**ANÁLISIS DEL RETO**

Mateo Sanchez, 20232154 m.sanchezz@uniandes.edu.co

Juan David Nieto, 202411007, j.nietom@uniandes.edu.co

Jeronimo Quevedo, 202422580, j.quevedo1@uniandes.edu.co

# **Requerimiento <<1>>**

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

## **Descripción**

Manejamos los crímenes como un rango y lo recorrimos con un for, después implementamos una función adicional sort key que nos ayuda a organizarlo de más reciente al más antiguo

|  |  |
| --- | --- |
| **Entrada** | Catalog, fecha inicial y fecha final |
| **Salidas** | Los crímenes ocurridos entre el rango de fechas ordenados del mas reciente al mas antiguo y muestra los primeros 5 crímenes encontrados |
| **Implementado (Sí/No)** | Si se implementó / mateo Sánchez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n) |
| Paso …. | O(n) |
| ***TOTAL*** | ***O(n al cuadrado)*** |

## **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)** |
| 08/07/2020 – 08/08/2021 | 256.20 ms |
| 04/06/2020 – 06/29/2021 | 115.10 ms |
| 08/18/2020 – 01/13/2021 | 68.07 ms |

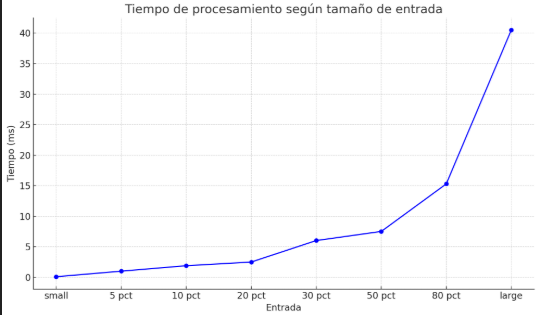
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 1 |
| 10 pct | 1.90 |
| 20 pct | 2.50 |
| 30 pct | 6.02 |
| 50 pct | 7.51 |
| 80 pct | 15.30 |
| large | 40.50 |

### **Graficas**



## **Análisis**

En este requerimiento implemente se uso un array list haciendo que se pudiera recorrer mucho mas sencillo que con alguna otra implementación, la complejidad aumento mas de lo esperado ya que tuvimos que recorrer dos veces la lista para sacar el rango pedido.

# **Requerimiento <<2>>**

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

## **Descripción**

Esta vez nos pedían un rango entre una fecha inicial y una fecha final también lo manejamos de la misma forma, lo probamos con las mismas fechas

|  |  |
| --- | --- |
| **Entrada** | Catalog, fecha inicial y fecha final |
| **Salidas** | Retorna los crímenes mas graves part 1 y entre dos fechas dadas |
| **Implementado (Sí/No)** | Si se implementó / mateo Sánchez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n) |
| Paso …. | O(n) |
| ***TOTAL*** | ***O(n al cuadrado)*** |

## **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).

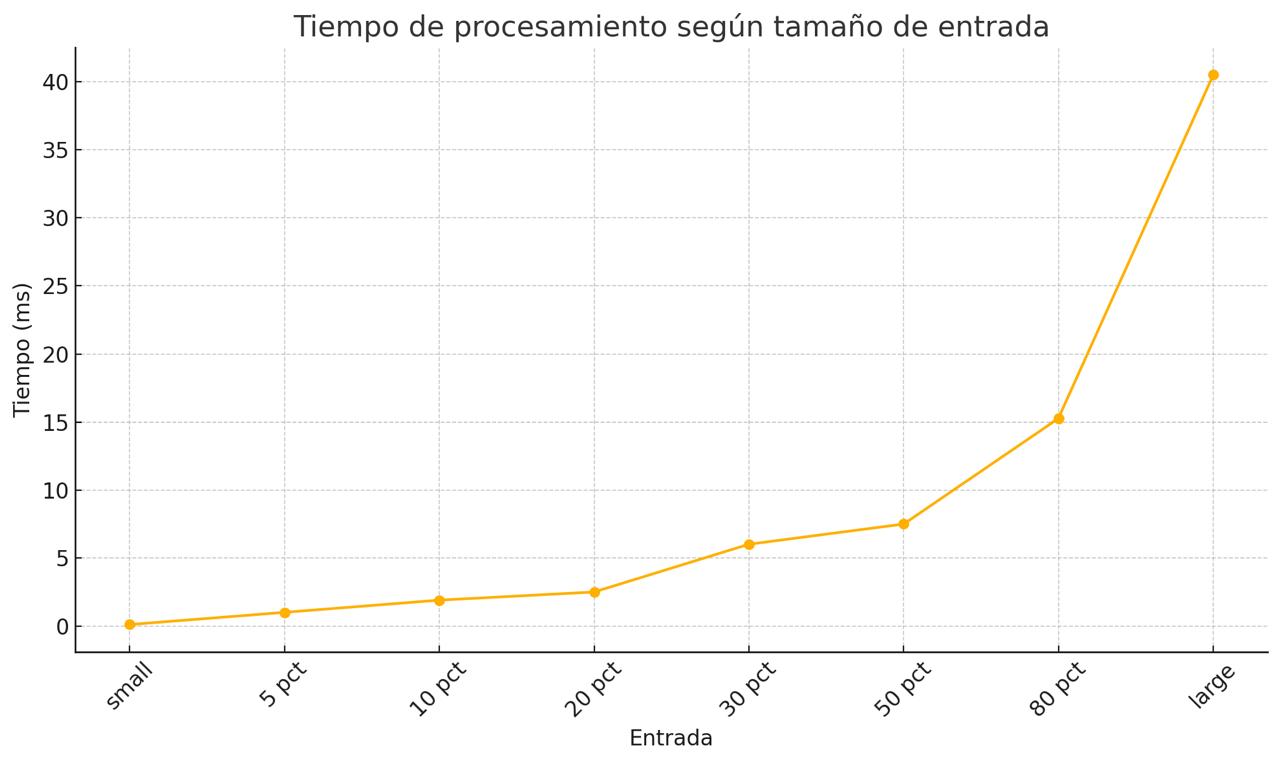
|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| 08/07/2020 – 08/08/2021 | 300 ms |
| 04/06/2020 – 06/29/2021 | 150 ms |
| 08/18/2020 – 01/13/2021 | 80 ms |

### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.18 |
| 5 pct | 1.25 |
| 10 pct | 2.10 |
| 20 pct | 2.80 |
| 30 pct | 6.24 |
| 50 pct | 8.06 |
| 80 pct | 16.52 |
| large | 45.20 |

### **Graficas**

|

## **Análisis**

En el segundo requerimiento logramos tener un resultado parecido al anterior en temas de complejidad usamos 2 for ya que teníamos que recorrer con dos índices la lista.

# **Requerimiento <<3>>**

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

## **Descripción**

Metimos los crímenes en una lista y la recorrimos hasta encontrar la posición de N

|  |  |
| --- | --- |
| **Entrada** | Catalog, área name, N |
| **Salidas** | Retorna los N crímenes mas recientes de un área |
| **Implementado (Sí/No)** | Si se implementó / todos |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n) |
| Paso …. | 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 (s)** |
| Mission | 50 ms |
| Pacific | 40.42 ms |
| Rampart | 70.32 ms |

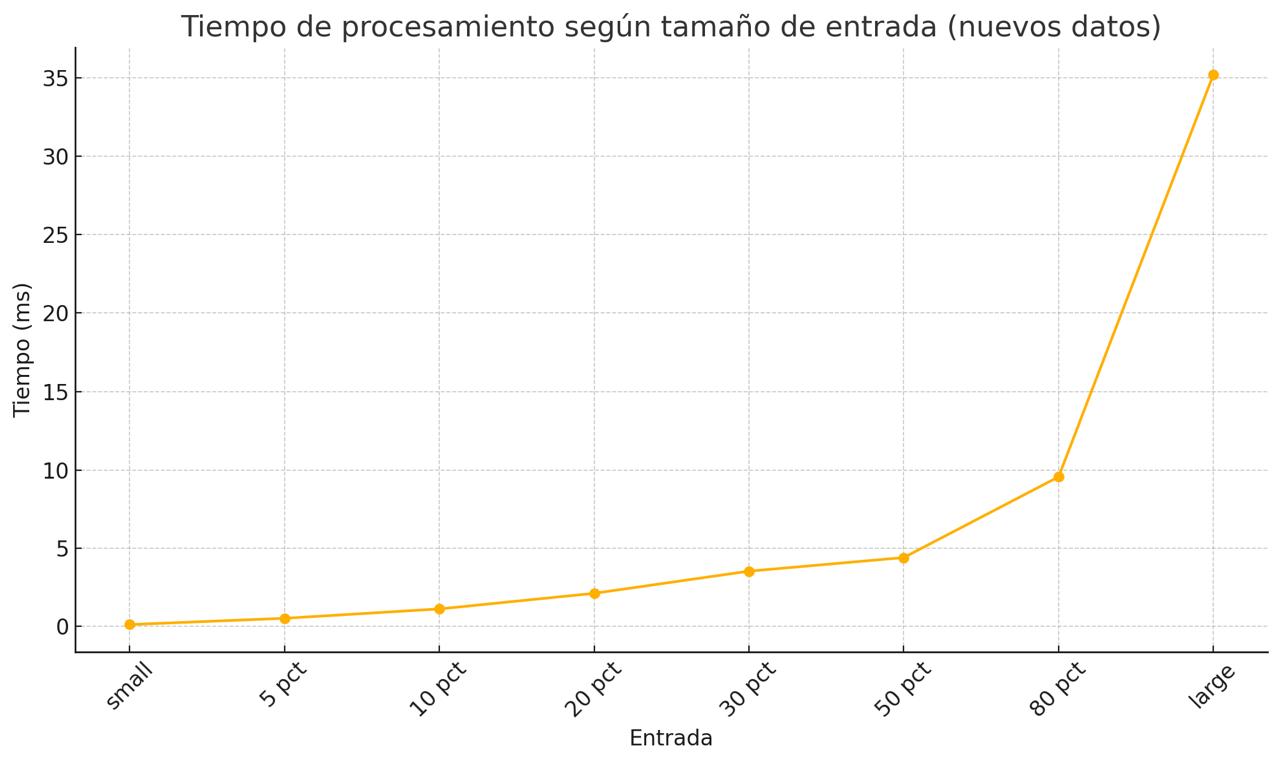
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 0.50 |
| 10 pct | 1.10 |
| 20 pct | 2.10 |
| 30 pct | 3.52 |
| 50 pct | 4.39 |
| 80 pct | 9.56 |
| large | 35.20 |

### **Graficas**



## **Análisis**

En este requerimiento implemente se usó un array list haciendo que se pudiera recorrer mucho más sencillo que con alguna otra implementación, la complejidad no aumento ya que solo tuvimos que recorrer una vez hasta encontrar N

# **Requerimiento <<4>>**

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

## **Descripción**

Hicimos una función adicional de sort criterio donde comparamos las edades para encontrar la pedida dentro de rango, usamos un árbol sacando los values siendo estos la edad mínima y máxima y después organizándolos con un merge sort invocando a la función anterior.

|  |  |
| --- | --- |
| **Entrada** | Catalog, N, Edad inicial, edad final |
| **Salidas** | Retorna el número total de crímenes que cumplen dentro del rango de edad, deben aparecer todas las víctimas de crímenes graves primero y luego las víctimas de crímenes menores ordenados descendentemente |
| **Implementado (Sí/No)** | Si se implementó / Juan David Nieto |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n logn) |
| Paso …. |  |
| ***TOTAL*** | ***O(nlogn)*** |

## **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)** |
| 28-34 | 36 .20ms |
| 26-47 | 48.9 ms |
| 25-51 | 78.32 ms |

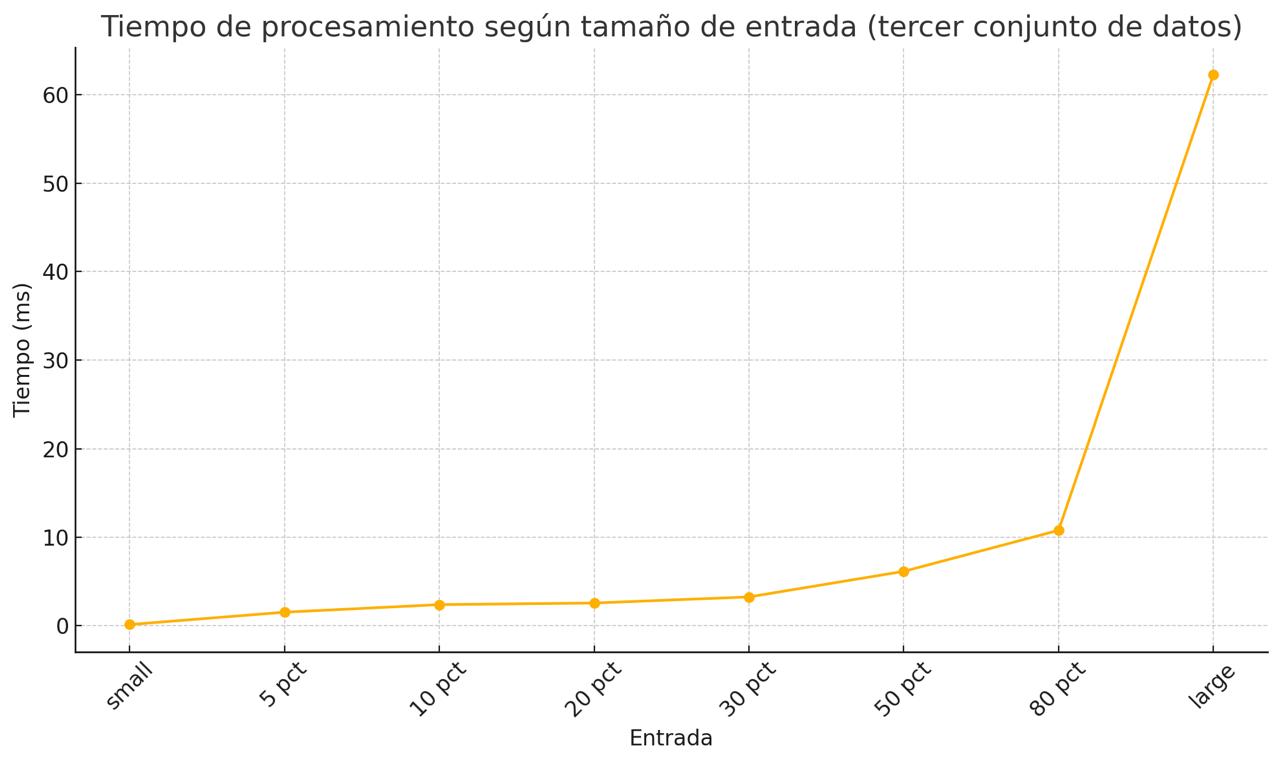
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 1.50 |
| 10 pct | 2.36 |
| 20 pct | 2.54 |
| 30 pct | 3.23 |
| 50 pct | 6.13 |
| 80 pct | 10.78 |
| large | 62.23 |

### **Graficas**



## **Análisis**

En este requerimiento se usó la implementación de arboles ya que se pudo hacer un recorrido recursivo al comparar y encontrar los parámetros pedidos.

# **Requerimiento <<5>>**

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

## **Descripción**

Hicimos otra función de sort criterio esta vez para encontrar los casos no resueltos y el nombre del área dentro de los rangos pedidos buscamos dentro del árbol si sí está presente y lo agregamos al diccionario nuevo, recorremos y buscamos si el caso esta resuelto y si es así se le suma 1 al elemento.

|  |  |
| --- | --- |
| **Entrada** | Catalog, N, date min date max |
| **Salidas** | Área en que ocurrió el crimen nombre del área cantidad de crímenes fecha del primer y último crimen |
| **Implementado (Sí/No)** | Si se implementó / Jeronimo Quevedo |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n logn) |
| Paso …. |  |
| ***TOTAL*** | ***O(nlogn)*** |

## **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)** |
| 12/22/2020 – 05/05/2021 | 40.35ms |
| 09/30/2020 – 05/28/2021 | 56.42ms |
| 10/10/2020- 07/31/2021 | 84.20 ms |

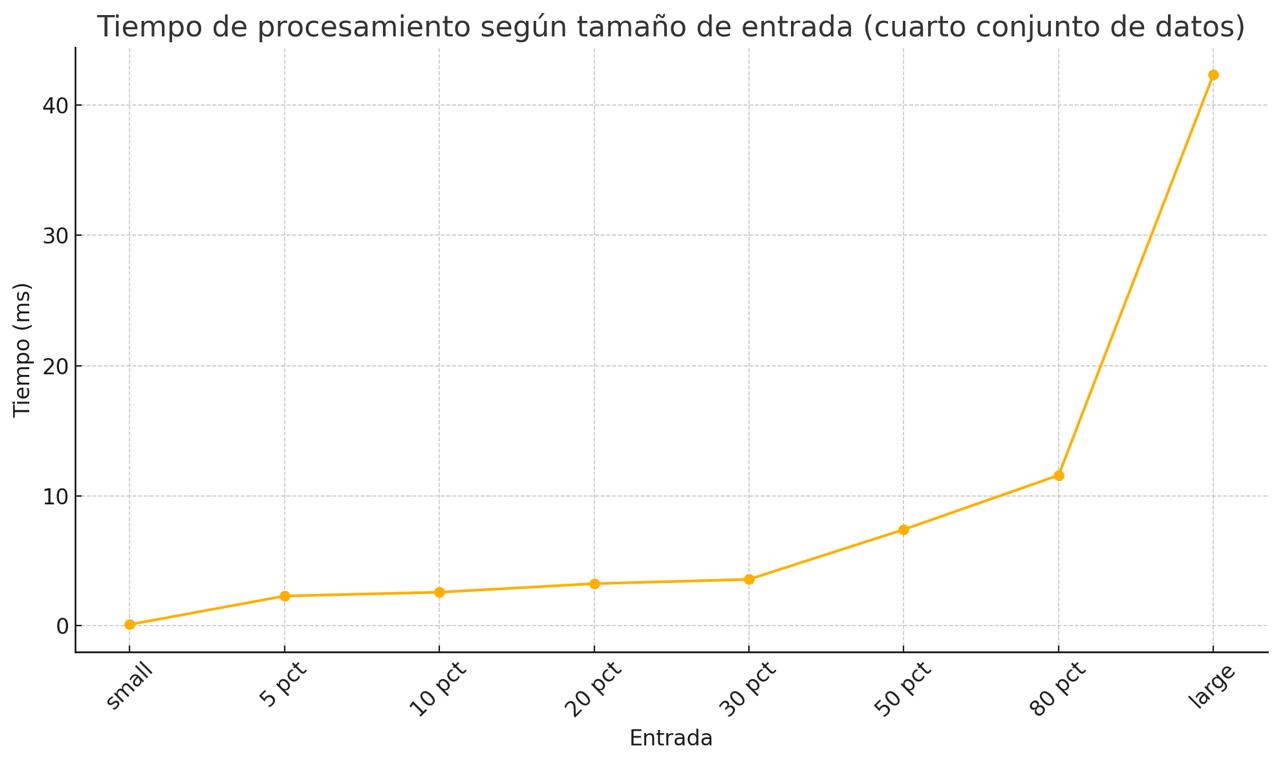
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 2.30 |
| 10 pct | 2.59 |
| 20 pct | 3.25 |
| 30 pct | 3.58 |
| 50 pct | 7.42 |
| 80 pct | 11.6 |
| large | 42.32 |

### **Graficas**



## **Análisis**

En el 5 requerimiento se usó una implementación parecida al 4 ya que se necesitaba una función adicional de comparación, al comparar ya que pedían retornar los parámetros organizados se uso un merge sort para tener organizado todo.

# **Requerimiento <<6>>**

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

## **Descripción**

Se creo un diccionario llamado áreas para poner encontrar las N mas seguras para un sexo y un mes especifico

|  |  |
| --- | --- |
| **Entrada** | Catalog, N, sexo, mes |
| **Salidas** | Retorna n áreas mas seguras, sexo en un mes especifico |
| **Implementado (Sí/No)** | Si se implementó / mateo Sánchez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n al cuadrado) |
| Paso …. |  |
| ***TOTAL*** | ***O(n al cuadrado)*** |

## **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)** |
| F 10 | 20.6 ms |
| M 02 | 32.5 ms |
| M 05 | 42.7 ms |

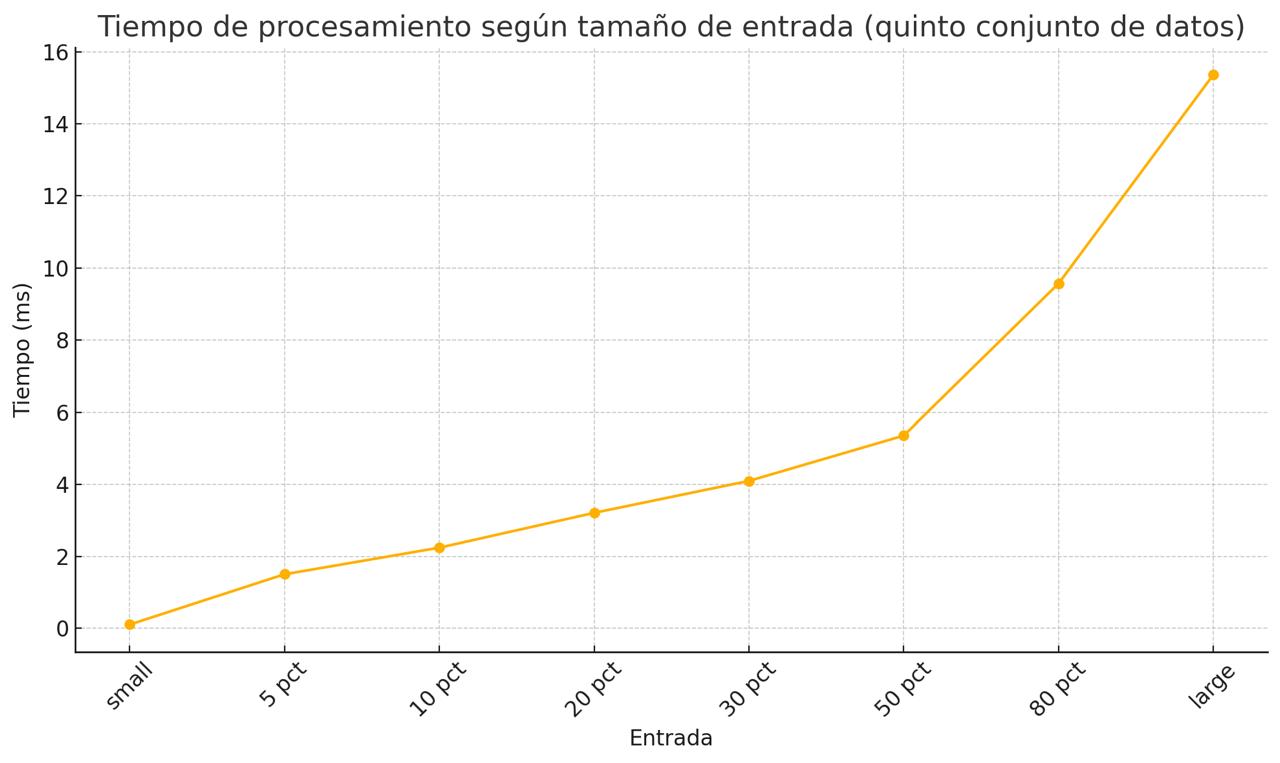
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 1.50 |
| 10 pct | 2.24 |
| 20 pct | 3.21 |
| 30 pct | 4.10 |
| 50 pct | 5.36 |
| 80 pct | 9.57 |
| large | 15.36 |

### **Graficas**



## **Análisis**

En este requerimiento se uso un diccionario inicial para ir metiendo el total de crímenes y los crímenes por año, se sacaron los valúes usando la implementación de arraylist y se uso un sort para organizar todo como se pedía, luego para el resultado creamos una lista donde fuimos metiendo los N variables pedidas, se mantuvo una complejidad baja para la complejidad del problema.

# **Requerimiento <<7>>**

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

## **Descripción**

Se creo un diccionario para meter los crímenes por tipo y poder organizarlos más fácilmente luego se recorrió el catalog y sacamos el sexo de la victima con un. get usamos una función adicional como criterio de ordenamiento para tener una mejor organización del resultado final.

|  |  |
| --- | --- |
| **Entrada** | Catalog, sexo, edad inicial, edad final, n |
| **Salidas** | retorna los n crímenes mas comunes en un rango de edades |
| **Implementado (Sí/No)** | Si se implementó / mateo Sánchez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Paso 1 | O(n) |
| Paso 2 | O(n al cuadrado) |
| Paso …. |  |
| ***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).

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| F 34-63 | 52.30 |
| M 30-59 | 42.25 |
| M 42-66 | 39.18 |

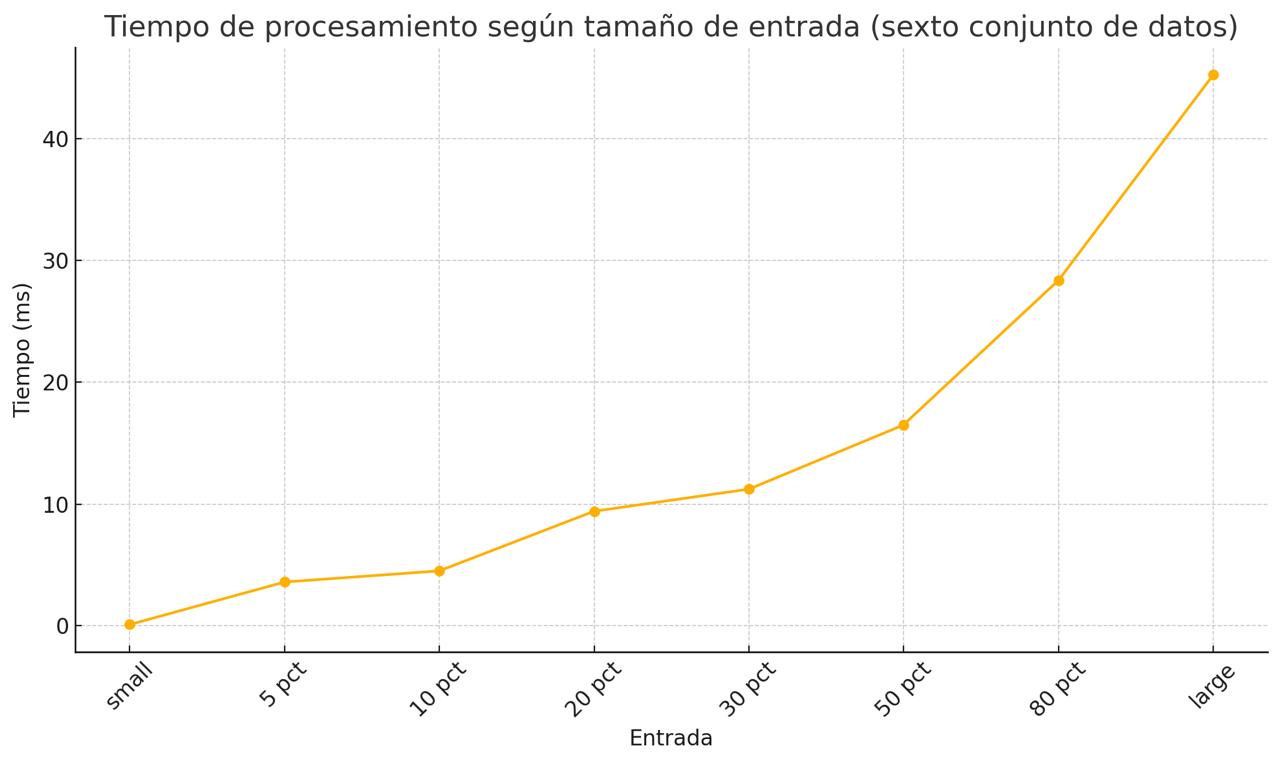
### **Tablas de datos**

Las tablas con la recopilación de datos de las pruebas.

|

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (ms)** |
| small | 0.10 |
| 5 pct | 3.60 |
| 10 pct | 4.52 |
| 20 pct | 9.43 |
| 30 pct | 11.25 |
| 50 pct | 16.54 |
| 80 pct | 28.37 |
| large | 45.22 |

### **Graficas**



## **Análisis**

En este requerimiento se usó una diccionario ya que observamos que al momento de usarlo así era mucho más fácil solucionar el problema de encontrar rápidamente el valor de N a su vez recorrimos el catalogo inicial buscando el sexo de la víctima donde después con un .get de la implementación del arraylist sacamos el código del crimen y la fecha, encontramos que al comparar dos criterios y organizarlos se encuentran de una manera mas sencilla los crímenes.