Trabajo práctico del Módulo 4

*TP Integrador Modulo 3 y 4: Where'is my books?*

*Se trata de un proyecto de desarrollo backend en NodeJS, utilizando API Rest y base de datos MySQL para conocer si los libros del usuario se encuentran en su biblioteca o prestados. En caso de estar prestado, a quien se los presto.*

*Se requiere conocer*

*De la persona a prestar los libros el nombre, apellido, email y alias. El email debe ser unico. Todos los datos son requeridos.*

*De los generos de los libros, solo los nombres, el campo nunca puede ser vacio o nulo y no pueden repetirse las categorias.*

*De los libros, el nombre, una descripcion, su categoria y la persona a la cual se le ha prestado el libro. Para representar que un libro se encuentra en la biblioteca se puede utilizar cualquiera de las siguientes estrategias: null para libros en la biblioteca en el campo de persona\_id, que el usuario se encuentre ingresado como una persona mas.*

*Se solicita verificar que el usuario no envie los campos requeridos solo con espacios en blanco.*

*Para asegurar los aciertos de las busquedas, se sugiere el guardado en mayusculas de todos los campos alfanumericos y recordar hacer las verificaciones teniendo en cuenta esto.*

*El servidor debera responser a la siguiente documentacion de API*

*CATEGORIA*

*POST '/categoria' recibe: {nombre: string} retorna: status: 200, {id: numerico, nombre: string} - status: 413, {mensaje: <descripcion del error>} que puede ser: "faltan datos", "ese nombre de categoria ya existe", "error inesperado"*

*GET '/categoria' retorna: status 200  y [{id:numerico, nombre:string}]  - status: 413 y []*

*GET '/categoria/:id' retorna: status 200 y {id: numerico, nombre:string} - status: 413, {mensaje: <descripcion del error>} que puede ser: "error inesperado", "categoria no encontrada"*

*DELETE '/categoria/:id' retorna: status 200 y {mensaje: "se borro correctamente"} - status: 413, {mensaje: <descripcion del error>} que puede ser: "error inesperado", "categoria con libros asociados, no se puede eliminar", "no existe la categoria indicada"*

*No se debe implementar el PUT*

*PERSONA*

*POST '/persona' recibe: {nombre: string, apellido: string, alias: string, email: string} retorna: status: 200, {id: numerico, nombre: string, apellido: string, alias: string, email: string} - status: 413, {mensaje: <descripcion del error>} que puede ser: "faltan datos", "el email ya se encuentra registrado", "error inesperado"*

*GET '/persona' retorna status 200 y [{id: numerico, nombre: string, apellido: string, alias: string, email; string}] o bien status 413 y []*

*GET '/persona/:id' retorna status 200 y {id: numerico, nombre: string, apellido: string, alias: string, email; string} - status 413 , {mensaje: <descripcion del error>} "error inesperado", "no se encuentra esa persona"*

*PUT '/persona/:id' recibe: {nombre: string, apellido: string, alias: string, email: string} el email no se puede modificar. retorna status 200 y el objeto modificado o bien status 413, {mensaje: <descripcion del error>} "error inesperado", "no se encuentra esa persona"*

*DELETE '/persona/:id' retorna: 200 y {mensaje: "se borro correctamente"} o bien 413, {mensaje: <descripcion del error>} "error inesperado", "no existe esa persona", "esa persona tiene libros asociados, no se puede eliminar"*

*LIBRO*

*POST '/libro' recibe: {nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null} devuelve 200 y {id: numero, nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null} o bien status 413,  {mensaje: <descripcion del error>} que puede ser "error inesperado", "ese libro ya existe", "nombre y categoria son datos obligatorios", "no existe la categoria indicada", "no existe la persona indicada"*

*GET '/libro' devuelve 200 y [{id: numero, nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null}] o bien 413, {mensaje: <descripcion del error>} "error inesperado"*

*GET '/libro/:id' devuelve 200 {id: numero, nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null} y status 413, {mensaje: <descripcion del error>} "error inesperado", "no se encuentra ese libro"*

*PUT '/libro/:id' y {id: numero, nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null} devuelve status 200 y {id: numero, nombre:string, descripcion:string, categoria\_id:numero, persona\_id:numero/null} modificado o bien status 413, {mensaje: <descripcion del error>} "error inesperado",  "solo se puede modificar la descripcion del libro*

*PUT '/libro/prestar/:id' y {id:numero, persona\_id:numero} devuelve 200 y {mensaje: "se presto correctamente"} o bien status 413, {mensaje: <descripcion del error>} "error inesperado", "el libro ya se encuentra prestado, no se puede prestar hasta que no se devuelva", "no se encontro el libro", "no se encontro la persona a la que se quiere prestar el libro"*

*PUT '/libro/devolver/:id' y {} devuelve 200 y {mensaje: "se realizo la devolucion correctamente"} o bien status 413, {mensaje: <descripcion del error>} "error inesperado", "ese libro no estaba prestado!", "ese libro no existe"*

*DELETE '/libro/:id' devuelve 200 y {mensaje: "se borro correctamente"}  o bien status 413, {mensaje: <descripcion del error>} "error inesperado", "no se encuentra ese libro", "ese libro esta prestado no se puede borrar"*

*Se evaluara la correcta implementacion, el uso de try/catch, async-await, cors, express, validacion de los datos solicitados, manejo de errores.*

*El tp se realizara en grupos de 5 o 6 personas que seran elegidas al azar por el campus*

*La entrega se realizara en formato ZIP y de deberan incluir: package.json, package-lock.json, app.json NO INCLUIR node\_modules (va a desconar puntaje el envio de esta carpeta).*

*Solo 1 entrega por equipo. Se debe incluir un archivo txt con el numero de grupo y el nombre y apellido de los integrantes.*