



universidad
de león



GRADO EN INGENIERÍA INFORMÁTICA

Sistemas de Información
de
Gestión y Business inteligente
(proyecto **RECOMENDIT**)

Autor: Angelo Cristian De Nicola

Fecha: 19/01/2023

INTRODUCCIÓN	3
PROBLEMA	4
HERRAMIENTAS	6
APLICACIÓN	7
ALGORITMO	15
DAFO	16
LÍNEAS DE FUTURO	17
LECCIONES APRENDIDAS	18
BIBLIOGRAFÍA Y ENLACES	19

INTRODUCCIÓN

En este trabajo analizaremos un problema específico de recomendación y lo resolveremos mediante un algoritmo de recomendación. Para ello, he creado un programa donde se presenta el problema: “**RECOMENDIT**”. El problema específico es recomendar una serie de películas de *Netflix*, especificando la categoría o una película en particular.

El problema concreto es recomendar una serie de películas, especificando la categoría o una película en sí mismo.

DESCRIPCIÓN DEL PROBLEMA

El problema que enfrenté fue la creación de un programa de recomendación de películas. Al comenzar a considerar las diferentes opciones que tenía, lo primero que se me vino a la mente fue cómo lo haría, si lo recomendaría según el género, el director, el reparto, ...

Después de reflexionar, decidí que lo mejor era darte dos opciones: seleccionar directamente las categorías que te interesen o seleccionar una película específica y recibir recomendaciones de películas similares.

Antes de desarrollar un sitio web de recomendación de películas, creo que es fundamental que pueda ser utilizado por cualquier persona, desde un niño que busca una película de animación interesante para ver en su tiempo libre, hasta una persona mayor que busca una película para ver en su tiempo libre.

Creo que la parte más difícil de todo esto es poder **crear una interfaz de usuario simple y comprensible para todos**, porque sí, la aplicación puede ser funcional y hermosa, pero si el usuario promedio no puede usarla, es un fracaso total. para esto me he centrado en una UI (interfaz de usuario) simple que todos puedan entender.

Creo que lo logré!

HERRAMIENTAS

Al comenzar este trabajo, me topé con varios desafíos, muchos de ellos relacionados con las herramientas que debía utilizar. Después de seleccionar el tema, tuve que buscar una **base de datos** adecuada para alcanzar mis objetivos.

Después de probar varias bases de datos a través de Kaggle, pude encontrar una base de datos que, si bien no es particularmente completa porque está restringida solo a títulos de Netflix, podría estar bien para el trabajo que tengo que hacer.

Entonces, mi primer reto fue asegurarme de que mi conjunto de datos estuviera organizado y limpio. Luego, procedí a limpiarlo (eliminando redundancias o incluso registros vacíos) y a eliminar columnas inútiles o innecesarias.

La herramienta principal que tuve que usar fue NEO4J, una herramienta de la que honestamente nunca había oído hablar, pero también fue un gran desafío aprender cosas nuevas. Entonces, lo primero que comencé a hacer esto en **NEO4J** y **Chipher**.

inicialmente la idea era crear **RECOMENDIT** en python, el mejor lenguaje para este tipo de cosas, y lo logré, pero no interactuando muy bien con NEO4J, tuve que pensar en otra cosa. Así que creé **RECOMENDIT** en vue3, pero tenía un error con neo4j y me impedía avanzar. como última versión, así que opté por Java.

Utilicé Java como lenguaje de programación. Una aplicación centrada en el usuario se divide en dos partes: *el front-end*, que es la parte visible para el usuario, y el *back-end*, que se encarga de las conexiones entre la aplicación y la base de datos. Para el front-end, elegí utilizar la librería **swing**, ya que la había utilizado anteriormente.

APLICACIÓN

Mi aplicación comienza creando un conjunto de datos que luego usaré para crear la base de datos. Empecé a buscar en Kaggle un conjunto de datos que encajara con lo que tenía en mente, probé muchos pero encontré uno que encajaba.

id	title	year	score	votes	genres	description
1	The Shawshank Redemption	1994	8.8	26	Drama	Two imprisoned men bond over a number of years, finding solace and hope while they await a chance for freedom.
2	The Godfather	1972	8.7	26	Drama	The story of the rise and fall of a Mafia family in New York City.
3	The Godfather: Part II	1974	8.6	26	Drama	The story of the rise and fall of a Mafia family in New York City.
4	The Godfather: Part III	1976	8.5	26	Drama	The story of the rise and fall of a Mafia family in New York City.
5	The Godfather: The Don	1978	8.4	26	Drama	The story of the rise and fall of a Mafia family in New York City.
6	The Godfather: The Godfather	1980	8.3	26	Drama	The story of the rise and fall of a Mafia family in New York City.
7	The Godfather: The Godfather	1982	8.2	26	Drama	The story of the rise and fall of a Mafia family in New York City.
8	The Godfather: The Godfather	1984	8.1	26	Drama	The story of the rise and fall of a Mafia family in New York City.
9	The Godfather: The Godfather	1986	8.0	26	Drama	The story of the rise and fall of a Mafia family in New York City.
10	The Godfather: The Godfather	1988	7.9	26	Drama	The story of the rise and fall of a Mafia family in New York City.

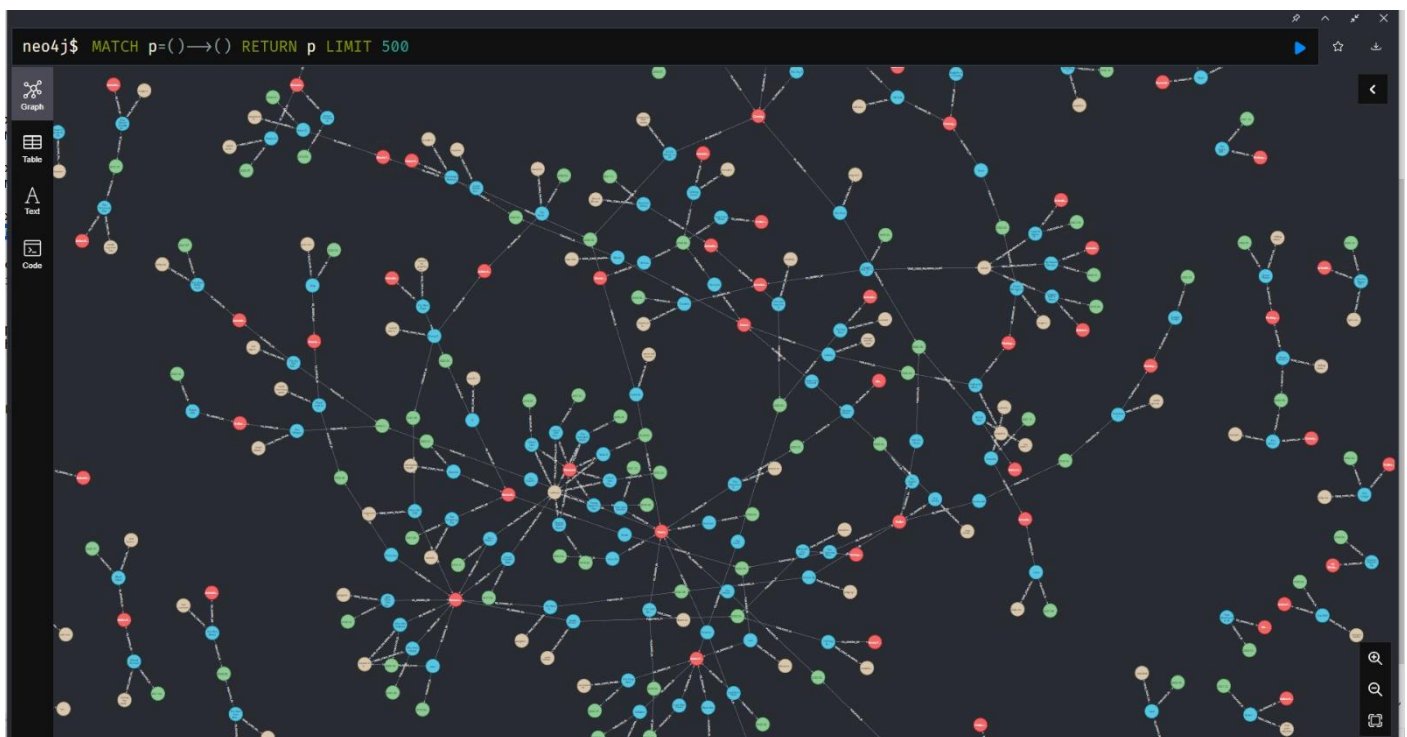
(esto es una query de demostracion para ver las tablas.)

Como se puede observar en el pequeño ejemplo, decidí que cada archivo en el documento será una película en la base de datos, y dentro del archivo tengo varios atributos, incluyendo el título de la película (que será su clave principal)

También me aseguré de que hubiera una forma de evaluación con respecto a la película, que más tarde utilizaría para recomendar películas basándome también en esto.

Una vez obtuve el conjunto de datos, comencé a crear la base de datos utilizando *Neo4J*. Fue bastante sencillo empezar a utilizarlo, ya que tomé dos cursos sobre Cypher y afortunadamente no se alejaba demasiado del lenguaje Sql que había estudiado anteriormente.

Debo decir que estoy muy satisfecho de haber descubierto Neo4j y haber podido trabajar con él, ya que me permitió realizar todos los cambios deseados y manipular el conjunto de datos sin que eso me llevara horas de trabajo para incorporar los cambios en la base de datos.



(Base de datos en NEO4J)

en la foto esta mi base de datos, cada nodo es un elemento conectado a otro nodo a través de una asociación. Los nodos que encontramos son: Películas (que es el nodo principal de esta base de datos), Género, para poder agrupar bien las películas por categoría, y la fecha de estreno, para facilitar las consultas de esa tarea específica.

Una vez configurado todo en Neo4j, llegó el momento de desarrollar lo que sería mi aplicación. Para ello, tuve que dividir el proyecto en dos partes: el frontend y el backend. El backend se encarga de la clase Neo4j, mientras que el frontend está en la clase Main.

El funcionamiento del backend no es muy complejo, además debo mencionar que tengo un algoritmo de recomendación sencillo. Desde el frontend, envío una película o una categoría al backend y mediante una consulta obtengo todas las películas de una determinada categoría. Explicaré con más detalle en la sección sobre el algoritmo de recomendación.

Una vez que tengo todas las películas, los guardo en una lista y la envío al frontend, donde los expondré.

Pasamos al frontend. Inicialmente, mi idea era mostrar inmediatamente las categorías de las películas y partir de ahí. Posteriormente, pensé que sería mejor ofrecer la opción de dar una recomendación basada en una categoría o una película, desarrollando así un frontend que actuaría como menú principal, donde el usuario puede seleccionar el tipo de recomendación que prefiera.



(MENÚ PRINCIPAL – First View)



choose your favorite movie

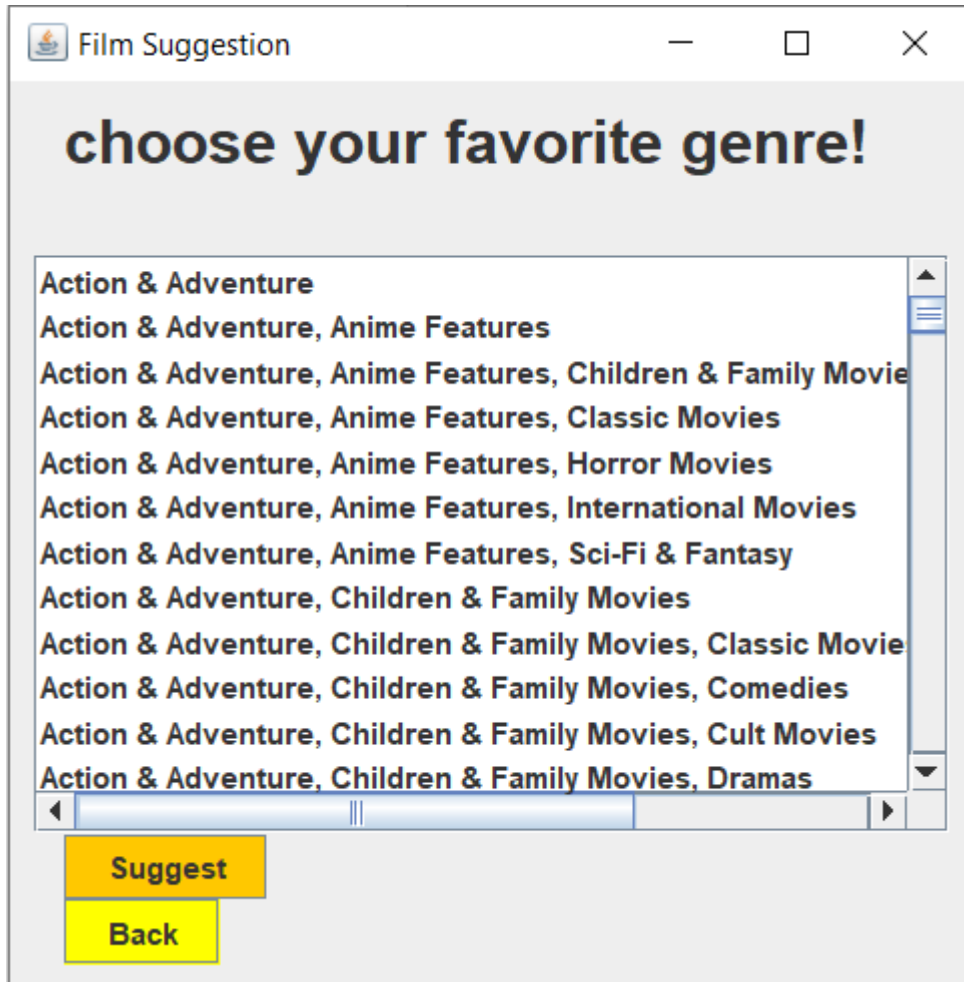
and go make the popcorn!

#Alive
#AnneFrank - Parallel Stories
#FriendButMarried
#FriendButMarried 2
#Roxy
#Rucker50
#Selfie

Suggest

Back

(Sugerencia según película)

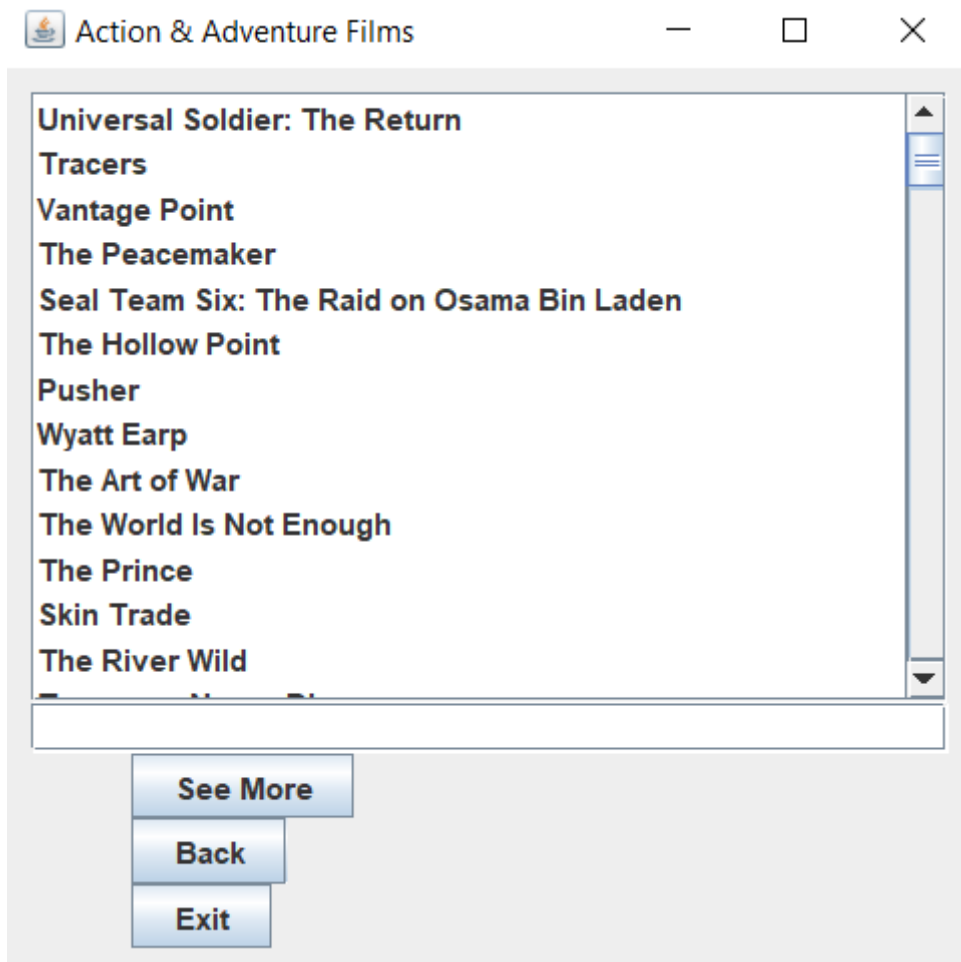


(Sugerencia según categoría)

Ambas páginas disponen de 2 **botones**:

- **Sugerir**: una vez seleccionado la película o la categoría, lo enviará al backend que realizará el sistema de recomendación.
- **Atrás**: para volver a la página anterior.

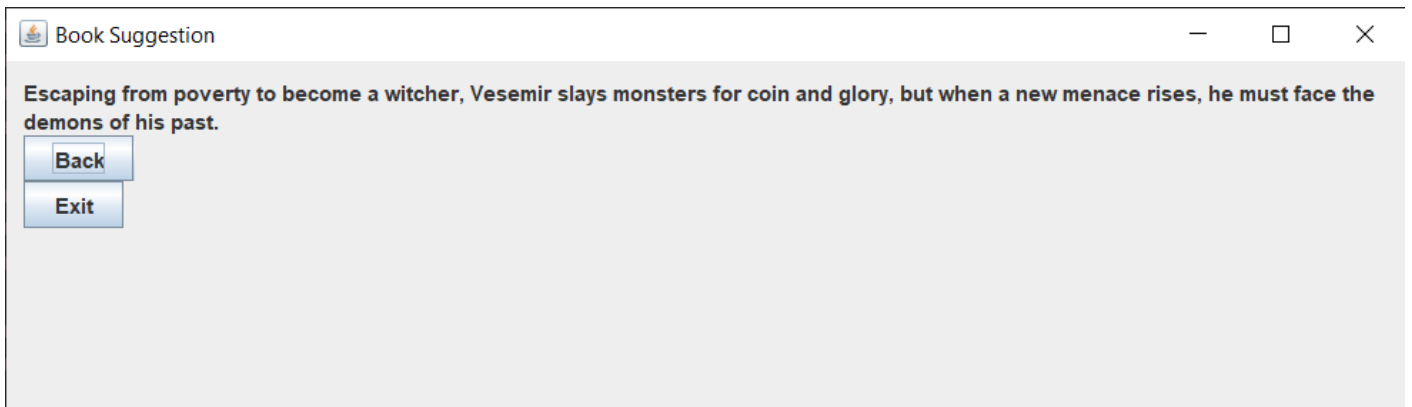
Al mostrar las películas, por comodidad, decidí añadir una barra de búsqueda que actualiza lo que se muestra en tiempo real.



(Películas recomendados por genre "adventura")

Se cambia un botón:

- **See More** :este botón servirá para mostrar la descripción de la películaseleccionada
- **Atrás**: para volver a la página anterior
- **Salir**: que cerrará el programa



(Descripción de la película)

ALGORITMO DE RECOMENDACIÓN

Mi *algoritmo de recomendación* no es muy complejo, básicamente consiste en un filtrado bastante sencillo: una vez que recibo la película (o la categoría), determino la categoría y muestro las películas de ese tipo. Las ordeno por calificación para mostrar primero los resultados más populares entre otros usuarios.

El frontend pasaría la película o la categoría al backend y éste haría la consulta a la base de datos. La comunicación entre el frontend y el backend tiene lugar entre '**App.java**' y '**Database.java**'.

El backend envía la lista al frontend, y en esta ventana introducimos cada película contenido, que ya ha sido previamente ordenado.

DAFO

En esta sección evaluaré las ventajas y desventajas de mi proyecto. Es cierto que mi aplicación cumplió con su función, pero también soy consciente de que podría haber mejorado en algunos aspectos.

Una ventaja de mi aplicación es que siempre devuelve un resultado, independientemente de su uso, porque si bien mi conjunto de datos es específico solo para los títulos de Netflix, es lo suficientemente completo como para devolver siempre un resultado.

Me hubiera gustado incluir una función para guardar las películas favoritas en una base de datos personal. Además, la aplicación cuenta con una gran capacidad de escalabilidad, ya que funciona igual de bien con pequeñas como con grandes cantidades de datos, lo que garantiza un rendimiento constante a medida que la base de datos se expande.

LINEAS DE FUTURO

Empece este proyecto no sabia que era un sistema de recomendacion o como podia moverme en un mundo similar, pero ahora que termino el proyecto debo decir que fue muy util y sin duda una cosa mas que saber.

en un futuro próximo me gustaría mejorar este sistema de recomendaciones y hacerlo más específico y mejor.

o añadir más filtros de búsqueda o mejorar la interfaz gráfica. Tengo tantos puntos en los que inspirarme y estoy seguro de que podría salir algo realmente muy interesante.

Definitivamente que sea una aplicación web, porque todos sabemos que cuando alguien no sabe qué mirar, la primera búsqueda se hace en google, por lo que será muy importante que sea una versión web.

LECCIONES APRENDIDAS

que aprendí? definitivamente la forma de trabajar con NEO4J y Chypher, cómo interactúan con los conjuntos de datos que les alimentamos, cómo se mueven a través de los nodos, cómo incluso interactúan con los diversos lenguajes de programación ya que he usado 3 de ellos.

luego también aprendí a crear un sistema de recomendación en python, ya sabía como funcionaba pero nunca había corrido uno así.

obviamente también el lenguaje VUE3, del cual aprendí que siempre hay una alternativa cuando algo anda mal, solo necesitas mirar hacia otro lado y poder mirar hacia otro lado.

BIBLIOGRAFÍA

- Página de Neo4j:

<https://neo4j.com/>

- Kaggle:

<https://www.kaggle.com>