

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC2016 – Algoritmos y Estructura de Datos

Sección 40

Ing. Moises Alonso



Proyecto 2

Fase 1

Diego Leiva	21752
Pablo Orellana	21790

GUATEMALA, 07 de mayo de 2023

Algoritmos de Recomendaciones

Los algoritmos de recomendación, también conocidos como motores de recomendación o sistemas de recomendación, son técnicas de inteligencia artificial y aprendizaje automático utilizadas para proporcionar sugerencias personalizadas a los usuarios de un servicio o producto. Estos algoritmos están diseñados para analizar patrones de comportamiento, preferencias y similitudes entre usuarios y elementos para brindar recomendaciones adecuadas (Franklin, 2022)

Los sistemas de recomendación son ampliamente utilizados en diversas aplicaciones, como comercio electrónico, redes sociales, sistemas de noticias, plataformas de streaming de música y video, entre otros. Estos sistemas ayudan a los usuarios a encontrar contenido relevante y de su interés, lo que puede mejorar la satisfacción del usuario y aumentar la retención y la conversión en aplicaciones y servicios (Koren, Bell, & Volinsky, 2009).

Hay varios tipos comunes de algoritmos de recomendación, entre los que destacan:

- **Filtrado colaborativo:** Este enfoque se basa en la premisa de que los usuarios que han mostrado comportamientos similares en el pasado tendrán intereses similares en el futuro. Los algoritmos de filtrado colaborativo pueden ser de dos tipos: basados en usuarios y basados en elementos
- **Filtrado basado en contenido:** Estos algoritmos analizan las características de los elementos y recomiendan a los usuarios elementos similares a los que han mostrado interés en el pasado
- **Filtrado híbrido:** Los sistemas híbridos combinan diferentes enfoques de recomendación, como filtrado colaborativo y basado en contenido, para mejorar la precisión y la diversidad de las recomendaciones.

(Franklin, 2022)

Los sistemas de recomendación basados en grafos son una variante que utiliza estructuras de datos de grafos para representar y analizar las relaciones entre usuarios y elementos. Estos algoritmos pueden aprovechar técnicas de análisis de redes y caminos más cortos para calcular similitudes y relaciones entre usuarios y elementos (Aggarwal, 2016).

A continuación, se presentan algunos ejemplos de algoritmos basados en grafos:

- **Algoritmo de propagación de etiquetas (Label Propagation Algorithm, LPA):** LPA es un algoritmo de propagación de información que asigna etiquetas a los nodos de un grafo basándose en las etiquetas de sus vecinos. En el contexto de los sistemas de recomendación, los nodos pueden representar usuarios y elementos, y las etiquetas pueden representar preferencias o intereses. El LPA puede utilizarse para propagar información de preferencias entre nodos similares y generar recomendaciones (Raghavan, Albert, & Kumara, 2007).

- Algoritmo de Camino más Corto (Shortest Path Algorithm, SPA): SPA es un algoritmo de análisis de redes que encuentra el camino más corto entre dos nodos en un grafo. En los sistemas de recomendación basados en grafos, SPA puede utilizarse para medir la similitud o la distancia entre usuarios o elementos en función de las relaciones en el grafo. Los elementos cercanos en el grafo se consideran más similares y, por lo tanto, más propensos a ser recomendados (Dijkstra, 1959).
- Algoritmo de PageRank: PageRank es un algoritmo de análisis de redes originalmente desarrollado por Google para medir la importancia de las páginas web en función de la estructura de enlaces en la web. En el contexto de los sistemas de recomendación, PageRank puede adaptarse para medir la importancia de los elementos en función de las relaciones entre usuarios y elementos en el grafo. Los elementos con un PageRank más alto tienen más probabilidades de ser recomendados (Page, Brin, Motwani, & Winograd, 1999).
- Redes Neuronales de Grafos (Graph Neural Networks, GNN): Las GNN son un tipo de redes neuronales que pueden operar directamente en estructuras de datos de grafos. Las GNN pueden aprender representaciones de nodos y aristas a través de la agregación de información de vecindarios locales en el grafo. Estas representaciones aprendidas pueden utilizarse para calcular similitudes entre usuarios y elementos y generar recomendaciones (Kipf & Welling, 2017)

Design Thinking

Empatía

Preguntas de la entrevista.

(Nota: esta entrevista está pensada para individuos que si tienen gusto por el anime)

1. ¿Cuándo y cómo comenzaste a interesarte por el anime? ¿Qué te atrajo de este tipo de animación?
2. ¿Cuáles son tus géneros de anime favoritos y por qué? ¿Podrías mencionar algunos de tus animes favoritos en esos géneros?
3. ¿Qué elementos específicos buscas en un anime? (por ejemplo, trama, personajes, animación, música, etc.)
4. ¿Cuánto tiempo dedicas a ver anime semanalmente? ¿Tienes un horario o rutina específica para verlo?
5. ¿Dónde sueles ver tus animes? (por ejemplo, plataformas de streaming, descargas, televisión, etc.)
6. ¿Sueles ver anime solo o con amigos/familiares? ¿Por qué prefieres esa forma de verlo?
7. ¿Sueles seguir a los creadores, directores o estudios de animación específicos? Si es así, ¿cómo influye esto en tu elección de nuevos animes para ver?
8. ¿Qué importancia le das a la calidad de la animación y el estilo artístico al elegir un anime para ver?

9. ¿Tienes preferencias en cuanto al doblaje (subtítulos en tu idioma vs. doblaje al idioma local) al ver anime? ¿Por qué prefieres una opción sobre la otra?
10. ¿Qué te motiva a recomendar un anime a alguien más? ¿Cómo eliges a quién recomendar un anime en particular?
11. ¿Qué papel juegan las bandas sonoras y la música en tu disfrute de un anime?
12. ¿Cómo te enteras de nuevos animes que podrían interesarte? (por ejemplo, amigos, redes sociales, blogs, foros, etc.)
13. ¿Alguna vez has utilizado herramientas o aplicaciones para buscar recomendaciones de animes? Si es así, ¿cuáles y cómo fue tu experiencia con ellas?
14. ¿Cuál ha sido la mejor recomendación de anime que has recibido y cómo la obtuviste?
15. ¿Qué factores influyen en tu decisión para ver un nuevo anime? (por ejemplo, calificaciones, reseñas, género, sinopsis, etc.)
16. ¿Qué crees que podría mejorar en las herramientas o aplicaciones actuales que te ayudan a descubrir nuevos animes?
17. Si pudieras diseñar la herramienta o aplicación perfecta para encontrar recomendaciones de animes, ¿qué características tendría?

Definición

Luego de haber analizado las respuestas que se obtuvieron a partir de las entrevistas se puede sintetizar lo siguiente:

- Los aficionados del anime ven importante el género y/o la trama para que un anime les llame la atención.
- La mayoría de las recomendaciones las obtienen a través de redes sociales como Instagram, Facebook o incluso YouTube.
- El rating del anime es importante a la hora de seleccionar el próximo anime por ver.
- Actualmente no existe una herramienta que les permita buscar y/o explorar nuevos animes que se acoplen a sus gustos y tendencias.
- Las plataformas actuales no cuentan con mucha personalización para recomendación de animes.
- Tanto los estudios de animación como los autores de las historias son parte importante en el proceso de selección de animes nuevos por ver, pues señalan que, si el autor tiene una muy buena historia, es muy probable que también tenga otras historias del mismo nivel.

Estas son algunos de los hallazgos mas importantes de las entrevistas. Por lo tanto, se puede inferir que la problemática principal actual para muchos aficionados al anime es que necesitan una forma eficiente y personalizada de descubrir nuevos animes los cuales se ajusten a sus preferencias en cuanto a géneros, estudios, autores, trama y/o rating. Ya que actualmente no cuentan con una herramienta especifica que facilite este proceso, o en su defecto desconocen de la existencia de herramientas útiles de este tipo.

Ideación

Diagrama lluvia de ideas



Ideas prioritarias:

- El usuario en si no tendrá mucha información personal más allá de su usuario y contraseña, sin embargo, el tener un usuario permite guardar información específica de cada individuo.
- Cuestionario de preferencias para el usuario, de esta forma se tiene un conocimiento base de que le gusta y que no le gusta al usuario.
- Clasificaciones, estas pueden ayudar para filtrar y recomendar de manera más eficiente diferentes tipos de anime al usuario, basándose en sus gustos.

Ideas secundarias, que son buenas, pero no son esenciales:

- Búsqueda por nombre de animes
- Filtrado por idioma o popularidad
- Exploración de contenido

Prototipos de baja fidelidad

Log in

Bienvenido

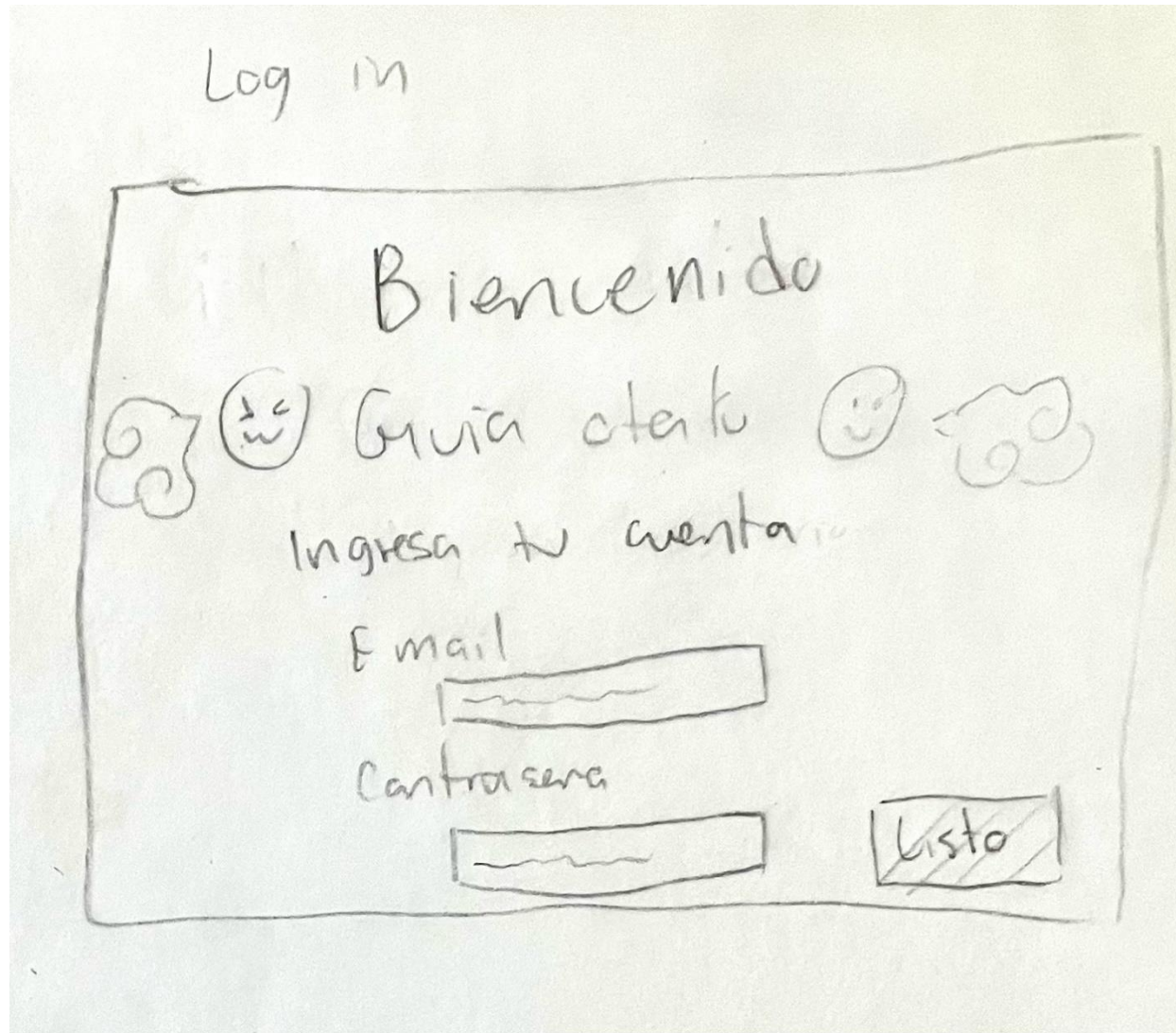
Guía de inicio

Ingresa tu cuenta

Email

Contraseña

Listo



2.5 Guía Otaku

Pantalla para crear usuario

Bienvenido a

Guía Otaku

crea tu cuenta

Usuario

.....

Email

.....

Contraseña

.....

☐ Listo!

Pantalla de inicio de sesión

Antes de comenzar

Conozcamos tus gustos 😊

¿Qué anime has visto?

escribir

Generos favoritos

escribir

Animes que te gustaría ver

escribir

opción 2 (inicio de sesión)

😊 Antes de comenzar 😊
Conozcamos tus gustos
* selección de 1 a 4 opciones.

Anime fav	Generos fav	Quiénes ver
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

👉 Listo! 👈

Menú (Opción 1)

😊 Guía Otaku 😊

Menú

Selecciona la opción de deseos

Explorar	Recomendar	Mis gustos
----------	------------	------------

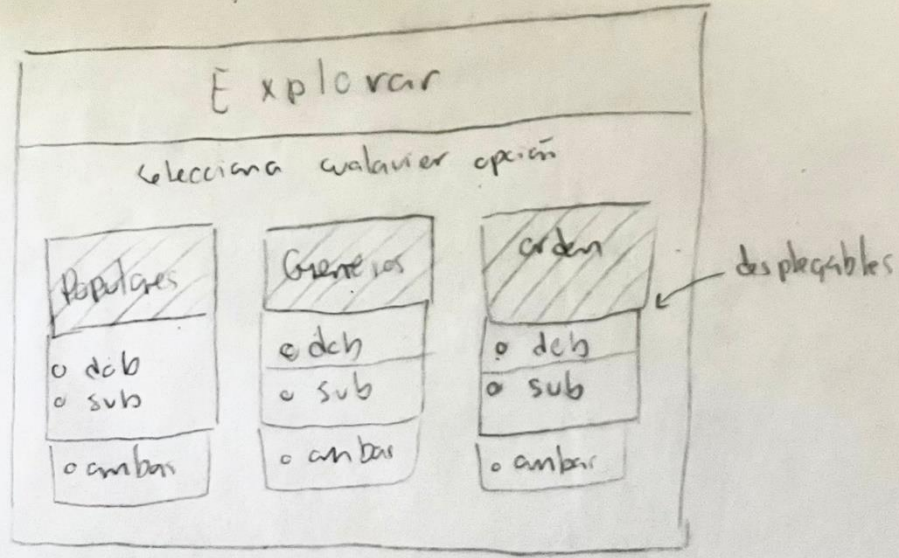
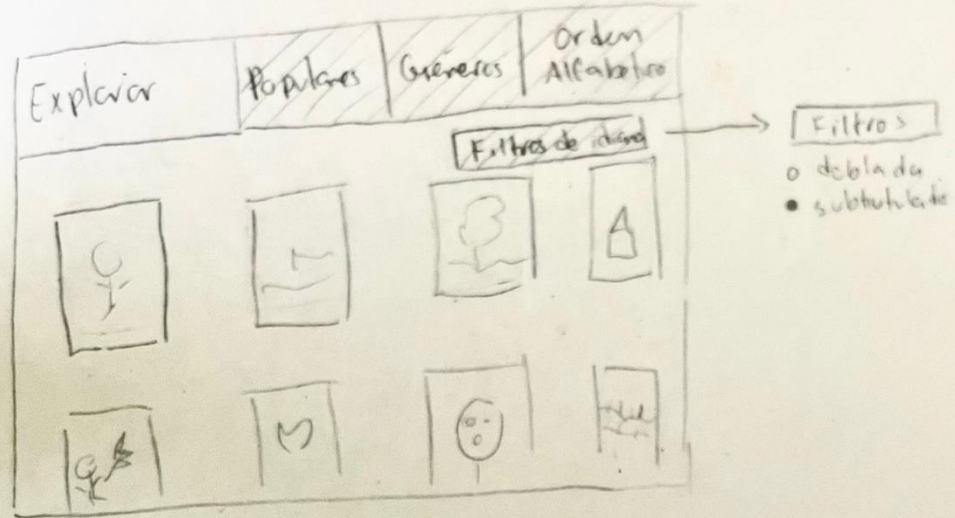
opción 2

Menú

explora	Recomen	Mis gustos
---------	---------	------------

😊 Guía otaku 😊

pestaña de explorar (opción 1)



pestaña recomendaciones

Recomendados

selecciona cualquier opción

Top 10

Crémenes
más vistos

Autores
favoritos

Capítulos
(25 min - 30 min)

pestaña mis gustos

Mis gustos

Crémenes fav

Autores fav

Idioma fav

Tiempo fav

Testing

Tomando en cuenta los diferentes nodos y atributos de estos, así como las relaciones entre ellos podemos concluir que el diseño presentado para la solución será efectivo, pues al obligar al usuario a responder un cuestionario inicial respecto a sus gustos, podemos generar las relaciones principales que nos darán la oportunidad de buscar otros animes que puedan ajustarse a los gustos de nuestros usuarios. La base de datos y el algoritmo están formulados para que estos se puedan modificar en el tiempo, pues sabemos que las personas tienen opiniones que cambian con el paso del tiempo.

La interfaz permitirá al usuario realizar diferentes acciones, como explorar animes, o ver que se le recomienda. Esto mediante el uso de los diferentes atributos de cada categoría de nodo, para hacer los cálculos de similitudes y puntajes de cada anime. Asimismo, con estos mismos atributos es posible filtrar animes que se acoplen a las tendencias que tienen los usuarios.

Ahora bien es importante mencionar que para que esta herramienta llegue a funcionar de forma adecuada, es necesario responder el cuestionario inicial, ya que sin este, no es posible generar ninguna relación que pueda llevar al algoritmo más allá de la fase de recopilación de datos del usuario.

Para corroborar lo anterior hemos hecho un pequeño testeo de nuestro prototipo con un posible usuario, su opinión está documentada en el repositorio de GitHub.

Algoritmo Diseñado

Breve explicación del algoritmo elaborado:

- Inicialización: Preprocesar y cargar el catálogo de animes y sus metadatos, como género, autor, estudio de animación, rating, etc.
- Recopilación de datos del usuario: Obtener las respuestas del cuestionario de preferencias del usuario, incluyendo géneros de interés, animes que le han gustado y no le han gustado, autores de interés y estudios de animación de interés.
- Crear un perfil de preferencias del usuario: A partir de las respuestas del cuestionario, crear un perfil de preferencias que represente los intereses del usuario en términos de género, autor, estudio de animación, entre otros.
- Calcular la similitud entre animes y preferencias del usuario: Utilizando una medida de similitud, calcular la similitud entre cada anime en el catálogo y el perfil de preferencias del usuario.
- Ordenar los animes por similitud: Ordenar los animes en función de las similitudes calculadas en el paso anterior, de mayor a menor similitud.
- Filtrar y seleccionar animes relevantes: Aplicar filtros adicionales, como idioma o duración, y seleccionar los animes más relevantes según las preferencias del usuario y los criterios de calidad o popularidad.
- Generar recomendaciones: Tomar los animes ordenados y filtrados y presentarlos como recomendaciones personalizadas para el usuario.
- Ajustar el algoritmo según la retroalimentación del usuario: Utilizar la información de interacción del usuario con las recomendaciones y su retroalimentación para ajustar y mejorar el algoritmo a lo largo del tiempo.

Pseudocódigo del algoritmo:

función obtener_recomendaciones(usuario, catálogo_animes, n_recomendaciones):

Paso 1: Cargar catálogo de animes y metadatos

catálogo = cargar_catálogo(catálogo_animes)

Paso 2: Recopilar datos del usuario

preferencias_usuario = obtener_preferencias(usuario)

Paso 3: Crear perfil de preferencias del usuario

perfil_usuario = crear_perfil_usuario(preferencias_usuario)

Paso 4: Calcular similitud entre animes y perfil del usuario

similitudes = []

para cada anime en catálogo:

similitud = calcular_similitud(perfil_usuario, anime)

similitudes.append((anime, similitud))

Paso 5: Ordenar los animes por similitud

similitudes.sort(reverse=True, key=lambda x: x[1])

Paso 6: Filtrar y seleccionar animes relevantes

animes_filtrados = filtrar_animes(similitudes, preferencias_usuario)

Paso 7: Generar recomendaciones

recomendaciones = animes_filtrados[:n_recomendaciones]

return recomendaciones

función cargar_catálogo(catálogo_animes):

Cargar el catálogo de animes y sus metadatos

return catálogo_animes

función obtener_preferencias(usuario):

Obtener las preferencias del usuario a partir del cuestionario

return preferencias_usuario

función crear_perfil_usuario(preferencias_usuario):

Crear un perfil de usuario basado en sus preferencias

return perfil_usuario

función calcular_similitud(perfil_usuario, anime):

Calcular la similitud entre el perfil del usuario y un anime

return similitud

función filtrar_animes(similitudes, preferencias_usuario):

Filtrar animes según criterios adicionales

return animes_filtrados

Modelo de base de datos

Diseño preliminar de la base de datos:

Tipos de nodos

- Usuario
 - Nombre
 - Contraseña
- Anime
 - Título
 - Descripción
 - Duración
 - Capítulos
 - Rating
 - Emisión
 - Subtitulado
 - Doblado
- Autor
 - Nombre
 - Edad
- Estudio
 - Nombre
- Genero
 - Nombre

Tipos de relaciones:

- Le gusto (de usuario a anime)
- No le gusto (de usuario a anime)
- Creador de (de autor a anime)
- Animación de (de estudio a anime)
- Pertenece a (de anime a genero)
- Le interesa (de usuario a género, autor, estudio)
- No le interesa (de usuario a genero)

Query generador de base de datos inicial:

```
CREATE(DiegoL:Usuario{username:'puxter',contraseña: 'admin123'})
CREATE(PabloO:Usuario{username:'pabloski',contraseña: 'admin345'})
```

```
CREATE(GegeAk:Autor{nombre:'Gege Akutami', edad: 31})
CREATE(HiromuA:Autor{nombre:'Hiromu Arakawa', edad: 50})
CREATE(HajimeI:Autor{nombre:'Hajime Isayama', edad: 36})
```

```
CREATE(JujutsuK:Anime{nombre:'Jujutsu Kaizen', descripcion:'Dificultades,
arrepentimiento, vergüenza... Los sentimientos negativos de los humanos se convierten en
Maldiciones que nos acechan en nuestra vida diaria.', duracion: 23, capitulos:24, rating:4.9,
emision:'en emision', subtitulado:'si',doblado:'si'})
```

```
CREATE(AOT:Anime{nombre:'Shingeki no Kyojin', descripcion:'Muchos años atrás, la
humanidad estuvo al borde de la extinción con la aparición de unas criaturas gigantes que
devoraban a todas las personas.', duracion: 20, capitulos:88, rating:4.9, emision:'finalizado a
medias', subtitulado:'si',doblado:'si'})
```

```
CREATE(FMA:Anime{nombre:'Full Metal Alchemist: Brotherhood', descripcion:'Los
hermanos Elric rompieron uno de los más importantes tabúes de la alquimia, de modo que Ed
perdió casi la mitad de su cuerpo, ahora reemplazado por partes mecánicas, y su hermano
menor Al, quedó convertido en un alma ligada a una armadura', duracion: 20, capitulos:64,
rating:4.9, emision:'finalizado', subtitulado:'si',doblado:'si'})
```

```
CREATE(MAPPA:Estudio{nombre:'MAPPA'})
CREATE(BONES:Estudio{nombre:'BONES'})
```

```
CREATE(Shonen:Genero{nombre:'Shonen'})
CREATE(Accion:Genero{nombre:'Accion'})
CREATE(FantasiaOsc:Genero{nombre:'Fantasia Oscura'})
```

```
CREATE
(DiegoL)-[:LE_GUSTA]->(JujutsuK),
(DiegoL)-[:LE_GUSTA]->(AOT),
(PabloO)-[:LE_GUSTA]->(FMA),
(PabloO)-[:LE_GUSTA]->(AOT),
```

```
(PabloO)-[:LE_INTERESA]->(MAPPA),
(PabloO)-[:LE_INTERESA]->(Shonen),
(DiegoL)-[:LE_INTERESA]->(HajimeI),
```

```
(AOT)-[:ANIMACION_DE]->(MAPPA),
(JujutsuK)-[:ANIMACION_DE]->(MAPPA),
```


(FMA)-[:ANIMACION_DE]->(BONES),

(HajimeI)-[:CREADOR_DE]->(AOT),

(HiromuA)-[:CREADOR_DE]->(FMA),

(GegeAk)-[:CREADOR_DE]->(JujutsuK),

(AOT)-[:PERTENECE_A]->(Shonen),

(AOT)-[:PERTENECE_A]->(Accion),

(AOT)-[:PERTENECE_A]->(FantasiaOsc),

(FMA)-[:PERTENECE_A]->(Shonen),

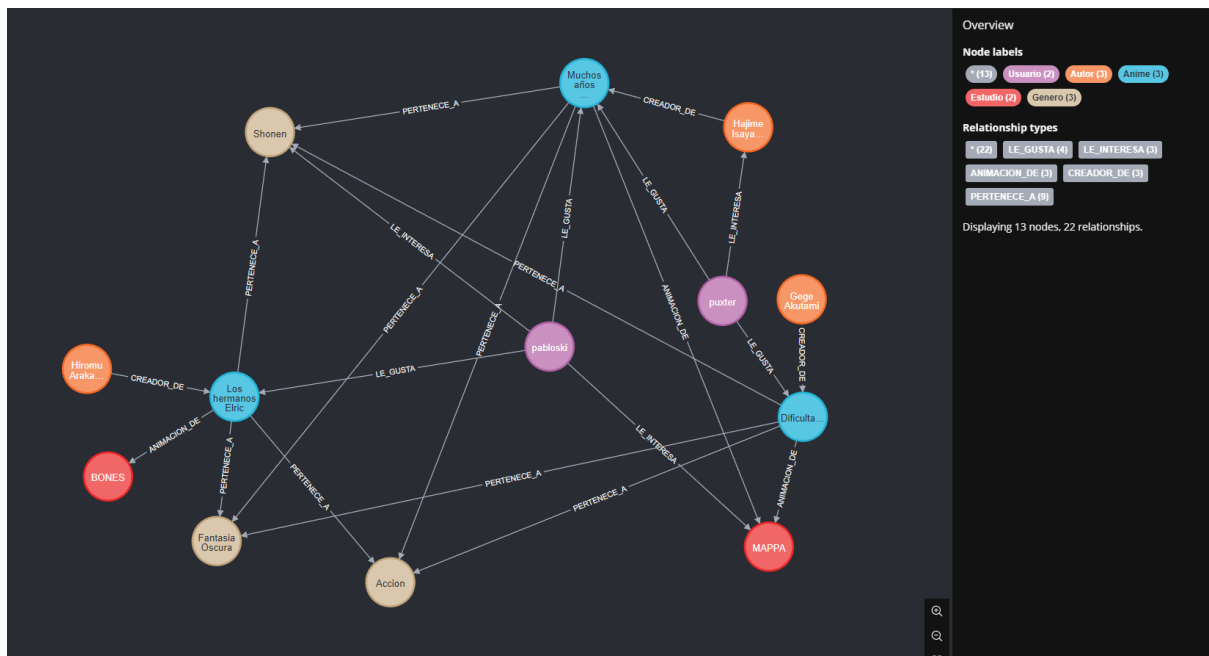
(FMA)-[:PERTENECE_A]->(Accion),

(FMA)-[:PERTENECE_A]->(FantasiaOsc),

(JujutsuK)-[:PERTENECE_A]->(Shonen),

(JujutsuK)-[:PERTENECE_A]->(Accion),

(JujutsuK)-[:PERTENECE_A]->(FantasiaOsc)



En esta base de datos se encuentran varias relaciones, unas relacionan a los autores y a los estudios con los anime donde tuvieron participación. Mientras que otra se encarga de relacionar el anime con uno o más géneros en específico, pues así como las películas y series, estas generalmente no entran en una sola categoría. Finalmente están las relaciones de gustos e intereses del usuario, en donde se hace una unión entre el nodo usuario y uno o varios nodos de autores, estudios, géneros o anime. Esto en combinación permitirá generar una recomendación más robusta para nuestro usuario objetivo, pues toma en cuenta más de un solo factor.

Referencias

- Aggarwal, C. (2016). Recommender Systems: The Text Book. Springer.
http://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269-271.
- Franklin, N. (2022). What Is a Recommendation Engine? How Recommenders Work. Amplitude.
<https://amplitude.com/blog/recommendation-engine#:~:text=Recommendation%20engines%20are%20advanced%20data%20filtering%20systems%20that%20use%20behavioral,businesses%20that%20offer%20personalized%20experiences>.
- Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Koren, Y., Bell, R., & Volinsky, C. (2009). MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS. *IEEE Computer Society*, 42(8).
[https://datajobs.com/data-science-repo/Recommender-Systems-\[Netflix\].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf)
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. *Stanford InfoLab*.
- Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 036106.

Enlace repositorio de GitHub:

<https://github.com/LeivaDiego/Proyecto2-Fase1-AED.git>