

## Tabla de contenido

Descripción del Proyecto .....	2
Requerimientos Técnicos .....	2
Endpoints de la API .....	2
Modelo de Datos .....	3
Evaluación .....	3

## Descripción del Proyecto

Estás a cargo de desarrollar una API RESTful para administrar una biblioteca virtual. Esta API debe permitir a los usuarios llevar a cabo diversas operaciones relacionadas con la gestión de libros y préstamos. Además, se requiere integrar una pasarela de pagos para que los usuarios puedan comprar libros.

### Requerimientos Técnicos

1. Utiliza Node.js y Express.js para el backend.
2. Emplea PostgreSQL como base de datos, junto con un ORM como Prisma, Sequelize o TypeORM.
3. Implementa TypeScript para la codificación del proyecto.
4. Sigue buenas prácticas de desarrollo, incluyendo patrones de diseño, manejo de errores y seguridad.
5. Implementa la autenticación utilizando JSON Web Tokens (JWT).
6. Documenta la API utilizando Swagger u otra herramienta similar.
7. Asegúrate de incluir pruebas unitarias e integración, utilizando herramientas como Jest, Mocha, o Chai.
8. Despliega la API REST en una plataforma de producción, como Heroku, Digital Ocean, AWS, o cualquier otra de tu elección.
9. Implementa un endpoint para la compra de libros, utilizando una pasarela de pagos como Stripe o una alternativa.

### Endpoints de la API

1. Autenticación
  - POST `/api/auth/register` - Registro de usuarios.
  - POST `/api/auth/login` - Inicio de sesión de usuarios.
2. Libros
  - 🔍 GET `/api/books` - Obtener todos los libros.
  - 🔍 GET `/api/books/:id` - Obtener un libro por su ID.
  - 🔍 POST `/api/books` - Crear un nuevo libro.
  - 🔍 PUT `/api/books/:id` - Actualizar la información de un libro.
  - 🔍 DELETE `/api/books/:id` - Eliminar un libro.
3. Préstamos
  - 🔍 POST `/api/loans/:userId/:bookId` - Prestar un libro a un usuario.
  - 🔍 PUT `/api/loans/:userId/:bookId` - Devolver un libro prestado por un usuario.

#### 4. Compra de Libros

📌 POST `/api/purchase` - Comprar un libro utilizando una pasarela de pagos.

#### Modelo de Datos

1. User (id, name, email, password, createdAt, updatedAt).
2. Book (id, title, author, publicationYear, createdAt, updatedAt).
3. Loan (id, userId, bookId, loanDate, returnDate).

#### Evaluación

**Se evaluará la claridad, eficiencia y mantenibilidad del código, así como la implementación de las características requeridas y el cumplimiento de los estándares mencionados.**