**Laboratorio EDA 2**

*María Paula Gonzales Escallon:* [*m.gonzaleze@uniandes.edu.co*](mailto:m.gonzaleze@uniandes.edu.co) *202012265*

*Jessica Alejandra Robles Moreno:* [*j.roblesm@uniandes.edu.co*](mailto:j.roblesm@uniandes.edu.co) *202013355*

*Martin Ubaque Forero:* [*m.ubaque@uniandes.edu.co*](mailto:m.ubaque@uniandes.edu.co) *201923281*

*Grupo 8*

1. Desempeño de las implementaciones de Lista (arreglo y lista encadenada) al cargar los datos \*–small.csv del reto

Datos Single Linked list

Películas (MoviesDetailsCleaned) 13265.625, 17218.75, 15796.875, 14390.625, 14765.625, 14156.25

Elenco (MoviesCastingRaw) 6734.375, 8359.375, 10468.75, 9000.0, 7046.875, 8734.375

Datos Array list

Películas (MoviesDetailsCleaned) 15718.75, 17375.0, 18796.875, 15828.125, 19984.375, 18937.5

Elenco (MoviesCastingRaw) 9703.125, 11328.125, 11140.625, 13656.25, 10640.625, 12500.0

|  |  |  |
| --- | --- | --- |
| **Fuente de datos** | **Arraylist [ms]** | **Singlelinkedlist [ms]** |
| Películas (MoviesDetailsCleaned) | 17773.4375 | 14932.29 |
| Elenco (MoviesCastingRaw) | 11494.7916 | 8390.625 |

Tabla 1. Tabla de rendimiento para cargar archivos como lista enlazada y arreglo.

|  |  |  |
| --- | --- | --- |
| **Algoritmo** | **Arraylist [ms]** | **Singlelinkedlist [ms]** |
| SelectionSort | 6176.69 | Muchísimo tiempo  (> 10 horas) |
| InsertionSort | 6531.25 | 10 horas |
| ShellSort | 14625 | Mucho tiempo |

*Tabla 2. Tabla de rendimiento para los distintos algoritmos de ordenamiento en listas enlazadas y arreglos.*

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

Con el arreglo *Arraylist* la carga de archivos se demora más que con *Singlelinkedlist.* Esto lo podemos notar tanto al cargar el archivo de películas como el de casting.

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

La que tiene mejor desempeño es *Singlelinkedlist.* Esto es debido a que, al momento de agregar datos, un arreglo tiene que acomodar sus datos para hacer espacio para los nuevos. Mientras que la lista encadenada no necesita acomodar sus datos para agregar elementos debido a que la información entrante se guarda en cualquier espacio de la memoria.

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

*MoviesDetailsCleaned* se demora más tiempo cargando que *MoviesCastingRaw* tanto en *Arraylist* como en *Singlelinkedlist.* Esto puede ser debido a que a pesar de que ambas listas tengan la misma cantidad de elementos (2000), la lista *MoviesDetailsCleaned* tiene elementos más largos ya que tiene que almacenar las descripciones de las películas, a diferencia de *MoviesCastingRaw* que todos los elementos son cortos.

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

Con el algoritmo *ShellSort* *e*l ordenamiento es el menos demorado de los tres algoritmos. Esto es debido a que presenta un orden de complejidad menor que los otros dos algoritmos. Mientras que los otros dos en su peor caso ambos algoritmos presentan el mismo orden de complejidad, el algoritmo *InsertionSort* es menos demorado que el algoritmo *SelectionSort,* ya que en sus mejores casos su orden de complejidad es menor que el de *SelectionSort.*

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

El *Arraylist d*emora menos en ordenar los datos en los 3 algoritmos ya que se necesita recorrer toda la lista y al saber la posición de los datos lo hace más fácil que el *SingleLinkedList,* que tiene que buscar la siguiente posición de la lista.