

TIENDA DE MÚSICA - PROYECTO FINAL

Las estructuras de datos son utilizadas para guardar información de manera organizada y para consultar o recuperar dicha información eficientemente, por lo tanto, el proyecto final del espacio académico estructura de datos consiste en desarrollar un proyecto donde se practique el uso de las estructuras vistas en clase durante todo el semestre. En consecuencia con lo anterior, la tienda de música Storify necesita una aplicación moderna donde sus usuarios puedan iniciar sesión, buscar canciones, guardarlas como favoritas y reproducirlas dentro de la misma aplicación.

La información requerida es:

- De una Canción se necesita un código único (generado aleatoriamente), nombre de la canción, nombre del álbum, carátula, año, duración, género (Rock, Pop, Punk, Reggaeton, Electrónica) y url de la canción en youtube.
- De un Autor/Artista se conoce un código, nombre (único), nacionalidad, un atributo que identifique si el Artista es un grupo o no, y una lista doblemente enlazada de Canciones.
- De un Usuario se conoce su username (único), contraseña, email, además el usuario tiene una lista propia de Canciones, dicha lista es una Lista circular.

La tienda guarda su catálogo de música agrupando las Canciones en sus respectivos autores en forma de lista, los Artistas se almacenan en forma de Árbol Binario, el orden del árbol está dado por el nombre de los Artistas. Además tenga en cuenta que los usuarios se guardan en un HashMap, donde la llave de cada usuario es su username.

La aplicación debe tener interfaz gráfica de usuario y además debe poder guardar toda la información en el disco duro, por lo tanto debe usar serialización de objetos.

Las **funcionalidades** de la tienda son:

Del lado del usuario:

1. **Iniciar sesión:** Los usuarios deben poder iniciar sesión usando su username y contraseña, una vez iniciada la sesión debe cargar todas las canciones que el usuario tenga guardadas.
2. **Buscar:** El usuario debe poder buscar música en la tienda, hay tres formas de buscar:
 - a. **Búsqueda Artistas:** Debido a que los artistas se guardan en un árbol binario, su búsqueda es muy eficiente, por lo tanto, dado el nombre de un Artista debe retornar su lista de canciones.
 - b. **Búsqueda O:** Dados los valores de dos o más atributos de una Canción, retorne una lista con las canciones con al menos un atributo que coincida.
 - c. **Búsqueda Y:** Dados los valores de dos o más atributos de una Canción, retorne una lista con las canciones con todos los atributos que coincidan.
 - d. **Para la búsqueda O y la búsqueda Y:** Usar Hilos de ejecución de tal manera que un hilo se encargue de buscar en el lado derecho del árbol, y otro hilo busque en el lado izquierdo del árbol.
3. **Guardar/Eliminar una canción en la lista del usuario:** El usuario puede guardar cualquier canción en su lista de canciones, así como eliminar de su lista las canciones que desee.
4. **Deshacer:** Deshacer una inserción o eliminación
5. **Rehacer:** Rehacer la función que se deshizo anteriormente.
6. **Ordenamiento:** El usuario debe poder ordenar su lista de canciones. Se debe poder ordenar usando cualquier atributo de la canción.
7. **Reproducir una canción:** Cuando se de click a una canción se debe poder reproducir en una ventana nueva, esta es la razón por la que cada canción tiene una url de youtube. Esta

funcionalidad necesita embeber una url que contiene un video de youtube en JAVA, se debe investigar cómo hacerlo. Como base pueden revisar las siguientes páginas:

<http://dproject.sourceforge.net/ns/> <https://stackoverflow.com/questions/20440484/embed-a-youtube-video-to-jframe> y otros.

Del lado del administrador:

1. **Iniciar sesión:** El administrador debe poder iniciar sesión (usar la misma interfaz de inicio de sesión del usuario), la aplicación solo tiene un administrador, cuyo username es *admin* y contraseña es *\$aDmiN*.
2. **Crear artistas - Agregar canción:** El administrador debe poder crear Artistas y agregarle canciones.
3. **Cargar información:** También se debe poder crear artistas cargando un archivo de texto plano con el siguiente formato:

```
#Artistas
CódigoArtista1;NombreArtista1;Nacionalidad1>true/false1
CódigoArtista2;NombreArtista2;Nacionalidad2>true/false2

#Canciones
NombreArtista1;NombreCanción;NombreAlbum;Año;Duración;Género;URLCanciónYoutube
NombreArtista1;NombreCanción;NombreAlbum;Año;Duración;Género;URLCanciónYoutube
NombreArtista1;NombreCanción;NombreAlbum;Año;Duración;Género;URLCanciónYoutube
NombreArtista2;NombreCanción;NombreAlbum;Año;Duración;Género;URLCanciónYoutube
```

4. **Consultar** el género con más canciones en la Tienda y el Artista más popular.

Requisitos adicionales:

- Todo el código debe estar debidamente documentado.
- Generar el JavaDoc.
- Pruebas unitarias al menos a 7 métodos.

Entrega:

El proyecto debe ser entregado en Google Classroom en un archivo .zip. La sustentación será en el horario de clase. Cada grupo será de máximo 3 personas.

Evaluación:

La entrega tendrá una calificación asignada entre 0 y 5, la sustentación tendrá una calificación individual entre 0 y 1. Para cada estudiante la calificación del proyecto final será la multiplicación entre la calificación de la entrega y la calificación de la sustentación.