

Documentación Técnica - Sistema de Recomendación de Películas

1. Resumen del Proyecto

Este proyecto presenta un sistema de recomendación de películas diseñado para ofrecer sugerencias personalizadas a los usuarios según sus gustos e historial. Fue desarrollado en Java con JavaFX para la interfaz gráfica y Neo4j como base de datos de grafos. Los usuarios pueden registrarse, indicar sus preferencias de género, actores favoritos y plataformas de streaming, así como seleccionar películas que ya han visto. El sistema responde generando recomendaciones variadas y precisas, además de permitir gestionar listas de deseos. Esta solución busca simular cómo plataformas reales como Netflix o Disney+ podrían personalizar su contenido para mejorar la experiencia del usuario.

2. Estructura del Sistema

El sistema está dividido en dos paquetes principales:

- `org.example.Model`: contiene las clases encargadas de manejar la lógica del sistema, la conexión con la base de datos Neo4j, la carga de datos desde archivos CSV, el algoritmo de recomendación, y los modelos de datos como películas, usuarios, géneros, etc.
- `org.example.View`: contiene todas las vistas creadas con JavaFX. Incluye pantallas para el registro, inicio de sesión, panel principal, recomendaciones, lista de deseos y selección de preferencias.

También se incluye un archivo `estilos.css` que mantiene una apariencia visual consistente en toda la interfaz.

3. Algoritmo de Recomendación

El sistema implementa un algoritmo híbrido que combina recomendaciones colaborativas con sugerencias basadas en contenido, ponderadas por las preferencias individuales del usuario. A continuación, se describen los pasos principales:

1. **Identificación de películas vistas:**
Se consultan todas las películas que el usuario ha visto, para evitar recomendar contenido repetido.
2. **Cálculo de pesos personalizados:**
Se analiza cuántas preferencias ha registrado el usuario en tres categorías:

- Géneros (PREFIERE_GÉNERO)
 - Actores (PREFIERE_ACTOR)
 - Plataformas (PREFIERE_PLATAFORMA)
- Con base en eso, se calculan pesos relativos (porcentaje de cada tipo sobre el total).
3. **Recomendación colaborativa:**
Se buscan usuarios que hayan visto películas en común con el usuario actual. Se calcula la similitud entre usuarios usando una fórmula basada en la raíz del producto del número de películas vistas (cosine-like), y se sugieren películas que esos usuarios han visto y el actual no.
 4. **Recomendación basada en contenido:**
 - Se consultan películas que coinciden con los géneros preferidos y no han sido vistas.
 - Se repite el mismo proceso con actores y plataformas.
 - A cada coincidencia se le suma un puntaje proporcional al peso de ese tipo de preferencia.
 5. **Películas populares:**
Se obtiene un top de las 10 películas mejor calificadas (por rating) que el usuario no ha visto, para asegurar que al menos parte del contenido recomendado tenga alta aceptación general.
 6. **Selección final:**
Se agrupan todas las películas candidatas no vistas, sumando los resultados anteriores. Se mezclan de forma aleatoria y se devuelven solo cinco recomendaciones finales para mostrar variedad.

4. Base de Datos Neo4j

La base de datos está construida sobre una estructura de grafos en Neo4j. Los nodos principales son:

- Usuario: representa a cada persona registrada. Relacionado con:
 - (:Usuario)-[:PREFIERE]->(:Genero)
 - (:Usuario)-[:PREFIERE]->(:Actor)
 - (:Usuario)-[:PREFIERE]->(:Plataforma)
 - (:Usuario)-[:VIO]->(:Pelicula)
 - (:Usuario)-[:DESEA_VER]->(:Pelicula)
 - (:Usuario)-[:CALIFICO {valor}]->(:Pelicula)

- Pelicula: nodo central con propiedades como nombre, duración, rating y suma de puntuaciones.

Se relaciona con:

- (:Pelicula)-[:PERTENECE_A]->(:Genero)
- (:Pelicula)-[:ACTUA_EN]->(:Actor)
- (:Pelicula)-[:DISPONIBLE_EN]->(:Plataforma)

- Genero, Actor, Plataforma: nodos auxiliares conectados tanto a películas como a usuarios.

Los datos iniciales fueron cargados desde archivos CSV por medio de la clase DataLoader.

5. Instrucciones de Uso e Instalación

Requisitos:

- Java JDK 17 o superior
- Apache Maven
- Neo4j Desktop o servidor Neo4j local
- Librerías utilizadas (agregadas por Maven):
 - Neo4j Java Driver
 - JavaFX SDK
- IDE recomendado: IntelliJ IDEA

Pasos para ejecutar el sistema:

1. Clonar el repositorio: `git clone https://github.com/MiguelRosas11/Proyecto-2-ED`
2. Abrir el proyecto en IntelliJ.
3. Configurar Neo4j: ejecutar el servidor local en `bolt://localhost:7687` con usuario y contraseña predefinidos.
4. Ejecutar la clase DataLoader para cargar los datos iniciales desde CSV con el comando: `mvn compile exec:java`
5. Ejecutar el sistema con el comando: `mvn clean javafx:run`
6. Utilizar la aplicación para registrarse, ingresar gustos, ver recomendaciones, marcar películas vistas, gestionar la lista de deseos y puntuar películas.

6. Documentación con usuarios

Link con el video	Puntos de mejora	Nivel de satisfacción (1-10)
https://youtu.be/DzTHuyDyz24	Algunas cajas de mensajes de confirmación de botones en algunos puntos porque el usuario necesita saber qué hace ese botón	8