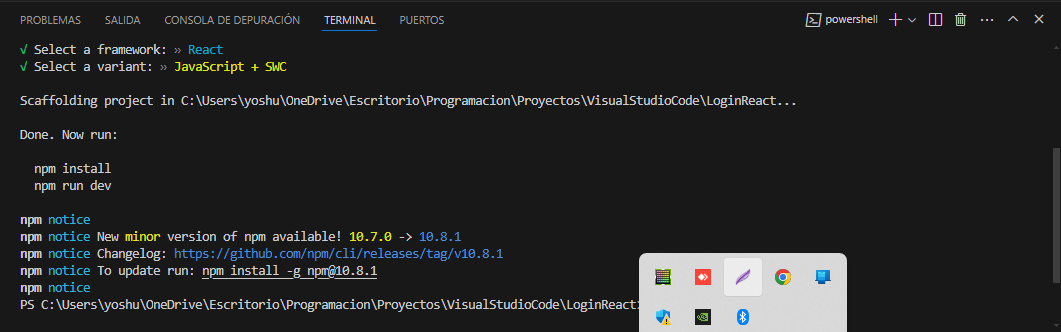


Luego nos salen unas preguntas que vamos a ir contestando, con las flechas escogemos react y damos enter.

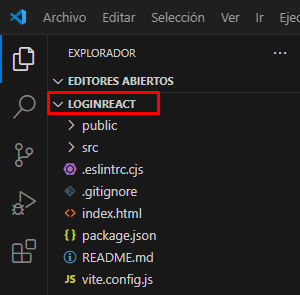




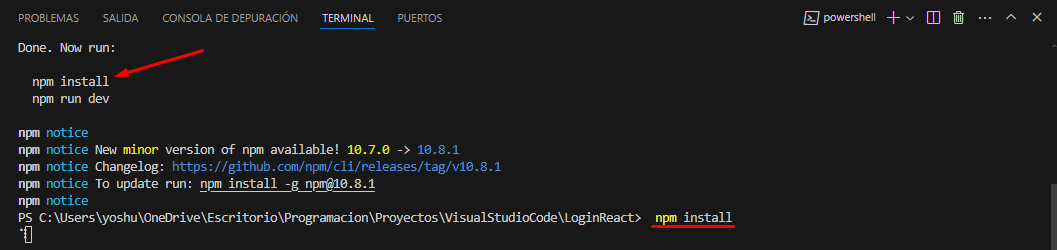
Volvemos a dar enter y nos sale lo siguiente



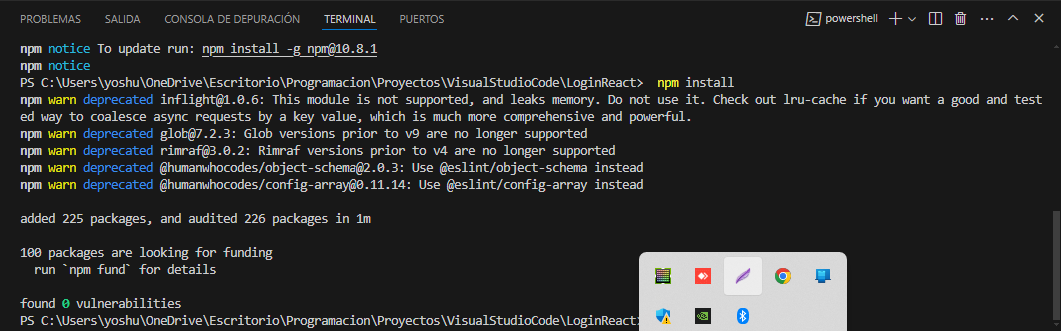
Y si vamos a nuestra carpeta vamos a ver que nos creó varios documentos



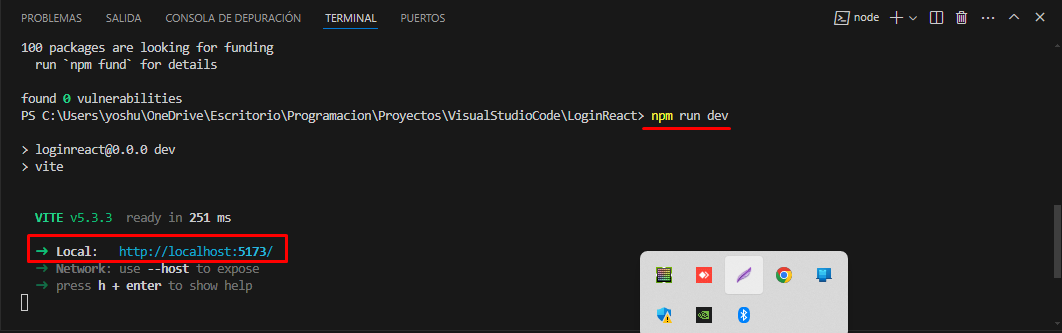
Ya tenemos nuestro proyecto base, perro ahora debemos instalar las dependencias, las dependencias son códigos que ya están creados y que nosotros vamos a reutilizar para ello instalamos el npm.



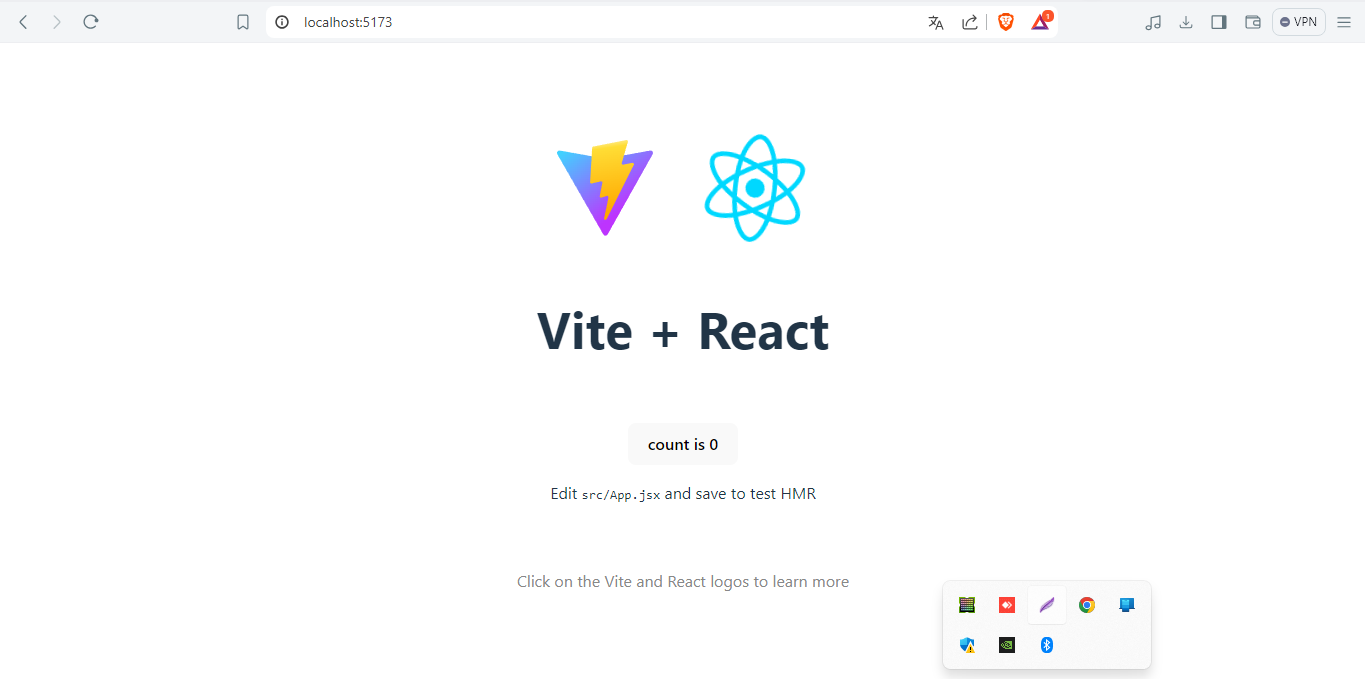
Va a demorar un poco mientras instala las dependencias que nuestro proyecto necesita para funcionar.



Ahora ejecutamos el otro código que estaba, y este nos va a crear un local host.



Si copeamos ese localhost y lo pego en el buscador nos va a salir

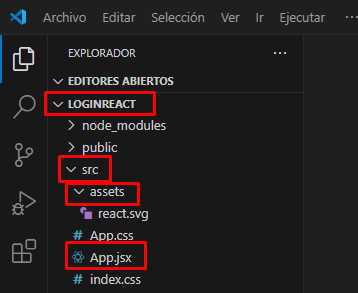


Debo tener en cuenta que cada vez que cierro VSC el servidor se va a detener, para lo cual vamos a ejecutar el código:

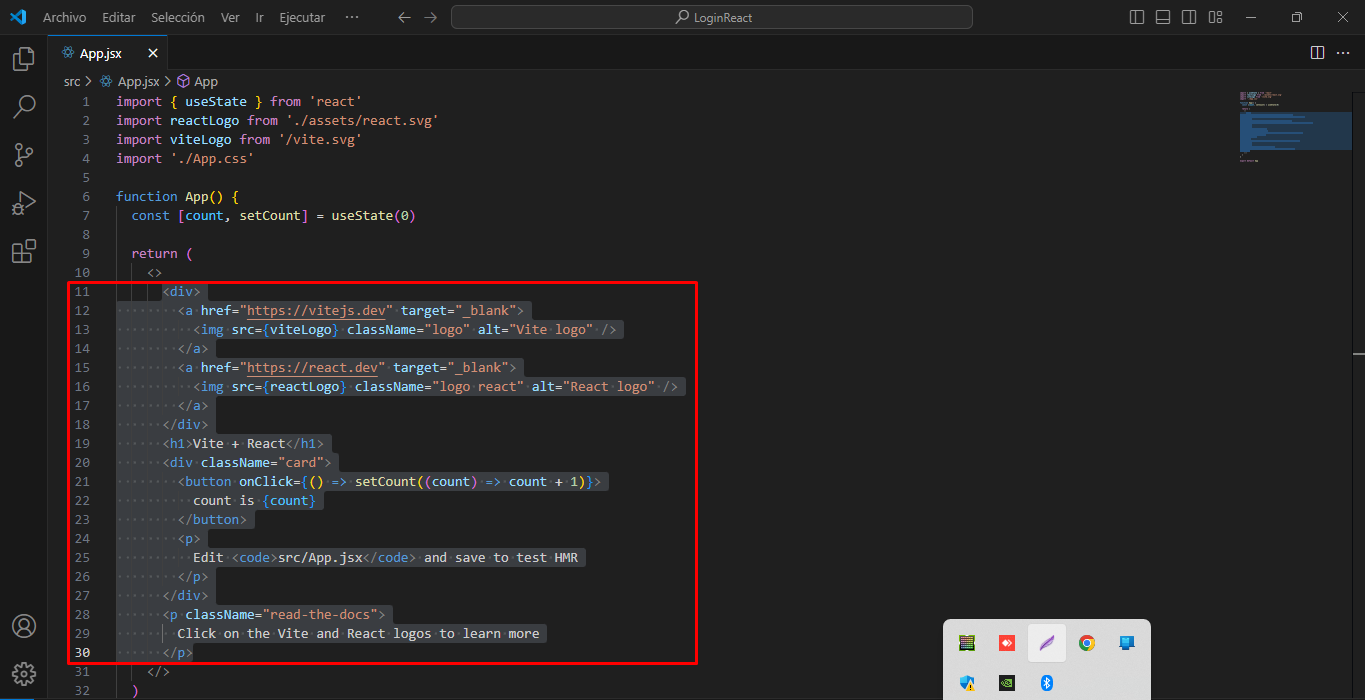
**npm run dev**

no debemos ejecutar los otros códigos porque los demás solo se crean una vez.

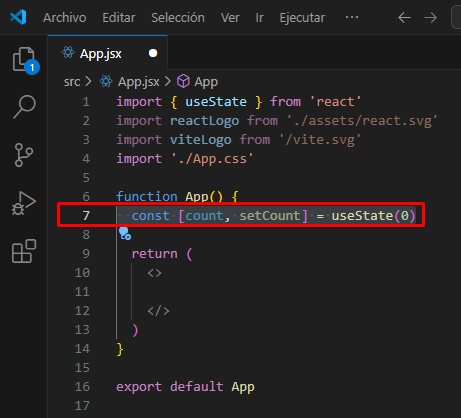
Ahora vamos a trabajar en nuestra pagina del local host. Vamos a crear dos inputs, uno de usuario uno de contraseña y un botón para ingresar para ellos seguimos el siguiente recorrido



Borramos de la línea 11 a la 30, pues ese código es lo que nos muestra la presentación del localhost



También vamos a eliminar la línea 7

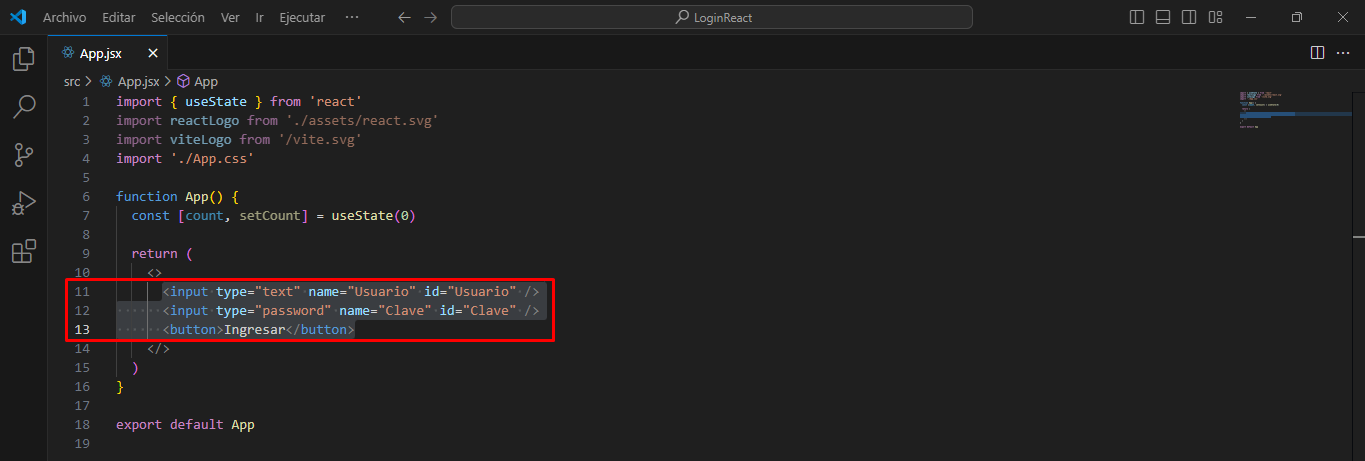


Ahora vamos a empezar a escribir el código:

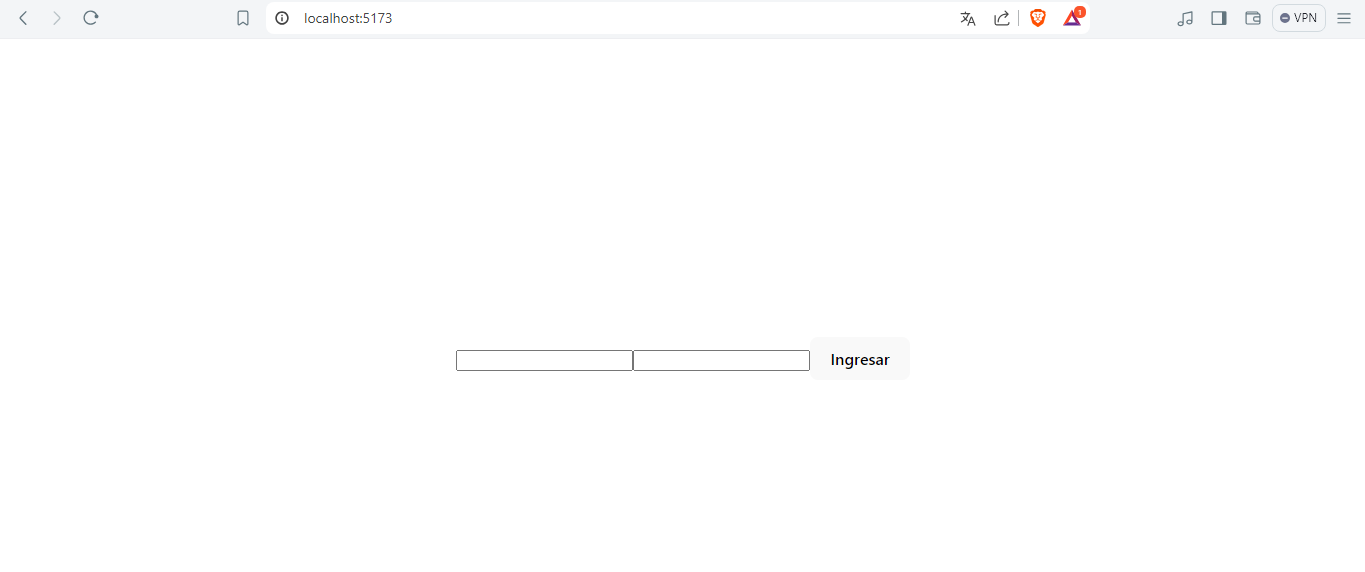
<input type="text" name="Usuario" id="Usuario" />

<input type="password" name="Clave" id="Clave" />

<button>Ingresar</button>

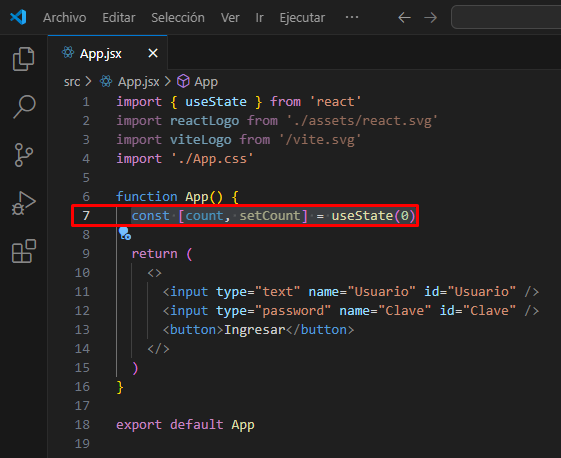


Guardamos y vamos a actualizar mi localhost

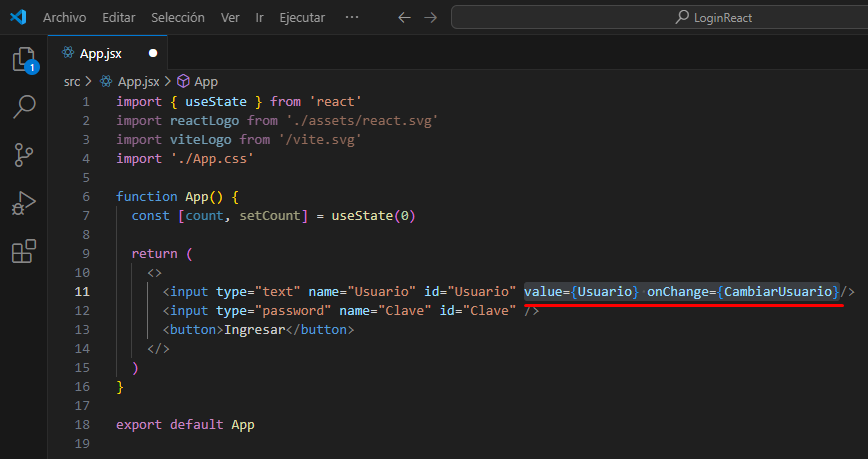


Ahora vamos a empezar a guardar los datos que coloque y darle funcionalidad a mi botón. Para ello vamos a crear las constantes para guardar los valores. Agregamos el siguiente código:

const [count, setCount] = useState(0)



value={Usuario} onChange={CambiarUsuario}



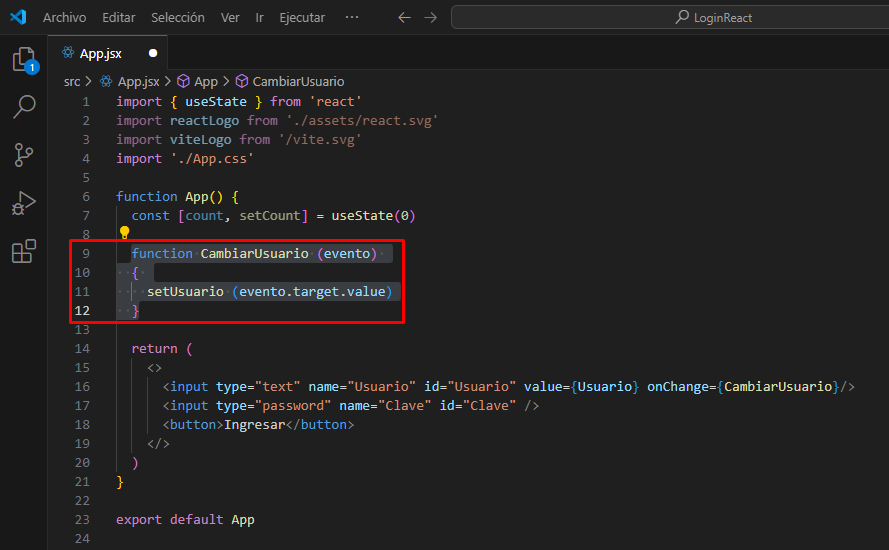
Agrego el siguiente código:

function CambiarUsuario (evento)

{

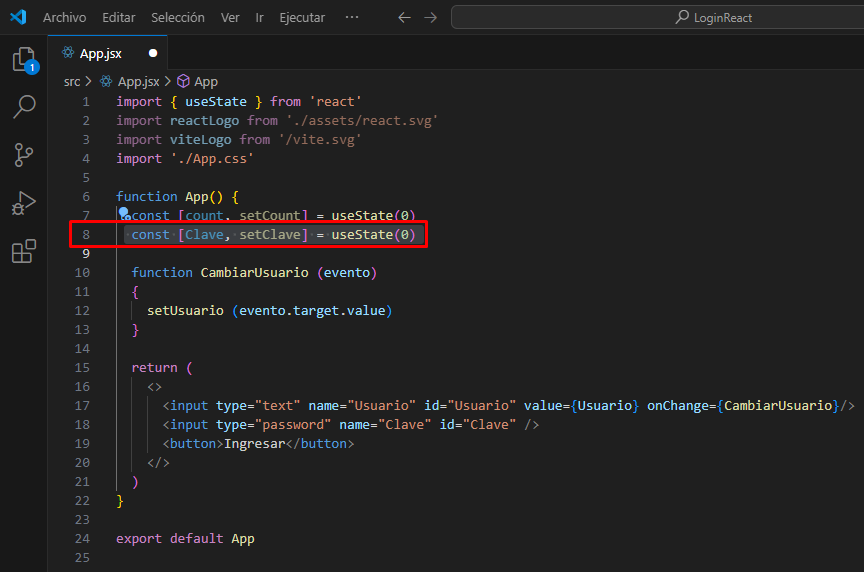
setUsuario (evento.target.value)

}



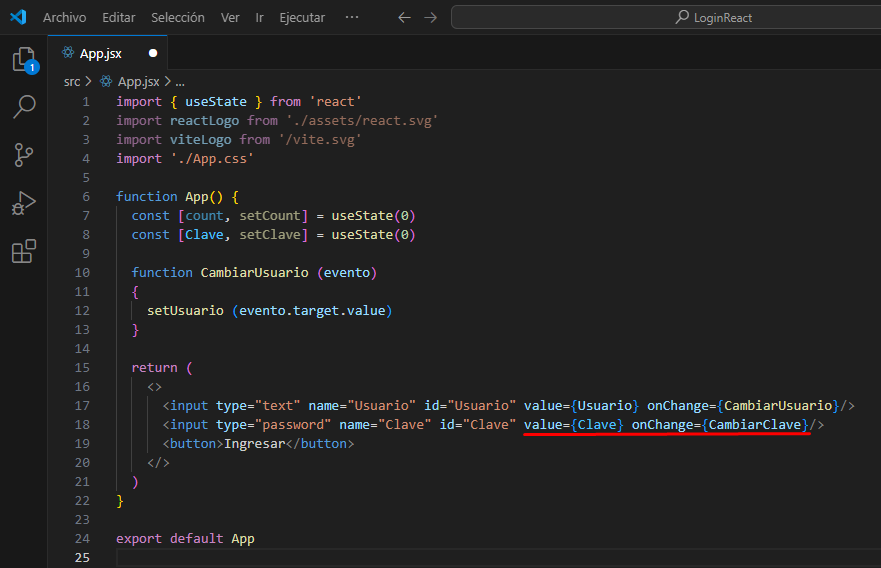
Agrego el siguiente código:

const [Clave, setClave] = useState(0)



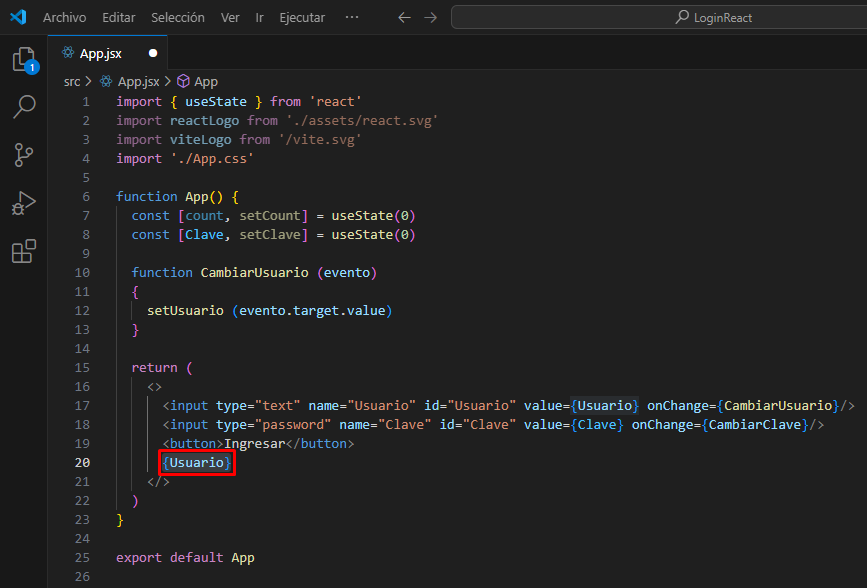
Agregamos el siguiente código:

value={Clave} onChange={CambiarClave}



Agrego el siguiente código:

{Usuario}



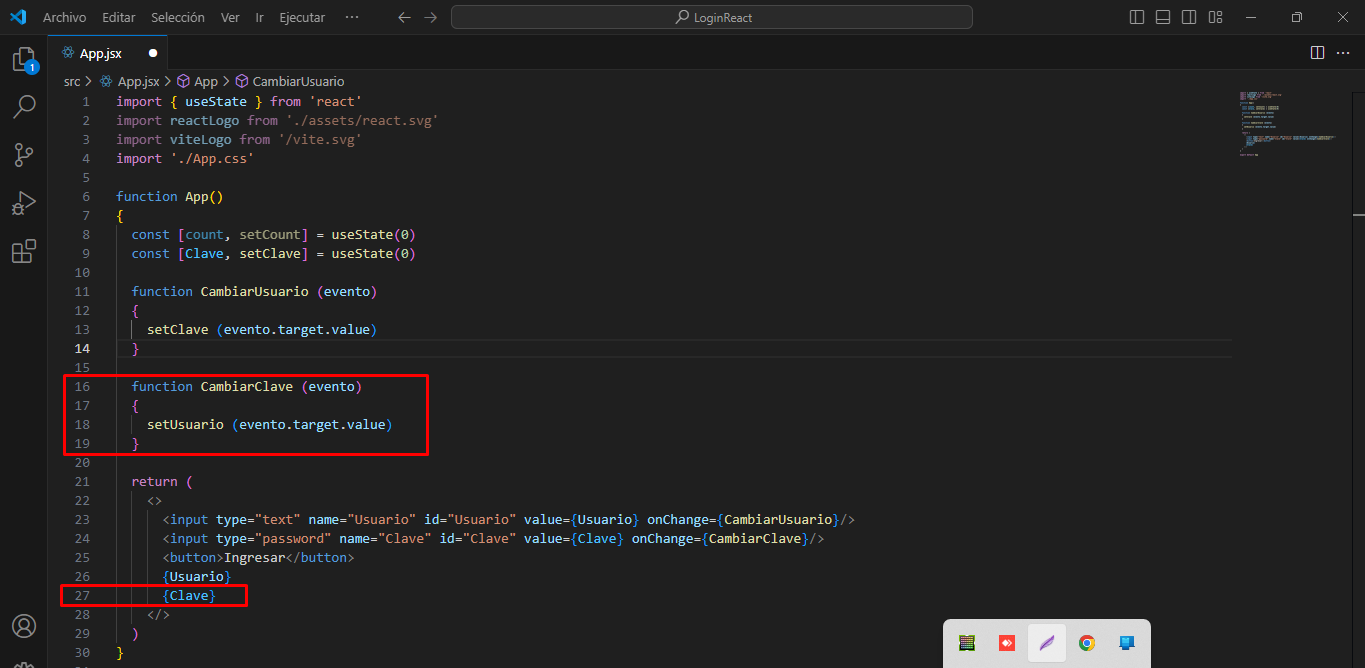
Ahora vamos a agregar el siguiente codigo:

function CambiarClave (evento)

{

setClave (evento.target.value)

}



Ahora vamos a trabajar sobre el botón

Agregamos el siguiente código:

function Ingresar ()

{

if (Usuario == "Admin" && Clave == "Admin")

{

alert ("Bienvenido")

}

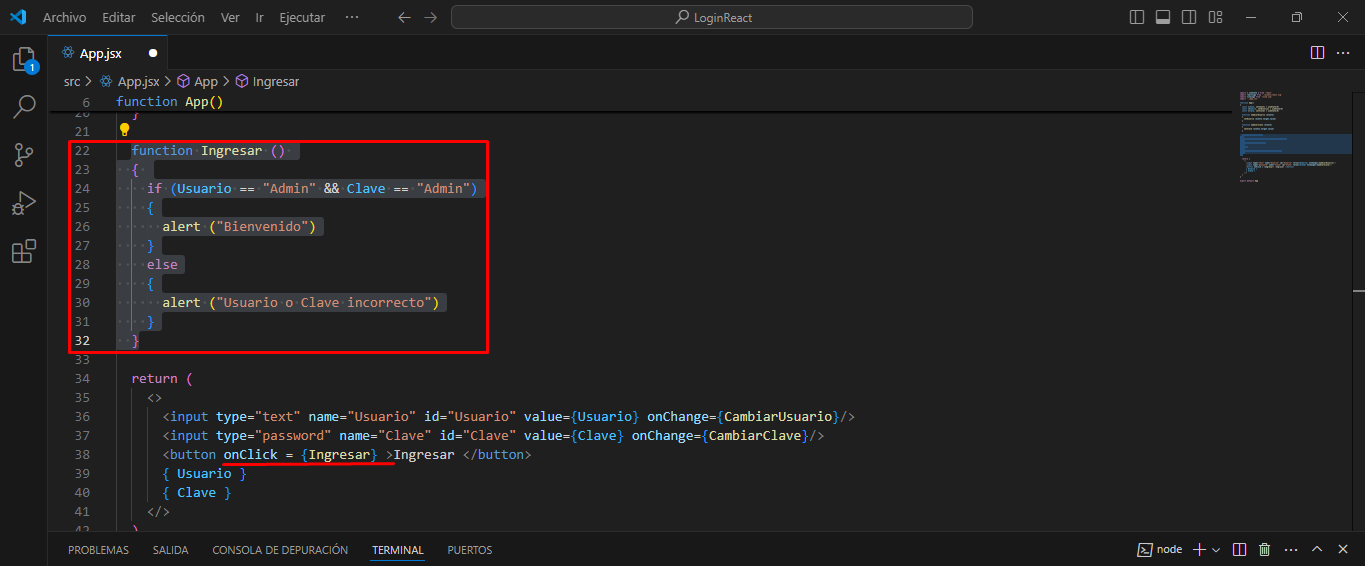
else

{

alert ("Usuario o Clave incorrecto")

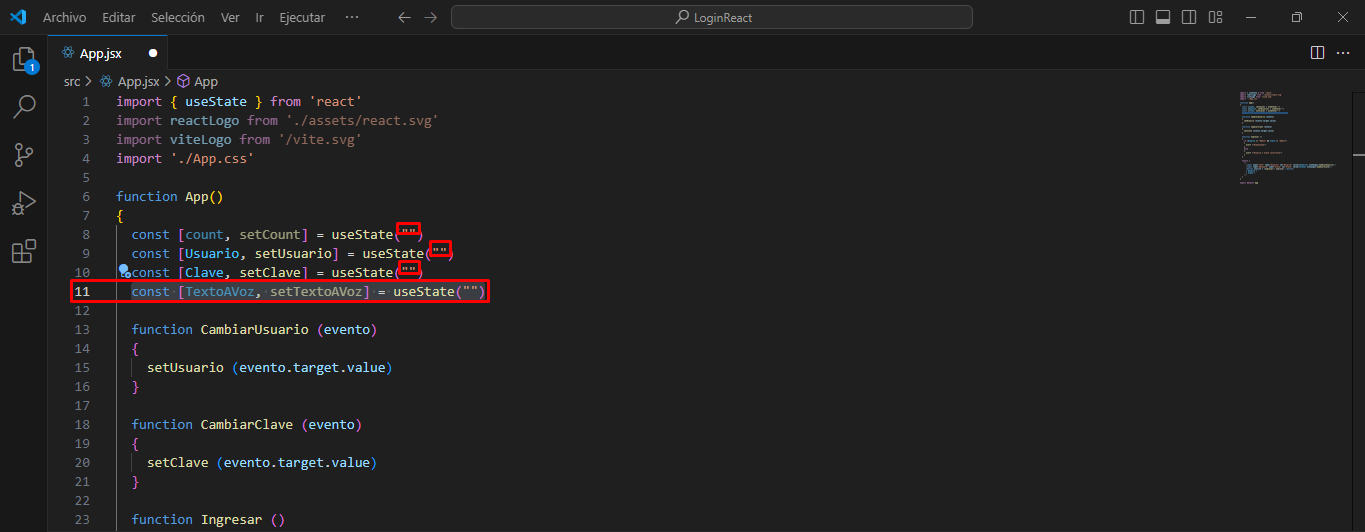
}

}



Ahora voy a crear otra constante:

const [TextoAVoz, setTextoAVoz] = useState("")



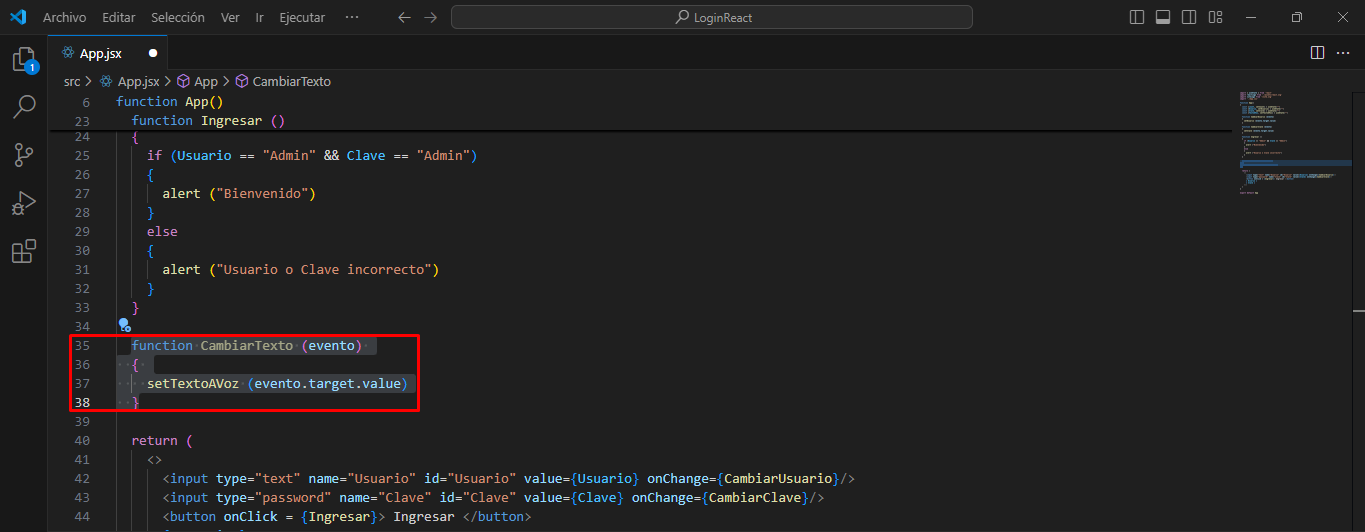
Ahora voy a crear una función:

function CambiarTexto (evento)

{

setTextoAVoz (evento.target.value)

}



Creamos otra función:

function ConvertirTextoAVoz ()

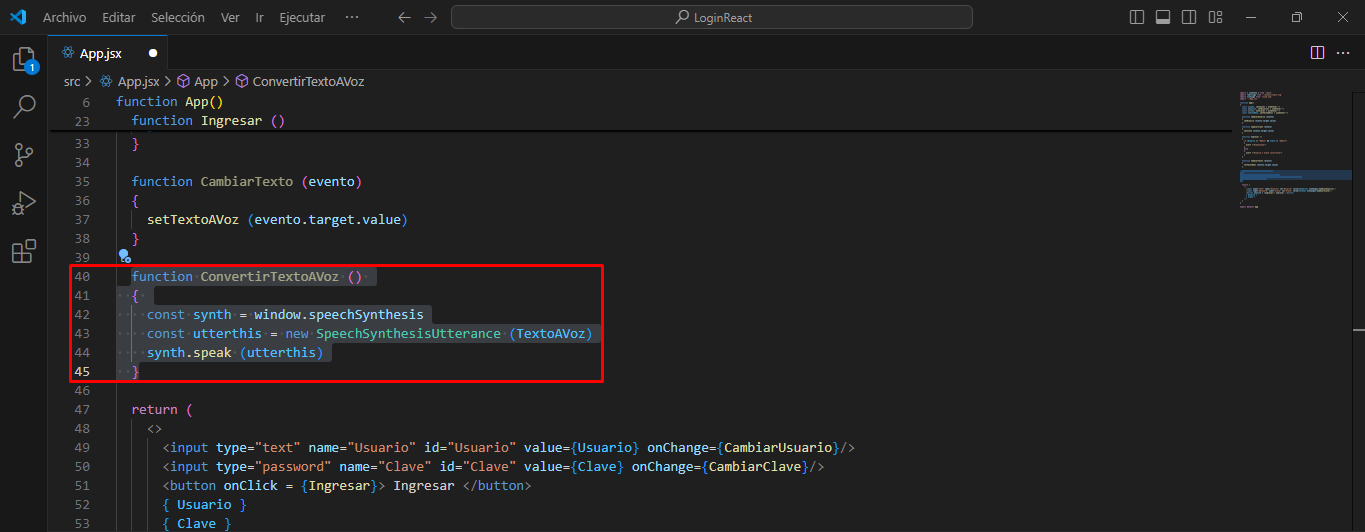
{

const synth = window.speechSynthesis

const utterthis = new SpeechSynthesisUtterance (TextoAVoz)

synth.speak (utterthis)

}

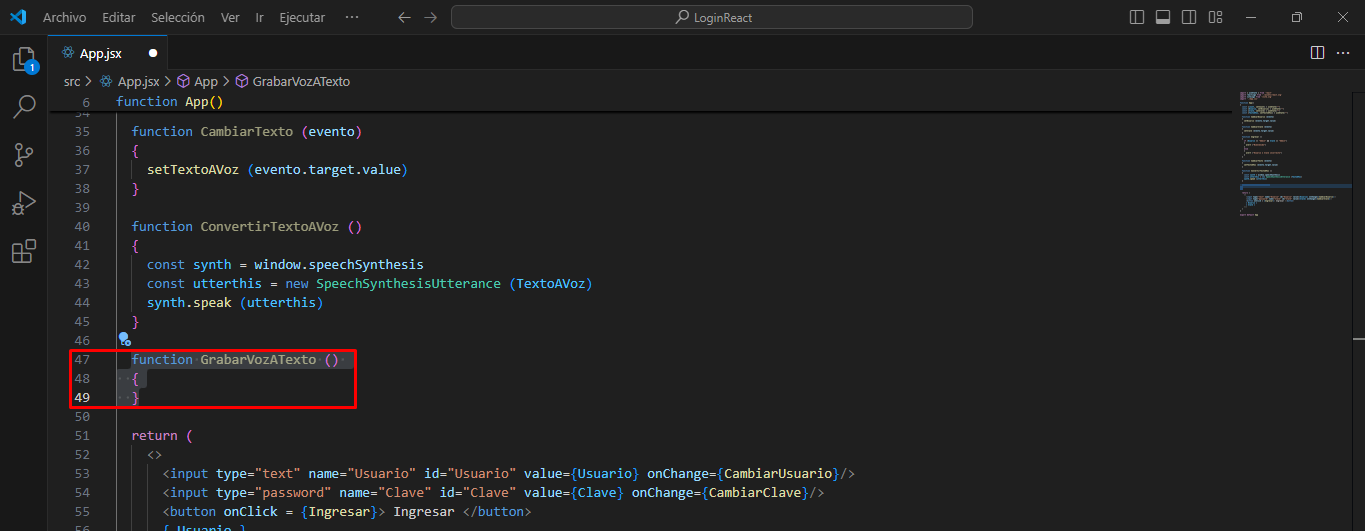


Vamos a crear otra función:

function GrabarVozATexto ()

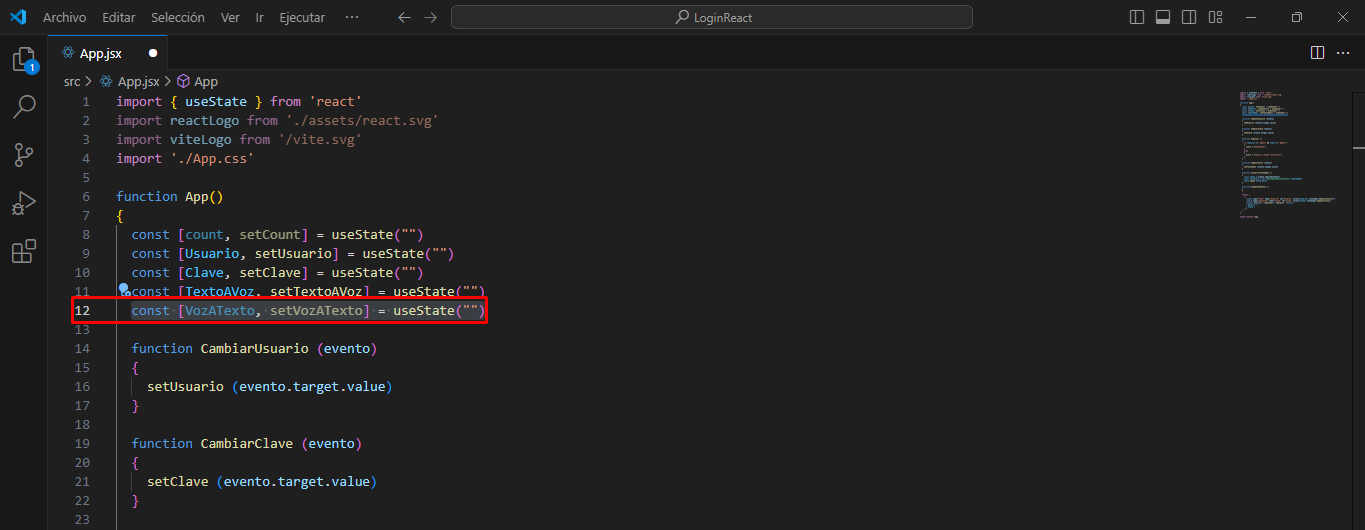
{

}



Ahora vamos a colocar otro estado o constante:

const [VozATexto, setVozATexto] = useState("")



Vamos a colocar el siguiente código en el que ya habíamos creado (Grabar VozATexto):

function GrabarVozATexto ()

{

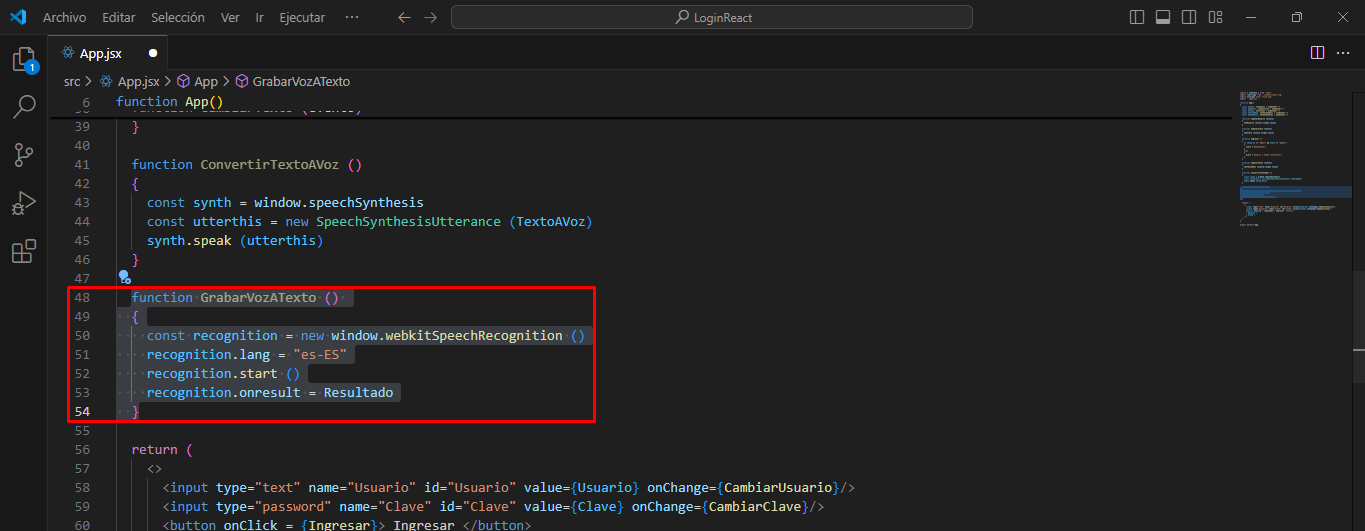
const recognition = new window.webkitSpeechRecognition ()

recognition.lang = "es-ES"

recognition.start ()

recognition.onresult = Resultado

}



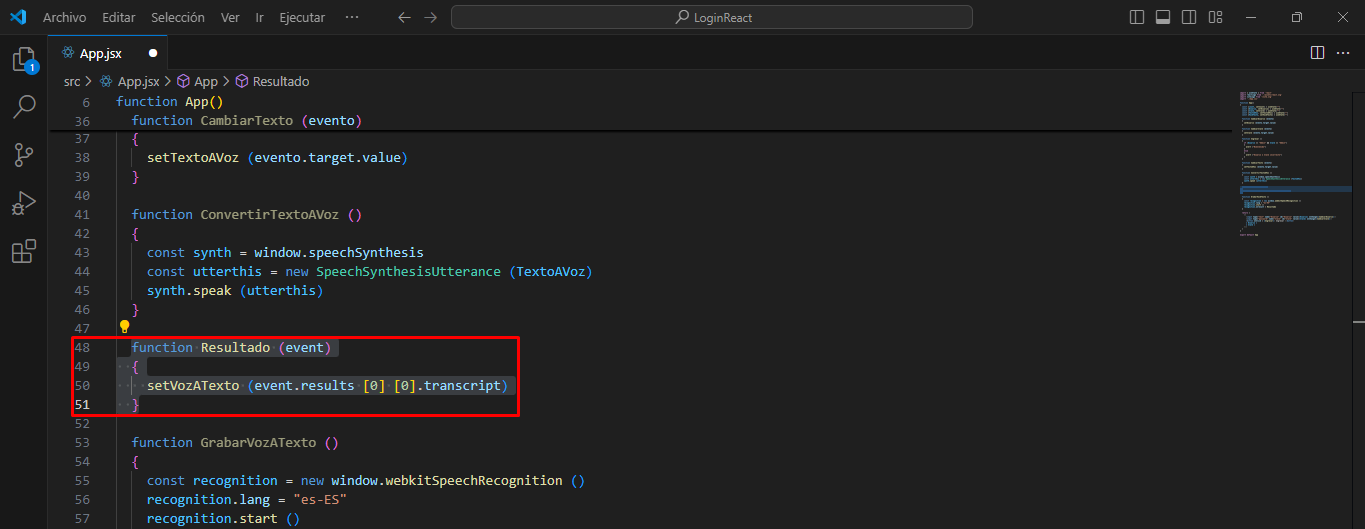
Vamos a crear encima de esta, otra función:

function Resultado (event)

{

setVozATexto (event.results [0] [0].transcript)

}



Ahora vamos a agregar el siguiente código:

<br />

<h3> Conversor de texto a voz </h3>

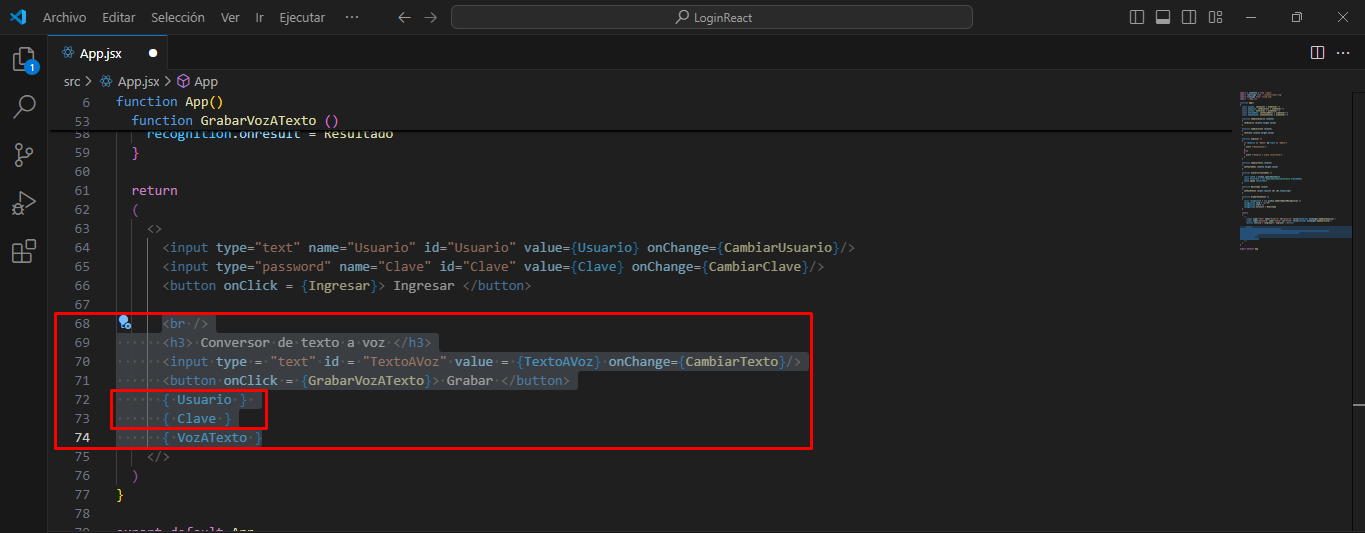
<input type = "text" id = "TextoAVoz" value = {TextoAVoz} onChange={CambiarTexto}/>

<button onClick = {GrabarVozATexto}> Grabar </button>

{ Usuario }

{ Clave }

{ VozATexto }

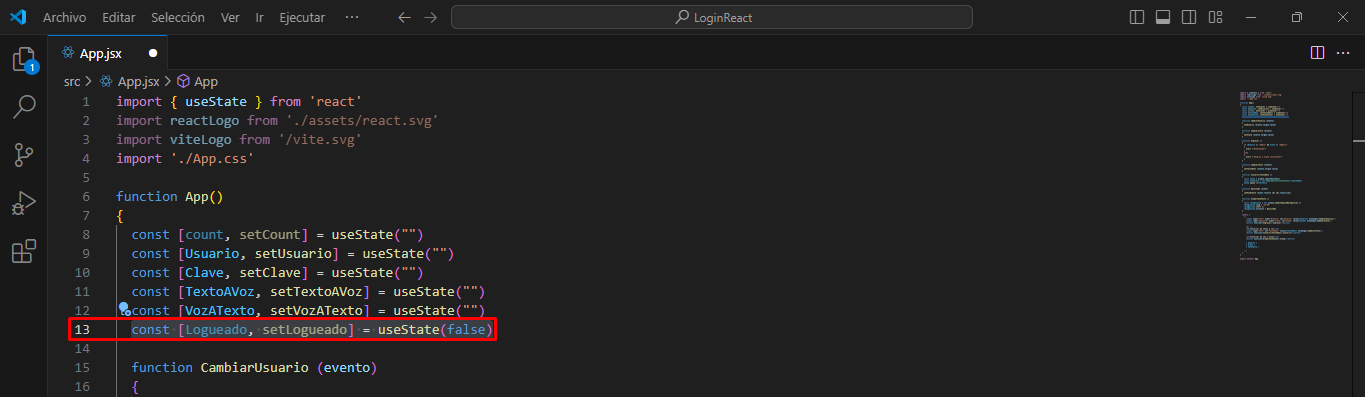


Grabamos y vamos a ver nuestra página. Hay un error y consiste en que no se debió bajar una llave.



Ahora vamos a crear otro estado

const [Logueado, setLogueado] = useState(false)



Ahora vamos a realizar un cambio en el código:

return (

<>

{Logueado ? (

<>

<br />

<h3>Conversor de texto a voz</h3>

<input type="text" id="TextoAVoz" value={TextoAVoz} onChange={CambiarTexto}/>

<button onClick={ConvertirTextoAVoz}>Convertir</button>

<h3>Conversor de voz a texto</h3>

<button onClick={GrabarVozATexto}>Grabar</button>

{ Usuario }

{ Clave }

{ VozATexto }

</>

) : (<>

<input type="text" name="Usuario" id="Usuario" value={Usuario} onChange={CambiarUsuario}/>

<input type="password" name="Clave" id="Clave" value={Clave} onChange={CambiarClave}/>

<button onClick={Ingresar}>Ingresar</button>

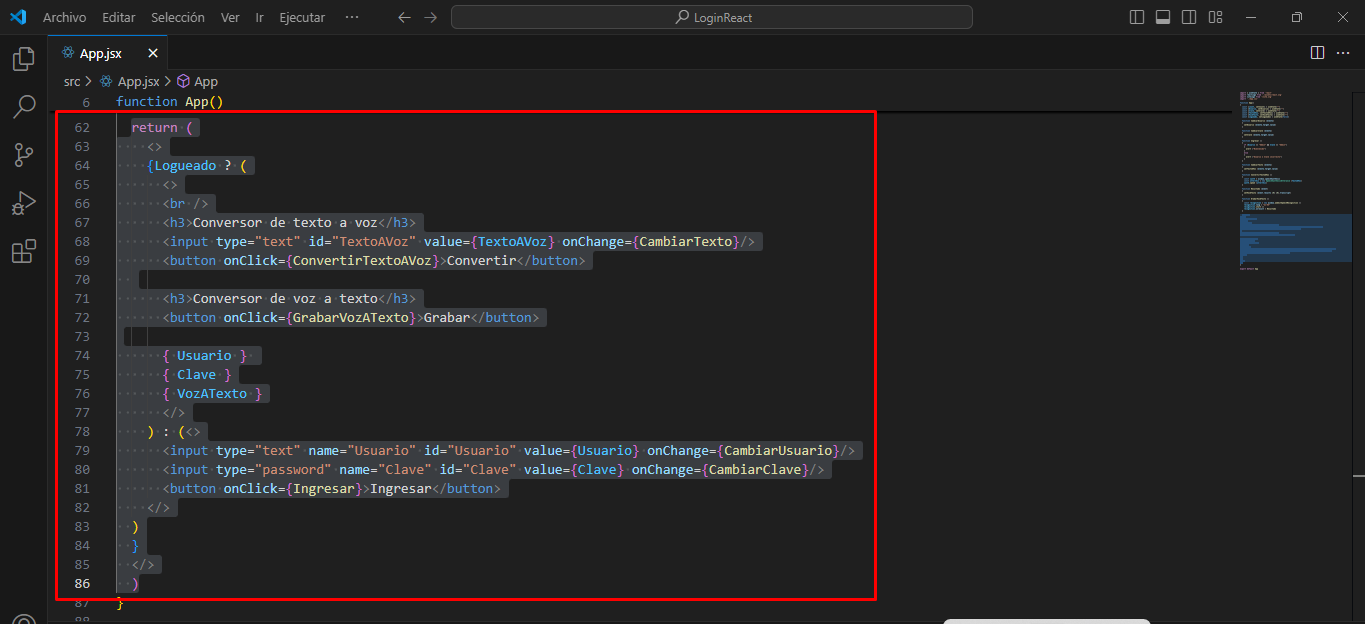
</>

)

}

</>

)



Ahora vamos a realizar otra modificación

}

return (

<>

<input type="text" name="Usuario" id="Usuario" value={Usuario} onChange={CambiarUsuario}/>

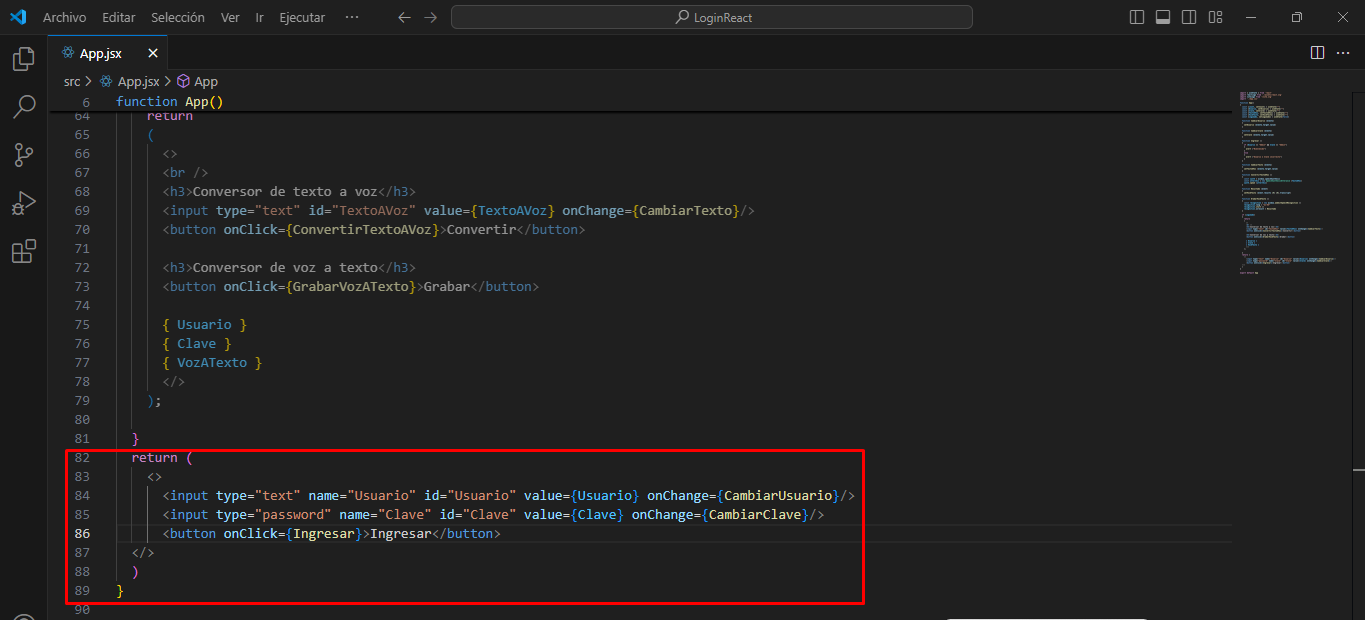
<input type="password" name="Clave" id="Clave" value={Clave} onChange={CambiarClave}/>

<button onClick={Ingresar}>Ingresar</button>

</>

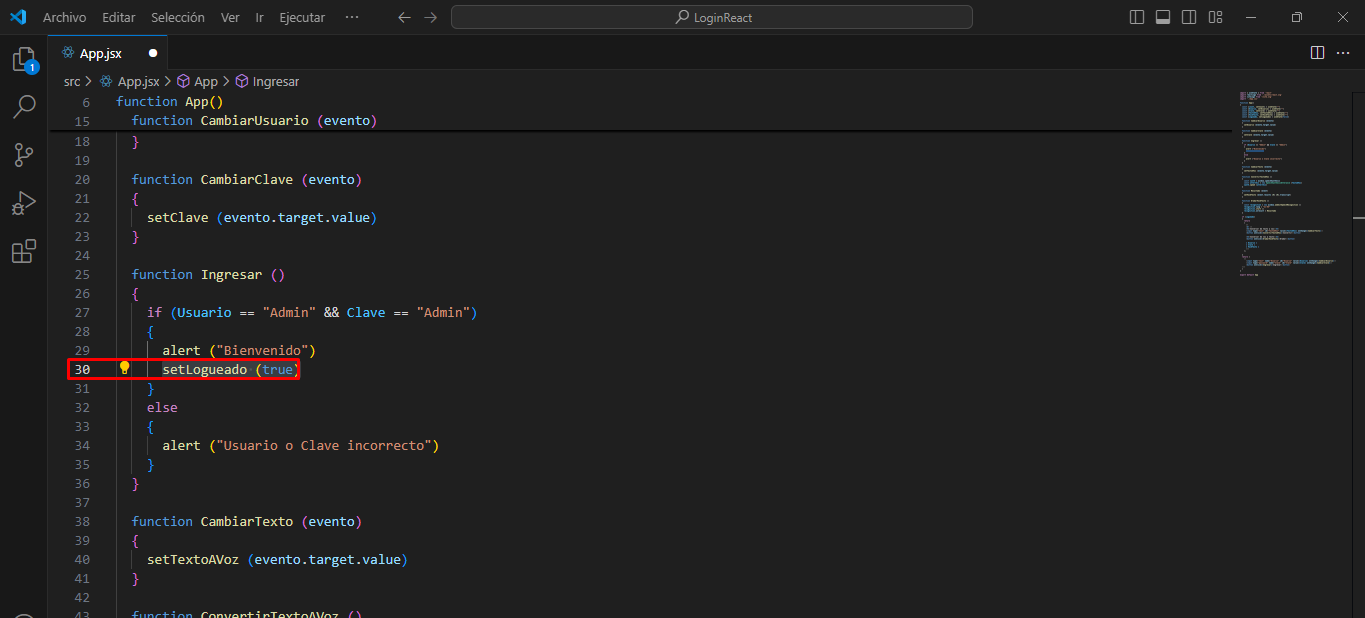
)

}



Ahora vamos a realizar otra modificación

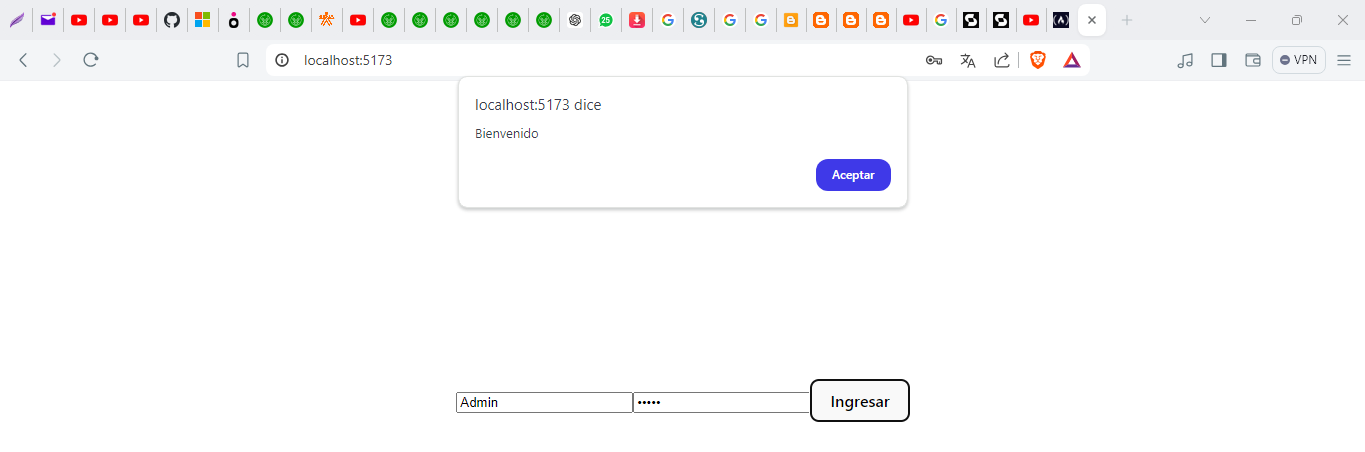
setLogueado (true)



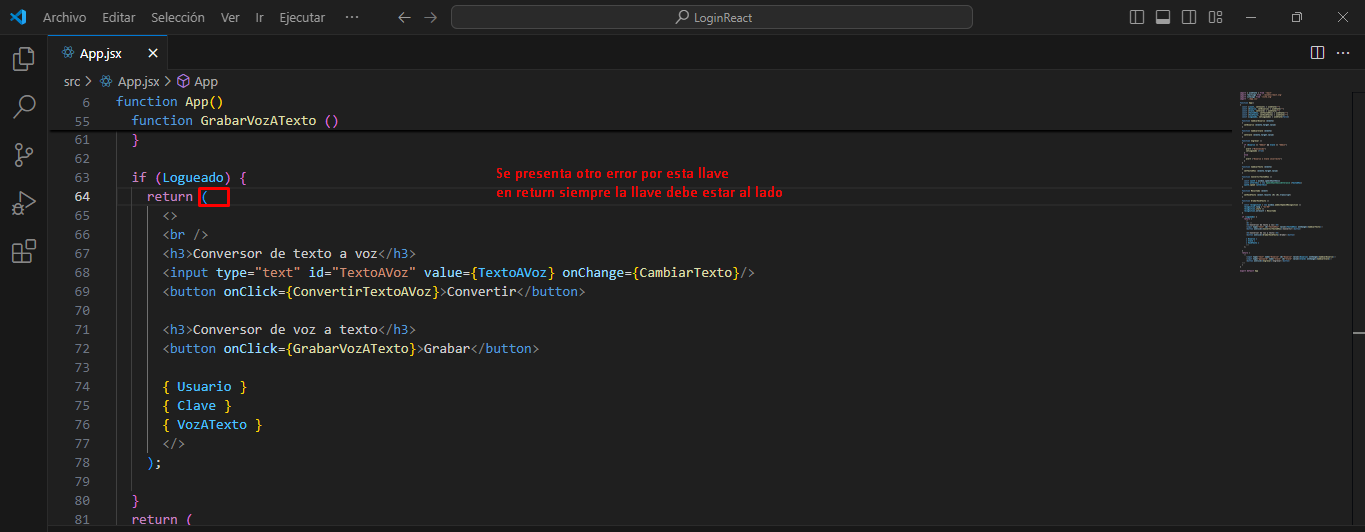
Si ejecuto mi programa y están mal las credenciales me aparecerá

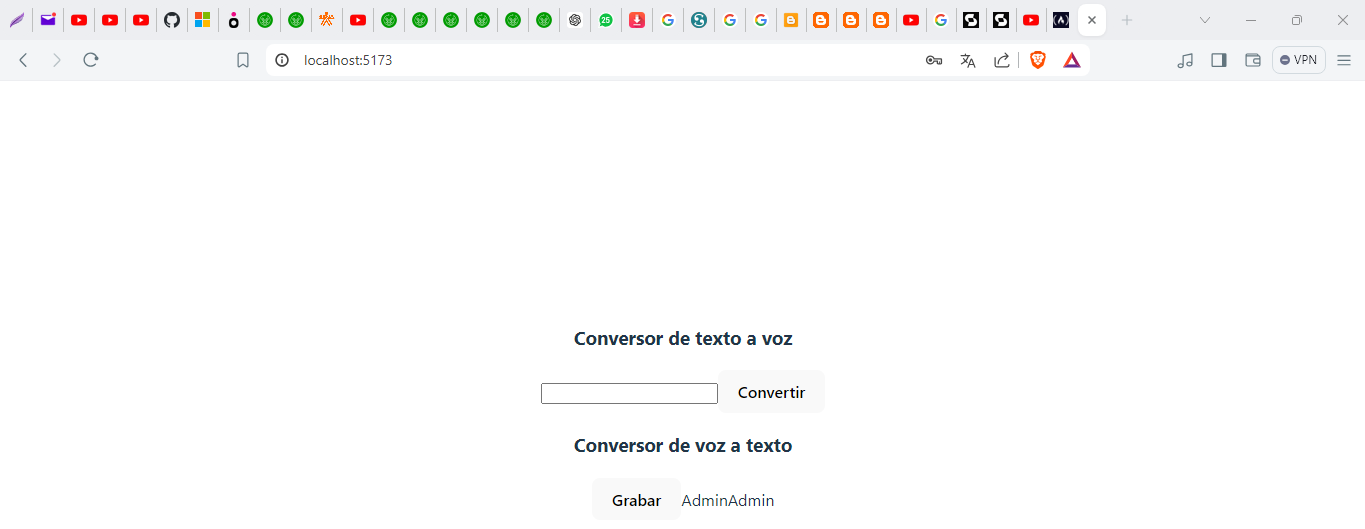


Si están bien aparecerá



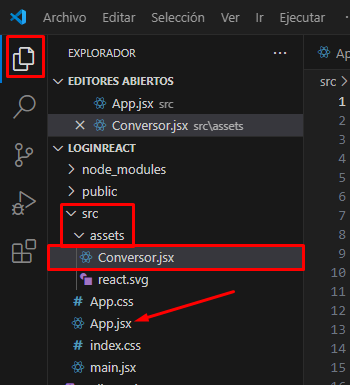
Y al dar aceptar



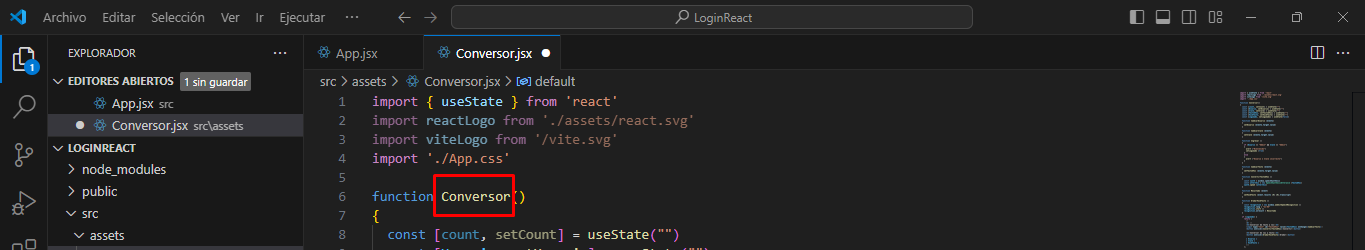


Ahora lo que voy a hacer es mejorar la apariencia de mi aplicación.

Y lo que voy a hacer es duplicar mi react y al nuevo lo voy a nombrar **converter**

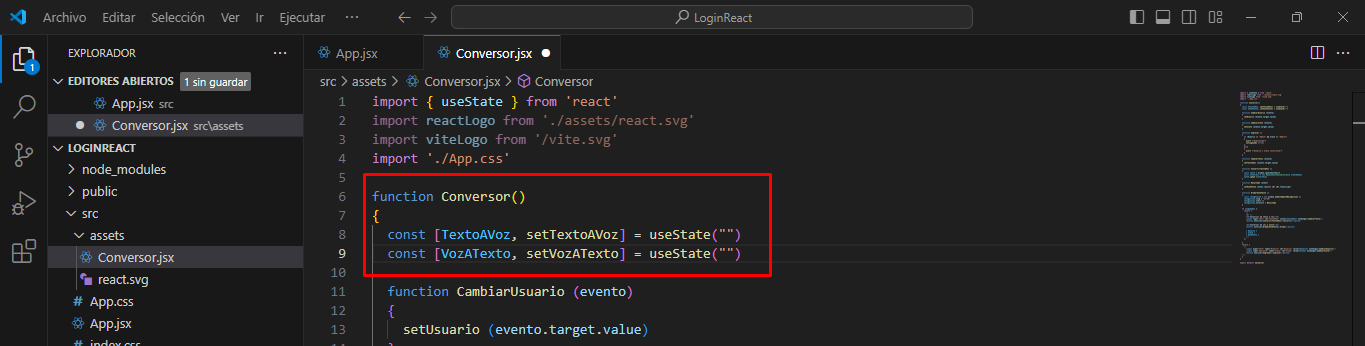


Ahora voy a cambiar la palabra app por conversor, al inicio y al final, pero al incicio entre parentetesis le voy a colocar: { Usuario, Clave }



Ahora voy a borrar varia información



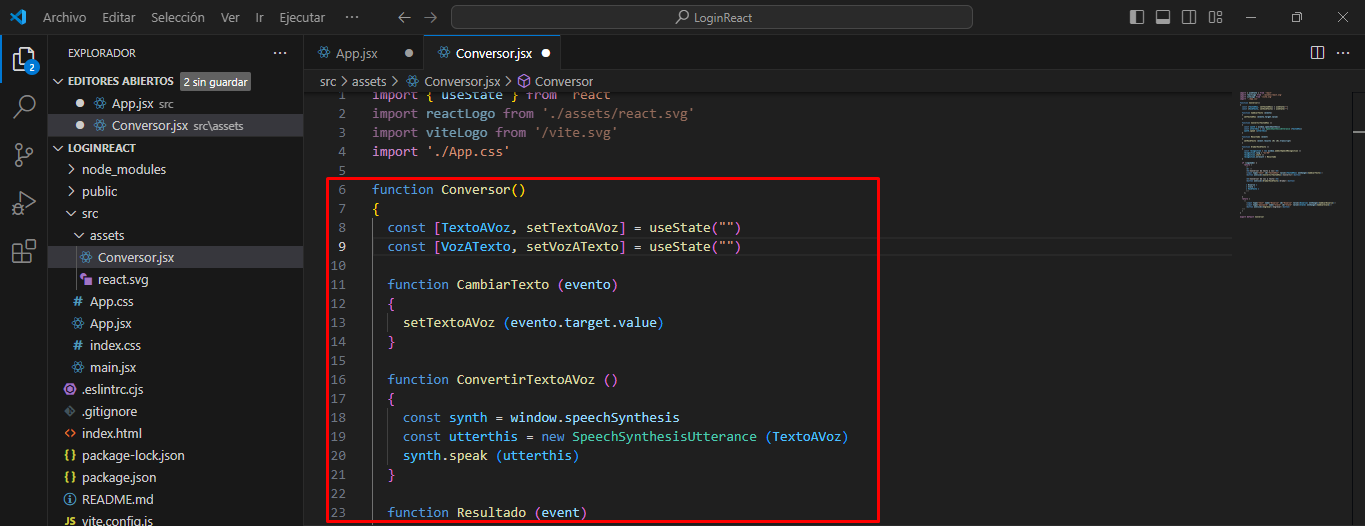


También voy a eliminar varias funciones como

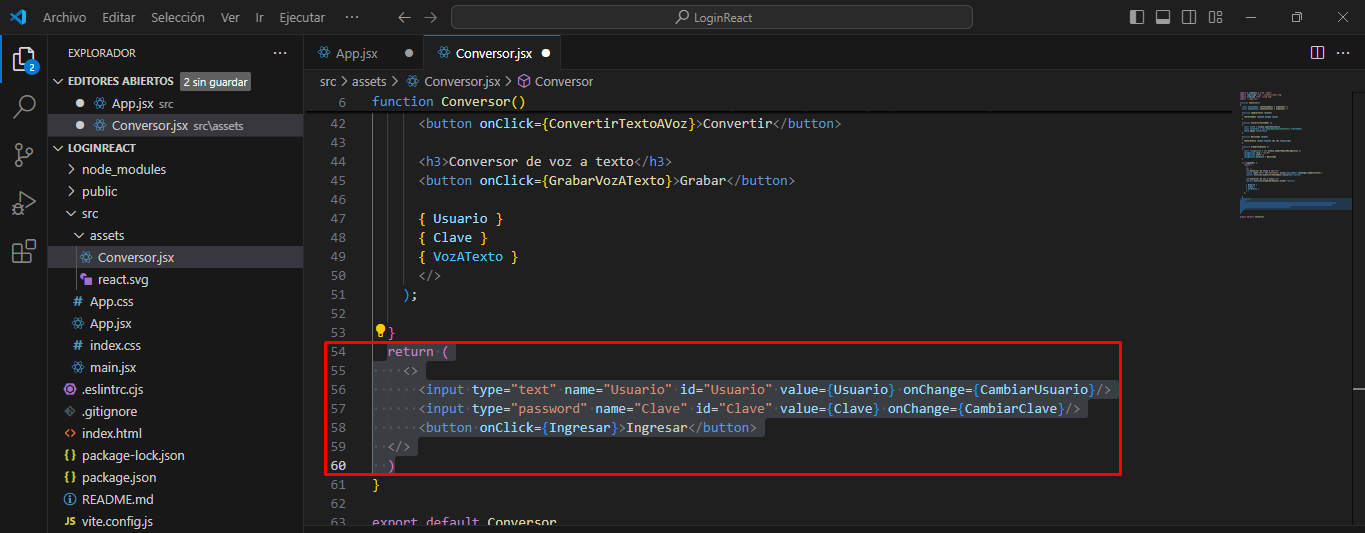
AambiarUsuario

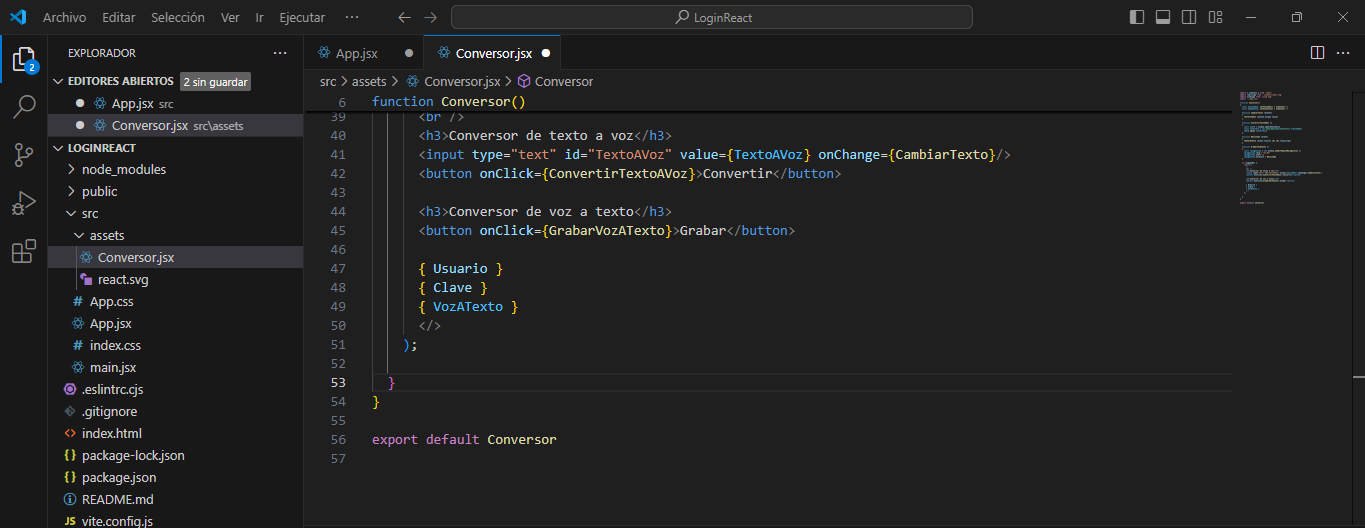
CambiarClave

Ingresar

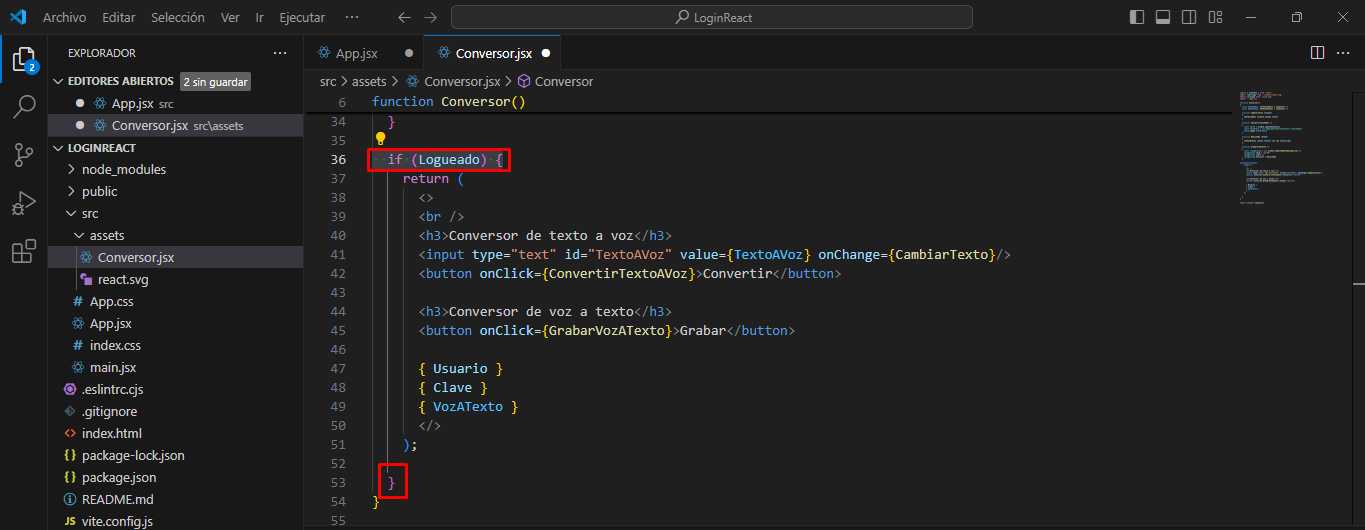


Voy a quitar el ultimo return que es el de inicio de sesión





Voy a quitar el if loqueado y su lave de cierre



De esta manera solo queda el código de voz a texto, de texto a voz, resultado y grabar.

Ahora necesitamos unir mi nuevo código, por lo que voy a mi código principal donde está todo y allí voy a eliminar todo lo que tenga que ver con el conversor.



Vamos a quitar las funciones

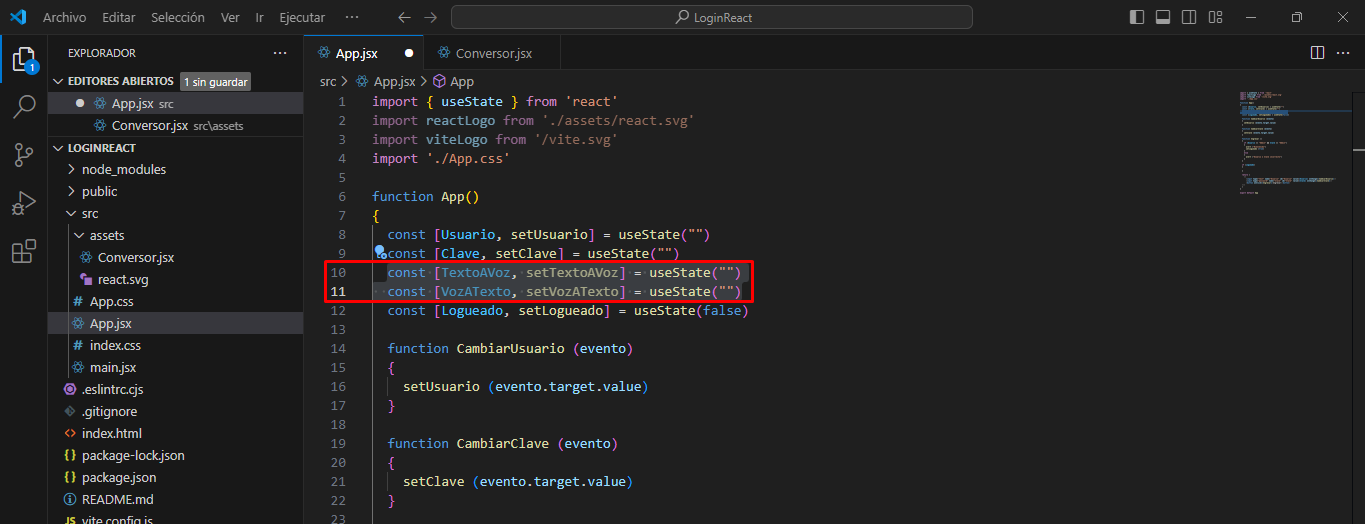
GrabarVozATexto

Resultado

ConvertirTextoAVoz

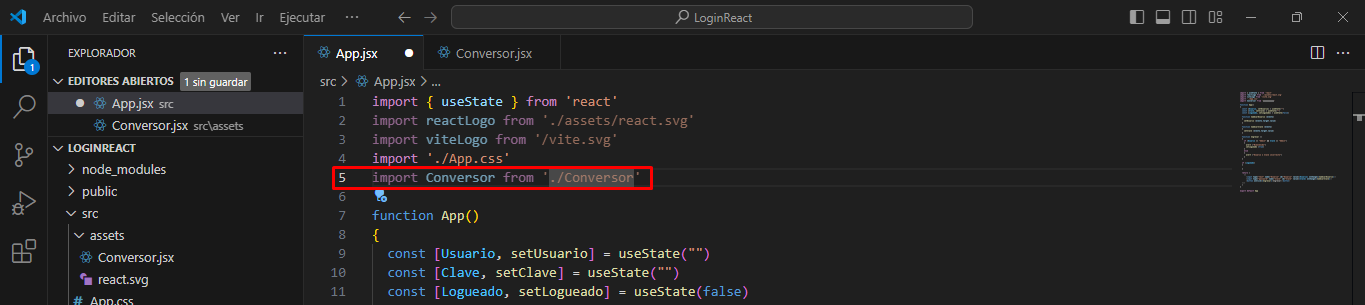
CambiarTexto

También quitamos los dos estados



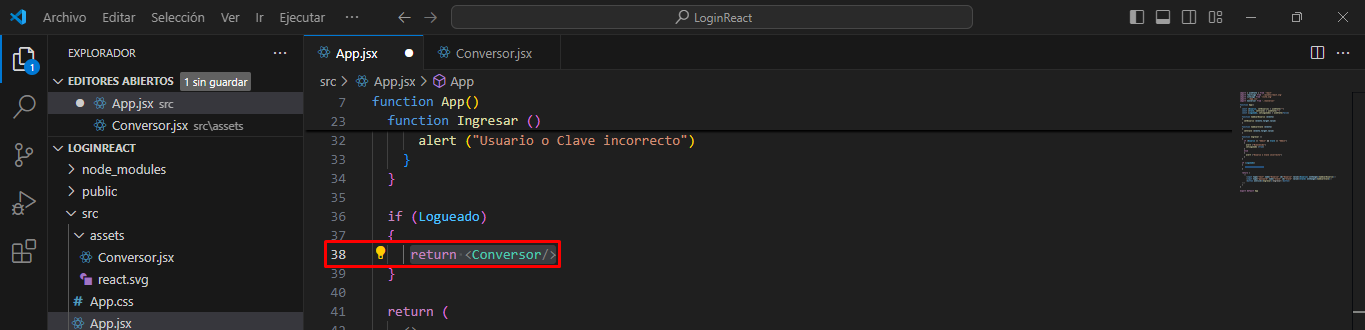
Ahora en el if de logueado que dejamos lo que podemos hacer es invocar el archivo de conversor:

import Conversor from './Converter'



Y en mi if colocar:

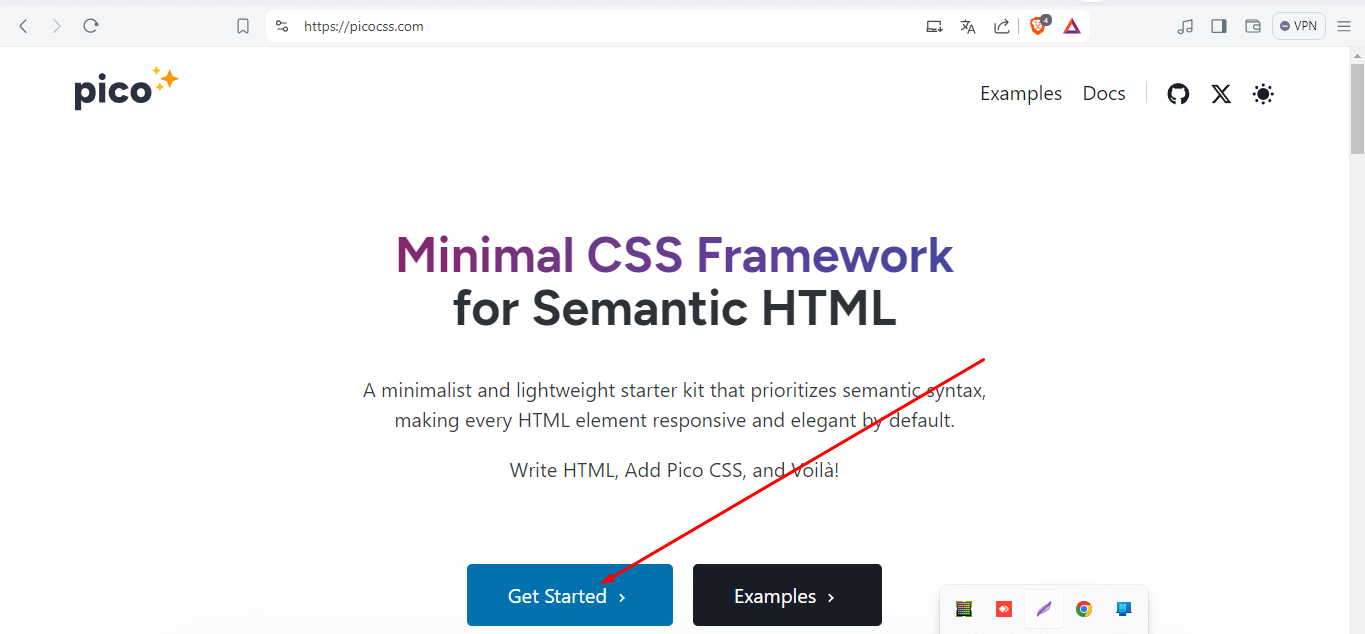
return <Conversor/>



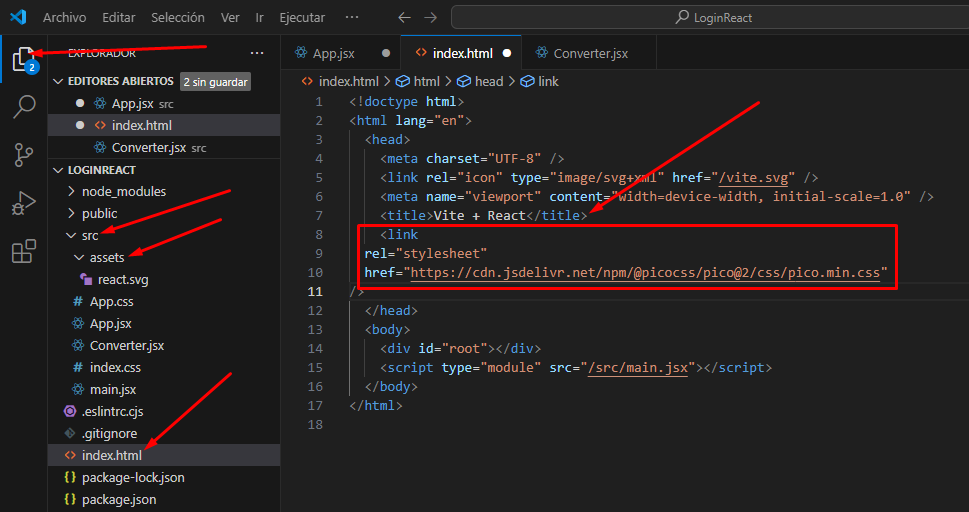
De esta manera si probamos de nuevo vamos a poder ver que funciona igual.

Ahora vamos a mejorar el Login, para lo cual vamos a actualizar el código de la siguiente manera.

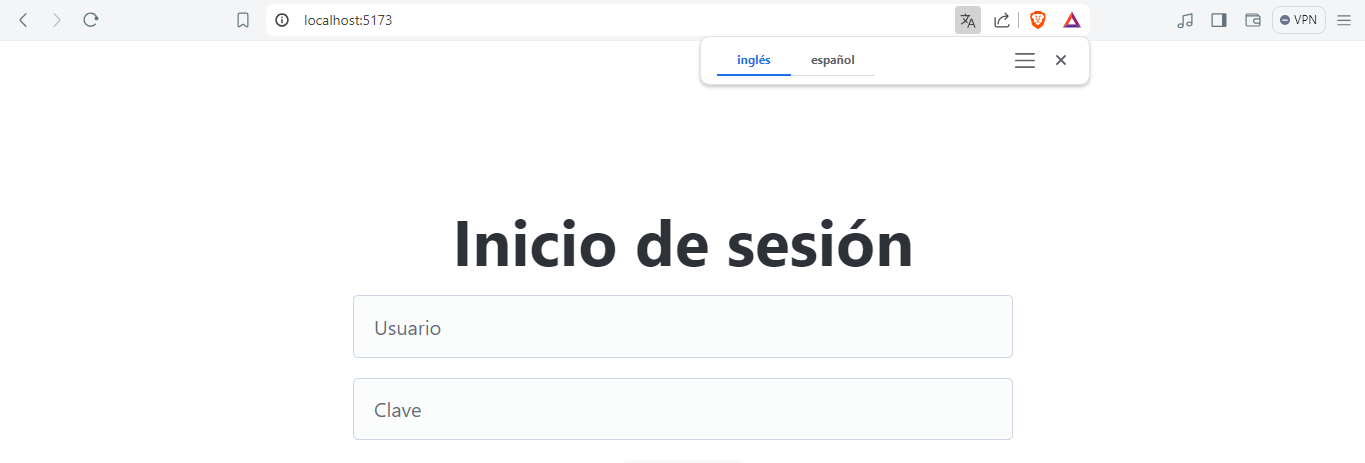
Para ello vamos a picoscss



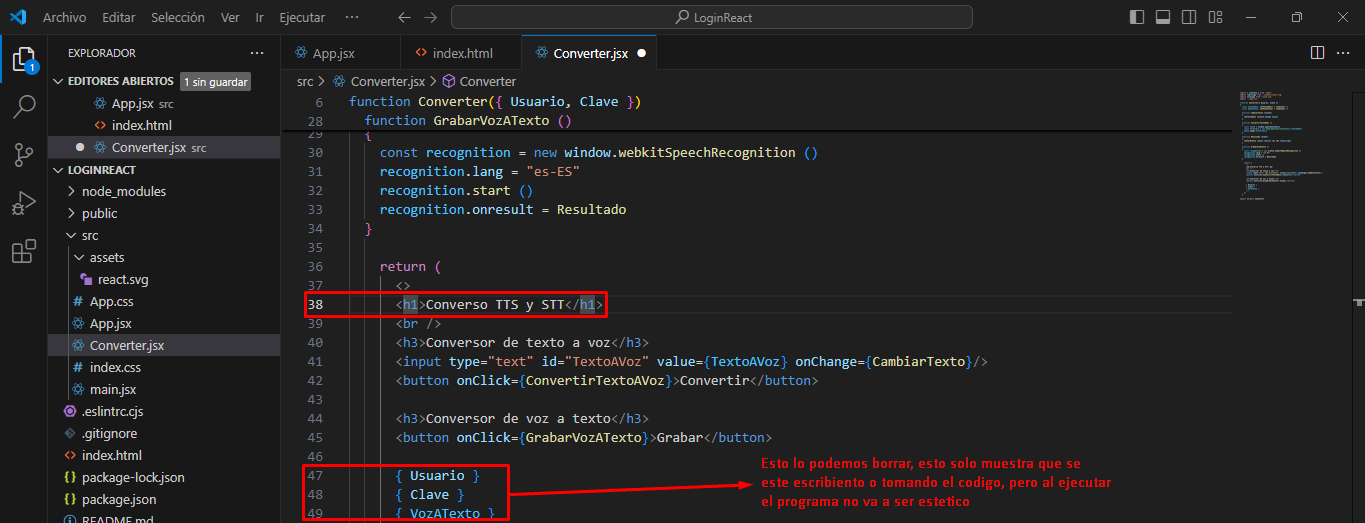




Guardamos y podemos ver el cambio



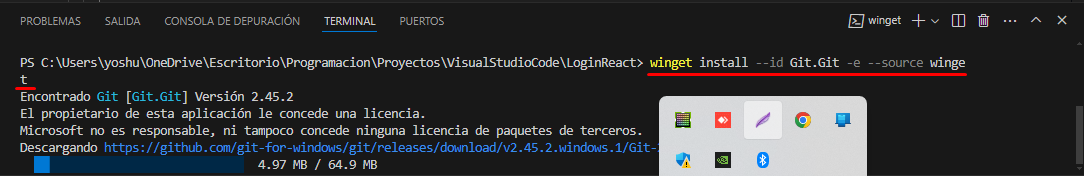
Ahora voy a agregar un titulo a mi pestaña convert



Ahora vamos a subir nuestro proyecto a github.

Si no tenemos git en nuestro VSC vamos a escribir el siguiente codigo en una nueva terminal

winget install --id Git.Git -e --source winget



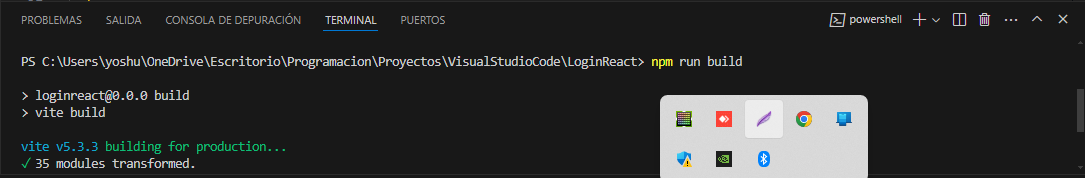
Este comando lo que hace es instalar el paquete en mi PC

Luego de la instalación me va a abrir git hub y me va a pedir permisos y luego me va a crear el repositorio.

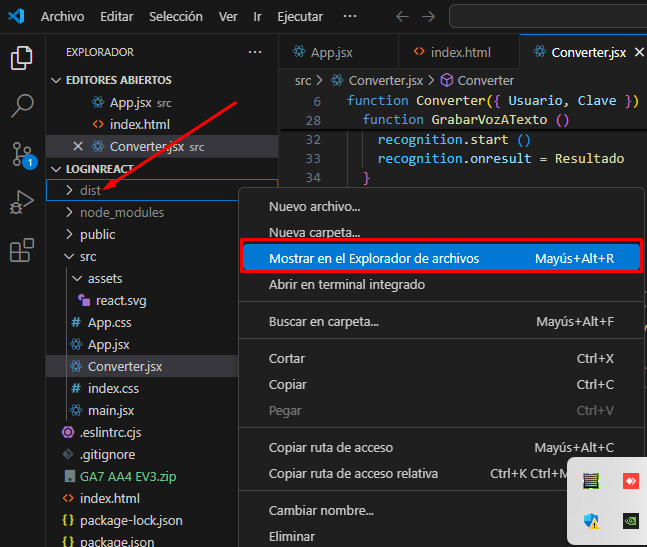
Ahora lo que vamos a hacer es publicar mi proyecto en la web.

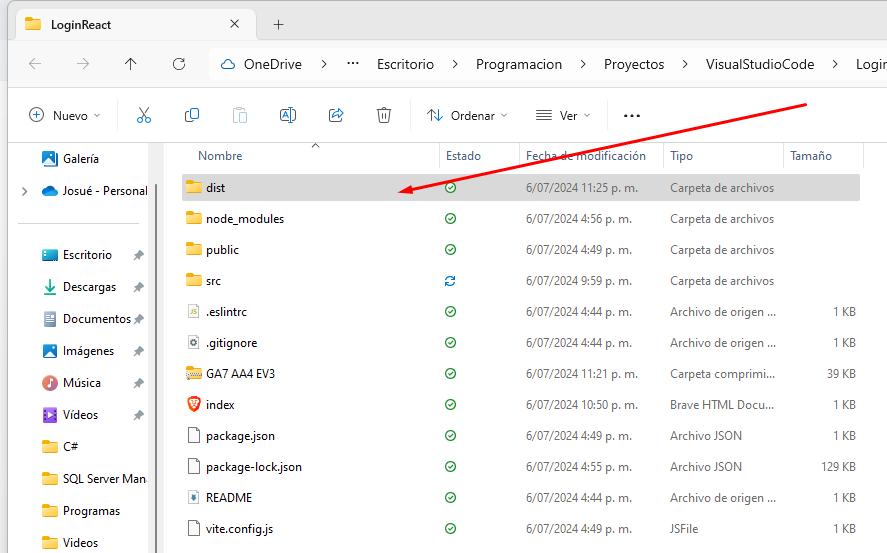
Para este procedimeinto abro una nueva termianl y escribo el código

npm run build



Esta me va a compilar todo mi proyecto para poderlo publicar en la web.





Después de ello me dirijo a netlufy drop y arrastro la carpeta de “dist”

