ANÁLISIS RETO 4 DE ESTRUCTURA DE DATOS Y ALGORITMOS

Integrantes:

- Estudiante 1: Jerónimo López / 202320969 / j.lopezm234@uniandes.edu.co
- Estudiante 2: Julián David Ramos González /202414411/ jd.ramosg1@uniandes.edu.co
- Estudiante 3: Juan Esteban Piñeros Barrera / 202412232 / Je.pineros@uniandes.edu.co

Requerimientos:

- Requerimiento 1: Grupal
- Requerimiento 2: Grupal
- Requerimiento 3: Julián Ramos (Estudiante 2)
- Requerimiento 4: Jerónimo López (Estudiante 1)
- Requerimiento 5: Juan Esteban Piñeros (Estudiante 3)
- Requerimiento 6: Grupal
- Requerimiento 7: Grupal
- Requerimiento 8: Grupal

Requerimiento 1:

Análisis: Este requerimiento utiliza un DFS para encontrar un camino entre dos usuarios. Además, recupera la información de cada usuario en el camino y se recorre todos los nodos y las aristas conectadas en el peor caso: O(V + E). Recuperar información de cada nodo tiene complejidad O(1) por nodo.

Datos de prueba:
user_id_a = 8236
user_id_b = 7918
Gráfica:

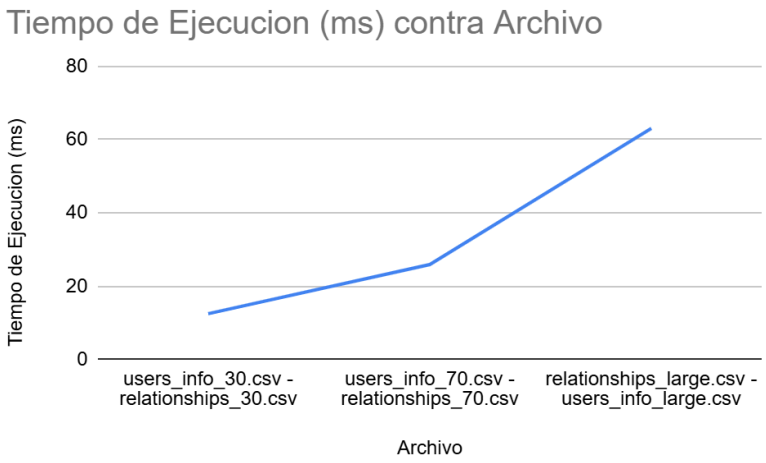

Tabla de comparación:

Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	12.4867
users_info_70.csv - relationships_70.csv	25.89358991
relationships_large.csv - users_info_large	63.09890707

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo sea O(V + E).

Requerimiento 2:

Busca el camino más corto entre dos usuarios de tipo "basic" usando BFS. Se verifica el catálogo que es O(1) por usuario (2 usuarios). Recorre los nodos y aristas conectados en el peor caso: O(V + E). Complejidad Total: O(V + E).

Datos de prueba:
user_id_a = 4242
user_id_b = 5174

Gráfica:

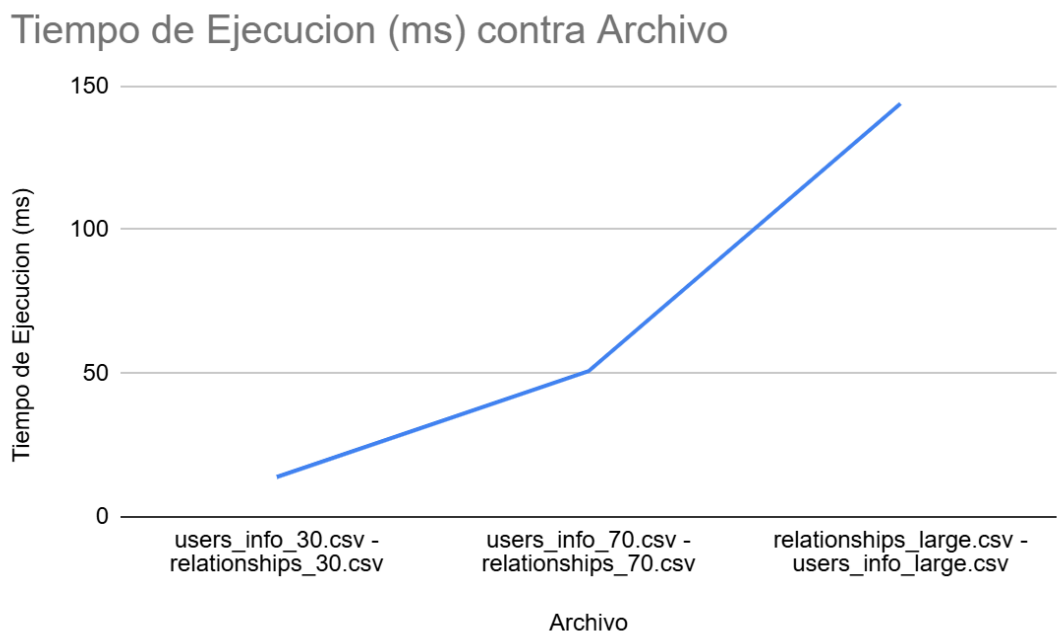

Tabla de comparación:

req 2	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	13.8883
users_info_70.csv - relationships_70.csv	50.7633
relationships_large.csv - users_info_large	143.8746

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo sea O(V + E).

Requerimiento 3:

Identifica al amigo de un usuario que tiene más seguidores. Obtenemos los amigos (graph.adjacents) y se accede a la lista de adyacencia: O(1). Para cada amigo, calcular el in_degree tiene complejidad O(V) en el peor caso. Comparación lineal entre los amigos: O(A), donde A es el número de amigos. Complejidad Total: O(A * V).

Datos de prueba:

user_id_a = 4242

Gráfica:

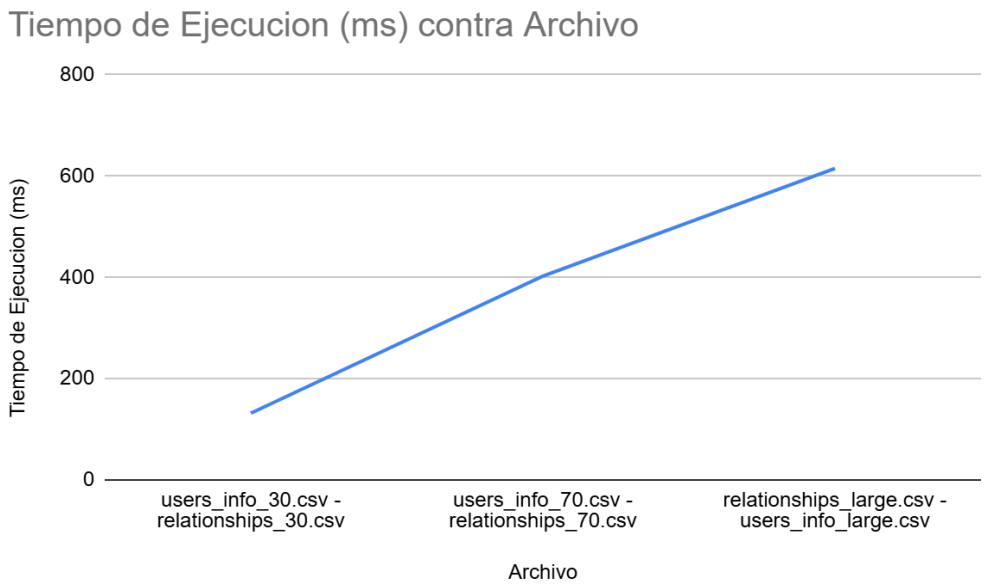

Tabla de comparación:

req 3	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	131.837
users_info_70.csv - relationships_70.csv	402.6739
relationships_large.csv - users_info_large	615.4294

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo sea O(A * V).

Requerimiento 4:

Encuentra amigos en común entre dos usuarios. Obtenemos los amigos para cada usuario que serían O(A1) y O(A2) para los dos usuarios. Se intersecta la listas de amigos donde sería comparación lineal entre las listas lo que daría una complejidad total O(A1 + A2).

Datos de prueba:

user_id_a = 4242
user_id_b = 5174

Gráfica:

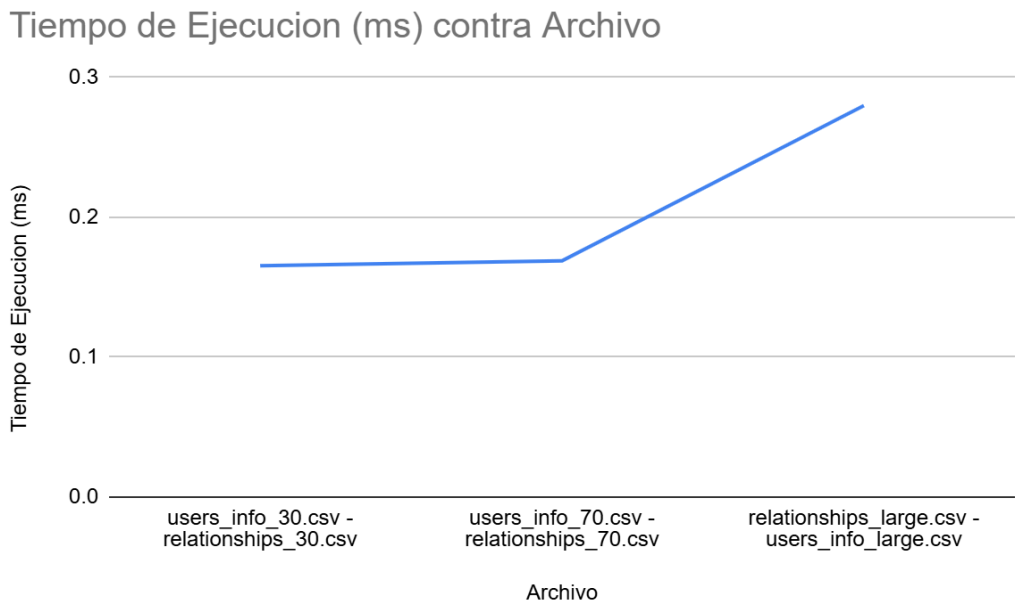

Tabla de comparación:

req 4	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	0.1652999967
users_info_70.csv - relationships_70.csv	0.1687999964
relationships_large.csv - users_info_large	0.2798000127

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo sea $O(A1 + A2)$.

Requerimiento 5:

Encuentra los N amigos que siguen a más usuarios y son amigos mutuos Se itera sobre la lista de adyacencia del usuario lo que sería O(A), donde A es el número de amigos. Para cada amigo, iterar sobre su lista de adyacencia: O(A * A'), donde A' es el promedio de amigos de cada amigo. Y, se usa merge sort lo que es O(N * log(N)). Complejidad Total: O(A * A' + N * log(N)).

Datos de prueba:

user_id_a = 4242
N = 90
Gráfica:

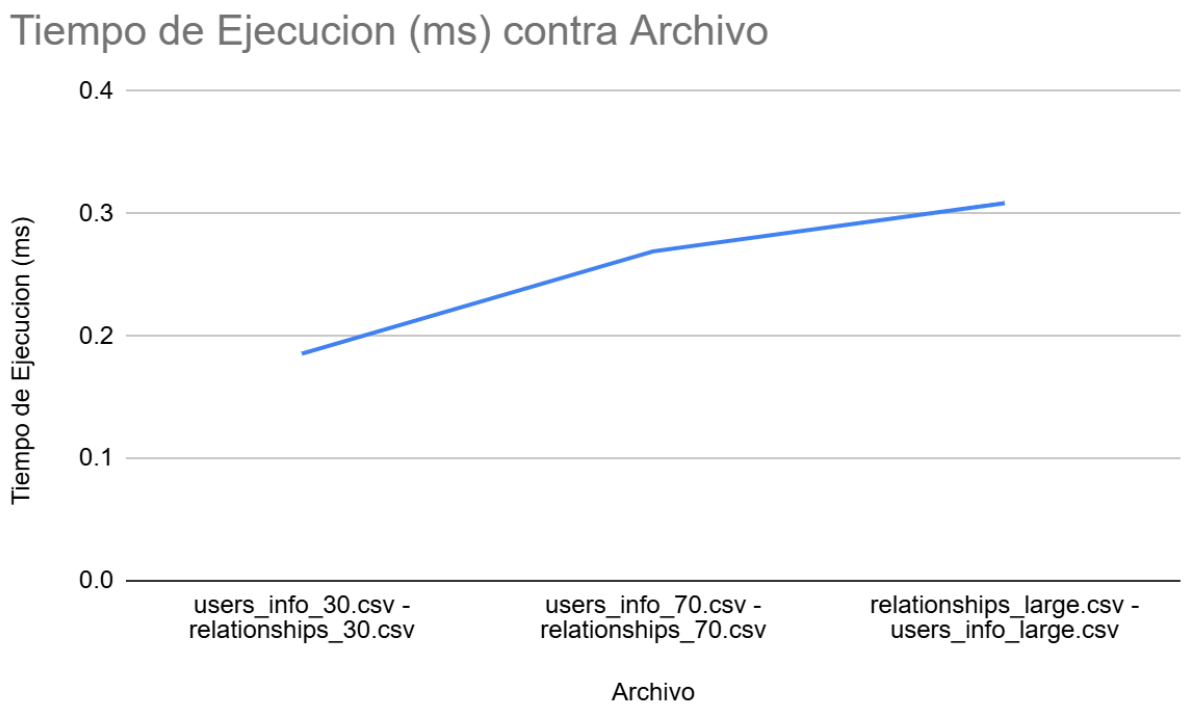

Tabla de comparación:

req 5	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	0.1854999959
users_info_70.csv - relationships_70.csv	0.2692000121
relationships_large.csv - users_info_large	0.3085999936

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo se resume en O(A * A')

Requerimiento 6:

Encuentra los usuarios más populares y construye un árbol de relaciones. Se itera sobre todos los vértices para calcular popularidad lo que sería O(V). Se ordenan de los usuarios populares lo que sería O(N * log(N)). Construcción del árbol con BFS lo que sería O(V + E). Complejidad Total: O(V + N * log(N) + V + E).

Datos de prueba:

N = 30

Gráfico:

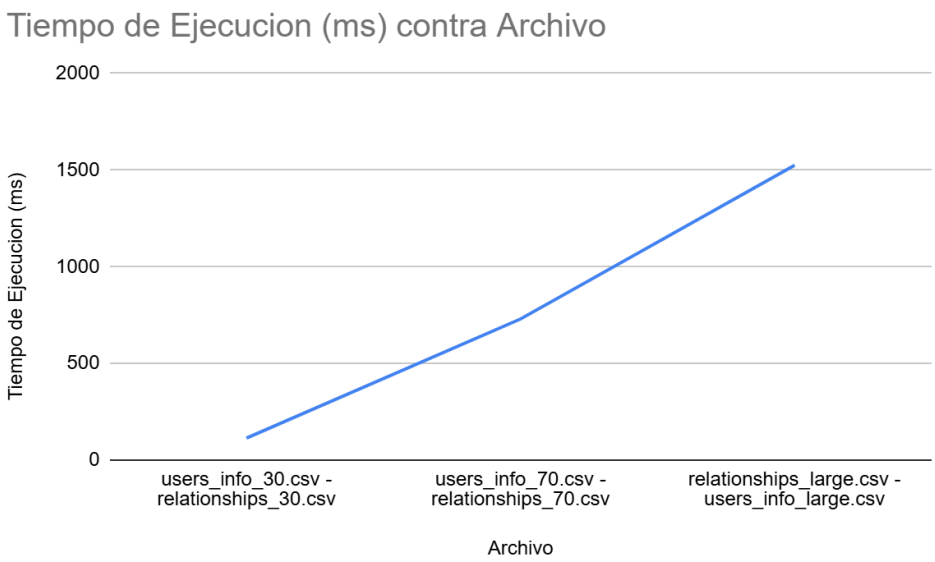

Tabla de comparación:

req 6	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	113.8698
users_info_70.csv - relationships_70.csv	729.9727
relationships_large.csv - users_info_large	1524.5274

Análisis final:

Según las gráficas y el análisis inicial es muy probable que el algoritmo se resume en O(V + N)

Requerimiento 7:

Encuentra usuarios con intereses similares basados en hobbies. Obtenemos los amigos directos lo que sería complejidad: O(A). Se comparan entre hobbies del usuario y los amigos: O(A * H), donde H es el número de hobbies. Y, se verifica para cada amigo: O(A * A'), donde A' es el promedio de amigos. Complejidad Total: O(A * H + A * A').

Datos de prueba:

user_id = 4242
hobbie = tango
Gráfica:

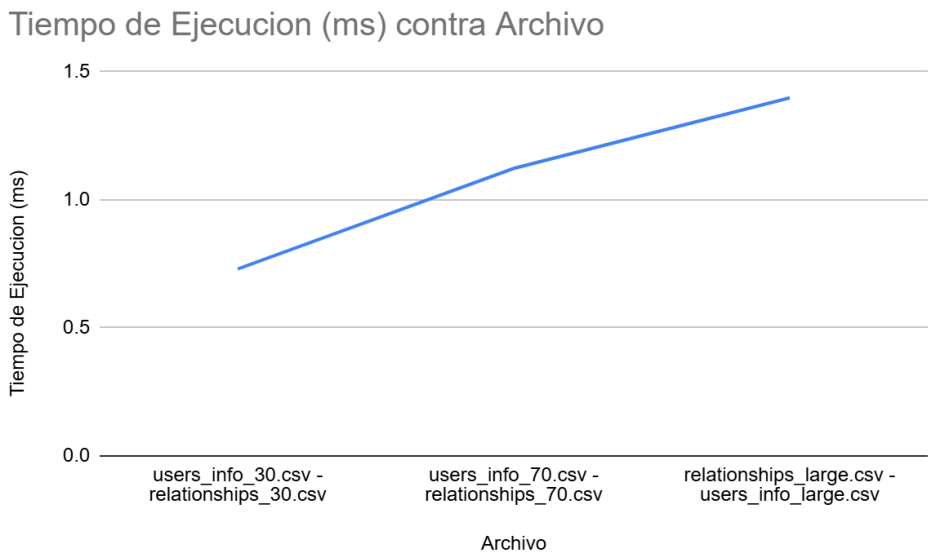

Tabla de comparación:

req 7	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	0.730099991
users_info_70.csv - relationships_70.csv	1.122899994
relationships_large.csv - users_info_large	1.39819999

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo se resume en O(A * H)

Requerimiento 8:

Grafica usuarios dentro de un radio específico utilizando Haversine y Folium. Se itera sobre todos los vértices del grafo lo que sería complejidad: O(V). Se hace cálculo de distancias (Haversine), y para cada vértice: O(V). Creamos marcadores en el mapa de complejidad: O(V). Complejidad Total: O(V).

Datos de prueba:

latitud del centro: 33.8989

longitud del centro: -118.2351

radio en kilómetros: 10000000

Gráfica:

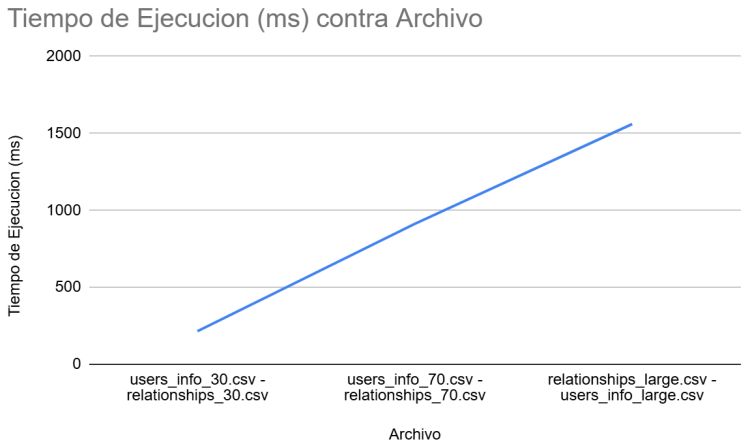

Tabla de comparación:

req 8	
Archivo	Tiempo de Ejecucion (ms)
users_info_30.csv - relationships_30.csv	215.1656
users_info_70.csv - relationships_70.csv	912.3081
relationships_large.csv - users_info_large	1560.6817

Análisis final:
Según las gráficas y el análisis inicial es muy probable que el algoritmo se resume en $O(V)$.