**Desarrollo Prueba New Inntech**

**Andres Felipe Devia O.**

Lo primero es desarrollar las API’s requeridas, en este caso se nos pide que todo sea en un solo directorio así que prodemos con sus comandos correspondientes para crear el entorno de trabajo

-npm init

Cabe aclarar que aquí solo mencionare los pasos agenos al código, ya que el desarrollo del mismo estará en el github

Luego instalamos todas las dependencias necesarias que se podrán consultar en el archivo package.json

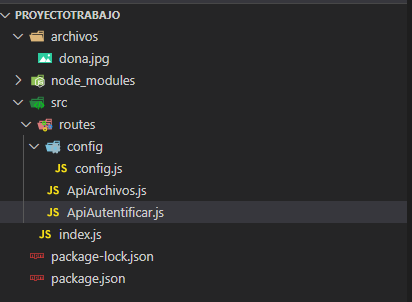
**Api para subir archivos pdf y jpg**

Para esta api el paso a paso no me funciono por algún error que desconozco, así que opte solo por subir cualquier tipo de archivo en su formato original. Para testearlo usamos postman con la ruta definida desde el código

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Podemos ver la ruta y el tipo de archivo que debe ser type=file con key=file, esta key puede dejarse vacia perfectamente, enviamos la imagen y nos retorna el mensaje de archivo subido, y si nos fijamos en la carpeta que destinamos para la subida de imágenes encontraremos la anteriormente mencionada



Podemos notar que la carpeta donde se guarda, tal como lo dice el postman es en el directorio “Archivos”

**Para el frontend**

Debido a que la guía suministrada estaba desactualizada y para la versión actual de vue no se podía ejecutar el vue-cli decidí hacer su frontend con react. Lo instale y ejecute para crear el entorno de trabajo, una vez allí modifique un poco las visuales y cree el botón para subir archivo, y que al subirlo me retorne un mensaje json de que se ha subido un archivo

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

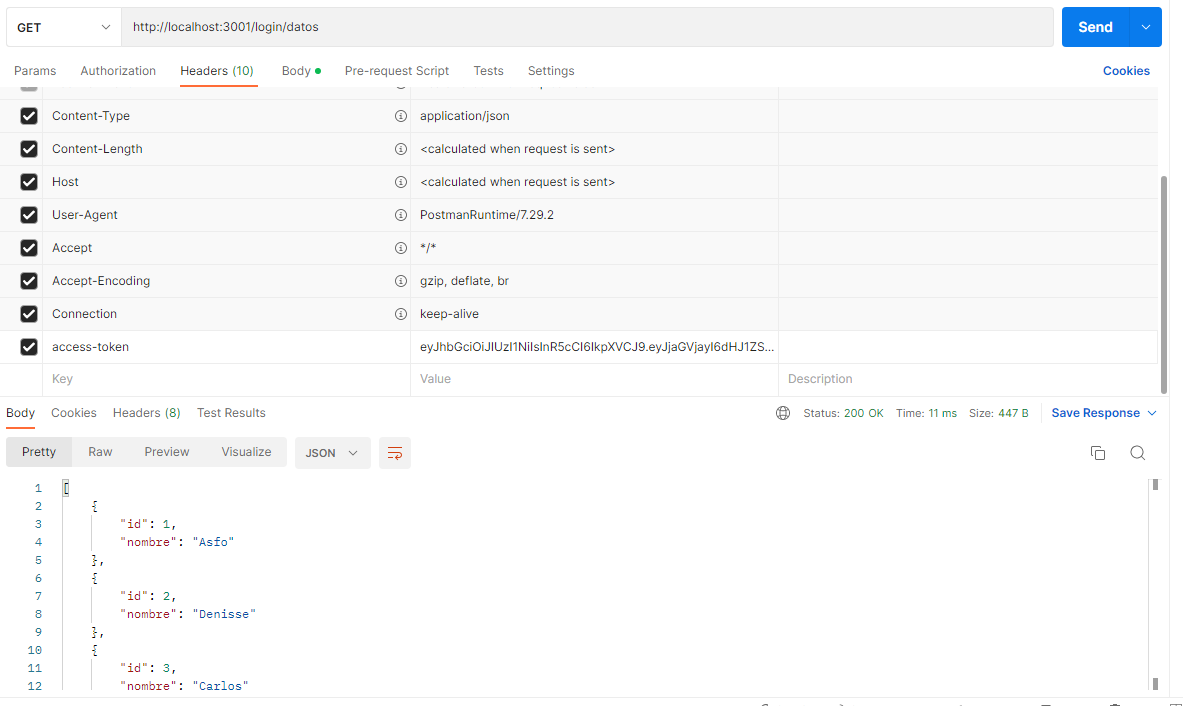
Podemos notar que al darle subir nos retornara un mensaje json que nos dira que el archivo se ha subido, y si revisamos de nuevo la carpeta “archivos” notaremos que en efecto, el archivo se encuentra allí.

**Api para Autenticar usuarios**

De la misma manera que el anterior, estará ubicada en la misma carpeta compartiendo un puerto para hacerlo mas simple, descargamos e instalamos las dependencias necesarias y como en este caso el código entero estaba en la guía solo lo pegamos y lo verificamos en el postman que nos retorne un token cuando el usuario es valido.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Podemos notar que como pusimos el usuario y la contraseña que se definió en el código como valido nos retorno el token, todo esto en archivo json, por tanto si copiamos el token y lo ejecutamos en un header con key “acces-token” y valor del token generado podemos acceder a los datos que nos brinda el token llendo a la ruta datos desde un get

**Para el frontend**

De la misma manera que con la primera API , realice su frontend desde react con fuentes similares, solo le cambie el tipo de input que recibía ya que este recibía un usuario y una contraseña

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Podemos ver que cuando ingresamos el usuario con acceso nos retorna un token, mientras que si ponemos un usuario no valido no nos retornara nada.

**Culminando…**

Para ejecutar estos archivos solo hace falta instalar las dependencias del package y ejecutar las api con “npm run dev” y los directorios de react con cd “NombredeLaCarpetaContenedora” y luego npm “start”

La creación del código en nodejs y react esta mas explicada en los comentarios del directorio, Estoy abierto a retroalimentaciones y sugerencias, gracias de antemano y feliz dia