

# TRABAJO INTEGRADOR DE NODE JS

**ALUMNO:** Palomo José Facundo

**Clúster :** 2

El proyecto consiste en una aplicación que gestiona librerías , cada una puede tener asociada o no una cantidad determinada de libros. Las acciones que se pueden realizar es obtener todas las librerías disponibles con el estado “a” (activo) , obtener una librería específica por medio de su “id” , crear una nueva , modificar una existente o eliminar. Estas mismas opciones se encuentran disponibles para los libros relacionados a su librería correspondiente. En caso de que se quiera eliminar un registro se hará de manera lógica y pasara a estado “e” (eliminado) y ya no estará visible la próxima vez que se realice la consulta.

No obstante para realizar algunas operaciones el usuario tendrá que estar logueado , en caso de que intente realizar una acción sin autenticarse el programa devolverá un mensaje de “usuario no autorizado”.

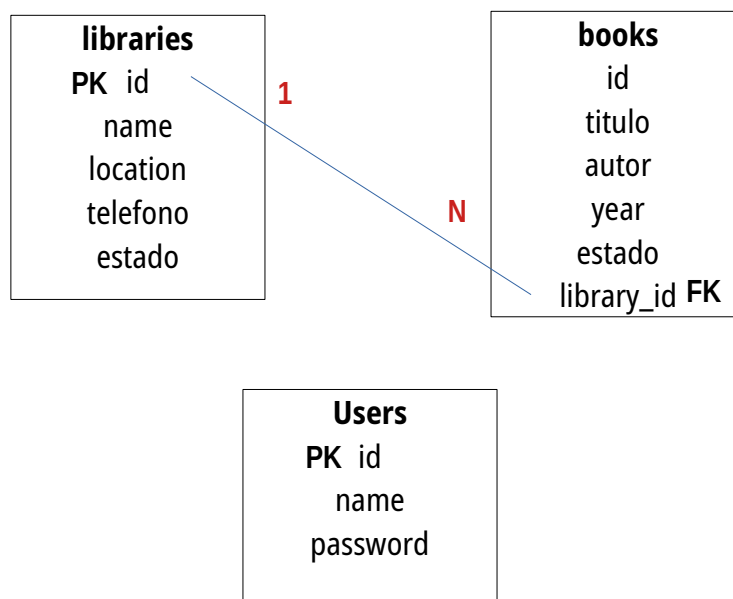
## Estructura del proyecto

Lo primero que realice fue preparar una base de datos en MYSQL con el nombre “integrador” , dentro de la misma cree tres tablas (users,libraries y books)

**users** ( id , name , password)

**libraries** ( id, name , location , telefono , estado )

**books** ( id , isbn, titulo , autor ,year , library\_id, estado)



A continuación procedí a instalar las dependencias de express, sequelize y el driver de la base de datos que iba a usar , en mi caso fue mysql.

Para facilitar el desarrollo y la visualización opte por instalar la dependencia de desarrollo de nodemon que actualiza el servidor cada vez que identifica cambios

Seguido de esto procedí a crear y configurar el repositorio de la api con el comando de "git init" y luego a crear el archivo principal index.html donde importe express y lo almacene en una variable "app" para su posterior uso.

Configure el puerto que iba a utilizar y a partir del método .listen levante la aplicación en el servidor para revisar que la conexión se realizara de manera correcta.

Luego realice la estructura de mi proyecto en la carpeta "src" donde almacene todo mi código fuente a excepción del archivo index.html .

### **Dentro de "src" cree las siguientes carpetas:**

**db:** En la carpeta "database" guarde todo la configuración que se necesita nuestra aplicación para acceder a la información de la base de datos, como el nombre , usuario y contraseña.

**models:** En esta carpeta definí los archivos de los atributos y tipo de datos de las entidades de la aplicación (users, libraries y books), que deben ser iguales a la información de sus tablas correspondientes en la base de datos. (Ej: id, name,lastname,email,password,etc)

**routes :** En esta carpeta configure las rutas o paths de cada una de las consultas que se van a realizar. Cada una seguida del middleware y controlador correspondiente.

**auth:** En esta carpeta se encuentran todos los archivos que contienen la configuración para la autenticación del usuario

**middlewares:** Esta carpeta puse los archivos que definen las funciones que se ejecutan antes o después de que se maneje una solicitud HTTP por el controlador correspondiente.

**controllers:** En esta carpeta se encuentran los archivos que encargan de procesar, recibir y responder las peticiones , es decir que su tarea principal va a ser gestionar las solicitudes HTTP.

**services:** En esta carpeta se encuentran los archivos que se encargan de la lógica de negocio y de interactuar con la base de datos , en este caso seria realizar las consultas básicas de un crud:

GET: Leer

POST: Insertar

PUT: Editar

DELETE: Borrar