Sesión 4 de Estructura de Datos: **Curso 2012/2013**

El objetivo de esta sesión es conocer las estructuras enlazadas, y sus implementaciones en Python. La fecha límite de entrega de esta sesión es el día 14-04-2013 para los grupos del jueves i el día 21-04-2013 para los grupos del lunes. Documentar el código.

Ejercicio

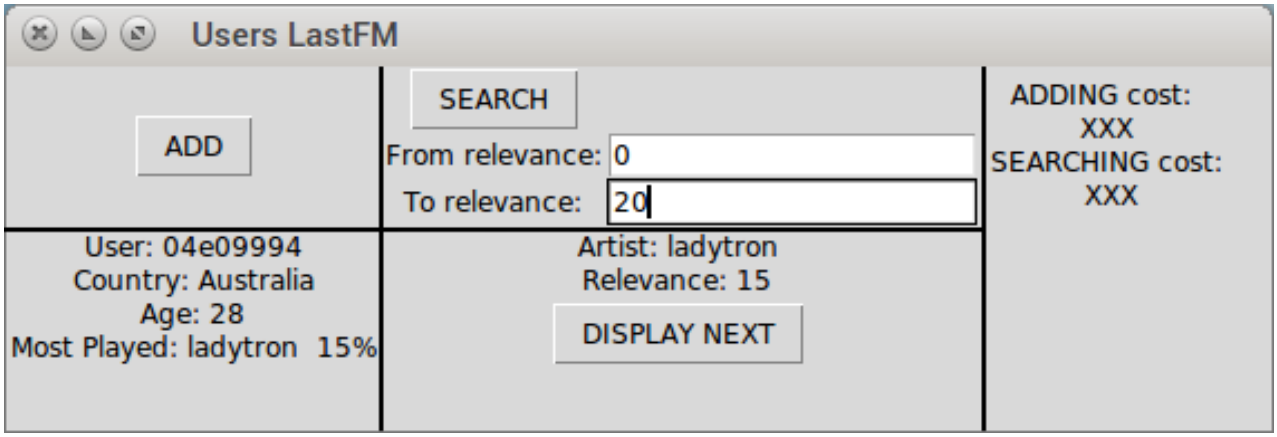
En esta sesión vamos a realizar una aplicación para ver la información de un conjunto de usuarios de LastFM a los que les gusta la música. Estos usuarios además de su información personal, nos han proporcionado la información de las canciones que más les gustan. Toda esta información está en un fichero de datos llamado ***LastFM_small.dat*** del que vamos a extraer información de los usuarios. En el campus virtual encontrareis un fichero llamado ***parserLastFM.py*** que contiene una función parser para leer dicho fichero de datos, el cual devuelve una lista no ordenada de objetos ***User***.

Cada usuario contiene los siguientes **atributos privados** que son:

- *uid*, como identificador del usuario
- *age*, edad del usuario
- *gender*, genero del usuario
- *country*, país de origen del usuario
- *songs_played*, lista de objectos ***Artist*** que son los artistas que le gustan al usuario
- *most_played*, el artista que más le gusta al usuario
- *relevance*, porcentaje de relevancia del artista más escuchado por el usuario

El objetivo principal es gestionar la colección de usuarios y para ello utilizaremos una lista enlazada. Esta gestión consiste en hacer operaciones de inserción y búsqueda. La eficiencia de estas operaciones se estimará calculando el tiempo de ejecución de estas operaciones.

Una posible interfície sería:



La interfície nos muestra "From relevance" y "To relevance" que sirven para definir el criterio de búsqueda de los usuarios. Cada vez que se aprieta el botón de "'NEXT", se nos muestra el siguiente en la lista con el criterio que se ha definido de "From"' y "To"'. Si se alcanza el final del "To", el "'NEXT" vuelve a la posición inicial del "From". En caso que no se haya especificado un criterio, el "From" es el inicio de la cola de prioridad y el "To" es el final

de la cola de prioridad. Además, la pantalla nos muestra el tiempo computacional consumido para inserir los usuarios (ADDING cost) y buscar (SEARCHING cost) usando la lista enlazada.

Para implementar la aplicación seguir las instrucciones:

Paso 1. Crear las clases necesarias para guardar la información de los usuarios

Paso 2. Implementar una cola como lista enlazada (podéis utilizar la clase Nodo y Queue)

Paso 3. Utilizando la función parser implementada en el fichero ***parserLastFM.py***, leer el fichero ***LastFM_small.dat***

Paso 4. Implementar la interfície de la aplicación en la clase **ColaInterface.py** con las siguientes funcionalidades:

a. ADD: añade 1000 usuarios del fichero en la cola (la del paso 2). Cada vez que se apriete el ADD se añaden los 1000 siguientes usuarios hasta que se acabe el fichero
b. SEARCH: dados dos valores "From relevance" y "To relevance", recupere todos los usuarios que tienen una relevancia dentro de este intervalo
c. DISPLAY NEXT: muestra secuencialmente el id del usuario, su país, edad, genero, artista y relevancia

Paso 5. Visualizar el tiempo de ejecución de la inserción (ADD) y la búsqueda (SEARCH) en la interfície (mirar la figura anterior)

Paso 6. Convertir la cola del paso 2 en una cola de prioridad (PriorityQueue) y crear una aplicación alternativa **PColaInterface.py**. La lista enlazada ha de quedar ordenada (sin usar un SORT) por orden descendiente de relevancia cada vez que se introduce un usuario.

Paso 7. Comparar el tiempo de ejecución promedio de insertar (ADD) y buscar (SEARCH) en las dos aplicaciones (como mínimo 10 veces). Para esto podéis utilizar el fichero ***LastFM_big.dat***

NOTAS:

- ***La entrega de la sesión ha de ser únicamente por el Campus Virtual dentro del límite de tiempo predefinido.***
- ***Entregad un fichero comprimido ("NombreApellidos_S4.zip") que contenga todo el código fuente (las dos aplicaciones).***
- ***Recordad que es obligatorio e*****structurar bien la información, documentarla y argumentarla*****.**