Universidad de los Andes
Facultad de Ingeniería

Departamento de Ingeniería de Sistemas y Computación
Estructuras de Datos y Algoritmos
ISIS-1225

ABET Engineering Accreditation Commission

ANÁLISIS DEL RETO 3

Tomás Sierra Sanchez, 202221567, t.sierras@uniandes.edu.co

Juan Francisco Rodríguez, 202214603, jf.rodriguezc1@uniandes.edu.co

Carlos Peña, 202222516, c.penaa@uniandes.edu.co

Requerimiento 1

Descripción

Este requerimiento se encarga de conocer los accidentes que ocurrieron en la ciudad durante un intervalo de fechas específico.

Entrada	Control: El catálogo de datos completo, La fecha inicial en formato aaaa/mm/dd, La fecha final en formato aaaa/mm/dd
Salidas	Una lista con todos los accidentes que ocurrieron en ese lapso de tiempo.
Implementado (Sí/No)	Si. Tomás Sierra

Análisis de complejidad

Pasos	**Complejidad**
Tomar los valores del árbol	O(d)
Recorrer todos las listas de días	O(d)
Realizar un quicksort por cada lista de dia	O(acc*log(acc))
Recorrer todos los accidentes por cada lista de dia	O(acc)
TOTAL	***O(d) + O(d*(acc + acc*log(acc))***

Pruebas Realizadas

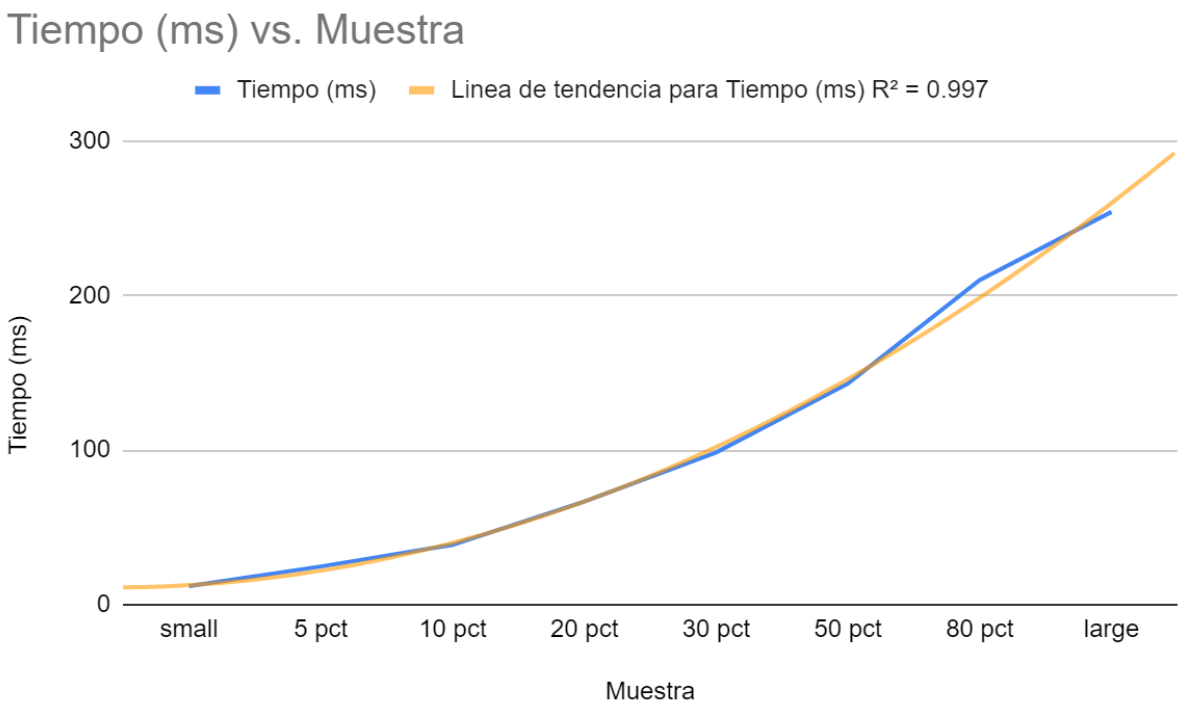
Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados la fecha de inicio es 01/11/2016 y la fecha de fin es 08/11/2016

Procesadores	**Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**
Memoria RAM	**8 GB**
Sistema Operativo	Windows 11 Single Language

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Lista con todos los accidentes que ocurrieron en ese lapso de tiempo	12.01
5 pct		24.71
10 pct		38.72
20 pct		66.68
30 pct		98.49
50 pct		143.13
80 pct		210.05
large		254.09

Gráfica



Análisis

Podemos observar en la práctica un comportamiento casi perfectamente cuadrático es decir O(n^2). Esto tiene sentido ya que vamos a recorrer por cada lista del día vamos a realizar un sort y luego un recorrido completo y teniendo en cuenta esto el comportamiento no podría ser lineal sino más bien algo del orden cuadrático en donde se revisa cada elemento por cada nuevo día.

Requerimiento 2

Descripción

Este requerimiento se encarga de conocer los accidentes ocurridos en un intervalo de tiempo del día (desde una hora y minutos iniciales hasta una hora y minutos finales) para un año y mes dados.

Entrada	Control: El catálogo de datos completo, El año a consultar, el mes a consultar, la hora inicial a consultar (HH:MM) y la hora final a consultar (HH:MM)
Salidas	La lista con todos los accidentes ocurridos en ese lapso de tiempo
Implementado (Sí/No)	Si. Tomás Sierra

Análisis de complejidad

Pasos	**Complejidad**
Tomar los valores del árbol	O(d)
Recorrer todos las listas de días	O(d)
Recorrer todos los accidentes por cada lista de dia	O(acc)
TOTAL	***O(d) + O(d * acc)***

Pruebas Realizadas

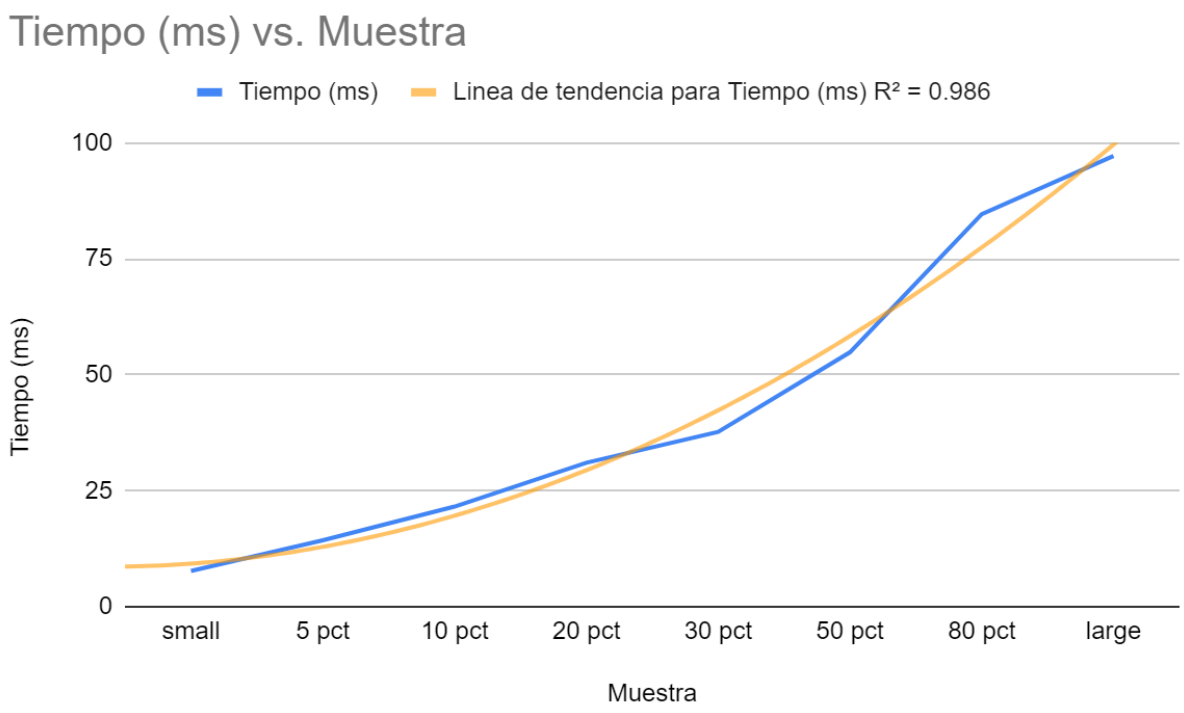
Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardado, el mes de noviembre, del año 2022 y las horas entre las 18:00 y las 21:00

Procesadores	**Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 11 Single Language

Tablas de datos

Muestra	**Salida**	**Tiempo (ms)**
small	Lista con todos los accidentes ocurridos en ese lapso de tiempo	7.7
5 pct		14.33
10 pct		21.59
20 pct		31.05
30 pct		37.7
50 pct		54.85
80 pct		84.63
large		97.17

Gráfica



Análisis

Se puede observar que en la práctica la implementación tiene una complejidad O(n^2), esto muy similar al requerimiento 1 puesto que el recorrido sobre el árbol es el mismo sin embargo podemos ahorrarnos el sort ya que ahora estamos comparando por la hora lo que reduce los tiempos en general sobre el req 1 aunque el comportamiento como tal sea similar.

Requerimiento 3

Descripción

Este requerimiento se encarga de Encontrar el subsector económico con el menor total de retenciones para un año específico

Entrada	Catálogo con todos los datos, clase del accidente, nombre de la vía de la ciudad
Salidas	Una lista con los tres accidentes más recientes ocurridos en esa vía bajo esa clase de accidente.
Implementado (Sí/No)	Sí, Tomás Sierra

Análisis de complejidad

Pasos	Complejidad
Recorrer todas las listas de dia	O(d)
Recorrer la lista completa de dia	O(acc)
Realizar un quicksort para ordenar	O(flog(f))
TOTAL	***O(d * acc) + O(flog(f))***

Pruebas Realizadas

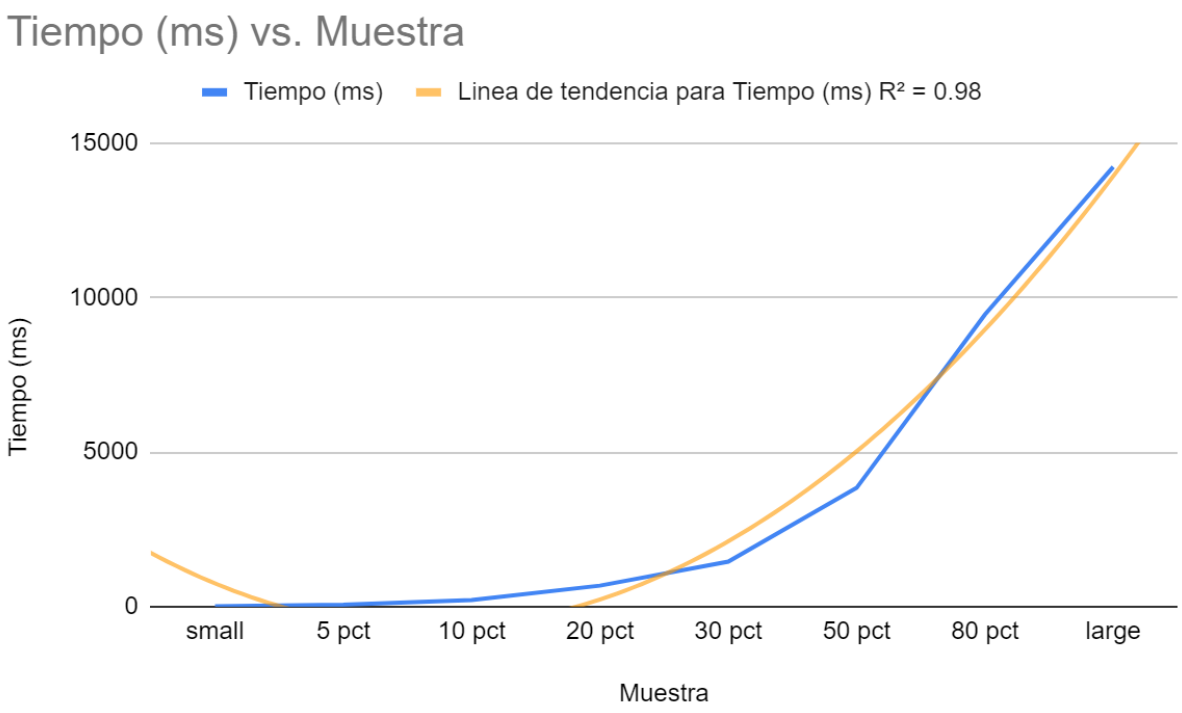
Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardado, la calle de AV Avenida de las américas, y el tipo choque.

Procesadores	**Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 11 Single Language

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Una tabla general del sub sector económico que más retenciones tuvo en el año. Una tabla donde se ve las actividades que más y menos aportaron a dichas retenciones	12.2
5 pct		64.47
10 pct		212.12
20 pct		681.08
30 pct		1457.13
50 pct		3841.52
80 pct		9450.47
large		14223.01

Gráfica



Análisis

Se puede observar que la curva modela un comportamiento cuadrático O(n^2) esto tiene sentido ya que estamos recorriendo toda la lista y por cada día estamos recorriendo todos los accidentes que ocurrieron ese día.

Requerimiento 4

Descripción

Este requerimiento se encarga de encontrar el subsector económico con los mayores costos y gastos de nómina para todos los años disponibles

Entrada	Data_structs: Es el catálogo de datos completo Inicio: La fecha de inicio de búsqueda Final: La fecha final de búsqueda Gravedad: El tipo de gravedad del accidente
Salidas	Una lista con los accidentes que cuenten con la gravedad ingresada por el usuario dentro del rango de fechas también estipuladas por el usuario.

Implementado (Sí/No)	Si, Juan Francisco Rodríguez Contreras

Análisis de complejidad

Pasos	**Complejidad**
Tomar los valores del árbol	O(v)
Recorrer todas las listas de los días	O(d)
Realizar un quicksort por cada lista de cada dia	O(acc*log(acc))
Recorrer todos los accidentes por cada lista de dia	O(l)
TOTAL	***O(v) + O(d*(l + acc*log(acc))***

Pruebas Realizadas

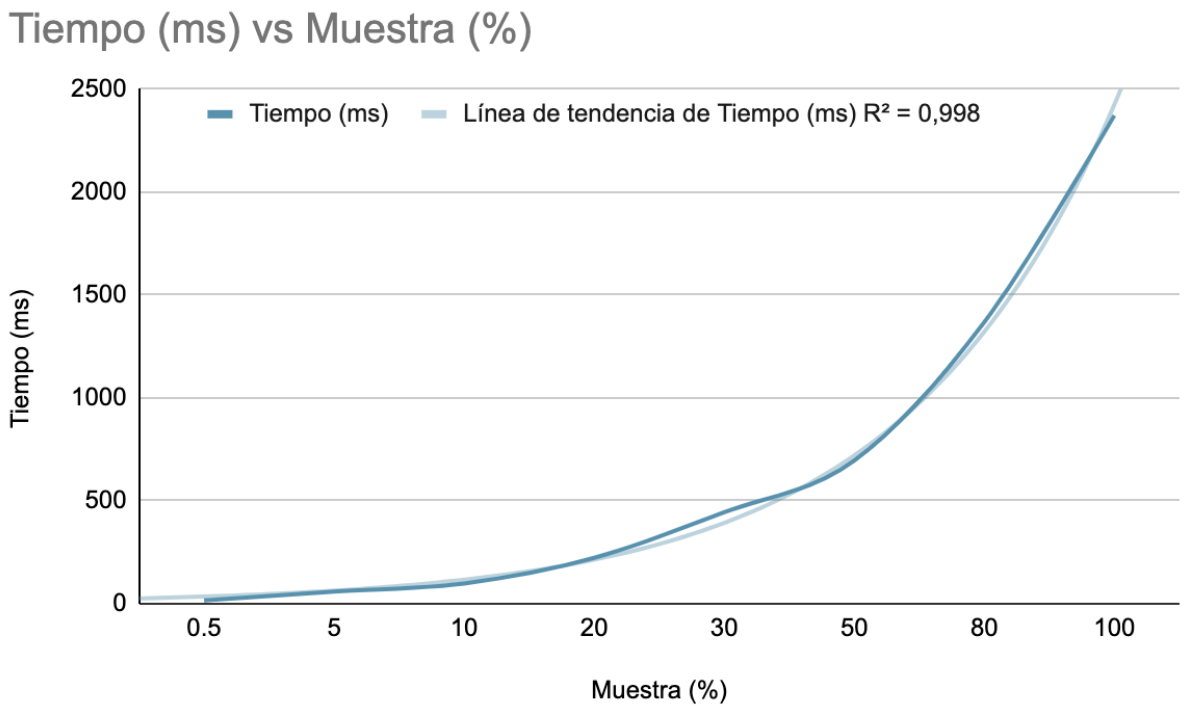
Los parámetros de entrada son: catálogo, CON MUERTOS, 01/10/2016 y 01/10/2018. Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	MacOS Catalina Versión 10.15.7

Tablas de datos

Muestra	Salidas	Tiempo (ms)
small	Una lista con los accidentes que cuenten con la gravedad ingresada por el usuario dentro del rango de fechas también estipuladas por el usuario.	15,57
5 pct		58,78
10 pct		98,29
20 pct		222,79
30 pct		444,28
50 pct		695,48
80 pct		1368,53
large		2372,55

Gráfica



Análisis

Podemos observar en la práctica un comportamiento casi perfectamente cuadrático es decir O(n^2). Esto tiene sentido ya que vamos a recorrer todas las listas de los días, luego realizamos quicksort a cada una de esas listas y por último recorrer cada accidente de las listas de cada día.

Requerimiento 5

Descripción

Este requerimiento se encarga de reportar los 10 accidentes menos recientes ocurridos en un mes y año para una localidad de la ciudad.

Entrada	control: El catálogo de datos completo, loc: Localidad a consultar, mes: Mes a consultar, anio: Año a consultar
Salidas	Lista con los 10 accidentes más recientes para los parámetros dados.
Implementado (Sí/No)	Si, Carlos Peña

Análisis de complejidad

Pasos	Complejidad

Se obtiene el último día del mes para dicho mes en dicho año. Obtenemos los valores para este mes con values() en forma de una lista filtrada (L).	O(n)
Para todas las listas de accidentes en la lista filtrada (L) se ordena con mergesort por 'HORA_OCURRENCIA_ACC' ascendentemente todos sus elementos (A). Para cada lista ordenada se recorren todos sus accidentes (A), si la localidad coincide, se guarda la información pertinente en una lista filtrada (L2) y se agrega 1 ocurrencia a (total de accidentes (tot_ac).	O(L * A log(A) A)
TOTAL	***O(n) + O(L * A² log(A))***

Pruebas Realizadas

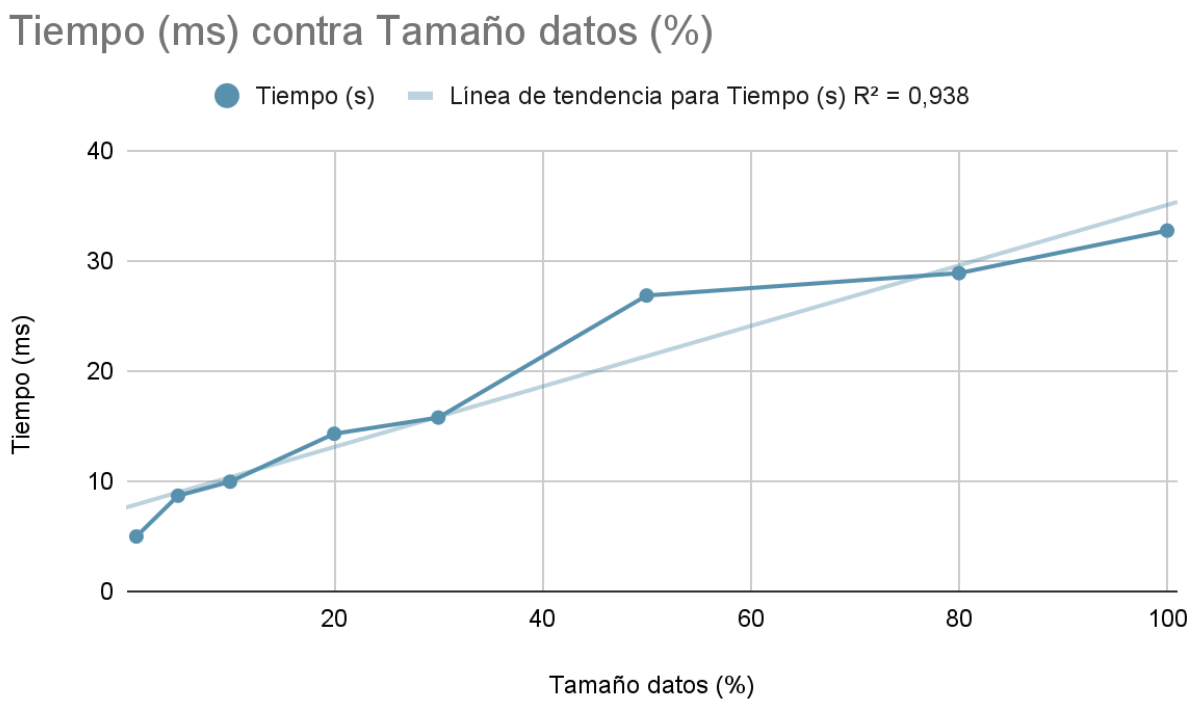
Los parámetros de entrada son: catálogo, FONTIBON (loc), 11 (mes), 2016 (anio). Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones:

Procesadores	**AMD Ryzen 7 5700G 3.80 GHz (8 cores 16 threads)**
Memoria RAM	32 GB 3600 MHz DDR4
Sistema Operativo	Windows 10 Home - 21H2

Tablas de datos

Tamaño datos (%)	Tiempo (ms)
1	4.98
5	8.68
10	9.95
20	14.3
30	15.77
50	26.84
80	28.87
100	32.73

Gráfica



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad lineal aproximada O(n) + O(L * A² log(A)). Esto se debe a que la complejidad que más aporta al tiempo es recorrer los datos con values(). Vemos que (L) se mantiene constante, pues son el mismo número de días para el mes de noviembre en todos los tamaños de datos y la variación se da en el número de accidentes (A).

Requerimiento 6

Descripción

Este requerimiento se encarga de mostrar los N accidentes ocurridos dentro de una zona específica para un mes y un año

Entrada	Control: El catálogo de datos completo, Mes, Año entre 2015 y 2022, Coordenadas del centro del área (Latitud y Longitud), Radio del área en km, Número de accidentes

Salidas	Tabla de sector económico con los mayores ingresos netos. Tabla de subsector económico que aportó lo mínimo, y tabla de subsector económico que aportó lo maximo
Implementado (Sí/No)	Si, Tomás Sierra

Análisis de complejidad

Pasos	**Complejidad**
Tomar los valores del árbol	O(d)
Recorrer todas las listas de dia	O(d)
Recorrer la lista completa de dia	O(acc)
Realizar un quicksort para ordenar	O(flog(f))
Recorrer i elementos de la lista final para ajustarla a la lista a tabular	O(i)
TOTAL	***O(d) + O(d*acc) + O(flog(f) + O(i)***

Pruebas Realizadas

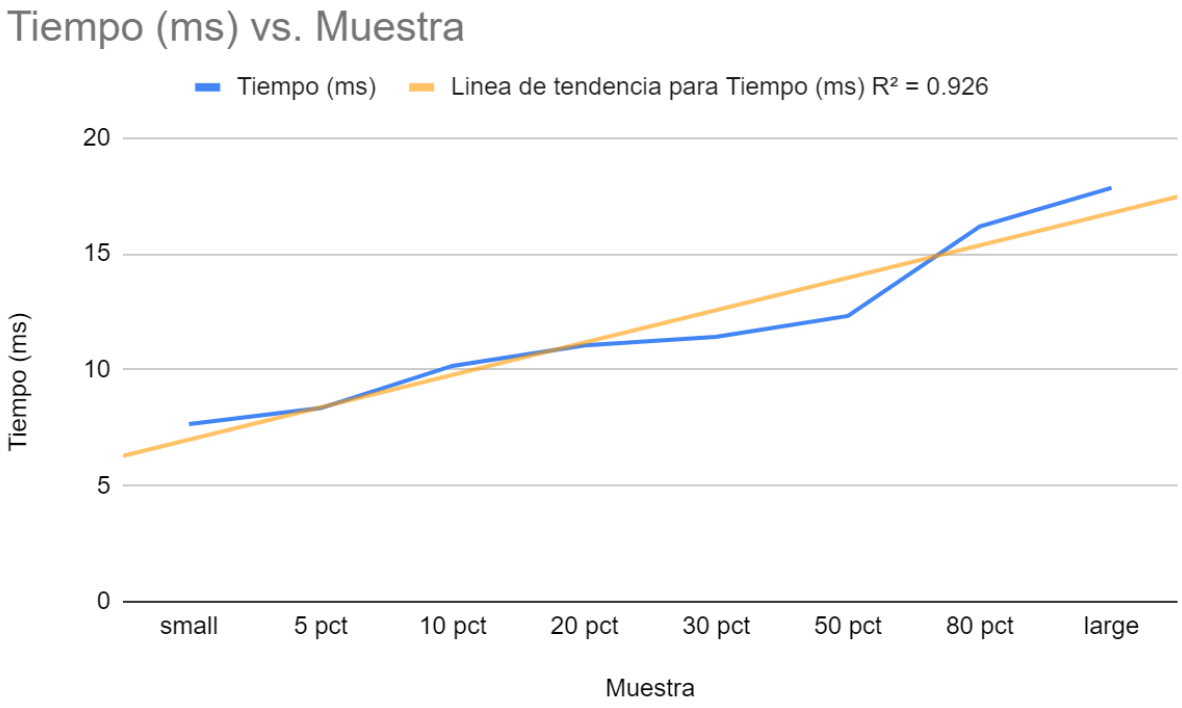
Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardado, y los 3 accidentes más cercanos al punto con latitud 4.674, longitud -74.068 dentro de un radio de 5.0 km para el mes de enero de 2022.

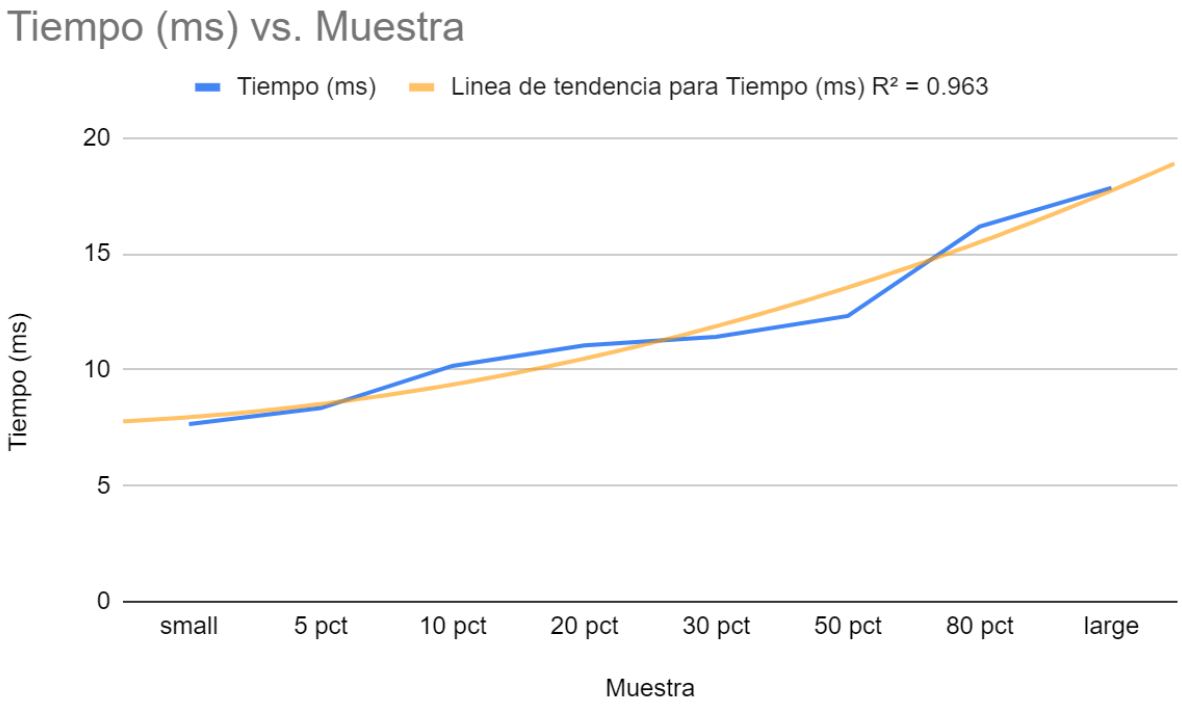
Procesadores	**Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 11 Single Language

Tablas de datos

Muestra	**Salida**	**Tiempo (ms)**
small	Tabla de sector económico con los mayores ingresos netos. Tabla de subsector económico que aportó lo mínimo, y tabla de subsector económico que aportó lo maximo	7.66
5 pct		8.35
10 pct		10.17
20 pct		11.05
30 pct		11.42
50 pct		12.32
80 pct		16.18
large		17.85

Gráficas


Tiempo (ms) vs. Muestra
Tiempo (ms)
Linea de tendencia para Tiempo (ms) R² = 0.926
20
15
10
5
0
Tiempo (ms)
small
5 pct
10 pct
20 pct
30 pct
50 pct
80 pct
large
Muestra


Tiempo (ms) vs. Muestra
Tiempo (ms)
Linea de tendencia para Tiempo (ms) R² = 0.963
20
15
10
5
0
Tiempo (ms)
small
5 pct
10 pct
20 pct
30 pct
50 pct
80 pct
large
Muestra

Análisis

En este requerimiento las líneas de tendencia se adaptan relativamente bien a un comportamiento lineal O(n) y cuadrático O(n^2) por lo que se puede concluir que el comportamiento exacto está entre estas dos líneas de tendencia.

Requerimiento 7

Descripción

Este requerimiento se encarga de reportar los accidentes más temprano y más tarde para cada día de un mes y año dado, y graficar el histograma de frecuencias de accidentes por hora para ese mismo mes y año.

Entrada	control: El catálogo de datos completo, mes: Mes a consultar, anio: Año a consultar
Salidas	Lista con datos a tabular y lista con datos para graficar.
Implementado (Sí/No)	Si, Carlos Peña

Análisis de complejidad

Pasos	**Complejidad**
Se obtiene el último día del mes para dicho mes en dicho año. Obtenemos los valores para este mes con values() en forma de una lista filtrada (L).	O(n)
Para todas las listas de accidentes en la lista filtrada (L) se ordena con mergesort por 'HORA_OCURRENCIA_ACC' ascendentemente todos sus elementos (A). Para cada lista ordenada se recorren todos sus accidentes (A), y se guarda la hora de ocurrencia en la lista con datos para graficar. Adicionalmente por cada lista se obtiene su primer y último elemento y se guarda la información pertinente en la lista con datos a tabular.	O(L * A log(A) A)
TOTAL	***O(n) + O(L * A² log(A))***

Pruebas Realizadas

Los parámetros de entrada son: catálogo, 12 (mes), 2019 (anio). Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones:

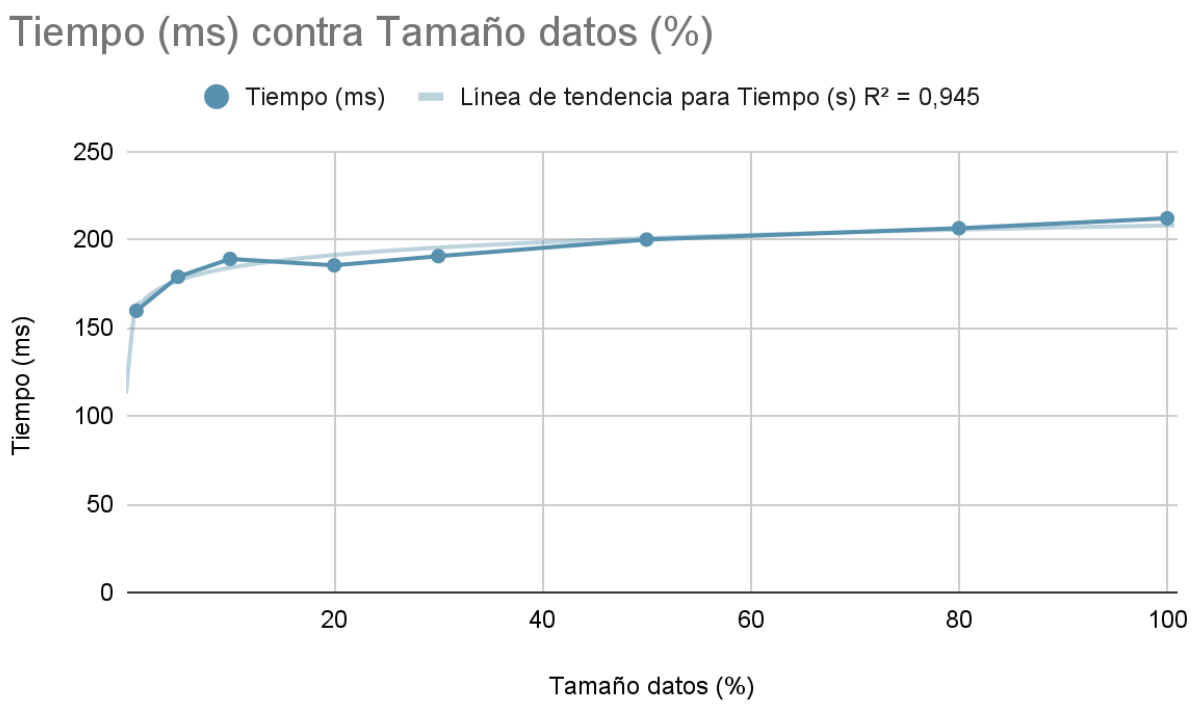
Procesadores	**AMD Ryzen 7 5700G 3.80 GHz (8 cores 16 threads)**
Memoria RAM	32 GB 3600 MHz DDR4

Sistema Operativo	Windows 10 Home - 21H2

Tablas de datos

Tamaño datos (%)	Tiempo (ms)
1	159.68
5	178.93
10	189.07
20	185.43
30	190.63
50	200.05
80	206.48
100	212.11

Gráfica



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad logarítmica aproximada $O(n) + O(L * A^2 \log(A))$. Esto se debe a que la complejidad que más aporta al tiempo es ordenar los datos.

Vemos que (L) se mantiene constante, pues son el mismo número de días para el mes de diciembre del 2019 en todos los tamaños de datos y la variación se da en el número de accidentes (A).

Requerimiento 8

Descripción

Este requerimiento se encarga de visualizar todos los accidentes de una clase particular para un rango de fechas en el mapa de Bogotá

Entrada	Control: El catálogo de datos completo, la fecha inicial y final (en formato aaaa/mm/dd), la clase del accidente que se quiere consultar
Salidas	En número total de accidentes que ocurrieron, un mapa interactivo mostrandolos.
Implementado (Sí/No)	Si, Tomas Sierra

Análisis de complejidad

Pasos	**Complejidad**
Tomar los valores del árbol	O(d)
Recorrer todas las listas de dia	O(d)
Recorrer la lista completa de dia	O(acc)
Construir el mapa	O(m)
TOTAL	***O(d) + O(d*acc) + O(m)***

Pruebas Realizadas

Los parámetros de entrada son: catálogo, entre las fechas de 01/11/2016 y el 08/11/2018. Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones:

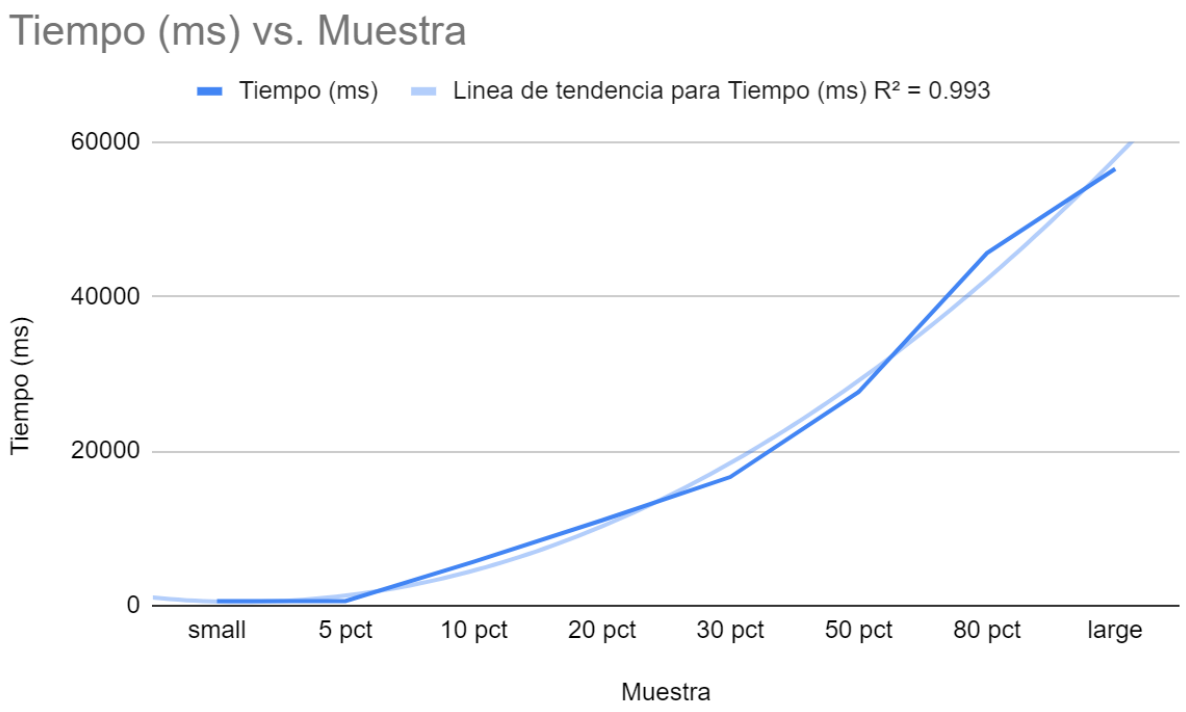
Procesadores	**Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 11 Single Language

Tablas de datos:

Muestra	**Salida**	**Tiempo (ms)**
small	En número total de accidentes que ocurrieron, un mapa	596.51
5 pct		602.87

10 pct	interactivo mostrandolos.	5671.58
20 pct		11041.13
30 pct		16649.17
50 pct		27639.97
80 pct		45605.27
large		56511.12

Gráfica



Análisis

Podemos observar que la gráfica tiene una tendencia cuadrática, en primer vistazo se podría pensar que la complejidad es O(n^2) sin embargo creemos que la gran mayoría del tiempo se gasta construyendo el mapa y no tanto recorriendo la lista para elegir que datos mostrar.

Carga de datos

Descripción

Este requerimiento se encarga de cargar los datos

Análisis de complejidad

Pasos	Complejidad
Recorrer todas las filas del csv	O(acc)
Insertar cada lista en la estructura de datos	O(log(acc))
TOTAL	***O(acc*log(acc))***

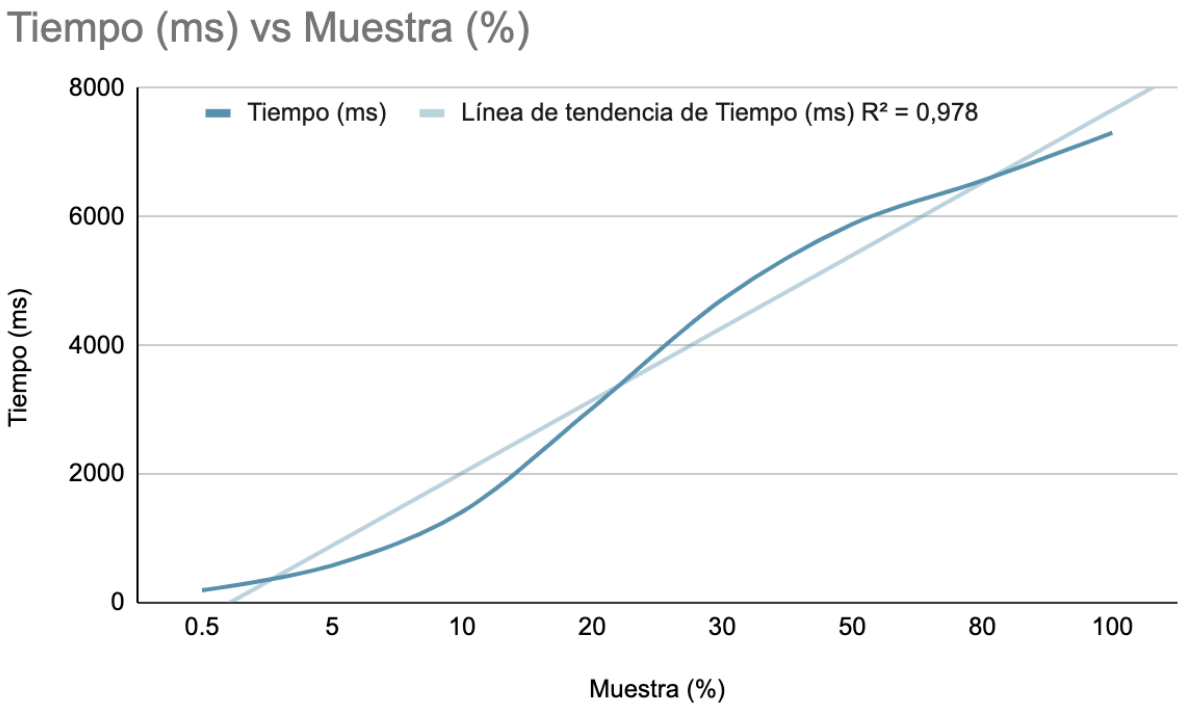
Pruebas Realizadas, Tablas de datos y Gráficas.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	MacOS Catalina Versión 10.15.7

Tablas de datos:

Muestra	Salida	Tiempo (ms)
small	En número total de accidentes que ocurrieron, un mapa interactivo mostrandolos.	193,46
5 pct		579,46
10 pct		1412,9
20 pct		3023,81
30 pct		4715,58
50 pct		5886,89
80 pct		6563,71
large		7308,18

Gráfica



Análisis

Podemos observar en la práctica un comportamiento aproximadamente lineal es decir O(n). Esto tiene sentido sin embargo como la complejidad ideal teórica es de O(n*log(n)) podemos ver como ese remanente de la dispersión de los datos se podría ajustar a una gráfica de esa forma.