Tablas 1 y 2.

Se utilizaron los archivos \*small.csv para las pruebas.

|  |  |  |
| --- | --- | --- |
| Fuente de Datos | ArrayList | LinkedList |
| Películas | 0.0625 | 0.046875 |
| Elenco | 0.03125 | 0.03125 |
| SelectionSort | 9.875 | 1082.015625 |
| InsertionSort | 9.875 | 1233.21875 |
| ShellSort | 0.34375 | 35.203125 |

Preguntas.

* ¿Qué diferencias se observan en desempeño de la carga de datos entre arreglo (Arraylist) y lista sencillamente encadenada (Singlelinkedlist)?

La lista sencillamente encadenada LinkedList tiene una mayor velocidad al momento de cargar los archivos .csv, pero no en los demás procesos.

* ¿Cuál de las dos implementaciones (Arraylist y Singlelinkedlist) tiene mejor desempeño? Y ¿Por qué?

ArrayList, ya que al realizar las pruebas de velocidad en las funciones *sort*, se dio una inmensa diferencia entre los tiempos de ordenamiento.

* ¿Qué diferencias existen entre cargar los archivos de películas (MoviesDetailsCleaned) y elenco (MoviesCastingRaw)?, ¿Por qué se presentan estas diferencias?

MoviesDetailsCleaned tarda más que MoviesCastingRaw y esto se debe a que tiene más columnas.

* ¿Qué diferencias en el desempeño se observan entre los tres algoritmos de ordenamiento?

ShellSort fue definitivamente el más rápido, pues, a pesar de que se tratara de 2000 lecturas, lo obtuvo en menos de un minuto.

* ¿Qué efectos tienen los dos tipos de lista en los tres algoritmos de ordenamiento?

El efecto que tiene ArrayList parece ser mucho más adecuado para este tipo de funciones (*sort*), mientras que LinkedList hasta 20 minutos en recopilar las 2000 lecturas y organizarlas.