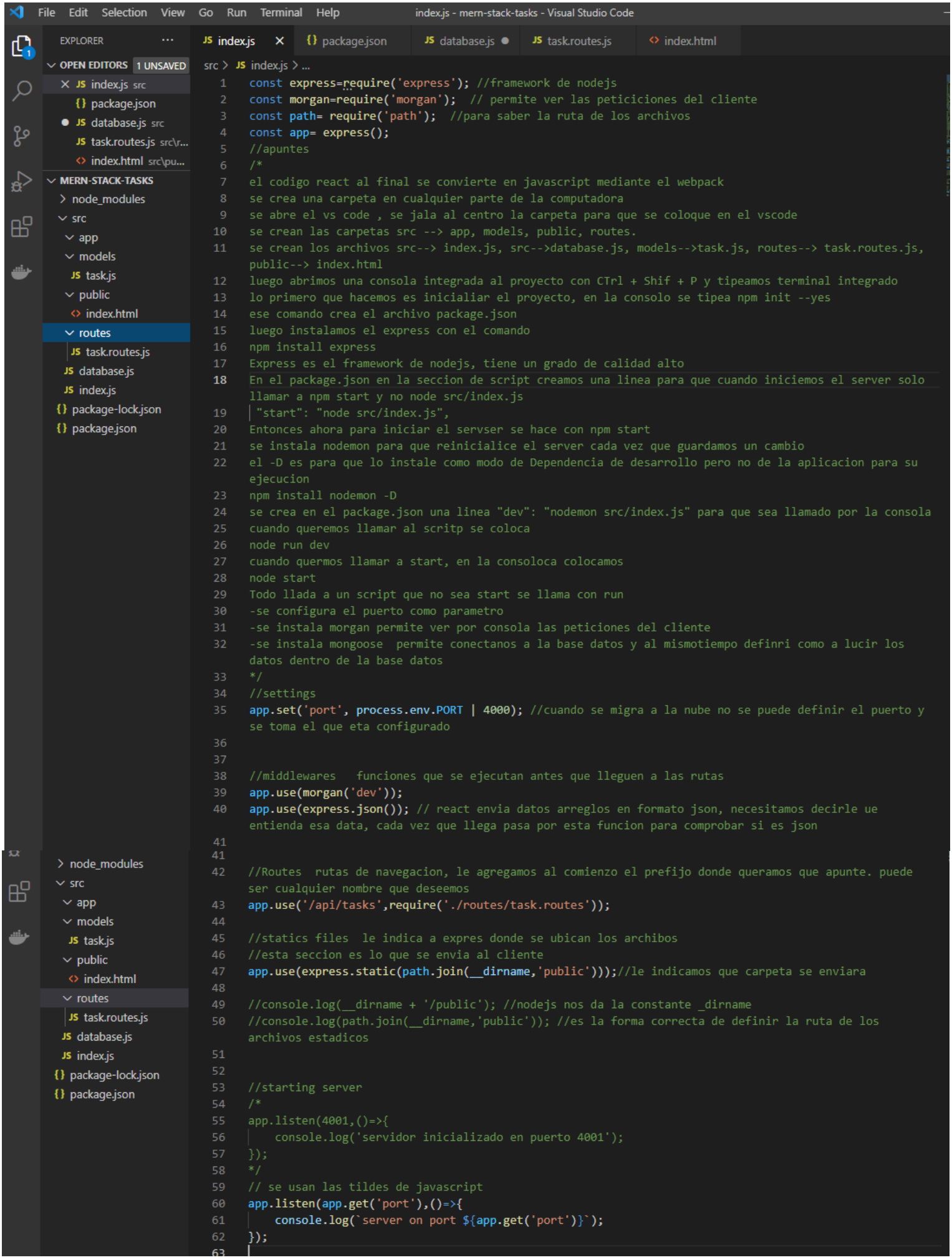
File Edit Selection View Go Run Terminal Help
index.js - mern-stack-tasks - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
index.js src
package.json
database.js src
task.routes.js src\r...
index.html src\pu...
MERN-STACK-TASKS
node_modules
src
app
models
task.js
public
index.html
routes
task.routes.js
database.js
index.js
package-lock.json
package.json
index.js
package.json
database.js
task.routes.js
index.html
src > index.js > ...
const express=require('express'); //framework de nodejs
const morgan=require('morgan'); // permite ver las peticiciones del cliente
const path= require('path'); //para saber la ruta de los archivos
const app= express();
//apuntes
/*
el codigo react al final se convierte en javascript mediante el webpack
se crea una carpeta en cualquier parte de la computadora
se abre el vs code , se jala al centro la carpeta para que se coloque en el vscode
se crean las carpetas src --> app, models, public, routes.
se crean los archivos src--> index.js, src-->database.js, models-->task.js, routes--> task.routes.js, public--> index.html
luego abrimos una consola integrada al proyecto con CTrl + Shif + P y tipeamos terminal integrado
lo primero que hacemos es inicialiar el proyecto, en la consolo se tipea npm init --yes
ese comando crea el archivo package.json
luego instalamos el express con el comando
npm install express
Express es el framework de nodejs, tiene un grado de calidad alto
En el package.json en la seccion de script creamos una linea para que cuando iniciemos el server solo llamar a npm start y no node src/index.js
"start": "node src/index.js",
Entonces ahora para iniciar el servser se hace con npm start
se instala nodemon para que reinicialice el server cada vez que guardamos un cambio
el -D es para que lo instale como modo de Dependencia de desarrollo pero no de la aplicacion para su ejecucion
npm install nodemon -D
se crea en el package.json una linea "dev": "nodemon src/index.js" para que sea llamado por la consola
cuando queremos llamar al scritp se coloca
node run dev
cuando quermos llamar a start, en la consoloca colocamos
node start
Todo llada a un script que no sea start se llama con run
-se configura el puerto como parametro
-se instala morgan permite ver por consola las peticiones del cliente
-se instala mongoose permite conectanos a la base datos y al mismotiempo definri como a lucir los datos dentro de la base datos
*/
//settings
app.set('port', process.env.PORT | 4000); //cuando se migra a la nube no se puede definir el puerto y se toma el que eta configurado
//middlewares funciones que se ejecutan antes que lleguen a las rutas
app.use(morgan('dev'));
app.use(express.json()); // react envia datos arreglos en formato json, necesitamos decirle ue entienda esa data, cada vez que llega pasa por esta funcion para comprobar si es json
//Routes rutas de navegacion, le agregamos al comienzo el prefijo donde queramos que apunte. puede ser cualquier nombre que deseemos
app.use('/api/tasks',require('./routes/task.routes'));
//statics files le indica a expres donde se ubican los archibos
//esta seccion es lo que se envia al cliente
app.use(express.static(path.join(__dirname,'public')));//le indicamos que carpeta se enviara
//console.log(__dirname + '/public'); //nodejs nos da la constante _dirname
//console.log(path.join(__dirname,'public')); //es la forma correcta de definir la ruta de los archivos estadicos
//starting server
/*
app.listen(4001,()=>{
console.log('servidor inicializado en puerto 4001');
});
*/
// se usan las tildes de javascript
app.listen(app.get('port'),()=>{
console.log(`server on port ${app.get('port')}`);
});