

COMPLEJIDAD REQUERIMIENTOS

1. Requerimiento 1

$O(n*m) + O(n)$

Esto debido a la iteración de dos datos completamente diferentes

2. Requerimiento 2

$O(m*p*n)$

3. Requerimiento 3 - ELABORADO POR DANIEL MOLANO

$O(n)$

4. Requerimiento 4

$O(n)$

5. Requerimiento 5

$O(n)$

6. Requerimiento 6 (Bono)

$O(n*m)$

TABLAS DE TIEMPOS

- Daniel Molano

CPU Ryzen 1200

RAM 16gb 2400

	REQ 1	REQ 2	REQ 3	REQ 4	REQ 5	Bono
Small	78,125	390,625	0	15,625	1109,375	46,875
5%	109,375	2671,875	0	31,25	86000	203,125
50%	125	9578,125	31,25	62,5		1546,875
Large	140,625	11.609	78,125	109,375		34968,75



- Germán Moreno
CPU Intel core i3 3240
RAM 8gb 800Mhz

	small	5pct	50pct	large
req1	78,13	125,00	140,63	187,50
req2	437,50	3203,13	11078,13	17046,87
req3	15,63	15,63	31,25	93,75
req4	15,63	31,25	78,13	109,38
req5	1218,75	97734,38	1899957,42	3809414,63
req6	62,50	171,88	843,75	1687,50



COMPARACION COMPLEJIDADES

-Req 1

$O(2n)$ debido a que se recorre con un For In solo una vez cada artista en el catálogo.

-Req 2

$O(4n^3+n)$ debido al for que recorre los ArtWork, y que dentro de este se recorre con otro for el ArtWork como tal buscando el constituent ID y los IsPresent que se utilizan.

-Req 3

$O(n*m)$ debido a que hay 2 for recorriendo listas distintas.

-Req 4

n = cantidad de Artwork

m = cantidad de Artistas en el artwork

k = cantidad de Artistas del catálogo

f = cantidad de nacionalidades

Complejidad = $O(n*m*k*f + f \log(f) + n)$ Debido a que se recorre n , luego por cada n se recorre un m y por cada m usa un isPresent con referencia a k y otro isPresent con referencia a f . Por último se realiza un ordenamiento shellsort sobre f .

-Req 5

$O(n + 2 \log(n) + n)$ Debido a que solo se recorre una vez la lista de Artworks para calcular el valor y se ordena 2 veces.

Al comparar las complejidades del reto 1 con las actuales podemos observar cómo algunas iteraciones son eliminadas al utilizar maps en cada uno de los requerimientos, así mismo esto permite comprobar la optimización que se ve en los tiempos.