universidad de león

GRADO EN INGENIERÍA INFORMÁTICA

Sistemas de Información de Gestión y BusinessIntelligenc

MEMORIA

Autor: Catalfamo Rosario

Profesor: Enrique López González

Fecha: 14/01/2023

ÍNDICE

INTRODUCCIÓN

En este trabajo analizaremos un problema concreto de recomendación y lo resolveremos mediante un algoritmo de recomendación. Para ello he realizado un programa donde se expone elproblema.

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

DESCRIPCIÓN DEL PROBLEMA

El problema al que me he enfrentado ha sido realizar un programa de recomendación de películas. Cuando empecé a plantearme las diferentes opciones que tenía lo primero que pensé es de qué forma lo iba a hacer, si iba a recomendar en función del género, del director, cast, ….

Después de pensarlo un poco, decidí que lo mejor era darte dos opciones: seleccionar directamente las categorías que te interesan o seleccionar una película y recibir recomendaciones sobre películas similares.

Ante un sitio de recomendación de películas, creo que es muy importante que cualquiera pueda utilizarlo, desde el niño que busca una película interesante para veer en su tiempo libre, hasta la persona mayor que busca una película para ver en su tiempo libre.

Lo más difícil de todo esto, creo, es hacerlo sencillo y comprensible, porque aunque al final lo más importante es que la aplicación haga su trabajo.

Así que me esforzado en que además de cumplir con su función sea atractiva tanto para niños como para adultos.

HERRAMIENTAS

Cuando empecé este trabajo me encontré con varias dificultades, muchas de ellas relacionadas con las herramientas que iba a utilizar. Una vez elegido el tema sobre el que trabajar, tuve que encontrar un conjunto de datos que se ajustara a lo que quería conseguir.

Luego al consultation de un montón de datos a través del Kaggle, conseguí encontrar una base de datos que, aunque no era especialmente completa, podría cumplir latarea para lo que estaba intentando hacer.

Así que el primer problema era asegurarme de que mi conjunto de datos estaba ordenado y limpio. Después procedí a limpiarlo (eliminando redundancias o incluso registros vacíos) y a eliminar columnas inservibles o simplemente inútiles.

La principal herramienta que tuve que utilizar fue NEO4J, que era totalmente desconocida para mí y debo decir que al principio era muy reacio a utilizarla. Me llevó un tiempo adaptarme, hice un par de cursos sobre Neo4J y al final conseguí cargar mi conjunto de datos.

Decidí tener un único informe para facilitar el filtrado vinculando los películas del conjunto de datos a sus géneros.

Inicialmente mi idea era hacer un sitio web con JavaScript. Esta idea se descartó al surgir problemas durante el desarrollo, como mi inexperiencia con el lenguaje y errores a los que no encontraba solución ni siquiera en línea.

Utilicé Visual Studio para desarrollar la aplicación, ya que lo había utilizado antes y me sentía

cómodo con él. Utilicé Java como lenguaje de desarrollo. Una aplicación centrada en el usuario se divide en dos partes: el front-end, que es la parte centrada en el desarrollo de lo que verá el usuario, y el back-end, que se centra en las conexiones entre la aplicación y la base de datos. Para el front-end, decidí usar la librería swing, ya que la había usado antes.

Para el backend he utilizado y controlador Neo4J para establecer la conexióncon Neo4J.

APLICACIÓN

Mi aplicación comienza con la creación de un conjunto de datos que luego utilizaré para crear la base de datos. Empecé a buscar en kaggle un conjunto de datos que se ajustara a lo que tenía en mente, hasta que encontré uno adecuado.

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmmaker Kirsten Johnson stages his death in inventive and comical ways to help them both face the inevitable.
s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sithole, Cindy Mahlangu, Ryle De Morny, Greteli Fincham, Sello Maake Ka-Ncube	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming star is her sister who was abducted at birth.
s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabiha Akkari, Sofia Lesaffre, Salim Kechiouche, Noureddine Farih, Geert Van Rampelberg, Bakary Diombera		September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Action & Adventure	To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent and deadly turf war.
s4	TV Show	Jailbirds New Orleans				September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down among the incarcerated women at the Orleans Justice Center in New Orleans on this gritty reality series.
s5	TV Show	Kota Factory		Mayur More, Jitendra Kumar, Ranjan Raj, Alam Khan, Ahsaas Channa, Revathi Pillai, Urvi Singh, Arun Kumar	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV Comedies	In a city of coaching centers known to train India's finest collegiate minds, an earnest but unexceptional student and his friends navigate campus life.
s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, Henry Thomas, Kristin Lehman, Samantha Sloyan, Igby Rigney, Rahul Kohli, Annarah Cymone, Annabeth Gish, Alex Essoe, Rahul Abburi, Matt Biedel, Michael Trucco, Crystal Balint, Louis Oliver		September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brings glorious miracles, ominous mysteries and renewed religious fervor to a dying town desperate to believe.
s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, Sofia Carson, Liza Koshy, Ken Jeong, Elizabeth Perkins, Jane Krakowski, Michael Mckean, Phil Lattner		September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria is divided. But a bright-eyed hero believes Earth Ponies, Pegasi and Unicorns should be pals — and, hoof to heart, she's determined to prove it.

(Pequeña muestra del conjunto de datos)

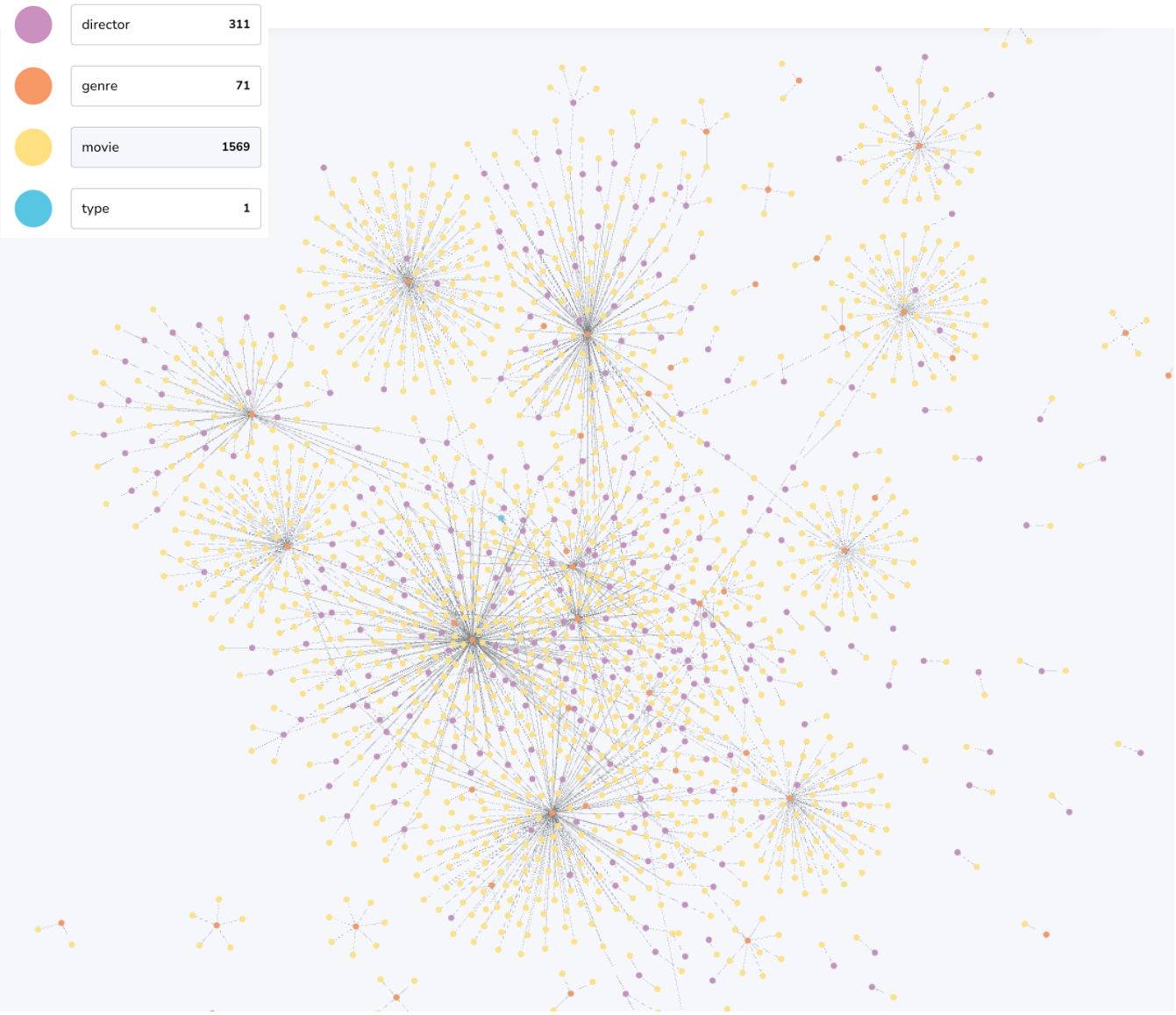
Como puedes ver en el pequeño ejemplo, he decidido que cada fila del documento será una película en la base de datos, y dentro de la fila tengo varios atributos, incluyendo el título de la película.

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 que tuve el conjunto de datos, empecé a crear la base de datos, para lo cual utilicé Neo4J. Fue bastante fácil empezar a usarlo, hice dos cursos sobre Cypher y afortunadamente no se desviaba demasiado del lenguaje Sql que había estudiado antes.

Sin embargo, mirando hacia atrás y viendo los resultados, debo decir que en comparación con otras plataformas que ya había utilizado, como MySQL, Neo4j es mucho más intuitiva y visible una vez que ya has creado la base de datos.

He de decir que estoy muy contento de haber descubierto Neo4j y haber podido trabajar con él, ya que me permitió hacer todos los cambios que quería y trastear mucho con el conjunto de datos sin que eso me llevara horas de trabajo para incluir los cambios en la base de datos.



(Base de datos)

Aquí muestro mi base de datos. Los círculos que aparecen en naranja son las películas, es decir, los atributos que componen la base de datos. Los círculos morados son las categorías de las películas.

El único problema que tuve con la base de datos fue la fiabilidad de mi conjunto de datos básicos. Como dije antes, todo lo que tenía que hacer era eliminar redundancias y filas vacías, lo que pude hacer directamente con neo4j gracias a Cypher.

Una vez que todo estaba configurado en Neo4j, era el momento de crear lo que sería mi aplicación. Para ello, tuve que dividir el proyecto en dos partes: el frontend y el backend. El backend es la parte de la clase Neo4j, mientras que el frontend está en la clase Main.

La parte de la conexión de datos debo decir que me costó un poco y la verdad es que estuve varias horas investigando el uso de la librería neo4j en Java.

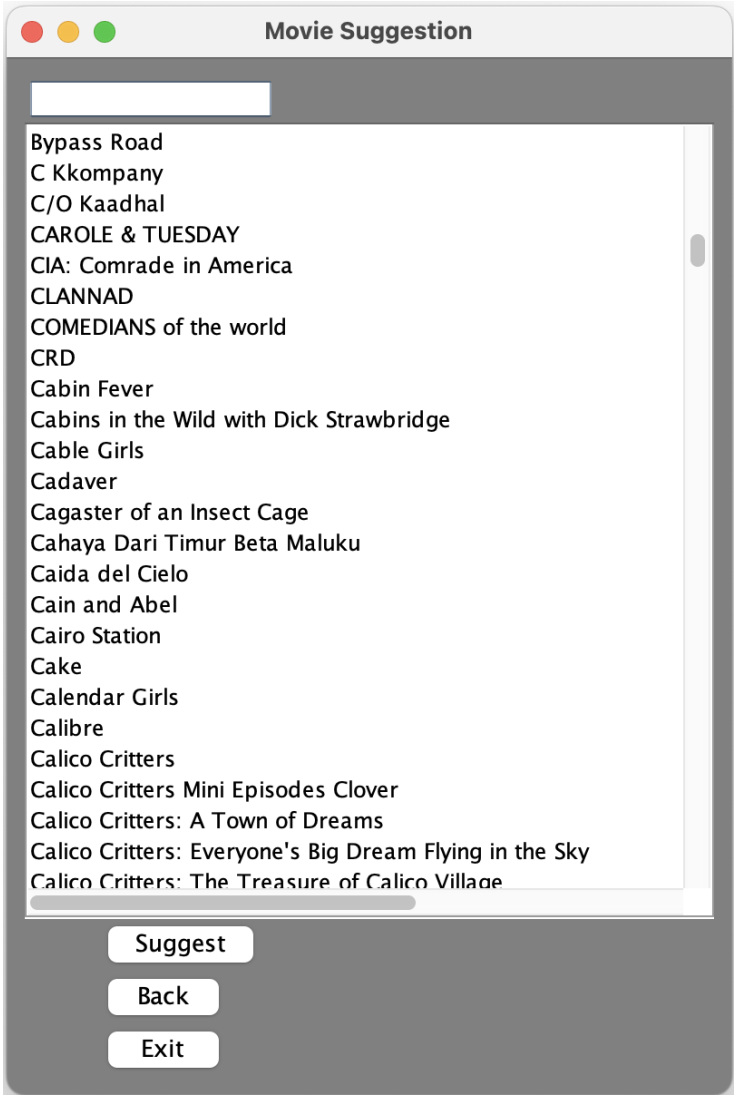
El funcionamiento del backend no es muy complicado, también debo decir que tengo un algoritmo de recomendación simple. Desde el frontend paso una película o una categoría al backend y mediante una consulta obtengo todos las películas de una determinada categoría. Explicaré más en la sección sobre el algoritmo de recomendación.

Una vez que tengo todos las películas, los guardo en una lista y la envío a fronted, donde los expondré.

Pasemos al frontend. En un principio, mi idea era mostrar las categorías de películas inmediatamente y seguir a partir de ahí. Más tarde, discutiendo con mis colegas, pensé que era mejor ofrecer la posibilidad de hacer una sugerencia a partir de una categoría o de una película, desarrollando así un frontend que servirá de menú principal, donde el usuario podrá seleccionar el tipo de sugerencia que más le guste.

(MENÚ PRINCIPAL)

v



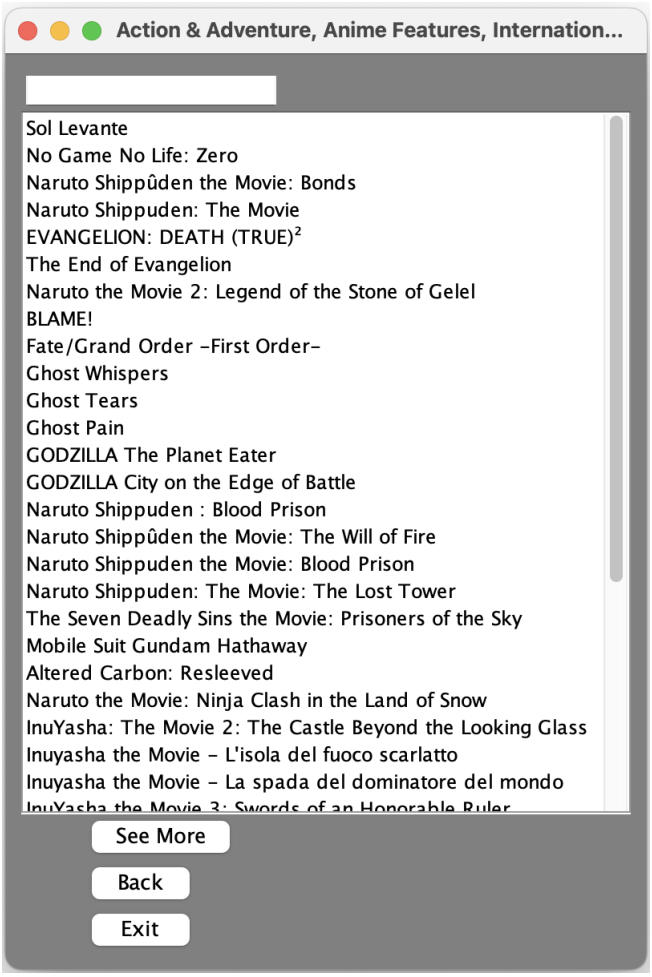
(Sugerencia según película)

(Sugerencia según categoría)

Ambas páginas disponen de 3 botones:

- Sugerir: una vez seleccionado la película o la categoría, lo enviará al backend que realizará una consulta
- Atrás: para volver a la página anterior
- Salir: que cerrará el programa

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

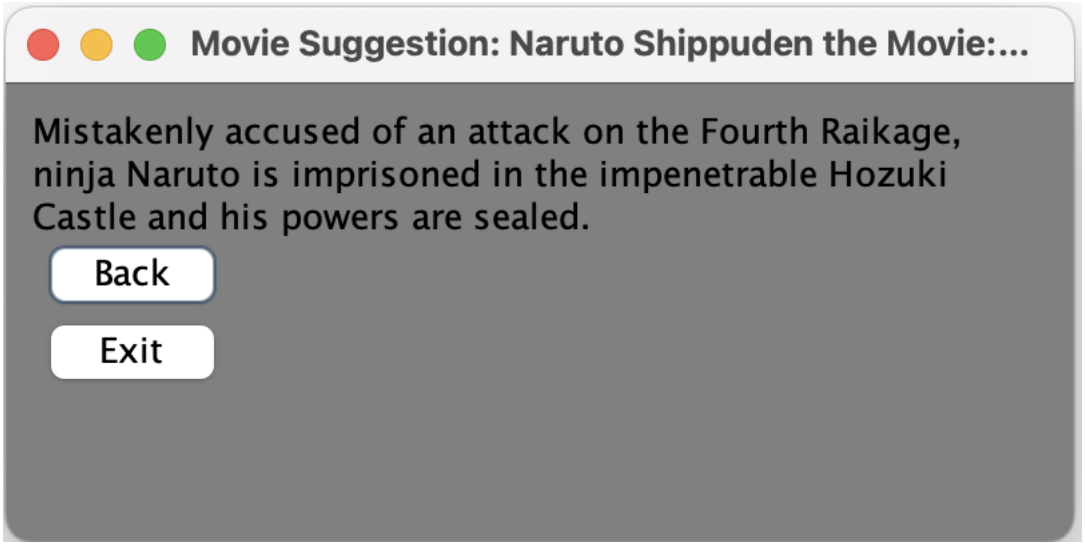


(Películas recomendados)

El título de la ventana cambiará según la sugerencia. Las películas mostradosestán ordenados según la calificación recibida.

Se cambia un botón:

- See More :este botón servirá para mostrar la descripción de la película seleccionada
- 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 complicado, básicamente consiste en un filtrado bastante sencillo: una vez que recibo la película (o la categoría), deduzco la categoría y muestro las películas de ese tipo. Los ordeno por puntuación para recibir primero los resultados más populares de 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 'Main.java' y 'QueryDB.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 analizaré los aspectos positivos y negativos de mi proyecto. Es cierto que mi aplicación cumple con su función, pero también sé que podría haber mejorado en algunos aspectos.

Una ventaja de mi aplicación es que siempre se obtiene un resultado, independientemente del uso que se le dé, debido a que aunque mi conjunto de datos no es muy grande (13000 películas), es suficientemente completo para obtener un resultado siempre. Además, me esforcé en asegurar que fuera eficiente y sin errores.

Un aspecto que me hubiera gustado incluir es la posibilidad de guardar las películas que a uno le gusten en una base de datos para tenerlas siempre a mano. También, una ventaja adicional es que tiene una capacidad de crecimiento muy buena, ya que funciona igual de bien con 200 películas que con 10,000, lo cual es muy ventajoso, ya que permite garantizar que, a medida que crece la base de datos, el rendimiento de la aplicación no disminuirá.

En el futuro, me gustaría agregar una ventana con las películas favoritas, que al ingresar un usuario y una contraseña, te muestre las películas que te han gustado o que te han parecido interesantes.

LÍNEAS DE FUTURO

Al comenzar este proyecto, no tenía una idea clara de lo que podría lograr o alcanzar. Sin embargo, una vez que he completado el trabajo, veo muchas oportunidades para el crecimiento y mejora.

En el futuro, me gustaría profundizar y perfeccionar la búsqueda, incluso considerando los gustos de otros usuarios.

Aunque estoy satisfecho con el trabajo realizado, a medida que avanzaba, me di cuenta de que podría haber hecho más. Por ejemplo, me hubiera gustado agregar un filtrado por género además de por edad, ya que ya tengo la relación en mi base de datos ya que tenía esa idea desde el principio.

Otra cosa que me hubiera gustado incluir es tener también los autores de las películas en la base de datos, para poder hacer recomendaciones basadas en el autor también.

LECCIONES APRENDIDAS

En primer lugar, quiero decir que estoy satisfecho con el trabajo que he realizado, porque es fruto de mi propio esfuerzo. Podría haber hecho más, pero tardé más de lo esperado en conseguir un conjunto de datos limpio y completo sobre el cual trabajar.

La verdad es que aprendí mucho haciendo este trabajo, porque tuve que investigar mucho por mi cuenta, ya que no sabía cómo funcionaba Neo4j, y tuve que esforzarme mucho para aprender Cypher.

Aunque el proyecto inicial en javascript resultó más difícil de lo esperado y me desanimó, volver sobre mis pasos y desarrollarlo en un lenguaje familiar como Java me ayudó mucho. La próxima vez, antes de lanzarme a un proyecto tan desafiante en un idioma desconocido para mí, me aseguraré de practicar primero para evitar quedarme atascado en trivialidades.

BIBLIOGRAFÍA

- Página de Neo4j:

  https://neo4j.com/graphacademy/training-intro-40/02-neo4j-graph- platform/

- Kaggle:
  https://www.kaggle.com