**Análisis de complejidad**

**Requerimiento 1:**

Evidenciamos que la complejidad función del requerimiento 1 es O(N(log(N)) lo que sería linearitmica, debido a que si bien usamos un ciclo que sería O(N), implementamos una función de ordenamiento por MergeSort lo cual la hace O(Nlog(N) al ser mayor que el crecimiento temporal lineal.

**Requerimiento 2:**

Nuestro requerimiento 2 posee una complejidad de O(N^2) debido a que usamos un doble recorrido el cual es el encargado de buscar autores y compararlos con el ConstituentID. Adicionalmente usamos un ordenamiento merge el cual tiene O(Nlog(N)) lo que lo hace menor a O(N^2).

**Requerimiento 3:**

Nuestro requerimiento 3 tiene una complejidad O(N^2) debido a que se implementa un doble recorrido encargado de comparar el nombre del autor con el ConstituentID de la obra para obtener el medio de la misma, adicionalmente se usó un merge O(Nlog(N)) para realizar el ordenamiento de la cantidad de veces que aparecía un medio en forma de contador y finalmente imprimir los datos

**Requerimiento 4:**

El en requerimiento 4 se evidencia una complejidad O(N) debido a la implementación de un solo recorrido y comparación suma en el contador de cada una de las nacionalidades.

**Requerimiento 5:**

Hallamos una complejidad en nuestro algoritmo de del requerimiento 5 de O(Nlog(N)) debido a que usamos principalmente dos recorridos por separados es decir O(N) y una función de ordenamiento merge lo que la haría O(Nlog(N)) debido a que esta tiene a ser mayor que O(N).

**Requerimiento 6:**

Hallamos una complejidad lineal O(N) en nuestro algoritmo del requerimiento 6 debido a que encontramos que no hubo necesidad de implementar una función de ordenamiento y usamos dos ciclos independientes durante la creación de esta función.

**Pruebas de rendimiento**

