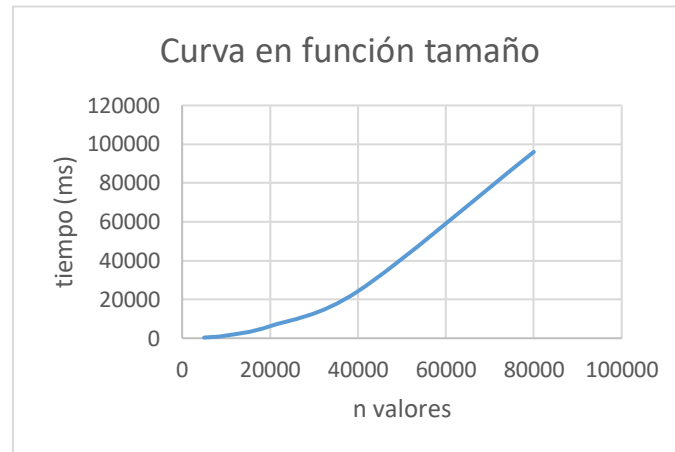


FACTOR1: TAMAÑO DEL PROBLEMA

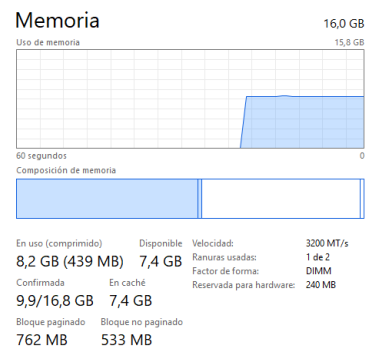
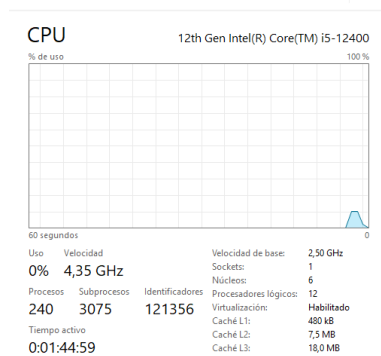
PROG / TIEMPO	5000	10000	20000	40000	80000
PythonA1.py	337	1387	6210	24234	96061



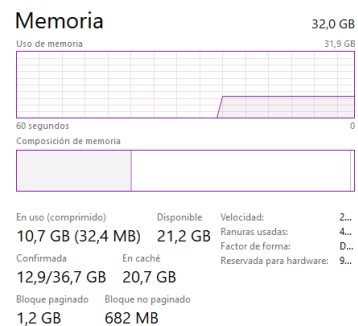
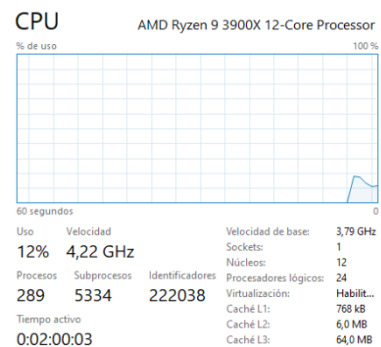
FACTOR2: POTENCIA DEL ORDENADOR

PC / TIEMPO	5000	10000	20000	40000	80000
PC 1	337	1387	6210	24234	96061
PC 2	574	2329	9460	37711	152489

PC 1



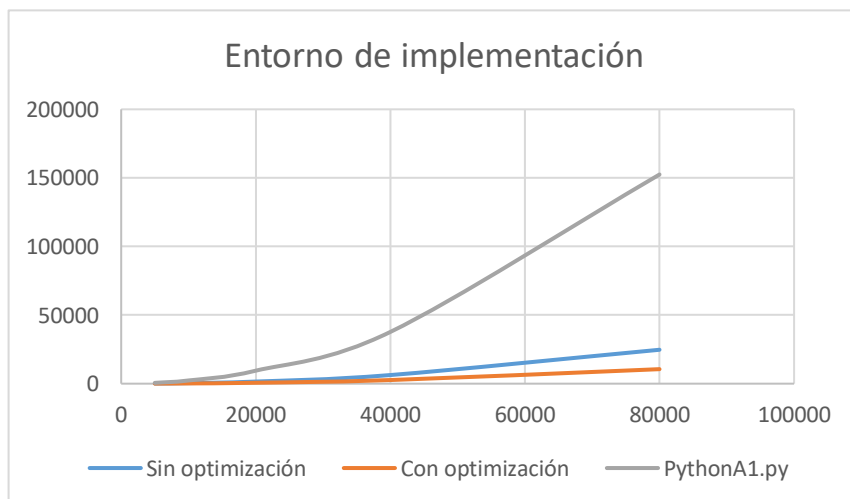
PC 2





FACTOR3: ENTORNO DE IMPLEMENTACIÓN

JavaA1.java / TIEMPO	5000	10000	20000	40000	80000
Sin optimización	106	391	1533	6273	24703
Con optimización	53	165	659	2640	10570
PythonA1.py	574	2329	9460	37711	152489



Se puede observar cómo ambas ejecuciones en Java son mucho más rápidas que el programa Python, así mismo ejecutando el programa con el JIT de java (optimizado) es prácticamente el doble de rápido que sin optimizar.

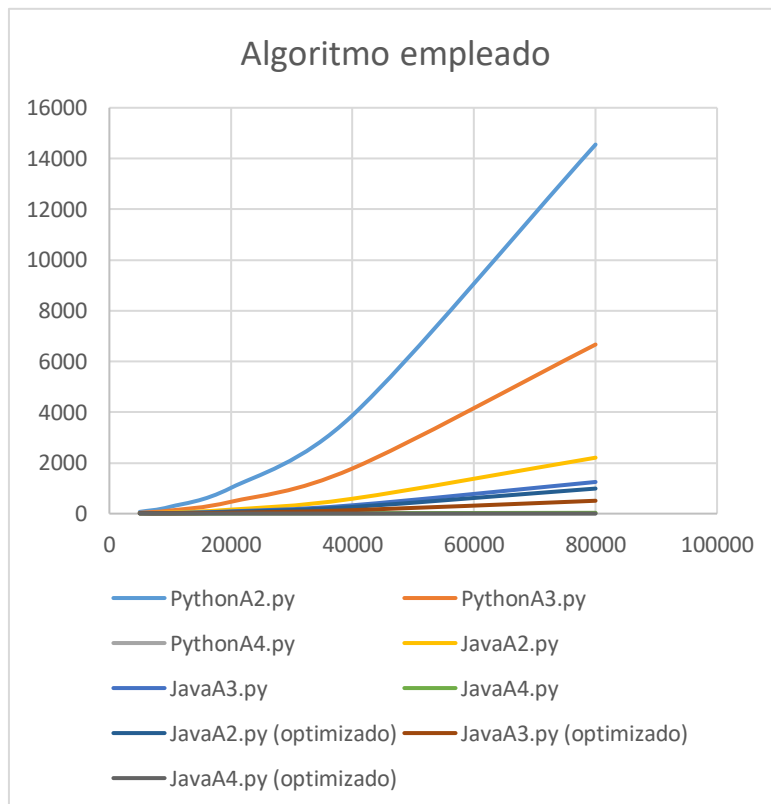
FACTOR4: ALGORITMO EMPLEADO

Python

PROG / TIEMPO	5000	10000	20000	40000	80000
PythonA2.py	70	269	1015	3874	14550
PythonA3.py	33	123	467	1781	6667
PythonA4.py	1	1	3	6	11

Java

PROG / TIEMPO	5000	10000	20000	40000	80000
JavaA2.py	22	44	158	588	2205
JavaA3.py	17	25	89	329	1248
JavaA4.py	12	4	8	17	33
JavaA2.py (optimizado)	17	20	71	263	989
JavaA3.py (optimizado)	15	10	38	137	508
JavaA4.py (optimizado)	12	2	4	2	3



De los datos obtenidos podemos observar que el algoritmo que se emplee marcara mucho la diferencia del tiempo de ejecución, siendo los algoritmos A4 los mejores, incluso el de java se puede mejorar su tiempo si le incorporamos la optimización. El mejor de todos a sido el PythonA4.