**Login Xpress**

<https://www.youtube.com/watch?v=53lYOIbJslI&list=PLAXdMOxCmqPvEukZ1iB_yl7oB6yHZIJBW&index=1>

Requisitos:

1. Node JS
2. Visual Studio Code
3. Xampp

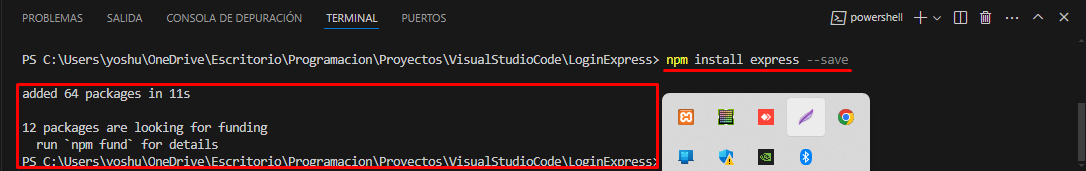
Complementos:

1. Xpress.js (<https://expressjs.com/>)
2. Mysql2 (<https://www.npmjs.com/package/mysql2>) conector de DB
3. Npm (<https://www.npmjs.com/package/cors>)
4. Sesion xpress (https://www.npmjs.com/package/express-session)

Paso a paso:

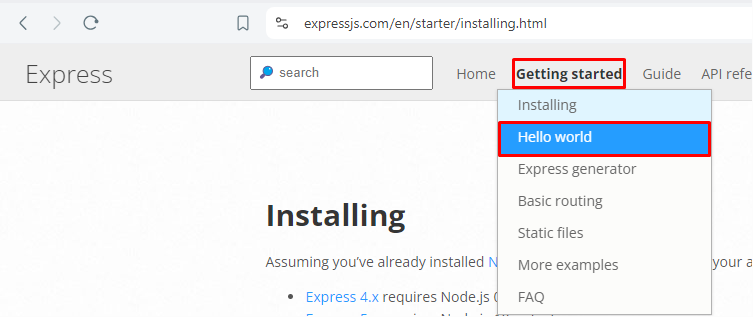
Crear una carpeta para mi proyecto y abrirla con visual studio code, pero también debo instalar el xpress.js en su terminal con el siguiente código:

npm install express –save



Esta es una librería o un proyecto que otras personas desarrollaron y yo solo voy a reutilizar su código.

Ahora voy a escribir el código para hacer uso de el, lo copiamos desde la pagina de xpress.



Copiamos el código y lo llevamos a VSC

const express = require('express')

const app = express()

const port = 3000

app.get('/', (req, res) => {

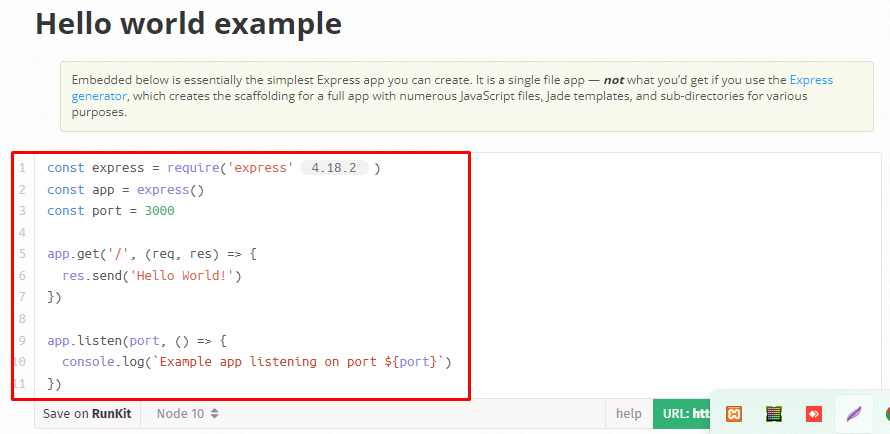
res.send('Hello World!')

})

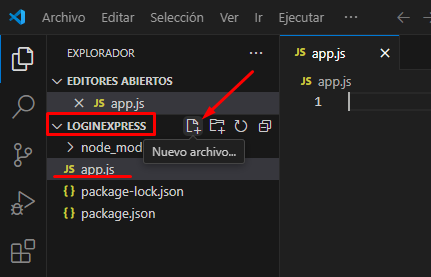
app.listen(port, () => {

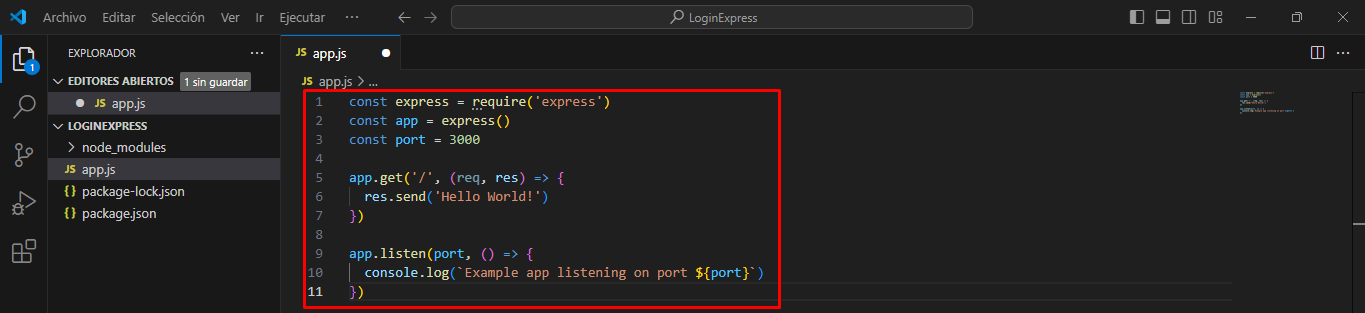
console.log(`Example app listening on port ${port}`)

})



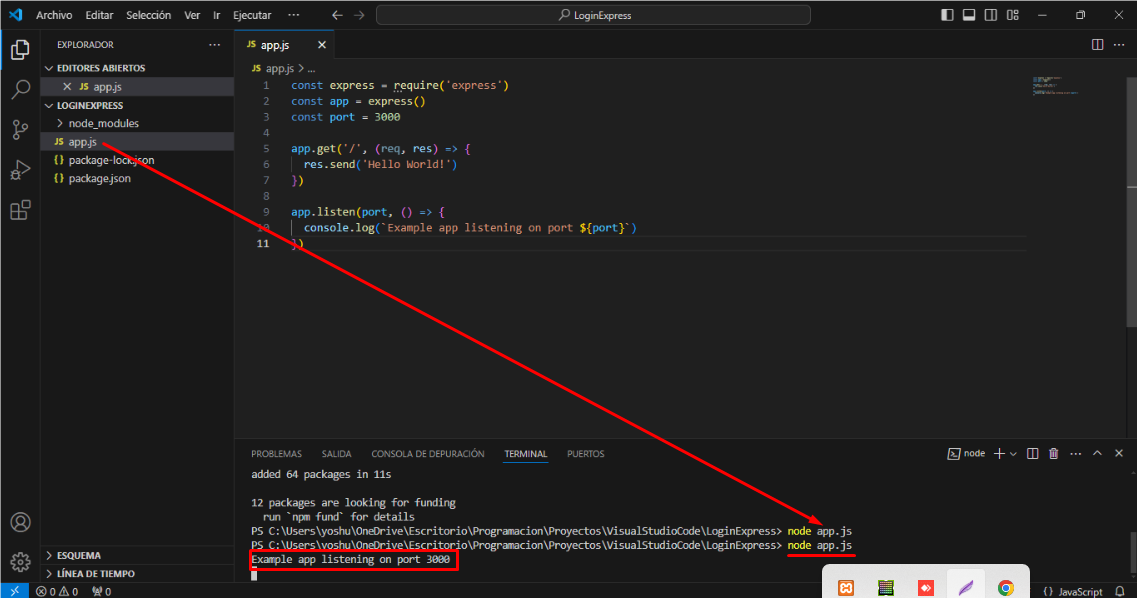
Creamos un archivo en VSC para pegar el código del xpress y lo nombramos app.js





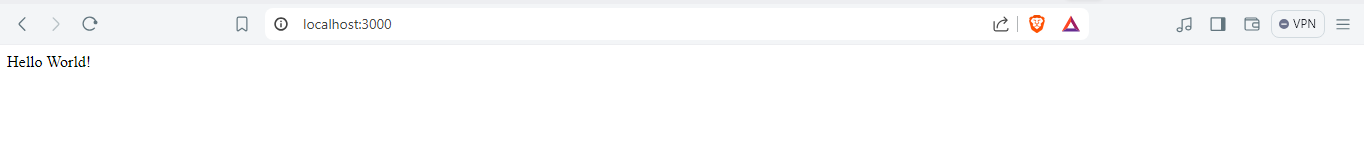
Y voy a correr el código por medio de la terminal y del siguiente código, pero antes debo haber guardado los cambios:

node app.js



Este me indica que ya se creo el archivo en el puerto 3000, por lo que voy a mi buscador y coloco en su buscador:

Localhost:3000



Ahora voy a duplicar mi sentencia de hellow Word y las voy a modificar.



Ahora cada vez que quiera ver los cambios debo ejecutar el código:

node app.js

pero va a ser muy tedioso por lo que voy a indicarle que automáticamente me realice dichos cambios por medio del siguiente código:

node -- watch .\app.js

si no funciona vamos a declarar los siguientes códigos

npm install -g nodemon

luego ejecutamos:

nodemon .\app.js

si tampoco funciona vamos a dclarar los siguientes códigos:

npm install -g node-dev

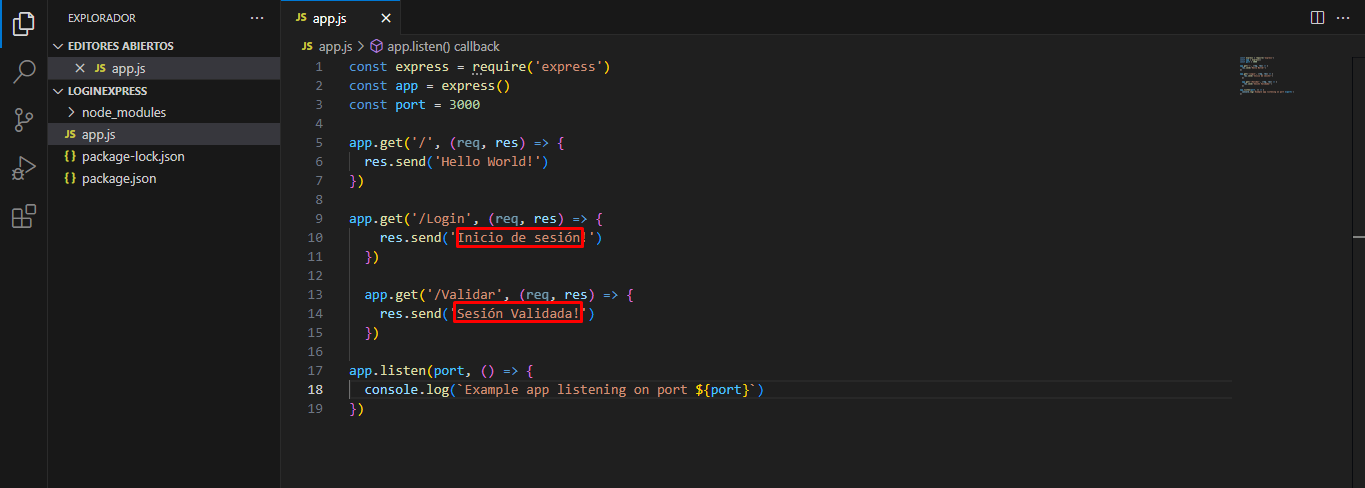
y después:

node-dev .\app.js

si el error persiste debemos ejecutar uno por uno los cambios.

node app.js

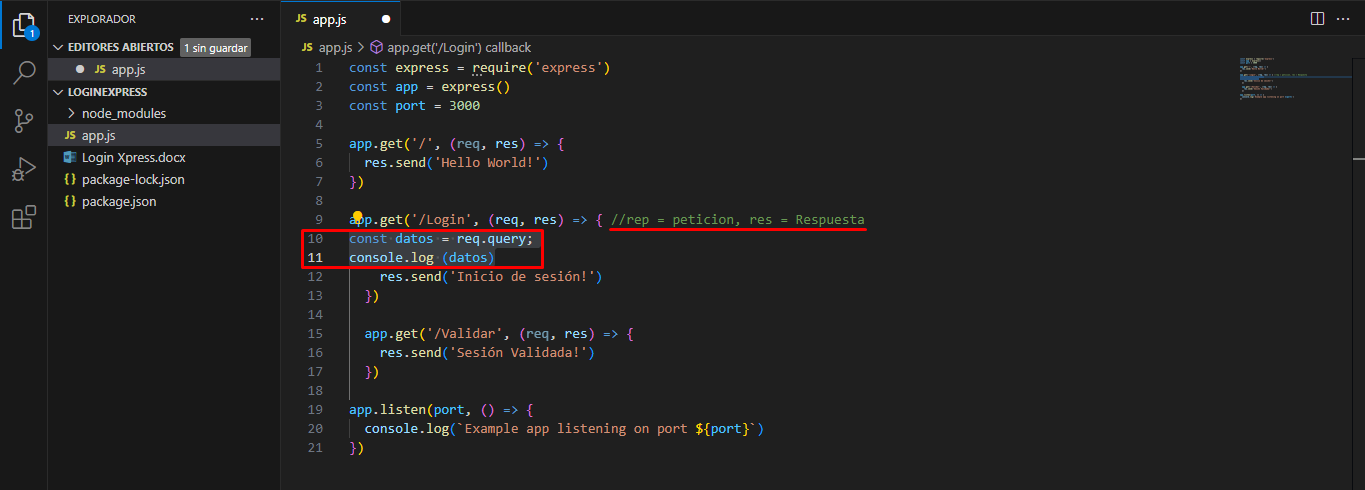
ahora sobre el código vamos a hacerlos siguientes cambios.



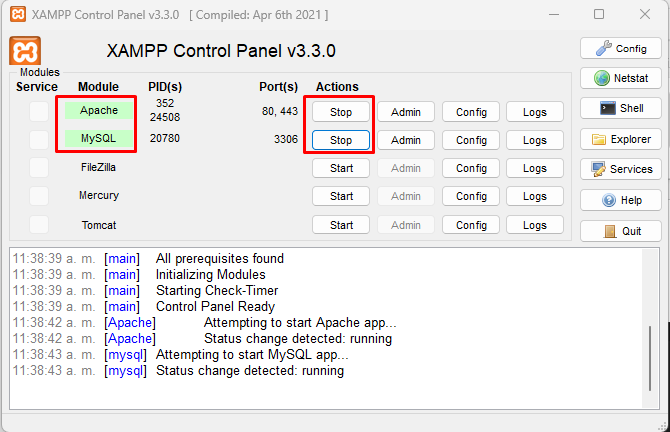
Ahora voy a hacer el siguiente cambio en la línea de codigo de Login dentro de sus llaves:

const datos = req.query;

console.log (datos)

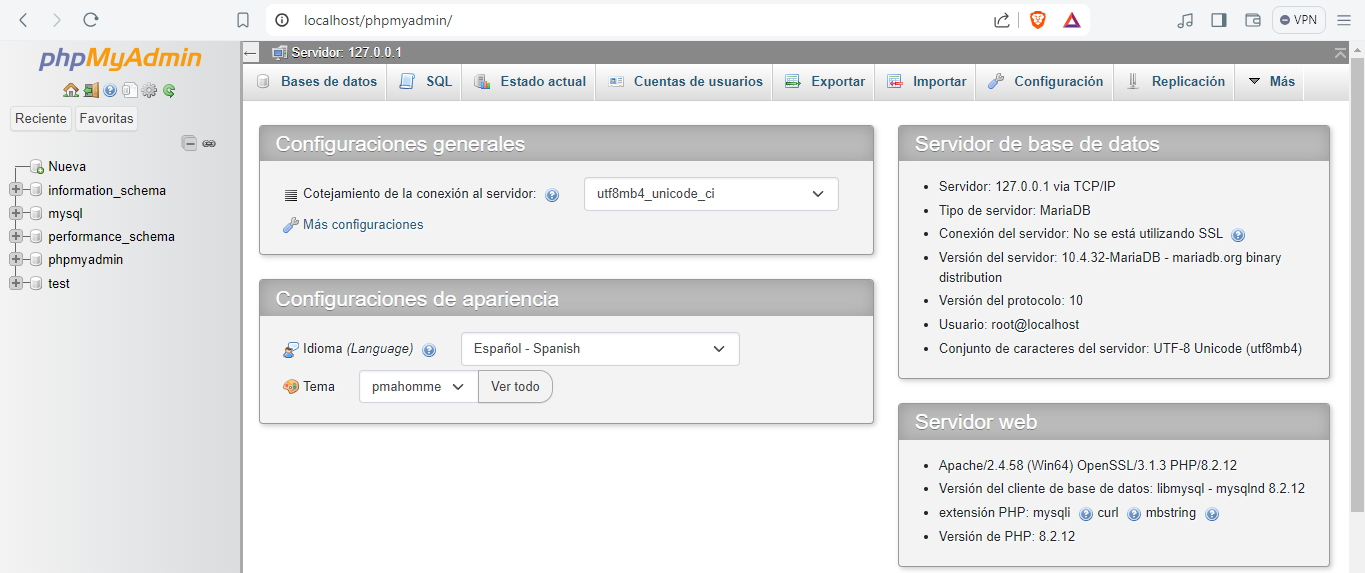


Ahora vamos a crear la base de datos, para ello vamos a abrir xampp y damos en iniciar en apache y sql

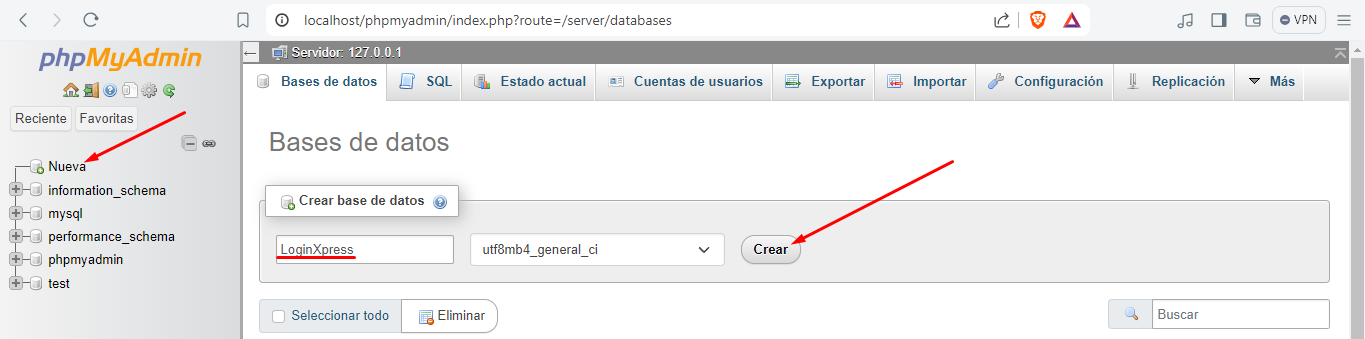


Y vamos a admin y luego vamos a abrir el gestor de base de datos.

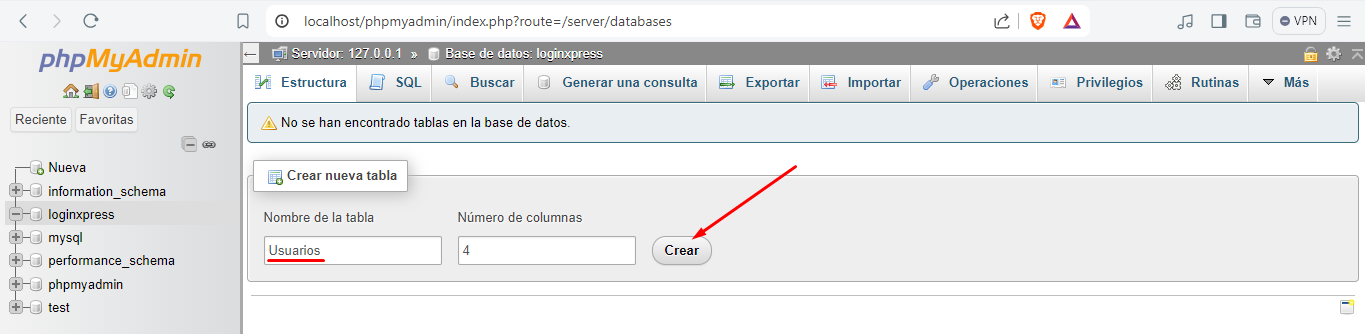




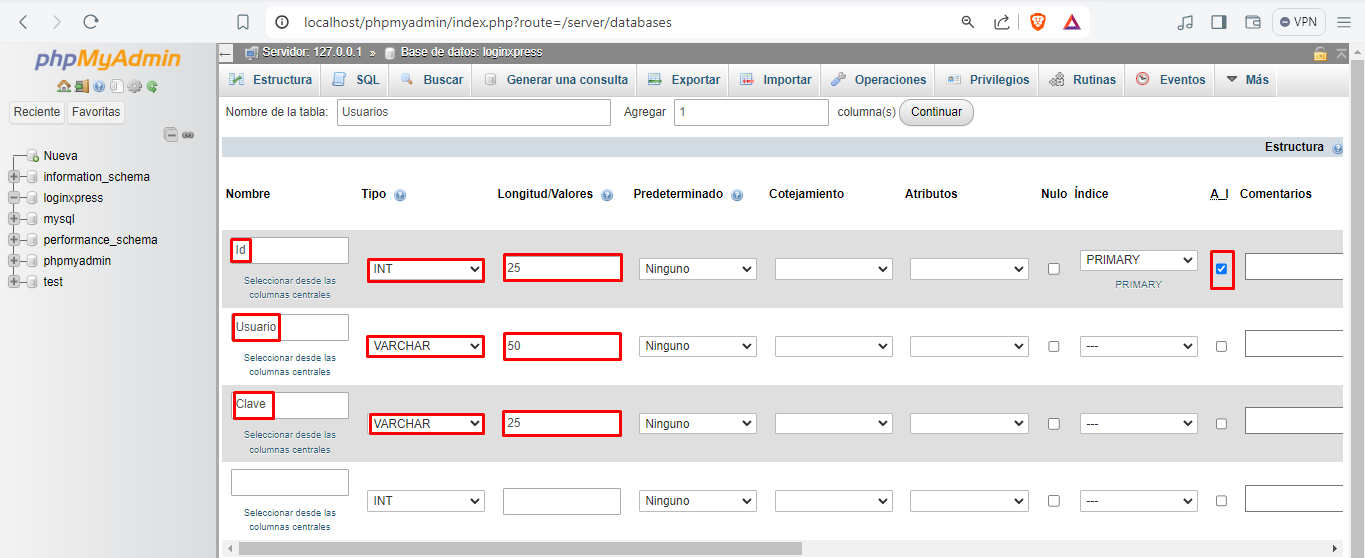
Y desde allí vamos a crear la base de datos.



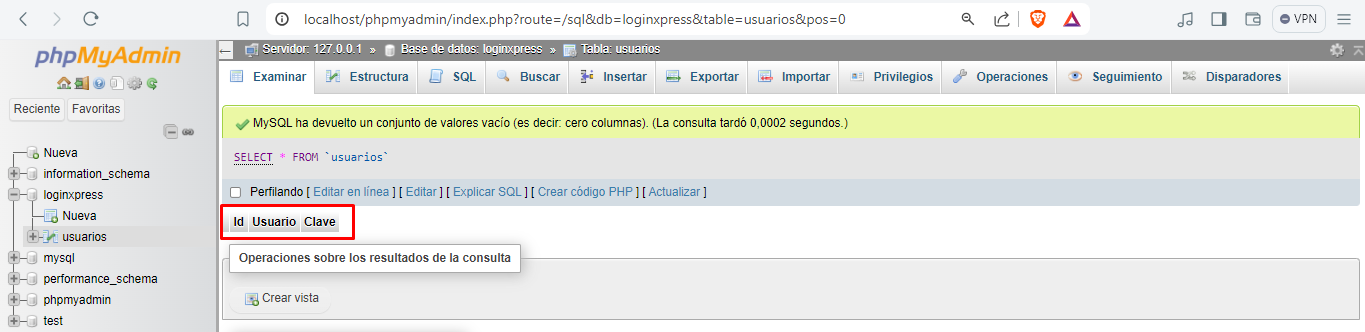
Voy a crear una tabla llamada usuarios



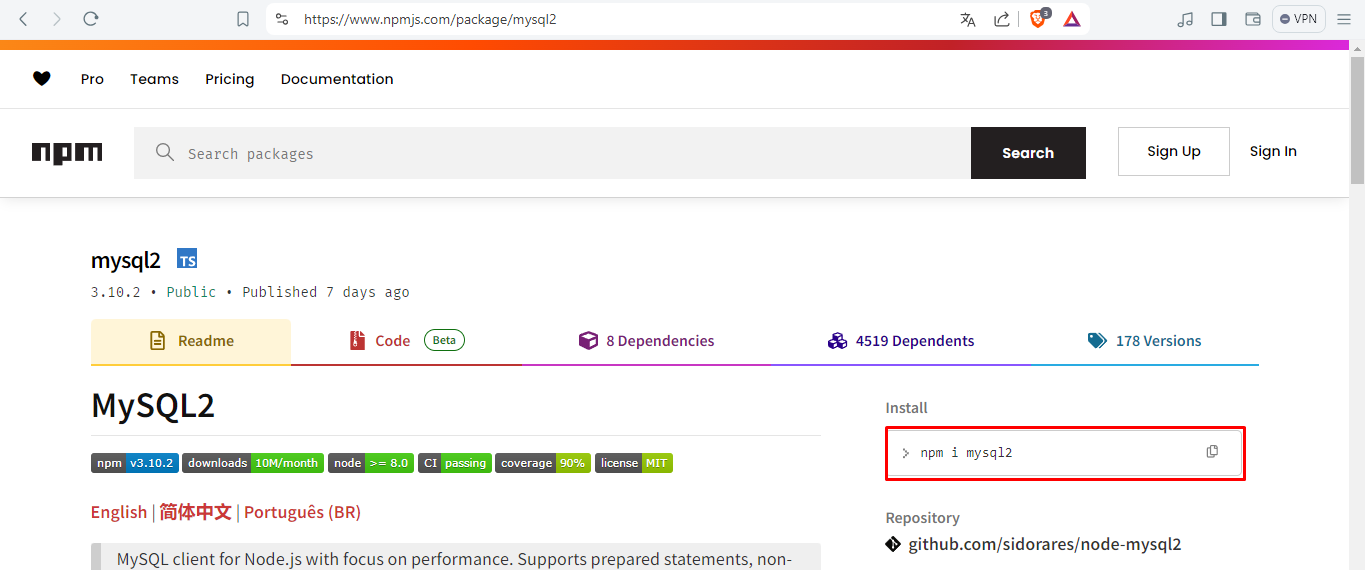
Vamos a ingresar los datos que voy a necesitar en mi DB y al final vamos a dar en guardar

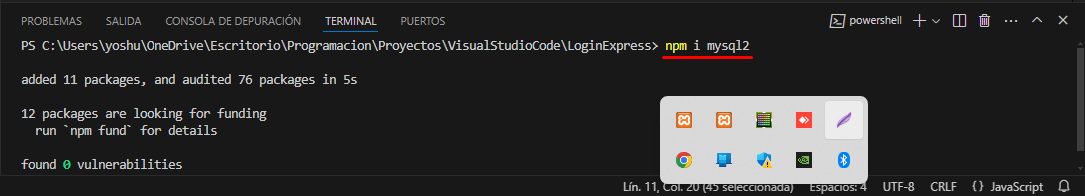


Ya que creamos nuestra base de datos vamos a conectarnos a esta base de datos desde mi VSC

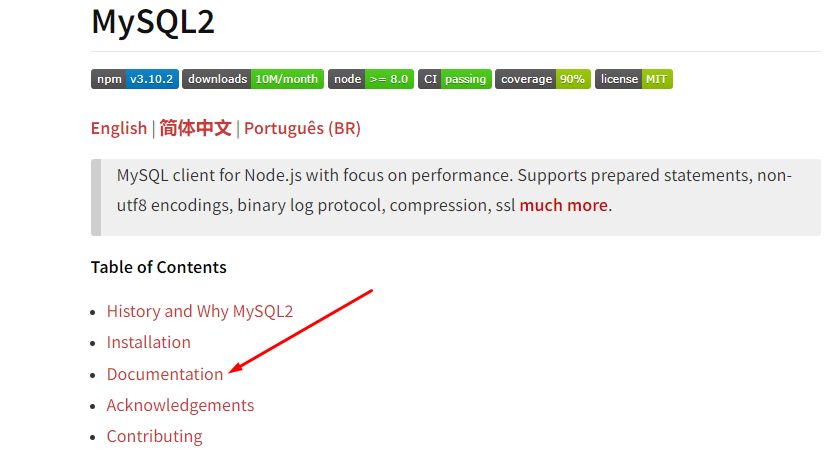


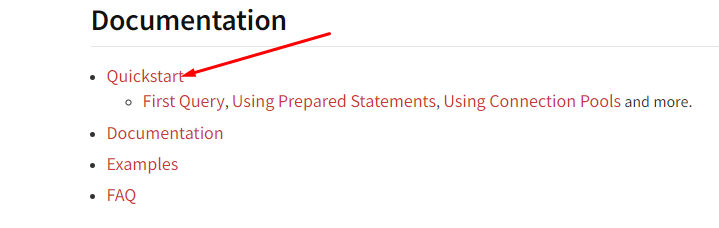
Voy a buscar MYSQL2 y voy a copear el código de instalación y lo voy a pegar en mi terminal.

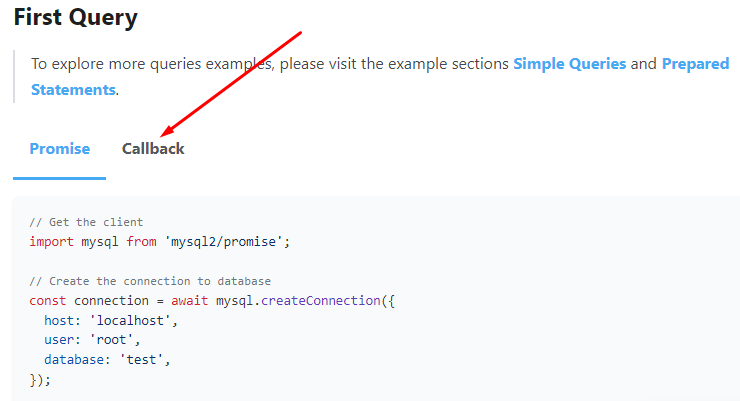




Ahora en la pagina de mysql2 voya seguir el siguiente recorrido









// Get the client

const mysql = require('mysql2/promise');

// Create the connection to database

const connection = mysql.createPool({

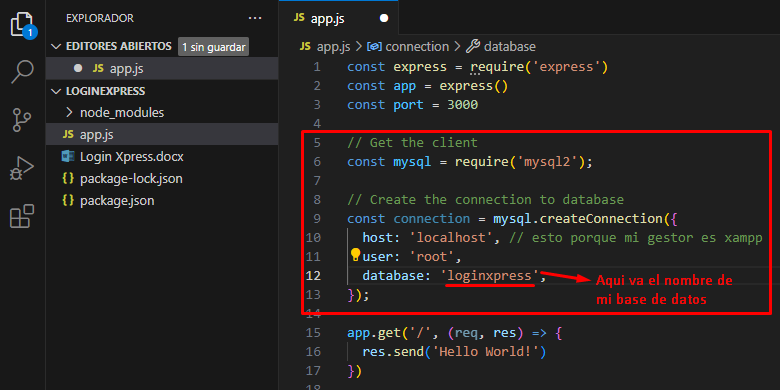
host: 'localhost', // esto porque mi gestor es xampp

user: 'root',

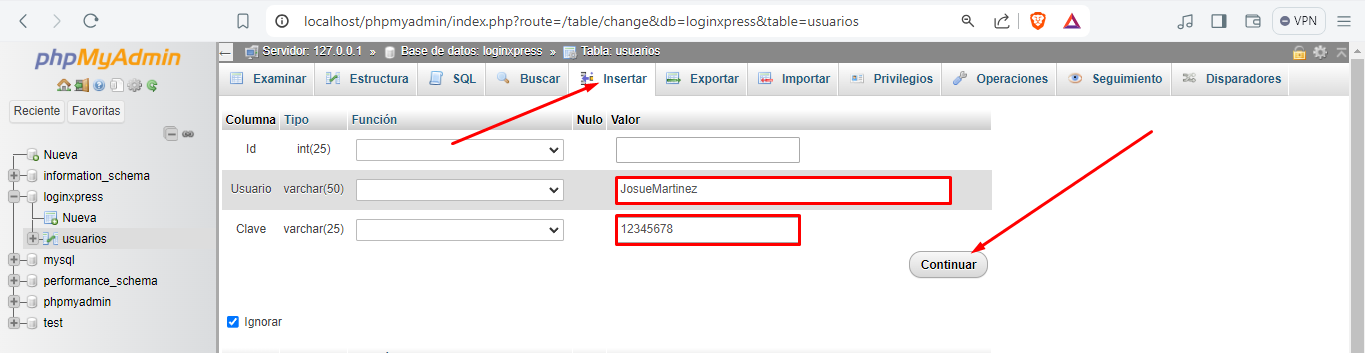
database: 'loginxpress',

});

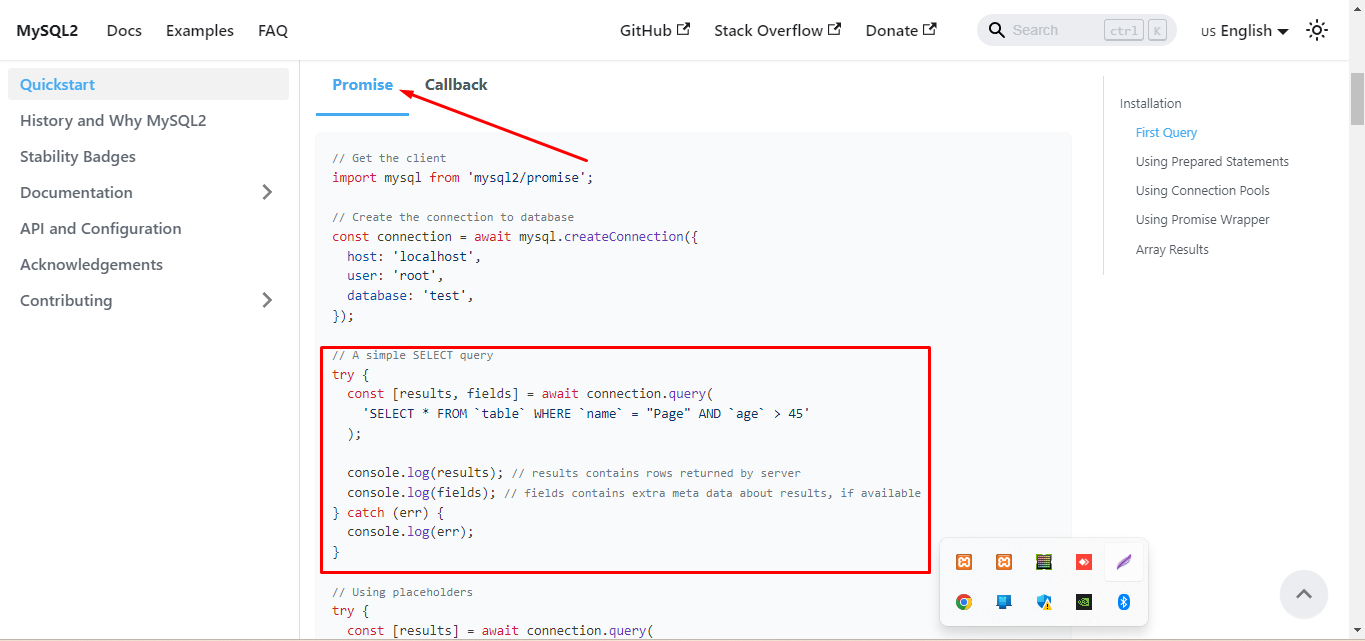
Y lo pego



Ahora vamos a hacer las consultas dentro de las rutas hacia nuestra base de datos, primero creamos un usuario



Ahora vamos a mi SQL2 y vamos a copear otro codigo



// A simple SELECT query

try {

const [results, fields] = await connection.query(

'SELECT \* FROM `table` WHERE `name` = "Page" AND `age` > 45'

);

console.log(results); // results contains rows returned by server

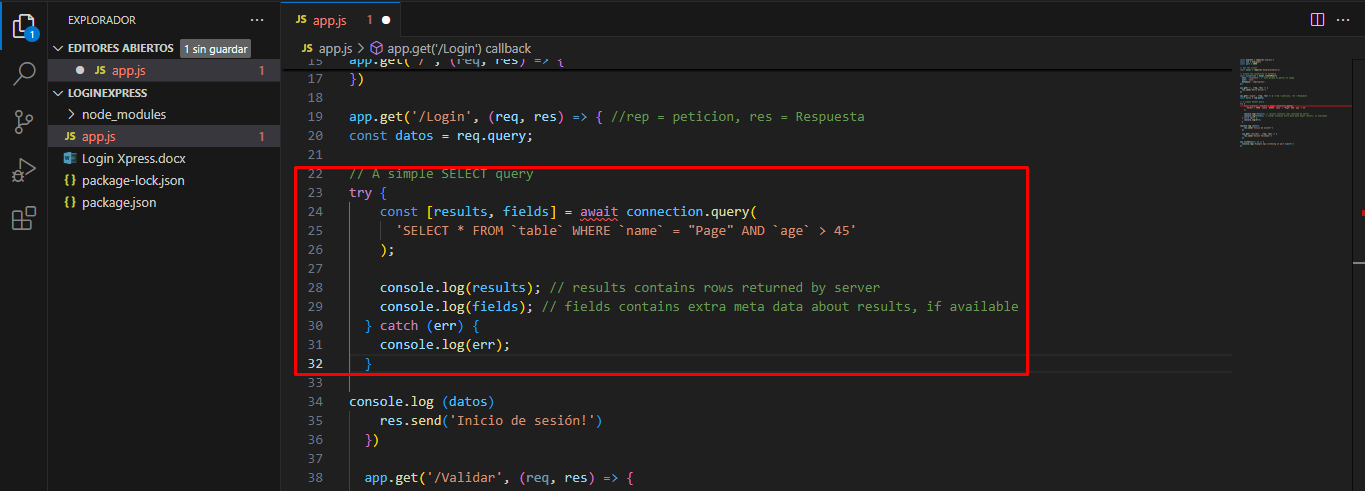
console.log(fields); // fields contains extra meta data about results, if available

} catch (err) {

console.log(err);

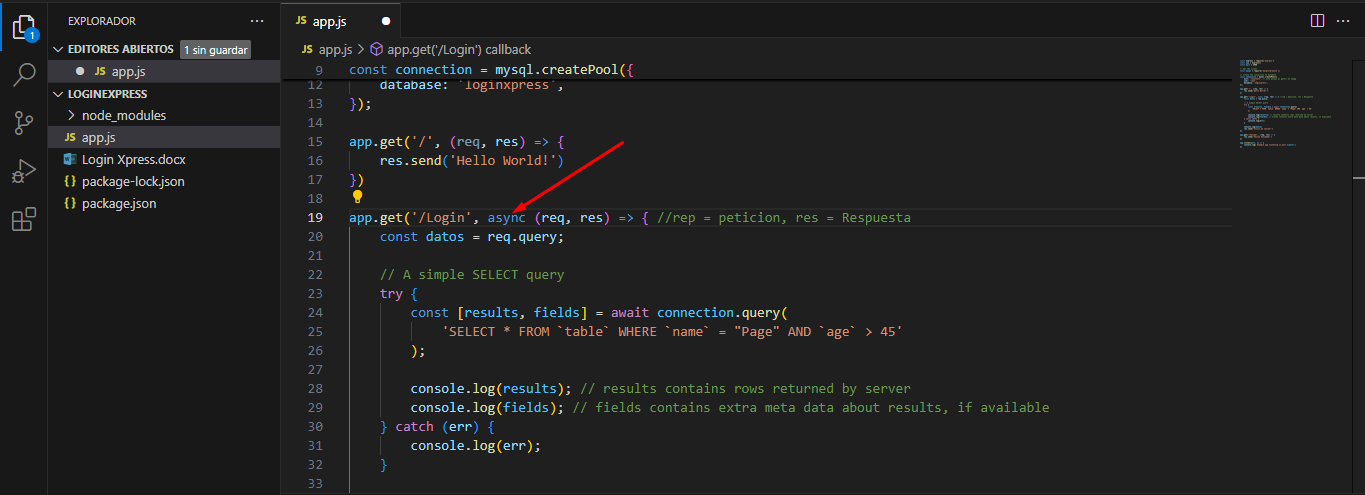
}

Y luego lo pego en mi VSC

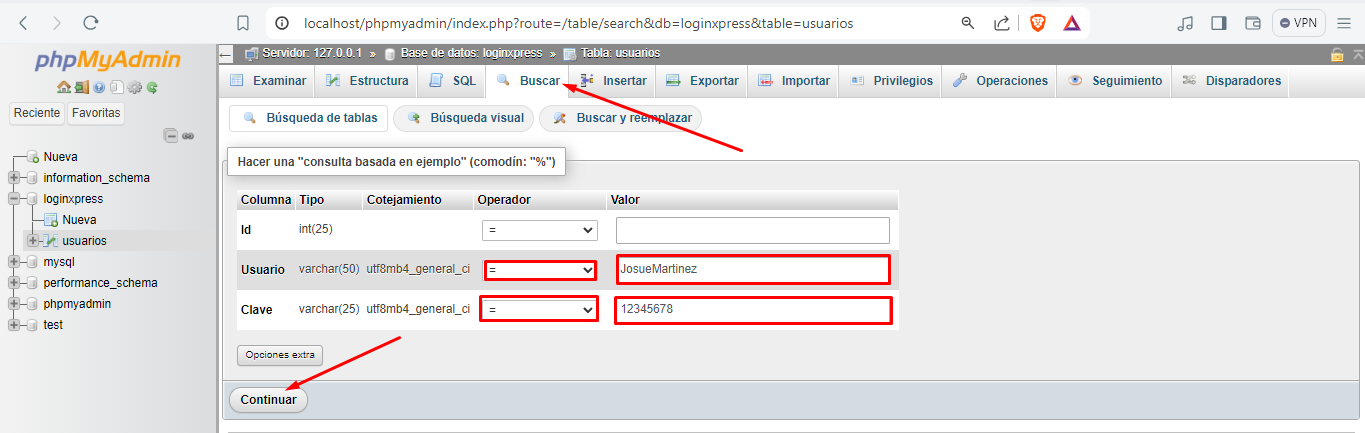


Si opimo alt + shift + f el codigo s eme va a organizar, pero hay un error, para ello vamos a agregar el codigo

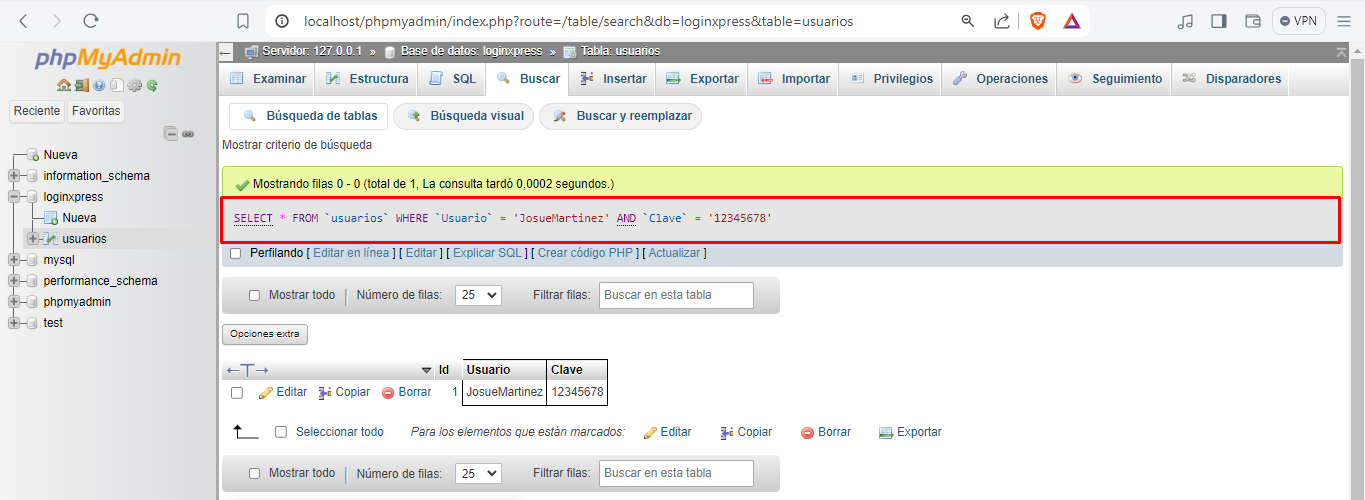
async



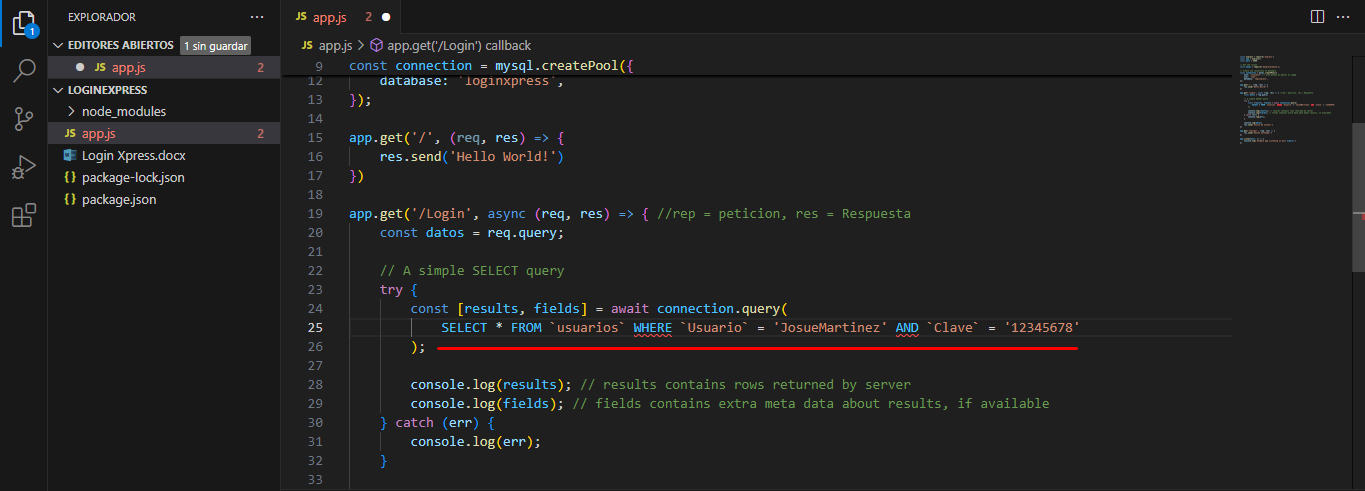
Ahora debemos hacer una consulta en nuestra base de datos y vamos a buscar el usuario que cree



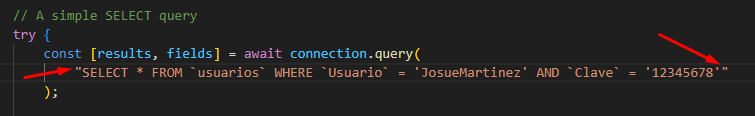
Vamos a copear el código que nos creo



Y vamos a reemplazarlo por otro código en VSC



Pero todo debió quedar en anaranjado porque todo es un texto así que debemos colocar comillas al inicio y al final



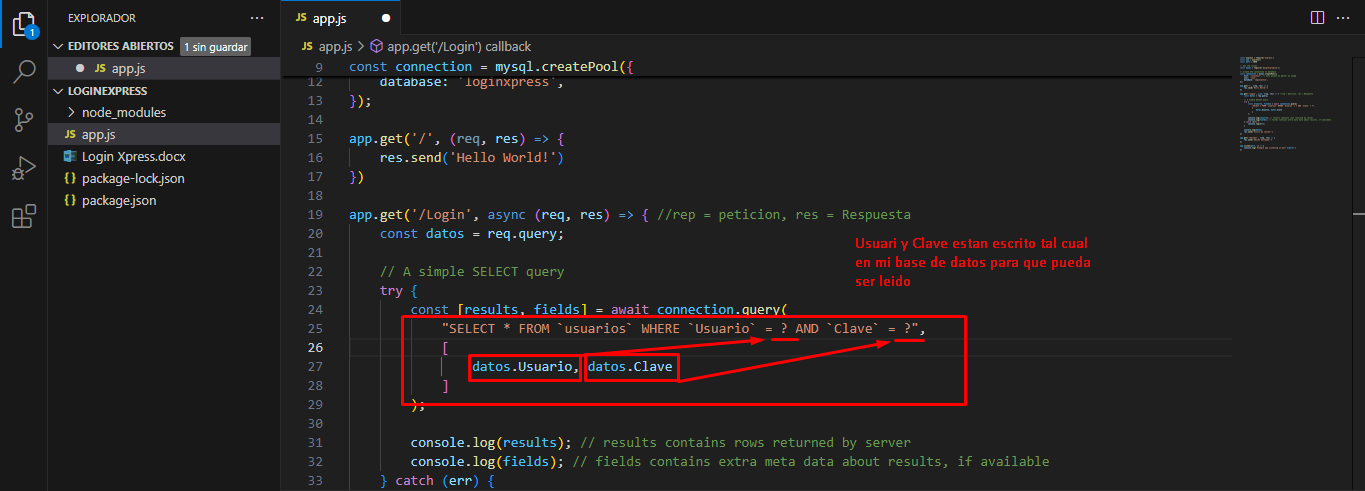
Ahora voy a ingresar la ruta del login y para esto debo hacer que mis datos sean más dinámicos y se cambien según mi búsqueda, para ello voy a realizar el siguiente cambio.

"SELECT \* FROM `usuarios` WHERE `Usuario` = ? AND `Clave` = ?",

[

datos.Usuario, datos.Clave

]



Ahora vamos a validar nuestro inicio de sesión

Para ello agregamos el siguiente código

if (results.length > 0)

{

res.status(200).send ("Inicio sesión correctamente")

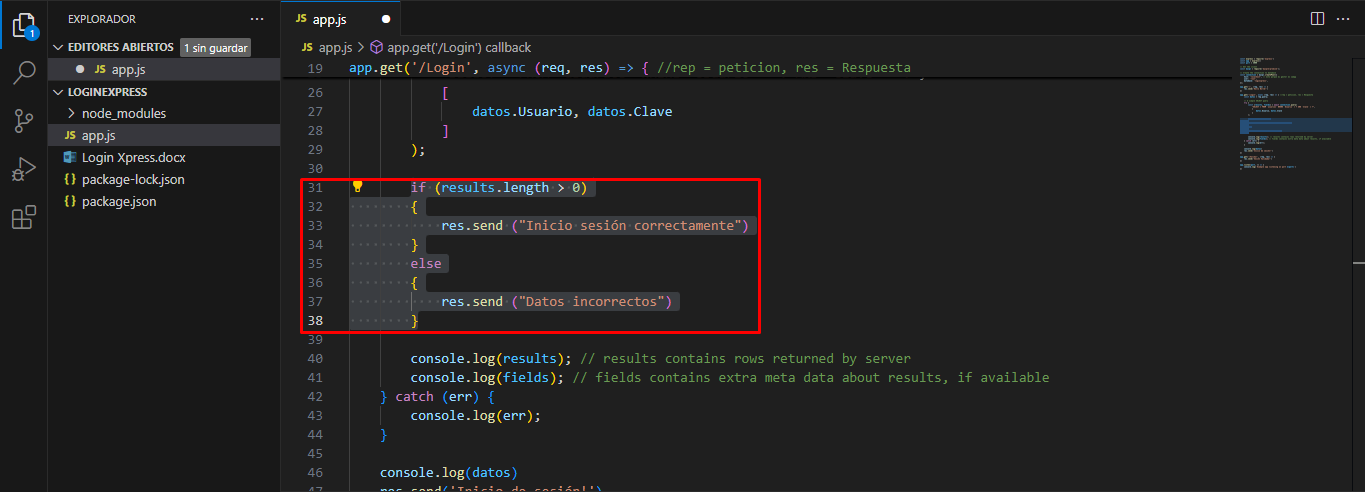
}

else

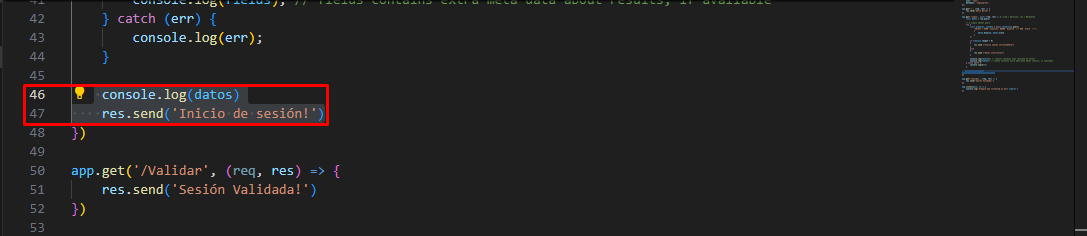
{

res.status(401).send ("Datos incorrectos")

}

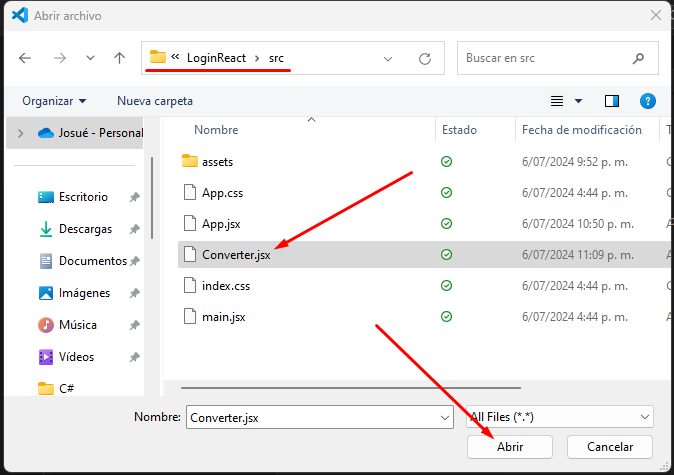


Y voy a eliminar parte de mi codido

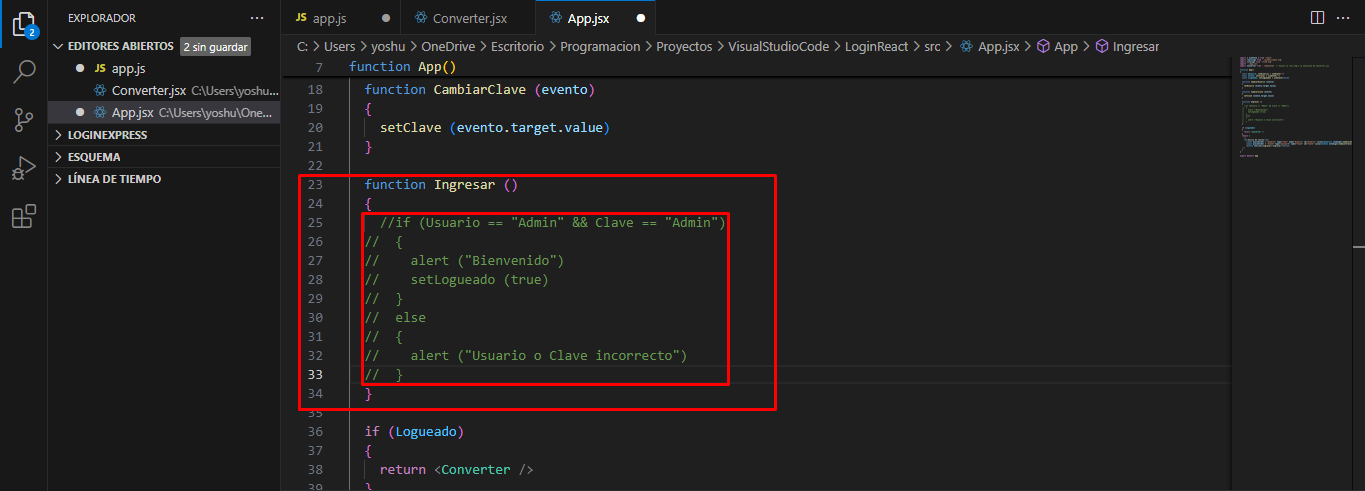


Ahora vamos a trabajar en el diseño de mi formulario ósea en el frontend.

Para ello voy a hacer uso de mi trabajo anterior voy a abrir mi archivo converter de mi proyecto anterior



Y me dirijo a mi función ingresar de mi app.js y voy a silenciar parte de mi código



Y voy a actualizar mi código

async function Ingresar ()

{

const Peticion = await fetch ("http://localhost:3000/Login?Usuario="+Usuario+"&Clave"+Clave)

if

(

Peticion.ok

)

{

setLogueado(true)

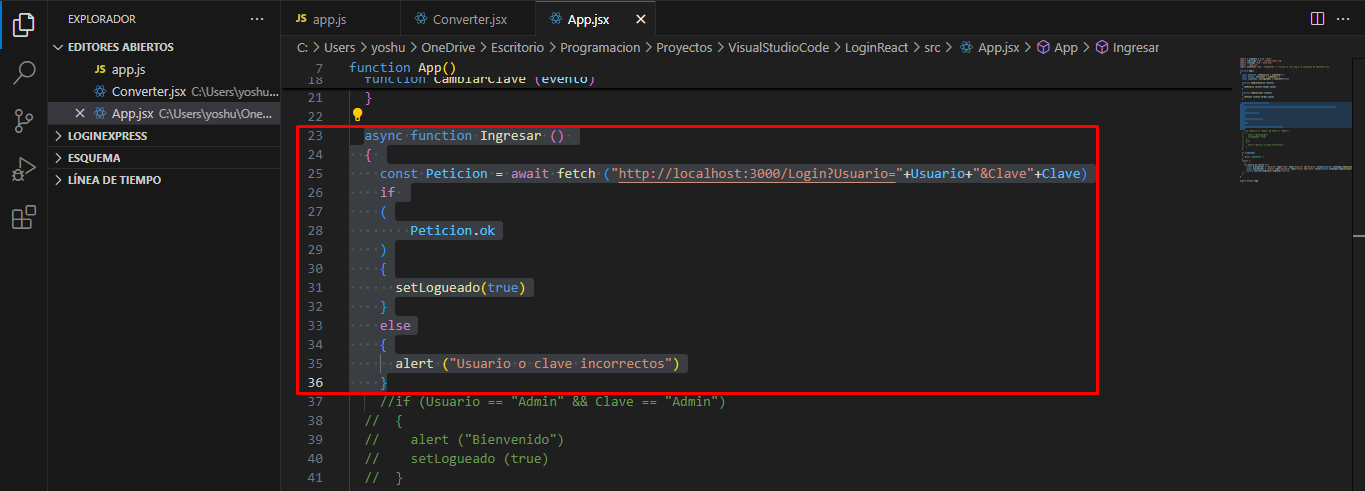
}

else

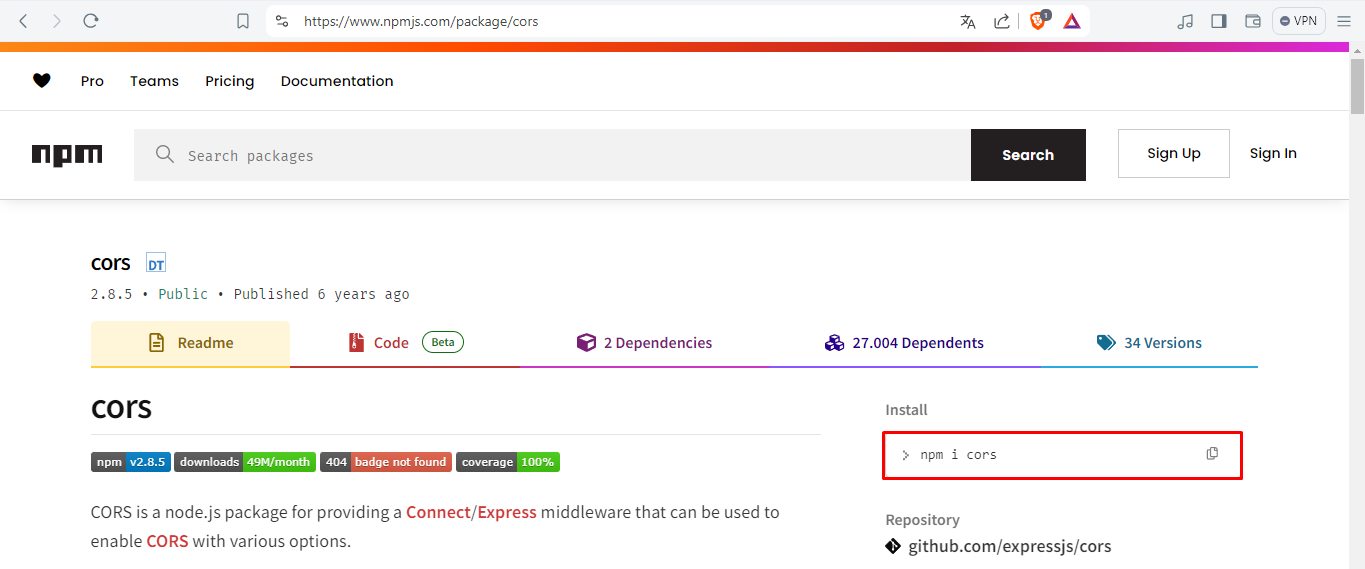
{

alert ("Usuario o clave incorrectos")

}



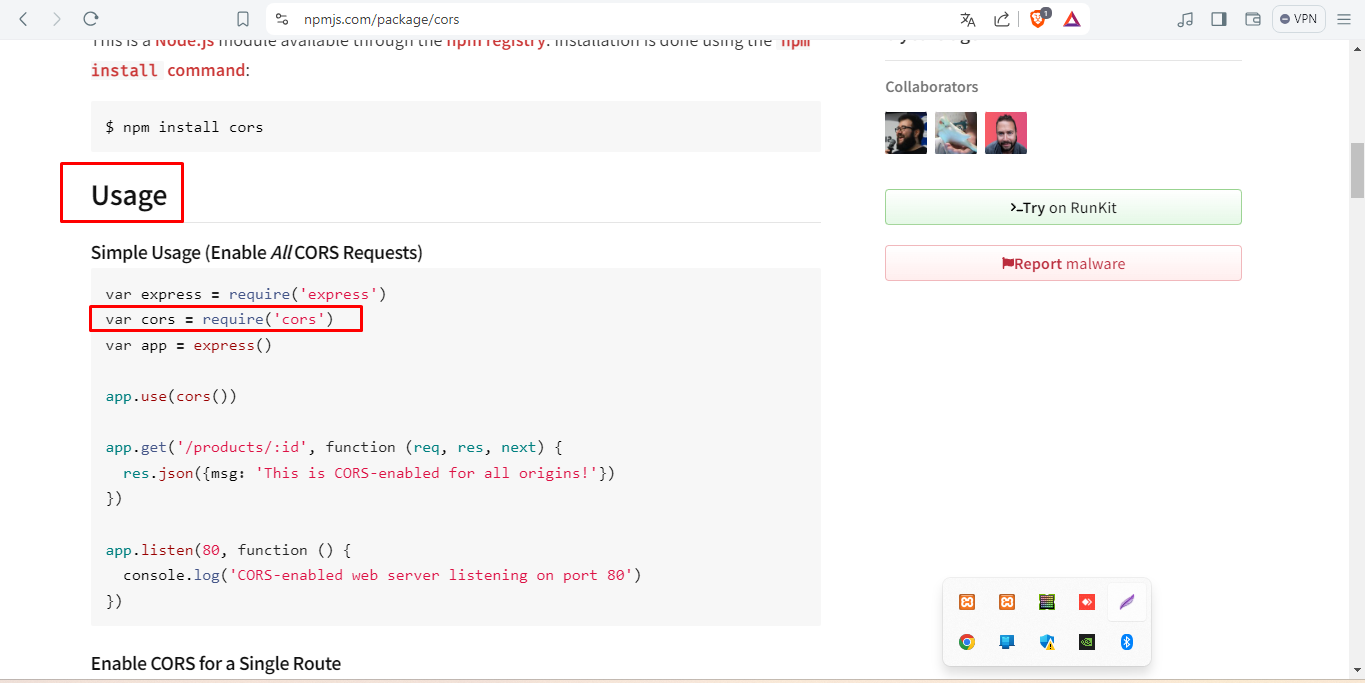
Ahora voy a cors npm



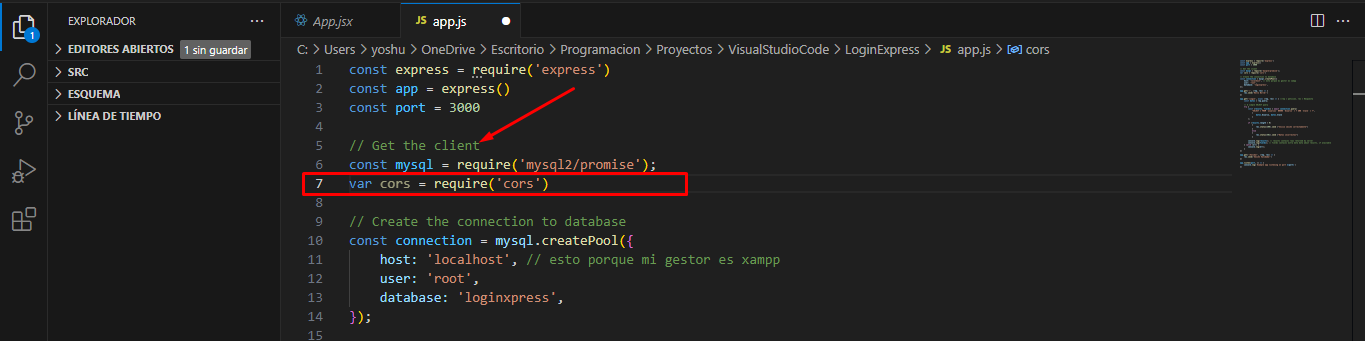
Copeo el código y lo pego en la terminal de mi proyecto.

También copeo el siguiente codigo:

var cors = require('cors')



Y lo pego en VSC



Le coloco una constante

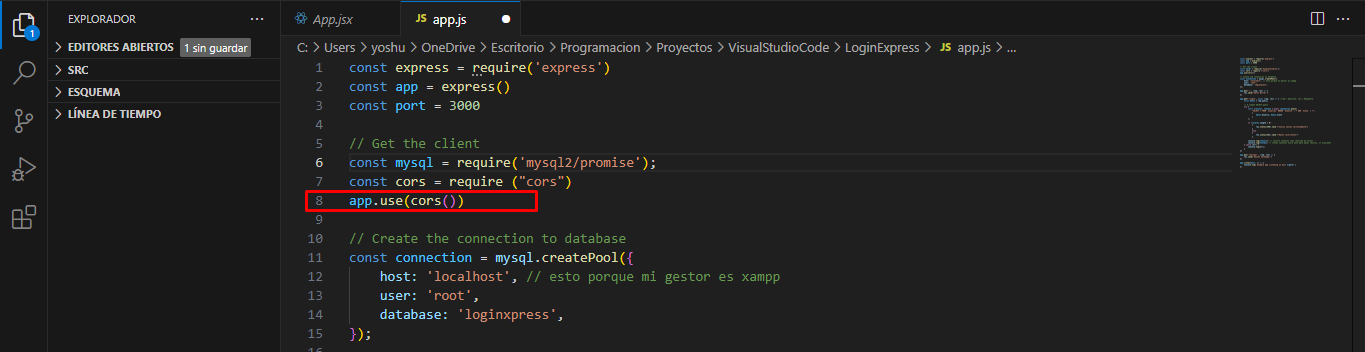
const cors = require ("cors")



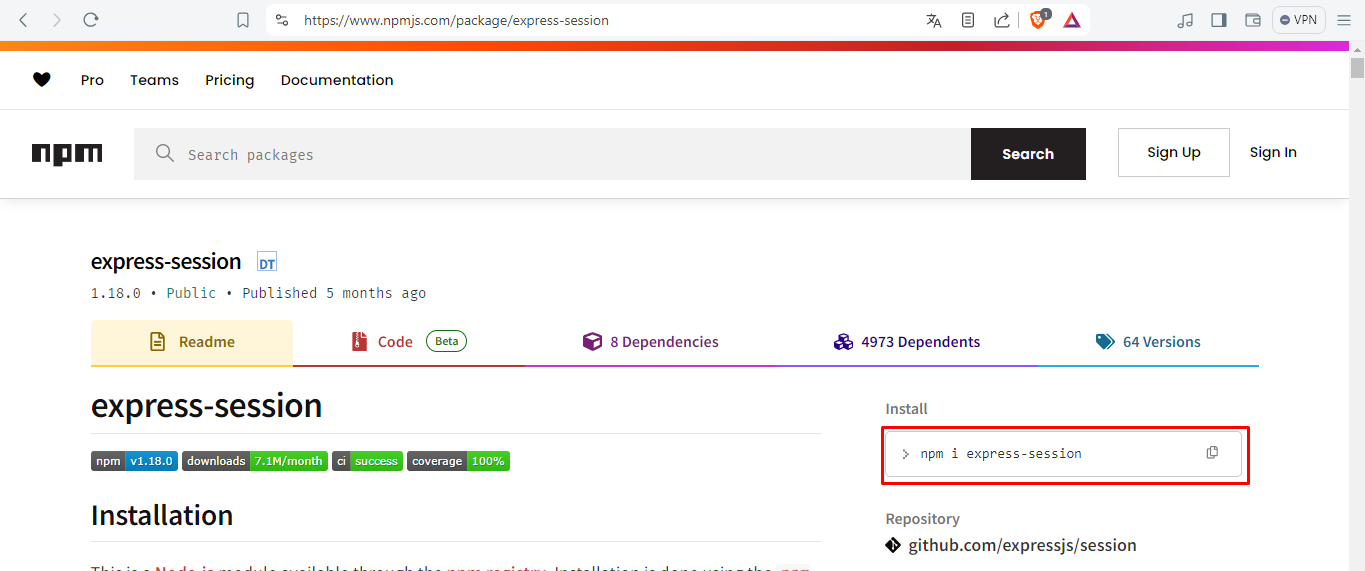
Y copeo otro código de cors

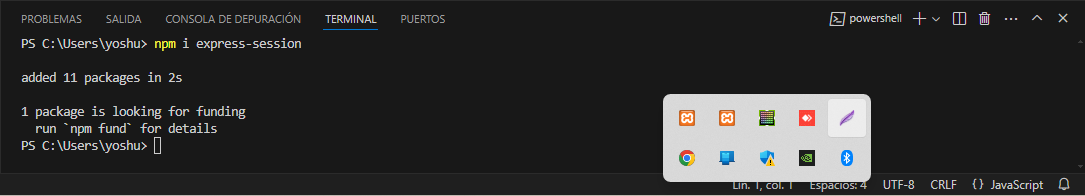
app.use(cors())





Ahora necesitamos otro paquete para trabajar en el inicio de sesión por lo que voy a donde esta alojado el código y posterior a ello lo instalo.

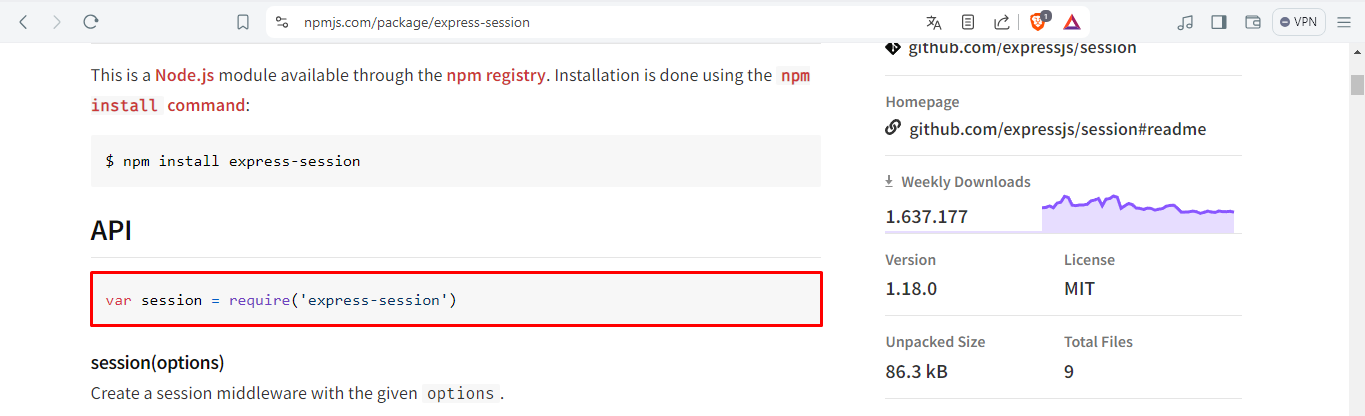




Copeo el código de uso

const session = require('express-session')

pero le quito la palabra var y la cambio por const

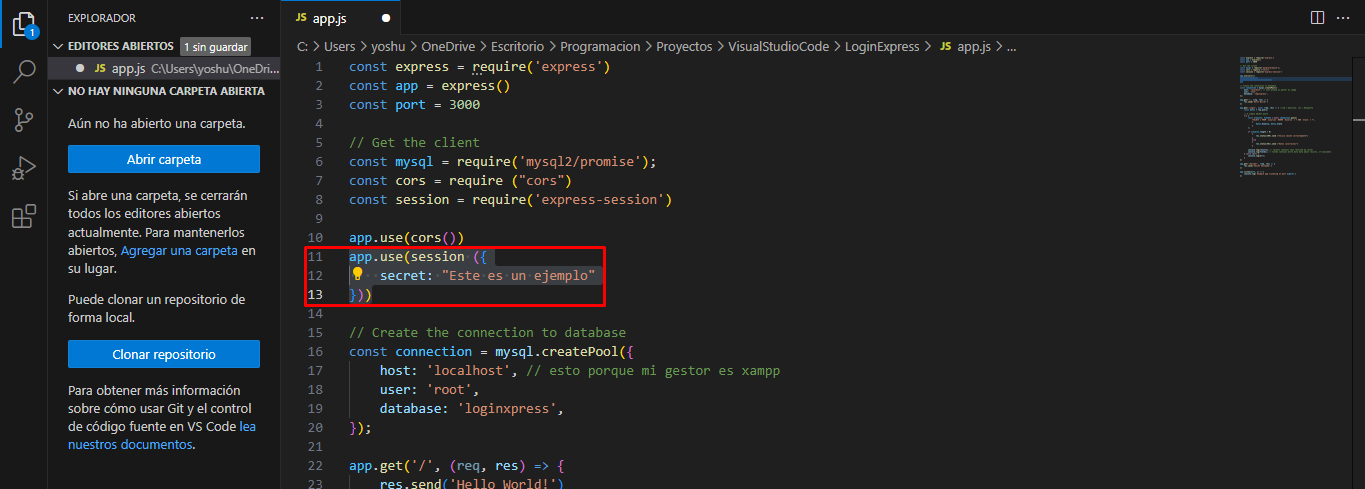


Luego uso el código

app.use(session ({

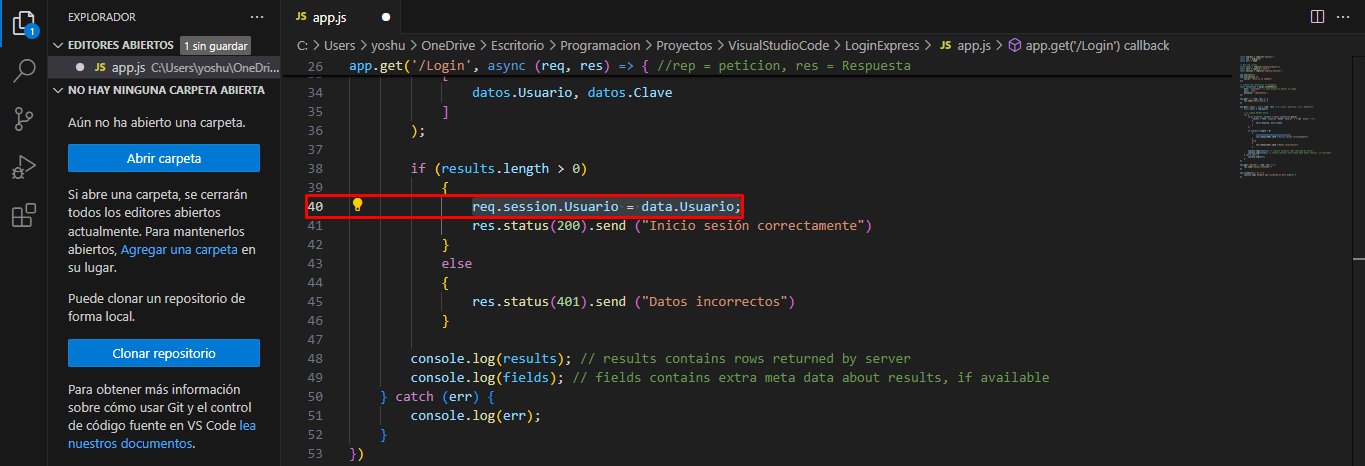
secret: "Este es un ejemplo"

}))



Y también voy a modificar este otro código

req.session.Usuario = datos.Usuario;



También vamos a modificar otra parte

if (req.session.Usuario)

{

res.status (200).send ("Sesión validada")

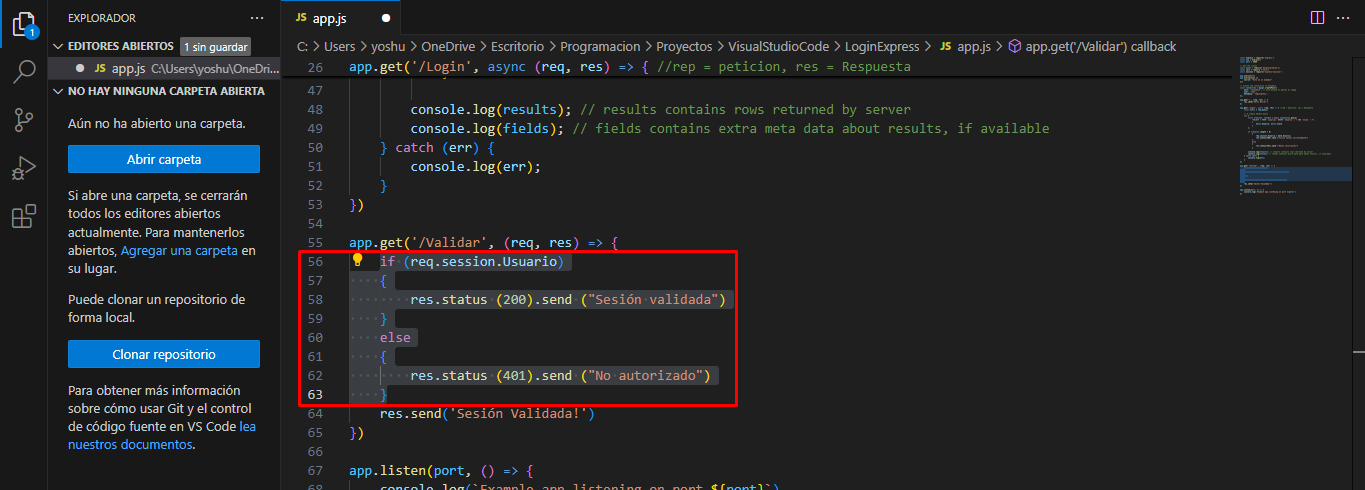
}

else

{

res.status (401).send ("No autorizado")

}



Ahora vamos a modificar otra parte del código

{

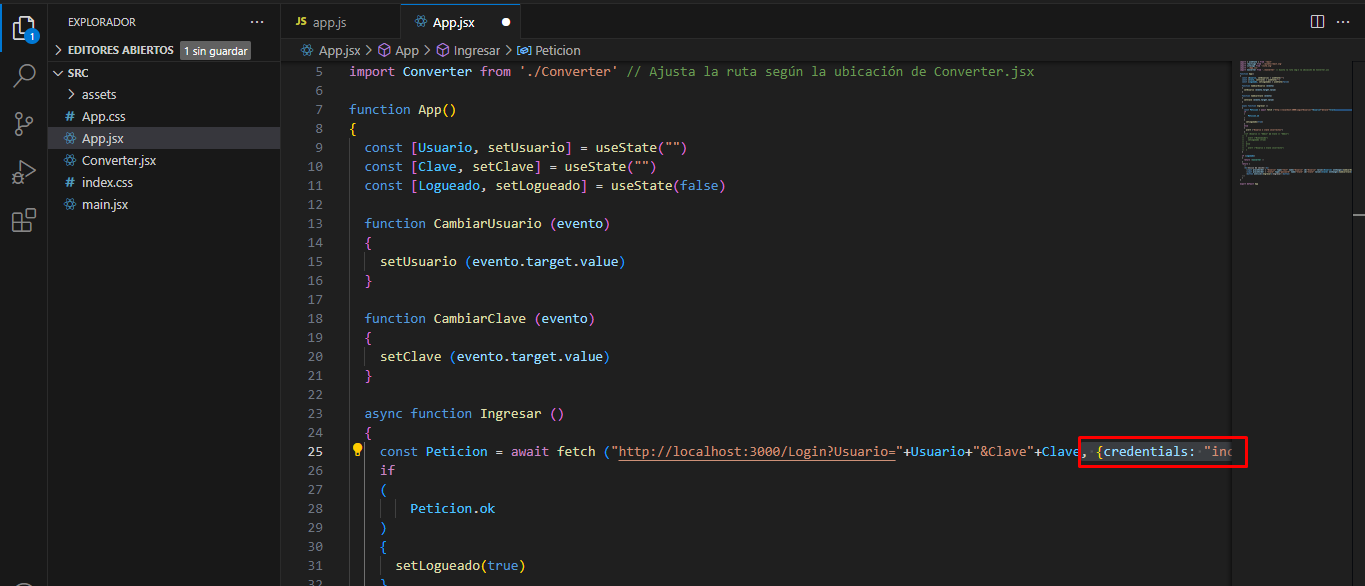
credentials : true

}



Ahora vuelvo a mi react y modifico el siguiente código

, {credentials: "include"}



Allí mismo vamos a crear otra función

async function Validar ()

{

const Peticion = await fetch ("http://localhost:3000/Validar", {credentials: "include"})

if

(

Peticion.ok

)

{

setLogueado(true)

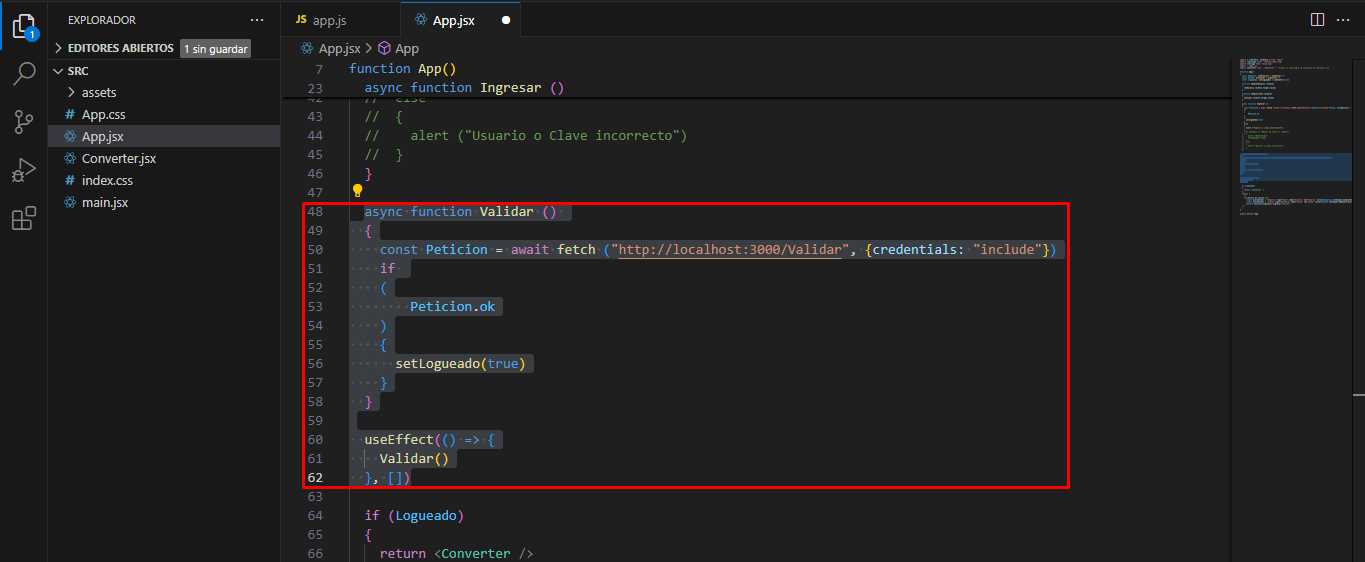
}

}

useEffect(() => {

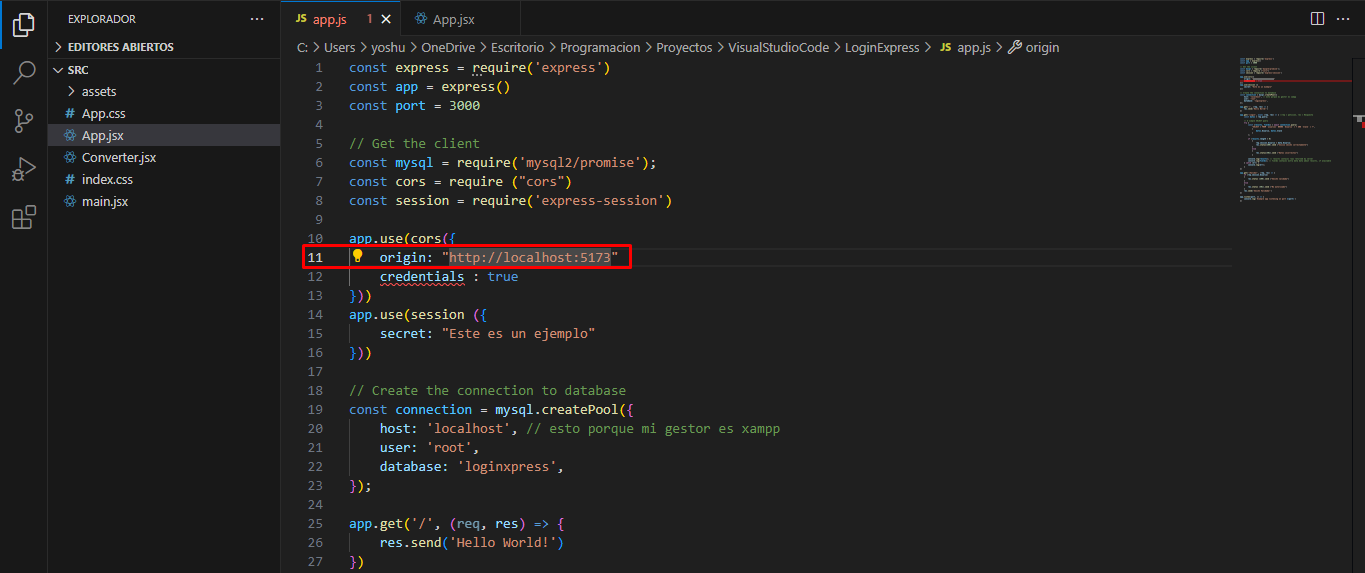
Validar()

}, [])

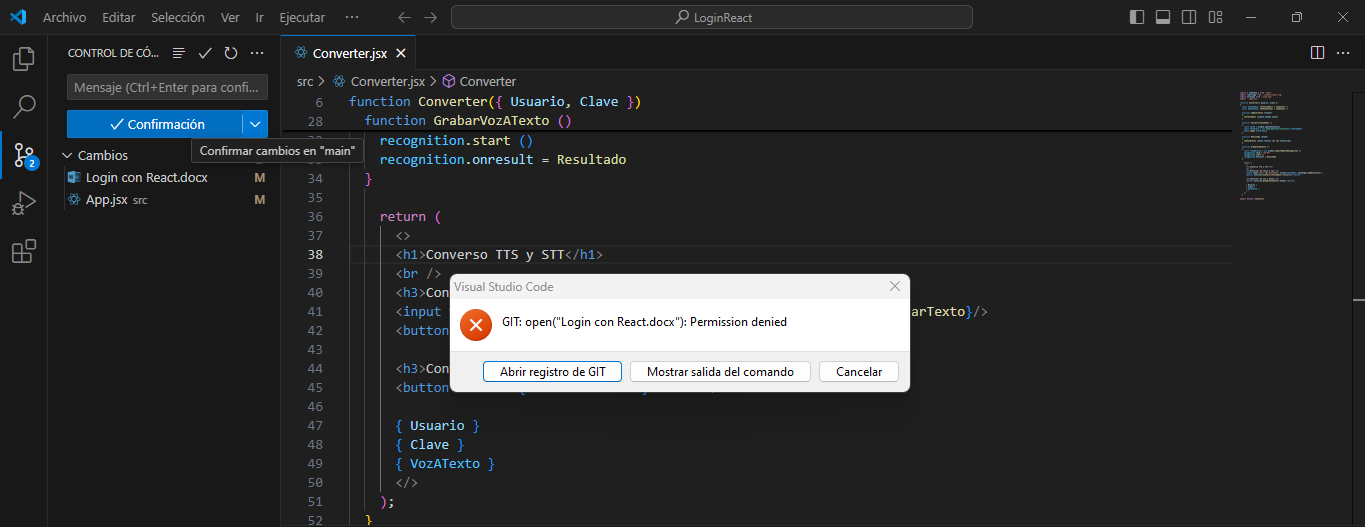


Y vamos a corregir otra parte de código

origin: “[http://localhost:5173](http://localhost:5173/)”,



Ahora vamos a subir los cambios a git, no puedo guardaros asi poruqe me sale un error



Esto se debe a que estoy usando código de otro lado u otro proyecto, por lo que debo ir a mi proyecto y darle la opción de que lo ignore.

Voy a crear un nuevo archivo con el nombre

.gitignore

