



Movie & Series Tracker

Una aplicación web **CRUD** donde el alumnado debe construir:

Frontend: Angular

- Componentes.
- Routing.
- Reactive Forms.
- `@for`, `@if`
- Eventos.
- Servicio de consumo de API.

Backend: Node + Express (API REST)

Base de datos: MongoDB

Sin autenticación

1. Requisitos del Proyecto

Convenciones OBLIGATORIAS

Todos los proyectos deben respetar exactamente estas configuraciones. No cumplirlas implica suspender.

Habr  dos carpetas, una llamada API y otra llamada WEB, en la carpeta API estar  vuestra API y en la carpeta WEB estar  el proyecto Angular.

Puertos obligatorios

| Servicio | Puerto |
|--------------------|--------|
| Angular (Frontend) | 4200 |
| API Node + Express | 3000 |
| MongoDB | 27017 |

Base de Datos (MongoDB)

Nombre de la base de datos: `movie_tracker_db`

Nombre de la colección (OBLIGATORIO): `movies`.

La conexión **debe apuntar exactamente a:**

`mongodb://mongoadmin:secret@localhost:27017/movie_tracker_db`

No se aceptan:

- Nombres de base de datos distintos
- Variables de entorno con otros nombres
- usuario mongoadmin, pass secret.

Frontend (Angular)

El alumnado debe implementar:

1. Componentes Standalone

Componentes mínimos requeridos:

- `MovieListComponent` — listado de películas/series
- `MovieDetailComponent` — vista de detalle
- `MovieFormComponent` — crear/editar
- `NavbarComponent`
- Componente raíz (`AppComponent`)

2. Routing

Mínimo **3 rutas**:

- `/movies`
- `/movies/new`
- `/movies/:id`

3. Formularios Reactivos (Reactive Forms)

El formulario de creación/edición debe usar:

- `FormGroup`

4. Funcionalidades de Angular

Debe incluir obligatoriamente:

- Bloques de control `@for` y `@if`
- Manejo de eventos (`click`), (`submit`), etc.)
- `@Input` y/o `@Output` cuando sea aplicable
- Un **servicio** para consumir la API de Node

Funcionalidades de la aplicación:

- Búsqueda de películas por **título o género**
- Filtro por **plataforma o género**
- **Paginación**
- Sistema de **valoraciones con estrellas**
- Marcar películas como **"vistas" (watched)**

5. CRUD

El frontend debe soportar:

- **Create** → crear película/serie
- **Read** → listar y ver detalle
- **Update** → editar
- **Delete** → eliminar

Campos recomendados:

- `title`
- `year`
- `genre`
- `rating` (0-10)
- `platform` (Netflix, HBO Max, Prime Video...)
- `imageUrl`

2. Backend (Node + Express)

Requisitos básicos

1. Endpoints REST

Debe implementar los siguientes endpoints:

GET /api/movies

GET /api/movies/:id

POST /api/movies

PUT /api/movies/:id

DELETE /api/movies/:id

2. Conexión con MongoDB

Debe usarse **Mongoose**.

Esquema con validaciones:

- **title**: String, **required**
- **year**: Number, **required**
- **genre**: String
- **rating**: Number (0–10)
- **platform**: String
- **imageUrl**: String

4. Gestión de errores

Debe incluir:

- **404** → recurso no encontrado
- **500** → error general del servidor
- Errores de validación de datos



3. Entregables

Archivo ZIP con la carpeta API y WEB. En ambas carpetas eliminad la carpeta Node_Modules. Además borrad la carpeta .Angular en Angular