

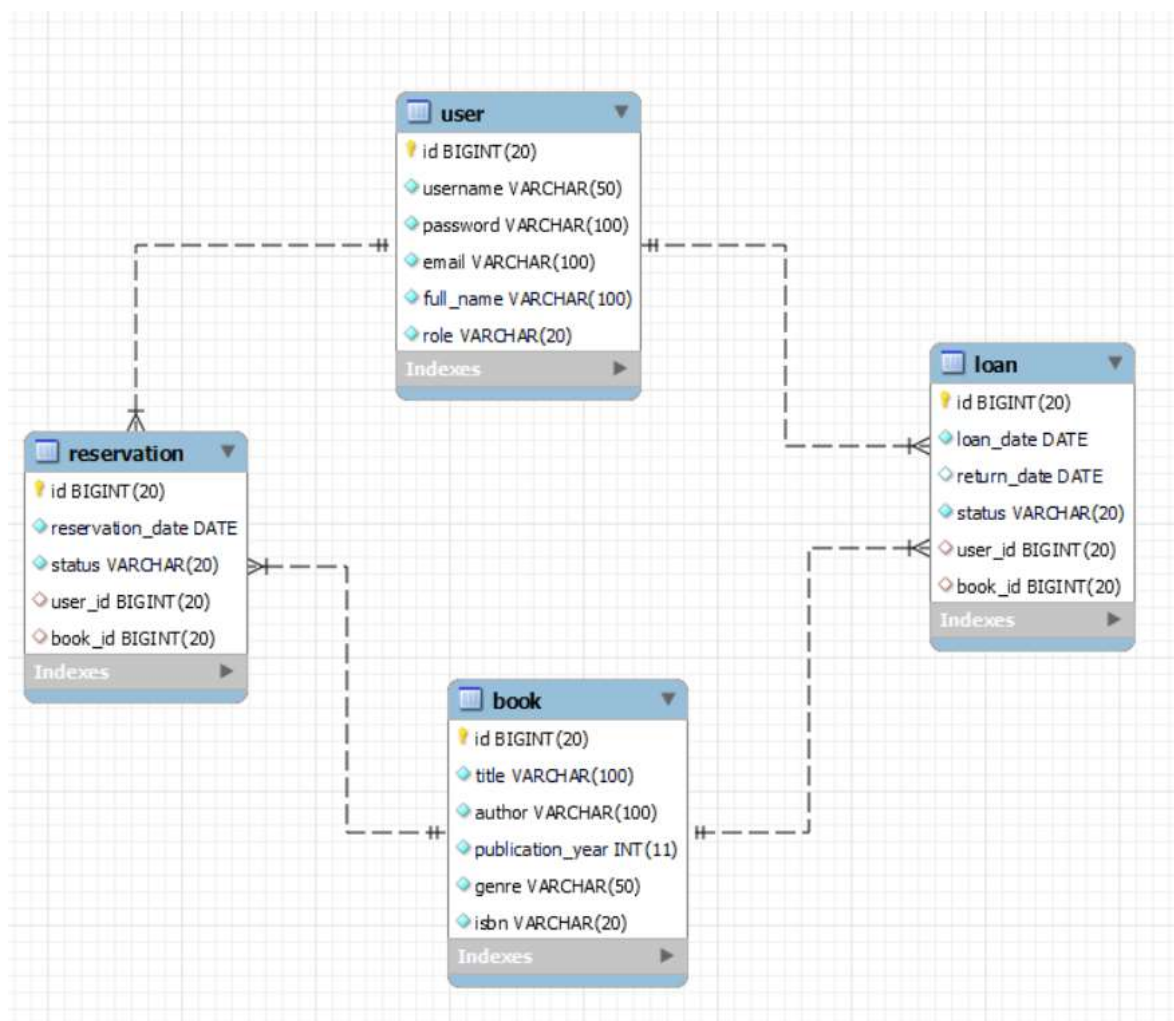
## Gestión de Biblioteca en Línea

**Introducción:** La biblioteca virtual "LibrosYa" necesita un sistema de gestión que permita a los usuarios buscar, reservar, y prestar libros, así como gestionar su cuenta personal.

**Objetivo:** Desarrollar un sistema de gestión de biblioteca en línea que permita almacenar y gestionar eficientemente los libros, usuarios, préstamos y reservas.

### Modelo Entidad-Relación Propuesto:

1. **User-Book:** Un usuario puede reservar muchos libros (uno a muchos) y cada libro puede ser reservado por muchos usuarios (muchos a muchos).
2. **Book-Loan:** Un libro puede ser prestado muchas veces (uno a muchos) y cada préstamo está asociado a un solo libro (muchos a uno).
3. **User-Loan:** Un usuario puede tener muchos préstamos y cada préstamo está asociado a un solo usuario.



### Criterios de aceptación generales:

- Todos los endpoints deben estar documentados en Swagger.
- Se deben manejar DTOs para la respuesta y entrada de datos utilizando MapStruct.
- La información debe venir de la base de datos.
- Todas las entradas deben tener validaciones para que no afecten a la base de datos.

- Si no se encuentra algún registro, se debe manejar el error y responder con un mensaje y su respectivo estatus de error, evitando los errores 500.
- Las relaciones con JPA deben ser bidireccionales.

## Endpoints Propuestos:

### Endpoints para Usuarios

#### 1. Registrar Usuario

- **POST /users**
- **Descripción:** Crear un nuevo usuario.
- **Datos de entrada:** username, password, email, full\_name, role
- **Datos de salida:** solo los datos del usuario

#### 2. Obtener Información del Usuario

- **GET /users/{user\_id}**
- **Descripción:** Obtener información detallada de un usuario específico.
- **Datos de salida:** datos del usuario con todas sus relaciones con las demás entidades

#### 3. Actualizar Usuario

- **PUT /users/{user\_id}**
- **Descripción:** Actualizar la información de un usuario.
- **Datos de entrada:** username, password, email, full\_name
- **Datos de salida:** solo los datos del usuario

#### 4. Eliminar Usuario

- **DELETE /users/{user\_id}**
- **Descripción:** Eliminar un usuario.

### Endpoints para Libros

#### 5. Crear Libro

- **POST /books**
- **Descripción:** Crear un nuevo libro.
- **Datos de entrada:** title, author, publication\_year, genre, isbn
- **Datos de salida:** solo los datos del libro

#### 6. Obtener Información del Libro

- **GET /books/{book\_id}**
- **Descripción:** Obtener información detallada de un libro específico.
- **Datos de salida:** datos del libro con todas sus relaciones con las demás entidades

#### 7. Actualizar Libro

- **PUT /books/{book\_id}**
- **Descripción:** Actualizar la información de un libro.
- **Datos de entrada:** title, author, publication\_year, genre, isbn
- **Datos de salida:** solo los datos del libro



## 8. Eliminar Libro

- **DELETE /books/{book\_id}**
- **Descripción:** Eliminar un libro.

## 9. Obtener Todos los Libros

- **GET /books**
- **Descripción:** Obtener una lista de todos los libros.
- **Datos de salida:** Solo los datos de cada libro
- **Busqueda:** búsqueda por título, autor y genero

## Endpoints para Préstamos

### 10. Crear Préstamo

- **POST /loans**
- **Descripción:** Crear un nuevo préstamo.
- **Datos de entrada:** user\_id, book\_id, loan\_date, return\_date
- **Datos de salida:** datos del préstamo con todas sus relaciones

### 11. Obtener Información del Préstamo

- **GET /loans/{loan\_id}**
- **Descripción:** Obtener información detallada de un préstamo específico.
- **Datos de salida:** datos del préstamo con todas sus relaciones

### 12. Actualizar Préstamo

- **PUT /loans/{loan\_id}**
- **Descripción:** Actualizar la información de un préstamo.
- **Datos de entrada:** loan\_date, return\_date, status
- **Datos de salida:** datos del préstamo con todas sus relaciones

### 13. Eliminar Préstamo

- **DELETE /loans/{loan\_id}**
- **Descripción:** Eliminar un préstamo.

### 14. Obtener Todos los Préstamos de un Usuario

- **GET /users/{user\_id}/loans**
- **Descripción:** Obtener todos los préstamos de un usuario específico.
- **Datos de salida:** datos del préstamo con todas sus relaciones

## Endpoints para Reservas

### 15. Crear Reserva

- **POST /reservations**
- **Descripción:** Crear una nueva reserva.
- **Datos de entrada:** user\_id, book\_id, reservation\_date
- **Datos de salida:** todos los datos de la reserva con sus relaciones

### 16. Obtener Información de la Reserva

- **GET /reservations/{reservation\_id}**
- **Descripción:** Obtener información detallada de una reserva específica.
- **Datos de salida:** todos los datos de la reserva con sus relaciones

### 17. Actualizar Reserva

- **PUT /reservations/{reservation\_id}**
- **Descripción:** Actualizar la información de una reserva.
- **Datos de entrada:** reservation\_date, status
- **Datos de salida:** todos los datos de la reserva con sus relaciones

### 18. Eliminar Reserva

- **DELETE /reservations/{reservation\_id}**
- **Descripción:** Eliminar una reserva.



#### 19. Obtener Todas las Reservas de un Usuario

- **GET /users/{user\_id}/reservations**
- **Descripción:** Obtener todas las reservas de un usuario específico.
- **Datos de salida:** todos los datos de la reserva con sus relaciones

#### 20. Obtener Todas las Reservas de un Libro

- **GET /books/{book\_id}/reservations**
- **Descripción:** Obtener todas las reservas de un libro específico.
- **Datos de salida:** todos los datos de la reserva con sus relaciones