

TAREA 4: Grape Loot

Manuel González[†], Pablo Gómez[†], Ayrton Morrison[†] y Emmanuel Velásquez[†]

[†] Universidad de Magallanes

Este informe fue compilado el 15 de diciembre de 2024

Resumen

En el presente informe técnico se detallará el planteamiento y desarrollo de una red social simplificada, mostrando el uso de algoritmos de similitud, manejo de archivos y uso de estructuras de datos eficientes para esto.

Keywords: Estructuras de datos, algoritmos, manejo de archivos

■ Índice

1	Introducción	2
2	Objetivo principal	2
3	Planteamiento del desarrollo del proyecto	3
4	Implementación	4
5	Gestión del equipo de trabajo	6
6	Posibles Mejoras a futuro	7
7	Ejemplo de uso de la red social	8
8	Conclusiones	9

1. Introducción

La presente tarea tiene como propósito consolidar los conocimientos adquiridos mediante el diseño e implementación de un sistema práctico. En esta ocasión, el proyecto se centra en el desarrollo de una red social simulada, un entorno que permitirá implementar y evaluar diversos algoritmos de recomendación y técnicas relacionadas con redes sociales. Este entorno incluye el uso de estructuras de datos avanzadas como grafos y listas enlazadas, integrando también algoritmos como la similitud de Jaccard. El objetivo general es simular un sistema que facilite recomendaciones personalizadas basadas en la importancia y similitud de nodos en una red, resolviendo desafíos técnicos relacionados con la gestión dinámica de datos en estructuras complejas.

2. Objetivo principal

El objetivo principal es diseñar e implementar un sistema funcional que permita gestionar perfiles de usuario, conexiones entre ellos y operaciones asociadas como publicaciones, búsquedas y recomendaciones. Este proyecto busca aplicar conceptos fundamentales de estructuras de datos y algoritmos avanzados para resolver problemas reales relacionados con la gestión y optimización de redes sociales simuladas.

3. Planteamiento del desarrollo del proyecto

El desarrollo de este proyecto se centra en la implementación de una red social simulada basada en grafos. Los nodos representan perfiles de usuario y los bordes, sus conexiones (seguidores y seguidos). Los algoritmos principales empleados incluyen:

- Similitud de Jaccard: Implementado para calcular la semejanza entre los intereses o conexiones de diferentes usuarios.

El proyecto incorpora una estructura modular, separando tareas como el almacenamiento de datos en archivos, la simulación de conexiones entre usuarios y el cálculo dinámico de métricas para optimizar el rendimiento.

Para la implementación del índice de Jaccard, se utilizó una tabla hash para las listas de preferencias de cada usuario. Esto permite comparar eficientemente las preferencias de dos usuarios y calcular el índice de Jaccard.

A continuación, se presenta un ejemplo gráfico de la implementación de la similitud de Jaccard:

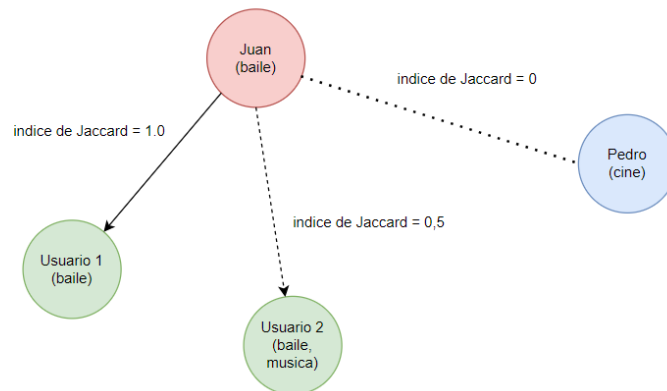


Figura 1. Ejemplo gráfico de la implementación de la similitud de Jaccard.

4. Implementación

La implementación se desarrolla en etapas:

■ Estructuras de datos:

- Uso de grafos para modelar las relaciones entre usuarios.
- Implementación de listas enlazadas y tablas hash para gestionar datos asociados a cada usuario.

■ Algoritmos:

- Cálculo de la similitud de Jaccard para sugerencias basadas en intereses comunes.

■ Manejo de datos:

- Almacenamiento estructurado en directorios para cada usuario.
- Archivos separados para guardar seguidores, seguidos y datos generales de los usuarios.

■ Generadores dinámicos:

- Creación automática y controlada de usuarios.
- Generación de conexiones aleatorias y recomendaciones personalizadas.

■ Interfaz de interacción:

- Desarrollo de una interfaz de línea de comandos para manejar la creación, modificación y búsqueda de usuarios.
- Ejecución a través de parámetros para una interacción dinámica.

Cálculo de la Similitud de Jaccard

- Para la implementación del índice de Jaccard, se utilizó una tabla hash para las representar las preferencias de cada usuario. Esto maximiza la eficiencia al realizar comparaciones necesarias entre las preferencias de dos usuarios y facilita el cálculo de dicho índice.

Creación Automática de Usuarios

- Para la creación automática de usuarios, se creó un modelo matemático que calcula la máxima cantidad de usuarios que podrían crearse según el número de perfiles en la red. Cada cierto intervalo de tiempo, se verifica el número de usuarios en la red y, en base a eso, se calcula la máxima cantidad de perfiles utilizando el siguiente modelo:

$$U_{\text{máx}}(n_0) = \left(\frac{n_0 + 180}{n_0} \right)^{\frac{2}{3}}$$

donde:

- $U_{\text{máx}}$ es la máxima cantidad de usuarios.
- n_0 es el número de usuarios actual en la red.

Después de usar el modelo matemático para limitar la cantidad de usuarios que se puedan crear, se genera un número aleatorio entre 0 y el valor devuelto por la función. Este número representa la cantidad de usuarios a crear.

Justificación del Modelo Matemático

- El modelo matemático simula el crecimiento controlado de la red social, asegurando que cuando la cantidad de usuarios en la red alcance o supere los 100, solo se pueda crear un usuario nuevo a la vez. Esto permite que la red crezca de forma continua, pero cada vez más suave.

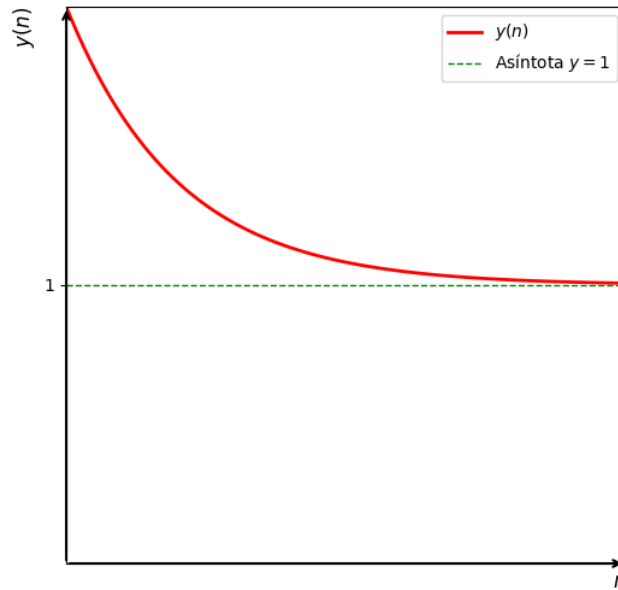


Figura 2. Ejemplo.

La creación del modelo se puede explicar en las siguientes expresiones matemáticas, en donde podemos observar el por qué este modelo toma los comportamientos indicados anteriormente.

$$\lim_{n \rightarrow \infty} y(n) = 1 \quad (\text{Asíntota horizontal en } y = 1) \quad (1)$$

$$\left. \frac{dy}{dn} \right|_{n=m} < 0 \quad \forall m \in \mathbb{R}^+ \quad (\text{Decreciente}) \quad (2)$$

$$U_{\max}(n) = \lfloor y(n) \rfloor \quad (\text{Discretización del modelo}) \quad (3)$$

5. Gestión del equipo de trabajo

El equipo de trabajo está compuesto por los siguientes integrantes: Ayrton Morrison, Emmanuel Velásquez, Manuel González y Pablo Gómez. La división de responsabilidades se organizó en dos grupos principales: el equipo de programación y el equipo de documentación.

Equipo de programación

El equipo de programación es responsable del diseño y desarrollo del código del sistema. Sus principales tareas incluyen:

- **Diseño de la arquitectura del sistema:** Planificación de la estructura y componentes del sistema para asegurar una implementación eficiente y escalable.
- **Desarrollo del algoritmo de similitud de Jaccard:** Implementación del algoritmo para calcular las similitudes entre los intereses de los usuarios, optimizando la precisión de las recomendaciones.
- **Integración de módulos y pruebas:** Asegurarse de que todos los componentes del sistema funcionen de manera coherente y realizar pruebas para detectar y corregir errores.
-

Equipo de documentación

El equipo de documentación se encarga de la redacción del informe y la elaboración de diagramas y documentación adicional. Sus principales responsabilidades son:

- **Redacción del informe del proyecto:** Documentación detallada del desarrollo del sistema, incluyendo objetivos, metodologías y resultados.
- **Elaboración de diagramas:** Creación de diagramas de arquitectura del sistema y otros gráficos que faciliten la comprensión del proyecto.
- **Documentación técnica:** Desarrollo de manuales de usuario y guías técnicas para asegurar que otros desarrolladores puedan entender y trabajar con el sistema.
- **Revisión y edición:** Asegurarse de que toda la documentación sea clara, precisa y esté libre de errores.

Coordinación y comunicación

Para asegurar una colaboración efectiva entre ambos equipos, se establecieron las siguientes prácticas de gestión:

- **Reuniones semanales:** Sesiones de actualización para revisar el progreso, discutir desafíos y planificar las siguientes etapas del proyecto.
- **Revisión de pares:** Proceso donde los miembros del equipo revisan y proporcionan retroalimentación sobre el trabajo de sus compañeros para mejorar la calidad del proyecto.

Estas prácticas garantizan que todos los miembros del equipo estén alineados con los objetivos del proyecto y que se mantenga una alta calidad en el desarrollo y la documentación.

6. Posibles Mejoras a futuro

En esta sección se describen algunas posibles mejoras que se podrían implementar en el futuro para optimizar y ampliar el proyecto:

- **Optimización del rendimiento:** Revisar y mejorar el código para reducir el tiempo de ejecución y el uso de recursos.
- **Ampliación de funcionalidades:** Añadir nuevas características que puedan ser útiles para los usuarios, basadas en el feedback recibido.
- **Mejora de la documentación:** Asegurarse de que toda la documentación esté actualizada y sea fácil de entender para nuevos desarrolladores.
- **Pruebas automatizadas:** Implementar un conjunto completo de pruebas automatizadas para garantizar la estabilidad y la calidad del código.
- **Interfaz de usuario:** Mejorar la interfaz de usuario para hacerla más intuitiva y fácil de usar.
- **Manejo de memoria:** El programa cuenta con un manejo óptimo de memoria, sin embargo aun se presentan ciertas fugas al liberar la memoria utilizada, si bien no es demasiado, es importante tenerlo en cuenta para una futura implementación.

7. Ejemplo de uso de la red social

El sistema de red social permite a los usuarios interactuar y conectar de diversas maneras. A continuación, se presentan algunos ejemplos de cómo los usuarios pueden aprovechar las funcionalidades del sistema:

Recomendación de conexiones

- El sistema internamente sugiere posibles conexiones basándose en intereses comunes entre usuarios, para simular el comportamiento de los usuarios en la web, utiliza un “Lanzamiento de moneda” con el fin de determinar el enlace de cada parte involucrada.
- Facilita la creación de nuevas relaciones y ampliando la red de contactos.

Publicación de contenido

- Los usuarios pueden publicar mensajes y otros contenidos en sus perfiles.
- Las publicaciones se pueden visualizar desde el menu interactivo de uso, al revisar amigos o buscar un perfil específico.

Búsqueda Optimizada

- El sistema permite a los usuarios realizar búsquedas eficientes para encontrar perfiles y contenidos específicos.

Visualización de intereses

- El sistema proporciona herramientas para visualizar los intereses de los usuarios, mostrando sus preferencias junto a su perfil, sin embargo se decidió por no mostrar el rango de similitud explícitamente.
- Ayuda a identificar conexiones potenciales y a comprender mejor las preferencias de la comunidad.

8. Conclusiones

El proyecto 'GrapeLoot' resultó ser una sólida aplicación de las estructuras de datos y los algoritmos en un entorno simulado de redes sociales. La implementación de estructuras de datos como grafos, tablas hash y listas enlazadas y algoritmos avanzados como el PageRank y la similitud de Jaccard, permitieron abordar problemas reales de gestión de datos y personalización de recomendaciones de forma bastante dinámica y eficiente, lo cual es un punto a favor en la implementación de la red social simulada.

En adición se consiguió implementar un sistema de manejo de archivos eficiente, dinámico y con un control de memoria destacable, lo cual indica nuevamente que se logró implementar estructuras de datos eficientes, las cuales en combinación con un manejo de archivos y directorios cauteloso, logró hacer un sistema eficiente de creación dinámica y automática de usuarios mientras se utiliza el programa. Simulando el uso de una red social real.

En resumen, este proyecto fue bastante exitoso, y si bien no fue fácil de implementar, debido a que se requiere mucha capacidad de abstracción e incluso también cierto dominio matemático para la parte de los algoritmos, fue una excelente forma de aplicar y consolidar los conocimientos adquiridos sobre estructuras de datos y algoritmos, lo que también servirá para irse adecuando a problemas de mayor complejidad a futuro.