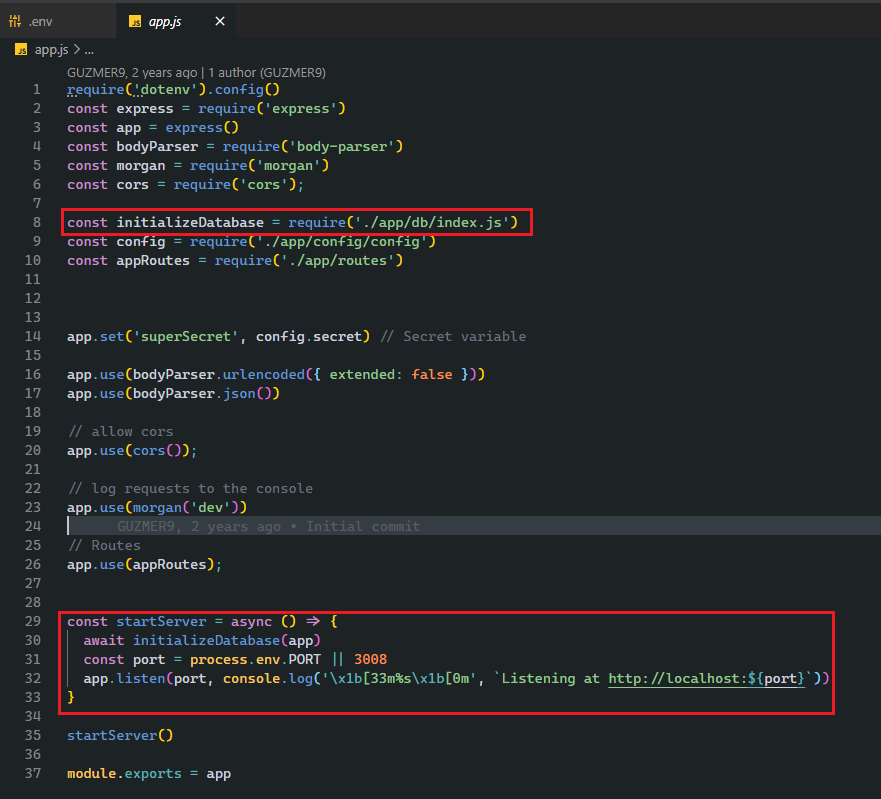
# **Conexión a base de datos usando NodeJS y React**

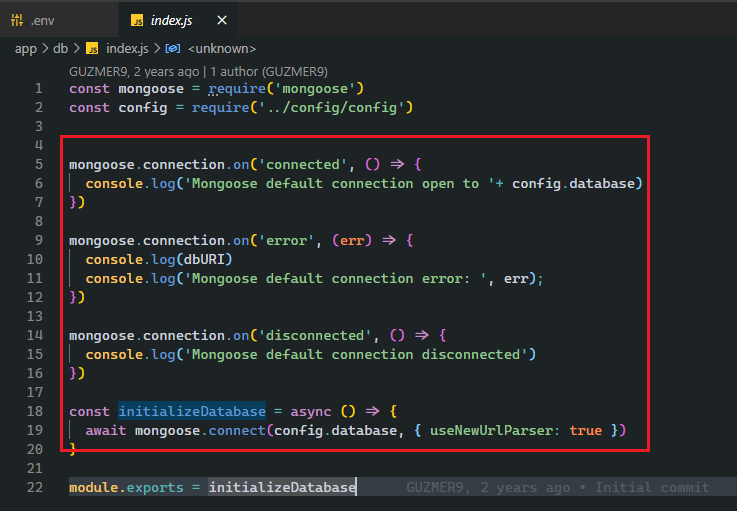
A continuación, explicaré cómo puedo conectar mi aplicación javascript a una base de datos, para este ejemplo he utilizado un backend, NodeJS, el cual, obviamente es javascript, y el cliente es manejado por una aplicación react, también javascript, esto facilita el desarrollo pues comparten el lenguaje de programación.

Estoy usando NodeJS en el lado del servidor pues porque es necesario tener un lenguaje o tecnología del lado del servidor para poder manejar este tipo de dato, básicamente necesitamos de dos aplicaciones, en la mayoría de casos, una aplicación destinada al cliente (frontend) y otra aplicación destinada al servidor (backend). Estas dos aplicaciones deben comunicarse usando un protocolo para este fin, por ejemplo, usando XML o mejor, usando JSON, JavaScript Object Notation, así compartimos un ecosistema javascript que nos facilita la programación, diseño y mantenimiento de nuestro sistema.

A continuación, veamos el archivo app.js de nuestra aplicación, en el creamos el servidor node, e inicializamos la base de datos



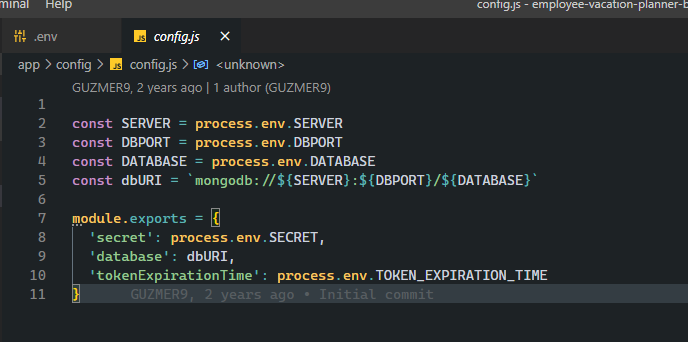
En la imagen podemos ver, en el primer recuadro, cómo importamos el archivo de base de datos, y luego como la inicializamos, pasándole la aplicación expressjs.

A continuación, veamos el archivo de conexión de base de datos:  
  


Podemos ver los diferentes middleware o funciones, que intentarán conectarse a base de datos, tenemos diferentes eventos, "**connected**", para cuando se conecta, "**error**", para cuando da error la conexión y "**disconected**" cuando la conexión falla. Al ejecutar initialize database, el servidor, usando una función asíncrona, intenta conectarse usando la librería "**mongoose**", la cual nos permite conectarnos a bases de datos MongoDB, una base de datos no relacional muy popular.

# Archivo de Configuración de la base de datos

Este es el archivo de configuración de la base de datos.



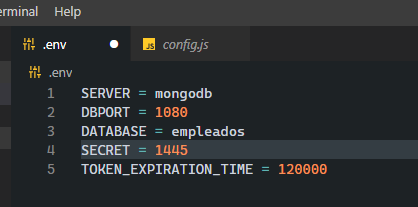
En el podemos ver que se describen las partes que nos ayudarán a conectarnos a una base de datos:

* Server – el servidor de base de datos que debe estar corriendo al momento de ejecución
* DBPort – el puerto abierto en dicho servidor de base de datos, el cual está a la espera de conexiones
* Database – la base de datos en si, su nombre, a la cual nos conectaremos
* DbURI – el uri de la base de datos a conectar, este se forma uniendo las partes previas de forma: `**mongodb://${SERVER}:${DBPORT}/${DATABASE}`**

Procedemos a exportar dicho módulo con el Uri, secret, el cual es básicamente el token o secreto usado para permitir a cualquier aplicación externa comunicarse con esta base de datos, digamos que es una clave de acceso, sin esta, no es posible comunicarse con nuestra base de datos.

Finalmente tenemos un tokenExpirationTime, basicamente un token que indica cuando se vence la sesión, cuando debes pedir al usuario que se logee nuevamente.

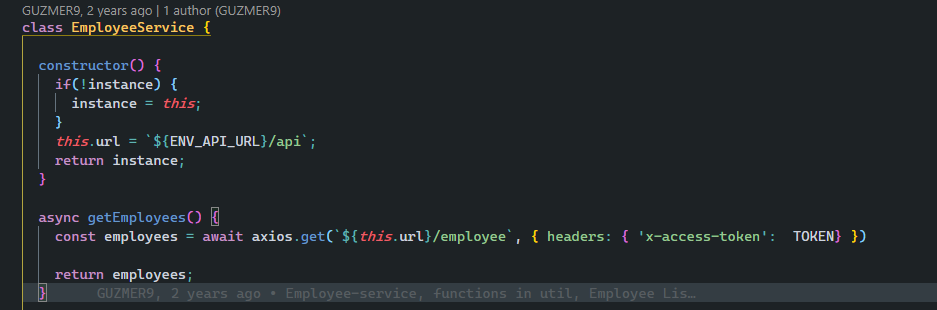
A continuación, vemos el archivo en el cual está la configuración bruta de la base de datos:



Es muy importante que este archivo no llegue a manos de nadie que no tenga acceso al sistema, pues en este archivo no solo se gestionan las variables globales de conexión, sino la clave de acceso y la expiración de tokens, si alguien tiene esta información, su sistema estaría comprometido.

# **¿Cómo conectar el FrontEnd y el Backend?**

A continuación veamos cómo hacemos un simple request a dicha base de datos.



Con ese servicio, podemos obtener los empleados, haciendo un request a nuestra base de datos:  
  
**export const ENV\_API\_URL = '**[**http://localhost:3001**](http://localhost:3001)**';**

Usando ese url, conectamos el frontend con el backend, el cual está corriendo en el puerto 3001 de mi laptop. Esa librería, axios, nos ayuda a manejar requests REST para tener un mejor manejo de los mismos, es necesario pasar el token de seguridad, e indicar el url.