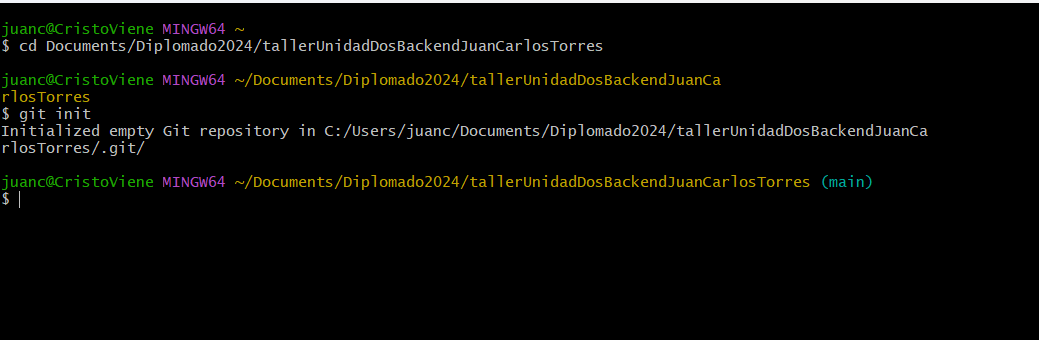
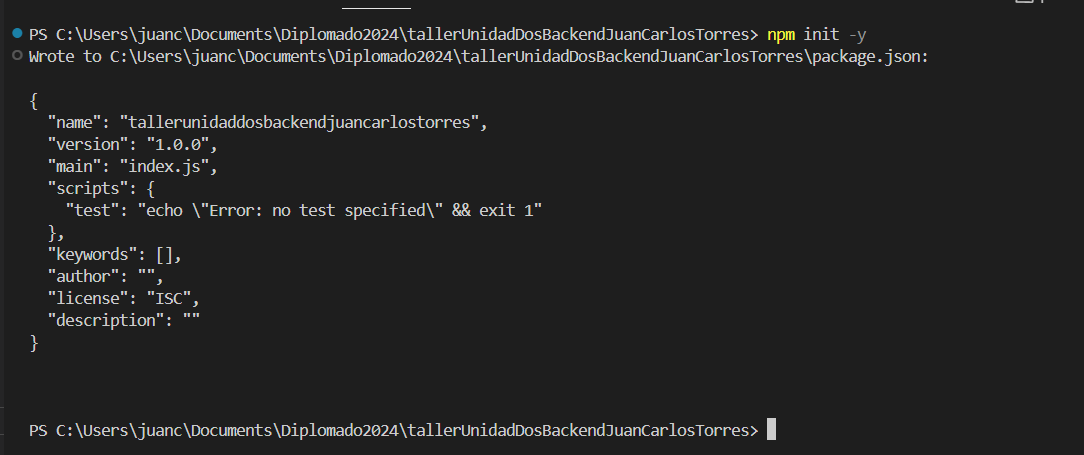
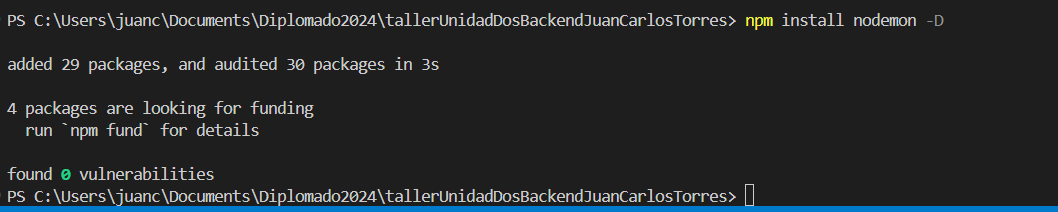
Entre a la carpeta tallerUnidadDosBackendJuanCarlosTorres para crear un repositorio para el seguimiento de versiones de mi proyecto



Desde visual Visual Studio entramos a la carpeta y escribimos el comando npm init -y para inicializar nuestro proyecto de backend



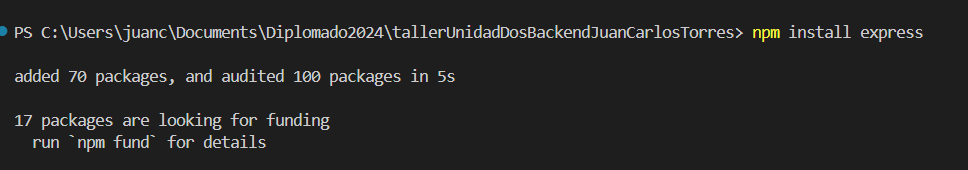
Escribimos el comando npm Install nodemon -D descarga un modulo para desarrollar el cual se encarga de reiniciar la aplicación de Node.js cada vez que detecta cambios en el código, esto evita que tenga que detener el servicio cada ves haga un cambio en mi codigo



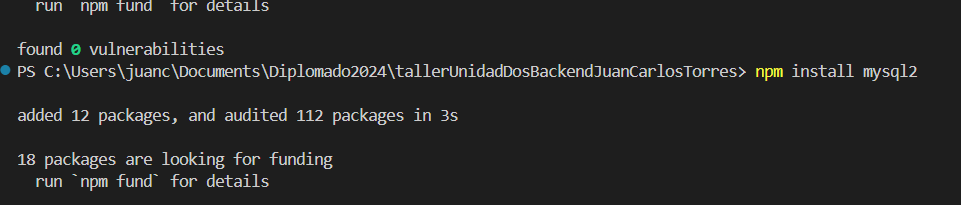
Cree una nueva caparte llamada src, es donde voy apilar todo mi código



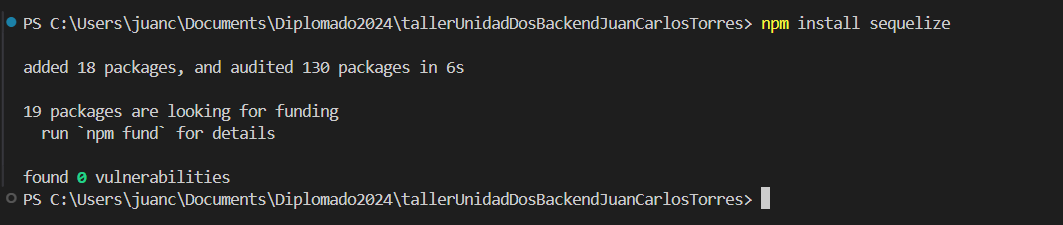
Instalamos express, que es el frameworks que me va permitir desarrollar el backend



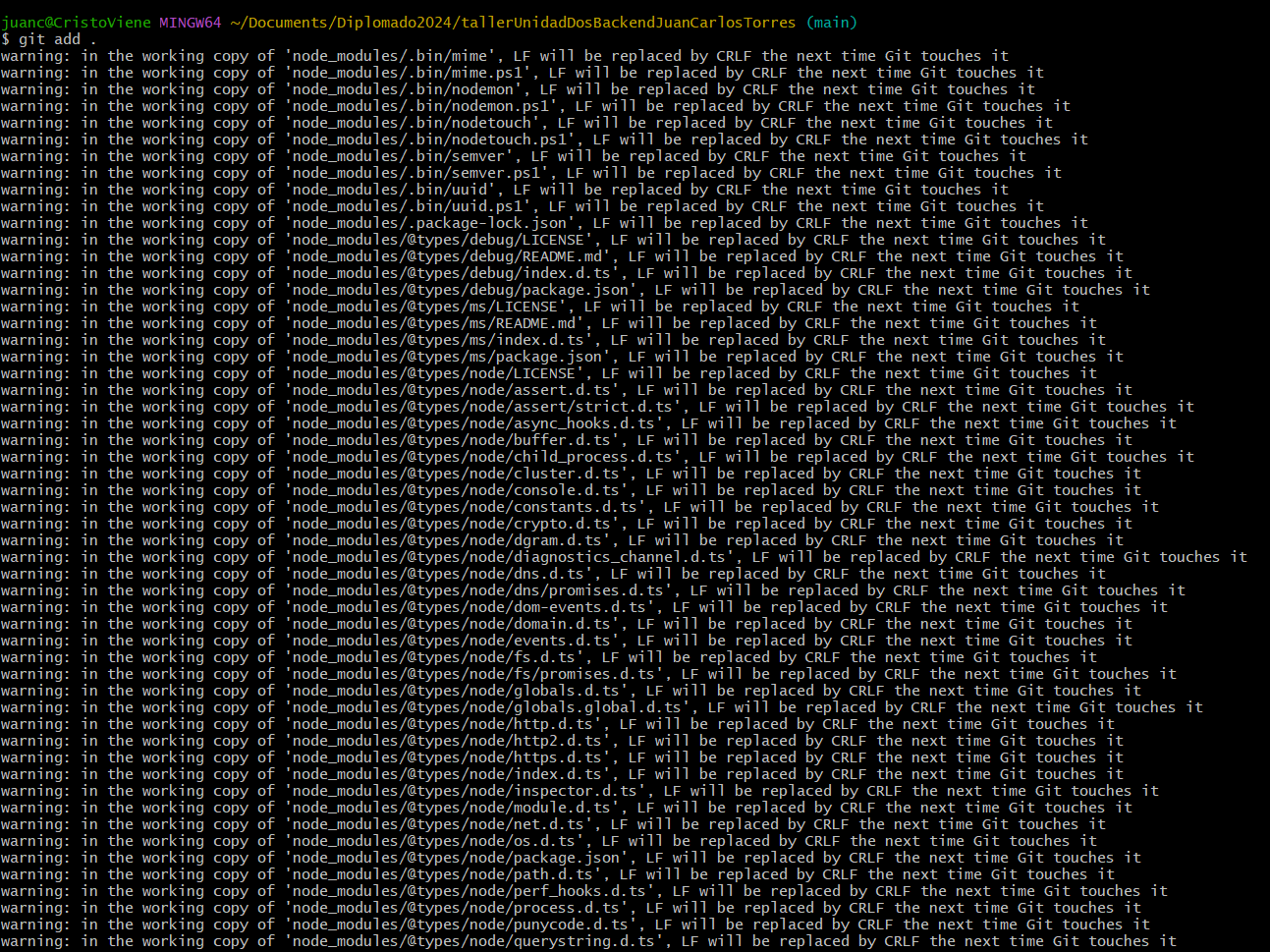
Instalamos mysql2 para que me permita conectarme base de datos mysql2



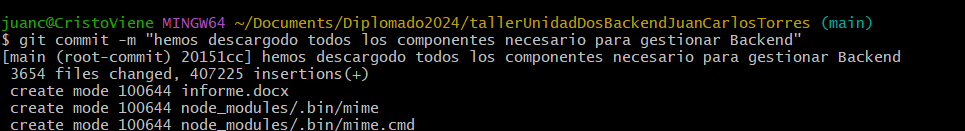
Instalamos sequelize porque este es un ORM el cual es un intermediario entre el desarrollador y la base de datos, evitando que este tenga que preocuparse por saber exactamente cuáles son las sentencias de mysql2 para gestionar la base de datos



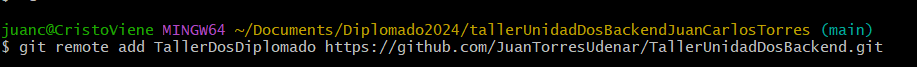
Me voy a mi consola Git y añado los cambios que tengo hasta el momento al área de preparación



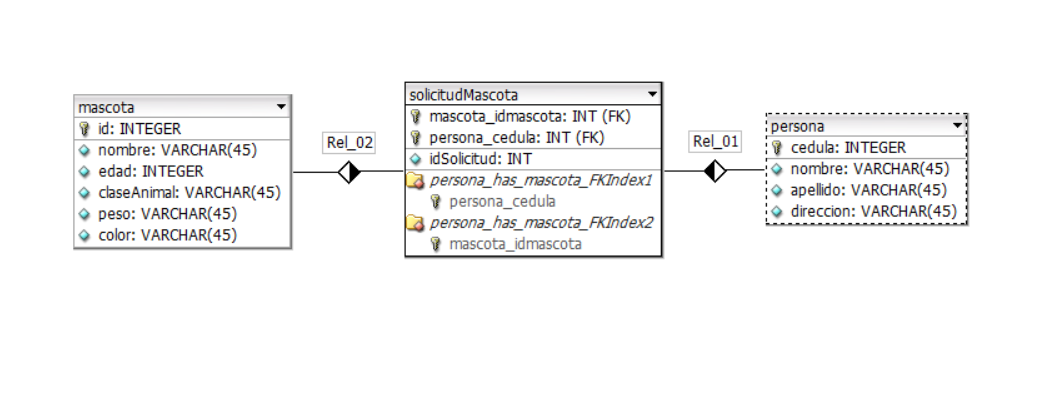
Hacemos un commit



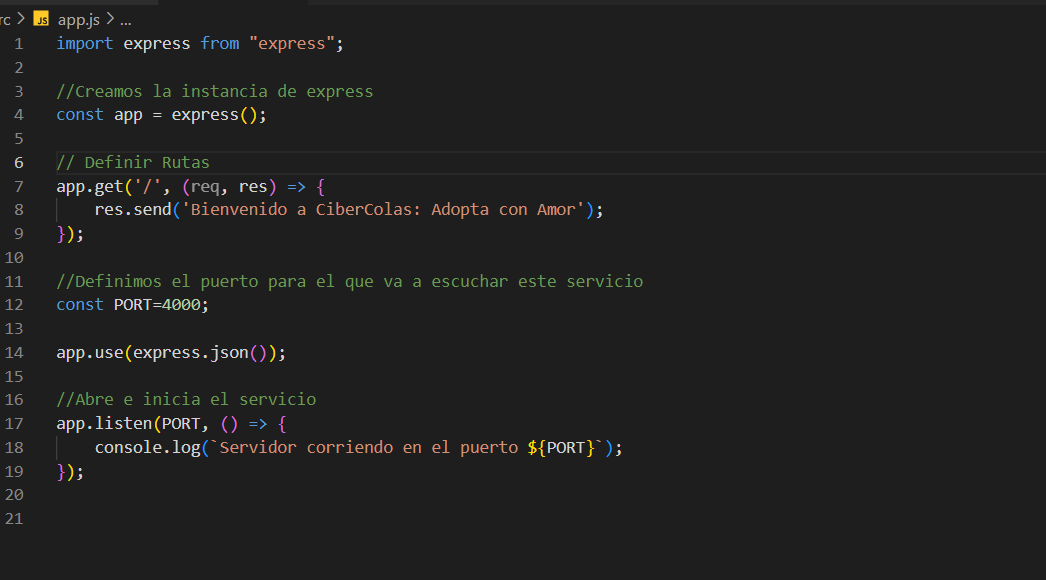
Sincronizamos el repositorio con el repositorio local



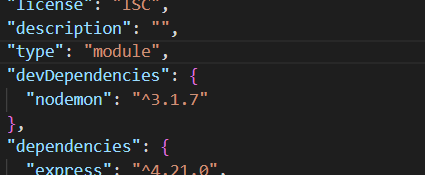
Hice un diagrama entidad relación para tener una idea clara de cuantas tablas hacer para mi base de datos, los atributos



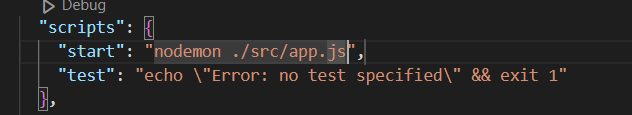
Empezamos a codificar nuestro archivo app.js



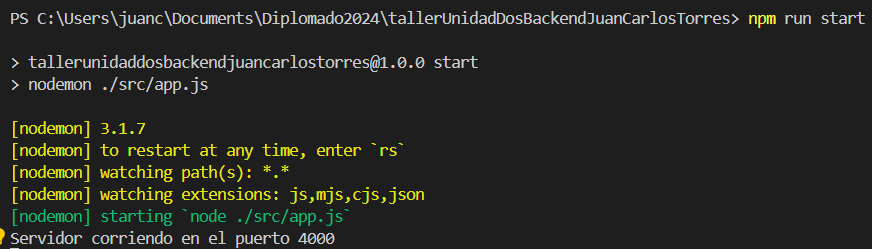
Se me había olvidado decirle que se debe comportar como un modulo



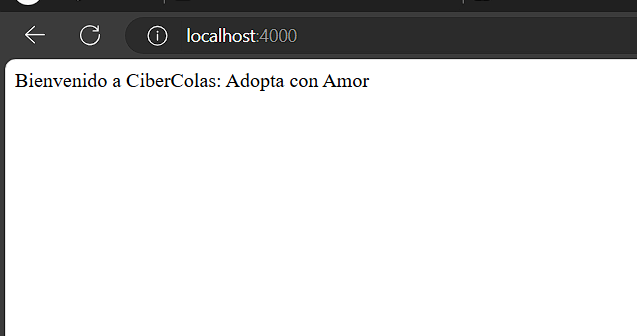
Me estaba olvidando de hacer un script que contiene la ruta donde esta el archivo app.js



Este comando es para correr el Scripts el cual se llama start, con esto el sistema esta pendiente de los cambios que yo haga y me los actualiza en caliente, evitando de que este bajando y subiendo el servicio a cada rato



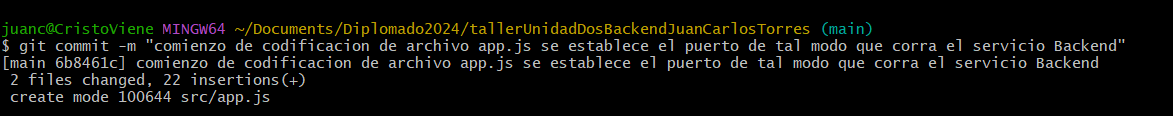
Podemos visualizar el mensaje de bienvenida y el nombre de la empresa la cual se encarga en dar en adopción mascotas



Añado los cambios que tengo hasta el momento al área de preparación



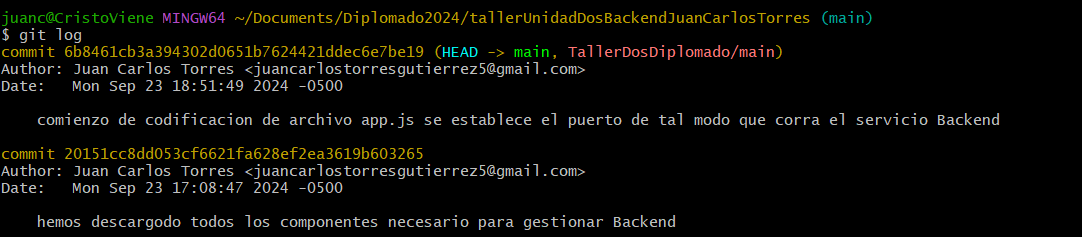
Hacemos un commit



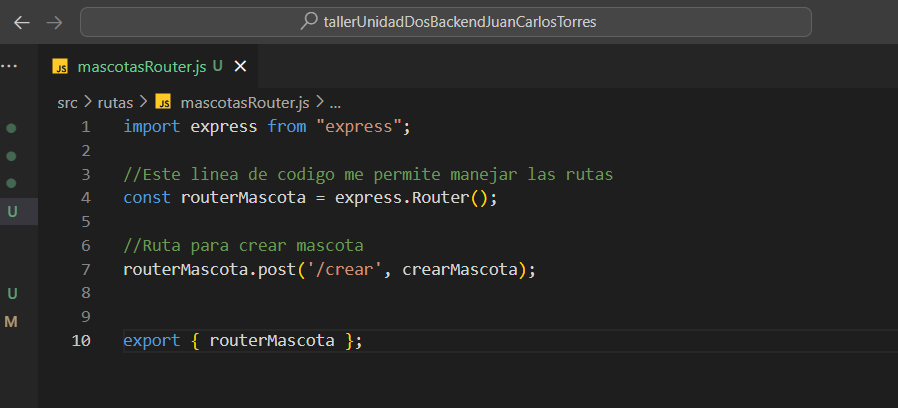
Subimos el avance



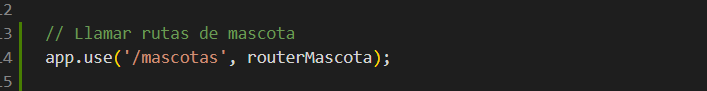
Evidenciamos los commit que tenemos hasta el momento de nuestro proyecto



Creamos una carpeta dentro de la carpeta src la cual se llama rutas, y establecemos la primera ruta para crear una mascota



En el archivo app.js añadimos esta línea de código que me permite llamar usar o implementar las ruta que ya establecí



E imputamos routerMascota en el archivo app.js

