ANÁLISIS DEL RETO

Catalina DeWasseige, 202220637, c.dew@uniandes.edu.co

Camilo Ramos, 202116748, cj.ramosh1@uniandes.edu.co

Luisa Gómez , 202222249, l.gomezo2@uniandes.edu.co

	Máquina
Procesadores	**Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz 2.21 GHz**
Memoria RAM (GB)	**16,0 GB**
Sistema Operativo	**Sistema operativo de 64 bits, procesador x64**

Introducción

El reto #1 se basa en el análisis y manejo de datos, sobre una base de datos externa. Atraves de este fueron necesarias habilidades de búsqueda e interpretación. En este reto se pone en práctica los conceptos aprendidos en clase acerca de las estructuras de datos lineales del módulo No. 1. Como lo son: las listas, pilas y colas, los algoritmos de búsqueda, y los algoritmos de ordenamiento.

Requerimiento <<1>>

Descripción

Este requerimiento pide identificar la actividad económica que tuvo el mayor saldo total de impuestos a pagar (Total saldo a pagar) para cada año disponible en la carga de datos. Lo abordamos primero haciendo un recorrido sobre la lista para así poder comparar el total saldo a pagar dentro de cada una. Después de esto se hace uso el quick Sort para ordenar los datos y una función auxiliar comparar_anio que sirve como parámetro del Sort.

Salidas	Lista que contiene las actividades económicas con mayor total saldo a pagar para todos los años disponibles
Implementado (Sí/No)	Si

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1: Crear una lista vacía que será la que se retornará	O(.1)
Paso 2: Recorrer la lista que contiene sublistas por año	O(n)
Paso 3: Recorrer cada sublista	O(n)

Paso 4: Comparar el valor del saldo total a pagar al asignarla a una variable “mayor” si el valor es mayor, y al finalizar el recorrido añadir el mayor a la lista de retorno	O(1)
Paso 5: Hacer un quick sort con el parámetro de la función comparar_anio	O(nlogn)
TOTAL	***O(n^logn)***

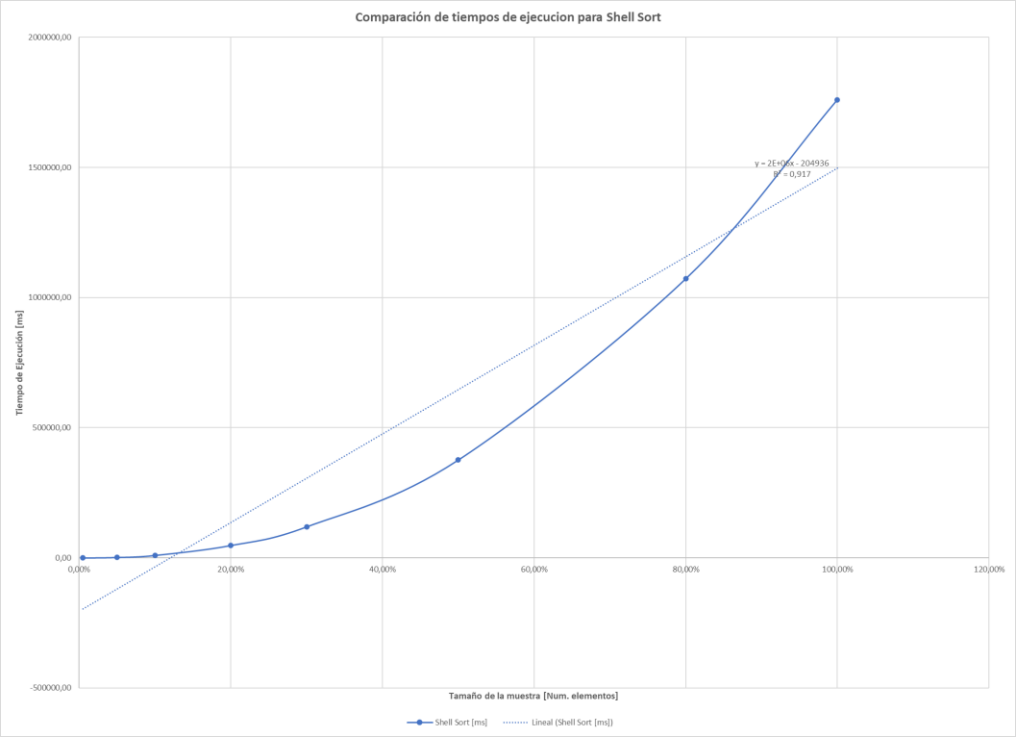
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	3.47
20%	1.61
50%	3.69

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Tras ver la toma de tiempos al cargar los diferentes tamaños de archivos se puede concluir que el tiempo es poco constante. Al tener una complejidad temporal de O(n^logn), esto es una complejidad buena ya que el tiempo no aumentara en grande cantidad en las diferentes cargas.

Requerimiento <<2>>

Descripción

Esta función busca la actividad económica con mayor total saldo a favor para todos los años disponibles. Los datos necesarios sol los Totales saldos a favor para cada año y código.

Entrada	La base de datos
Salidas	Lista con la actividad económica con mayor total saldo a favor para todos los años disponibles
Implementado (Sí/No)	Si

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1: Crear una lista vacía que será la que se retornará	O(1)
Paso 2: Recorrer la lista que contiene sublistas por año	O(n)
Paso 3: Recorrer cada sublista	O(n)
Paso 4: Comparar el valor del saldo total a favor al y asignarla a una variable "mayor" si el valor es mayor, y al finalizar el recorrido añadir el mayor a la lista de retorno	O(nlog)
Paso 5: Hacer un quick sort con el parámetro de la función comparar_anio	O(1)
TOTAL	***O(n^2logn)***

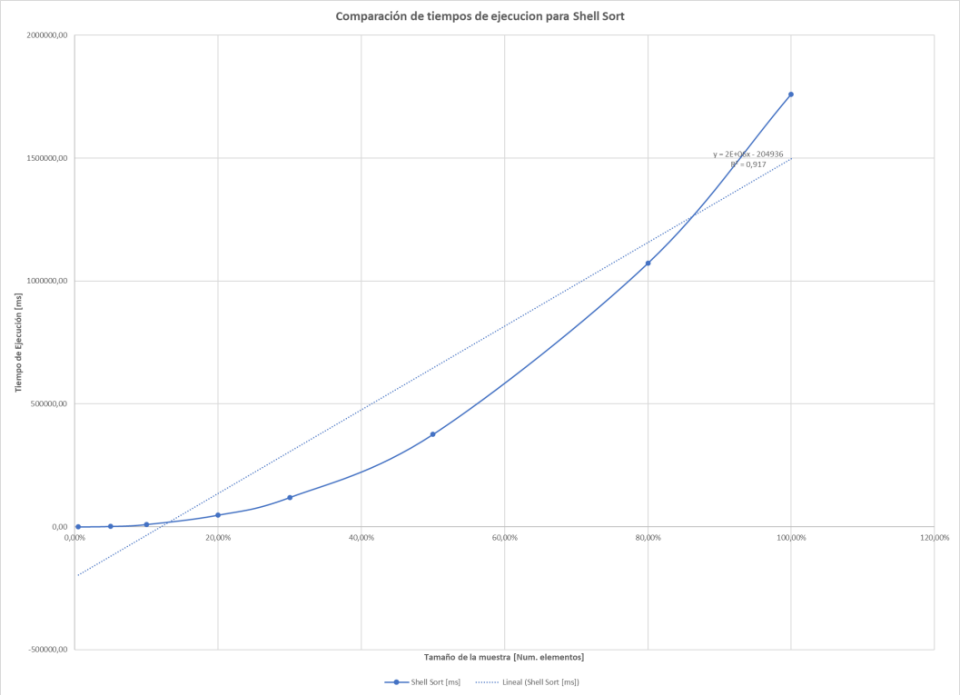
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	2.97
20%	2.87
50%	3.94

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Tras ver la toma de tiempos al cargar los diferentes tamaños de archivos se puede concluir que el tiempo es poco constante. Al tener una complejidad temporal de O(n^logn) esto es una complejidad buena ya que el tiempo no aumentara en grande cantidad en las diferentes cargas.

<u>Requerimiento <<3>></u>

Descripción

Esta función encuentra el subsector económico que tuvo el menor total de retenciones (Total retenciones) para cada año disponible en los datos cargados. Además, si existen menos de 6 actividades económicas en el subsector, liste todas ordenadas de menor a mayor por el valor total de retenciones.

Entrada	Base de datos
Salidas	Lista con el subsector económico con el menor total de retenciones para cada año
Implementado (Sí/No)	Si

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1: Creación de las listas de retorno y de otra que se usara en el problema	O(1)

Paso 2: Recorrer la lista que contiene sublistas por año	O(n)
Paso 3: Llama a la función que crea una lista por subsector económico y que además suma cuanto aporta cada sector. Devuelve una lista con las sumas que además contiene los aportes de cada subsector	O(n+n+n)
Paso 4: Dentro del anterior recorrido se hace otro recorrido sobre la lista por subsector, dentro del cual se comparan el total de retenciones. Este recorrido encuentra el menor de cada año que es una lista	O(n)
Paso 5: Dentro del primer recorrido se hace un quick sort que ordena de menor a mayor por retenciones. Ademas se hacen dos addlast que agrega en la lista de retorno el menor de cada año, y otro que en la lista de las actividades agrega el aporte del menor	O(1)
Paso 6: Se hace un quick sort, por fuera de todo recorrido, que ordena la lista de retorno de menor año a mayor año	O(nlogn)
TOTAL	***O(n^2)***

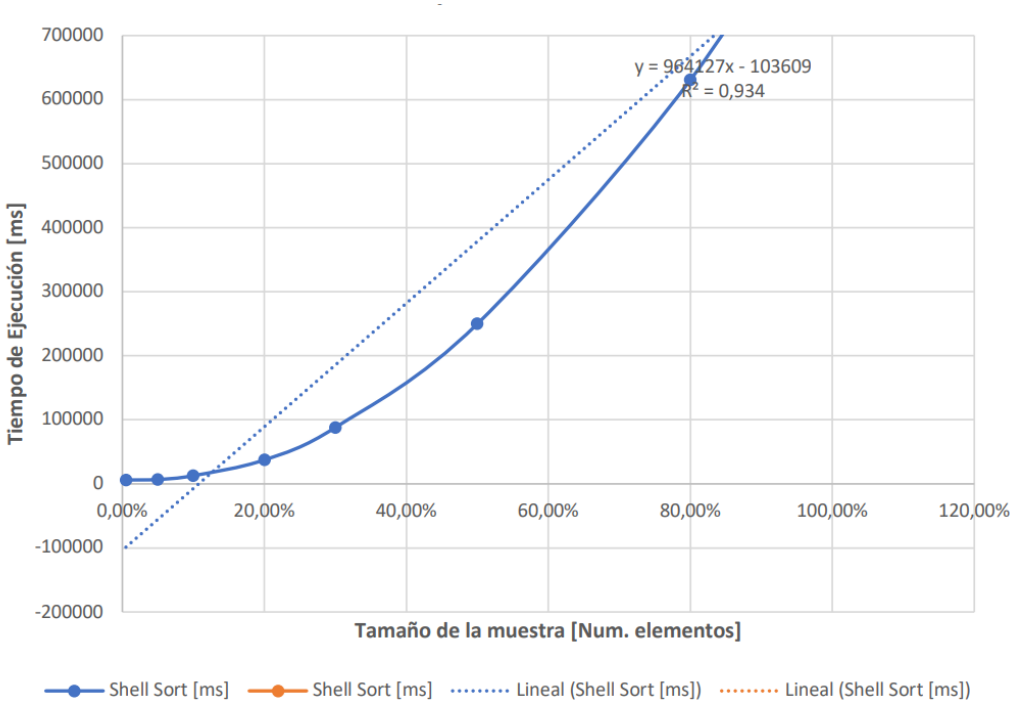
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	6.19
20%	55.91
50%	130.93

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Tras ver la toma de tiempos al cargar los diferentes tamaños de archivos se puede concluir que el tiempo es poco constante. Al tener una complejidad temporal de O(n^2) se puede decir que es exponencial, lo que significa que a medida que entre mas grande el archivo mas tiempo tardara el algoritmo en ejecutar. Esto

Requerimiento <<4>>

Descripción

Como analista económico necesito identificar el subsector económico que tuvo los mayores costos y gastos de nómina (Costos y gastos nómina) para cada año disponible en los datos cargados. Además, si existen menos de 6 actividades económicas en el subsector, la función las lista todas ordenadas de menor a mayor por el valor total de costos y gastos de nómina.

Entrada	Base de datos
Salidas	Lista con el subsector económico que tuvo los mayores costos y gastos de nómina (Costos y gastos nómina) para cada año disponible en los datos cargados
Implementado (Sí/No)	Si

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**

Paso 1: Creación de las listas de retorno y de otra que se usara en el problema	O(1)
Paso 2: Recorrer la lista que contiene sublistas por año	O(n)
Paso 3: Llama a la función que crea una lista por subsector económico y que además suma cuanto aporta cada sector. Devuelve una lista con las sumas que además contiene los aportes de cada subsector	O(n+n+n)
Paso 4: Dentro del anterior recorrido se hace otro recorrido sobre la lista por subsector, dentro del cual se comparan el total costos y gastos nómina del subsector. Este recorrido encuentra el mayor de cada año que es una lista	O(n)
Paso 5: Dentro del primer recorrido se hace un quick sort que ordena de menor a mayor por retenciones. Además, se hacen dos addlast que agrega en la lista de retorno el menor de cada año, y otro que en la lista de las actividades agrega el aporte del mayor	O(1)
Paso 6: Se hace un quick sort, por fuera de todo recorrido, que ordena la lista de retorno de menor año a mayor año	O(nlogn)
TOTAL	***O(n^2)***

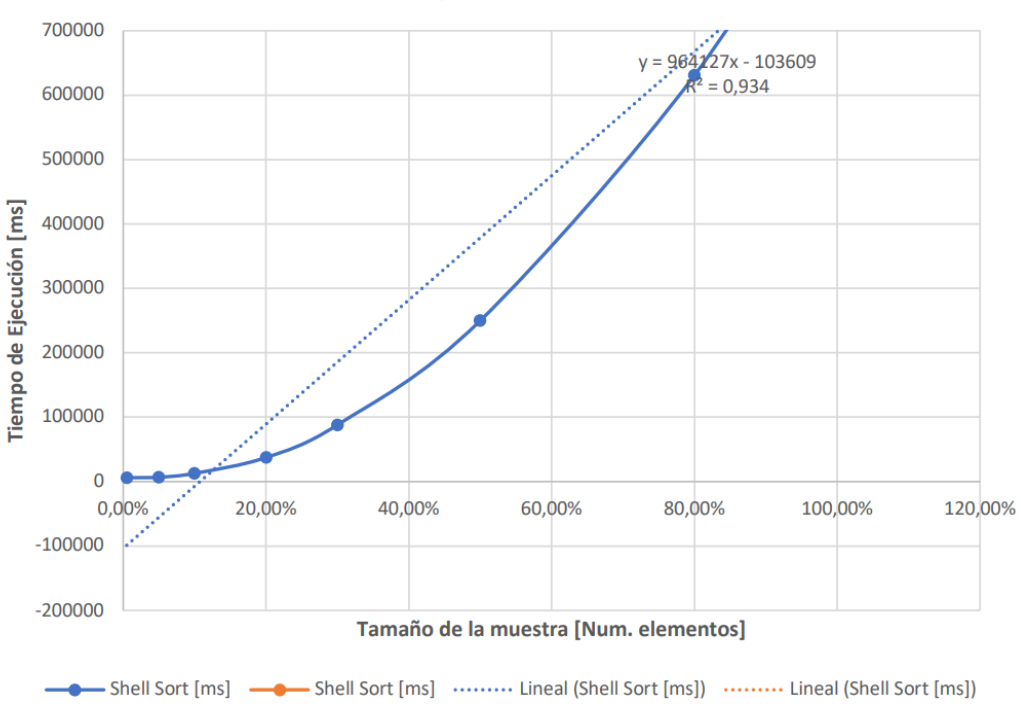
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	6.63
20%	63.35
50%	154.87

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Tras ver la toma de tiempos al cargar los diferentes tamaños de archivos se puede concluir que el tiempo es poco constante. Al tener una complejidad temporal de O(n^2) se puede decir que es exponencial, lo que significa que a medida que entre más grande el archivo más tiempo tardara el algoritmo en ejecutar. Un crecimiento exponencial es regular en términos de la implementación del requerimiento; sin embargo, logra concluir tiempos eficientes de ejecución.

Requerimiento <<5>>

Descripción

Como analista económico necesito encontrar el subsector económico que tuvo los mayores descuentos tributarios (Descuentos tributarios) para cada año disponible en los datos cargados. Si existen menos de 6 actividades económicas en el subsector, lístelas, todas ordenadas de mayor a menor por el valor total de descuentos tributarios.

Entrada	Base de datos
Salidas	Lista con el subsector económico que tuvo los mayores descuentos tributarios (Descuentos tributarios) para cada año disponible en los datos cargados
Implementado (Sí/No)	Si se implementó y quien lo hizo.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1: Creación de las listas de retorno y de otra que se usara en el problema	O(1)
Paso 2: Recorrer la lista que contiene sublistas por año	O(n)
Paso 3: Llama a la función que crea una lista por subsector económico y que además suma cuanto aporta cada sector. Devuelve una lista con las sumas que además contiene los aportes de cada subsector	O(n+n+n)
Paso 4: Dentro del anterior recorrido se hace otro recorrido sobre la lista por subsector, dentro del cual se comparan el Total de descuentos tributarios del subsector económico. Este recorrido encuentra el mayor de cada año que es una lista	O(n)
Paso 5: Dentro del primer recorrido se hace un quick sort que ordena de menor a mayor por retenciones. Ademas se hacen dos addlast que agrega en la lista de retorno el mayor de cada año, y otro que en la lista de las actividades agrega el aporte del mayor	O(1)
Paso 6: Se hace un quick sort, por fuera de todo recorrido, que ordena la lista de retorno de menor año a mayor año	O(nlogn)
TOTAL	***O(n^2)***

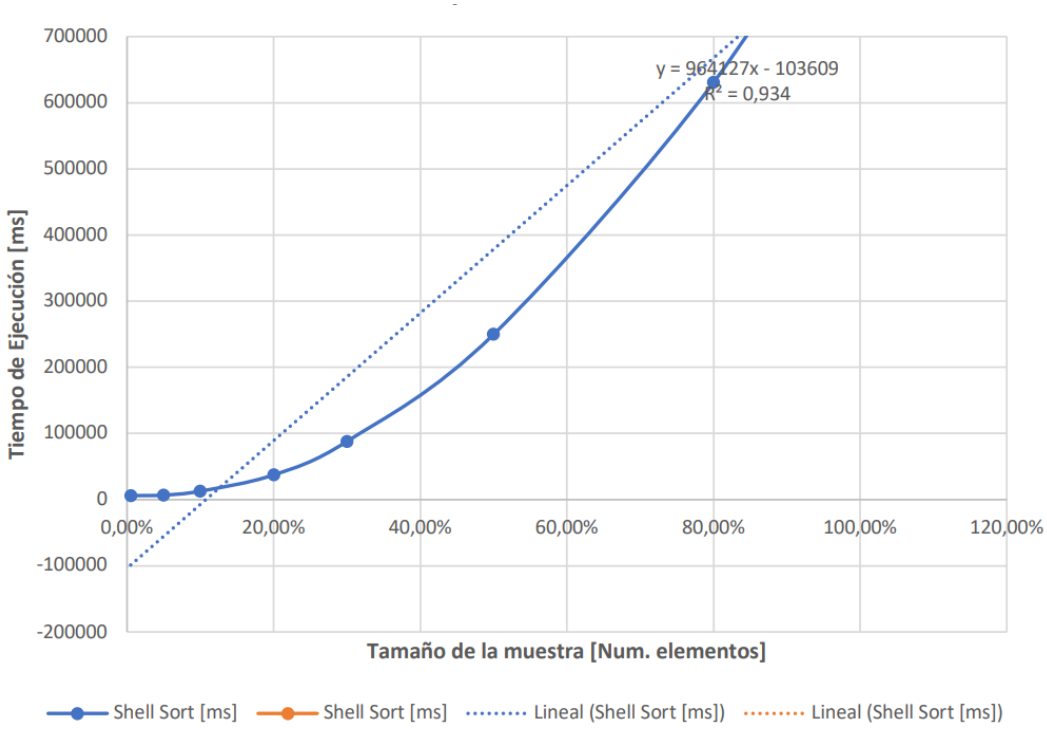
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	7.65
20%	78.48
50%	115.25

Graficas

Las gráficas con la representación de las pruebas realizadas.



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

Tras ver la toma de tiempo al ver los diferentes tamaños de archivos se puede notar que la función es medianamente buena, gracias a los tiempos de ejecución. Teniendo en cuenta la gráfica, se ve con claridad un crecimiento exponencial (n ^2) por lo que el análisis del código tiene una buena aproximación. Además, un crecimiento exponencial es regular en términos de la implementación del requerimiento; sin embargo, logra concluir tiempos eficientes de ejecución.

Requerimiento <<6>>

Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción
Breve descripción de como abordaron la implementación del requerimiento

Entrada	Parámetros necesarios para resolver el requerimiento.
Salidas	Respuesta esperada del algoritmo.
Implementado (Sí/No)	Si se implementó y quien lo hizo.

Análisis de complejidad
Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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).

Entrada	**Tiempo (ms)**
small	
20%	
50%	

Graficas
Las gráficas con la representación de las pruebas realizadas.

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

Requerimiento <<7>>

Descripción
Esta función lista las actividades económicas que tuvieron los menores valores totales de costos y gastos (Total costos y gastos), para un periodo de tiempo específico.

Entrada	Base de datos, El número (N) de actividades económicas a identificar, año inicial, año final.
Salidas	Una lista con las actividades económicas que tuvieron los menores valores totales de costos y gastos (Total costos y gastos), para un periodo de tiempo específico.
Implementado (Sí/No)	Si

Análisis de complejidad
Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1: Crear una lista que será la lista que se retornará	O(1)

Paso 2: Se hace un recorrido de la lista que contiende las sublistas por año	O(n)
Paso 3: Condicional que busca el año en el rango de años dado por parámetro. Que además adentro de este crea otro condicional por si el número de top que quieres es mayor al número de elementos que hay en la lista	O(1)
Paso 4: Dentro de este condicional se hace otro recorrido para coger cada elemento del top, y dentro de este se le agrega a la lista la línea correspondiente. Para encontrar la línea se hace un getElement de la lista del año.	O(n)
TOTAL	***O(n)***

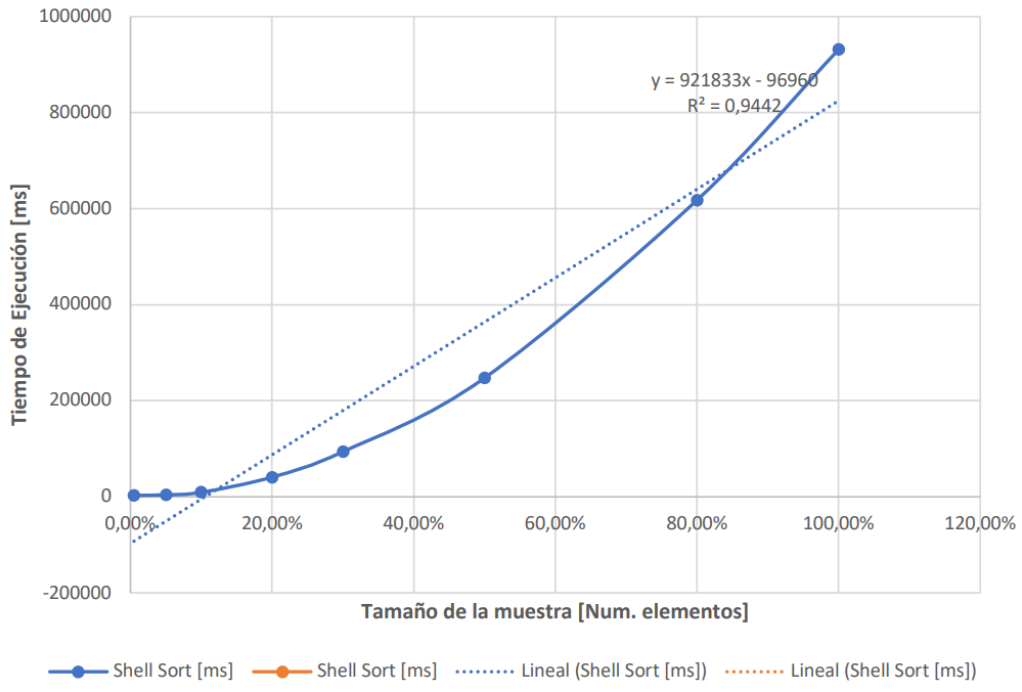
Pruebas Realizadas

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).

Entrada	**Tiempo (ms)**
small	2.48
20%	7.87
50%	15.05

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

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

Tras ver la toma de tiempo al ver los diferentes tamaños de archivos se puede notar que la función es bastante buena, gracias a los tiempos de ejecución. Teniendo en cuenta la gráfica, se ve con claridad un crecimiento constante por lo que el análisis del código tiene una buena aproximación. Además, un crecimiento constante es bastante bueno en términos de la implementación del requerimiento.