

INF 780 VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE

Examen de Primer Parcial

Clona el proyecto BooksApp de la dirección:

<https://github.com/HuascarFedor/booksapp>

1. Configura las pruebas en Thunder Client para probar los end points del proyecto.

1) Crear libro — POST /books

Thunder

- **Method:** POST
- **URL:** {{baseUrl}}/books
- **Body (JSON):**

```
{
  "titulo": "Clean Code",
  "autor": "Robert C. Martin",
  "isbn": "9780132350884",
  "anioPublicacion": 2008,
  "categoria": "tecnico",
  "stock": 5
}
```

Respuesta esperada (201 Created):

```
{
  "id": 1,
  "titulo": "Clean Code",
  "autor": "Robert C. Martin",
  "isbn": "9780132350884",
  "anioPublicacion": 2008,
  "categoria": "tecnico",
  "stock": 5,
  "creadoEn": "2025-10-26T21:00:00.000Z",
  "actualizadoEn": "2025-10-26T21:00:00.000Z"
}
```

cURL

```
curl -X POST "{{baseUrl}}/books" \
  -H "Content-Type: application/json" \
  -d '{"titulo":"Clean Code","autor":"Robert C. Martin","isbn":"9780132350884","anioPublicacion":2008,"categoria":"tecnico","stock":5}'
```

Errores comunes

- 400 Bad Request: falta titulo o autor, anioPublicacion fuera de rango, isbn inválido.
- 409 Conflict: ya existe un libro con **mismo titulo + autor**.

2) Listar libros — GET /books (con paginación y filtros)

Thunder

- **Method:** GET

- **URL:** `{{baseUrl}}/books?q=clean&conStock=1&page=1&limit=10`

Respuesta esperada (200 OK):

```
{
  "data": [
    {
      "id": 1,
      "titulo": "Clean Code",
      "autor": "Robert C. Martin",
      "isbn": "9780132350884",
      "anioPublicacion": 2008,
      "categoria": "tecnico",
      "stock": 5,
      "creadoEn": "2025-10-26T21:00:00.000Z",
      "actualizadoEn": "2025-10-26T21:00:00.000Z"
    }
  ],
  "total": 1,
  "page": 1,
  "limit": 10
}
```

Parámetros útiles

- q: busca en titulo y autor (ej. q=martin)
- categoria: ficcion | no_ficcion | tecnico | academico | infantil
- anioDesde, anioHasta: enteros (ej. anioDesde=2000&anioHasta=2010)
- conStock=1 (o true): solo con stock > 0
- page, limit: paginación

cURL

`curl "{{baseUrl}}/books?q=clean&conStock=1&page=1&limit=10"`

3) Obtener detalle — GET /books/:id

Thunder

- **Method:** GET
- **URL:** `{{baseUrl}}/books/1`

Respuesta esperada (200 OK):

```
{
  "id": 1,
  "titulo": "Clean Code",
  "autor": "Robert C. Martin",
  "isbn": "9780132350884",
  "anioPublicacion": 2008,
  "categoria": "tecnico",
  "stock": 5,
  "creadoEn": "2025-10-26T21:00:00.000Z",
  "actualizadoEn": "2025-10-26T21:00:00.000Z"
}
```

cURL

`curl "{{baseUrl}}/books/1"`

Errores comunes

- 404 Not Found: id inexistente.

4) Actualizar parcialmente — PATCH /books/:id

Thunder

- **Method:** PATCH
- **URL:** {{baseUrl}}/books/1
- **Body (JSON):**

```
{  
  "stock": 12,  
  "categoria": "tecnico"  
}
```

Respuesta esperada (200 OK):

```
{  
  "id": 1,  
  "titulo": "Clean Code",  
  "autor": "Robert C. Martin",  
  "isbn": "9780132350884",  
  "anioPublicacion": 2008,  
  "categoria": "tecnico",  
  "stock": 12,  
  "creadoEn": "2025-10-26T21:00:00.000Z",  
  "actualizadoEn": "2025-10-26T21:05:30.000Z"  
}
```

cURL

```
curl -X PATCH "{{baseUrl}}/books/1" \  
-H "Content-Type: application/json" \  
-d '{"stock":12,"categoria":"tecnico"}
```

Errores comunes

- 404 Not Found: id inexistente.
- 409 Conflict: cambiar titulo o autor a uno que choca con otro registro existente.
- 400 Bad Request: stock negativo, anioPublicacion fuera de rango, isbn inválido.

5) Eliminar — DELETE /books/:id

Thunder

- **Method:** DELETE
- **URL:** {{baseUrl}}/books/1

Respuesta esperada

{ } // o vacío con 200/204, según tu implementación

cURL

```
curl -X DELETE "{{baseUrl}}/books/1"
```

Errores comunes

- 404 Not Found: id inexistente.

2. Escribe las PRUEBAS UNITARIAS para `books.service.ts`
3. Escribe las PRUEBAS UNITARIAS para `books.controller.ts`

Entregables:

- Muestra tus pruebas de Thunder Client en el laboratorio
- Crea un repositorio, sube tu proyecto y envía el enlace mediante WhatsApp.