

Análisis de resultados

Integrantes:

- Daniel Cortes, d.cortesl@uniandes.edu.co, 202316263
- Salomon Fullea, s.fullea@uniandes.edu.co, 202225323

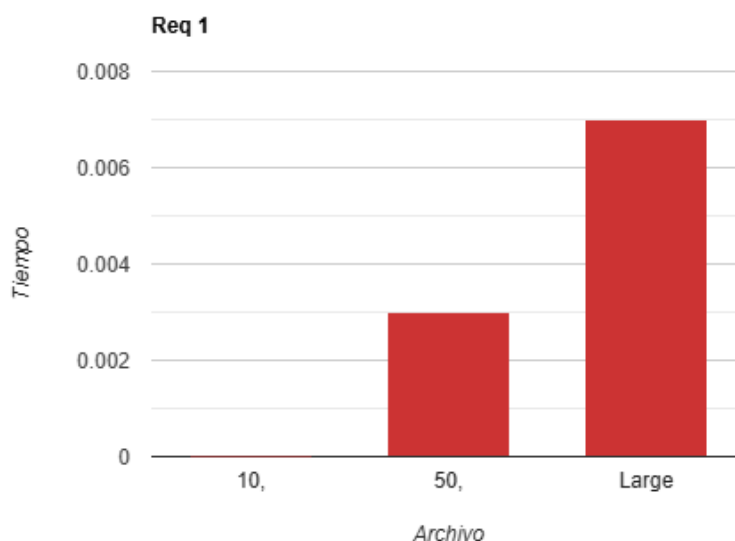
Requerimientos Individuales:

- Requerimiento 3: Daniel Cortes
- Requerimiento 4: Salomon Fullea

Requerimiento 1:

-Las líneas “vertices = catalog[“vertices”][“table”][“elements”]”, “user_data = catalog[“information”][“table”][“elements”]”, “if start_id not in vertices or end_id not in vertices” son búsquedas en diccionarios por lo tanto se tiene una complejidad de $O(1)$. Después se tiene una búsqueda por BSF la cual tiene una complejidad $O(V+E)$ en el peor de los casos por último se recorren todos los caminos del grafo lo cual nos da una complejidad de $O(P)$, simplificando todo nos da finalmente una complejidad de $O(V+E)$.

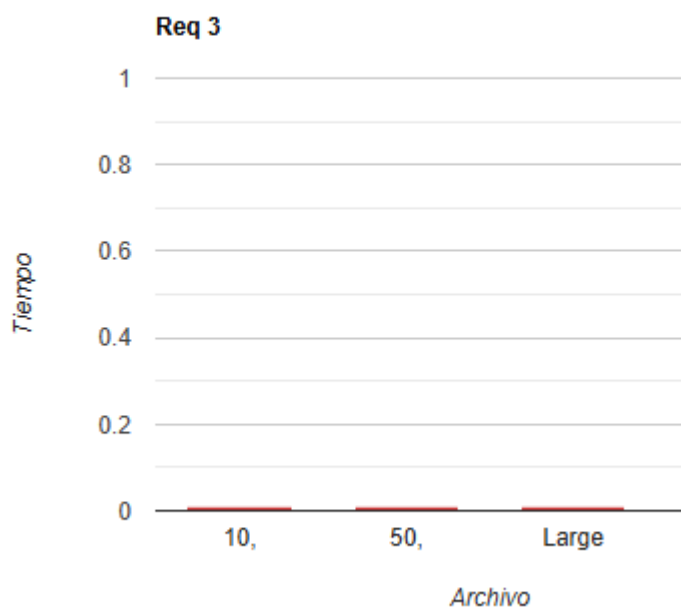
| Req_1 | | |
|---------|-----------------|-----------|
| Archivo | Datos | Tiempo(S) |
| 10 | 9830.0 - 7073.0 | 0.0 |
| 50 | 9830.0 - 7073.0 | 0.003001 |
| Large | 9830.0 - 7073.0 | 0.007006 |



Requerimiento 3:

-Las líneas “vertices = catalog[\"vertices\"]['table']['elements]”, “if start_id not in vertices”, “friends = vertices.get(start_id, [])” son operaciones de diccionarios y listas las cuales tienen complejidad de $O(1)$. Después tenemos el ciclo “for friend_id in friends” el cual en el peor de los casos tiene una complejidad de $O(v)$ donde v son los vertices que se iteran, por último, tenemos la operación len la cual tiene complejidad $O(1)$ sin embargo la complejidad total del algoritmo sería $O(v)$

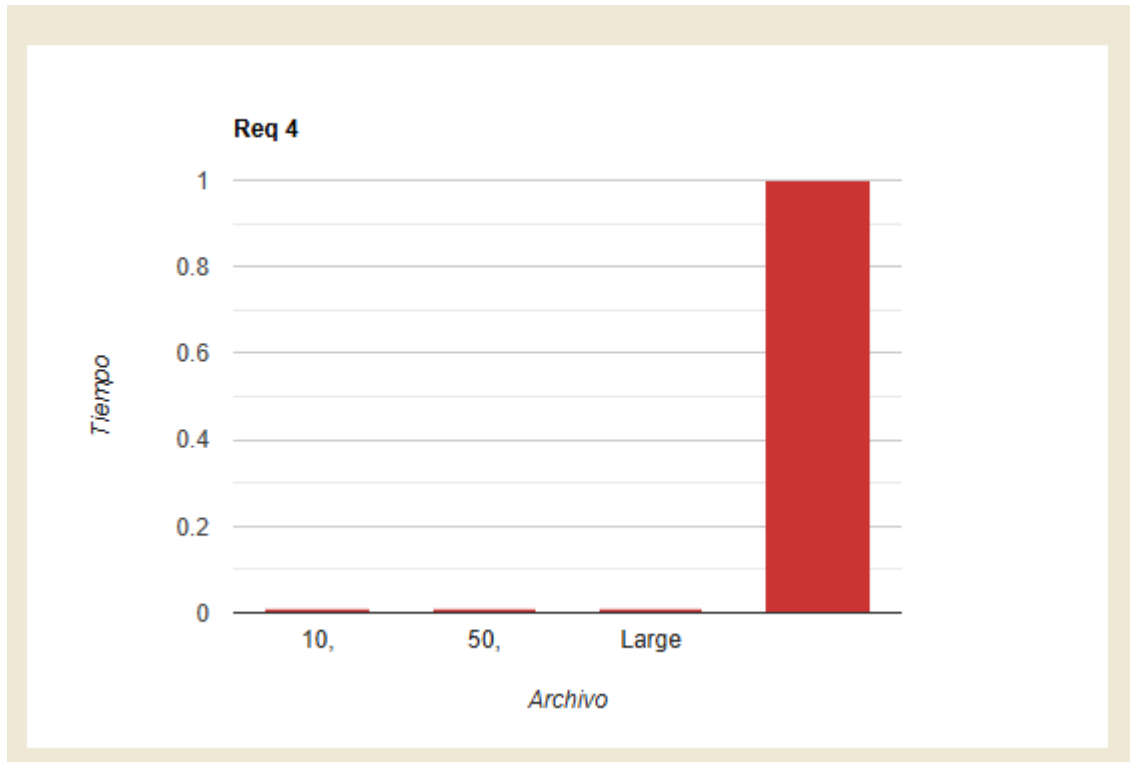
| Req_3 | | |
|---------|--------|-----------|
| Archivo | Datos | Tiempo(S) |
| 10 | 9830.0 | 0.0 |
| 50 | 9830.0 | 0.0 |
| Large | 9830.0 | 0.0 |



Requerimiento 4:

-Las líneas “catalog[\"vertices\"]['table']['elements]” y “catalog[\"information\"]['table']['elements]” tienen complejidad $O(1)$, después al hacer “Afriends = relationshipgraph[cuentaA]” y “Bfriends = relationshipgraph[cuentaB]” también se tiene una complejidad de $O(1)$ después en el ciclo “for amigo in Afriends:” en el peor de los casos se tiene una complejidad de $O(n)$, dentro de este ciclo se realizan varias verificaciones con if las cuales tienen complejidad de $O(m)$ por ende la complejidad final es $O(n * m)$

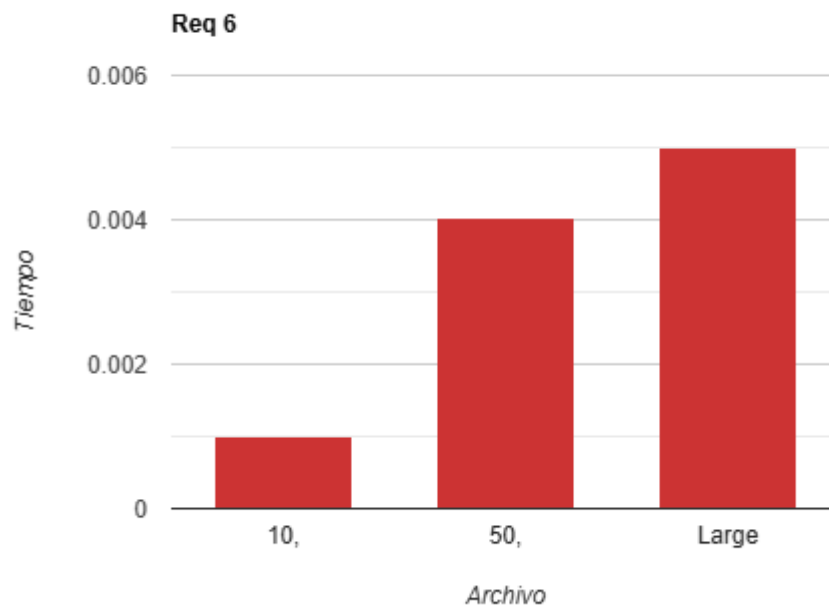
| Req_4 | | |
|---------|--------|-----------|
| Archivo | Datos | Tiempo(S) |
| 10 | 3 y 54 | 0.0 |
| 50 | 3 y 54 | 0.0 |
| Large | 3 y 54 | 0.0 |



Requerimiento 6:

-Primeramente, tenemos una comparación en la línea “if $N < 2$ ” la cual tiene complejidad de $O(1)$ igualmente que las líneas de operaciones básicas de diccionarios y listas, después tenemos el ciclo “for user_id, followed_users in vertices.items():” el cual en el peor de los casos la complejidad es de $O(E)$ siendo “E” la cantidad de aristas, después tenemos la línea sorted la cual tiene una complejidad de $O(V \log V)$, después la línea “popular_users = sorted_users[:N]” tiene una complejidad de $O(N)$ y por ultimo tenemos la llamada a la función “check_if_tree_exists(catalog, popular_user_ids)” la cual es un recorrido de BFS el cual tiene complejidad de $O(V+E)$ por lo tanto la complejidad total es $O(E + V \log V + N)$

| Req_6 | | |
|---------|-------|-----------|
| Archivo | Datos | Tiempo(S) |
| 10 | 9 | 0.001012 |
| 50 | 9 | 0.004014 |
| Large | 9 | 0.005012 |



Requerimiento 7:

-Las líneas “`catalog["vertices"]["table"]["elements"]`” y “`catalog["information"]["table"]["elements"]`”, la creación de `visited` y la creación de `queue` tienen complejidad de $O(1)$, después se tiene una búsqueda de tipo BFS la cual como en los otros requerimientos tiene una complejidad de $O(V + E)$. En el bucle “for neighbor in neighbors” se tienen varias operaciones sin embargo la mayoría tienen complejidad de $O(1)$, la operación la cual cambia la complejidad es la de `.split()` y calcular la intersección la cual tiene una complejidad de $O(h+k)$ por ende la complejidad total de esta función es de $O(V + E(h + k))$

| Req_7 | | |
|---------|---|-----------------|
| Archivo | Datos | Tiempo(S) |
| 10 | 7488.0 - ['parapente', 'voleibol', 'rafting'] | <u>0.021208</u> |
| 50 | 7488.0 - ['parapente', 'voleibol', 'rafting'] | 0.085139 |
| Large | 7488.0 - ['parapente', 'voleibol', 'rafting'] | 0.136818 |

