Facultad de Ciencias Exactas

Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

Informe de proyecto:

Trabajo Practico Especial Programación 3 Primer entrega

Bianco, Manuel*: manucobianco@gmail.com*
Rampoldi, Santiago*: santirampoldi12@gmail.com*

Fecha de entrega: Viernes 04/05/2018

Introducción al proyeto:

En este proyecto se nos pide implementar una herramienta que simplifique la búsqueda dentro de una colección de libros, permitiendo realizar dicha búsqueda por genero. La colección de libros se nos es dada en un archivo .csv con un formato predefinido y nosotros debemos llevarlo a memoria, armando un índice por género que permita buscar mas rápidamente.
Los libros poseen atributos fijos que serán especificados en el .csv provisto por los profesores con el siguiente formato:

Nombre, Autor, Cantidad de Paginas, Géneros (Pueden ser varios).

Se le pide a el usuario que ingrese un género de su elección y luego se genera un nuevo archivo .csv, de nombre ListadoDeLibros que contenga los nombres de todos los libros pertenecientes a ese género. En el caso de que el género ingresado por el usuario no exista dentro de la colección, el archivo generado estará vacío.

También en la consigna se nos pide realizar discusiones y análisis respecto a las estructuras disponibles para llevar a cabo el proyecto y su injerencia en los costos de tiempo e interacciones en las distintas partes del proyecto. En base a esto debemos elegir la estructura que mejor funcionalidad nos provea y justificar nuestra elección en este informe.

Géneros:

Usamos la Arraylist predefinida de java por la poca cantidad de elementos que puede contener ahora y que no incrementaría en gran número en el futuro, en el caso de querer agregar un género nuevo hay que hacer pocos corrimientos y es una lista que a futuro no va a crecer exponencialmente.

Decidimos aplicar la búsqueda binaria ya que al tener que agregar una gran cantidad de libros, recorrer toda la lista O(n) cada vez que agregamos un libro tomaría bastante más tiempo. En cambio con la búsqueda binaria aplicada tendríamos O(log n) que llevado a una gran cantidad de libros significa una diferencia importante en cuanto a tiempo y una muy pequeña en cuanto al desarrollo del código.

Libros:

Utilizamos una lista simplemente vinculada implementada por nosotros en una clase Lista, ya que al manejar una gran cantidad de libros nos es muy conveniente que al insertar un elemento nuevo, la injerencia sea O(1) y no haya que hacer un corrimiento que podría conllevar millones de interacciones.

En cuanto al orden insertamos siempre primero, ya que no tenemos necesidad de que está lista quede ordenada según ningún criterio y eligiendo este método de inserción nos evitamos recorrer la lista innecesariamente.

Tiempos:
En todos los casos, los tiempos de búsqueda son mínimos, ya que el método encargado solo debe devolver una lista que ya está armada.
En cuanto al tiempo uno de los procesos que más consume es llevar esas listas al archivo .csv de salida, lo cual es bastante fijo ya que no hay grandes cambios que se puedan realizar desde el código. El proceso que más tiempo consume es leer los datos provistos en el archivo .csv de entrada y llevarlos a memoria, armando en ese mismo proceso las listas. Para reducir este tiempo al mínimo posible, además de implementar el índice de géneros pedido en la consigna, implementamos una búsqueda binaria, porque aunque sea una cantidad relativamente pequeña de géneros, al insertar millones de libros hay una diferencia de tiempo considerable. Creemos conveniente armar la listas de libros que hay en cada género una sola vez, que tener que armarlas cada vez que nos la piden, lo cual llevaría mucho más tiempo y más interacciones. En el caso de agregar libros nuevos este proceso se hace sin tener que agregar todos los libros de vuelta.

Tiempo de inserción sin búsqueda binaria:

1. Csv1. Entre 5 y 8 milisegundos
2. Csv2. Entre 32 y 40 milisegundos
3. Csv3. Entre 450 y 1500 milisegundos
4. Csv4. Entre 9400 y 16000 milisegundos
   (Entre 9,4 y 16 segundos)

Tiempo de inserción con búsqueda binaria:

1. Csv1. Entre 5 y 8 milisegundos
2. Csv2. Entre 30 y 40 milisegundos
3. Csv3. Entre 450 y 550 milisegundos
4. Csv4. Entre 8000 y 15500 milisegundos
   (Entre 8 y 15,5 segundos)

Las ventajas de aplicar búsqueda binaria se vuelven mas notables a medida que crece la cantidad de libros ingresados, como se puede apreciar en el gráfico abajo.

Tiempo de escritura (armado y creación del archivo .csv de salida): Considerando que el género ingresado existe:

Csv1. Entre 1,5 y 2 milisegundos
Csv2. Entre 2 y 2,5 milisegundos
Csv3. Entre 14 y 1000 milisegundos
Csv4. Entre 70 y 110 milisegundos

Si el género ingresado no existe, el tiempo es siempre de entre 1,5 y 2 milisegundos, ya que el archivo generado no va a contener ningún libro.

El proceso de buscar la lista de libros del género ingresado es siempre de 0,2 milisegundos porque la lista de libros ya existe en cada género y solo se debe devolver eso.

En conclusión vimos que a medida que agregamos más datos a analizar los tiempos se diferencian mucho más entre siendo la búsqueda binaria mucho más eficiente que la búsqueda normal, aunque con poca cantidad de datos las diferencias son casi minimas.