

**Trabajo Práctico Especial**  
**Programación 3**  
**TUDAI**  
**Primera Entrega**

**Facultad de Ciencias Exactas**  
**Universidad Nacional del Centro**

**Alumno: Sandoval, Fernando Daniel**

**Email: [fdsandoval@gmail.com](mailto:fdsandoval@gmail.com)**

**Fecha de entrega: 4/5/2018**

# Introducción

La cátedra nos solicita que implementamos una herramienta que permita simplificar la búsqueda de libros por géneros. Y además caracterizar el comportamiento de los usuarios mientras realizan dichas búsquedas. En una primera etapa se desea implementar la logica para obtener una colección de libros que contenga un género en particular. Para simplificar el proceso de búsqueda se requiere implementar un índice por género.

## Desarrollo del trabajo

Se analizaron previamente diversas posibles estructuras para la implementación, e inicialmente se optó por utilizar una lista de listas para volcar los datos de los archivos .csv en memoria y a partir de allí generar los índices de acceso por género. Luego se detectó que podía generarse el índice directamente desde la lectura de los archivos .csv, sin necesidad de una estructura auxiliar.

## Implementación

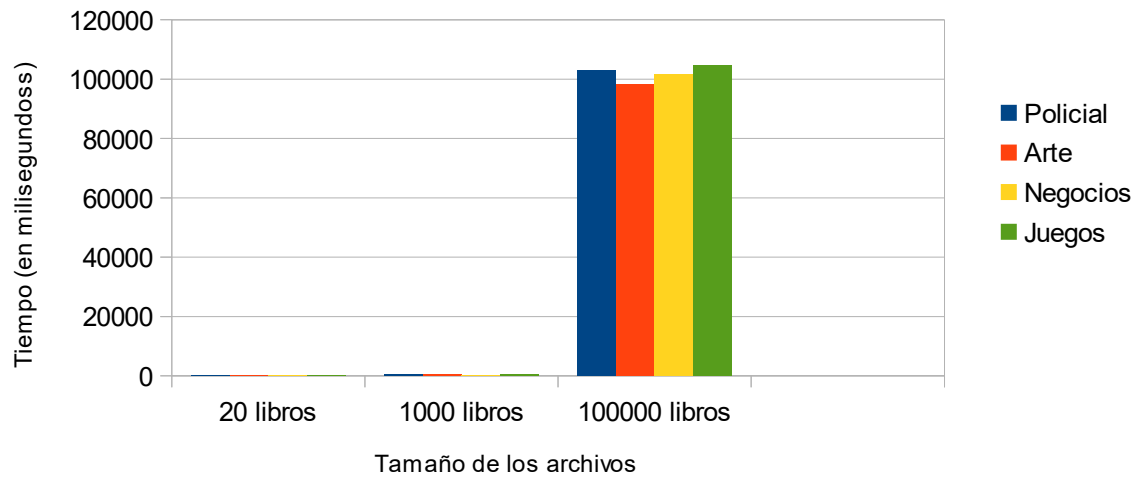
Se utilizó una lista cuyos nodos tienen cada uno el nombre del género y una lista de libros pertenecientes a ese género. La lista de generos se va creando a medida que se va recorriendo el archivo csv, como así también la lista de los libros que pertenecen a cada género. Una vez creada, fácilmente pueden obtenerse los datos correspondientes a los títulos de los libros que pertenecen a un determinado género, haciendo una búsqueda simple en la lista, pasándole como parámetro un String con el género buscado.

En el análisis previo a la implementación no se tuvo en cuenta la elección de una estructura óptima en cuanto a complejidad o rendimiento según el volumen de datos (como un árbol binario de búsqueda), sino que se eligieron en función de su facilidad para poder implementarse (como una lista simplemente vinculada). En este caso, el costo de la búsqueda en el mejor caso sería de  $O(1)$  si el género buscado se encuentra en el primer lugar de la lista, en el peor caso sería de  $O(n)$  si el género se encuentra en la última posición, y en promedio tendríamos un  $O(n/2)$ , tomando este caso tenemos la mitad de comparaciones con respecto a la cantidad de elementos. La velocidad de ejecución dependerá linealmente del tamaño de la lista.

## Mediciones

Se tuvo en cuenta para realizar las distintas mediciones el ingreso por teclado del género a ser buscado en el índice, ya que es lo único que puede hacer variar en mayor o menor medida el tiempo de ejecución. Los tiempos de lectura de los datos desde el archivo csv sólo varían dependiendo del tamaño del archivo csv. Tomamos diferentes muestras de tiempos y las comparamos a continuación:

Tiempo de carga de los archivos csv  
en milisegundos



	Policial	Arte	Negocios	Juegos
20 libros	31	38	91	28
1000 libros	560	480	305	528
100000 libros	103145	98350	101547	104637

En la muestra de datos vemos que la carga de 20 libros en promedio duplica la cantidad de accesos, mientras que para 1000 se tiene el caso promedio ( $n/2$ ), y finalmente para 100000 se obtiene un número similar al del tamaño de la lista. No se han considerado en esta tabla los valores correspondientes a la carga de 1000000 de libros dado que se obtuvieron valores muy altos, del orden de los 10000000 de milisegundos.

# Conclusiones

Como podemos ver, con ésta estructura utilizada (lista simple) se tiene un crecimiento lineal de accesos con respecto a los datos, lo cual se ve reflejado en la tabla. A mayor número de datos, mayor es el tiempo utilizado. Utilizando otras estructuras seguramente obtendremos mejores tiempos de acceso a los datos buscados, con la consecuente mejora en cuanto a la complejidad del algoritmo.