

Análisis de resultados

Integrantes:

- **Est -1.** Jesus M Otero
B 202410261 jm.otero@uniandes.edu.co
- **Est-2.** Carlos A Amado A 202410873
c.amadoa@uniandes.edu.co

Req_1:

- `mp.value_set(catalog)` recorre todos los valores del mapa, lo cual es $O(n)$, donde n es el número de películas en el catálogo.
- las películas en el catálogo y comparas el título y el idioma original, lo cual también es $O(n)$.
- **Complejidad total: $O(n)$.**

Tamaño datos	Tiempo de ejecución (ms)
45031	60.33 ms
45031	80.24 ms
45031	64.39 ms
14860	30.23 ms
14860	18.36 ms
14860	25.42 ms

Req_4:

Hecho por Jesús Otero

- De nuevo, `mp.value_set(catalog)` tiene una complejidad de $O(n)$.
- El bucle que compara las fechas y el estado de las películas también es $O(n)$.
- El ordenamiento con Merge Sort tiene una complejidad de $O(n \log n)$.
- **Complejidad total: $O(n \log n)$.**

Req_6:

- `mp.value_set(catalog)` es $O(n)$.
- Luego, recorres las películas y filtras las que cumplen con las condiciones, lo cual es $O(n)$.
- El bucle que agrega elementos al diccionario `dicc_anios` y actualiza los valores es $O(n)$.
- **Complejidad total: $O(n)$.**

Req_7:

- `mp.value_set(catalog)` es $O(n)$.
- Luego, se recorren las películas y las compañías productoras, lo que es $O(n*m)$, donde m es el número de compañías productoras por película.

- El bucle que actualiza el diccionario `dicc_anios` es $O(n)$.
- **Complejidad total: $O(n*m)$.**