



INSTITUTO PROFESIONAL
SAN SEBASTIAN

Evaluación Sumativa Unidad 1

INTRODUCCIÓN CLOUD

Profesor
Víctor Cofre Farias
Integrantes
Sebastian lucero
Osvaldo Maulén

ACREDITACIÓN 2025
Construimos juntos una
formación de calidad





1. Introducción

El presente trabajo tiene como objetivo el desarrollo de una API REST para la gestión de álbumes de láminas coleccionables, permitiendo administrar álbumes y sus respectivas láminas de manera eficiente.

El sistema desarrollado permite realizar operaciones CRUD sobre los álbumes, agregar láminas de forma individual o por lote, y realizar consultas específicas para identificar láminas faltantes y repetidas dentro de un álbum.

Para la implementación se utilizaron tecnologías backend modernas como Spring Boot y MongoDB, incorporando validaciones, manejo de errores y una arquitectura por capas que facilita la mantenibilidad y escalabilidad del sistema.

2. Desarrollo

La aplicación fue desarrollada utilizando Spring Boot como framework principal, lo que permitió una configuración rápida y una correcta integración con Spring Data MongoDB para la persistencia de datos.

MongoDB fue seleccionada como sistema de base de datos debido a su modelo orientado a documentos, el cual se adapta de forma natural a la estructura del sistema, donde un álbum contiene una lista de láminas embebidas.

La conexión a la base de datos se realizó mediante el archivo de configuración **application.properties**, centralizando los parámetros de conexión y facilitando la administración del entorno.





```
_id: ObjectId('694b59b14efd9cf1d1d7f2fc')
nombre : "Album Mundial 2026"
imagenUrl : "https://ejemplo.com/album2.png"
fechalanzamiento : 2026-12-24T03:00:00.000+00:00
tipoLaminas : "Futbol"
laminas : Array (empty)
createdAt : 2025-12-24T03:10:41.320+00:00
updatedAt : 2025-12-24T03:22:37.293+00:00
_class : "com.laminas.dealbumes.model.Album"

_id: ObjectId('694b5a1c4ef9cf1d1d7f2fd')
nombre : "Dragon Ball"
imagenUrl : "https://ejemplo.com/album.png"
fechalanzamiento : 2022-11-01T03:00:00.000+00:00
tipoLaminas : "Anime"
laminas : Array (empty)
createdAt : 2025-12-24T03:12:28.066+00:00
updatedAt : 2025-12-24T03:12:28.066+00:00
_class : "com.laminas.dealbumes.model.Album"

_id: ObjectId('694b5ea84efd9cf1d1d7f303')
nombre : "Album Mundial 2030 - Actualizado"
imagenUrl : "https://ejemplo.com/album2.png"
fechalanzamiento : 2030-12-24T03:00:00.000+00:00
tipoLaminas : "Futbol"
laminas : Array (4)
0: Object
1: Object
2: Object
3: Object
createdAt : 2025-12-24T03:31:52.906+00:00
updatedAt : 2025-12-24T03:36:21.054+00:00
_class : "com.laminas.dealbumes.model.Album"
```

Figura 1. Base de datos MongoDB utilizada por la aplicación

3. Arquitectura del Proyecto

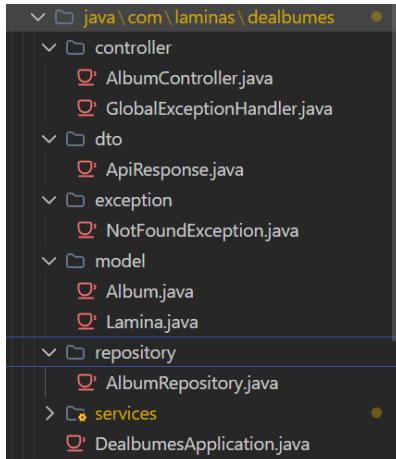
El sistema fue desarrollado siguiendo una arquitectura por capas, separando claramente las responsabilidades de cada componente.

Las capas utilizadas son:

- **Controller:** expone los endpoints REST y gestiona las solicitudes HTTP.
- **Service:** contiene la lógica de negocio del sistema.
- **Repository:** maneja el acceso a datos mediante Spring Data MongoDB.
- **Model:** define las entidades del sistema.
- **Exception y DTO:** permiten un manejo centralizado de errores y respuestas.

Esta arquitectura permite un código ordenado, fácil de mantener y escalable.





4. Modelo de Datos

Entidad Álbum

- id
- nombre
- imagenUrl
- fechaLanzamiento
- tipoLaminas
- lista de láminas

Entidad Lámina

- codigo
- nombre
- imagenUrl
- cantidadRepetidas
- pegada

La relación entre las entidades es de tipo **uno a muchos**, donde un álbum puede contener múltiples láminas embebidas dentro del documento.



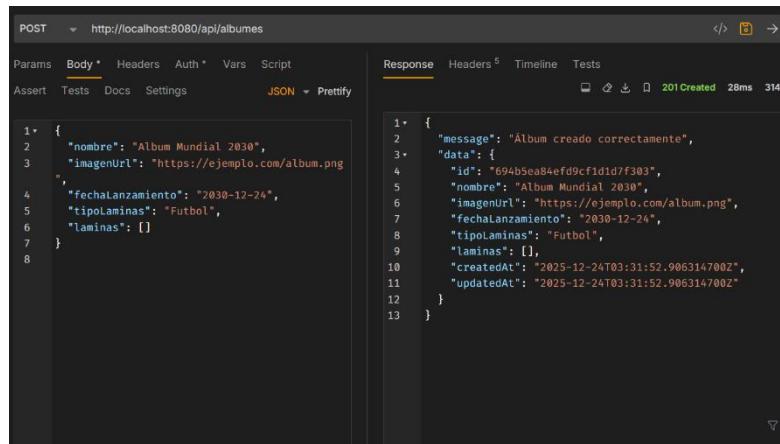


5. Implementación CRUD de Álbumes

Se implementaron operaciones CRUD completas para la entidad Álbum.

Creación de Álbum

Permite registrar un nuevo álbum en el sistema.



```

POST http://localhost:8080/api/albums
{
  "nombre": "Álbum Mundial 2030",
  "ImagenUrl": "https://ejemplo.com/album.png",
  "fechalanzamiento": "2030-12-24",
  "tipolaminas": "Futbol",
  "laminas": []
}
  
```

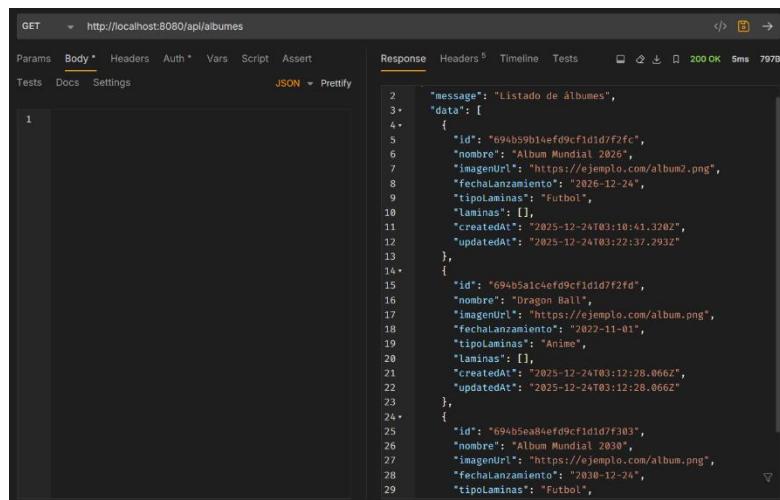
```

Response Headers: 201 Created 28ms 3148
{
  "message": "Álbum creado correctamente",
  "data": {
    "id": "694b5ea84ef9cf1d1d7f303",
    "nombre": "Álbum Mundial 2030",
    "ImagenUrl": "https://ejemplo.com/album.png",
    "fechalanzamiento": "2030-12-24",
    "tipolaminas": "Futbol",
    "laminas": [],
    "createdAt": "2025-12-24T03:31:52.906314700Z",
    "updatedAt": "2025-12-24T03:31:52.906314700Z"
  }
}
  
```

Figura 2. Creación de un álbum mediante endpoint POST

Listado de Álbumes

Permite obtener la lista completa de álbumes registrados.



```

GET http://localhost:8080/api/albums
  
```

```

Response Headers: 200 OK 5ms 7978
{
  "message": "Listado de Álbumes",
  "data": [
    {
      "id": "694b59b14ef9cf1d1d7f2fc",
      "nombre": "Álbum Mundial 2026",
      "ImagenUrl": "https://ejemplo.com/album2.png",
      "fechalanzamiento": "2026-12-24",
      "tipolaminas": "Futbol",
      "laminas": [],
      "createdAt": "2025-12-24T03:10:41.326Z",
      "updatedAt": "2025-12-24T03:22:37.293Z"
    },
    {
      "id": "694b5ac4ef9cf1d1d7f2fd",
      "nombre": "Dragon Ball",
      "ImagenUrl": "https://ejemplo.com/album.png",
      "fechalanzamiento": "2022-11-01",
      "tipolaminas": "Anime",
      "laminas": [],
      "createdAt": "2025-12-24T03:12:28.066Z",
      "updatedAt": "2025-12-24T03:12:28.066Z"
    },
    {
      "id": "694b5ea84ef9cf1d1d7f303",
      "nombre": "Álbum Mundial 2030",
      "ImagenUrl": "https://ejemplo.com/album.png",
      "fechalanzamiento": "2030-12-24",
      "tipolaminas": "Futbol",
      "laminas": []
    }
  ]
}
  
```

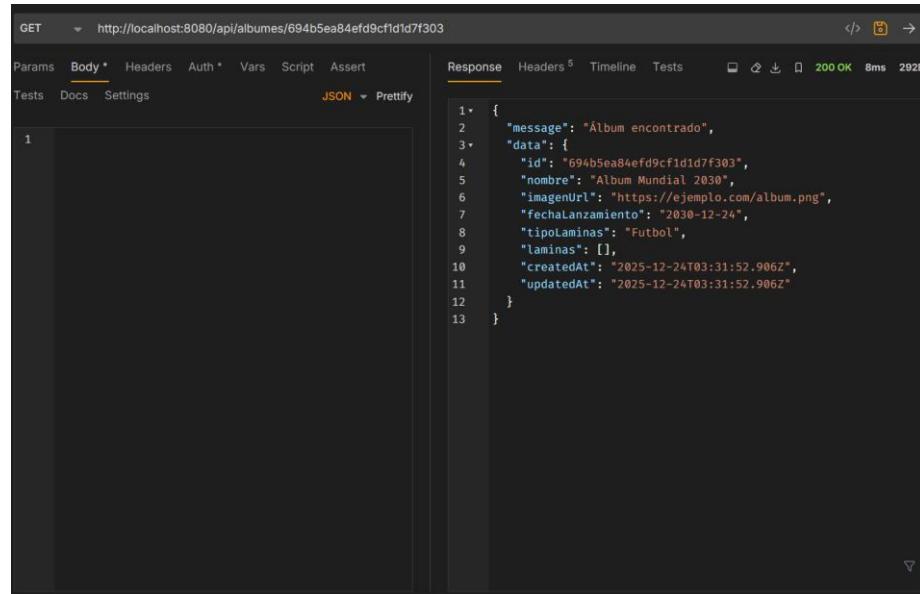
Figura 3. Listado de álbumes registrados





Obtención de Álbum por ID

Permite obtener la información de un álbum específico mediante su identificador.



```

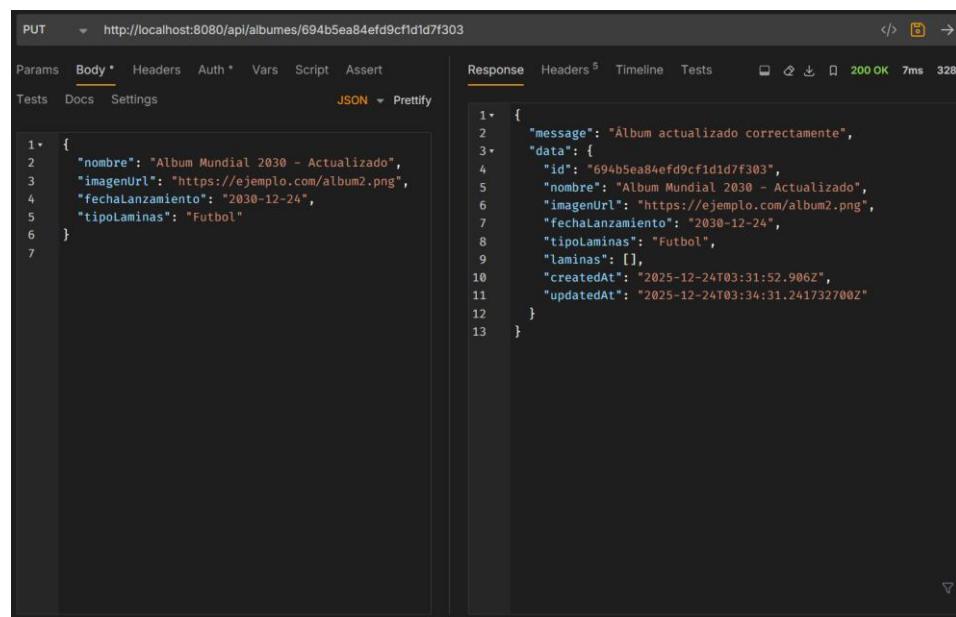
GET http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303
{
  "message": "Álbum encontrado",
  "data": {
    "id": "694b5ea84efd9cf1d1d7f303",
    "nombre": "Album Mundial 2030",
    "imagenUrl": "https://ejemplo.com/album.png",
    "fechaLanzamiento": "2030-12-24",
    "tipolaminas": "Futbol",
    "laminas": [],
    "createdAt": "2025-12-24T03:31:52.906Z",
    "updatedAt": "2025-12-24T03:31:52.906Z"
  }
}

```

Figura 4. Obtención de un álbum por su identificador

Actualización de Álbum

Permite modificar la información de un álbum existente.



```

PUT http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303
{
  "nombre": "Album Mundial 2030 - Actualizado",
  "imagenUrl": "https://ejemplo.com/album2.png",
  "fechaLanzamiento": "2030-12-24",
  "tipolaminas": "Futbol"
}

```

```

{
  "message": "Álbum actualizado correctamente",
  "data": {
    "id": "694b5ea84efd9cf1d1d7f303",
    "nombre": "Album Mundial 2030 - Actualizado",
    "imagenUrl": "https://ejemplo.com/album2.png",
    "fechaLanzamiento": "2030-12-24",
    "tipolaminas": "Futbol",
    "laminas": [],
    "createdAt": "2025-12-24T03:31:52.906Z",
    "updatedAt": "2025-12-24T03:34:31.241732700Z"
  }
}

```

Figura 5. Actualización de información de un álbum





Eliminación de Álbum

Permite eliminar un álbum del sistema.

```
DELETE http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303
```

Response Headers: 5 200 OK 7ms 56B

```
1 {  
2   "message": "Álbum eliminado correctamente",  
3   "data": null  
4 }
```

Figura 6. Eliminación de un álbum mediante endpoint DELETE

6. Gestión de Láminas

El sistema permite gestionar láminas asociadas a un álbum específico.

Agregar Lámina Individual

Permite agregar una lámina a un álbum existente.

```
POST http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303/laminas
```

Response Headers: 5 200 OK 14ms 413B

```
1 {  
2   "message": "Lámina agregada correctamente",  
3   "data": {  
4     "id": "694b5ea84efd9cf1d1d7f303",  
5     "nombre": "Album Mundial 2030 - Actualizado",  
6     "imagenUrl": "https://ejemplo.com/album2.png",  
7     "fechaLanzamiento": "2030-12-24",  
8     "tipoLaminas": "Futbol",  
9     "laminas": [  
10       {  
11         "codigo": "A1",  
12         "nombre": "Jugador 1",  
13         "imagenUrl": "",  
14         "cantidadRepetidas": 2,  
15         "pegada": true  
16       }  
17     ],  
18     "createdAt": "2025-12-24T03:31:52.906Z",  
19     "updatedAt": "2025-12-24T03:35:26.224707500Z"  
20   }  
21 }
```

Figura 7. Agregación de una lámina individual a un álbum





Agregar Lote de Láminas

Permite agregar múltiples láminas de forma simultánea.

The screenshot shows a Postman interface with a POST request to `http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303/laminas/lote`. The request body contains a JSON array of sticker objects:

```
[{"codigo": "A2", "nombre": "Jugador 2", "imagenUrl": "", "cantidadRepetidas": 0, "pegada": false}, {"codigo": "A3", "nombre": "Jugador 3", "imagenUrl": "", "cantidadRepetidas": 1, "pegada": true}, {"codigo": "A4", "nombre": "Jugador 4", "imagenUrl": "", "cantidadRepetidas": 0, "pegada": false}]
```

The response is a JSON object indicating success:

```
{ "message": "Listado de láminas agregado correctamente", "data": { "id": "694b5ea84efd9cf1d1d7f303", "nombre": "Album Mundial 2030 - Actualizado", "imagenUrl": "https://ejemplo.com/album2.png", "fechaLanzamiento": "2030-12-24", "tipoLaminas": "Futbol", "laminas": [ { "codigo": "A1", "nombre": "Jugador 1", "imagenUrl": "", "cantidadRepetidas": 2, "pegada": true}, { "codigo": "A2", "nombre": "Jugador 2", "imagenUrl": "", "cantidadRepetidas": 0, "pegada": false}, { "codigo": "A3", "nombre": "Jugador 3", "imagenUrl": "", "cantidadRepetidas": 1, "pegada": true} ] }}
```

Figura 8. Agregación de múltiples láminas a un álbum

7. Consultas Especiales

Consulta de Láminas Faltantes

Permite obtener las láminas que aún no han sido pegadas en un álbum.

The screenshot shows a GET request to `http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303/laminas/faltantes`. The response is a JSON object indicating the list of missing stickers:

```
{ "message": "Listado de láminas faltantes", "data": [ { "codigo": "A2", "nombre": "Jugador 2", "imagenUrl": "", "cantidadRepetidas": 0, "pegada": false}, { "codigo": "A4", "nombre": "Jugador 4", "imagenUrl": "", "cantidadRepetidas": 0, "pegada": false} ] }
```

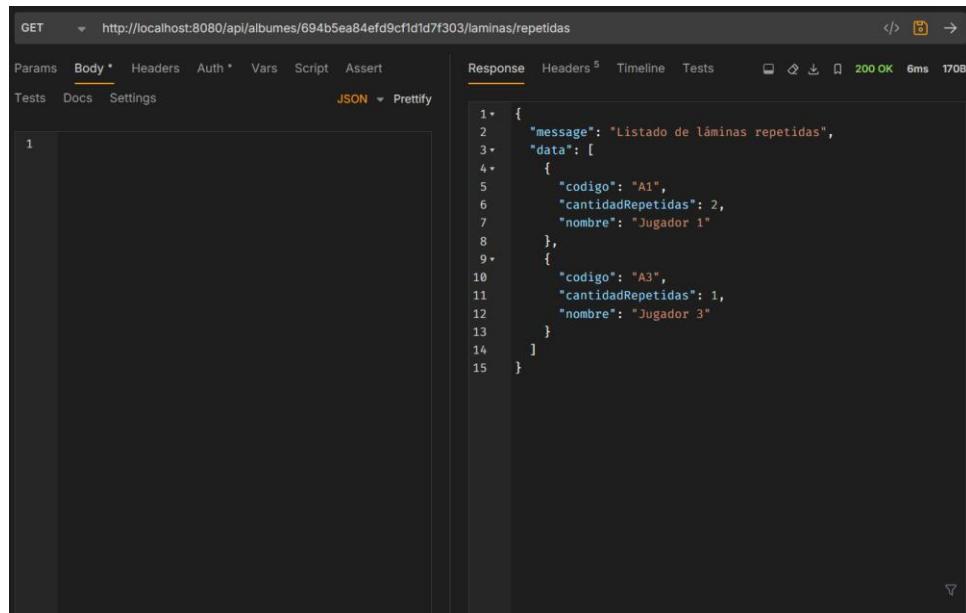
Figura 9. Consulta de láminas faltantes en un álbum





Consulta de Láminas Repetidas

Permite identificar las láminas repetidas dentro de un álbum.



```

GET http://localhost:8080/api/albums/694b5ea84efd9cf1d1d7f303/laminas/repetidas
Params Body * Headers Auth * Vars Script Assert
Tests Docs Settings JSON Prettify
Response Headers 5 Timeline Tests
200 OK 6ms 170B
1
1+ {
2+   "message": "Listado de láminas repetidas",
3+   "data": [
4+     {
5+       "codigo": "A1",
6+       "cantidadRepetidas": 2,
7+       "nombre": "Jugador 1"
8+     },
9+     {
10+      "codigo": "A3",
11+      "cantidadRepetidas": 1,
12+      "nombre": "Jugador 3"
13+    }
14+  ]
15+ }

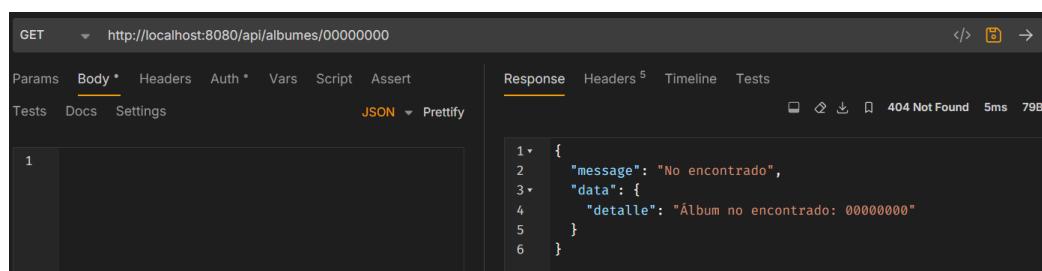
```

Figura 10. Consulta de láminas repetidas en un álbum

8. Validaciones y Manejo de Errores

Para asegurar la integridad de los datos, se utilizaron validaciones mediante Jakarta Validation, evitando el ingreso de información inválida.

Además, se implementó un manejo global de excepciones que permite retornar respuestas claras y estructuradas ante errores, tales como recursos no encontrados o solicitudes incorrectas.



```

GET http://localhost:8080/api/albums/00000000
Params Body * Headers Auth * Vars Script Assert
Tests Docs Settings JSON Prettify
Response Headers 5 Timeline Tests
404 Not Found 5ms 79B
1
1+ {
2+   "message": "No encontrado",
3+   "data": [
4+     {
5+       "detalle": "Álbum no encontrado: 00000000"
6+     }
6+   ]

```

Figura 11. Error al intentar acceder a un álbum inexistente (404)





9. Uso de Inteligencia Artificial

Durante el desarrollo del proyecto se utilizó una herramienta de Inteligencia Artificial como apoyo académico para resolver dudas técnicas, comprender conceptos y organizar el desarrollo del sistema.

La IA fue utilizada únicamente como herramienta de apoyo, mientras que el diseño, implementación y pruebas del sistema fueron realizadas por los integrantes del grupo.

10. Conclusión

Se desarrolló exitosamente una API REST para la gestión de álbumes y láminas utilizando Spring Boot y MongoDB.

El sistema cumple con todos los requerimientos solicitados, implementando operaciones CRUD, validaciones, manejo de errores y consultas especializadas, manteniendo una arquitectura clara y ordenada.

Este proyecto demuestra el correcto uso de tecnologías backend modernas y sienta las bases para futuras mejoras, como la incorporación de mecanismos de seguridad y autenticación.

