# **Observaciones del reto 2**

Ernesto Perez – 202112530 – ec.perez@uniandes.edu.co

Nicolás Saavedra – 20212963 - [n.saavedrag@uniandes.edu.co](mailto:n.saavedrag@uniandes.edu.co)

[**https://github.com/EDA2021-2-SEC01-G07/Reto3-G-07**](https://github.com/EDA2021-2-SEC01-G07/Reto3-G-07)

Requerimiento 3: Nicolás Saavedra

Requerimiento 4: Ernesto Perez

# **Análisis de complejidad**

# **Pruebas de rendimiento**

Especificaciones de la máquina de prueba

|  |  |
| --- | --- |
|  |  |
| Procesadores | AMD Ryzen 5 3600 6-Core Processor, 3593 Mhz, 6 Core(s), 12 Logical Processor(s) |
| Memoria RAM | 16GB |
| Sistema operativo | Microsoft Windows 10 pro 64-bits |

**Análisis de complejidad**

|  |  |  |
| --- | --- | --- |
| Operación | Complejidad temporal | Justificación |
| Cargar Datos |  |  |
| Requerimiento 1 |  |  |
| Requerimiento 2 |  |  |
| Requerimiento 3 |  |  |
| Requerimiento 4 |  |  |
| Requerimiento 5 |  |  |

**Tiempo de ejecución promedio por requerimiento y crecimiento temporal**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tiempo de ejecución para cada función [ms]** | | | | | |
| **Porcentaje de datos** | **Cargar datos** | **Req1** | **Req2** | **Req3** | **Req4** | **Req5** |
| **10%** |  | 31.25 | 0.0 |  | 0.0 |  |
| **30%** |  | 62.5 | 0.0 |  | 0.0 |  |
| **50%** |  | 83.33 | 0.0 |  | 3.91 |  |
| **80%** |  | 130.21 | 0.0 |  | 11.72 |  |
| **100%** |  | 156.25 | 10.42 |  | 15.63 |  |

*En las funciones con entradas se usaron los mismos datos de entrada de los ejemplos.*

**Graficas de tiempo de ejecución promedio por requerimiento y crecimiento temporal**

**Complejidad temporal requerimiento 1:** O(n)

El requerimiento 1 tiene una complejidad lineal que se evidencia muy bien en el comportamiento de la gráfica. Esto se debe ya que se necesita conocer la ciudad con más avistamientos y la cantidad de avistamientos para esa ciudad por lo que ese necesario recorrer todo el árbol con índices de ciudades para poder conocer que tantos avistamientos hubo para dicha ciudad y de esta manera conocer cuál es la ciudad con mayor cantidad de avistamientos.

**Complejidad temporal requerimiento 2:** O(n)

Es poco intuitivo saber la complejidad temporal a partir de la gráfica pues el intervalo de duración tiene que ser muy grande para poder ver un comportamiento lineal. Sin embargo, la función para sacar las llaves dado un rango tiene complejidad en el peor caso de O(n), una vez se saca este rango de valores el código es mucho más eficiente pues solo saca los 3 primeros y últimos en O (1).

**Complejidad temporal requerimiento 4:** O(n)

Este requerimiento función similar al requerimiento 2 y es más fácil ver en la gráfica un comportamiento lineal para la complejidad temporal. Esto se debe a que la función para sacar las llaves dado un rango tiene complejidad en el peor caso de O(n), una vez se saca este rango de valores el código es mucho más eficiente pues solo saca los 3 primeros y últimos en O (1).