

ANÁLISIS DEL RETO

Manuela Rivas, 202021971, m.rivas2@uniandes.edu.co

Lucas Nieto, 202012861, l.nietom@uniandes.edu.co

Diego Londoño, 202012494, d.londonob@uniandes.edu.co

Ambientes de pruebas

Máquina	
Procesadores	AMD RYZEN R7
Memoria RAM (GB)	16
Sistema Operativo	Windows 10

Requerimiento 1: Listar películas estrenadas en un periodo de tiempo

Descripción

Entrada	Ingresa el catálogo general de películas, un año inferior y otro superior para delimitar el rango de búsqueda
Salidas	Retorna ordenada la cantidad de películas estrenadas en ese rango, además de la información de las primeras y últimas 3.
Implementado (Sí/No)	Implementado correctamente por Diego Londoño y Lucas Nieto.

Análisis de complejidad

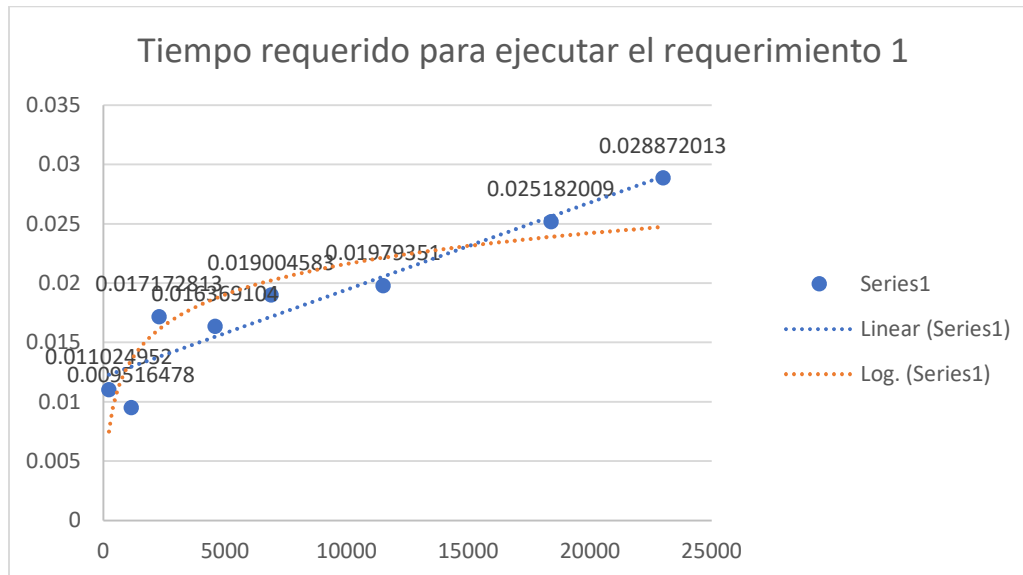
Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megsort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.011025
5% (N = 1148)	0.009516
10% (N = 2298)	0.017173
20% (N = 4598)	0.016369
30% (N = 6898)	0.019005
50% (N = 11498)	0.019794
80% (N = 18397)	0.025182

large (N = 22998)	0.028872
-------------------	----------

Graficas



Análisis

Dado que la aproximación lineal falla para pequeñas muestras de datos, mientras que el ajuste logarítmico falla para las mayores cantidades de datos, lo que mejor modelaría este ordenamiento temporal sería la combinación de estas dos funciones: $N\log(N)$.

Requerimiento 2: Listar programas de televisión agregados en un periodo de tiempo

Descripción

Entrada	• Fecha inicial del periodo (con formato "%B %d, %Y").
Salidas	El número total de programas de TV estrenados en ese periodo adicionalmente imprime la información de los primeros y últimos 3 programas de la lista ya ordenada.
Implementado (Sí/No)	Implementado correctamente por Diego Londoño.

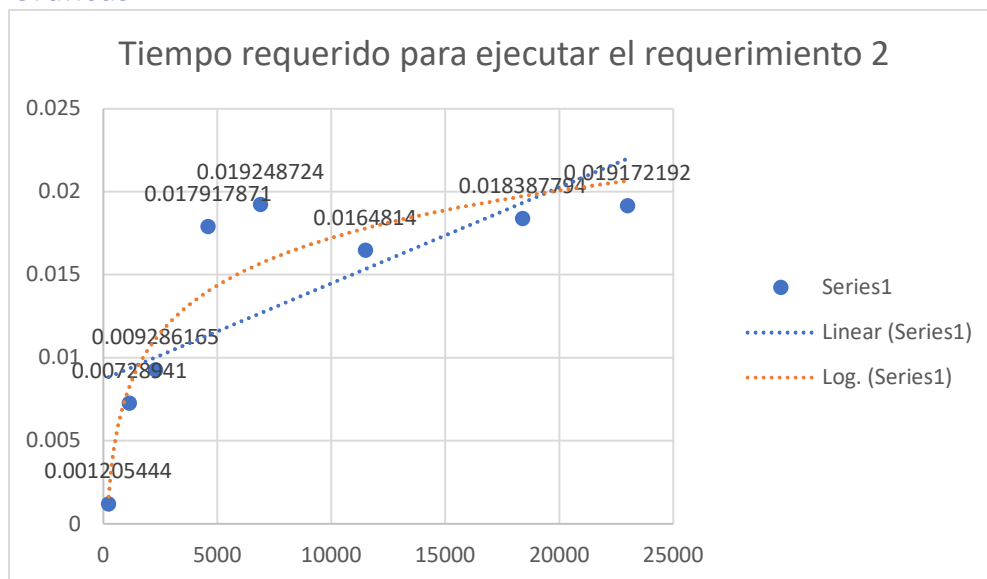
Análisis de complejidad

Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megesort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.001205
5% (N = 1148)	0.007289
10% (N = 2298)	0.009286
20% (N = 4598)	0.017918
30% (N = 6898)	0.019249
50% (N = 11498)	0.016481
80% (N = 18397)	0.018388
large (N = 22998)	0.019172

Graficas



Análisis

Dado que la aproximación lineal falla para pequeñas muestras de datos, mientras que el ajuste logarítmico falla para las mayores cantidades de datos, lo que mejor modelaría este ordenamiento temporal sería la combinación de estas dos funciones: $N\log(N)$.

Requerimiento 3: Encontrar contenido donde participa un actor

Descripción

Entrada	Un string con el nombre de un actor del cual el usuario quiere encontrar contenido
Salidas	Dos tablas, la primera contiene conteos del contenido en el que ha participado el actor y la segunda la información ordenada de los primeros y últimos 3 títulos de ese actor.
Implementado (Sí/No)	Implementado correctamente por Diego Londoño.

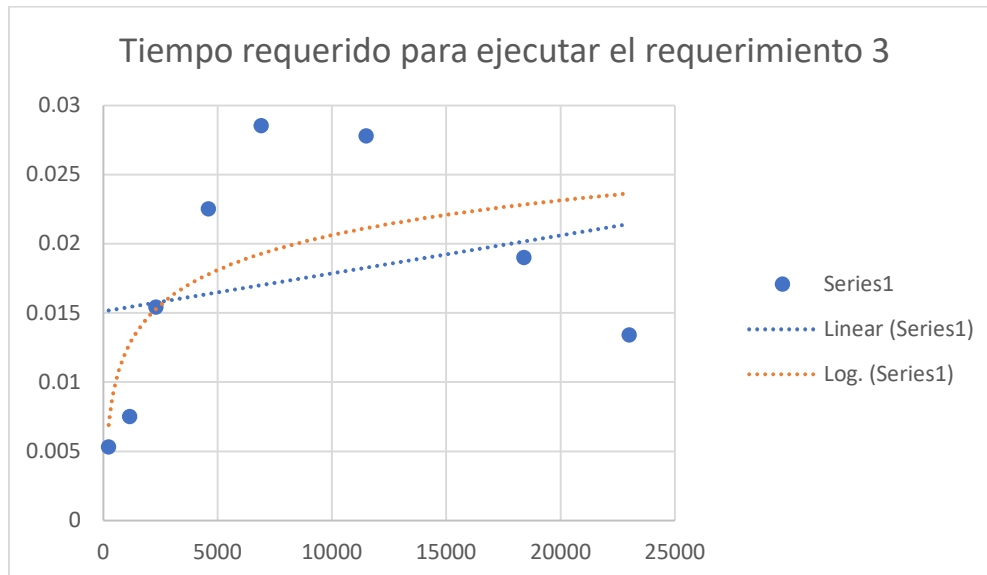
Análisis de complejidad

Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megsort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.005346
5% (N = 1148)	0.007533
10% (N = 2298)	0.015443
20% (N = 4598)	0.022533
30% (N = 6898)	0.038556
50% (N = 11498)	0.027807
80% (N = 18397)	0.019027
large (N = 22998)	0.013423

Graficas



Análisis

Con estos datos acontece algo particular, el equipo en el que se realizaron estas pruebas midió una distribución casi que normal en cuanto al tiempo requerido para completar el requerimiento, esto se puede deber a una falla en el análisis del ordenamiento o que la máquina que corrió las pruebas falló al medir los tiempos de los procesos o dedico hilos de memoria a otros procesos mientras se tomaban estas pruebas y se ralentizaron en los valores medios.

Requerimiento 4: Encontrar contenido por género

Descripción

Entrada	Un string que contiene al género que el usuario está interesado en buscar
Salidas	La cantidad de contenido correspondiente a este género y una tabla ordenada con los 3 primeros y últimos títulos correspondientes a este género.
Implementado (Sí/No)	Implementado correctamente por Lucas Nieto.

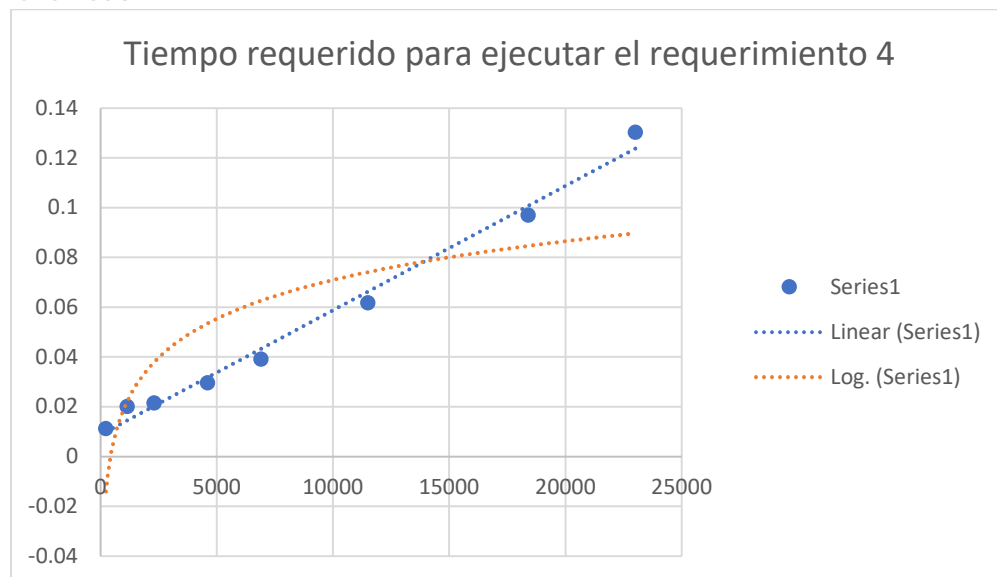
Análisis de complejidad

Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megsort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.011272
5% (N = 1148)	0.020139
10% (N = 2298)	0.0215
20% (N = 4598)	0.0296
30% (N = 6898)	0.039157
50% (N = 11498)	0.061771
80% (N = 18397)	0.097022
large (N = 22998)	0.130404

Graficas



Análisis

La tendencia que mejor ajusta los datos de las pruebas de este requerimiento es una recta, mientras que el ajuste logarítmico a penas predice un poco mejor los tiempos de las muestras pequeñas de datos, sin embargo no puede ser lineal porque hay un ordenamiento merge involucrado, por lo que esto puede deberse a la sensibilidad de la máquina de pruebas.

Requerimiento 5: Encontrar contenido producido en un país

Descripción

Entrada	Un String que contenga el país del cual el usuario quiere encontrar contenido.
Salidas	Los títulos y el tipo de contenido que se ha producido en dicho país.
Implementado (Sí/No)	Implementado correctamente por Manuela Rivas.

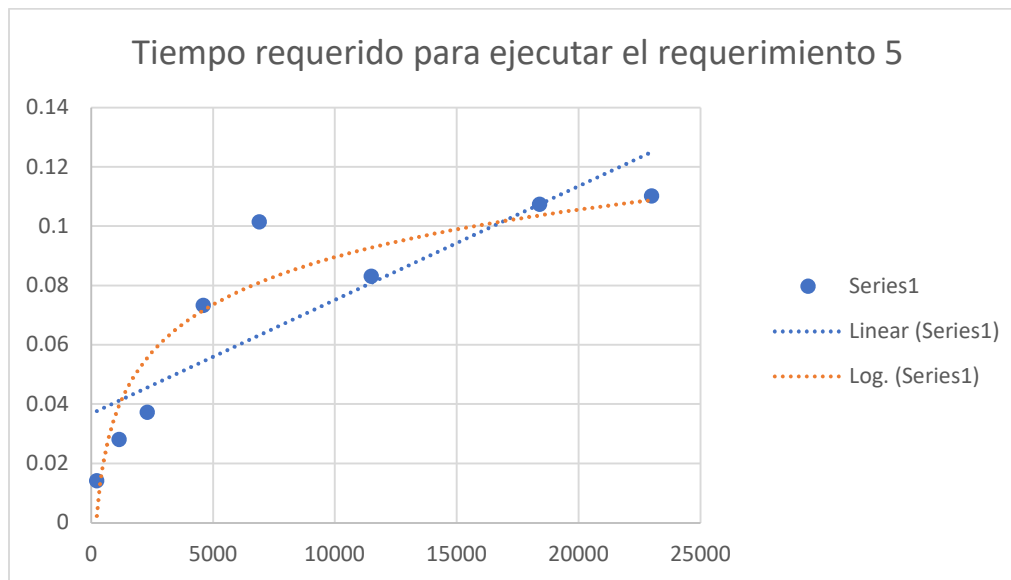
Análisis de complejidad

Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megsort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.014207
5% (N = 1148)	0.028174
10% (N = 2298)	0.037303
20% (N = 4598)	0.073366
30% (N = 6898)	0.101578
50% (N = 11498)	0.183179
80% (N = 18397)	0.007498
large (N = 22998)	0.010285

Graficas



Análisis

El ajuste logarítmico ajusta mucho mejor estos datos que una regresión lineal, hay discrepancia con el ordenamiento temporal linealítico esperado, por lo que hubo un error ya sea en la máquina de pruebas o en el cálculo del ordenamiento.

Requerimiento 6: Encontrar contenido con un director involucrado

Descripción

Entrada	Un string con el nombre de un director del cual el usuario quiere encontrar contenido
Salidas	Un reporte con la cantidad de películas o programas de TV que ha dirigido el director buscado y detallado por nombre de los títulos, plataformas, etc.
Implementado (Sí/No)	Implementado correctamente por Lucas Nieto.

Análisis de complejidad

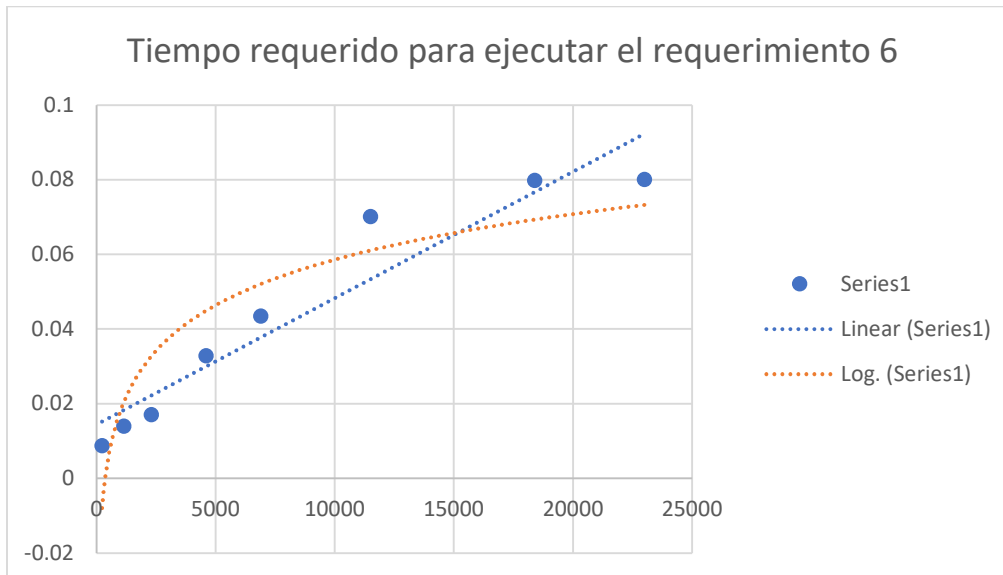
Pasos	Complejidad
Asignación de variable	$O(1)$
mp.contains	$O(N)$
Asignación a mp.get	$O(1)$
Asignación con me.getValues	$O(N)$
Megsort peor caso	$N\log(N)$
TOTAL	$N\log(N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.008749

5% (N = 1148)	0.013976
10% (N = 2298)	0.017076
20% (N = 4598)	0.032797
30% (N = 6898)	0.04346
50% (N = 11498)	0.010174
80% (N = 18397)	0.01181
large (N = 22998)	0.012879

Graficas



Análisis

Dado que la aproximación lineal falla para pequeñas muestras de datos, mientras que el ajuste logarítmico falla para las mayores cantidades de datos, lo que mejor modelaría este ordenamiento temporal sería la combinación de estas dos funciones: $N\log(N)$.

Requerimiento 7: Listar el TOP (N) de géneros con más contenido

Descripción

Entrada	Un número entero que represente la cantidad de elementos a considerar en el top
Salidas	Tabla del top de géneros con más contenido con sus respectivos conteos por tipo de contenido y plataforma
Implementado (Sí/No)	Implementado correctamente por Manuela Rivas.

Análisis de complejidad count by

Pasos	Complejidad
Asigna variable	$O(1)$

Asigna valor a element	$O(N)$
Compara != y ==	$O(N)$
+=1 peor caso	$O(N)$
TOTAL	$O(N)$

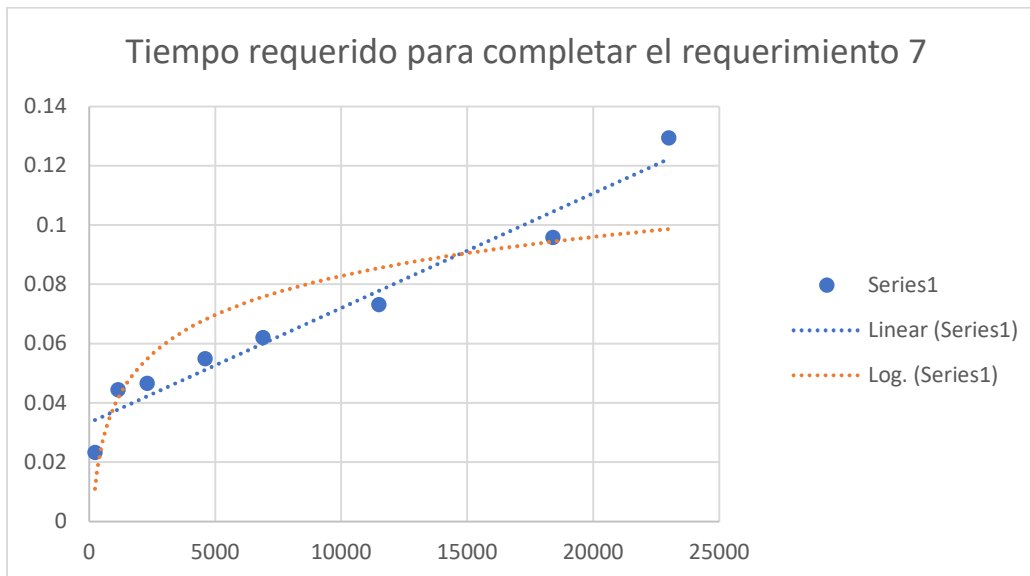
Análisis de complejidad

Pasos	Complejidad
Crea un diccionario	$O(1)$
Asignación con Valueset	$O(N)$
Mergesort	$N\log(N)$
Crea una lista	$O(1)$
Agrega variables a una lista	$O(M)$ con M igual al número de top que desea incluir el usuario.
Count_by	$O(N)$
TOTAL	$O(N\log N)$

Pruebas Realizadas

Muestra de datos (N)	Tiempo de respuesta (s)
small (N = 228)	0.023351
5% (N = 1148)	0.044534
10% (N = 2298)	0.046667
20% (N = 4598)	0.055017
30% (N = 6898)	0.062156
50% (N = 11498)	0.073195
80% (N = 18397)	0.095936
large (N = 22998)	0.129481

Graficas



Análisis

Dado que la aproximación lineal falla para pequeñas muestras de datos, mientras que el ajuste logarítmico falla para las mayores cantidades de datos, lo que mejor modelaría este ordenamiento temporal sería la combinación de estas dos funciones: $N\log(N)$.