CINETICA

Catálogo de Películas – Integración de APIs con Arquitectura MVC

Proyecto académico desarrollado para el Parcial – Corte III de la asignatura
Base de datos II

Santiago Contreras Carvajal

Juan Sebastian Ayala Fernandez

Keyner David Vera Diaz
Universidad de Pamplona

Descripción general del proyecto

CINETICA es una aplicación web tipo catálogo de películas, desarrollada bajo una arquitectura MVC, que integra dos APIs públicas para el consumo, persistencia y visualización de información cinematográfica.

El sistema permite:

- Consultar un catálogo local de películas.
- Buscar películas externas mediante una API pública.
- Persistir información relevante en una base de datos no relacional (MongoDB).
- Gestionar reseñas y calificaciones mediante operaciones CRUD.
- Visualizar información enriquecida como sinopsis, géneros, plataformas de streaming y calificaciones.

Este proyecto cumple con los requisitos técnicos y académicos establecidos para el parcial del tercer corte .

Arquitectura utilizada (MVC)

La aplicación sigue estrictamente el patrón **Modelo – Vista – Controlador (MVC)**:

Modelo (Model)

- Gestión de datos.
- Conexión a bases de datos relacionales y no relacionales.
- Consumo de APIs externas.
- Persistencia de información obtenida de las APIs.

Vista (View)

- Interfaz web desarrollada con React.
- Visualización de datos provenientes de MongoDB y APIs externas.
- Componentes reutilizables (catálogo, modal de película, formulario de reseñas).

Controlador (Controller)

- Orquestación entre vistas y modelos.
- Manejo de lógica de negocio.
- Validación de datos.
- Manejo de errores y respuestas HTTP.

APIs Integradas

API 1 – The Movie Database (TMDB)

Propósito:
Obtener información estructurada sobre películas.

Datos consumidos:

- Título
- Sinopsis
- Géneros
- Calificación
- Imágenes (poster y backdrop)
- Plataformas de streaming
- Año de estreno

Uso en el proyecto:

- Búsqueda global de películas.
- Enriquecimiento del catálogo local.
- Persistencia parcial de datos en MongoDB.

API 2 – TMDB Watch Providers

Propósito:
Complementar la funcionalidad mostrando dónde ver una película.

Datos consumidos:

- Plataformas de streaming (Netflix, HBO, Amazon Prime, etc.).
- Disponibilidad por región (Colombia).

Uso en el proyecto:

- Visualización dinámica dentro del modal de detalle.
- No se persiste toda la información, solo se integra a la lógica de negocio..

Bases de datos utilizadas

Base de datos relacional (SQL)

- Usuarios
- Roles
- Autenticación
- Control de acceso

Base de datos no relacional (MongoDB)

- Catálogo de películas
- Información enriquecida desde TMDB
- Reseñas y calificaciones

Operaciones CRUD implementadas

Entidad: Películas

- Crear películas (seed + enriquecimiento desde TMDB).
- Consultar catálogo.
- Actualizar información enriquecida.
- Eliminar registros (si se requiere).

Entidad: Reseñas

- Crear reseñas autenticadas.
- Consultar reseñas por película.
- Actualizar reseñas.
- Eliminar reseñas.

Manejo de errores y respuestas HTTP

La aplicación maneja correctamente:

- 200 OK – operaciones exitosas.
- 400 Bad Request – errores de validación.
- 401 Unauthorized – falta de autenticación.
- 404 Not Found – recursos inexistentes.
- 500 Internal Server Error – errores del servidor.

Los mensajes de error se muestran tanto en backend como en frontend para una mejor experiencia de usuario

Cómo ejecutar el proyecto

Backend (Django + MongoDB)

```
# Crear entorno virtual
python -m venv venv
source venv/bin/activate  # Windows: venv\Scripts\activate

# Instalar dependencias
pip install -r requirements.txt

# Ejecutar servidor
python manage.py runserver
```

Frontend (React)

```
npm install
npm run dev
```

Variables de entorno

Ejemplo de archivo .env.example:

```
TMDB_API_KEY=”aqui va el api key”
MONGO_URI=mongodb://localhost:27017/cinetica
```

Funcionalidades principales

- Búsqueda de películas (MongoDB / TMDB).
- Catálogo visual estilo streaming.
- Detalle completo de película (modal).
- Sistema de calificaciones.
- Reseñas autenticadas.
- Filtros por género y tipo.
- Autenticación con JWT.

Estructura del proyecto

```
cinetica/
├── backend/
│   ├── models/
│   ├── views/
│   ├── controllers/
│   └── api/
├── frontend/
│   ├── components/
│   ├── pages/
│   ├── api/
│   └── styles/
└── README.md
```

Video demostrativo

El video (máx. 5 minutos) muestra:

- Consumo de APIs.
- Operaciones CRUD.
- Arquitectura MVC.
- Funcionamiento general del sistema.

Conclusión

El proyecto CINETICA cumple con todos los requisitos establecidos en el parcial del tercer corte:

- Integración de dos APIs.
- Uso de arquitectura MVC.
- Persistencia en base de datos no relacional.
- Implementación de CRUD.
- Documentación clara y completa.

Este sistema demuestra la correcta aplicación de conceptos de integración de servicios, arquitectura de software y desarrollo web moderno.