

ANÁLISIS DEL RETO

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 listar la actividad económica con mayor saldo a pagar para todos los años disponibles. Y las retorna en una tabla.

Entrada	Control: El catálogo de datos completo
Salidas	Lista de la actividad económica con el mayor saldo por cada año
Implementado (Sí/No)	Si. Tomás Sierra

Análisis de complejidad

Pasos	Complejidad
Declarar variables auxiliares	$O(1)$
Recorrer por todos los años disponibles	$O(a)$
Recorrer cada elemento para guardar el año mayor	$O(n)$
TOTAL	$O(n) + O(a)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

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

Entrada	Tiempo (ms)
small	15.35

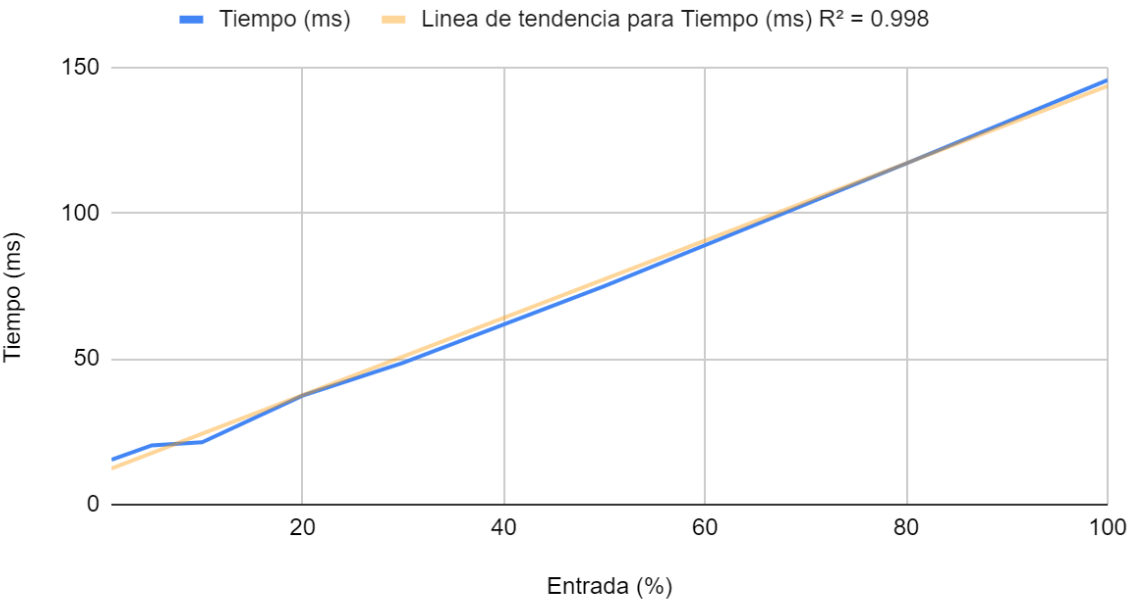
5 pct	20.32
10 pct	21.38
20 pct	37.41
30 pct	48.65
50 pct	74.96
80 pct	117.01
large	145.62

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Lista de la actividad económica con el mayor saldo por cada año	15.35
5 pct		20.32
10 pct		21.38
20 pct		37.41
30 pct		48.65
50 pct		74.96
80 pct		117.01
large		145.62

Gráficas

Tiempo (ms) vs. Entrada (%)



Análisis

Se puede observar que en la práctica la implementación tiene una complejidad $O(n)$, esto se puede deber a que aunque se recorren todos los años en nuestro caso esos siempre se mantienen constantes sin importar el tamaño de datos ya que las muestras van desde 2012 hasta 2021 por lo que la complejidad del primer for se mantiene constante y se vuelve un comportamiento lineal.

Requerimiento 2

Descripción

Este requerimiento se encarga de listar la actividad económica con mayor total saldo a favor para todos los años disponibles. Y los retorna en una tabla con cada año.

Entrada	Control: El catálogo de datos completo
Salidas	Lista de la actividad económica con el mayor saldo por cada año
Implementado (Sí/No)	Si. Tomás Sierra

Análisis de complejidad

Pasos	Complejidad
Declarar variables auxiliares	$O(1)$
Recorrer por todos los años disponibles	$O(a)$
Recorrer cada elemento para guardar el año mayor	$O(n)$
TOTAL	$O(n) + O(a)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

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

Entrada	Tiempo (ms)
small	18.73
5 pct	19.29
10 pct	23.38
20 pct	37.76

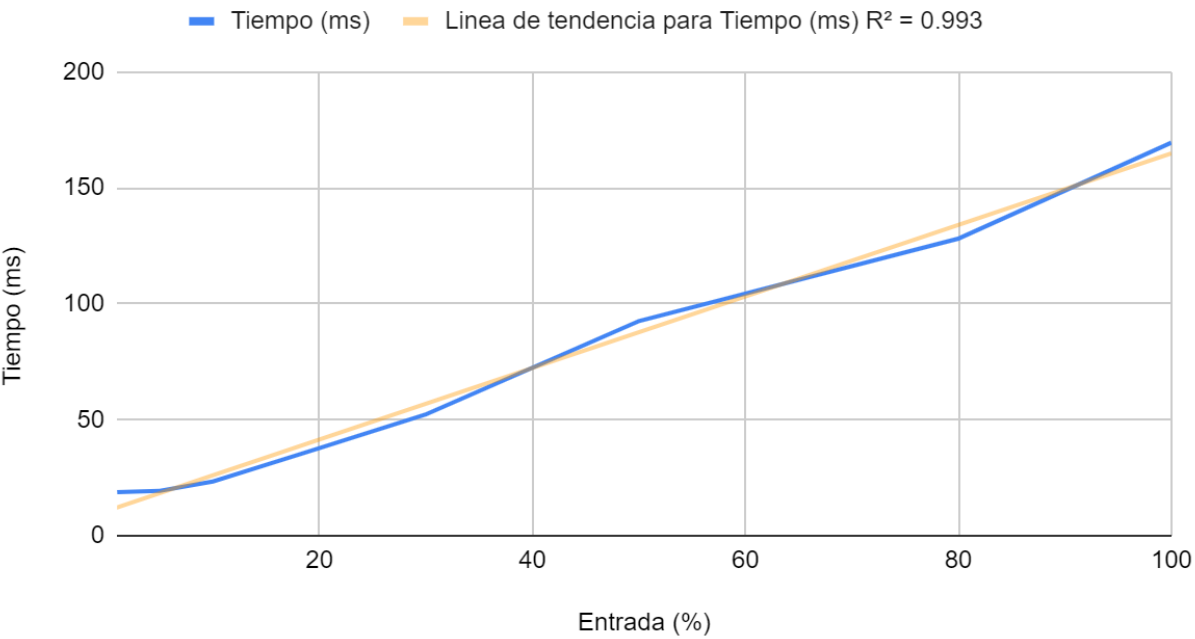
30 pct	52.28
50 pct	92.58
80 pct	128.1
large	169.49

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Lista de la actividad económica con el mayor saldo por cada año	18.73
5 pct		19.29
10 pct		23.38
20 pct		37.76
30 pct		52.28
50 pct		92.58
80 pct		128.10
large		169.49

Gráficas

Tiempo (ms) vs. Entrada (%)



Análisis

Se puede observar que en la práctica la implementación tiene una complejidad $O(n)$, esto muy similar al requerimiento 1 se puede deber a que aunque se recorren todos los años en nuestro caso esos siempre se mantienen constantes sin importar el tamaño de datos ya que las muestras van desde 2012 hasta 2021 por lo que la complejidad del primer for se mantiene constante.

Requerimiento 3

Descripción

Este requerimiento se encarga de encontrar el subsector económico con las menores retenciones para cada año

Entrada	Catálogo con todos los datos
Salidas	Una tabla general del sub sector económico que más retenciones tuvo en el año. Una tabla por cada año donde se ve las actividades que más y menos aportaron a dichas retenciones
Implementado (Sí/No)	Sí, Tomás Sierra

Análisis de complejidad

Pasos	Complejidad
Procesar la lista completa para conseguir las sublistas de años y llamar la función procesar año. Luego retornar lista de tuplas	$O(n)$
Procesar la sub lista de cada año para devolver la información requerida. Retornar tuplas	$O(a)$
Procesar la lista de tuplas para imprimir	$O(t)$
TOTAL	$O(n) + O(a) + O(t)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

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

Entrada	Tiempo (ms)
---------	-------------

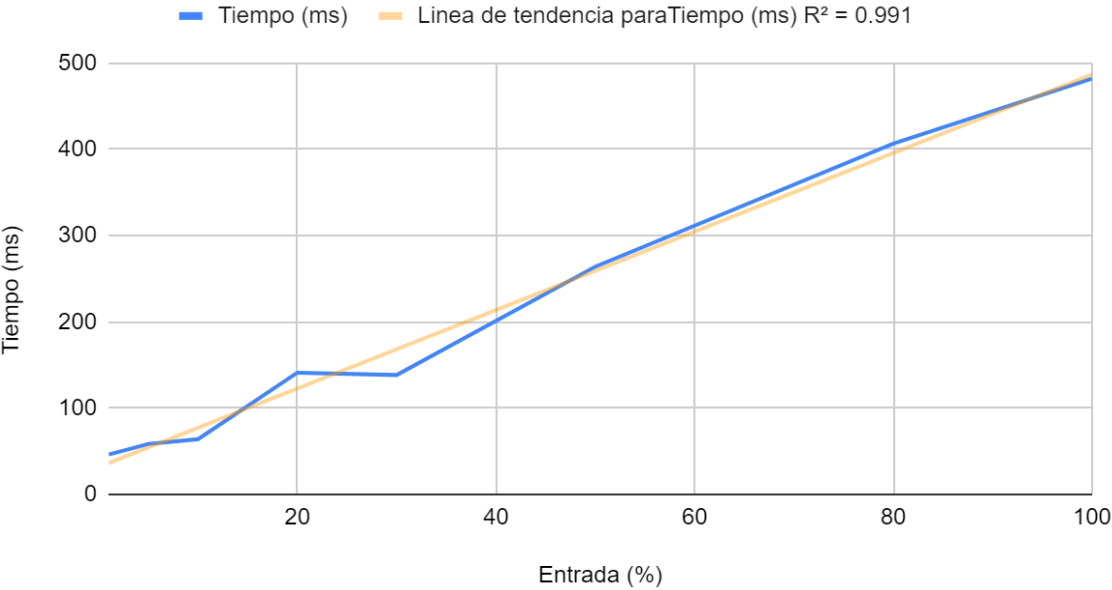
small	46.3
5 pct	58.49
10 pct	64.03
20 pct	141.16
30 pct	138.38
50 pct	264.04
80 pct	406.77
large	481.97

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Una tabla general del sub sector económico que más retenciones tuvo en el año. Una tabla por cada año donde se ve las actividades que más y menos aportaron a dichas retenciones	26.43
5 pct		34.66
10 pct		30.18
20 pct		53.76
30 pct		79.94
50 pct		97.47
80 pct		152.32
large		178.51

Gráficas

Tiempo (ms) vs. Entrada (%)



Análisis

Se puede observar que la curva se apega a un comportamiento lineal más que todo, esto pienso que se debe a que como tanto a como t son en realidad sublistas de la original. La complejidad temporal va a estar dada por n que es el número de datos que ingresan a la función.

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
Salidas	Una tabla general del subsector económico con los mayores costos y gastos de nómina del año. Una tabla por cada año donde se ve las actividades que más y menos aportaron a dichos costos y gastos de nómina.
Implementado (Sí/No)	Sí, Juan Francisco Rodríguez Contreras

Análisis de complejidad

Pasos	Complejidad
Procesar el catálogo de datos completos para conseguir las sublistas de cada año.	$O(n)$
Procesar la sublista de cada año para devolver la información requerida y retornar tuplas con tabla general del subsector económico con los mayores costos, una tabla por cada año donde se ve las actividades que más y menos aportaron a dichos costos y gastos de nómina y gastos de nómina tuvo en el año y los títulos de cada información para proyectarlos en el view.	$O(a)$
Procesar la lista de tuplas para imprimir en el view.	$O(t)$
TOTAL	$O(n) + O(a) + O(t)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardado como un Arraylist y el algoritmo de ordenamiento usado fue Merge Sort.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
--------------	--------------------------------------

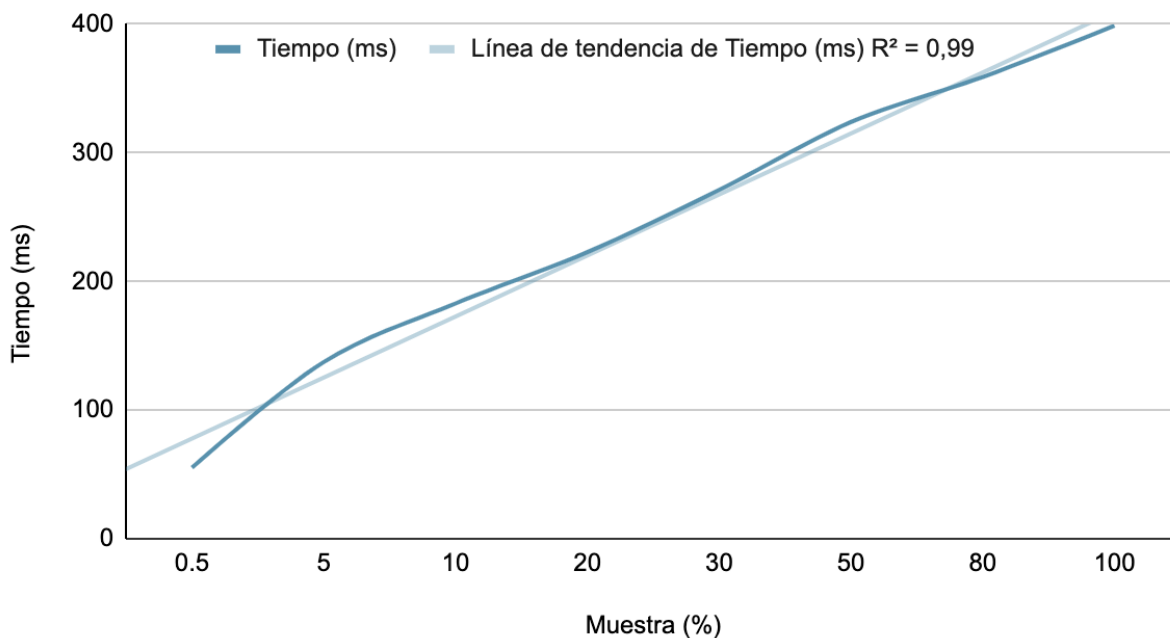
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	MacOS Catalina Versión 10.15.7

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	El subsector económico que tuvo los mayores costos y gastos de nómina para cada año disponible en los datos cargados.	55.2
5 pct		137.4
10 pct		182.93
20 pct		222.78
30 pct		271.18
50 pct		323.96
80 pct		359.02
large		398.89

Gráficas

Tiempo (ms) vs Muestra (%)



Análisis

Viendo el resultado que nos arroja la gráfica se puede notar una tendencia lineal la cual es el fruto de que a la hora de resolver el requerimiento se utilizó una complejidad en casi todo momento de $O(n)$ al estar creando sublistas de una lista principal.

Requerimiento 5

Descripción

Este requerimiento se encarga de listar el subsector económico con los mayores descuentos tributarios para todos los años disponibles.

Entrada	Control: El catálogo de datos completo
Salidas	Lista del subsector económico con los mayores descuentos tributarios para todos los años disponibles. En este, las tres actividades económicas que menos aportaron y las tres actividades económicas que más aportaron al descuento tributario en cada año.
Implementado (Sí/No)	Sí, Carlos Peña

Análisis de complejidad

Pasos	Complejidad
Agrupar datos (Por sector, subsector económico, y las actividades que aportaron 'Descuentos tributarios' a dicho sector con su respectivo subsector). Recorre todos los datos, luego recorre todos los datos en la lista agrupada "g". (Primero se itera sobre todos los datos de la lista original, después se itera dos veces sobre la lista nueva para comprobar si el año, el sector y subsector económico ya se encuentran listados.)	$O(n^2g)$
Ordena para cada uno de los elementos en la lista de los subsectores "s" descendientemente por su 'El total de descuentos tributarios del subsector económico'. Adicionalmente ordena para cada uno de los elementos de la lista de actividades "a", para el primer elemento de de cada lista ordenada "s" de subsectores, las actividades ascendientemente por su 'Total descuentos tributarios'. (Primero se itera sobre todos los datos de la lista filtrada, luego se ordena con quicksort dos veces, primero para los subsectores económicos, posteriormente se ordena para las actividades en el subsector que más descuentos totales tiene.)	$O(g * [s \log(s) + a \log(a)])$
TOTAL	$O(n^2g) + O(g * [s \log(s) + a \log(a)])$

Pruebas Realizadas

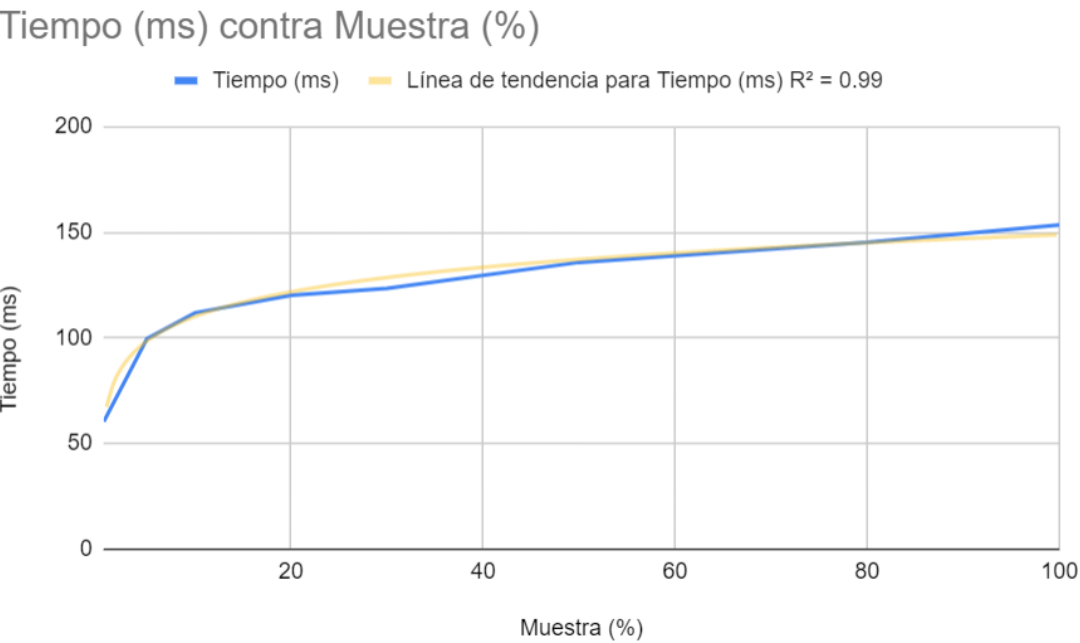
Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	Mac OS Monterrey Versión 12.6.3

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Los subsectores económicos con los mayores descuentos tributarios disponibles.	60.38
5 pct		99.64
10 pct		111.92
20 pct		120.22
30 pct		123.5
50 pct		135.8
80 pct		145.22
large		153.56

Gráficas



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad aproximada $O(n \log(n))$. Esto se debe a que la complejidad que más aporta al tiempo es el ordenamiento de los datos. Pues filtrarlos y agruparlos no requiere de tantos recursos como ordenarlos en este caso.

Requerimiento 6

Descripción

Este requerimiento se encarga de listar la actividad económica con el mayor total de ingresos netos (Total ingresos netos) para cada sector económico en un año específico.

Entrada	Control: El catálogo de datos completo, anio: Año a analizar
Salidas	Lista de la actividad económica con el mayor total de ingresos netos (Total ingresos netos) para cada sector económico en un año específico ordenada de menor a mayor por el código del sector económico.
Implementado (Sí/No)	Sí, Carlos Peña

Análisis de complejidad

Pasos	Complejidad
Recorrer todos los datos y filtrar para obtener solo los datos para el año deseado.	$O(n)$
Agrupar datos filtrados (Por sector, subsector económico, y las actividades que aportaron 'Total ingresos netos' a dicho subsector). Recorre todos los datos en la lista filtrada "f", luego recorre todos los datos en la lista agrupada "g" 2 veces para comprobar si el sector y el subsector ya se encuentran.	$O(f*2g)$
Ordenar la lista ascendentemente por el código del sector económico. Recorre todos los datos en la lista agrupada "g".	$O(g \log(g))$
Ordena descendentemente cada uno de los elementos de la lista de subsectores "s" en la lista agrupada ordenada "g", descendentemente por su 'Total ingresos netos del subsector económico'. Adicionalmente ordena para cada uno de los elementos de la lista de actividades "a", para el primer y último elemento para cada lista ordenada "s" de subsectores, las actividades descendentemente por su 'Total ingreso neto' atribuido al subsector.	$O(g * (s \log(s) + 2a \log(a)))$
TOTAL	$O(n) + O(f*2g) + O(g \log(g)) + O(g * [s \log(s) + 2a \log(a)])$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	Mac OS Monterrey Versión 12.6.3

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	La actividad económica de cada uno de los sectores económicos que tuvieron los mayores ingresos netos en el año 2021	85.64
5 pct		136.7
10 pct		198.07
20 pct		213.59
30 pct		238.94
50 pct		302.94
80 pct		283.19
large		318.15

Gráficas

Tiempo (ms) contra Muestra (%)



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad aproximada $O(n \log(n))$. Esto se debe a que la complejidad que más aporta al tiempo es el ordenamiento de los datos. Pues filtrarlos y agruparlos no requiere de tantos recursos como ordenarlos en este caso.

Requerimiento 7

Descripción

Este requerimiento se encarga de listar las actividades económicas que tuvieron los menores valores totales de costos y gastos (Total costos y gastos), para un periodo de tiempo específico.

Entrada	Control: El catálogo de datos completo, n: El número (N) de actividades económicas a identificar, bi: Año de inicio, bs: Año de fin
Salidas	Lista del año, sector económico y su subsector económico con el menor totales de costos y gastos, organizado descendientemente en años y ascendientemente en totales de costos y gastos por año.
Implementado (Sí/No)	Sí, Carlos Peña

Análisis de complejidad

Pasos	Complejidad
Recorrer todos los datos y filtrar para obtener solo los datos para el periodo en años deseado.	$O(n)$
Ordenar la lista filtrada "f" ascendientemente por 'Total costos y gastos'	$O(f \log(f))$
Segundo filtro que obtiene el número de elementos "N" a identificar de la lista "f" y los guarda en una lista "t".	$O(N)$
Ordena la lista filtrada "t" de manera estable por años descendientemente.	$O(t)$
TOTAL	$O(n) + O(f \log(f)) + O(N) + O(t)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	Mac OS Monterrey Versión 12.6.3

Tablas de datos

Muestra	Salida	Tiempo (ms)
small	Las 10 actividades económicas con menores costos y gastos totales para el	26.43
5 pct		34.66
10 pct		30.18

20 pct	periodo comprendido entre 2014 y 2021.	53.76
30 pct		79.94
50 pct		97.47
80 pct		152.32
large		178.51

Gráficas

Tiempo (ms) contra Muestra (%)



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad aproximada $O(n)$. Esto se debe a que la complejidad que más aporta al tiempo es el filtrado de datos. Pues ordenarlos no requiere de tantos recursos como filtrarlos en este caso.

Requerimiento 8

Descripción

Este requerimiento se encarga de listar el Top (N) de actividades económicas de cada subsector económico con los mayores totales de impuestos a cargo para un periodo de tiempo

Entrada	Control: El catálogo de datos completo, n: El número (N) de actividades económicas a identificar, bi: Año de inicio, bs: Año de fin
Salidas	Lista de sector económico y su subsector económico que más aportó 'Total de impuestos a cargo para el subsector', adicionalmente las N actividades económicas que más aportaron a este subsector.
Implementado (Sí/No)	Sí, Carlos Peña

Análisis de complejidad

Pasos	Complejidad
Recorrer todos los datos y filtrar para obtener solo los datos para el periodo en años deseado.	$O(n)$
Agrupar datos filtrados (Por sector y subsector económico, y las actividades que aportaron a dicho subsector). Recorre todos los datos en la lista filtrada "f", luego recorre todos los datos en la lista agrupada "g".	$O(f * g)$
Ordenar la lista alfabéticamente por el nombre del subsector Recorre todos los datos en la lista agrupada "g".	$O(g \log(g))$
Ordena para cada uno de los elementos de la lista de actividades "a" en la lista agrupada ordenada "g", las actividades descendientemente por su 'Total Impuesto a cargo' atribuido al subsector.	$O(g * a \log(a))$
TOTAL	$O(n) + O(f * g) + O(g \log(g)) + O(g * a \log(a))$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una máquina con las siguientes especificaciones. Los datos de entrada fueron el catálogo guardados como un Arraylist y el algoritmo de ordenamiento usado fue Quick Sort.

Procesadores	1,8 GHz Intel Core i5 de dos núcleos
Memoria RAM	8 GB 1600 MHz DDR3
Sistema Operativo	Mac OS Monterrey Versión 12.6.3

Tablas de datos

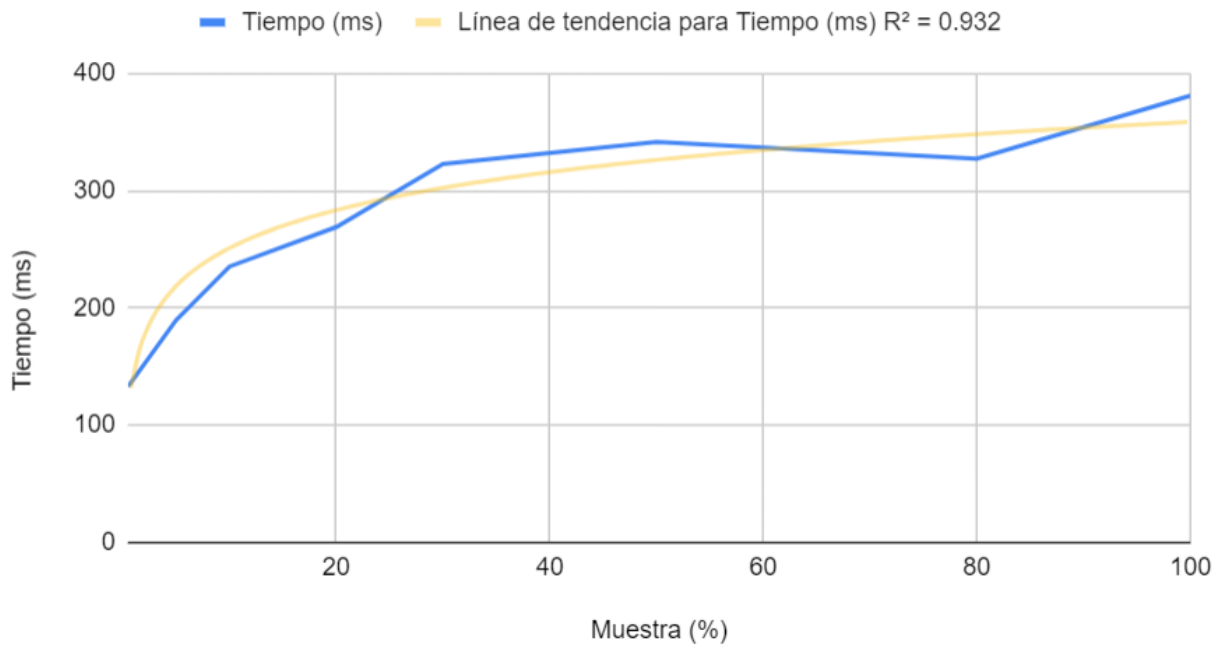
Muestra	Salida	Tiempo (ms)
small	Las 5 actividades económicas que aportaron el mayor impuesto a cargo, de cada subsector, para el periodo	133.07
5 pct		189.72
10 pct		235.35
20 pct		269.04

subsector, para el periodo

30 pct	comprendido entre 2014 y 2021.	322.85
50 pct		341.79
80 pct		327.46
large		381.11

Gráficas

Tiempo (ms) contra Muestra (%)



Análisis

Se puede observar que en la práctica el requerimiento tiene una complejidad aproximada $O(n \log(n))$. Esto se debe a que la complejidad que más aporta al tiempo es el ordenamiento de los datos. Pues filtrarlos y agruparlos no requiere de tantos recursos como ordenarlos en este caso.

Carga de datos

Descripción

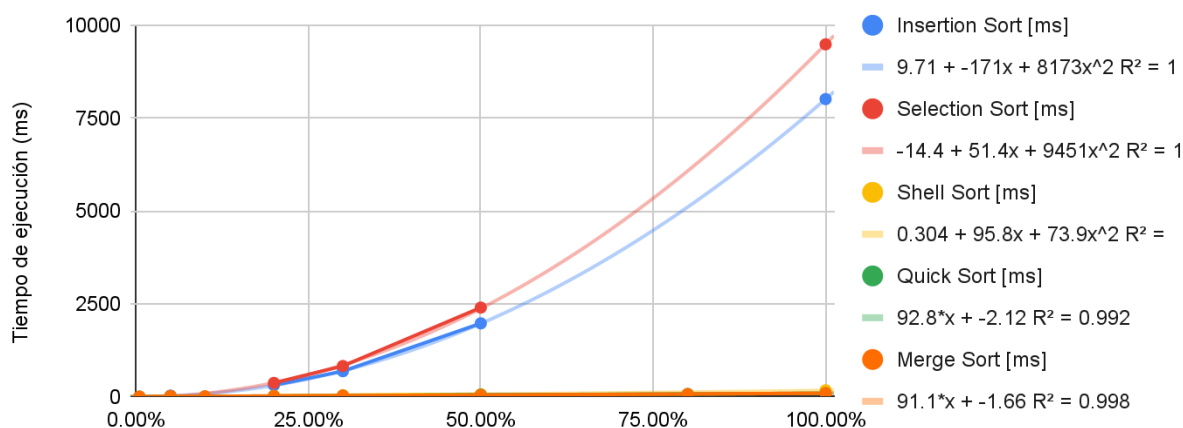
Este requerimiento se encarga de cargar los datos y asignarles un ID único en la base de datos.

Pruebas Realizadas, Tablas de datos y Gráficas.

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

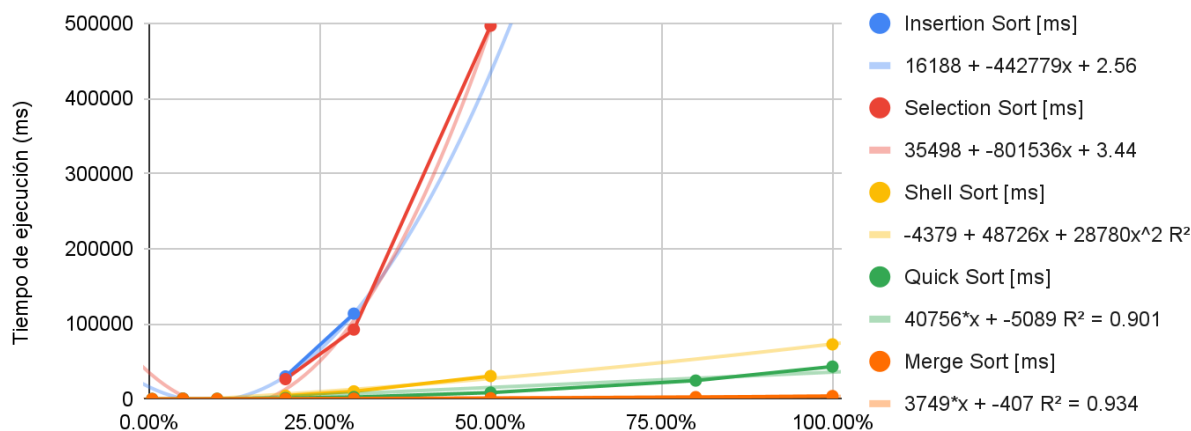
Tabla 1. Especificaciones de la máquina para ejecutar las pruebas de rendimiento.

Algoritmos (ms) vs. Tamaño Muestra (%) M1 (AL)



Gráfica 1. Tiempo de ejecución de algoritmos de ordenamiento contra tamaño de muestra en % para ArrayList.

Algoritmos (ms) vs. Tamaño Muestra (%) M1 (SL)



Gráfica 2. Tiempo de ejecución de algoritmos de ordenamiento contra tamaño de muestra en % para ArrayList.

Para ver las tablas de datos sugerimos visitar el siguiente vinculo e ir a la pestaña Lab5-M1 <https://docs.google.com/spreadsheets/d/1U1a-tytwRgBI05hb0W4QZb5vzWjm11AkofJY145OtUs/edit?usp=sharing>.

Análisis de complejidad sobre algoritmos de ordenamiento y carga de datos.

Los algoritmos se comportan acorde a lo establecido teóricamente ya que para los casos cuando el algoritmo tiene una complejidad teórica de $O(n^2)$ las gráficas mostraron una correlación muy precisa con ese comportamiento cuadrático. Dadas las limitaciones para regresiones en la herramienta en que las llevamos a cabo, no pudimos realizar una regresión de acorde a lo teórico en los algoritmos Quick, Merge y Shell sort. A pesar de esto podemos evidenciar que el comportamiento visual es de acorde a lo teórico. Durante la toma de datos no se vio alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas. Esto debido a que las máquinas que tenían un menor poder de procesamiento se demoran más tiempo en general en procesar los datos. Sin embargo, esto no significaba que el comportamiento era diferente, la variación se generaba en el tiempo de ejecución, mas no en como se comportaba del mismo. De las gráficas mostradas anteriormente (Gráfica 1 y Gráfica 2) podemos ver que teniendo en cuenta el tiempo de ejecución, la mejor estructura de datos es el ARRAY LIST ya que este fue el que menos tiempo se demoró sin importar el algoritmo de ordenamiento o la máquina. Adicionalmente, el algoritmo de ordenamiento preferido para el reto varió entre Mergesort (cuando era necesario un algoritmo estable que mantuviera el orden de entrada para datos iguales) y Quicksort en casos en que el uso eficiente de la memoria de la máquina prevalecía. Esto debido al análisis obtenido de la carga de datos, en la cual presenciamos una diferencia abismal entre los algoritmos de ordenamiento iterativos y los recursivos (exceptuando a Shellsort).