

Trabajo Práctico Especial

Programación 3

PRIMERA ETAPA:

Buscador de libros por géneros.



Carrera:

TUDAI

Autores:

González, Fernanda Magali (fgonzalez@alumnos.exa.unicen.edu.ar)

Martino Martinez, Fabián (fabian.martino.m@gmail.com)

Fecha de entrega:

04/05/2018

Contenido

INTRODUCCIÓN	3
Primera Etapa	3
DESARROLLO DEL TRABAJO	4
Discusión y análisis	4
Decisiones de diseño	4
Resultados obtenidos en pruebas	5
Volcado de datos en gráfico	6
CONCLUSIONES	7

INTRODUCCIÓN

Primera Etapa

Partiendo de una colección de libros, se nos pidió implementar una herramienta que permita simplificar la búsqueda de libros por géneros. Esta comenzará llevando a memoria la colección completa de libros para luego realizar un filtrado por un género dado, presentando al usuario la colección de libros resultante.

Para optimizar el proceso de búsqueda, se debió implementar un índice por género, el cual simplifica el acceso a solo un subconjunto de todos los libros existentes.

DESARROLLO DEL TRABAJO

Discusión y análisis

Para esta primera etapa podíamos utilizar distintas estructuras como Listas, Arreglos, Arboles o incluso combinaciones de estas para la implementación de la herramienta.

Por un lado debíamos elegir una estructura que almacene los libros dados, la cual no nos interesa si esta ordenada o no, por esto último el árbol no nos era de utilidad.

Por otro, se debía generar una estructura con los géneros, a los que puede pertenecer un libro, la cual se debía decidir si queríamos que estuviese ordenada porque se le debía aplicar una búsqueda. Para esta parte podíamos elegir, también, entre varias estructuras, incluida la lista simplemente vinculada que habíamos implementado nosotros. Dado que entre los miembros del grupo contábamos con dos implementaciones de esta última, decidimos descartar esta opción.

Decisiones de diseño

Las decisiones que se tomaron fueron:

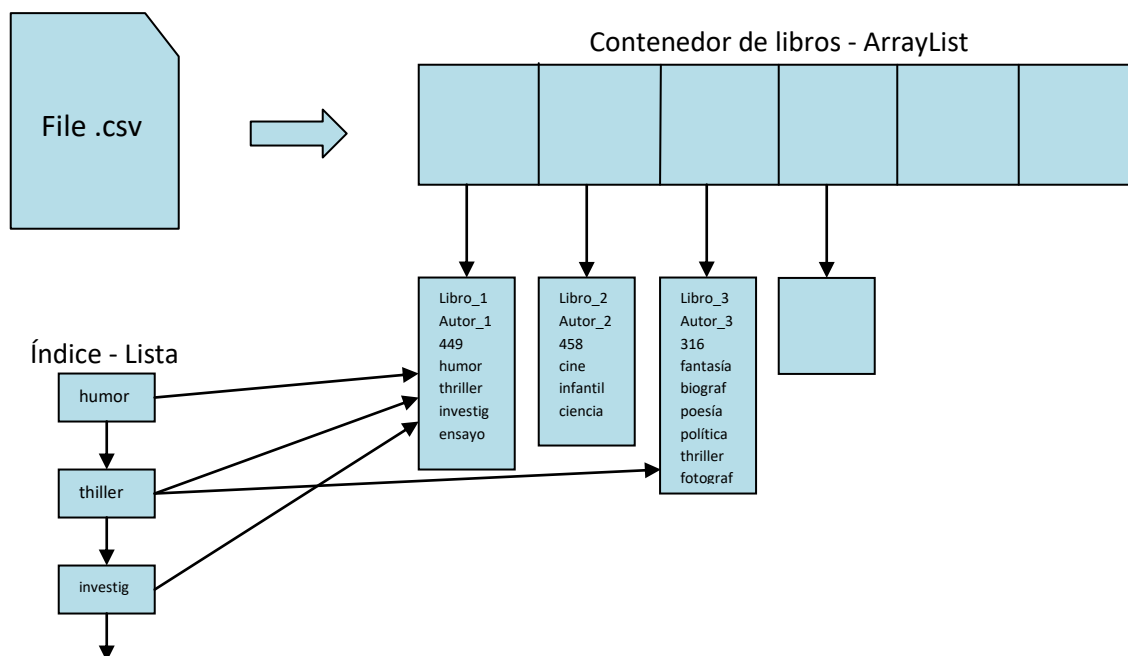
- Utilizar un ArrayList, de la interface List de Java, como estructura principal para el volcado en memoria de los libros. Se la eligió porque, como no se debe realizar búsquedas, la complejidad para la inserción es de $O(1)$.
- Utilizar una lista que se llame a sí misma (recursiva), como estructura principal para el acceso por género. Los datos se insertan, sin ordenar, siempre al final de la lista por lo que su complejidad es $O(n^1)$. Para la búsqueda, como los elementos no están ordenados, la complejidad es también $O(n)$.

La implementación la pensamos así:

- Una clase Libro que contenga un arreglo con los géneros que posea cada libro.
- Una clase Colección que sería donde se van a guardar los libros.
- Una clase Índice que guardaría los géneros nuevos que va encontrando y asociaría el libro que pertenece a ese género.
- Una clase Main que levantaría los datos del archivo, los guardaría en la estructura y generaría el archivo de salida.

¹ Siendo n cantidad de elementos.

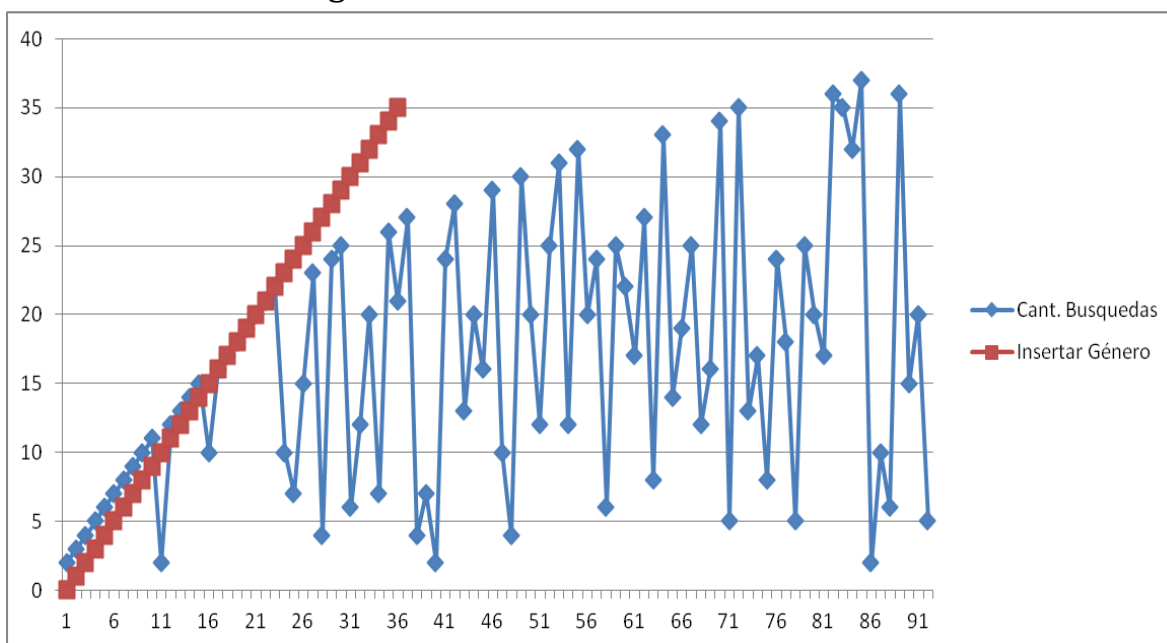
Esquema que representa la implementación:



Resultados obtenidos en pruebas

Durante las pruebas realizadas a la herramienta vimos que la forma en que la desarrollamos no fue la más eficiente. Para el agregado de géneros en el Índice se observa un incremento lineal mientras que para la búsqueda de todos los géneros de cada libro varía según el orden en el que se encuentran en la estructura (ya que no está ordenada).

Volcado de datos en gráfico



El eje vertical son la cantidad de géneros distintos que encontramos y el horizontal son la suma de los géneros en todos los libros.

El grafico es sobre el dataset 1. Para los demás dataset el grafico se hace ilegible pero mantiene el mismo formato.

CONCLUSIONES

Luego de realizados los análisis llegamos a la conclusión de que se podrían encontrar estructuras y/o funciones más eficientes para la implementación de la herramienta, en especial en la carga de los elementos.

En futuros trabajos se tratara de solucionar la eficiencia de la herramienta.