

## MANUAL DE FUNCIONAMIENTO APP BIBLIOTECA

Se entregan dos carpetas comprimidas, en una de ellas que es biblio se entrega el backend y en la otra se entrega parte del front en angular. para ambos proyectos se deben descargar sus dependencias en el caso del backend se utilizó Maven y en el de front angular CLI.

En el caso del front se puede utilizar el comando `npm install` para sus dependencias y `ng serve` para su despliegue.

En el back se recomienda usar Eclipse para hacer el update de Maven y así leer el `pom.xml`

### FUNCIONALIDADES BACKEND DE USUARIOS

Nota: después de subir el API se debe crear un usuario en la base de datos con rol 1

URL: `http://localhost:8080`

MOTOR DE BD: MySQL

- CREAR USUARIO (POST)  
PATH: `/Registrar`  
BODY: `{ "Nombre": "",  
          "email": "",  
          "pass": ""  
}`
- USUARIO login (POST)  
PATH: `/Login`  
BODY: `{  
      "email": "",  
      "pass": ""  
}`
- USUARIO LOGOUT (PATCH)  
PATH: `/Login/"id"`  
BODY: `{}`  
RETURN: `false/true`

Nota: los métodos login y registrar retornan un JSON con el usuario con su ID si es exitosa la petición y si no lo es retorna un JSON con todos sus atributos en `null`.

## FUNCIONALIDADES BACKEND DE LIBROS

ULR: <http://localhost:8080>

MOTOR DE BD:mysql

- CREAR LIBRO (POST)  
PATH: Libro /"id"  
BODY:{nombre:"",  
"autor": "",  
"categoría": "",  
"cant":int,  
"disp":int,  
"tarifa":int  
}  
}
- actualizar (PATCH)  
PATH: /Libro/"id"  
BODY: {  
nombre:"",  
"autor": "",  
"categoría": "",  
"cant":int,  
"disp":int,  
"tarifa":int  
}  
}
- Eliminar LIBRO (DELETE)  
PATH: /Libro/"id"  
BODY: {"id":int}  
RETURN: false/true
- listar (GET)  
PATH: /Libro/query  
BODY: {}  
RETURN: lista de libros

NOTA: el id solicitado en el path es el id del usuario que debe ser un usuario con roll = true es decir es administrador de lo contrario no funcionara.