

Ejercicio: Diseño de API REST y Arquitectura para una App de Películas

En este ejercicio, deberás diseñar una **API REST** para una aplicación móvil que permite a los usuarios interactuar con una base de datos de películas y realizar diversas acciones sociales, como seguir a otros usuarios y calificar películas. El objetivo es que apliques buenas prácticas de desarrollo de software, tanto en el diseño de la API como en la definición del modelo de dominio y la arquitectura preliminar de la aplicación.

Contexto de la App:

1. Los usuarios pueden descargar la app y crear una cuenta.
2. **Sin estar logueado**, los usuarios pueden buscar películas recomendadas, con filtros por título, actores y categorías.
3. **Estando logueado**, cada usuario puede:
 - Calificar películas con una puntuación del 1 al 10.
 - Buscar otros usuarios, seguirlos y ver las películas que han calificado.
 - Aceptar o rechazar solicitudes de seguimiento.
 - Ver su lista de seguidores y a quién está siguiendo.
 - Ver y editar su perfil, incluyendo la opción de eliminar su cuenta.

Atributos del Usuario:

Un usuario debe tener al menos los siguientes atributos:

- Email
- Nombre
- Apellido
- Foto de perfil (avatar)
- Fecha de nacimiento
- Género

Funciones de un Usuario ADMIN:

Además de las funcionalidades estándar, un usuario con rol **ADMIN** puede:

- Iniciar sesión como administrador.
- Crear, editar y eliminar otros usuarios de rol **ADMIN**.
- Crear, editar y eliminar categorías de películas.
- Crear, editar y eliminar actores.
- Crear, editar y eliminar películas, asignando actores y categorías (sin modificar las calificaciones de los usuarios).

Requisitos Técnicos:

- Autenticación con **Basic Authentication** para el inicio de sesión y **JWT** para las acciones autenticada con manejo de refresh token
- Paginación en el listado de películas.
- Buenas prácticas en el uso de códigos HTTP, verbos y nombres de los endpoints.

Tareas a Realizar:

1. **Definir una API REST:**
 - Diseña los endpoints necesarios para soportar todas las funcionalidades descritas, incluyendo los roles de usuario estándar y ADMIN.
 - Sigue el ejemplo dado a continuación para el login, utilizando los códigos de estado HTTP y verbos apropiados.
2. **Modelo de Dominio Preliminar:**
 - Elabora un modelo de dominio que represente los elementos principales de la aplicación definiendo sus relaciones.
3. **Propuesta de Arquitectura Preliminar:**
 - Define una arquitectura básica para la aplicación.

Ejemplo de Login:

Descripción: Permite que los usuarios inicien sesión.

Request:

- Headers: `Authorization: Basic {base64(email:password)}`
- **POST** `/api/v1/users/token`

Response:

Código HTTP: `200 OK`

```
{  
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "refresh_token": "m9obiBEb2UiLCJpYXQiOiE1MTYyMzkwMj..."  
}
```

Código HTTP: `401 Unauthorized` si las credenciales son incorrectas.