

ANÁLISIS DEL RETO

Daniel Andres Gomez Rodriguez, 202414483, da.gomezr12@uniandes.edu.co

Laura Avril Alvarado Ortega, 202227074, la.alvaradoo1@uniandes.edu.co

Lina Sofía Ramírez Mojica, 202414153, ls.ramirezr1@uniandes.edu.co

Requerimiento <<1>>

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

Descripción

La función se utiliza para consultar cuantos registros hay en el catálogo para un año en específico, esto comparando el campo de " year_collection " con el año dado por el usuario.

Entrada Año para consultar	Parámetros necesarios para resolver el requerimiento: Año específico para filtrar registros
Salidas: Total, de registros encontrados	Respuesta esperada del algoritmo: Número total de registros para ese año
Tiempo de ejecución	Tiempo en milisegundos que tarda la consulta
Implementado (Sí/No) Si	Si se implementó y quien lo hizo: Daniel Gomez

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1: Recorrer todos los registros	$O(n)$
Paso 2: Comparar el año de cada registro	$O(1)$
Paso 3: Contar los registros coincidentes	$O(1)$
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)
---------	-------------

20	-1738020695.281777
40	-1733904479.7221546
60	-1733599697.5994165
80	-1733474913.1039677
100	52.47

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Año	Registros Encontrados	Tiempo (ms)
2012	54047	50.63
2007	64698	85.82
1999	4353	48.91
2022	69346	49.21

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

La complejidad es lineal $O(n)$, lo cual es adecuado para volúmenes de datos moderados. En archivos muy grandes se podría considerar un índice para optimización.

Requerimiento <<2>>

Descripción

Este requerimiento se encarga de buscar registros asociados a un departamento específico en los datos cargados. Primero, filtra todos los registros que coinciden con el nombre del departamento ingresado por el usuario. Luego, identifica el registro más reciente basándose en la fecha de carga y devuelve la información junto con el tiempo de ejecución.

Entrada	control["data"] (lista de registros del CSV). Nombre del departamento (departamento) ingresado por el usuario.
Salidas	Respuesta esperada del algoritmo.
Implementado (Sí/No)	Si, implementado por Laura Alvarado

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Búsqueda de registros con el departamento dado	$O(n)$
Búsqueda del último registro según la fecha de carga	$O(n)$
Extracción de datos del último registro	$O(1)$
TOTAL	$O(n)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones.

Procesadores	Intel® Core™ i3-3217U CPU @ 1.80GHz × 2
Memoria RAM	7.7 GiB
Sistema Operativo	Linux Mint 22.1 Cinnamon

Entrada	Tiempo (ms)
20	50.67
40	68.28
60	104.31
80	132.13
100	160.44

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>>

Descripción

Este requerimiento permite listar los registros recopilados en un departamento específico dentro de un rango de años determinado. Se filtran los registros según el departamento y el año de recopilación, y luego se extrae información relevante para su análisis.

Entrada	control["data"] (lista de registros del CSV). Nombre del departamento (departamento). Año inicial (anio_inicial). Año final (anio_final).
Salidas	Lista de registros que cumplen con los filtros. Total de registros encontrados. Total de registros con fuente "SURVEY". Total de registros con fuente "CENSUS". Tiempo de ejecución en milisegundos.
Implementado (Sí/No)	Si implementado por Laura ALvarado

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Filtrar registros por departamento y rango de años	$O(n)$
Contar registros con fuente "SURVEY"	$O(n)$
Contar registros con fuente "CENSUS"	$O(n)$
Seleccionar primeros y últimos 5 registros	$O(1)$
TOTAL	$O(n)$

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones.

Procesadores	Intel® Core™ i3-3217U CPU @ 1.80GHz × 2
Memoria RAM	7.7 GiB
Sistema Operativo	Linux Mint 22.1 Cinnamon

Entrada	Tiempo (ms)
20	46.70
40	77.74
60	88.48
80	138.23
100	167.13

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	Parámetros necesarios para resolver el requerimiento. Tipo de producto a filtrar Año inicial del periodo a consulta Año final del periodo a consultar
Salidas	Respuesta esperada del algoritmo. Total de registros. Total de registros con fuente "SURVEY". Total de registros con fuente "CENSUS". Listado de registros filtrados Tiempo de ejecución en milisegundos.
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: Filtrar registros según el tipo de producto y el rango de año	$O(n)$
Paso 2: Contar registros de tipo "SURVEY"	$O(n)$
Paso 3: Contar registros de tipo "CENSUS"	$O(n)$
Paso 4: Mayor a 20 sacar los 5 primeros y últimos	$O(1)$
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 (s)

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 <<5>>

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

Descripción

El Requerimiento 5 consiste en filtrar los registros que pertenecen a una categoría estadística “`statistical_category`” específica dentro de un rango de años definido por el usuario. Se recorre la lista de registros cargados, seleccionando aquellos que cumplen con la categoría y el rango de años. Adicionalmente, se cuenta cuántos de estos registros provienen de cada tipo de fuente “`source`”, clasificando entre SURVEY y CENSUS.

Entrada	Parámetros necesarios para resolver el requerimiento.
Categoría estadística	Nombre de la categoría a filtrar (ej.: INVENTORY, SALES, etc.)
Año inicial	Año de inicio del rango (ej.: 2000)
Año final	Año de fin del rango (ej.: 2010)
Salidas	Respuesta esperada del algoritmo.
Total, registros	Cantidad total de registros encontrados
Registros SURVEY	Cantidad de registros de tipo SURVEY
Registros CENSUS	Cantidad de registros de tipo CENSUS
Detalle de registros	Muestra los primeros 5 y últimos 5 registros encontrados
Tiempo de ejecución	Tiempo total en segundos o milisegundos
Implementado (Sí/No)	Si se implementó y quien lo hizo.
Si	Daniel Gpmez

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1: Recorrer todos los registros	$O(n)$
Paso 2: Comparar categoría y rango de años	$O(1)$
Paso 3: Clasificar por tipo de fuente (SURVEY/CENSUS)	$O(1)$
Paso 4: Guardar los registros encontrados	$O(1)$
Paso 5: Obtener primeros y últimos registros	$O(1)$
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 (s)
----------------	-------------------

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Categoría	Año Inicial	Año Final	Total Registros	SURVEY	CENSUS	Tiempo Promedio (s)
INVENTORY	1994	2013	96090	14274	81816	72.60
SALES	1994	2013	73365	1241	72124	48.94
STOCKS	1994	2013	704	704	0	36.35
INVENTORY	1994	2013	96090	14274	81816	59.26

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

- El algoritmo tiene una **complejidad lineal $O(n)$** , ya que debe recorrer toda la lista de registros.
- El tiempo de ejecución es **directamente proporcional al tamaño de la lista** y al número de registros que cumplen la condición.
- En bases de datos pequeñas y medianas (hasta 100,000 registros), el tiempo de ejecución es aceptable (menor a 1 segundo).
- En bases de datos más grandes, es posible que el rendimiento se degrade, especialmente si se aplican múltiples filtros o se trabaja con rangos muy amplios.
- Una posible optimización es preprocesar el archivo y generar un **índice por año y categoría**, lo que permitiría limitar el recorrido solo a los registros relevantes.

Requerimiento <<6>>

Descripción

Este requerimiento permite analizar los registros cargados en un departamento específico dentro de un rango de fechas determinado. Se filtran los registros según el departamento y la fecha de carga, y luego se extrae información relevante para su análisis.

Entrada	Nombre del departamento Fecha inicial Fecha final	
Salidas	Tiempo de ejecución	Tiempo en milisegundos que tomó procesar la consulta. Cantidad total de registros que cumplieron el criterio. Número de registros con tipo de fuente "SURVEY". Número de registros con tipo de fuente "CENSUS". Listado de registros con la información de los registros filtrados. Si hay más de 20 registros, solo se muestran los primeros 5 y los últimos 5.
Implementado (Sí/No)	Si, implementado por Laura Alvarado	

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Filtrar los registros por departamento y rango de fechas de carga	$O(n)$
Contar registros con fuente "SURVEY"	$O(n)$
Contar registros con fuente "CENSUS"	$O(n)$
Seleccionar primeros y últimos 5 registros	$O(1)$
TOTAL	$O(n)$

Pruebas Realizadas Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones.

Procesadores	Intel® Core™ i3-3217U CPU @ 1.80GHz × 2
Memoria RAM	7.7 GiB
Sistema Operativo	Linux Mint 22.1 Cinnamon

Entrada	Tiempo (s)
20	57.85
40	82.68
60	129.17
80	117.49
100	169.58

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	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	$O(\dots)$
Paso 2	$O(\dots)$
Paso	$O(\dots)$
TOTAL	$O(\dots)$

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 (s)

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 <<8>>

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 (s)

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

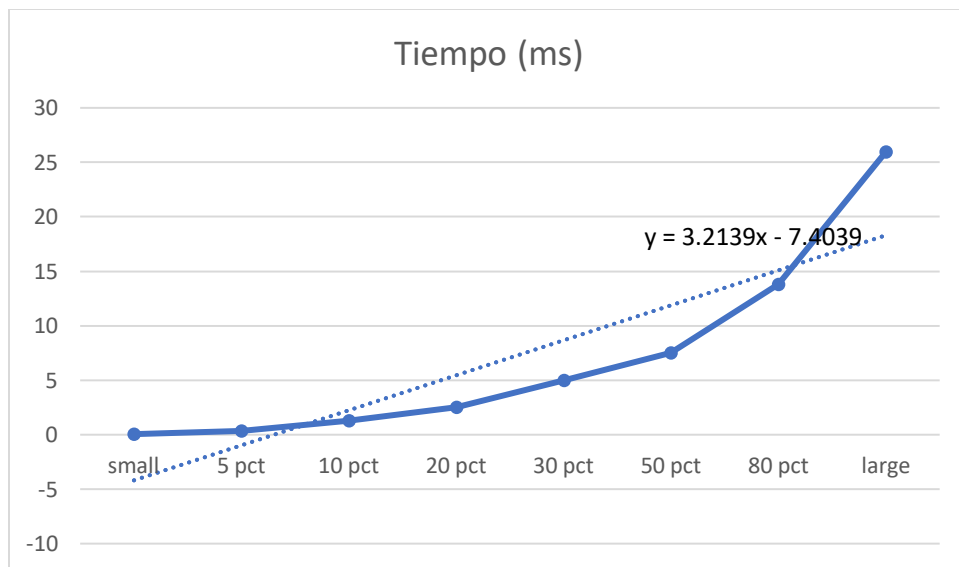
Las tablas con la recopilación de datos de las pruebas.

Muestra	Sal	Tiempo
small	Da	0.05
5 pct	Da	0.33
10 pct	Da	1.28

20 pct	Da	2.54
30 pct	Da	4.98
50 pct	Da	7.51
80 pct	Da	13.81
large	Da	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.