

## Taller 4

### Técnicas de ordenación

#### ***Preámbulo***

Los distintos métodos de ordenación se presentan convencionalmente para operar sobre arreglos, más sin embargo, en muchas situaciones prácticas los objetos a ordenar se encuentran en colecciones implementadas por ejemplo mediante listas (simple/doblemente enlazadas). El método de ordenación Mergesort suele ser utilizado para ordenar colecciones representadas mediante listas, pues se presta para una implementación más sencilla.

Por otro lado, una consideración importante para la implementación de cualquier método de ordenación es como implementar la comparación de objetos, en particular cuando son ADT compuestos y existen múltiples opciones respecto al criterio de ordenación.

El ejercicio propuesto busca explorar ambos problemas en un caso práctico.

#### ***Ejercicio a desarrollar***

Es común tener colecciones relativamente grandes de ADTs con múltiples atributos y para los cuales se desea poder hacer una visualización ordenada por uno de los atributos.

Un ejemplo es el *dataset* de canciones con su ranking, que se puede obtener aquí:

<https://think.cs.vt.edu/corgis/csv/music/music.html>

Descargar el *dataset* y realizar las siguientes operaciones con base en este:

1. Definir el ADT Cancion, el cual represente los atributos de una canción. Para facilitar considerar estos atributos (artist.name, release.name, song.hottness, title, year, location). El ADT debe implementar un constructor que devuelva una instancia a partir de una línea del archivo `.csv`.
2. Definir la clase ColeccionCanciones. Implementar el método leerCanciones que toma el nombre del archivo y construye una lista simplemente enlazada con todos los ADT Cancion.
3. Implementar un Comparator de ADTs Cancion como una clase interna de ColeccionCanciones que haga uso del atributo de la clase sortField, de forma que sea posible elegir con cual campo se realiza la comparación en el método compare. Dar la implementación del método less haciendo uso de este Comparator.

4. Implementar el método mergesort operando con lista simple.
5. Implementar un método top20 que acepte la cabeza de la lista y el atributo sortField. El método debe ordenar la lista con respecto al atributo seleccionado e imprimir en pantalla las primeras 20 canciones de acuerdo a este atributo.
6. Dar una implementación del main que invoque toda la secuencia: leerCanciones, top20. Top20 a su vez crea el comparador, organiza e imprime las top 20 en orden descendente.

## **Entregables**

- Código fuente (no incluir el .jar)
- Resultados obtenidos por top20 ordenando por *song.hottnesss*, por *title* y por *year*. Para cada uno de los casos reportar el tiempo requerido por la ordenación.

Trabajar en grupos de máximo 3 personas.

Fecha de entrega: Abril 24

## **Referencias**

baeldung. [Comparator and Comparable in Java](#), 2019.

Oracle. [Interface Comparator](#).

GeeksforGeeks. [Comparator Interface in Java with Examples](#).