

ANÁLISIS DEL RETO

*Andrés Cárdenas Layton 202122083**Esteban Benavides 202220429*

Requerimiento 1

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

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	No requiere parametros de entrada
Salidas	Actividad economica con mayor total saldo a pagar para cada año
Implementado (Sí/No)	Se implemento y fue hecha en grupo por ambos integrantes

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1: Organizar los datos por año (1 while)	$O(N)$
Paso 2 : Organizar datos por total saldo a pagar (1 While)	$O(N)$
TOTAL	$O(2N) \rightarrow O(N)$

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).

Procesadores	AMD Ryzen 9 5900HS with Radeon Graphics 3.30 GHz
Memoria RAM	16Gb
Sistema Operativo	Windows 10 home

Entrada	Tiempo (s)
---------	------------

5pct	2.682 ms
10pct	6.669 ms
20pct	10.043 ms
30pct	28.284 ms

50 pct	34.624 ms
80 pct	59.238 ms
large	78.119 ms

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

Requerimiento 3

Plantilla para documentar y analizar cada uno de los requerimientos.

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	No requiere parámetros
Salidas	subsector con menor total de retenciones
Implementado (Sí/No)	si, Andrés Cárdenas Layton

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Organizar datos por año (while)	$O(N)$
Organizar datos por subsector para cada año (for dentro de while)	$O(N^2)$
Determinar el subsector con mayores costos y gastos nomina (While)	$O(N)$
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)
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

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 4

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

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	No requiere parametros
Salidas	subsector con mayor costo y gasto nomina por año

Implementado (Sí/No)	si(parcialmente), implementado por Esteban Benvides
----------------------	---

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Organizar datos por año (while)	$O(N)$
Organizar datos por subsector para cada año (for dentro de while)	$O(N^2)$
Determinar el subsector con mayores costos y gastos nomina (While)	$O(N)$
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)
5 pct	2.303 ms
10 pct	3.920 ms
20 pct	7.546 ms
30 pct	10.964 ms
50 pct	18.36 ms
80 pct	21.41 ms
large	35.69 ms

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

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

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

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Top N, Año inicio, Año final
Salidas	Top N actividades con mayor total costo y gasto para el periodo de tiempo dado
Implementado (Sí/No)	Si, Grupal, desarrollado por ambos integrantes

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Organizar datos por año (While)	$O(N)$
Adquirir las listas de los años requeridos (While)	$O(N)$
Adquirir Top N actividades de las listas de los Años (While)	$O(N)$
TOTAL	$O(3N) \rightarrow O(N)$

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).

Utilizando como Top N = Top 3 entre 2020 y 2021

Entrada	Tiempo (ms)
5 pct	1.113 ms
10 pct	2.181 ms
20 pct	7.546 ms
30 pct	8.658 ms
50 pct	10.132 ms
80 pct	17.866 ms
large	22.327 ms

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

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 3

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
Organizar datos por año	$O(N)$
Organizar por subsector	$O(N^2)$
Comparar por retencion	$O(N)$
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)
5 pct	2.253
10 pct	3.964 ms
20 pct	7.273 ms
30 pct	10.83 ms
50 pct	14.627 ms
80 pct	17,829ms

large	21.427 ms
-------	-----------

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

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 Ejemplo

Descripción

```
def get_data(data_structs, id):
    """
    Retorna un dato a partir de su ID
    """
    pos_data = lt.isPresent(data_structs["data"], id)
    if pos_data > 0:
        data = lt.getElement(data_structs["data"], pos_data)
        return data
    return None
```

Este requerimiento se encarga de retornar un dato de una lista dado su ID. Lo primero que hace es verificar si el elemento existe. Dado el caso que exista, retorna su posición, lo busca en la lista y lo retorna. De lo contrario, retorna None.

Entrada	Estructuras de datos del modelo, ID.
Salidas	El elemento con el ID dado, si no existe se retorna None
Implementado (Sí/No)	Si. Implementado por Juan Andrés Ariza

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
-------	-------------

Buscar si el elemento existe (isPresent)	$O(n)$
Obtener el elemento (getElement)	$O(1)$
TOTAL	$O(n)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos de entrada fueron el ID 1.

Procesadores	AMD Ryzen 7 4800HS with Radeon Graphics
Memoria RAM	8 GB
Sistema Operativo	Windows 10

Entrada	Tiempo (ms)
small	0.05
5 pct	0.33
10 pct	1.28
20 pct	2.54
30 pct	4.98
50 pct	7.51
80 pct	13.81
large	25.97

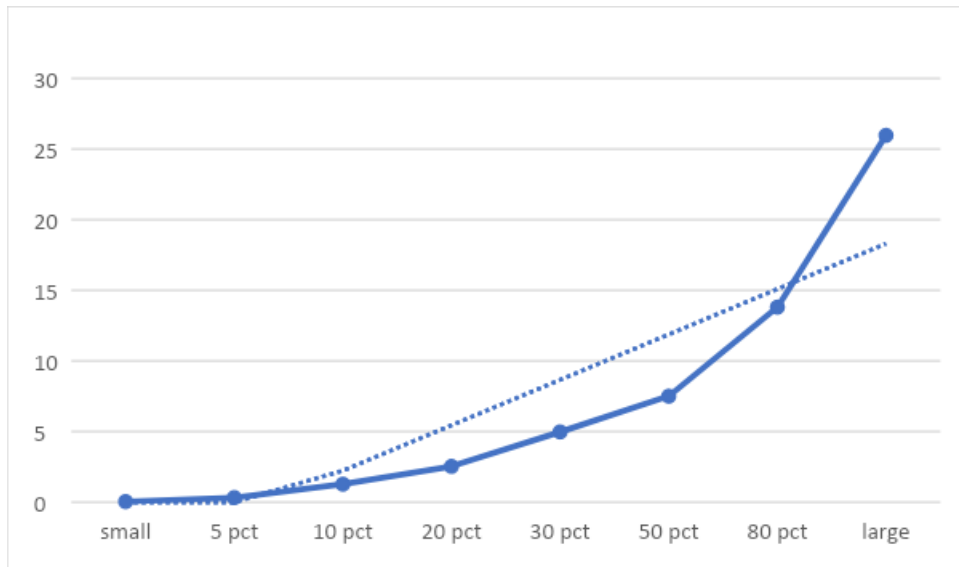
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	0.05
5 pct	Dato2	0.33
10 pct	Dato3	1.28
20 pct	Dato4	2.54
30 pct	Dato5	4.98
50 pct	Dato6	7.51
80 pct	Dato7	13.81
large	Dato8	25.97

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

A pesar de que obtener un elemento en un *ArrayList*, dada su posición, tiene complejidad constante, la implementación de este requerimiento tiene un orden lineal $O(n)$. Esto debido a que, lo primero que se hace es verificar si el elemento hace parte de la lista. Específicamente, a la hora de buscar un elemento en una lista, en el peor de los casos es necesario recorrer toda la lista, es decir, complejidad lineal.

Este comportamiento se puede evidenciar experimentalmente en la gráfica. Ya que, gracias a que los datos no se encuentran tan dispersos con respecto a la línea de tendencia, la curva coincide con el comportamiento lineal esperado.