



Proyecto: Videoclub con Node.js y React



Descripción del Proyecto

Este proyecto consiste en desarrollar una aplicación tipo **Videoclub** similar a **TheMovieDB**, donde los usuarios puedan explorar información de películas, agregarlas a favoritos, dejar comentarios y visualizar detalles. El backend en **Node.js** gestionará la autenticación mediante **JWT con HTTPOnly**, almacenará la información obtenida desde **TheMovieDB API** y proveerá una API REST para el frontend en **React + TailwindCSS 4.0**.



Funcionalidades Principales

- 🎬 **Obtención y almacenamiento de películas desde TheMovieDB API** en una base de datos.
- 🗝️ **Autenticación con JWT (HTTPOnly)** permitiendo registro, login y logout.
- ⭐ **Sistema de favoritos** para agregar y gestionar películas preferidas.
- 💬 **Sistema de comentarios** para cada película.
- 🌐 **Backend con API REST** que gestione las funcionalidades mencionadas.
- 🛡️ **Protección de rutas en el frontend con React Router v7**, asegurando que solo usuarios autenticados puedan ver detalles de películas, gestionar favoritos y comentar.
- 🏠 **Página de inicio accesible para todos**, mostrando las películas más populares.
- 🎨 **Frontend en React + TailwindCSS 4.0 + React Router v7**.
- 🐳 **Dockerización completa** del backend y frontend.
- 📖 **README.md** detallado con instrucciones de instalación y despliegue.



Tecnologías a Utilizar

- **Node.js** (v20+)
- **Express.js** para la gestión del servidor
- **MongoDB** o alternativa para almacenamiento
- **Fetching** para consumir TheMovieDB API
- **JSON Web Tokens (JWT)** para autenticación segura
- **React + TailwindCSS 4.0** para el frontend
- **React Router v7** para la navegación
- **Docker** para la contenedorización



Requisitos del Proyecto

1. **Backend en Node.js** que obtenga datos de **TheMovieDB API**, los almacene en la base de datos y exponga endpoints para el frontend.






2. Endpoints requeridos:




- **POST /api/auth/register** → Registro de usuario.
- **POST /api/auth/login** → Inicio de sesión con JWT (almacenado en HTTPOnly cookie).
- **POST /api/auth/logout** → Cierre de sesión.
- **POST /api/auth/check-auth** → Verificar si estoy logueado actualmente.
- **POST /api/users/me** → Información del usuario que está logueado actualmente.
- **GET /api/movies** → Devuelve todas las películas populares.
- **GET /api/movie/:id** → Devuelve la información detallada de una película.
- **POST /api/endpoint_a_tu_elección** → Agregar una película a favoritos.
- **DELETE /api/endpoint_a_tu_elección** → Eliminar una película de favoritos.
- **POST /api/endpoint_a_tu_elección** → Agregar un comentario a una película.

3. Frontend en React que permita:

- **Explorar películas populares.**
- **Autenticarse (registrarse, iniciar sesión y cerrar sesión).**
- **Agregar y eliminar películas de favoritos.**
- **Dejar comentarios en películas.**
- **Proteger rutas con React Router v7, impidiendo acceso a funciones avanzadas sin estar autenticado.**

4. Dockerización completa del backend y frontend.**5. README.md** detallado con instrucciones de instalación y uso.**Rúbrica de Evaluación**

Criterio	Descripción	Puntuación
 Obtención de datos y almacenamiento	Consume datos de TheMovieDB API y los almacena correctamente	1 pts
 Interfaz gráfica Responsive y prof	Generar una interfaz Web Responsive y moderna	1,5 pts
 Interfaz con protecciones	Protección de rutas en función de si estamos registrados o no	1 pts
 Autenticación con JWT	Implementa login, registro y logout con HTTPOnly cookies	2 pts
 Sistema de favoritos	Permite agregar y eliminar películas de favoritos	1,5 pts

Criterio	Descripción	Puntuación
 Comentarios en películas	Implementa la funcionalidad para comentar en películas	1,5 pts
 Dockerización	Backend y frontend correctamente dockerizados	1 pts
 Documentación	Instrucciones detalladas en README.md para instalación y uso	0,5 pts
TOTAL	10 pts	

Ejemplo de Interfaz

(Se recomienda incluir una capturas de pantalla de la interfaz final para referencia en el README)