Recuperación de Información y

Recomendaciones en la Web

¿Qué ver hoy?

Recomendación de Películas y Series

Estudiantes

Gabriel Kryger	4.933.558-9
Ezequiel Labarrere	4.965.278-3
Agustín Soria	4.759.138-9
Tali Yovine	5.668.252-7

Docente

Libertad Tansini

Contenido

Introducción

Actualmente el mercado audiovisual maneja cientos de miles de contenidos para ofrecer. El objetivo de nuestro trabajo será enfocarnos en este contexto y poder facilitar de alguna manera esa difícil decisión a los consumidores cuando deciden que mirar, para ello aplicaremos algunos de los temas vistos en el curso utilizando las herramientas necesarias para generar una solución a la problemática. Sumado a esto detallaremos la arquitectura, tecnologías y fuentes de información utilizados.

Problema

Elegir qué mirar es una problemática común, en épocas actuales el contenido audiovisual va en aumento a un ritmo acelerado, provocando que a las personas se le dificulte cada vez más el encontrar series o películas de su agrado. Normalmente esta tarea puede consumir horas de la semana, que se podrían estar disfrutando con la familia, amigos, entre otros.

Por otro lado, la variedad de plataformas y multitud de contenido que se posee actualmente hace difícil mantenerse al tanto de nuevos contenidos que puedan resultar atractivos, sin contar los títulos viejos que se agregan día a día.

Adicionalmente, buscar recomendaciones, ya sea por género, actores, directores o un título en particular, se debe hacer sobre un grupo reducido de contenido sobre la plataforma en la que estoy. Esto implica tener que pasar de una a otra sin obtener un buen resultado en mi búsqueda ya que esta termina muy limitada por la falta de cruzamiento de datos y el no acceso a alguna de ellas.

Enfoque de la solución

El equipo plantea crear una plataforma de búsqueda de contenido audiovisual centralizada, en donde se podrá realizar búsquedas personalizadas tomando en cuenta la preferencia de los usuarios y generando una respuesta que contemple las plataformas en las que se encuentra suscripto, la región donde se encuentra entre otras.

Para lograr esto el equipo propuso un sistema de recomendación utilizando filtrado colaborativo basado en memoria, para el cálculo de la predicción. A su vez, para computar la medida de similitud entre dos usuarios, se utilizará, el método del coeficiente de correlación de Pearson. Además de esto, se le agregara un sistema de “peso” al orden de las recomendaciones, calculado por el promedio de las calificaciones por los usuarios más parecidos, y el promedio más alto determinara el orden a mostrar. Por último se hace un filtrado para mostrar al usuario solo las películas y/o series que se encuentren en las plataformas la cual está suscrito entre otras opciones de filtrado que se detallan en la sección *Funcionalidades y uso*. El algoritmo en general es O(m*n) siendo m la cantidad de películas y n la cantidad de usuarios.

La idea principal de estos sistemas es automatizar el proceso “boca a boca”, donde las personas recomiendan que ver con sus allegados. Cuando normalmente se tiene que elegir entre una variedad de opciones se confía en las opiniones positivas recibidas. Estos sistemas actúan como guías que orientan en la toma de decisiones relacionadas con los gustos personales y han tenido gran aceptación debido a que las personas están acostumbradas a recibir y solicitar recomendaciones.

El filtrado colaborativo consiste entonces en recomendar elementos que les han gustado a usuarios con preferencias similares, basándose únicamente en el puntaje que estos asignan a los ítems del sistema. Para ello las personas puntúan los elementos de acuerdo a sus intereses y a partir de ellos se calculan las recomendaciones. Específicamente el algoritmo basado en memoria que utilizamos refiere a que, el cálculo de la predicción se realizará considerando la similitud entre el usuario activó (el que hace la solicitud) y los demás usuarios del sistema.

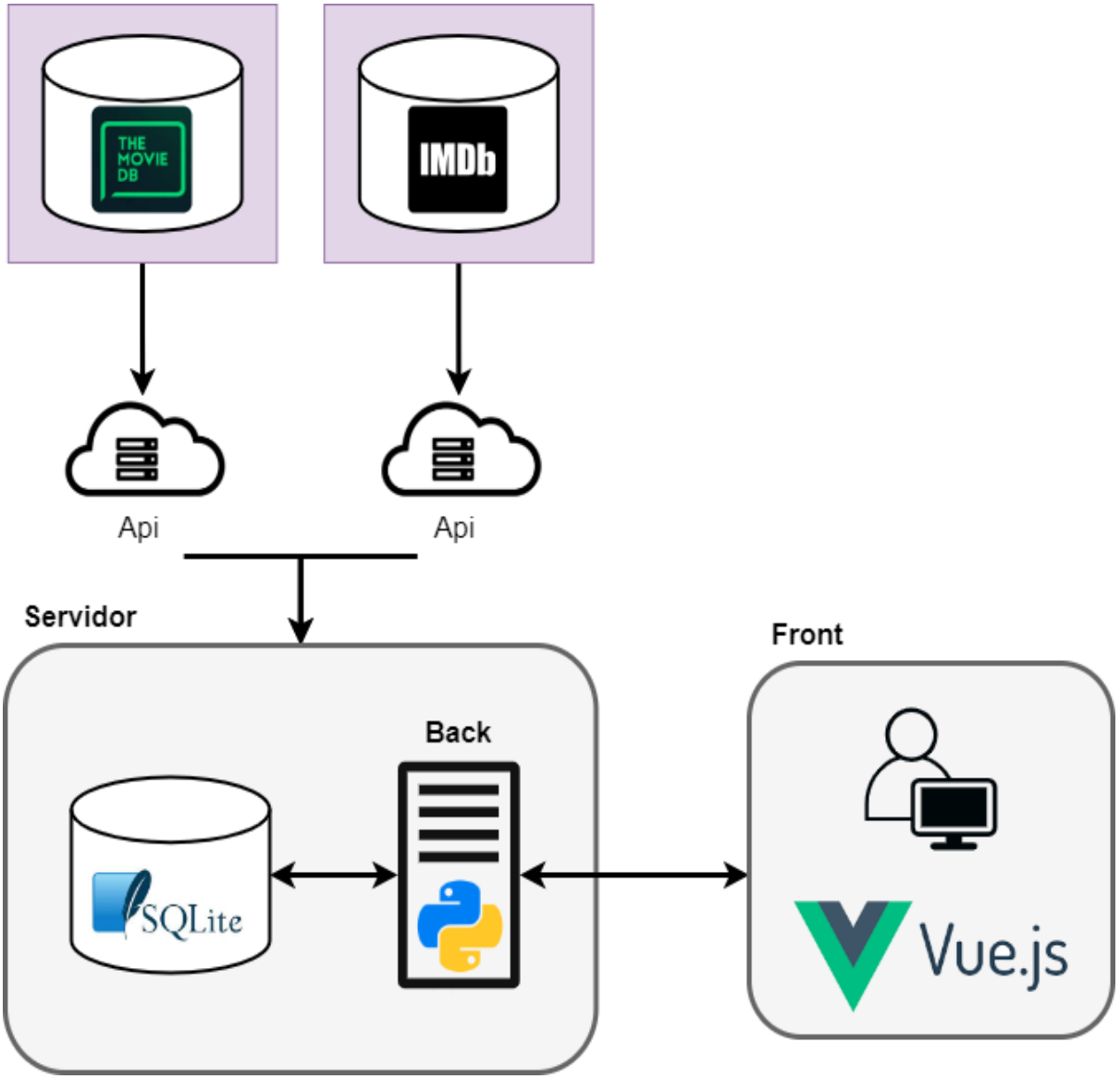
Entonces el procedimiento general aplicado en el sistema se resume de la siguiente forma:

1. Los usuarios expresan sus valoraciones sobre las series y películas mediante una calificación de 1 a 5, siendo, siendo 1 una mala calificación y 5 excelente.
2. A partir de esta información, se intenta predecir el puntaje que daría el usuario que solicita la recomendación, al contenido multimedia no conocido hasta el momento por el.
3. De los resultados calculados se seleccionan los elementos con valores más altos para realizar la recomendación. A estos se los ordena por el promedio total de valoraciones más alto.
4. Por último filtra según las plataformas a las que el usuario se encuentra suscrito, entre otras opciones presentadas.

Arquitectura

Para la implementación de la solución, se optó por una arquitectura **cliente servidor.** Donde la interfaz gráfica se comunica con el servidor y este con las bases de películas a través de Apis. A continuación se listan los componentes por separado:

- Interfaz gráfica web del usuario, implementado en Vue.js
- Servidor que contiene un Backend implementado en Python.
- Base de datos implementada con SQLite para manejar las estructuras utilizadas en la lógica del back.
- Acceso a las Apis de las páginas de películas *The movie db* y *IMDb* para la obtención del catálogo de películas.



Componentes

Backend

Para la implementación de Backend se utilizó el lenguaje python. En este es donde se implementa el filtrado colaborativo además de las opciones de filtrado mencionadas anteriormente. Por otro lado este se conecta tanto con nuestra base de datos como con las Apis donde se recupera posteriormente toda la información de las películas y/o series.

Frontend

Para la implementación del Frontend se utilizó Vue.js, esto nos permite una fácil integración por métodos POST utilizando arquitectura REST, ya que utiliza el lenguaje JavaScript. Todo esto complementado por una documentación extensa y de fácil acceso, por ende nos permitió utilizar una plantilla de código abierto y poderla adaptar para nuestra realidad de forma fácil.

Base de datos

En la BD se tienen los datos indispensables para realizar el filtrado colaborativo y los filtros del usuario, esto se decidió por dos motivos. El primero es la masividad de los datos, los cuales no son posibles de manejar de manera local ya que solamente las calificaciones de los contenidos multimedia pesan aproximadamente 2 Megas. El segundo es que no es necesario tener todos los datos de las series y/o películas porque estos los obtenemos una vez hecho el procesamiento y antes de desplegar el resultado al usuario.

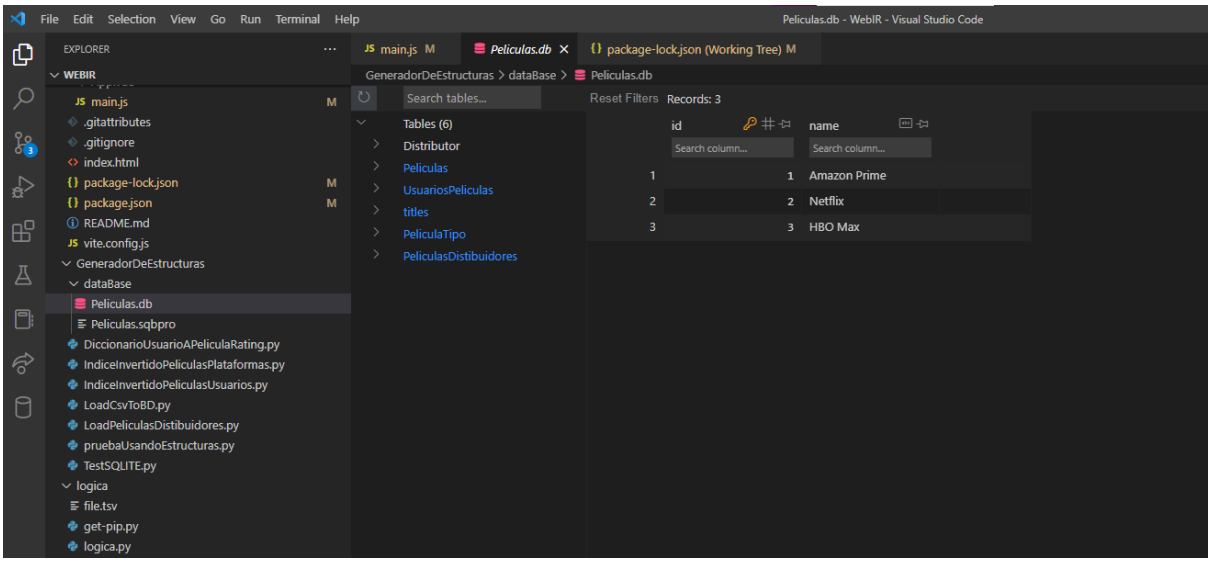
Se utilizaron las siguientes tablas:

- Distribuidor(id, name): se utiliza para la administración de plataformas, las cuales son Netflix, Amazon Prime y HBO max
- Películas (id, título): índice para almacenar un conjunto de nombres de películas/series a partir de sus ids.
- UsuariosContenido (idUsuario, idPelicula, calificación): diccionario donde para cada usuario, contiene la calificación que le dió a cada película que vió.
- PeliculasDistribuidores (títulos, distribuidores):índice invertido el cual permite obtener a partir de los títulos de película, los distribuidores de la misma.
- PeliculasTipo (título, tipo): índice que permite obtener el tipo de contenido (películas/series) que posee cada título del sistema.

Herramientas

SQLite

SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño , rápido , autónomo , de alta confiabilidad y con todas las funciones. El mismo se utiliza para el manejo de índices y diccionarios para la lógica del back. Para su manejo se utilizó SQLite viewer en cual se encuentra dentro de los complementos de visual studio.



Pandas

Es una herramienta de manipulación y análisis de datos de código abierto construida sobre el lenguaje de programación Python. Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales. En nuestro caso se utilizó para la creación de estructuras temporales, análisis de datos y ejecución de algoritmos de recomendación.

Vue.js

Vue.js es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Utilizado por nuestra App para la interfaz que verá e interactúa el usuario.

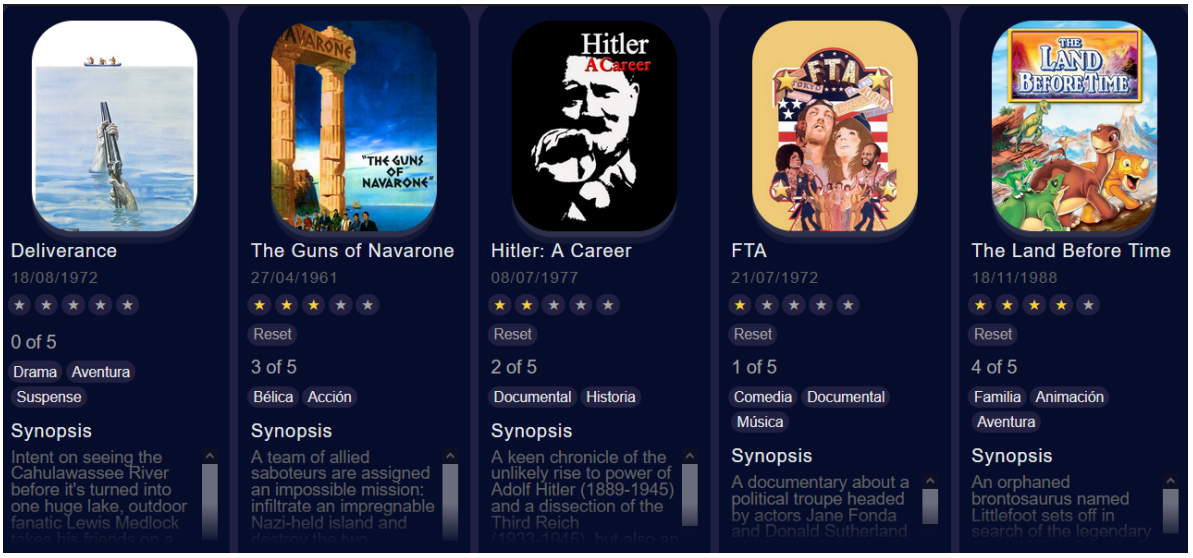
Otras herramientas

- Datasets: Obtenidos para la obtención de los datos iniciales sobre los que trabaja el algoritmo de recomendación. Los mismos fueron obtenidos de MovieLens y kaggle.com.
- FastApi: Para poder utilizar los desarrollos de Backend sin tener que utilizar el Frontend integrado.
- VisualStudio: se estandarizó la utilización de este IDE por el conocimiento y experiencia que se poseía previamente.
- Git: Se utilizó github para el control de versiones.

Funcionalidades y uso

Al ingresar a la aplicación, el usuario se encontrará con una pantalla principal donde se podrá intercambiar entre tres vistas de la aplicación (Películas, Valorados y Tus Configuraciones).

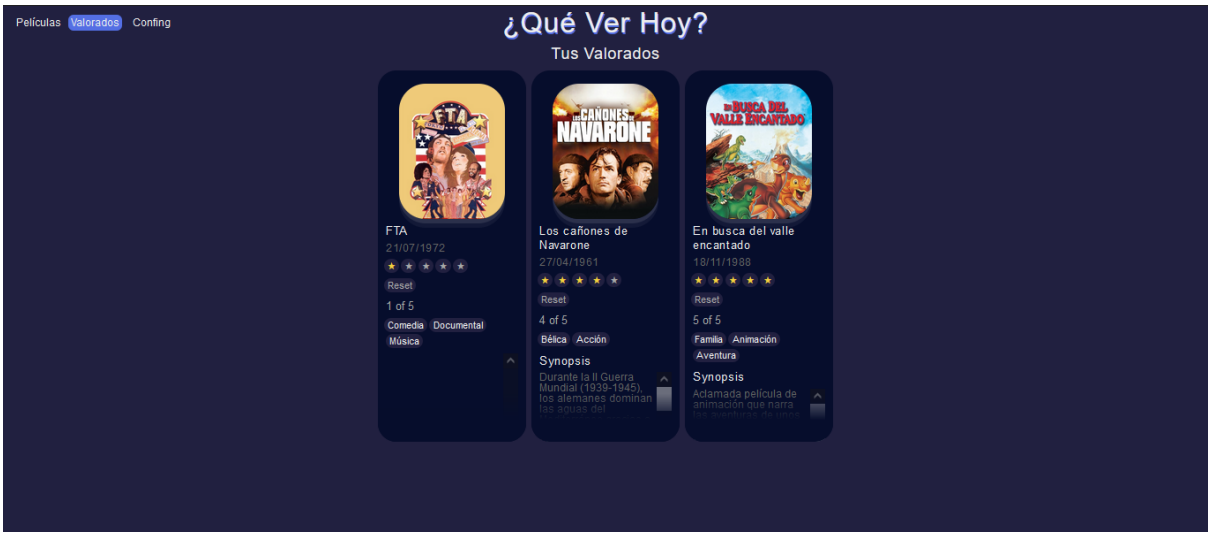
Por defecto, la primera que se visualizará es la de Películas. En ella, si es la primera vez que se ingresa, se verá una lista de películas/series cargadas al azar y el usuario deberá elegir un mínimo de 10 para calificar mediante un sistema de estrellas. Luego de esto se deberá presionar el botón buscar y se pasará a mostrar una lista de recomendaciones generadas a partir de las elecciones anteriores. A medida que el usuario sigue calificando esta lista debería ir mejorando su fiabilidad ya que se contaría con más datos acerca de las preferencias del mismo.



Para cada película en el listado se mostrará una imagen, título, fecha de estreno, categoría y sinopsis.

Adicionalmente para facilitar la búsqueda dentro del listado, se ofrece una lista de filtros a aplicar. Estos son por título del contenido, tipo de contenido, plataforma que posee el usuario.

Por otro lado, si el usuario lo desea, podrá acceder a la vista de Valorados, donde se podrá visualizar una lista de los contenidos que ya fueron calificados por el usuario.



Para mejorar la experiencia de usuario, al acceder a la vista de Config se podrá elegir las plataformas de streaming a las que se tiene acceso, la región donde se encuentra y el lenguaje del contenido que se quiere ver.

Por último se tendrá la vista de Tus Configuraciones, en la cual se podrá elegir la región donde se encuentra el usuario e idioma en el que se desea consumir el contenido.

Limitaciones

- Para el funcionamiento de la aplicación es necesario una lista de preferencias mínima del usuario actual y un cuantioso conjunto de preferencias de otros usuarios.
- A su vez los dataset utilizados son estáticos y no cuentan con los últimos títulos en estreno, lo que podría provocar recomendaciones no del todo actualizadas.
- En el caso que todas las personas que tienen una lista de películas similar a la del usuario actual lleguen a recomendar menos de 10 películas/series (según el algoritmo de filtrado colaborativo) se agregan películas al azar de la base de datos.
- La solución planteada no es escalable debido a algunas de las estructuras utilizadas.

Conclusiones

Se logró obtener una aplicación que cumple los objetivos propuestos, utilizando las herramientas dadas en el curso y profundizando los conocimientos adquiridos mediante ejemplos reales. Además el agregado de opciones de filtrado da una mejor experiencia de usuario ya que complementa el algoritmo y le da robustez a la solución.

Se pudo corroborar el correcto funcionamiento del algoritmo de filtrado colaborativo para el problema planteado y la utilidad de las estructuras como, índices, índices invertidos y diccionarios. Si bien este algoritmo produce buenos resultados cuando es aplicado sobre grandes cantidades de información, la performance es su principal limitación.

El parámetro más importante a definir del algoritmo implementado es la cantidad máxima de vecinos que se considera en el cálculo de la predicción. Existen trabajos que profundizan más sobre esto, haciendo pruebas para comprobar la existencia de un patrón de comportamiento, se trabaja mediante un error medio absoluto en función de la cantidad máxima de vecinos, donde se define un valor adecuado para que el error sea lo más chico posible. En nuestro caso no manejamos este tipo de pruebas ya que la cantidad de vecinos que tomamos fueron todos los usuarios del sistema.

Finalmente, mediante la experiencia realizada se pudo observar la fuerte dependencia del algoritmo de filtrado colaborativo basado en memoria con los datos previamente cargados.

Mejoras y Trabajo a futuro

- Prescindir del dataset inicial para pasar a usar una fuente de datos externa, más completa y actualizada en tiempo real.

- Poder elegir películas según mi preferencia por nombre y no solo las que se me están recomendando.

- En el caso de no tener ninguna película para recomendar, a partir de otros usuarios, poder generar una nueva lista de recomendaciones a partir de otros parámetros.

- Reflejar en la base de datos el género del contenido, para poder generar recomendaciones más específicas y poder obtener algún algoritmo de recomendación en el caso de no recibir resultados por el método principal.

- Añadir nuevas plataformas como Star+, Paramount+, Apple TV y más.

- Añadir la posibilidad de ver trailers a partir de youtube.

- Poder obtener datos de los usuarios a partir de las acciones que realizan sobre el sistema, como tiempo que se permanece en una página o cantidad de clicks en un botón en específico.

Referencias

- Python
- IMDb
- The Movie Database
- SQLite
- Pandas
- PGMÚSICA Sistema de Recomendación de Música - UDELAR
- Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo - Facultad de Ciencias de la Computación - BUAP