

Astor Prieto DehghanPour (aprietde10)

Programació II

31 – 03 – 2014

Lliurament 2, ReproductorUB2

# Índex

- **Introducció**
- **Anàlisi**
  - Classes Implementades
  - Codi Reutilitzat
- **Desenvolupament**
  - Proves Realitzades
  - Exemples
- **Resultats**

## **Introducció**

L'objectiu d'aquesta practica és poder organitzar totes les cançons en una biblioteca musical i poder crear llistes de reproducció que contindran una selecció de cançons de la biblioteca.

Tot fent servir o reutilitzant parts de codi del Lliurament1 (FitxerAudio, MenuReproductor...), hem de implementar una classe LlistaFitxers, que ens servirà per a definir com serà una llista de fitxers de audio, i quina forma tindran les llistes de reproducció.

A partir de la biblioteca musical, on tindrem tots els arxius de audio, definirem les llistes de reproducció, que vindran definides per la classe LlistaReproducció (que hereta de la classe LlistaFitxers) i seran llistes que contindran una selecció de cançons de la biblioteca.

El nostre reproductor pot tenir nomes una sola biblioteca musical, mentre que pot tenir indefinides llistes de reproducció.

Seguint el model MVC (Model-Vista-Controlador), tindrem una classe DadesReproductor que contindrà tota la informació del reproductor i que al mateix temps, es comunicarà amb la classe CtrlReproductor, que serà la que es comuniqui amb la Vista, per tal de garantir una seguretat a les dades del reproductor.

## **Anàlisi**

En aquest lliurament, i tenint com a recolzament les classes fetes al lliurament primer, trobem que hem hagut de implementar classes noves basades en algunes antigues, pero que també hem pogut reutilitzar algunes de les ja existents.

## Classes Implementades

Com ja hem mencionat abans, per tal de seguir el model MVC, i per a garantir la seguretat de les dades del reproductor, implementarem la classe DadesReproductor, on trobarem totes les dades de la biblioteca musical així com també tota la informació de les llistes de reproducció. Aquesta classe contindrà tots els mètodes interns per a comunicar-se amb les classes més bàsiques (LlistaFitxers, LlistaReproduccio...), i alhora, serà la classe que utilitzi CtrlReproductor. La classe CtrlReproductor serà la classe del controlador, que farà ús dels mètodes creats a DadesReproductor per a modificar les dades i així mai donar un accés directe des de la Vista al Model. La classe CtrlReproductor constarà així de un objecte del tipus DadesReproductor on es farà crida de tots els mètodes necessaris per a implementar la funcionalitat del menú.

Finalment, la classe MenuReproductor contindrà tota la informació i esquelet del menú del reproductor, així com els seus sub-menús i totes les opcions. Mitjançant un objecte del tipus CtrlReproductor farem accés des del menú de tots els mètodes del reproductor i així únicament haurem de crear un objecte del tipus MenuReproductor a la classe main.

## Codi Reutilitzat

Ja que aquest és un projecte amb diferents parts, trobem que algunes parts del codi poden ser reutilitzades per a posterior ús. Trobem que podem aprofitar classes senceres com FitxerAudio (la classe bàsica per als arxius de música), o simplement modificant classes ja existents tenim quasi tot el esquelet del reproductor muntat. El

MenuReproductor o la TaulaFitxers en són bons exemples. A la classe TaulaFitxers, i partint de la implementació del ReproductorUB1-B, podem observar que gran part del codi i dels mètodes empleats, poden ser directament usats, o modificant-los lleugerament, a la nostra nova classe LlistaFitxers. De mateixa manera passa amb MenuReproductor, que únicament cal estendre el codi per a afegir diversos sub-menús a les opcions 1, 2 i 2.4, i tenim el esquelet del reproductor muntat.

Mètodes antics com el `demanaDadesTeclat()` o `guardarDades()` i `recuperarDades()` han sofert modificacions, però en la seva majoria han pogut ser aprofitats.

En quant a canvis, el principal canvi és la implementació de `CtrlReproductor` al programa, que impedeix que el usuari accedeixi directament a les dades del Model, creant així una barrera que protegeix `DadesReproductor`.

## **Desenvolupament**

### **Proves Realitzades**

Per tal de poder comprovar el correcte funcionament del reproductor, s'han realitzat un nombre de proves dins de cada apartat del menú. Dins de la primera opció, la gestió de la biblioteca, comprovem si afegeix, elimina i mostra la llista de cançons com es demana, així com si mostra correctament els missatges d'error si troba que una pista que volem afegir ja es troba a la biblioteca o si una pista que intentem eliminar no existeix. Ja que en aquest lliurament el punt 1.4 (play) no era corresponent a aquest lliurament, s'ha deixat sense implementar, mostrant únicament un missatge informatiu.

A la segona opció, la gestió de les llistes de reproducció, trobem la opció de crear una llista de reproducció nova, que per tal de crear-la ens demanen posar-li nom, també trobem la opció de mostrar una llista amb totes les llistes de reproducció creades fins al moment i una tercera opció per a eliminar llistes de reproducció del nostre programa.

A la opció 2.4 trobem un sub-submenú que ens porta a la gestió de una llista de reproducció donant-li un nom al reproductor. Aquest submenú conté les opcions de afegir, eliminar i mostrar els arxius de la "playlist" en qüestió. Podem comprovar que el menú de gestió de les llistes no ens deixa ni crear una playlist ja existent ni borrar una que no hi és, i també observem que mitjançant el mètode `eliminarFitxerBiblioteca()`, quan borrem un arxiu de la biblioteca musical, també és eliminat de totes les playlists on hagi estat contingut. Aquesta acció es fa mitjançant un `Iterator`, que itera sobre la llista de llistes de reproducció `IRepro` de la classe `DadesReproductor` i va eliminant llista per llista el element que ha sigut suprimit de la biblioteca.

Finalment, a la tercera i quarta opció del menu principal, trobem els mètodes per a guardar i recuperar dades, que donant el nom de un arxiu i la seva extensió, guarda un objecte del tipus DadesReproductor en un arxiu extern (si no existeix el crea el programa), que conté les dades de la biblioteca i la llista de llistes. De la mateixa forma, al recuperar el arxiu donant-li el nom de la biblioteca a recuperar desitjada, comprova si el arxiu existeix a la capreta del projecte i en carrega el objecte DadesReproductor que s'ha guardat previament.

## Exemples

En els següents exemples podem veure com es mostra la informació en les diferents parts del menu:

Troblem, per exemple, com es mostra la llista de cançons de la biblioteca musical.

```
-----
Entra una opcio >> 2
Opcio 2...
[1] | Titol: 1 | Autor: 1 | Disc: 1 | Pista: 1 | Dicografica: 1 | Duracio: 1.0 | Any: 1 | Fitxer: <1> |
[2] | Titol: 2 | Autor: 2 | Disc: 2 | Pista: 2 | Dicografica: 2 | Duracio: 2.0 | Any: 2 | Fitxer: <2> |
[3] | Titol: 4 | Autor: 4 | Disc: 4 | Pista: 4 | Dicografica: 4 | Duracio: 4.0 | Any: 4 | Fitxer: <4> |
[4] | Titol: 5 | Autor: 5 | Disc: 5 | Pista: 5 | Dicografica: 5 | Duracio: 5.0 | Any: 5 | Fitxer: <5> |
[5] | Titol: 3 | Autor: 3 | Disc: 3 | Pista: 3 | Dicografica: 3 | Duracio: 3.0 | Any: 3 | Fitxer: <3> |

-----
REPRODUCTOR UB2
-----
1.- Gestio Biblioteca:
2.- Gestio Llistes Reproduccio:
3.- Guardar Dades:
4.- Recuperar Dades:
5.- SORTIR
-----
Entra una opcio >>
```

També podem veure com es mostren la llista de playlists dins de la gestió de llistes de reproducció.

```
-----
Entra una opció >> 2
Opció 2...
--- LLISTES REPRODUCCIO ---
1 - Rock
2 - Jazz
3 - Electro
-----
REPRODUCTOR UB2
-----
1.- Gestio Biblioteca:
2.- Gestio Llistes Reproduccio:
3.- Guardar Dades:
4.- Recuperar Dades:
5.- SORTIR
-----
Entra una opció >>
```

Finalment, veiem com es mostra la informació de una playlist en concret (Rock) en el submenu 2.4. Observem també com a les playlists si que es pot repetir una pista, cosa que a la biblioteca no pot ocórrer.

```
-----
Entra una opció >> 4
Nom de la llista a Gestionar:
Rock
-----
2.4. GESTIONAR LLISTA:
-----
1.- Afegir Fitxer Audio:
2.- Mostrar Llista:
3.- Eliminar Fitxer:
4.- Menu Anterior.
-----
Entra una opció >> 2
Opció 2...
[1] | Titol: 4 | Autor: 4 | Disc: 4 | Pista: 4 | Dicografica: 4 | Duracio: 4.0 | Any: 4 | Fitxer: <4> |
[2] | Titol: 2 | Autor: 2 | Disc: 2 | Pista: 2 | Dicografica: 2 | Duracio: 2.0 | Any: 2 | Fitxer: <2> |
[3] | Titol: 4 | Autor: 4 | Disc: 4 | Pista: 4 | Dicografica: 4 | Duracio: 4.0 | Any: 4 | Fitxer: <4> |
```

## **Resultats**

Després de comprovar la majoria de errors que han pogut sorgir a l'hora de implementar el codi, mitjançant “throws” i capturant-los amb “catch” en la seva majoria al controlador, repassem els objectius i fites assolides:

- S'ha realitzat una biblioteca musical en Java.
- Aquesta biblioteca musical dona suport també a llistes de reproducció.
- S'ha dominat el ús tant com de ArrayLists, herències i sobrecàrregues, streams de entrada i sortida i Iterators per recorre elements.
- S'ha après a usar el model MVC en la seva totalitat (incloent Controlador), per així oferir una major robustesa als nostres programes.