Tecnológico de Monterrey

Proyectos Finales

Leon Daniel Vilchis Arceo A01641413

Álvaro González Martínez A01646343

Gregorio Alejandro Orozco Torres A01641967

Daniel Hernández Gutiérrez A01640706

Lorenzo Orrante Román A01641580

1 de diciembre del 2025

Programación de Estructuras de Datos y Algoritmos Fundamentales

Grupo 604

Algoritmo de Dijkstra con Animación:

Descripción del algoritmo de Dijkstra:

En esta actividad usamos el algoritmo de Dijkstra, método usado en teoría de grafos para determinar la ruta más corta de un nodo origen hacia todos los demás nodos que hay en el grafo, tomando en cuenta que ninguna de las aristas tenga un peso negativo. Este algoritmo funciona asignando progresivamente el recorrido mínimo conocido a cada uno de los nodos, iniciando en el origen y actualizando las rutas cuando se calculan caminos más cortos. En cada paso, el algoritmo siempre selecciona el nodo con menor distancia aun ni procesada, y revisa si los nodos vecinos pueden mejorar su distancia entre ellos. De esta manera, Dijkstra expanda los nodos visitados hasta que acabe de cubrir todo el grafo y permite que se haya encontrado el camino más corto entre el nodo de inicio.

También se implementó una visualización animada del funcionamiento de este algoritmo, esta animación deja observar como Dijkstra navega el grafo nodo a nodo hasta encontrar la ruta más óptima desde el origen hasta el final.

Justificación del modelo implementado:

El algoritmo de Dijkstra es el indicado en para este problema porque garantiza encontrar el camino más corto entre los nodos con una complejidad de O($nodos^2$). Dijkstra también está hecho para esta actividad ya que no cuenta con ningún tipo de pesos negativos, cosa que rompería la funcionalidad del algoritmo.

La visualización animada funciona para ilustrar claramente qué nodos se exploran y cómo se actualizan las distancias para determinar la mejor ruta. Es útil para aprender cómo funciona el algoritmo y depurar errores que pudieran ocurrir.

Complejidad temporal y espacial:

La complejidad de tiempo de este tiempo es aproximadamente de O(N^2) donde N es el número de nodos, en cada iteración se busca el nodo no visitado con el recorrido más corto, y para eso se tiene que checar los nodos que no han sido visitados. La complejidad a la hora de uso de

memoria es de O(N), ya que el programa guarda el recorrido de cada nodo, los que han sido visitados y la ruta anterior necesaria para poder armar la ruta final.

Balanceador de Carga (Round Robin):

Descripción del algoritmo Round Robin:

Es un algoritmo que se utiliza en sistemas que requieren administrar el tiempo de ejecución de diferentes procesos. Se llama Round Robin ya que lo que hace el algoritmo es rotar entre todas las acciones que debe hacer, dándoles a cada uno una cantidad de tiempo fija y sin tomar en cuenta la prioridad del proceso. Lo que busca hacer este algoritmo es darle a todos los procesos la misma cantidad de tiempo para completarse, garantizando equidad entre las tareas.

Justificación del modelo implementado:

Round Robin es efectivo en este caso ya que cumple con la distribución equitativa, es O(1) aunque tenga muchas solicitudes. Además que en la industria es la elección estándar para este tipo de situaciones.

A la hora de la visualización, Round Robin es muy útil para ver cómo se ejecuta cada paso del programa.

Complejidad del temporal y espacial:

La complejidad temporal para este balanceador de carga con Round Robin es de O(1) por cada ejecución, ya que el tiempo de ejecución siempre será algo constante. Mientras que la complejidad espacial es de O(N), N siendo el número de requests que se deben de hacer a la hora de ejecutar el algoritmo.

Algoritmo de Recomendaciones Basado en Amigos (Red Social):

Descripción del algoritmo de similitud o recomendación utilizado:

El algoritmo implementado utiliza un sistema de recomendación basado en el filtrado colaborativo que se tiene en la memoria para extraer específicamente la similitud que hay entre los datos guardados de los usuarios.
Para medir la similitud entre cada uno de los usuarios en base a sus gustos se utiliza el coeficiente de Jaccard, el cual se calcula a través de esta fórmula: similitud = (gustos_compartidos) / (gustos_totales_únicos). Resultando en un rango de 0 a 1 en donde uno significa que sus gustos son idénticos y 0 que no tienen ningún gusto en común. Después el método recomendar intereses identifica los amigos directos del usuario, extrae los gustos de cada amigo y los utiliza para evaluar el interés en base al rango de similitud calculado por el método anterior entre el amigo y el usuario. Este score se acumula y así forma un interés no poseído que al final se filtran para excluir los gustos que el usuario ya posea y se ordenan de mayor a menor en base a su score.

Justificación del modelo implementado:

El uso de grafos con NetworkX funciona como una estructura ideal para modelar las relaciones entre las amistades interpretadas como aristas y los usuarios como nodos, así calculamos eficientemente las estadísticas dentro de la red social y obtenemos los datos vecinos de amigos. Después el coeficiente de Jaccard atribuye a esto midiendo directamente la superposición de gustos entre usuarios amigos y funciona como una métrica clara para el análisis de estos conjuntos. Decidimos usar este método sobre otros útiles para el análisis de datos porque a diferencia de la similitud por el método de Pearson o del Coseno, este resultó ser más adecuado para datos categóricos.

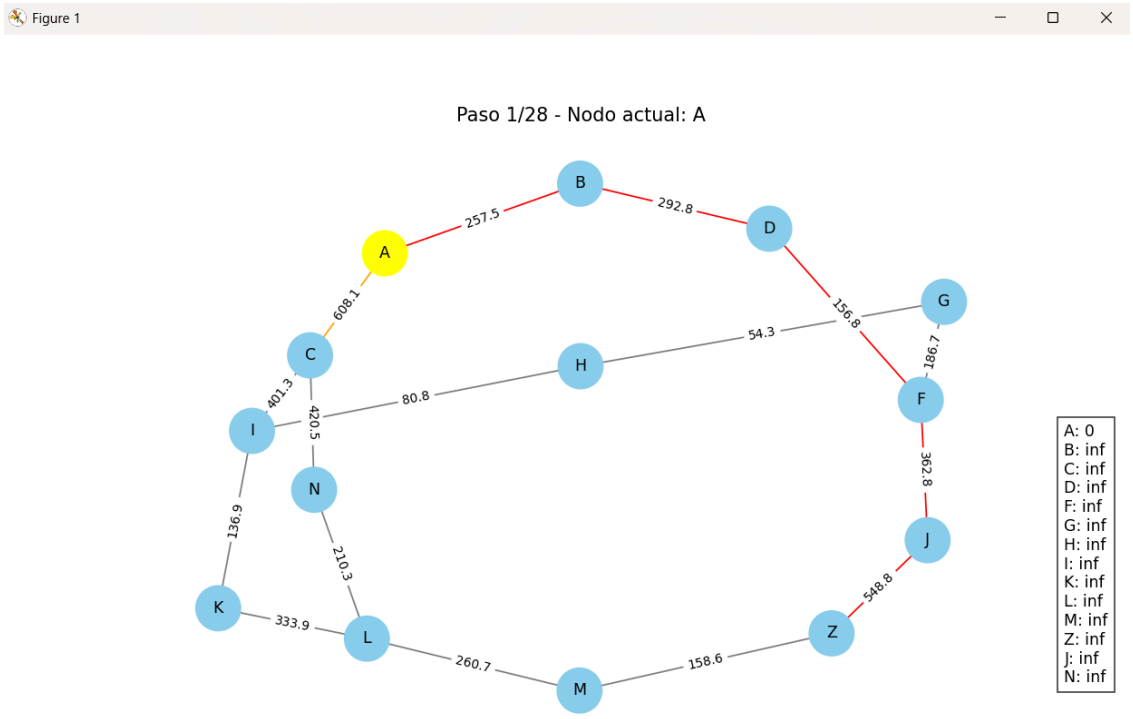
Complejidad temporal y espacial:

La complejidad temporal para el metodo de recomendacion es de O(D*K*log(D)) donde D es el número máximo de amigos, K es el máximo de gustos que se tienen por usuario y se utilizan cada vez que se quieren saber las recomendaciones por usuario. El tiempo de ejecución no es igual a O(1) ya que depende de la cantidad de amigos y su número de gustos. La complejidad espacial se centra en el almacenamiento de cómo está estructurada la red y los datos de cada usuario, por lo que sería una complejidad espacial de O(N*K+M) donde N son los usuarios y M

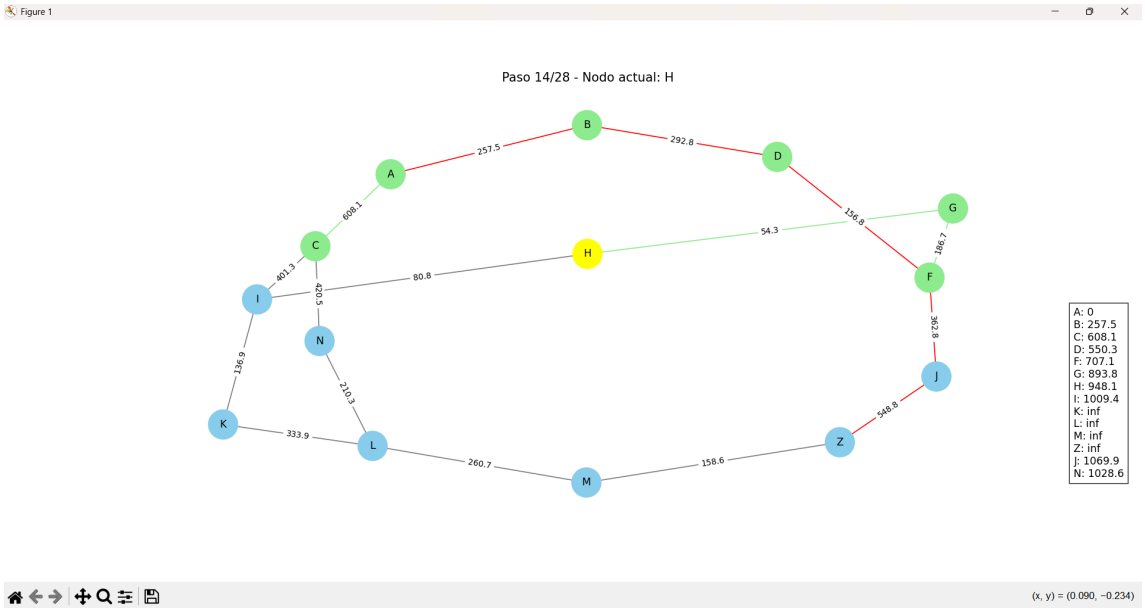
las amistades. El sistema primero almacena la lista de usuarios y sus gustos O(N*K) para después sumar la lista del grafo de amistades M.

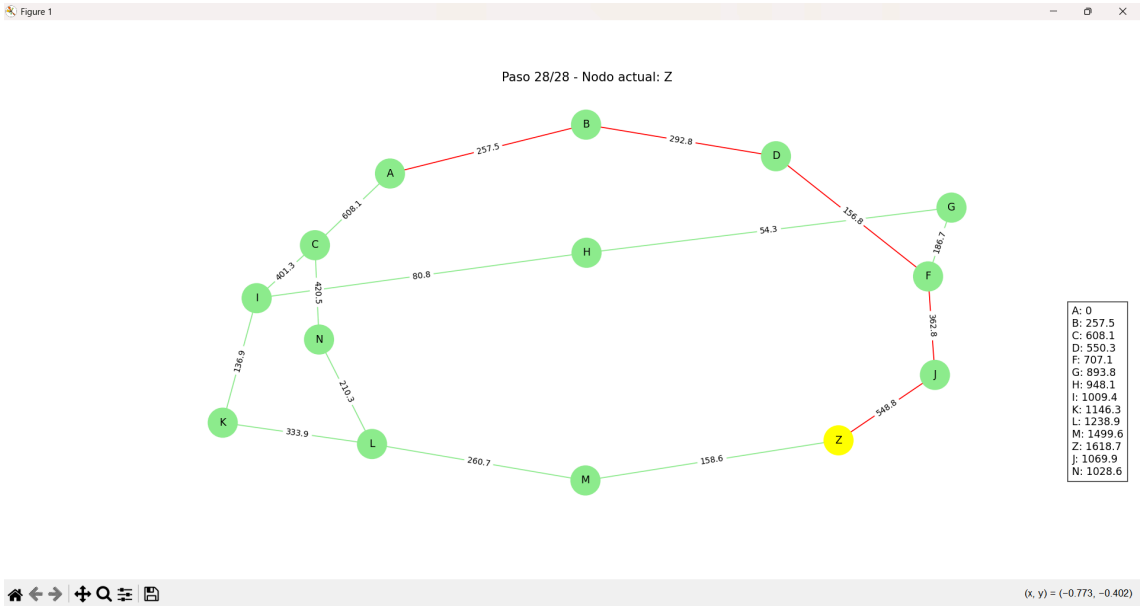
Resultados observados y capturas de ejecución:

Algoritmo de Dijkstra con Animación:



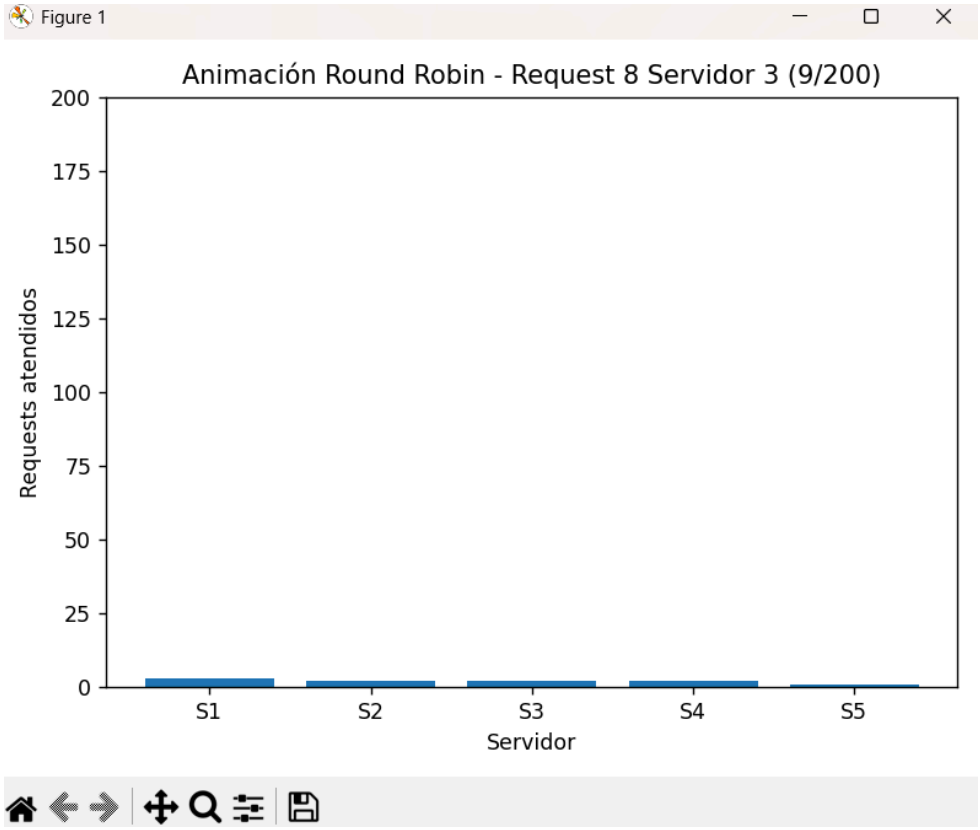
O





Podemos observar como el algoritmo calcula la distancia más corta desde el nodo A, que es el origen, hacia todos los demás nodos en el grafo. Para cada nodo objetivo, se consigue la distancia total y la secuencia de nodos que generan el camino más corto. La animación enseña como Dijkstra visita todos los nodos de forma gradual, iniciando en el origen y abriéndose camino hacia los nodos vecinos, eligiendo con las distancias más pequeñas encontradas.

Balanceador de Carga (Round Robin):



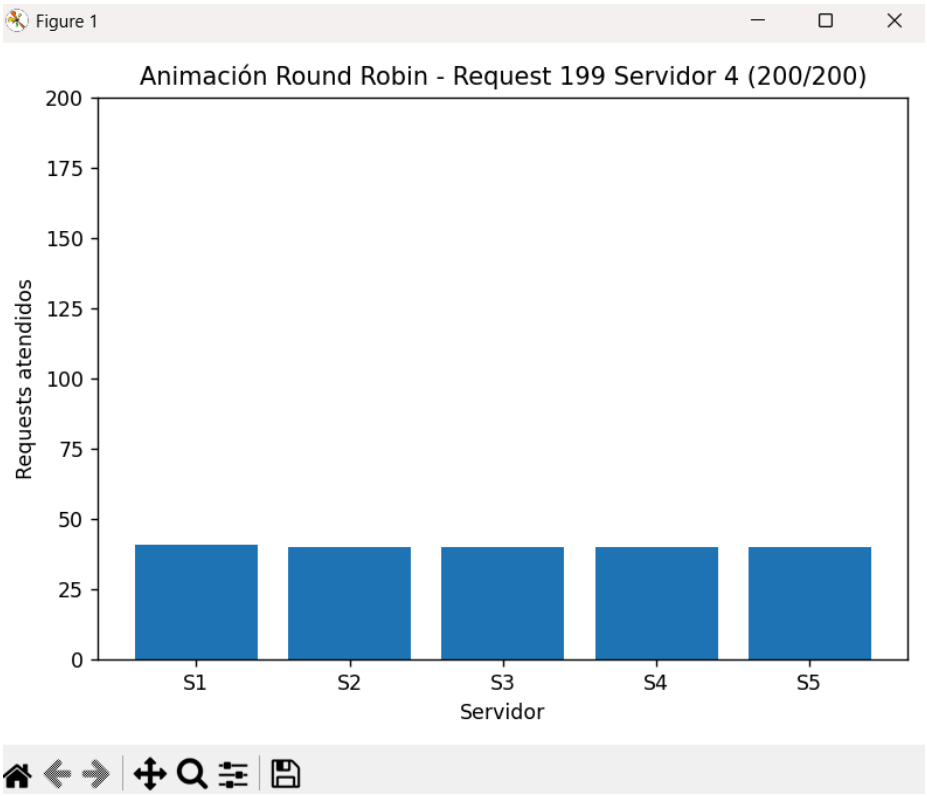


Figure 1
Animación Round Robin - Request 199 Servidor 4 (200/200)
Requests atendidos
200
175
150
125
100
75
50
25
0
S1
S2
S3
S4
S5
Servidor

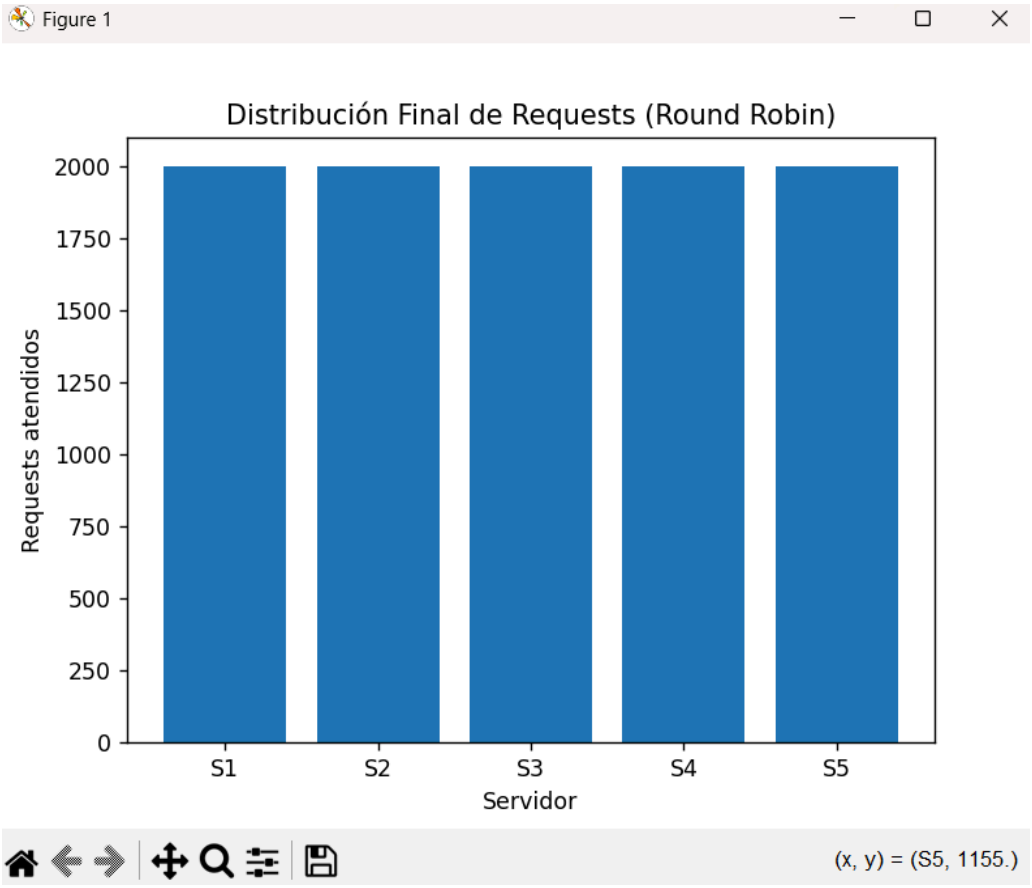


Figure 1
Distribución Final de Requests (Round Robin)
Requests atendidos
2000
1750
1500
1250
1000
750
500
250
0
S1
S2
S3
S4
S5
Servidor
(x, y) = (S5, 1155.)

Se puede apreciar cómo el algoritmo se encarga de asignar las request de forma uniforme entre todos los servidores, distribuyendo todo en un patrón circular que se repite. La gráfica de barras muestra como la altura para cada barra es casi igual, demostrando la distribución equitativa

Algoritmo de Recomendaciones Basado en Amigos (Red Social):

```
================================================================================
SISTEMA DE RECOMENDACIONES BASADO EN AMIGOS
================================================================================

ESTADÍSTICAS DE LA RED SOCIAL
   • Número de usuarios: 8
   • Número de amistades: 12
   • Densidad de la red: 0.429
   • Componentes conexas: 1
   • Grado promedio: 3.00

--------------------------------------------------------------------------------
```

```
USUARIO: Ana
Gustos actuales: fitness, música, tecnología, viajes
Amigos (3): Carlos, María, Pedro

Similitud con amigos:
      • Carlos: 33.33%
      • María: 33.33%
      • Pedro: 33.33%

RECOMENDACIONES:
      1. deportes (score: 0.667)
      2. cocina (score: 0.333)
      3. videojuegos (score: 0.333)
      4. programación (score: 0.333)
      5. fotografía (score: 0.333)

--------------------------------------------------------------------------------
```

```
USUARIO: Carlos
Gustos actuales: cocina, deporte, fitness, música
Amigos (3): Ana, María, Diego

Similitud con amigos:
      • Diego: 100.00%
      • Ana: 33.33%
      • María: 14.29%

RECOMENDACIONES:
      1. tecnología (score: 0.476)
      2. viajes (score: 0.333)
      3. videojuegos (score: 0.143)
      4. programación (score: 0.143)

--------------------------------------------------------------------------------
```

```
USUARIO: María
Gustos actuales: música, programación, tecnología, videojuegos
Amigos (4): Ana, Carlos, Juan, Laura

Similitud con amigos:
      • Juan: 100.00%
      • Ana: 33.33%
      • Laura: 33.33%
      • Carlos: 14.29%

RECOMENDACIONES:
      1. fitness (score: 0.476)
      2. viajes (score: 0.333)
      3. arte (score: 0.333)
      4. fotografía (score: 0.333)
      5. deportes (score: 0.143)

--------------------------------------------------------------------------------
```

```
USUARIO: Pedro
Gustos actuales: deportes, fitness, fotografía, viajes
Amigos (3): Ana, Sofía, Diego

Similitud con amigos:
      •Sofía: 60.00%
      • Ana: 33.33%
      • Diego: 33.33%

RECOMENDACIONES:
      1. cocina (score: 0.933)
      2. música (score: 0.667)
      3. tecnología (score: 0.333)

--------------------------------------------------------------------------------
```

```
USUARIO: Laura
Gustos actuales: arte, fotografía, música, tecnología
Amigos (3): María, Sofía, Juan

Similitud con amigos:
      • María: 33.33%
      • Juan: 33.33%
      • Sofía: 14.29%

RECOMENDACIONES:
      1. videojuegos (score: 0.667)
      2. programación (score: 0.667)
      3. viajes (score: 0.143)
      4. fitness (score: 0.143)
      5. cocina (score: 0.143)

--------------------------------------------------------------------------------
USUARIO: Juan
Gustos actuales: música, programación, tecnología, videojuegos
Amigos (2): María, Laura

Similitud con amigos:
      • María: 100.00%
      • Laura: 33.33%

RECOMENDACIONES:
      1. arte (score: 0.333)
      2. fotografía (score: 0.333)
```

```
USUARIO: Sofía
Gustos actuales: cocina, fitness, fotografía, viajes
Amigos (3): Pedro, Laura, Diego

Similitud con amigos:
      • Pedro: 60.00%
      • Diego: 33.33%
      • Laura: 14.29%

RECOMENDACIONES:
      1. deportes (score: 0.933)
      2. música (score: 0.476)
      3. arte (score: 0.143)
      4. tecnología (score: 0.143)

--------------------------------------------------------------------------------
```

```
USUARIO: Diego
Gustos actuales: cocina, deportes, fitness, música
Amigos (3): Carlos, Pedro, Sofía

Similitud con amigos:
      • Carlos: 100.00%
      • Pedro: 33.33%
      • Sofía: 33.33%

RECOMENDACIONES:
      1. viajes (score: 0.667)
      2. fotografía (score: 0.667)
```

```
================================================================================
EJEMPLO DETALLADO: Ana
================================================================================

Usuario: Ana
Gustos: {'música', 'viajes', 'fitness', 'tecnología'}

Amigos: ['Carlos', 'María', 'Pedro']

Análisis de similitud con amigos:
  • Carlos:
    - Similitud: 33.33%
    - Gustos compartidos: {'música', 'fitness'}
    - Gustos únicos del amigo: {'deportes', 'cocina'}
  • María:
    - Similitud: 33.33%
    - Gustos compartidos: {'música', 'tecnología'}
    - Gustos únicos del amigo: {'videojuegos', 'programación'}
  • Pedro:
    - Similitud: 33.33%
    - Gustos compartidos: {'viajes', 'fitness'}
    - Gustos únicos del amigo: {'deportes', 'fotografía'}

Recomendaciones generadas:
  • deportes: score 0.667
    ← Recomendado por: Carlos (sim: 33.33%), Pedro (sim: 33.33%)
  • cocina: score 0.333
    ← Recomendado por: Carlos (sim: 33.33%)
  • videojuegos: score 0.333
    ← Recomendado por: María (sim: 33.33%)
  • programación: score 0.333
    ← Recomendado por: María (sim: 33.33%)
  • fotografía: score 0.333
    ← Recomendado por: Pedro (sim: 33.33%)
```

En la ejecución se demuestra el funcionamiento del sistema de recomendaciones basado en el filtrado colaborativo de los amigos. Se aprecian los resultados que calculamos con el coeficiente de Jaccard para la similitud de gustos entre pares de usuarios y se observa la ponderación de los scores entre gustos y como se acumulan para darle mayor peso a los gustos populares entre amigos con la mayor similitud al usuario.

Referencias APA:

GeeksforGeeks. (2025, 25 agosto). Round Robin Scheduling in Operating System. https://www.geeksforgeeks.org/operating-systems/round-robin-scheduling-in-operating-system/

Lagares, J. (2022, 27 agosto). How does Dijkstra's Algorithm work? Easy explanation in Less than 5 Minutes. *Medium*. https://medium.com/codex/how-does-dijkstras-algorithm-work-easy-explanation-in-less-than-5-minutes-e27f46795c18