

BookShop

Vas a construir una API REST para una plataforma de compra/venta de libros.

La plataforma ejecuta procesos automáticos para analizar la actividad y notificar a los usuarios.

Entidades

User { id email password createdAt }	Reglas <ul style="list-style-type: none">• Email único• Password hasheado
---	---

Book { id title description price author status: "PUBLISHED" "SOLD" ownerId soldAt createdAt }	Reglas <ul style="list-style-type: none">• Un libro pertenece a un único usuario.• Los libros en estado SOLD no son visibles públicamente.• Los libros creados tienen el estado PUBLISHED.• Solo el dueño del libro puede editar y eliminar el libro.• La propiedad soldAt es null mientras el libro esté en venta.• Los usuarios no pueden comprar sus propios libros.
--	--

Endpoints requeridos

Autenticación

La autenticación de usuarios tendrá que realizarse mediante el uso de token JWT.

1. POST authentication/signup → Creación de usuarios
2. POST authentication/signin → Identificación de usuarios

Libros

Endpoints privados

1. POST /books → Crear libro
2. PUT /books/:id → Editar libro. Sólo se puede editar el título, descripción, precio y autor.
3. POST /books/:id/buy → Comprar libro. Cuando se compra un libro, pasa a estar en estado SOLD
4. GET /me/books → Consultar mis libros

Endpoint público

1. GET /books → Consultar libros
 - a. Solo devolverá libros con estado PUBLISHED
 - b. Búsqueda por título o autor
 - c. Endpoint paginado

Casos de uso

Compra de libro

Cuando un usuario compre un libro, el vendedor recibirá un email siendo notificado de ello.

Sugerencia de bajada de precio

Una vez a la semana, se ejecutará una tarea programada que revisará los libros que llevan publicados más de 7 días. Por cada uno de ellos, el vendedor recibirá un email para sugerirles que bajen el precio.

Testing

Habrá que testear, como mínimo, los siguientes endpoints:

- POST /books/:id/buy
- POST /books
- GET /books