

OBSERVACIONES DEL RETO 04

David Álvarez Cod 202410841
Daniel Matabajoy Cod 202326861
Julian Sanchez Cod 202412119

Requerimiento	Responsable	Código	Correo
Requerimiento 1	Grupal(David)	202410841	df.alvarezr1@uniandes.edu.co
Requerimiento 2	Grupal(David)	202410841	df.alvarezr1@uniandes.edu.co
Requerimiento 3	David Alvarez	202410841	df.alvarezr1@uniandes.edu.co
Requerimiento 4	Daniel Mata	202326861	d.matabajoy@uniandes.edu.c o
Requerimiento 5	Julian Sanchez	202412119	j.sanchezf@uniandes.edu.co
Requerimiento 6	Grupal(Daniel)	202410841	d.matabajoy@uniandes.edu.c o
Requerimiento 7	Grupal(Julian)	202412119	j.sanchezf@uniandes.edu.co
Bono			

Requerimiento	Complejidad Temporal n=vértices, m=aristas
req_1	$O(n + m)$
req_2	$O(n + m)$
req_3	$O(n+m)$
req_4	$O(n)$
req_5	$O(n + m \log m)$
req_6	$O(n + m \log m)$
req_7	$O(n + m)$
req_8	

Ambientes de pruebas

	Máquina 1	Máquina 2	Máquina 3
Procesadores	12th Gen Intel(R) Core(TM) i5-12500H	AMD Ryzen 5 5500u	12th Gen Intel(R) Core(TM) i5-12450H
Memoria RAM (GB)	8 gb	6gb	8gb
Sistema Operativo	Windows	Windows	Windows

Tipos de algoritmos usados:

- Grafos
- Queue
- Stack
- Index Priority Queue

- bfs
- Dijkstra

Máquina 1

Resultados requerimientos

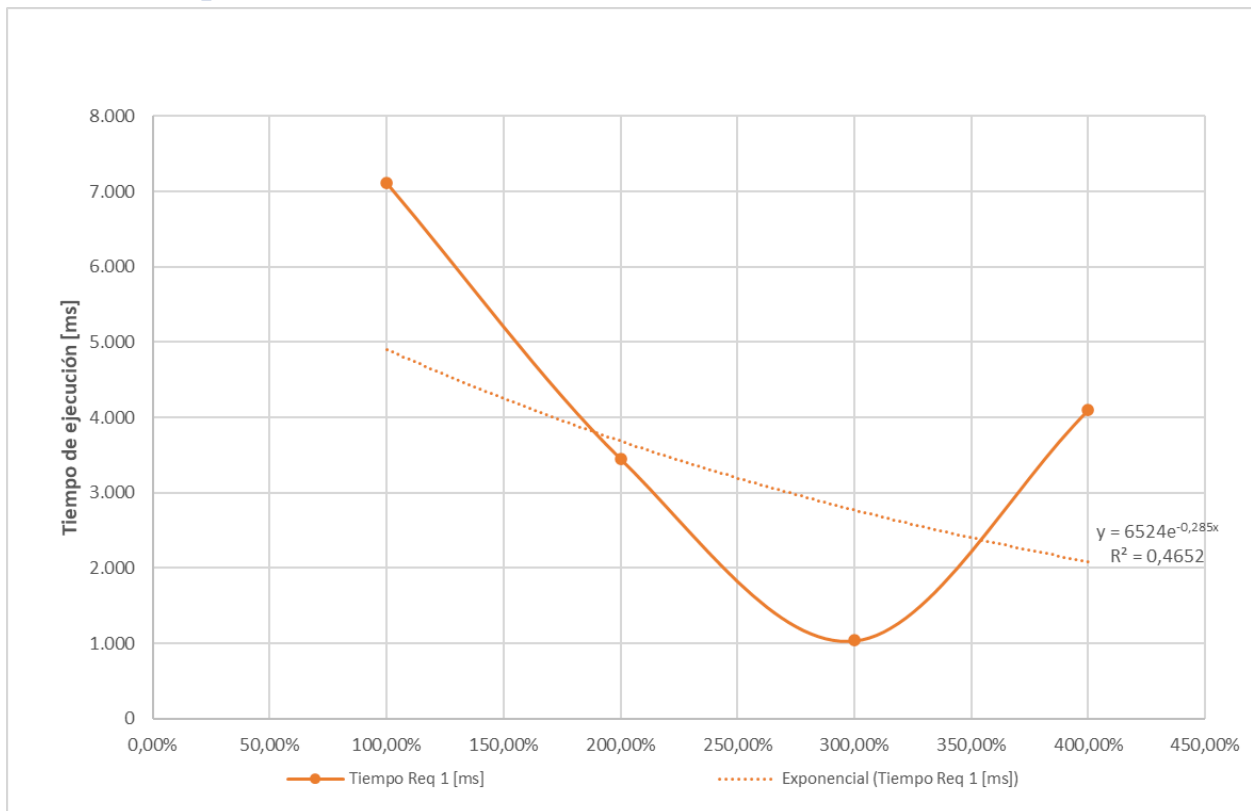
Carga de datos :

Carga de datos	Tiempo[ms]:19.174
----------------	-------------------

Req 1 :

Req 1(start_id, end_id)	Tiempo Req 1 [ms]
804, 639	7.111
6350, 45	3.451
5521, 1927	1.037
3643,57	4.100

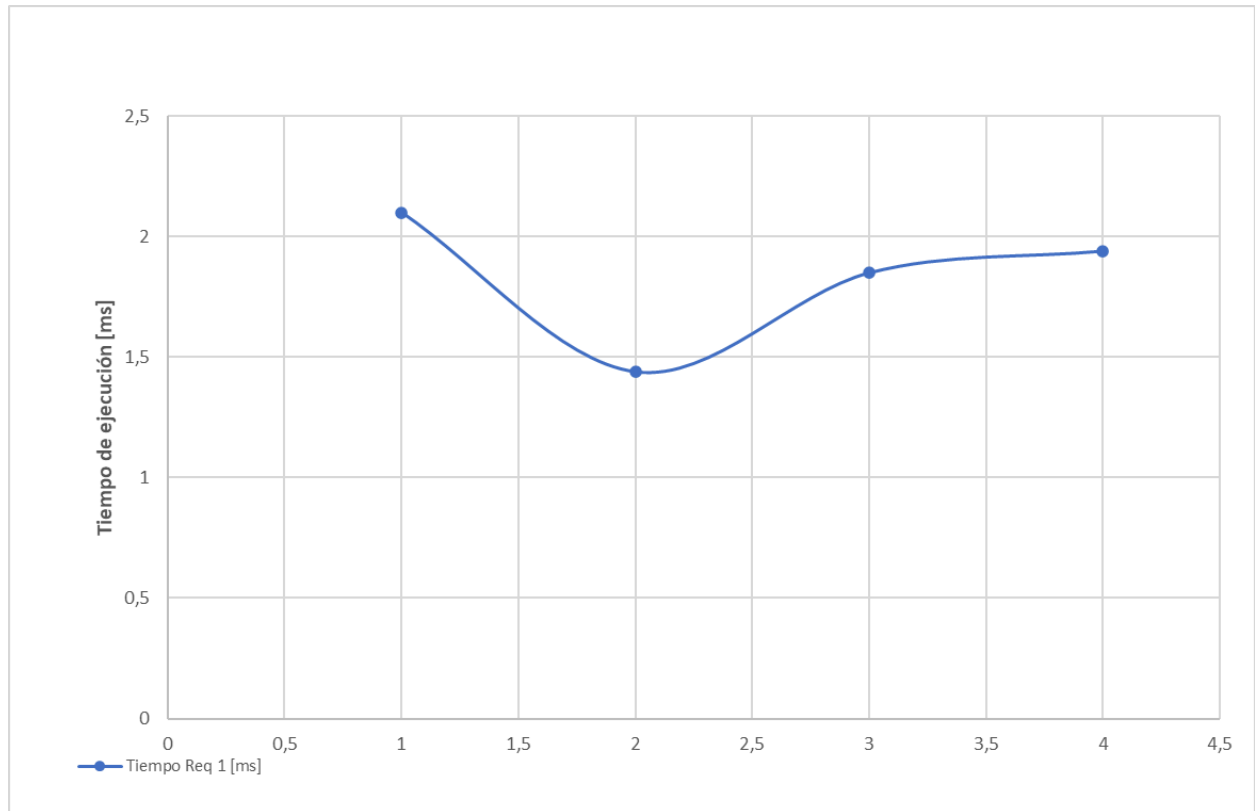
Gráficas Req 1 :

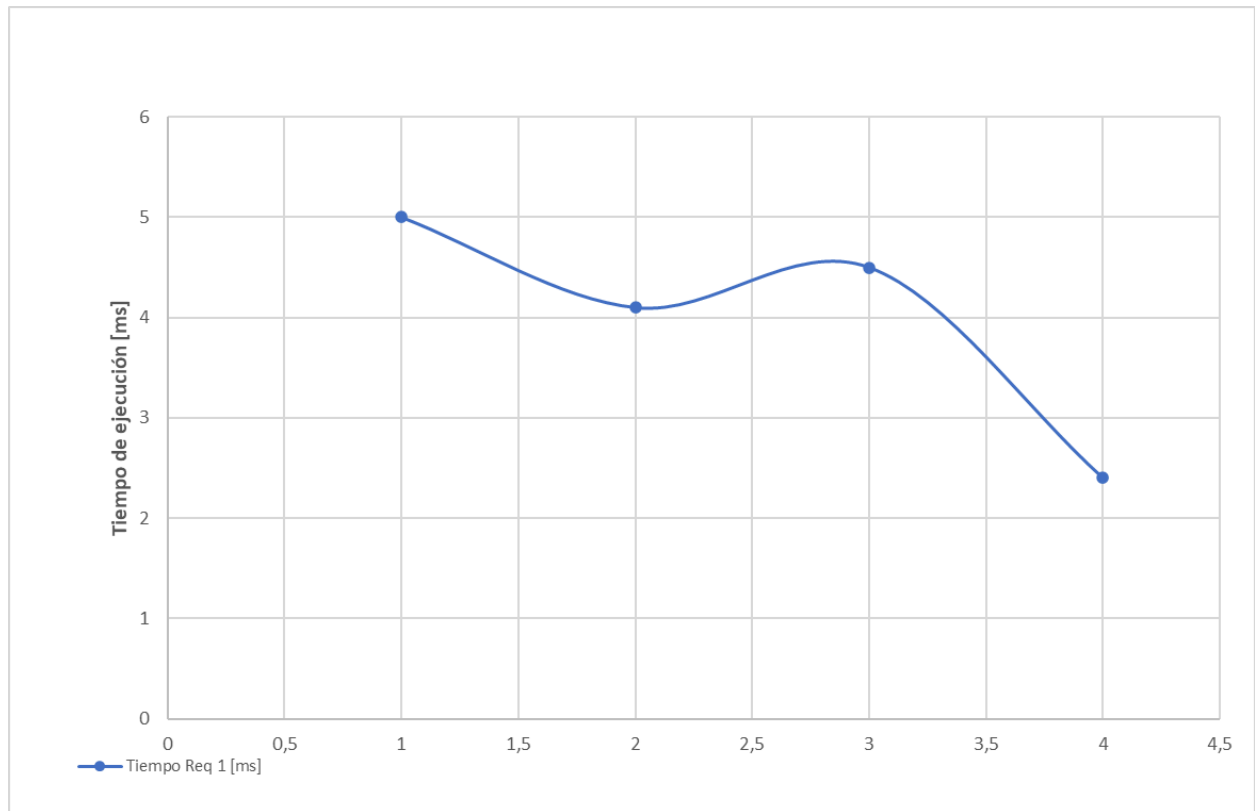


Req 2 :

Req 2(start_id, end_id)"basic"	Tiempo Req 2 [ms]	Req 3(user_id)	Tiempo Req 3 [ms]
1388, 3757	0.021	57	0.050
6976, 2022	0.144	5672	0.041
2011, 2751	0.185	583	0.045
8083, 4756	0.194	3409	0.024

Gráficas Req 2 y Req 3:

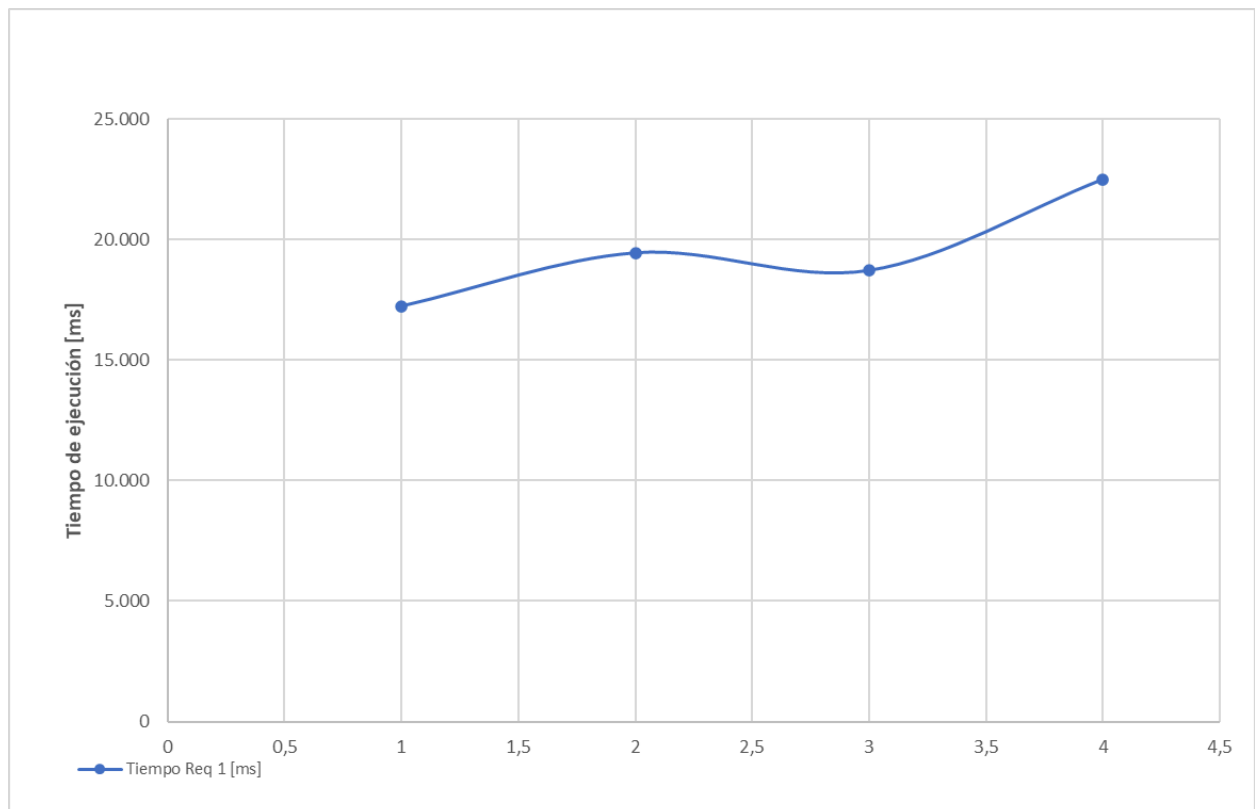
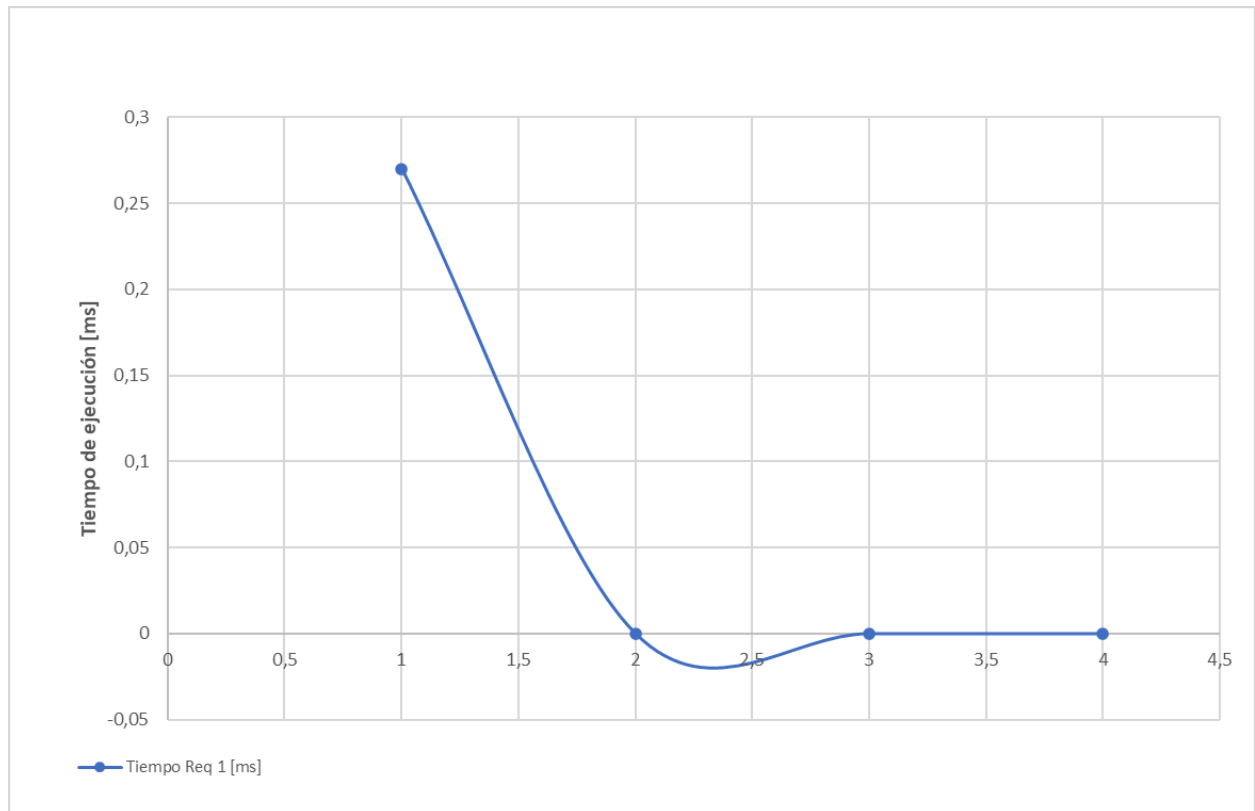




Req 4 y 5:

Req 4()	Tiempo Req 4 [ms]	Req 5()	Tiempo Req 5 [ms]
1,2.000	2.7	Cualquier id n=5	17.245
1379,352	0.395	Cualquier id n=10	19.452
1999,417	0.598	Cualquier id n=15	18.723
33,1	0.267	Cualquier id n=20	22.498

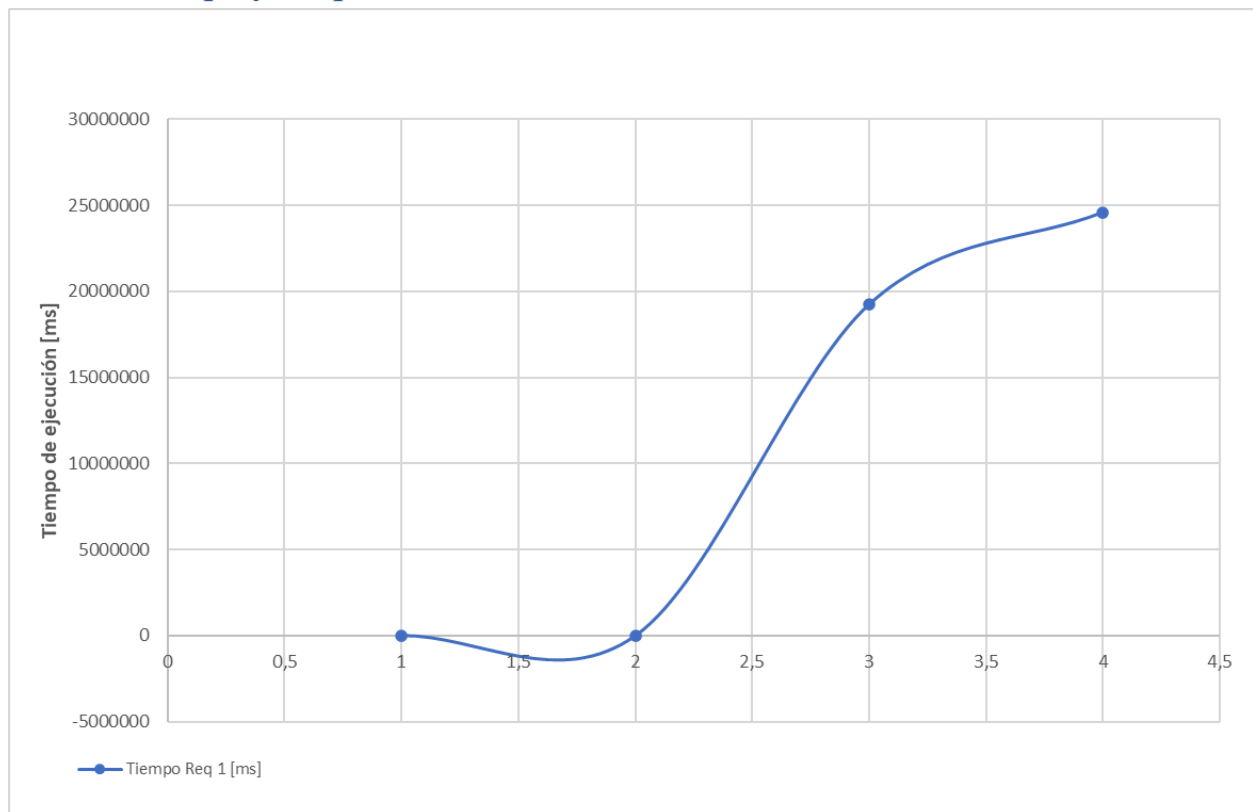
Gráficas Req 4 y Req 5:



Req 6 y Req 7

Req 6()	Tiempo Req 6 [ms]	Req 7()	Tiempo Req 7 [ms]
10	18571.703	Cualquier id golf	
15	26023.501	boxeo,lucha	
20	19251.261	piano,surf	
25	24590.069	remo,golf,tenis	

Gráficas Req 6 y Req 7:



Máquina 2

Resultados requerimientos

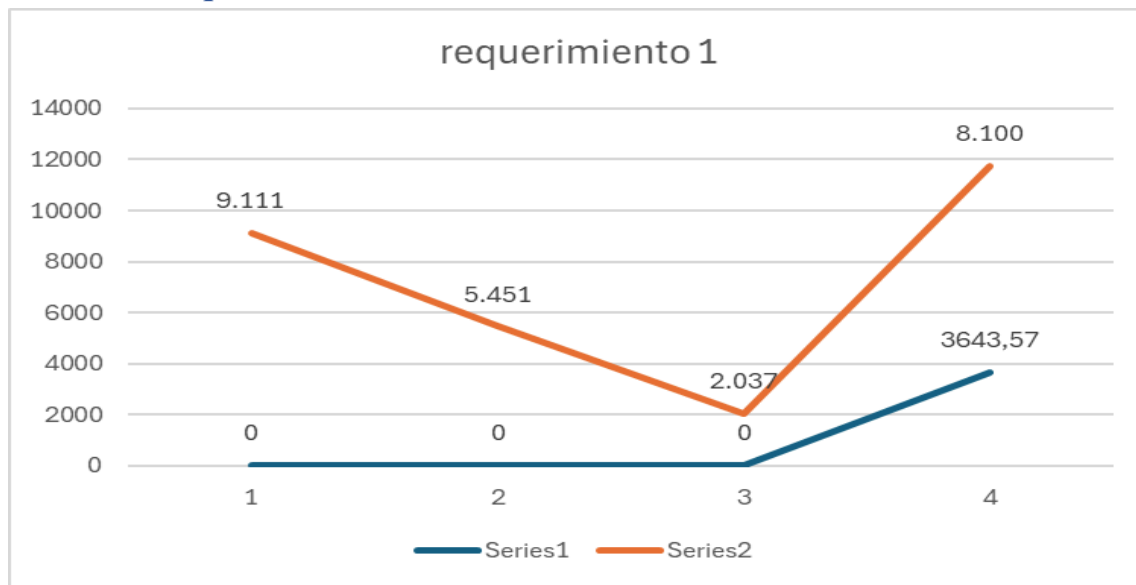
Carga de datos :

Carga de datos	Tiempo[ms]:37.448
----------------	-------------------

Req 1 :

Req 1(start_id, end_id)	Tiempo Req 1 [ms]
804, 639	9.111
6350, 45	5.451
5521, 1927	2.037
3643,57	8.100

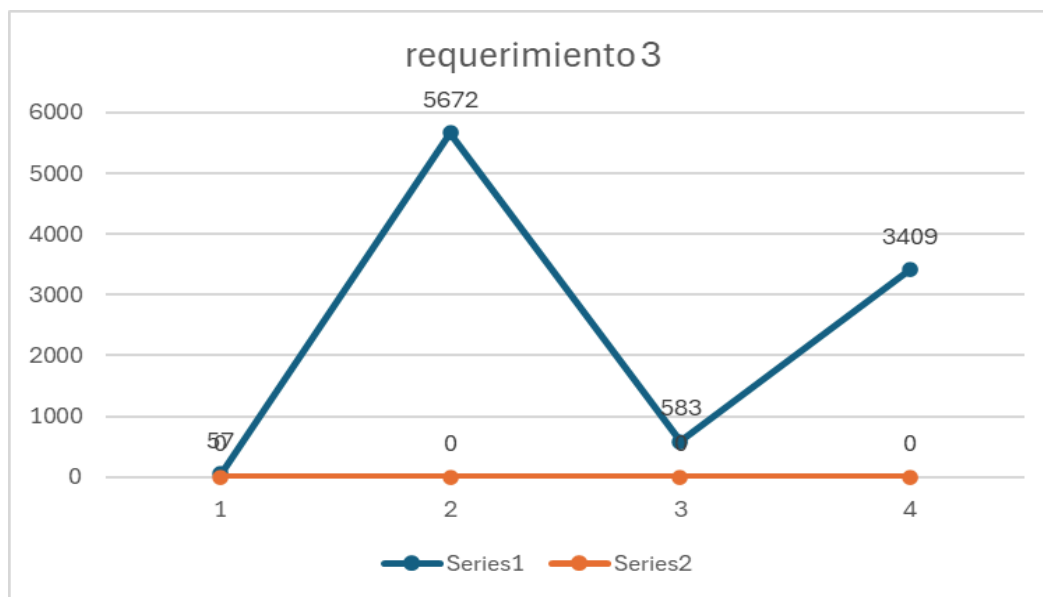
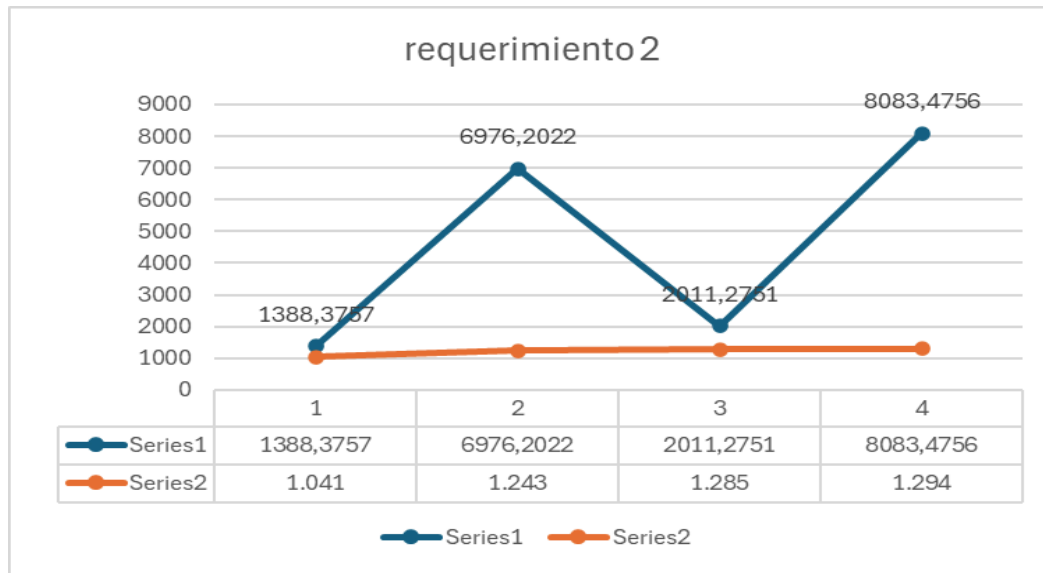
Gráficas Req 1 :



Req 2 :

Req 2(start_id, end_id)"basic"	Tiempo Req 2 [ms]	Req 3(user_id)	Tiempo Req 3 [ms]
1388, 3757	0.041	57	0.150
6976, 2022	0.243	5672	0.141
2011, 2751	0.285	583	0.085
8083, 4756	0.294	3409	0.064

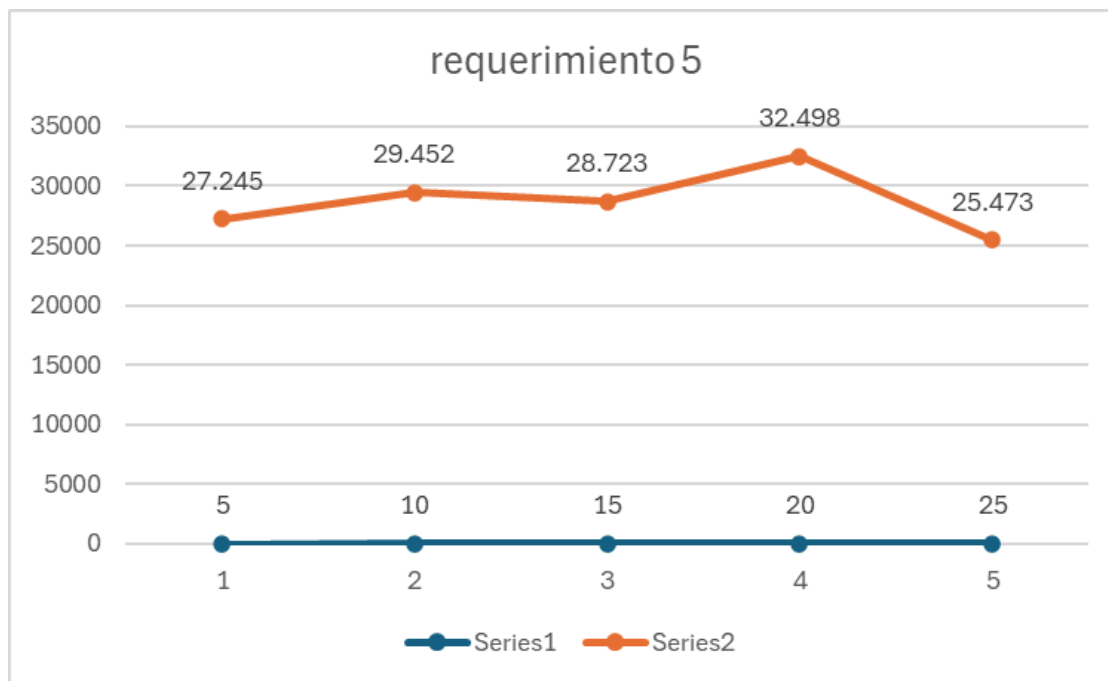
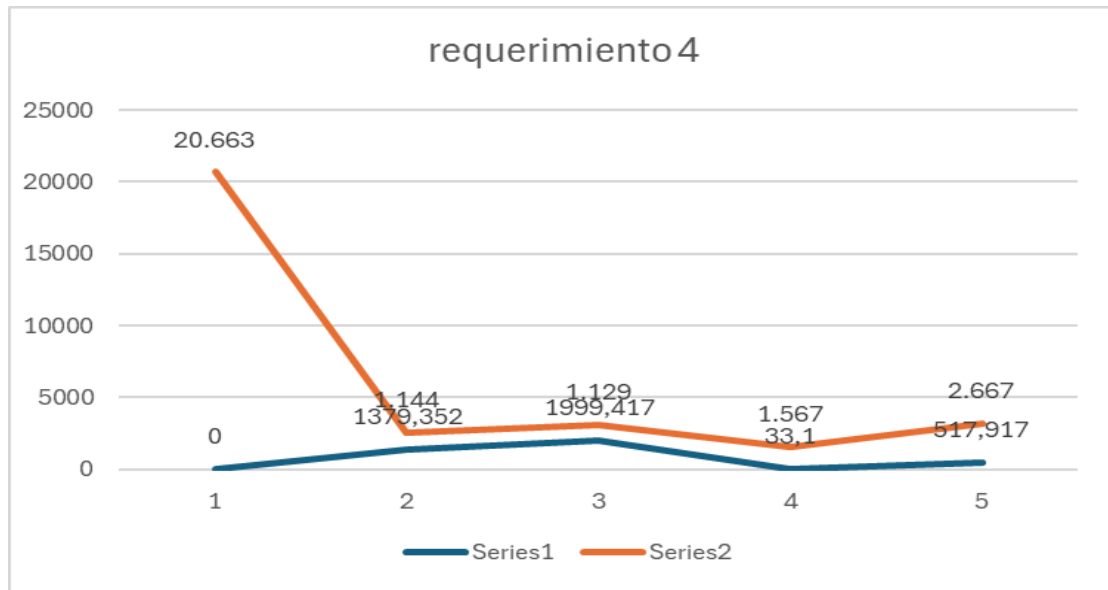
Gráficas Req 2 y Req 3:



Req 4 y 5:

Req 4()	Tiempo Req 4 [ms]	Req 5()	Tiempo Req 5 [ms]
1,2.000	20.663	5	27.245
1379,352	1.144	10	29.452
1999,417	1.129	15	28.723
33,1	1.567	20	32.498
517,917	2.667	25	25.473

Gráficas Req 4 y Req 5:

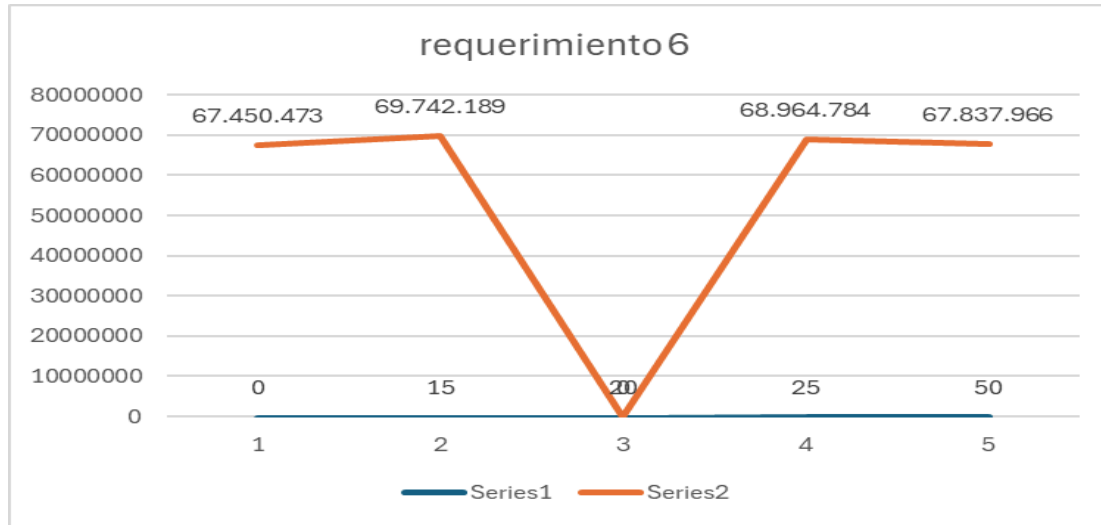


Req 6 y Req 7

Req 6()	Tiempo Req 6 [ms]	Req 7()	Tiempo Req 7 [ms]
10	67450.473	golf	
15	69742.189	boxeo,lucha	
20	69791.136	piano,surf	

25	68964.784	remo,golf,tenis
50	67837.966	baloncesto,cant o,pesas

Gráficas Req 6 y Req 7:



Máquina 3

Resultados requerimientos

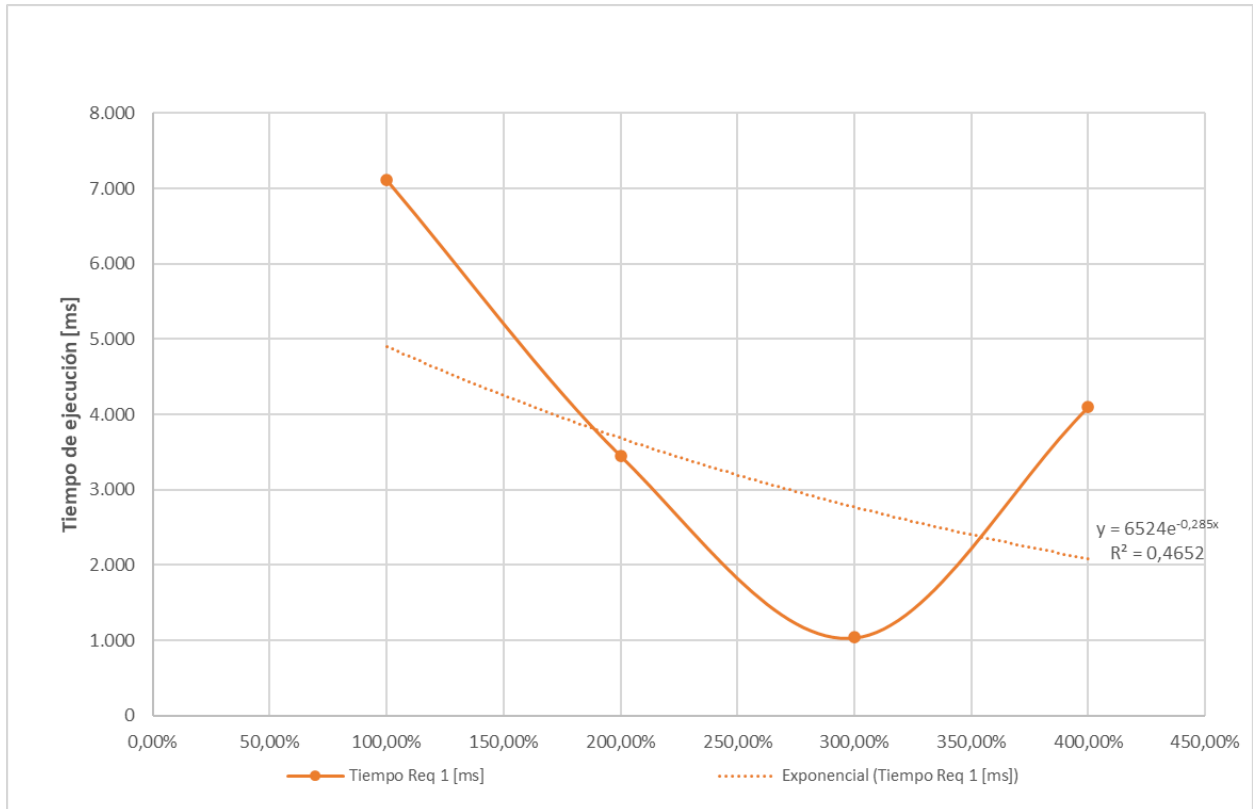
Carga de datos :

Carga de datos	Tiempo[ms]:22.987
----------------	-------------------

Req 1 :

Req 1(start_id, end_id)	Tiempo Req 1 [ms]
804, 639	14.164
6350, 45	6.202
5521, 1927	3.196
3643,57	7.237

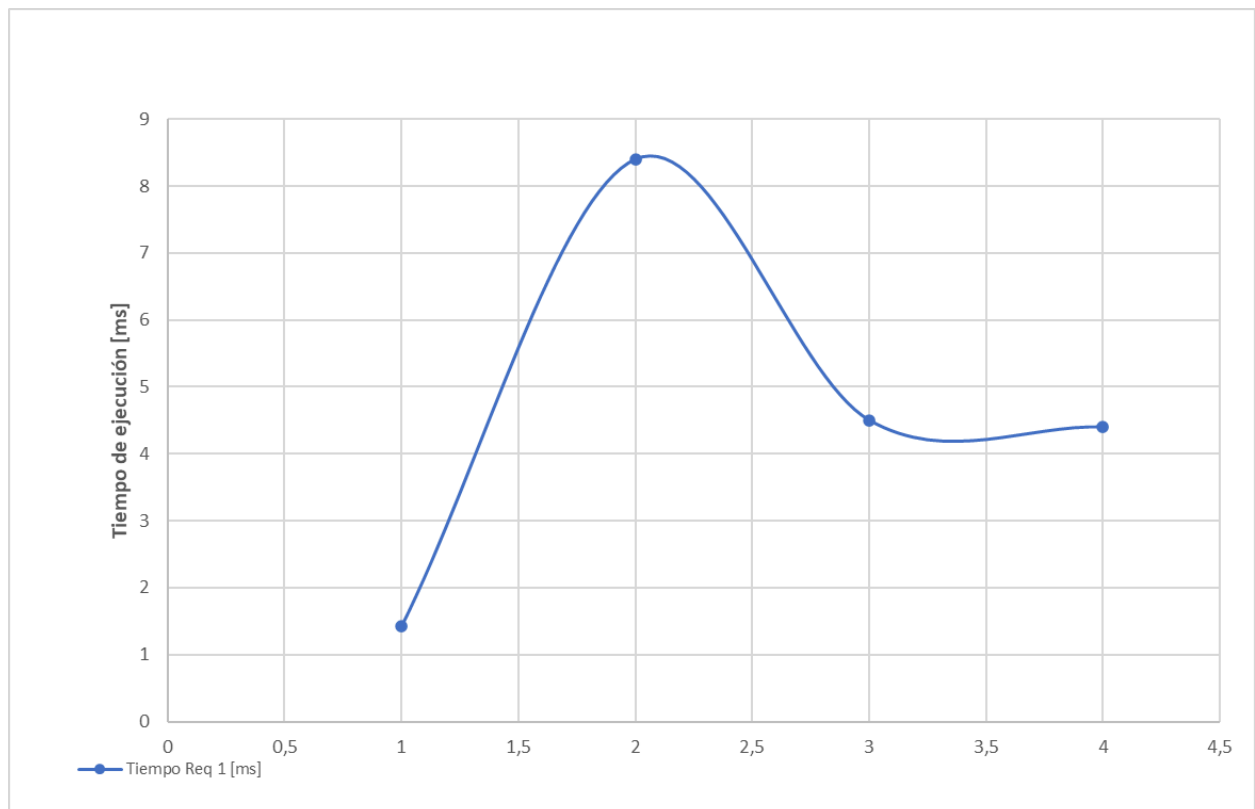
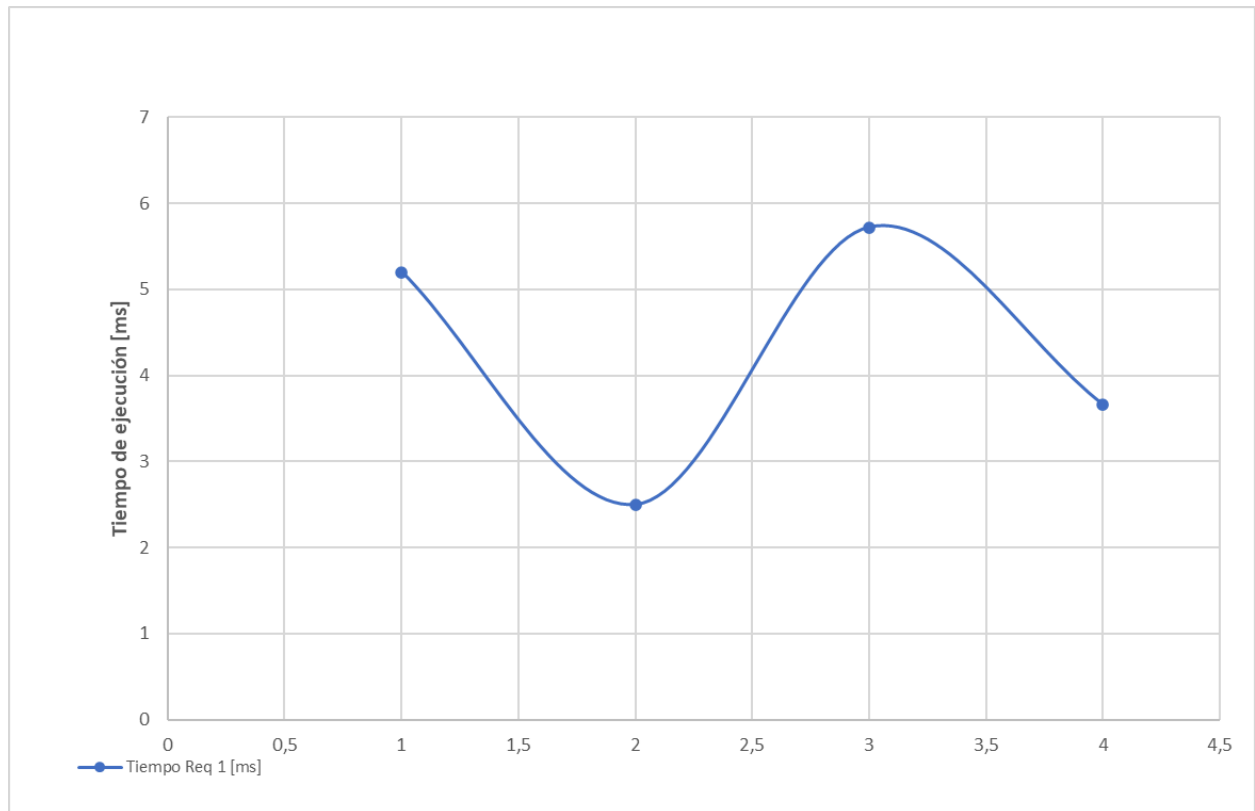
Gráficas Req 1 :



Req 2 :

Req 2(start_id, end_id)"basic"	Tiempo Req 2 [ms]	Req 3(user_id)	Tiempo Req 3 [ms]
1388, 3757	0.051	57	0.143
6976, 2022	0.285	5672	0.084
2011, 2751	0.572	583	0.045
8083, 4756	0.366	3409	0.044

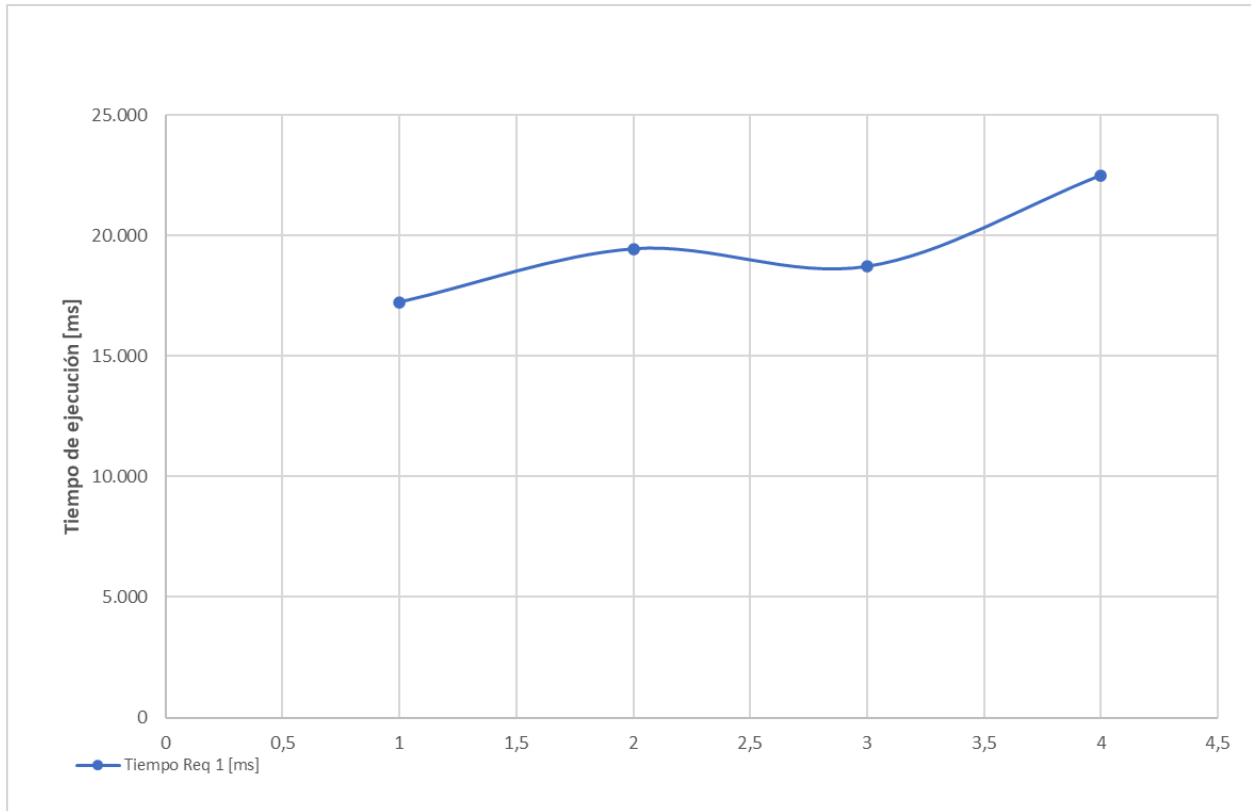
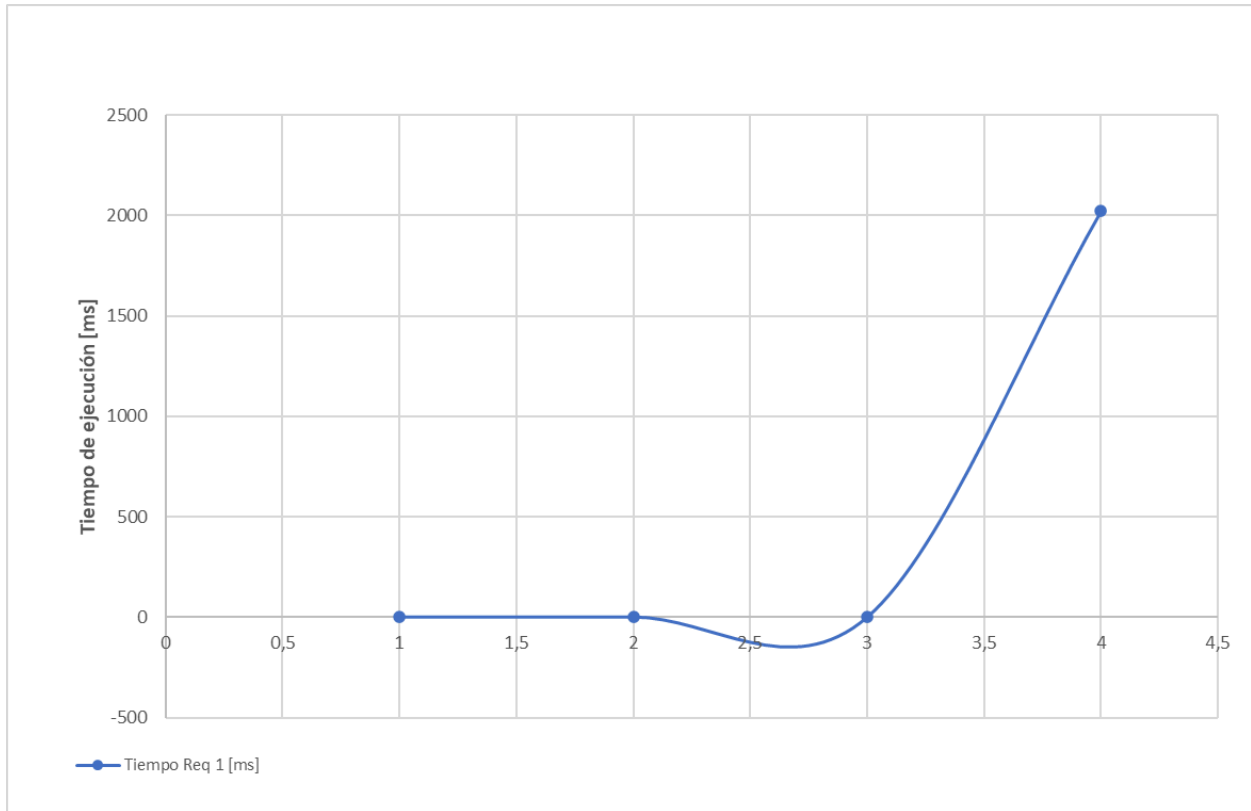
Gráficas Req 2 y Req 3:



Req 4 y 5:

Req 4()	Tiempo Req 4 [ms]	Req 5()	Tiempo Req 5 [ms]
1,2.000	1.005	Cualquier id n=5	17.245
1379,352	0.999	Cualquier id n=10	19.452
1999,417	0.990	Cualquier id n=15	18.723
33,1	2.023	Cualquier id n=20	22.498
517,917	1.422	Cualquier id n=25	25.473

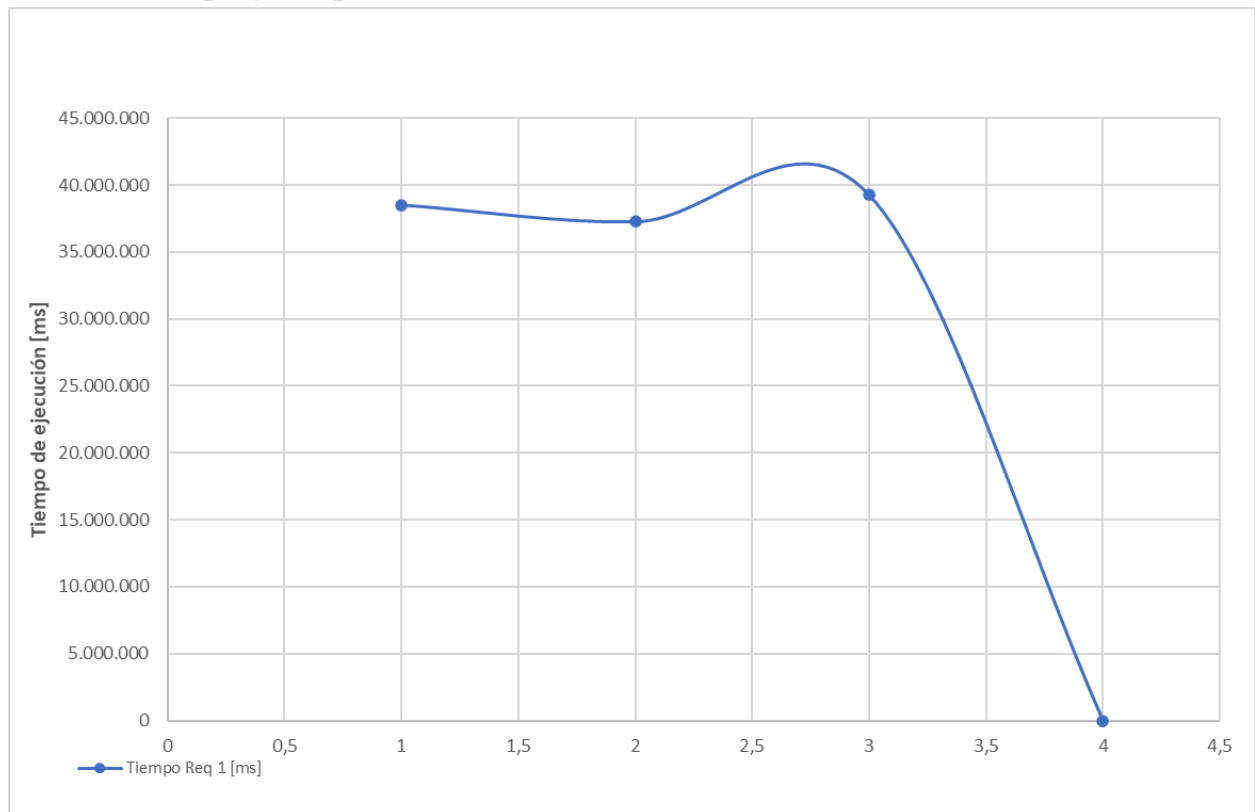
Gráficas Req 4 y Req 5:



Req 6 y Req 7

Req 6()	Tiempo Req 6 [ms]	Req 7()	Tiempo Req 7 [ms]
10	38523.517	Cualquier id golf	
15	37270.610	boxeo,lucha	
20	39251.261	piano,surf	
25	38860.898	remo,golf,tenis	
30	41528.701	baloncesto,cant o,pesas	

Gráficas Req 6 y Req 7:



Análisis de resultados comparando los resultados obtenidos con el análisis de complejidad:

1. Carga de Datos³⁴

La función load_data cumple con:

- **Carga de relaciones:** Agrega aristas entre usuarios en el grafo a partir de las relaciones cargadas del archivo CSV.
- **Carga de usuarios:** Inserta vértices en el grafo con información detallada del usuario, incluyendo datos opcionales como hobbies y foto.
- **Estadísticas:** Proporciona información resumida sobre el número de usuarios, conexiones, tipos de usuarios, promedio de seguidores, y la ciudad con más usuarios.

Dificultades:

- Formato inconsistente en las fechas (%Y-%m-%d %H:%M:%S para relaciones y %d/%m/%Y para usuarios).
- Manejo de valores nulos o incorrectos en los campos opcionales.

Solución: Implementación de validaciones para cada campo y un manejo robusto de excepciones.

2. Requerimiento 1: Identificar la red de personas entre dos usuarios

Utiliza un enfoque basado en BFS para encontrar el camino más corto entre dos usuarios. Genera una lista detallada del camino.

Reto: Optimizar la búsqueda para grafos grandes y evitar redundancias. **Solución:** Uso de una lista de visitados y almacenamiento de caminos parciales para minimizar recomputaciones.

3. Requerimiento 2: Ruta con menos personas intermedias entre dos usuarios básicos

Implementa Dijkstra simplificado para encontrar la ruta con menor cantidad de pasos. Solo considera usuarios de tipo basic.

Reto:

- Implementar correctamente la lógica para filtrar usuarios básicos en tiempo real.
- Manejar grafos con rutas múltiples de igual longitud.

Solución: Ajuste en las condiciones de búsqueda y mejora en el seguimiento de caminos previos.

4. Requerimiento 3: Amigo más popular

Determina el amigo de un usuario con más seguidores mediante un conteo iterativo.

Reto: Procesar grandes cantidades de datos sin comprometer la eficiencia. **Solución:** Evitar recorridos redundantes al calcular seguidores de cada amigo una sola vez.

5. Requerimiento 4: Amigos en común entre dos usuarios

Identifica los amigos compartidos entre dos usuarios y detalla su información.

Reto:

- Comparar grandes listas de amigos sin sobrecargar la memoria.
- Manejo de usuarios sin amigos.

Solución: Uso de conjuntos para intersección eficiente y manejo robusto de excepciones.

6. Requerimiento 5: Top N amigos con más conexiones

Genera una lista ordenada de amigos de un usuario según el número de usuarios que siguen.

Reto:

- Ordenar eficientemente amigos por su nivel de conexiones.
- Filtrar adecuadamente los amigos relevantes.

Solución: Implementación de un criterio de ordenación en merge_sort para listas grandes.

7. Requerimiento 6: N usuarios más populares

Encuentra los N usuarios con más seguidores y construye un árbol de conexiones entre ellos.

Reto:

- Conectar a los usuarios seleccionados en una estructura arbórea significativa.
- Evitar ciclos y redundancias.

Solución: Uso de BFS para construir un árbol con enlaces padres-hijos y garantizar conectividad.

Reflexión sobre la experiencia del desarrollo

El equipo enfrentó varios retos, incluyendo:

- **Complejidad de los datos:** Manejar múltiples formatos y casos especiales en los CSV.
- **Eficiencia:** Optimizar algoritmos para evitar tiempos de ejecución elevados en un grafo extenso.
- **Coordinación:** Sincronizar trabajo en varias funciones para garantizar consistencia y evitar errores de integración.

Lecciones aprendidas:

- La validación de datos es clave para evitar errores en etapas posteriores.
- Diseñar pruebas para cada módulo agiliza la detección y corrección de fallos.
- Dividir responsabilidades por requerimientos ayuda a enfocar esfuerzos y garantizar claridad.