

Trabajo Pratico Especial

Parte 1

Materia: Programacion 3.

Integrantes: Gonzalez,J. Benjamín; Cabrer Lombardi, Paul J.

Descripción del trabajo realizado:

Para resolver el problema, realizamos 9 aplicaciones, usando 3 estructuras diferentes :

- con un arreglo de tamaño de 500.000 de usuarios.
- con un arreglo de tamaño de 1 millón de usuarios.
- con un arreglo de tamaño de 3 millones de usuarios.
- con una lista que se carga al final con 500.000 usuarios.
- con una lista que se carga al final con 1 millón de usuarios.
- con una lista que se carga al final con 3 millones de usuarios.
- con una lista que se carga al principio con 500.000 de usuarios.
- con una lista que se carga al principio con 1 millón de usuarios.
- con una lista que se carga al principio con 3 millones de usuarios.

Para las nueve aplicaciones utilice 4 clases por cada una:

- Reader: posee un método que devuelve un array string llamado reader(). Se encarga de importar el "dataset_'numeroDeUsuario'.csv" a la estructura de usuarios (arreglo/lista) dentro de la aplicación.

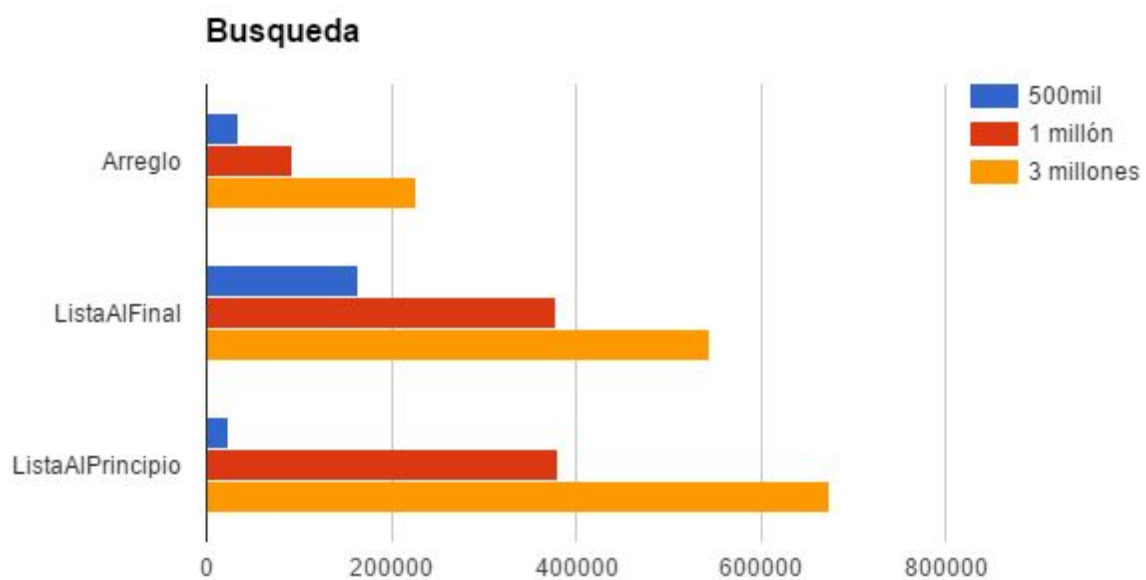
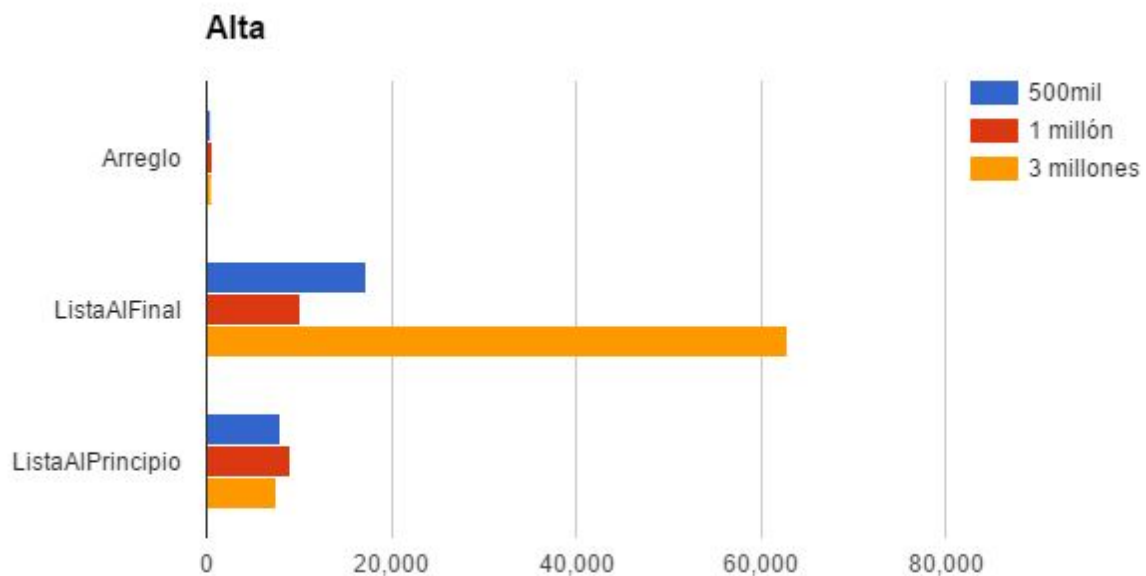
- WriterRegister: la cual contiene un método writerRegister() que recibe como parámetro una estructura de tipo usuario(arreglo/lista). Crea un archivo "salidaAlta.csv" que contiene 10000 usuarios y los tiempos que tardaron en insertarse en la estructura tipo usuario.

- WriterSearch : la cual contiene un método writerSearch() que recibe como parámetro una estructura de tipo usuario(arreglo/lista). Crea un archivo "salidaBusqueda.csv" que contiene 10000 usuarios , los tiempos que tardaron en insertarse en la estructura tipo usuario y además un booleano que indica si se encontró el usuario o no dentro de dicha estructura.

- Test(cantidad de usuarios)Usuarios : Contiene el método main ,este se encarga de primero crear una estructura de datos con la cual se

quiera trabajar , a esta estructura se le carga el dataset y luego genera los 2 archivos (.csv).

Comparación de los tiempos de “alta” y “búsqueda” en las diferentes estructuras:



Llegamos a la conclusión que utilizar arreglos es más efectivo a la hora de la Alta y la búsqueda en la mayoría de las ocasiones, debido a que en la alta para agregar un elemento, solo se tiene que conocer la posición y no es necesario tener que recorrer toda la estructura como en la lista, y que en la búsqueda es mucho más rápido iterar sobre una estructura de longitud conocida en comparación a una que no se conoce como en el caso de la lista