Examen Práctico

Backend - NodeJS

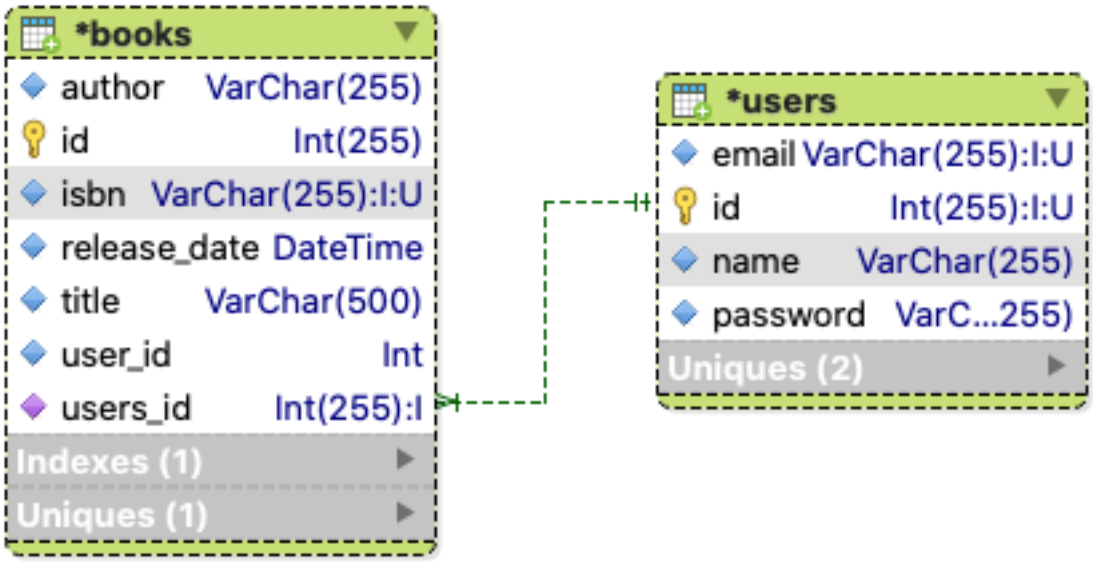
DIGITAL IGNITION

ID: MX01-35	Tipo:	Examen Técnico
Versión: 2.1 Noviembre, 2023	Tecnología:	Node JS
	Equipo:	Back

PROBLEMA POR RESOLVER

Una librería necesitar mantener un registro de los usuarios que acceden a su plataforma y registrar los libros disponibles en su acervo digital. Para ello, se solicita la creación de una serie de servicios REST que permitan el registro de usuarios, inicio de sesión, creación de libros, consulta de los libros disponibles, actualización y borrado de los mismos.

El administrador de base de datos a cargo presentó el siguiente diagrama ER que muestra la estructura que tendrá la base de datos que usará el sistema:



ACTIVIDADES PARA EVALUAR

1. Desarrolle un servicio en el lenguaje de programación **Node JS** que de respuesta a peticiones HTTP.
2. Cree una base de datos en el manejador de su preferencia siguiendo el diagrama presentado anteriormente.
3. Desarrolle los siguientes servicios para el control de usuarios:
    a. **POST /auth/register**: Este endpoint recibirá los campos name, email y password en su body y deberán ser almacenados en la base de datos. Si la operación es exitosa, deberá devolver un true como respuesta. En caso contrario, devolver un error.
    b. **POST /auth/login:** Este endpoint recibe los campos email y password. Si el usuario existe, el servicio deberá generar un token de autenticación y

devolverlo en la respuesta. Si el usuario no se encuentra registrado o la contraseña no coincide, devolver un error.

c. **GET /books:** Este endpoint deberá devolver un arreglo con todos los libros registrados en la base de datos y el usuario que lo registró. Recuerda no enviar la contraseña en la respuesta.
d. **POST /books:** Este endpoint recibirá los campos isbn, title, author y release_date en su body y creará un nuevo registro en la base de datos ÚNICAMENTE si el ISBN no ha sido registrado antes. Deberá recibir, además, el token de autenticación del usuario para registrar el libro a su nombre. Si el proceso es exitoso, devolver el id del nuevo libro. En caso contrario, devolver un error.
e. **DELETE /books/:id**: Este endpoint recibe el id del libro en la url y el token de autenticación del usuario y elimina el registro del libro de la base de datos. El endpoint deberá devolver true si la operación fue exitosa o error en caso de que el libro no exista, el usuario no esté autenticado, etc.

CONSIDERACIONES Y RECOMENDACIONES

1. En el endpoint **POST /auth/register** todos los campos son obligatorios. Si algún campo no es enviado, devuelva un error.
2. Almacene la contraseña del usuario de forma encriptada en la base de datos.
3. Haga uso de JWT para la creación y control de los tokens de autenticación de los usuarios.
4. El token de autenticación de un usuario recién loggeado deberá tener un periodo de expiración de 24 horas.
5. Podrá usar cualquier framework que desee.
6. Se revisará la claridad y estructuración del código.

ENTREGA

1. Crea un repositorio en GitHub o GitLab que sea público y sube tu código.
2. Exporte e incluya la base de datos en formato .sql que creó en la carpeta raíz de su repositorio.
3. Envía el enlace del repositorio a los siguientes correos: jose.esparza@digitalignition.com.mx, joel.carranza@digitalignition.com.mx, y roberto@digitalignition.com.mx, Si tienes algo que decir, usa el cuerpo del correo para expresar lo que quieras.
4. El tiempo máximo para la entrega es de 36 horas, a partir del envío de la prueba.