

## Reto No. 3

### UFOS SIGHTINGS

#### INTEGRANTES

Estudiante 1 Mateo López Cespedes

Cod 202014481

Correo [m.lopez23@uniandes.edu.co](mailto:m.lopez23@uniandes.edu.co)

Estudiante 2 Valentina Perea

Cod 202013095

Correo [v.peream@uniandes.edu.co](mailto:v.peream@uniandes.edu.co)

Link repositorio:

<https://github.com/EDA2021-2-SEC03-G17/Reto3-G17.git>

#### REQUERIMIENTOS INDIVIDUALES:

-REQ2: Valentina Perea Márquez

-REQ3: Mateo López Cespedes

#### AMBIENTES DE PRUEBA

MAQUINA	1: Mateo López	2: Valentina Perea
Procesadores	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz	AMD Ryzen 5 3400G with Radeon Vega Graphics 3.70
Memoria RAM (GB)	8,00 GB	16,0 GB
Sistema Operativo	Windows 10 Home	Windows 10 Pro 2004 64 bits

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

#### COMPLEJIDAD DE REQUERIMIENTOS:

REQUERIMIENTO	PROMEDIO	ESPECIFICACION
1	$O(n+2\log(n))$	Se hace una copia del mapa cuya llave corresponde a la ciudad ingresada por parámetro, para esto se usa KeySet y getValue. Luego para hallar los 3 primeros y 3 ultimos se usa 3 veces MinKey y 3 veces MaxKey
2	$O(2n+2n\log(n))$	Se pide el size de un árbol, luego un maxKey, luego un values para hallar valores dentro de un rango de los cuales se saca una muestra usando dos iteradores que ingresan a dos addlast y un shellsort. Y una última iteración para hallar la cantidad total de elementos.
3	$O(2\log(n)+2n)$	Se usa values para filtrar el árbol de acuerdo a los limites pasados por parámetro. Luego para hallar los 3 primeros y los 3 últimos se usa 3 veces minKey y 3 veces Maxkey. Finalmente, para hallar los avistamientos más tardíos, se hace una copia del árbol RBT que maneja como llave la hora usando keyset y se usa 5 veces maxKey para hallar los avistamientos más tardíos.
4	$O(2n+2n\log(n))$	Se pide el size de un árbol, luego un minKey, luego un values para hallar valores dentro de un rango de los cuales se saca una muestra usando dos iteradores que ingresan a dos addlast y un shellsort. Y una última iteración para hallar la cantidad total de elementos
5	$O(2\log(n))$	Se realiza dos veces .value() con el fin de filtrar primero la longitud y segundo la latitud. Luego para hallar los 5 primeros y los 5 últimos se realiza el mismo proceso que en requerimientos pasados y se usa 5 veces Minkey y 5 veces Maxkey
6	$O(2\log(n))$	Hace exactamente los mismo que el requerimiento 5 solo que ahora hace uso de la librería folium que a partir de coordenadas y diferentes valores de formato presenta un mapa

Tabla 2. Complejidades de los requerimientos en notación BigO.

NOTA: El requerimiento 6 genera un archivo html en la carpeta del Reto y que al abrirse muestra el mapa

