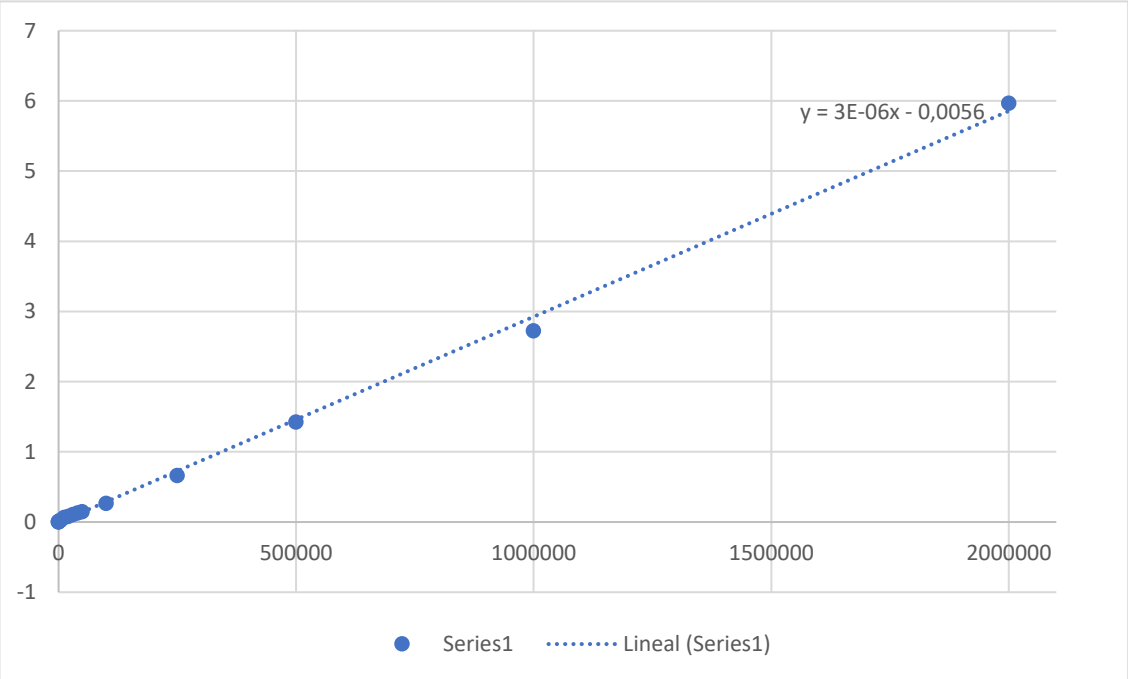
ANALISIS PRÁCTICO

ejemplo.txt	**0,005**	**15**
test100.txt	**0,006**	**100**
test500.txt	**0,008**	**500**
test1000.txt	**0,009**	**1000**
test2000.txt	**0,016**	**2000**
test5000.txt	**0,03**	**5000**
test12000.txt	**0,066**	**12000**
test20000.txt	**0,079**	**20000**
test30000.txt	**0,107**	**30000**
test40000.txt	**0,13**	**40000**
test50000.txt	**0,149**	**50000**
test100000.txt	**0,269**	**100000**
test250000.txt	**0,666**	**250000**
test500000.txt	**1,427**	**500000**
test1000000.txt	**2,728**	**1000000**
test2000000.txt	**5,97**	**2000000**



Las anteriores etapas de orden $O(n^2)$ ahora se han unificado mediante el algoritmo disjointSet (búsqueda-unión) y ahora al realizar todas las operaciones mediante HashTable, su orden máximo será de O(n).
Realmente hay alguna parte del algoritmo cuyo orden es de orden O(log n) pero se tomará como orden O(1). Esto lo podemos tomar así gracias al teorema que dice que si unimos el grumo pequeño con el más grande los niveles que se forman en el árbol tendrán un orden de complejidad de O(log n) pero como es tan pequeño se puede aproximar a O(1) por lo tanto el orden final del programa en sí será de O(n).