

ANÁLISIS EFICIENCIA PRÁCTICA 1.

Evaluación de la eficiencia respecto al tiempo, mediante medidas de tiempo utilizando distintos tamaños del fichero de entrada.

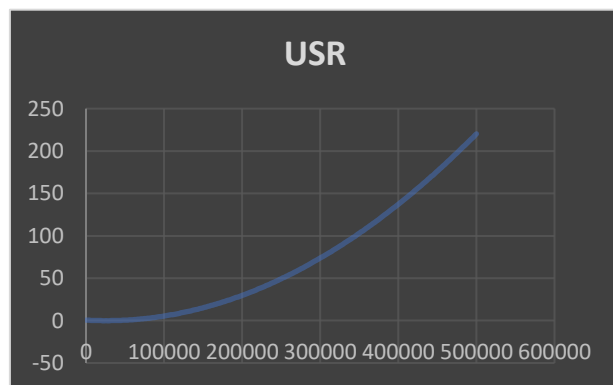
Se analizará el análisis correspondiente a las etapas 2, 3 y 4 del algoritmo, las cuales son: creación de la lista de usuarios, creación de la lista de grupos y ordenación y selección de los grupos.

Tomadas las respectivas medidas, mediante Excel, se obtienen estos resultados con y podemos corroborar el orden de magnitud que tienen las siguientes etapas.

Creación de la lista de usuarios.

Respecto a su análisis sobre la complejidad obtenemos lo siguiente:

ejemplo.txt	0	15
test100.txt	0,001	100
test500.txt	0,003	500
test1000.txt	0,01	1000
test2000.txt	0,012	2000
test5000.txt	0,036	5000
test12000.txt	0,098	12000
test20000.txt	0,237	20000
test30000.txt	0,486	30000
test40000.txt	0,861	40000
test50000.txt	1,431	50000
test100000.txt	7,689	100000
test250000.txt	47,157	250000
test500000.txt	220,652	500000



```
public static void listaUsr(ArrayList<Integer> usr, ArrayList<listaUsuarios> red, int numConex) {
    for(int i = 0; i<numConex;i++) {
        int usr1 = red.get(i).getUsr1();
        int usr2 = red.get(i).getUsr2();
        if (!usr.contains(usr1))
            usr.add(usr1);
        else if (!usr.contains(usr2))
            usr.add(usr2);
    }
}
```

Como los add y son $O(1)$ y los contains $O(i)$.

$$\sum_{i=0}^n O(1) + O(1) + O(i) + O(i) = \sum_{i=0}^n O(i)$$

Mediante la fórmula que dice:

$$\sum_{i=c1}^{n+c2} O(i^a) = O(n^{a+1}) \Rightarrow \sum_{i=0}^n O(i) = O(n^{1+1}) = O(n^2)$$

En el peor de los casos en cambio, entrará en una sola comparación y por lo tanto será $O(1)$.