

ANÁLISIS DEL RETO

Jaime Esteban Moreno Jaramillo, je.morenoj1@uniandes.edu.co, 202220189

David Santiago Carrillo Vargas, d.carrillov@uniandes.edu.co, 202225276

Juan Diego Sánchez, jd.sanchez@uniandes.edu.co, 202214625

Requerimiento <<Carga de Datos>>

Descripción

Pensamos nuestra carga de datos con un gran mapa de hash con tantas parejas llave valor como años en el archivo. La llave de estas parejas es el año, y su valor son otras tres parejas llave-valor; respectivamente: “elements”-Arraylist (con todos los registros del año), “subsector”-mapa (Lo llamaremos mapa 2) y “sector”-otro mapa (mapa 3). El mapa 2 contiene parejas cuya llave es el código del subsector y el valor es otro mapa, el cual contiene 8 parejas llave-valor, el primero “elements”-Arraylist (con todos los datos del subsector correspondiente) y otras 7 parejas que tienen las sumatorias necesarias para los requerimientos individuales. Por otro lado, el mapa 3 tiene como llave el código del sector, y como valor un mapa con dos parejas llave-valor “elements”-Arraylist (con todos los elementos del sector) y la otra es una sumatoria para los requerimientos básicos.

```
424
425 def add_data(data_structs,data,tipo_mapa,factor_carga):
426
427     if not(mp.contains(data_structs["años"],data["Año"])): # se revisa si el año existe
428         create_base(data_structs,data,tipo_mapa,factor_carga)
429     add(data_structs,data,tipo_mapa,factor_carga)
430
431     return data_structs
432
433
```

Carga de datos en el model

Entrada	La estructura de datos (control), Valor booleano sí o no se quiere conocer el uso de memoria (memflag).
Salidas	Total filas cargadas, tiempo usado en ms, y espacio usado si es que se pidió o no
Implementado (Sí/No)	Si se implementó

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Creación mapa años	$O(\#años)$
Itera el archivo y añade los datos a la estructura	$O(N(\text{complejidad añadir datos}))$
Añadir datos. (Revisa si la llave existe, si existe añade la información a las estructuras existentes. De lo	Complejidad añadir datos = $O(\#años + \#sector + \#subsector + 1)$

contrario crea estructuras para los nuevos sectores, subsectores o años). Añadir datos es $O(1)$. Crear las estructuras es $O(\# \text{datos que se ingresan})$. Al final, la magnitud de la carga de datos es $O(N)$	
TOTAL	$O(N)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

	Máquina 1	Máquina 2	Maquina 3
Procesadores	10th Gen Intel Core i7-10750H	11th Gen Intel Core i5-1135G7	AMD Ryzen 3 3250U
	8	8	12
Sistema Operativo	Windows 11	Windows 11	Windows 11

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Maquina 1

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	31393.35	1592.72
0.5	26940.70	1623.44
0.7	26622.62	1737.46
0.9	26508.56	1761.34

Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 1.

Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	27347.20	2073.56
4.00	27136.13	1659.74
6.00	27048.00	1814.40
8.00	27036.99	1818.86

Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

Maquina 2

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	30779.69	1540,23
0.5	26415,99	1598,65
0.7	26409,62	1615,63
0.9	26395,14	1675,31

Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Máquina 2.

Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	27162.96	1989.64
4.00	27106.16	1619.17
6.00	27027.46	1801,97
8.00	27011.13	1809,89

Tabla 6. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Máquina 2.

Maquina 3

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	31413.95	3882.11
0.5	26958.30	3630.17
0.7	26642.19	3702.23
0.9	26527.50	4132.12

Tabla 7. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Máquina 3.

Carga de Catálogo CHAINING

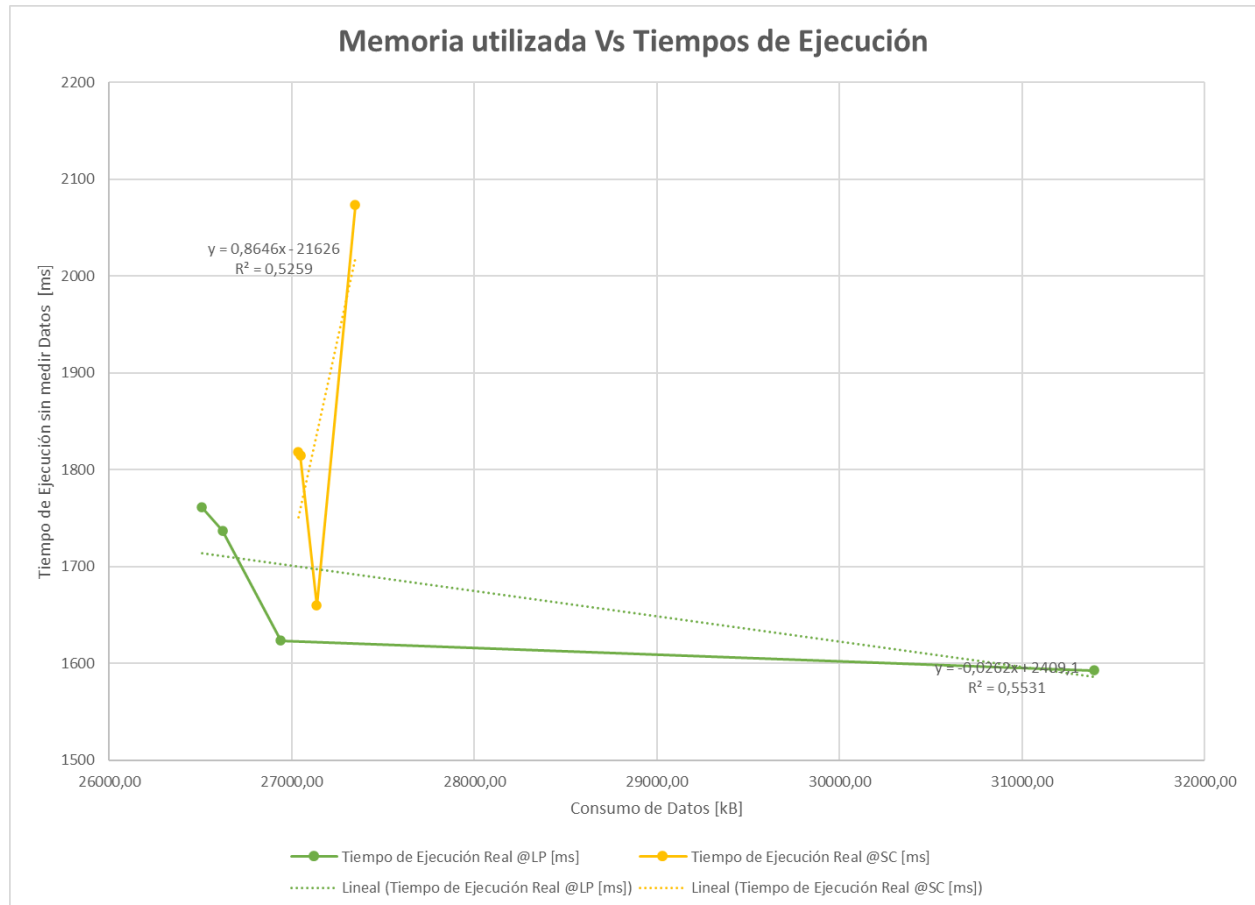
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	27337.93	3784.98
4.00	27154.17	3864.77
6.00	27066.80	4286.91
8.00	27053.58	4161.31

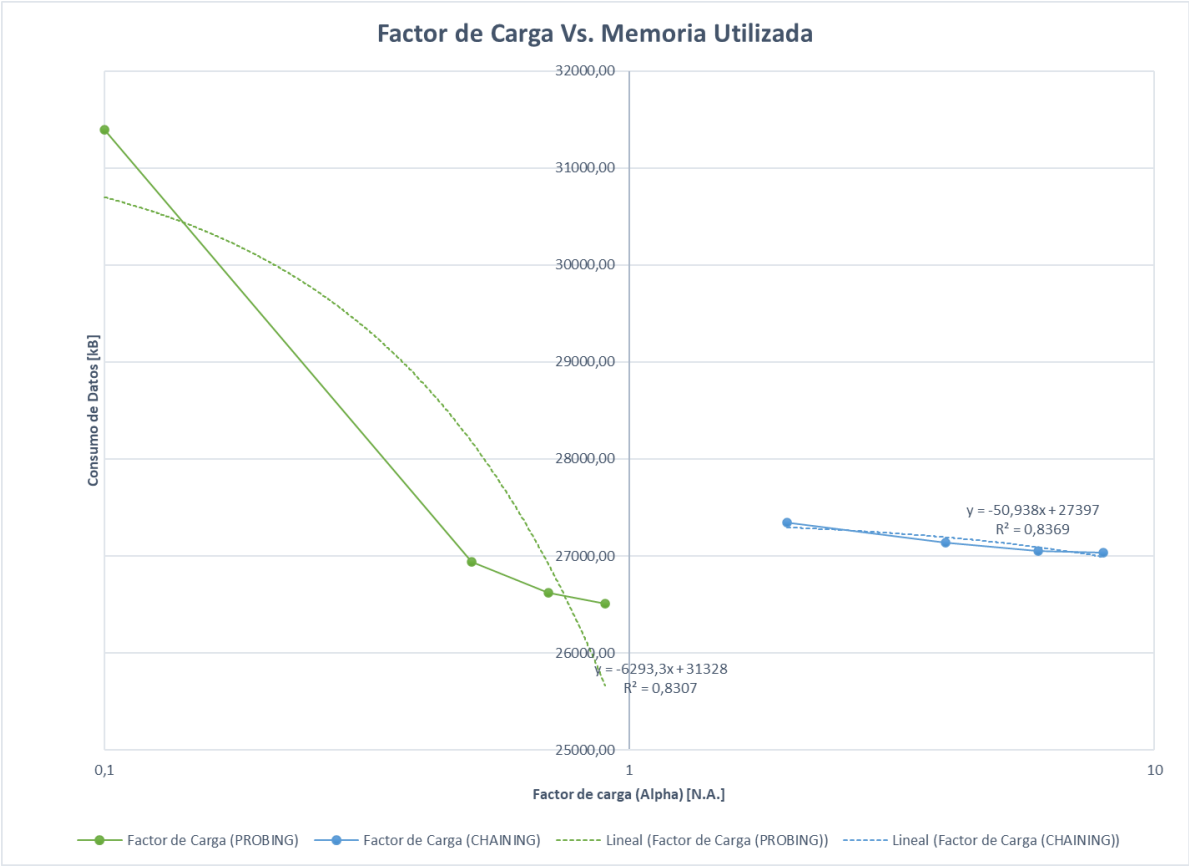
Tabla 8. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Máquina 3.

Graficas

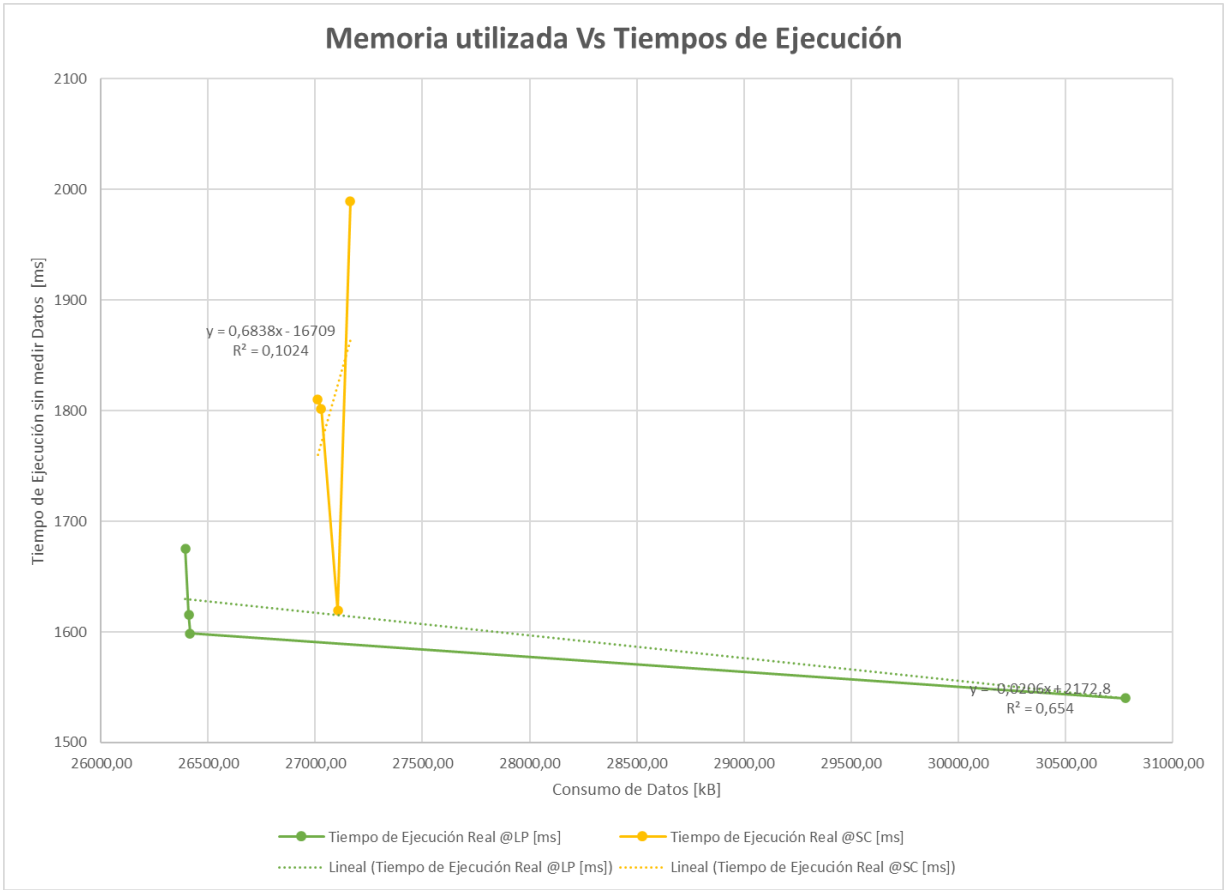
Las gráficas con la representación de las pruebas realizadas.

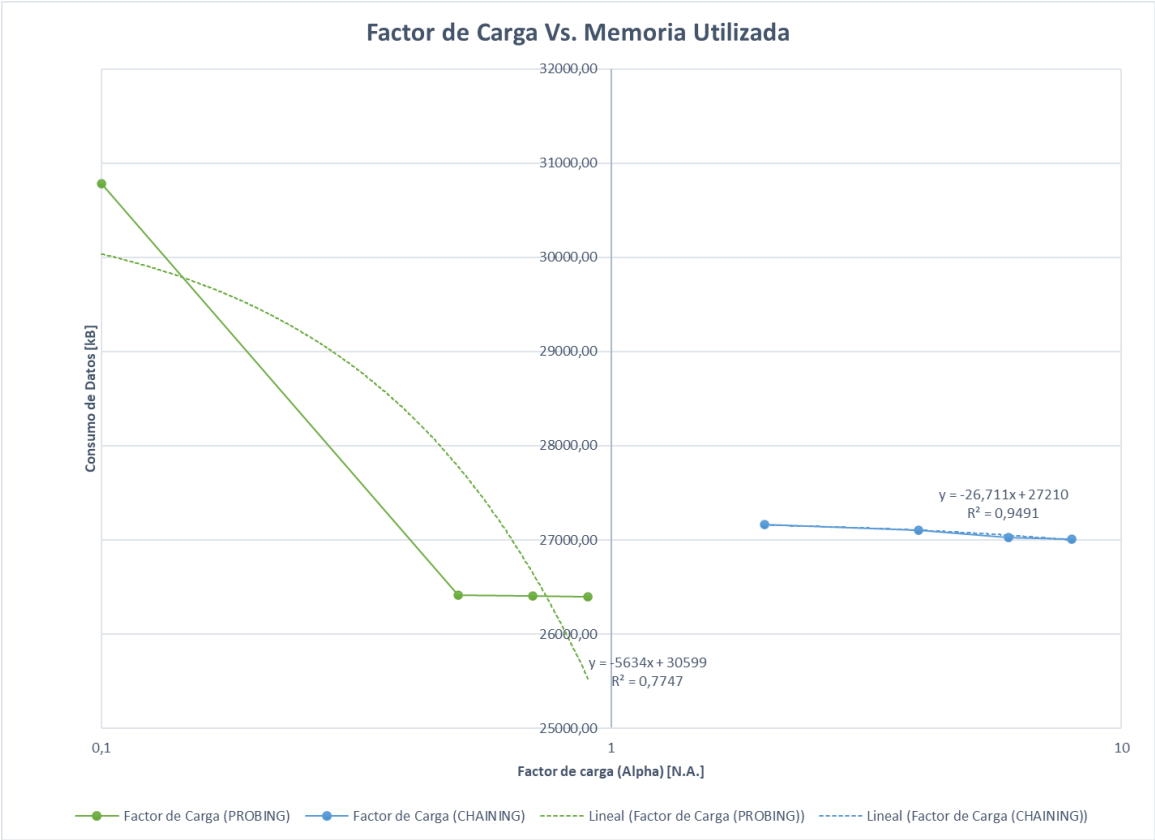
Maquina 1



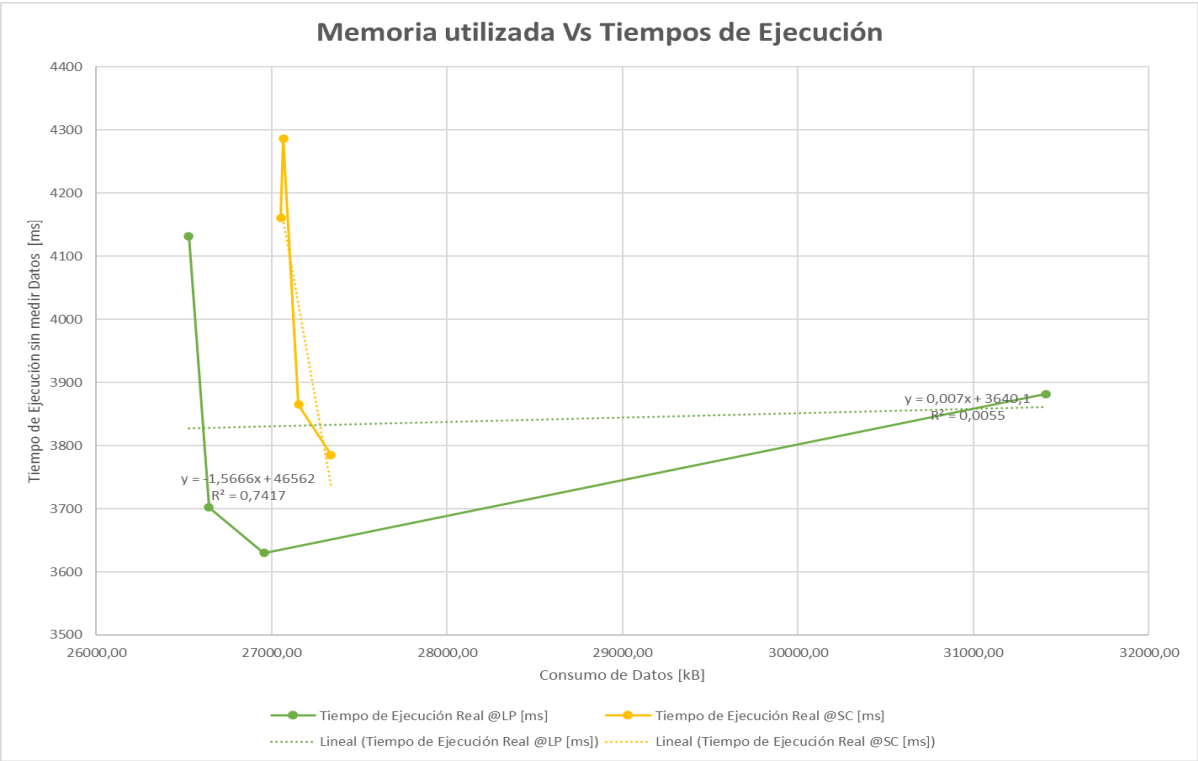


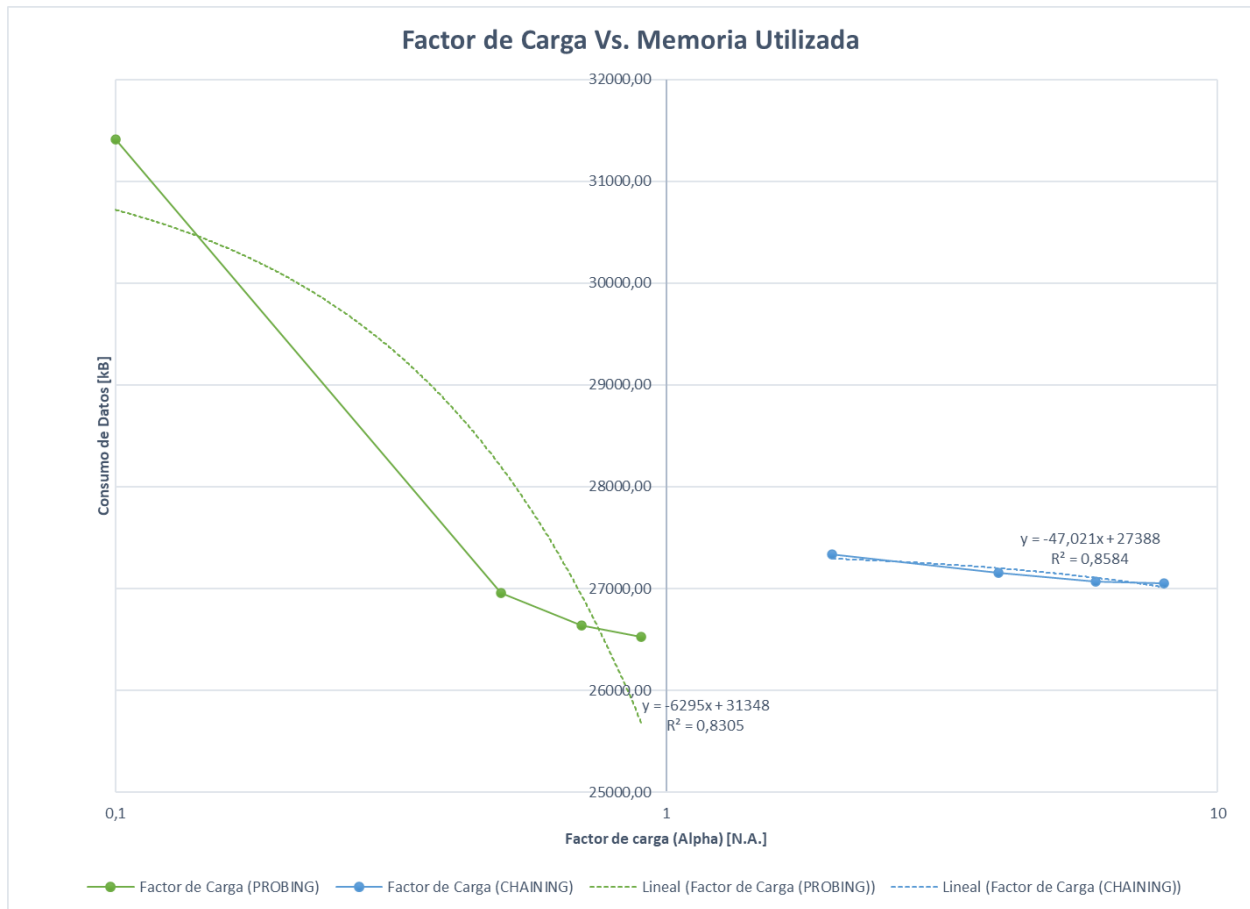
Maquina 2





Maquina 3





Análisis

En base a los resultados obtenidos en las pruebas y basados en la teoría del funcionamiento de los algoritmos linear probing y separate chaining concluimos que tomamos la decisión correcta al escoger probing como el tipo de mapa y 0.5 como el mejor tiempo de carga. Además, gracias al factor de carga acceder a datos es $O(1)$ en mapas de hash. Por esta razón decidimos hacer las sumatorias necesarias para los requerimientos directamente en el mapa. Es así, como obtenemos tiempo realmente muy bajos en la ejecución de los requerimientos.

Requerimiento << No. 1 (Grupal): Obtener la actividad económica con mayor saldo a pagar para un sector económico y un año específico.>>

Descripción

Primero creamos una lista array list vacía, y en la variable act guardamos el valor de llevar a cabo la función como tal, enviándole al controlador los datos (control), el año y el sector que se quieren conocer, el controlador le comunica la solicitud al modelo, en el modelo primero creamos la función La función `diccio_req1` la cual toma como entrada un diccionario data y devuelve un nuevo diccionario con las mismas claves pero solo los valores correspondientes a las claves que se especifican en la función. Luego la función `req_1` toma como entrada tres parámetros: `data_structs`, `anio` y `sector`, y devuelve un diccionario con la información correspondiente al elemento con el mayor valor de la clave "Total saldo a pagar" dentro del subsector económico especificado, finalmente ese diccionario es regresado al controlador y a su vez a la vista guardado en la variable act, la cual posteriormente es añadida a la lista array list inicialmente creada para ser posteriormente mostrada en pantalla.

Entrada	<ul style="list-style-type: none">• Año.• Código sector económico.• Datos (control)
Salidas	Lista con el diccionario de la actividad económica con mayor saldo a pagar para el subsector dado
Implementado (Sí/No)	Si se implementó

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener el mapa del año requerido por el usuario	$O(1)$
Obtener el mapa con todos los sectores economicos del año	$O(1)$
Obtener el mapa del sector requerido por el usuario	$O(1)$
Recorrer todas las actividades en esta lista y comparar uno por uno hasta elegir el mayor	$O(\text{\#elementos de la lista})$
Devolver el diccionario de dicha actividad	
TOTAL	$O(\text{\#elementos de la lista})$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

	Máquina 1	Máquina 2	Maquina 3
Procesadores	10th Gen Intel Core i7-10750H	11th Gen Intel Core i5-1135G7	AMD Ryzen 3 3250U
	8	8	12
Sistema Operativo	Windows 11	Windows 11	Windows 11

Entrada	Tiempo (s)
small	0.171
5 pct	0.186
10 pct	0.197
20 pct	0.233
30 pct	0.240
50 pct	0.287
80 pct	0.372
large	0.401

Entrada	Tiempo (s)
small	0.012
5 pct	0.057
10 pct	0.104
20 pct	0.206
30 pct	0.279
50 pct	0.201
80 pct	0.297
large	0.176

Las pruebas se realizaron con el año 2016 y con el sector 3

Entrada	Tiempo (ms)
small	0.476
5 pct	0.588
10 pct	0.702
20 pct	0.691
30 pct	0.878
50 pct	0.704
80 pct	2.565
large	3.061

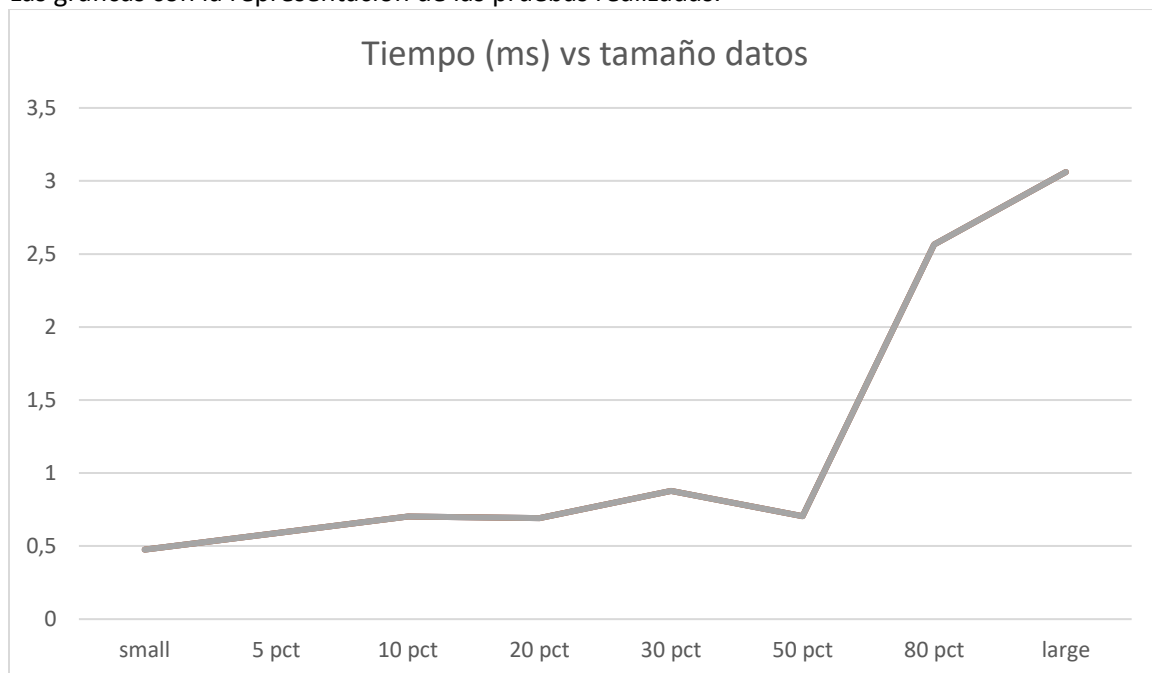
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Actividad economica: 130	0.171
5 pct	Actividad economica: 125	0.186
10 pct	Actividad económica: 125	0.197
20 pct	Actividad económica: 125	0.233
30 pct	Actividad económica: 125	0.240
50 pct	Actividad económica: 125	0.287
80 pct	Actividad económica: 125	0.372
large	Actividad económica: 125	0.401

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En base a los resultados obtenidos y el análisis de complejidad realizado podemos ver un comportamiento de $O(n)$ el cual aumenta progresivamente en relación con los datos dados.

Requerimiento << No. 2 (Grupal): Obtener la actividad económica con mayor saldo a favor para un sector económico y un año específico. >>

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	<ul style="list-style-type: none">• Año• Código sector económico• Datos (control)
Salidas	<ul style="list-style-type: none">• Código actividad económica• Nombre actividad económica• Código subsector económico• Nombre subsector económico• El total ingresos netos• El total costos y gastos• El total saldo a pagar• El total saldo a favor
Implementado (Sí/No)	Si se implementó

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener la lista con los elementos del sector consultado	$O(1)$
Se ordena con Merge esa lista	$O(n \cdot \log n)$ n =número elementos en la lista
Sacar el mayor	$O(1)$
TOTAL	$O(n \cdot \log n)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

	Máquina 1	Máquina 2	Maquina 3
Procesadores	10th Gen Intel Core i7-10750H	11th Gen Intel Core i5-1135G7	AMD Ryzen 3 3250U
	8	8	12
Sistema Operativo	Windows 11	Windows 11	Windows 11

Año 2021, Sector 3

Entrada	Tiempo (ms)
small	2.990
5 pct	4.887
10 pct	4.935
20 pct	4.994
30 pct	5.283
50 pct	5.623
80 pct	5.963
large	7.546

Los datos se tomaron con el año 2016 y sector 1

Entrada	Tiempo (ms)
small	0.227
5 pct	0.762
10 pct	2.964
20 pct	1.802
30 pct	3.134
50 pct	3.433
80 pct	4.265
large	6.402

2

Entrada	Tiempo (ms)
small	0.318
5 pct	0.577
10 pct	1.060
20 pct	1.740
30 pct	2.878

50 pct	4.032
80 pct	8.165
large	9.907

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida (Actividad económica)	Tiempo (ms)
small	130	0.227
5 pct	150	0.762
10 pct	150	2.964
20 pct	145	1.802
30 pct	145	3.134
50 pct	161	3.433
80 pct	161	4.265
large	161	6.402

Graficas

Las gráficas con la representación de las pruebas realizadas.

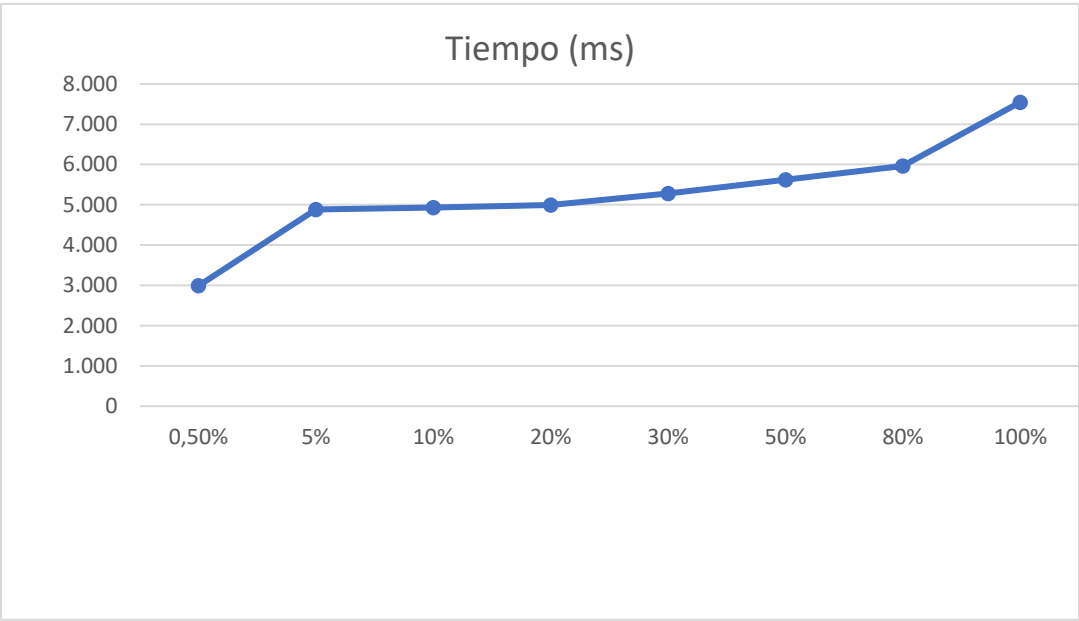
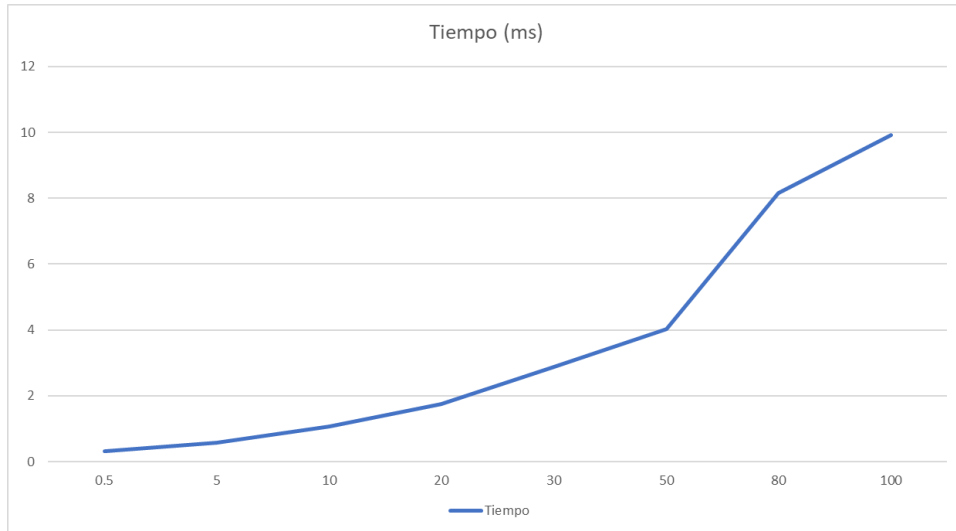


Tabla 2



Análisis

En base a los resultados obtenidos y el análisis de complejidad realizado podemos ver un comportamiento de $O(n \log(n))$ el cual aumenta en poca medida en relación con los datos dados pues el proceso de mayor complejidad temporal viene siendo el ordenamiento con merge.

Requerimiento << No. 3 (Individual): Encontrar el subsector económico con el menor total de retenciones para un año específico. >>

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Año y código del subsector
Salidas	Una tupla la cual contiene dos diccionarios, uno con los datos del mayor subsector y el otro con las 3 primeras y las 3 ultimas actividades económicas.
Implementado (Sí/No)	Si se implementó David Carrillo 202225276.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Hallar el mapa de subsectores de un año específico	$O(1)$
Recorrer todos los subsectores del año y hallar el que tiene mayores costos y gastos de nómina.	$O(S)$ $S = \text{numero de subsectores del año}$
Hallar la lista del subsector con mayores costos y gastos nómina.	$O(1)$
Ordenar la lista de actividades económicas de un subsector	$O(ns \cdot \log(ns))$ $ns = \text{actividades económicas de un subsector}$
Obtener los 3 elementos que más aportaron	$O(1)$
Obtener los 3 elementos que menos aportaron	$O(1)$
TOTAL	$O(ns \cdot \log(ns))$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Máquina 2	
Procesador	11th Gen Intel Core i5-1135G7
	8
Sistema Operativo	Windows 11

--	--

Año 2013

Entrada	Tiempo (ms)
0.5%	1.230
5%	1.312
10%	1.404
20%	1.562
30%	1.659
50%	1.835
80%	2.022
100%	2.235

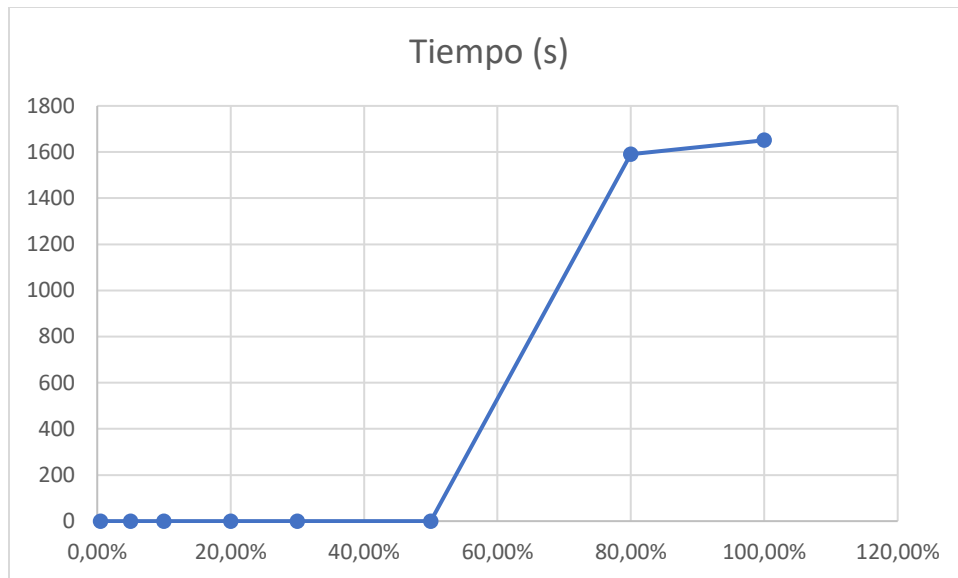
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small		
5 pct		
10 pct		
20 pct		
30 pct		
50 pct		
80 pct		
large	Subsector: 20 Cod actividad: 9810 9820 9700 9700	

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

En base a los resultados obtenidos en la gráfica y en la tabla podemos decir que la tendencia de la complejidad tiende a una complejidad de $n\log(n)$ pues el aumento entre los puntos es pequeño en relación a la cantidad de datos.

Requerimiento << No. 4 (Individual): Encontrar el subsector económico con los mayores costos y gastos de nómina para un año específico >>

Descripción

En la carga de datos se hicieron las sumatorias de todos los datos que se necesitaban. Por esta razón, la función compara todos los costos y gastos de nómina que tiene cada subsector y así encuentra cual es el mayor. Luego ordena los datos que se encuentran en el subsector mayor y encuentra las actividades económicas que más y menos aportan los costos y gastos de nómina del subsector.

Entrada	Año y el código del subsector
Salidas	Una tupla la cual contiene dos diccionarios, uno con los datos del mayor subsector y el otro con las 3 primeras y las 3 últimas actividades económicas.

Implementado (Sí/No)	Si se implementó y lo hizo Jaime Esteban Moreno Jaramillo
-----------------------------	---

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Hallar el mapa de subsectores de un año específico	$O(1)$
Recorrer todos los subsectores del año y hallar el que tiene mayores costos y gastos de nómina.	$O(S)$ $S = \text{numero de subsectores del año}$
Hallar la lista del subsector con mayores costos y gastos nómina.	$O(1)$
Ordenar la lista de actividades económicas de un subsector	$O(ns \cdot \log(ns))$ $ns = \text{actividades económicas de un subsector}$
Obtener los 3 elementos que más aportaron	$O(1)$
Obtener los 3 elementos que menos aportaron	$O(1)$
TOTAL	$O(ns \cdot \log(ns))$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Máquina 1	
Procesador	10th Gen Intel Core i7-10750H
	8
Sistema Operativo	Windows 11

Entrada	Tiempo (ms)
0.5	1.324
5	1.930
10	2.355
20	3.383
30	6.145
50	8.804
80	13.075
100	14.955

Tablas de datos

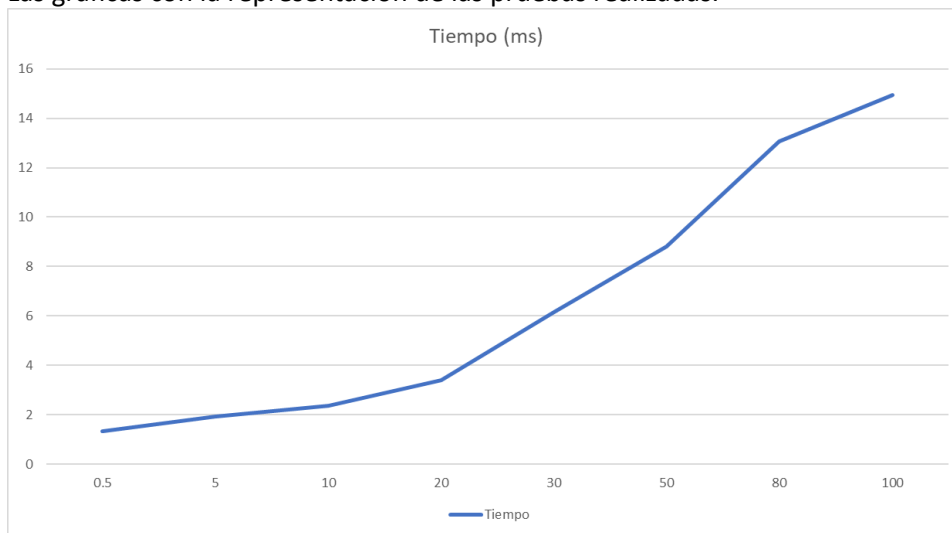
Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Subsector: 3 Cod act: 2750 1392 3320	1.324
5 pct	Subsector: 7 Cod act: 4663 4520 4520 4541 4512 4782	1.930
10 pct	Subsector: 7 Cod act: 4645 4690 4663 4541 4512 4782	2.355
20 pct	Cod subsector: 7 Cod act: 4645 4690 4663 4512 4512 4782	3.383
30 pct	Cod subsector: 3 Cod act: 2023 2229 1089 2818 3220 2680	6.145
50 pct	Cod subsector: 3	8.804

	Cod act: 3312 2023 2229 3220 2826 2680	
80 pct	Cod subsector: 3 Cod act: 1410 2100 3312 2826 2680 1922	13.075
large	Subsector: 3 Cod act: 1410 2100 3312 2826 2680 1922	14.955

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Como se puede ver la grafica concuerda con la complejidad $n \log n$, porque crece casi lineal, pero llega un punto (con muchos datos) en el que dispara y crece considerablemente mas que antes.

Requerimiento << No. 5 (Individual): Encontrar el subsector económico con los mayores descuentos tributarios para un año específico >>

Descripción

En la carga de datos, como se especificó anteriormente se hicieron las sumatorias de todos los datos que se necesitaban. Por esta razón, la función compara todos los totales de descuentos tributarios que tiene cada subsector y así encuentra cual es el mayor. Luego ordena los datos que se encuentran en el subsector y encuentra las actividades económicas que más y menos aportan al subsector.

Entrada	El diccionario con la estructura de la carga de datos y el año
Salidas	Dos tablas, una con la información del subsector con mayores descuentos tributario. Y otra con las actividades económicas que más y menos aportaron en el subsector.
Implementado (Sí/No)	Sí se implementó por Juan Diego Sánchez - 202214625

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener el mapa del año y del subsector	$O(\text{factor de carga}(fc)) = O(1)$
Un <i>for</i> recorre los subsectores en el mapa "subsector" y elige el mayor	$O(\text{número subsectores})$
Obtener la lista de actividades del subsector	$O(fc) = O(1)$
Ordenar las actividades con <i>Merge</i>	$O(n \cdot \log(n)) \rightarrow n$: número de actividades en la lista
Creación de "sub_arraylist" y añadir los elementos a devolver.	$O(1)$
Creación de la información para devolver la información pedida en el requerimiento del subsector con mayores descuentos tributarios.	$O(1)$
TOTAL	$O(n \cdot \log(n))$

Nota: Se omitió la complejidad de usar *tabulate* para darle el formato de tablas que se observa en el view. La complejidad de esta sería la complejidad total y es $O(n^*) \rightarrow n^*$: número de celda (menos los headers) que hay en la tabla más grande.

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Maquina 3	
Procesador	AMD Ryzen 3 3250U
	12
Sistema Operativo	Windows 11

Tabla de datos:

Se tomaron con el año 2021

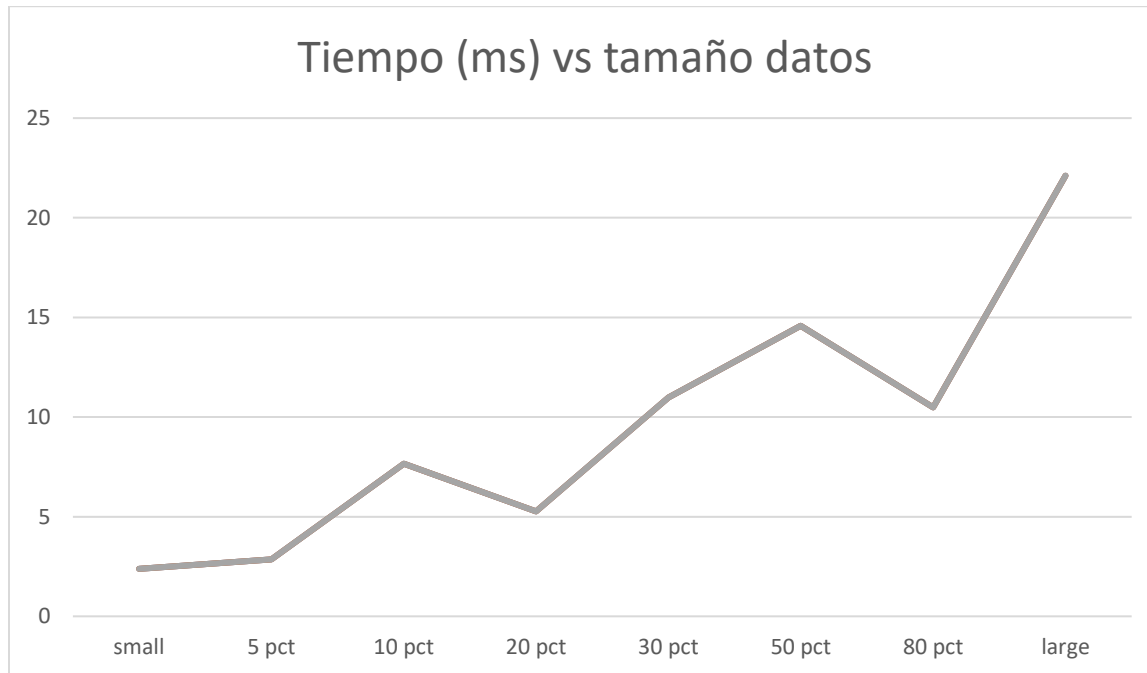
Entrada	Tiempo (ms)
small	2.383
5 pct	2.862
10 pct	7.651
20 pct	5.269
30 pct	10.975
50 pct	14.576
80 pct	10.485
large	22.109

Tablas de datos

Muestra	Salida (Subsector – mayor actividad – menor actividad)	Tiempo (ms)
small	3 – 2750 – 3320	2.383
5 pct	7 – 4663 – 4782	2.862
10 pct	7 – 4690 – 4782	7.651
20 pct	7 – 4690 – 4782	5.269
30 pct	3 – 1104 – 3099	10.975
50 pct	3 – 1104 – 3099	14.576
80 pct	11 – 6532 – 6411	10.485
large	3 – 1103 – 3220	22.109

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Análisis de resultados de la implementación, tener cuenta las pruebas realizadas y el análisis de complejidad.

Los tiempos son buenos, tal vez debería buscarse una forma de reducir la complejidad a la hora de ordenar las actividades económicas. Merge fue el que mejores resultados arrojó en mis pruebas. Sin embargo, tal vez si se ordenaran estos datos en la carga de datos podrían mejorarse los tiempos. No obstante, esto no es viable ya que estos datos son utilizados en otros requerimientos. Adicionalmente, considero que se logra una gran mejoría a los tiempos del primer reto gracias a la implementación que le dimos a los mapas de hash.

Requerimiento << Requerimiento No. 6 (Grupal): Encontrar el sector económico con el mayor total de ingresos netos para un año específico >>

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Año
Salidas	Una tabla con la información del sector que mayores ingresos netos obtuvo en el año ingresado. Y 2 tablas adicionales con información

	de los subsectores que más y menos aportaron y las respectivas actividades que más y menos aportaron a cada subsector.
Implementado (Sí/No)	Sí se implementó

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener el mapa de sectores del año ingresado	$O(1)$
Recorrer los sectores en el mapa sectores del año ingresado $O(\#sectores)$. Y elegir el mayor $O(\#Sectores(1))$. Nos queda $O(\#Sectores)$	$O(\#Sectores)$
Recorrer los subsectores que se encuentran en el sector mayor $O(\#subsectores \text{ en el sector})$. Comparaciones para elegir el mayor y el menor $O(\#subsectores \text{ en el sector}(1))$	$O(\#Subsectores \text{ en el sector})$
Crear las listas para las respuestas $O(1+1+O(n*\log(n)))$ n es el número de elementos en el subsector, mayor o menor, que tiene más elementos.	$O(n*\log(n))$
TOTAL	$O(n*\log(n))$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

	Máquina 1	Máquina 2	Máquina 3
Procesadores	10th Gen Intel Core i7-10750H	11th Gen Intel Core i5-1135G7	AMD Ryzen 3 3250U
	8	8	12
Sistema Operativo	Windows 11	Windows 11	Windows 11

Año 2021

Entrada	Tiempo (ms)
small	1.993
5 pct	8.460
10 pct	7.842
20 pct	5.530
30 pct	2.766
50 pct	3.352
80 pct	2.620
large	9.346

Año 2021

Entrada	Tiempo (ms)
small	7.345
5 pct	5.308
10 pct	3.424
20 pct	8.465
30 pct	7.547
50 pct	1.174
80 pct	7.968
large	5.955

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida (Sector económico – subsector que más aportó – subsector que menos aportó)	Tiempo (ms)
small	8 – 11 – 11	7.345
5 pct	6 – 7 – 9	5.308
10 pct	6 – 7 – 9	3.424
20 pct	6 – 7 – 9	8.465
30 pct	6 – 7 – 9	8.394
50 pct	8 – 11 – 12	5.996
80 pct	8 – 11 – 12	7.968
large	8 – 11 – 12	5.955

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

Análisis de resultados de la implementación, tener cuenta las pruebas realizadas y el análisis de complejidad.

Para este requerimiento consideramos que se obtuvieron muy buenos resultados. Ya que se necesitan obtener muchos datos, y los tiempos son menores a 10 ms en las pruebas. Adicionalmente, observamos que nuevamente la complejidad de la organización termina siendo la complejidad de todo el algoritmo. Pensamos que esta sigue siendo una de las limitaciones de usar las estructuras de datos que hemos

visto. Sin embargo, la complejidad de buscar el sector con mayores ingresos netos logra ser igual a $O(\#sectores)$ gracias a nuestra carga de datos.

Requerimiento << Requerimiento No. 7 (Grupal): Listar el TOP (N) de las actividades económicas con el menor total de costos y gastos para un subsector y un año específicos >>

Descripción

Obtiene la lista que contiene las actividades economicas del año y del subsector requerido por el usuario, ordena las actividades por el total de costos y gastos y si la lista es mayor a la cantidad de actividades requeridas por el usuario obtiene de la lista de todas las actividades las que van desde la pos 1 hasta la que el usuario haya puesto y las va añadiendo a una lista vacia la cual será la que devolverá

Entrada	Mapa grande,año,subsector, top de actividades
Salidas	Lista con top de actividades economicas
Implementado (Sí/No)	Si se implementó

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Obtener el mapa del año requerido por el usuario	$O(1)$
Obtener el mapa con los codigos de los subsectores del año.	$O(1)$
Obtener el mapa con los datos del subsector requerido	$O(1)$
Obtener la lista de las actividades economicas requeridas por el usuario	$O(1)$
Organizar la lista usando shell short	$O(A * \log(A))$ A = numero de actividades
Si la lista es menor que el top de elementos requeridos por el usuario devolver la lista	$O(1)$
Si la lista es mayor al top requerido por el usuario crear una nueva lista vacía	$O(1)$

Obtener las actividades economicas que se encuentras desde la posición 1 hasta el top requerido por el usuario	O(top) Top = numero de elementos requeridos por el usuario
TOTAL	O(A)

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

	Máquina 1	Máquina 2	Máquina 3
Procesadores	10th Gen Intel Core i7-10750H	11th Gen Intel Core i5-1135G7	AMD Ryzen 3 3250U
	8	8	12
Sistema Operativo	Windows 11	Windows 11	Windows 11

Entrada	Tiempo (s)
0,5 %	0,181
5 %	0,154
10 %	0,198
20 %	0,196
30 %	0,242
50 %	0,939
80 %	1,54
100 %	0.825

Entrada	Tiempo (s)
0,5 %	0.186
5 %	0.185
10 %	0.283
20 %	0.256
30 %	0.356
50 %	0.567
80 %	1.590
100 %	1.651

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Cod act: 6422	0,181
5 pct	Cod act: 6422	0,154
10 pct	Cod act: 6422 6532	0,198
20 pct	Cod act: 6422 6423 6521 6532	0,196
30 pct	Cod act: 6522 6424 6613 6422 6423 6521 6532	0,242
50 pct	Cod act: 6522 6615 6629 6424 6421 6613 6422 6492	0,939
80 pct	Cod act: 6513 6531 6522 6615 6629 6493 6424 6421 6494	1,54
large	Cod act: 6514 6513 6531 6522	2,925

	6491	
	6615	
	6496	
	6629	
	6611	

Graficas

Las gráficas con la representación de las pruebas realizadas.

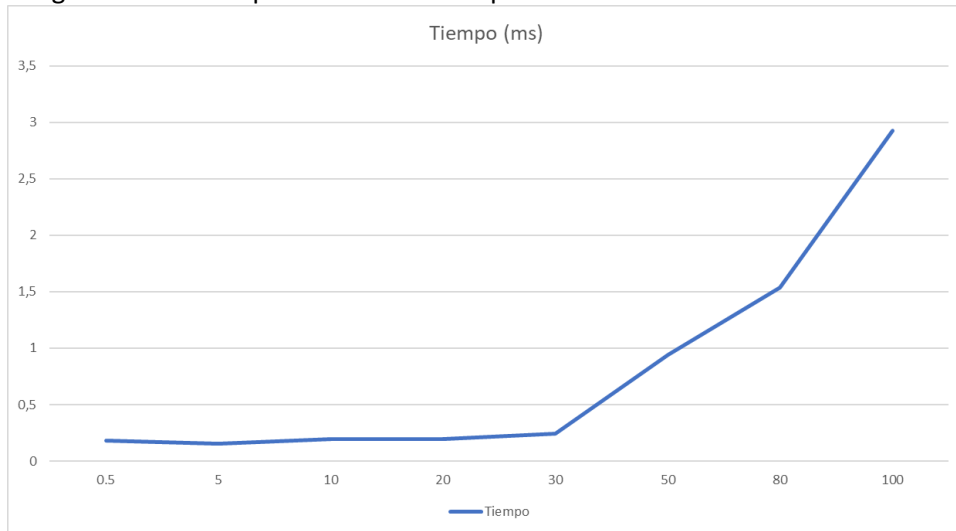
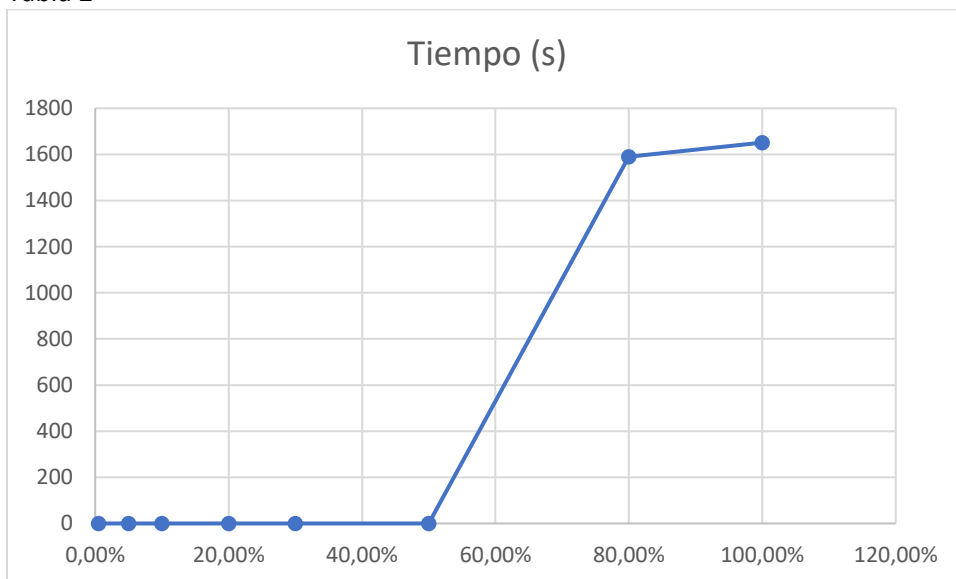


Tabla 2



Análisis

La grafica crece en $n \log(n)$, empieza creciendo lento, pero cuando hay mucho datos se dispara y deja de ser n .