La carpeta backend contiene todo lo concerniente al servidor, el cual es desarrollado sobre Node.js con la ayuda de express para facilitar en gran medida el trabajo, la base de datos se manejara en mongo DB y usando como lenguaje el JSON para el envió de información entre el cliente (React.js) y el servidor, este conjunto de herramientas se denomina o también es conocido como Stack MERN(Mongodb Express React.js Node.js).

El archivo package.json contiene la información de todas las tecnologías o dependencias usadas para el servidor, así como sus extensiones las cuales son: cors, dotenv, express, mongoose, el archivo package-lock.json contiene información más detallada sobre las versiones

**.env** contiene las variables de entorno que puede usar el servidor

**node\_modules** contiene archivos que node.js descarga e incorpora al proyecto automáticamente, pero, aun así, algunos es necesario invocarlos o requerirlos desde el código

En la carpeta **src** podemos encontrar 5 carpetas más, controllers, models, routes, libs, middlewares y 4 archivos más, los cuales son:

**config.js** que contiene la palabra secreta que ocupa el algoritmo de encriptacion

**app.js** define el puerto al cual se conectará el servidor, también define middlewares y routas

**index.js** requiere de app,js y usa esa información para montar el servidor por medio de la función asíncrona main()

**database.js** es el archivo que se encarga de la conexión a la base de datos en mongodb

En la carpeta **models** se encuentran los esquemas que se usaran en los archivos Json así como los ejemplos.

En la carpeta **routes** encontramos las rutas de cada acción y de cada apartado o modulo, en este caso una para turno, una para usuarios y una para lecciones.

En la carpeta **controllers** están todos aquellos controladores que se encargan de cada acción por modulo, en el archivo se importa el esquema y la ruta correspondiente y se define que datos requiere cada función como lo es eliminar, crear, actualizar entre otras.

En la carpeta **libs** encontraremos un archivo auxiliar que se encargara de verificar y/o crear ciertos elementos que son necesarios por el servidor a la hora de su inicialización, este archivo se llama initialSetup.js.

En la capeta **middlewares** encontraremos las funciones de confirmación y autorización para los tokens y otras herramientas, estos se importan a un archivo index.js para manejar de manera más fácil aquellas funciones, permitiendo así importar solo un archivo que llama a los archivos donde están dichas funciones.

**Dependencias:**

**express** facilita en gran manera la comunicación con el framework node.js

**cors** ayuda a la comunicación entre servidores, en este caso node.js y react.js

**mongoose** librería para node.js que es muy útil a la hora de usar mongodb

**dotenv** habilita la creación de variables de entorno que servirán para aumentar la seguridad del servidor

**morgan** muestra las consultas que llegan a la base de datos por consola.

**helmet** aumenta la seguridad del servidor no mostrando datos de más al cliente.