

Chalange backend java spring boot api

[https://www.youtube.com/watch?v=6EGfM5sBMlg&ab\\_channel=RodrigoCaro](https://www.youtube.com/watch?v=6EGfM5sBMlg&ab_channel=RodrigoCaro)

objetivo:

desarrollar una API para explorar el mundo de Disney, la cual permitirá conocer y modificar los personajes que lo componen y entender en que películas estos participaron. Por otro lado, deberá exponer la información para que cualquier frontend pueda consumirla.

- Utilizar Spring Boot
- No es necesario armar el Frontend
- Las rutas deberán seguir el patrón REST

Requerimientos técnicos

## 1) Modelado de BD

**Personaje:**

- Imagen
- Nombre
- Edad
- Peso
- Historia
- Película o series asociadas

**Pelicula o serie**

- Imagen
- Titulo
- Fecha de creación
- Calificación (del 1 al 5)
- Personajes asociados

**Genero:**

- Nombre
- Imagen
- Peliculas o series asociadas

## 2. Autenticacion de Usuarios

Para realizar peticiones a los endpoints subsiguientes el usuario deberá contar con un token que obtendrá al autenticarse. Para ello, deberán desarrollarse los endpoints de registro y login, que permitan obtener el token.

Los endpoints encargados de la utenticacion deben ser:

- /auth/login
- /auth/register

### **3. Listado de Personajes**

El listado deberá mostrar:

- Imagen
- Nombre

El endpoint deberá ser:

- /characters

### **4. Creacion, Edicion y Eliminacion de Personajes (CRUD)**

Deberán existir las operaciones básicas de creación, edición y eliminación de personajes.

### **5. Detalle de Personaje**

Deberá mostrares el detalle de las películas en las que haya participado un personaje.

### **6. Busqueda de Personaejs**

Deberán Prmitir buscar por nombre, por edad, peso o peliculas/series en las que participo.

Para especificar el termino de búsqueda o filtros se deberán enviar como parámetros de query:

- GET /characers?name=nombre
- GET /characers?age=edad
- GET /characers?movies=idMovie

### **7. Listado de peliculas**

Debera mostrar solamente los campos imagen, título y fecha de creación

El endpoint deberá ser:

- GET /movies

### **8. Detalle de película / serie con sus personajes**

Devolverá todos los campos de la película o serie junto a los personajes asociados a la misma

### **9. CRUD peliculas / serie**

### **10. Busqueda de Pelicuals o series**

Deberán permitir buscar por título y por género. Además, permitir ordenar los resultados por fecha de creación de forma ascendiente o descendente.

El termino de búsqueda, filtro u ordenación se deberán especificar como parámetros query:

- /movies?name=nombre
- /movies?genre=idGenero

- `/movies?order=ASC | DESC`

### **Documentaicon**

Es deseable documentar los endpoints utilizando alguna herramienta como Postman o Swagger.

### **Tests**

De forma opcional, se podrán agregar tests de los diferentes endpoints de la APP, verificando posibles escenarios de error:

- Campos faltantes o con un formato invalido en BODY de las peticiones
- Acceso a recursos inexistentes en endpoints de detalle

Los tests pueden realizarse utilizando JUnit y Mockito.