FilmApp API Practica

Entidades

User:

- **Id**: Identificador único del usuario
- **name**: Nombre del usuario y tipo String
- **email**: Correo y tipo String, único y con validaciones
- **password**: Contraseña de usuario y encriptada
- **role**: Por defecto será “user” y lo cambiaremos a mano en base de datos si queremos hacer un usuario “admin”
- **favorites**: Array de MovieId de peliculas que le gusta al usuario, tipo ObjectID y hace referencia al modelo Movies.

Movie:

- **Id**: Identificador único de la peli
- **Title**: Nombre de la peli, tipo String
- **Description** : Descripción de la peli , tipo String
- **Category**: Array de Strings para guardar los géneros de la peli, tipo enum.
- **Director:** Nombre del director y tipo String
- **Rating**: Valoración de la peli, tipo String
- **PosterUrl**: Portada de la peli, tipo String

- **TrailerUrl**: Trailer de la peli, tipo String
- **Year**: Año de la peli, tipo String
- **CreateAt**: Fecha de creación

EndPoints

Sin autenticación:

- `GET /movies` (Para obtener todas las películas)
- `GET /movies/:id` (Para obtener el detalle de una película)
- `GET /recent_movies` (Para obtener las películas recientes, obtener las 10 peliculas insertadas recientemente)
- `GET /most_popular` (Para obtener las películas mejor valoradas, obtener las 10 películas mejora valoradas)
- `POST /login` (EndPoint para loguearse en la app)
- `POST /signup` (EndPoint para registrase en la app)

Con autenticación:

- `GET /user/:id/favorite` , para obtener la lista de películas favoritas del usuario logueado
- `POST /user/:id/favorite` , para añadir películas a favoritos
- `DELETE /user/:id/favorite` , para borrar películas de favoritos
- `POST /movies` , para añadir películas solo los administradores
- `DELETE /movies/:id` , para borrar películas solo los administradores
- `PATCH /movies/:id` , para actualizar películas solo los administradores
- `GET /genereToken` Para recuperar el token normal y token de refresco y es el único donde utilizamos el REFRESH_TOKEN

Documentación

- Swagger para tener todos los endPoints comentados y con ejemplos.
- Collection de Thunder Client

Seguridad

- Para el token, JWT, para generar TOKEN normal y REFRESH_TOKEN para refrescar, uno de 15 minutos y otro de 60min.
- Encriptado de contraseña del usuario con Bcrypt
- Middleware para controlar que los token no han expirado
- Middleware para controlar que el usuario es admin o no.

Extras

- Mandar un email de agradecimiento cuando se registre un usuario nuevo.
- EndPoint `GET /movies` sistema de paginación, donde le pasemos la página y limite, y si no se pasa que tenga por defecto.
- Sistema de comentarios en las películas, donde se guarde el usuario, la película, el comentario y la fecha del comentario.
- EndPoint para editar usuario
- EndPoint para obtener la media de valoraciones de las películas.
- Botón para descargar un PDF con el detalle de la película.