

Music-To-Play

Gestió d'un catàleg musical

Metodologia i Tecnologia de la Programació II

Maig de 2025

Índex

1	Introducció	2
2	Comandes de l'aplicació	2
3	Disseny de l'aplicació	3
3.1	Canco	3
3.2	Gestor	4
4	Suport a la programació	4
4.1	Formats	5
4.2	Cerca en un string	5
5	Format del fitxer d'un catàleg	5
5.1	Tipus de dades	6
5.2	Estructura del fitxer	6
6	Criteris d'avaluació i procediment d'entrega	7
7	Exemples proporcionats	7

1 Introducció

Us proposem desenvolupar un programa en C++ per a gestionar catàlegs musicals i el control de la reproducció de les cançons d'un catàleg. El programa gestor s'anomena **Music-To-Play (MTP)**. Quan s'inicia MTP, el primer que es demana és quin catàleg volem gestionar (el catàleg es llegirà d'un fitxer, veure Secció 5). A continuació MTP oferirà un conjunt d'opcions per a poder gestionar les cançons del catàleg escollit (veure Secció 2). De les cançons tenim diverses informacions, com ara el títol, l'autor, i l'any de llançament, entre d'altres. Les funcionalitats principals del gestor són:

- Afegir noves cançons al catàleg.
- Mantenir una cua de reproducció.
- Mantenir un registre (pila) de cançons escoltades recentment.
- Seleccionar una cançó. **MTP tindrà en tot moment una *cançó seleccionada***, sobre la qual podrem realitzar accions com ara reproduir-la o afegir-la a la cua de reproducció.
- Reproduir una cançó (se simularà escrivint un text per pantalla).
- Mostrar les cançons del catàleg filtrades per diferents criteris, així com de la cua de reproducció i de la pila de les cançons ja escoltades.

2 Comandes de l'aplicació

Tot seguit enumerem totes les comandes que poden ser utilitzades pel gestor MTP. L'entrada/sortida de qualsevol comanda cal consultar-la en els exemples d'execució annexos.

- **afegir**: afegeix una cançó al catàleg. Inicialment pregunta per les dades **títol** i **artista**, que juntes seran un identificador únic. Si la cançó no existeix, pregunta per la resta de camps essencials. Altrament, s'informa a l'usuari que aquesta cançó ja existeix en el catàleg. En qualsevol cas, aquesta cançó passa a ser la *cançó seleccionada*.
- **encuar**: la cançó seleccionada s'afegeix a la cua de reproducció. Aquesta comanda no canvia quina és la cançó seleccionada.
- **seleccio**: escriu les dades de la cançó seleccionada en format complet.
- **buidar**: buida la pila de cançons escoltades recentment.
- **repseleccio**: simula que sona la cançó seleccionada. Apareix el missatge "**Reproduint...**" seguit de les dades de la cançó en format curt (una línia). La cançó passa a la pila d'escoltades, i el comptador d'escoltades de la mateixa cançó també s'incrementa. Aquesta comanda no canvia quina és la cançó seleccionada.
- **repcua**: funcionament similar al de la comanda **repseleccio**, però respecte a la primera cançó de la cua de reproducció. La cançó reproduïda s'afegeix a la pila d'escoltades, desapareix de la cua, i s'incrementa el seu comptador d'escoltades.
- **sortir**: el programa finalitza, i s'escriu el missatge d'acomiadament "**Fins aviat!**".

- **seleccionar** <subcomanda>: les comandes **seleccionar** tenen com objectiu seleccionar una cançó dins del catàleg (cançó seleccionada). Tenim diverses alternatives de selecció dins del conjunt global de comandes, totes elles anàlogues. Consisteixen en escriure un llistat enumerat on apareix en format curt cada cançó segons el filtre escaient. L'ordenació és sempre per **títol** i **artista**. Si el filtre no es compleix per cap cançó, apareix el missatge “**Cap coincidència.**”. Si el filtre només el compleix una cançó, apareix el missatge “**Seleccio automatica (un sol resultat).**” i aquesta passa a ser la cançó seleccionada directament. Si hi ha més d'una cançó, apareix el missatge: “**Seleccio [1..n]?** ” on **n** és el valor màxim de l'enumeració. A continuació s'entrarà un nombre de 1 a **n** per teclat, i la cançó que hem triat passarà a ser la cançó seleccionada.
 - **totes**: la subcomanda **totes** indica que no hi ha filtre. Volem seleccionar d'entre totes les cançons.
 - **artista** <paraula_clau>: es filtren les cançons que contenen l'string <paraula_clau> dins de l'atribut **artista**.
 - **anim** <valor>: es filtren les cançons que tenen com estat d'ànim el valor entrat (ha de coincidir exactament).
 - **títol** <paraula_clau>: es filtren les cançons que contenen l'string <paraula_clau> dins de l'atribut **títol**.
- **escriure** <subcomanda>: les comandes **escriure** mostren un llistat enumerat de cançons en format curt, filtrat segons diferents criteris. Per qualsevol subcomanda de **escriure**, si el llistat resultant és buit, apareix el missatge “**Cap coincidència.**”.
 - **catàleg**: es llisten totes les cançons del catàleg ordenades per **títol** i **artista**.
 - **gènere**: es llisten totes les cançons del catàleg ordenades per **gènere** i desempat per **títol** i **artista** si escau.
 - **pendents**: es llisten les cançons de la cua de cançons pendents.
 - **recents**: es llisten les cançons de la pila de cançons recents.
 - **top10**: es llisten les 10 cançons amb més reproduccions, en ordre descendent de nombre de reproduccions (el desempat segueix essent ascendent per **títol** i **artista** si escau).

3 Disseny de l'aplicació

D'entre les diferents classes que caldrà implementar, n'hi ha dues de principals: la classe **Canco** (cançó), i la classe **Gestor**.

3.1 Canco

Aquesta classe encapsula totes les dades referents a una cançó, que són les següents:

- **UID**: identificador numèric únic per a cada cançó (enter).
- **títol**: cadena de caràcters (amb espais).
- **artista**: cadena de caràcters (amb espais).
- **any**: nombre natural (any en què apareix la cançó).

- **durada:** nombre natural (durada de la cançó en segons).
- **gènere:** cadena de caràcters sense espais en blanc. Entre d'altres, aquest camp pot prendre per valor: *blues, pop, jazz, rock, soul, funk, latin, hip-hop, folk, easy*. Podeu ampliar les opcions i emprar etiquetes pròpies sempre que la restricció d'un sol mot es respecti.
- **estat d'ànim:** cadena de caràcters sense espais en blanc. És una etiqueta que fa referència a l'estat d'ànim/sensació musical que aporta la cançó. Pot pendre els següents valors: *trist, calmat, alegre, energetic*.
- **nombre de reproduccions:** nombre natural (nombre de vegades que la cançó ha estat escoltada).

3.2 Gestor

Aquesta classe conté totes les dades d'un catàleg, i implementa tota la lògica del gestor musical MTP. L'estat d'un gestor musical es defineix per les dades següents:

- **nom:** cadena de caràcters (nom del catàleg).
- **cançons:** **taula dinàmica ampliable** de Canco (cançons). És el repositori on s'emmagatzemen les cançons.
- **recents:** **pila** d'identificadors de cançons (UIDs). Permet saber quines cançons han estat escoltades recentment.
- **pendents:** **cua** d'identificadors de cançons (UIDs). Conté les cançons pendents d'escoltar.
- **selecció:** enter, és l'UID de la *cançó seleccionada* (recordeu que sempre n'hi ha una).

Fixeu-vos que només la taula dinàmica conté realment els objectes que representen les cançons, la resta de camps emmagatzemen el seu UID. Haurem de garantir que l'UID d'una cançó coincideixi amb la posició que ocupa a la taula dinàmica ampliable. D'aquesta manera, podrem accedir als objectes Canco amb un accés directe (operador []). Per tant, cal tenir en compte que quan afegim una nova cançó se li ha d'assignar el següent UID disponible, que és el primer índex lliure de la taula.

Per generar qualsevol subconjunt ordenat per qualsevol camp (per exemple, **gènere** o **estat d'ànim**), caldrà copiar la taula en una de secundària i fer l'ordenació que s'escaigui (amb filtratge previ si fos necessari). Recordeu que amb un sol mètode d'ordenació és suficient si s'incorpora un paràmetre que especifiqui el criteri de comparació. **Atenció: us demanem que implementeu totes les ordenacions fent servir l'algorisme *quicksort*.**

4 Suport a la programació

Recordeu que és imprescindible, abans de començar a programar, dedicar un temps a fer un bon disseny amb les estructures de dades adients, a pensar bé els mètodes involucrats i, finalment, a fer el codi. La implementació s'ha de fer per etapes per a poder anar testejant el funcionament correcte dels mètodes i de les accions i funcions que es van afegint.

4.1 Formats

Podem assumir que els valors numèrics necessaris per enumerar llistats o UUIDs pertanyen a l'interval $[0, 99]$ i, per tant, la instrucció: `setw(2)` és útil.

Alguns llistats contenen separadors de grups per a facilitar la lectura (especialment per a depurar). Per a més detalls, cal veure els exemples, però cal saber que la longitud de tots és de 70 caràcters. Es pot utilitzar un mètode de la classe `string` per a replicar un caràcter.

Exemple: `const string LINE = string(70, '-');`

4.2 Cerca en un string

La classe `string` ens ofereix moltes eines útils per a treballar amb cadenes de caràcters. Entre elles, trobem la funció `find`, que permet comprovar si una cadena es troba dins d'una altra. Aquesta funció retorna la posició on comença la subcadena dins de la cadena principal (tenint en compte que la primera posició és 0). Si no es troba cap coincidència exacta amb tots els caràcters, la funció retorna la constant `string::npos`.

Exemple:

```
string exemple = "Escoltant MTP";
string cerquem = "MTP";
exemple.find(cerquem); // retorna 10
exemple.find("MIT"); // retorna string::npos
```

Per tant, amb el següent patró `if (exemple.find("MIT") != string::npos)` podem saber si una cadena de caràcters apareix en una altra.

5 Format del fitxer d'un catàleg

Els catàlegs s'emmagatzemen en fitxers de text amb extensió `.mtp`. Aquests fitxers no només contenen les dades de les cançons del catàleg, sinó també un estat del gestor musical, és a dir, la cançó seleccionada, el nombre de vegades que s'ha reproduït cada cançó, les cançons recents i les cançons pendents. Tot fitxer compleix estrictament les següents característiques, que podem assumir quan implementem el programa:

1. El format de fitxer és correcte.
2. Conté com a mínim 10 cançons.
3. Totes les dades que conté són correctes.
4. Les cançons apareixen al fitxer ordenades per UUID, sense cap altre criteri d'ordenació, i no hi ha forats, és a dir, el fitxer conté tots els UUIDs entre 0 i $n - 1$, on n és el nombre de cançons del fitxer. Ara bé, en el fitxer, les cançons estan enumerades de 1 a n , és a dir, el seu UUID ha estat incrementat en 1, però per a la gestió de la taula l'hem de disminuir en 1 (perquè la taula de cançons s'indexa a partir de 0).

5.1 Tipus de dades

Les dades d'un fitxer corresponen a aquests quatre tipus:

- **Paraula** = cadena sense espais en blanc (ex: "rock", "trist")
- **PARAULES** = cadena amb alta probabilitat de tenir espais en blanc (ex: "Madonna", "The Rolling Stones", "Another Break In The Wall")
- **Nombre** = valor natural (ex: 2001).

5.2 Estructura del fitxer

L'estructura del fitxer és la següent:

Primera línia:

- **MTP[CATALEG]** (Paraula).
- **UID** de la cançó actual del catàleg +1 (Nombre).

Segona línia:

- **MTP[CANCONS]** (Paraula).
- **Nombre** de cançons (Nombre).

Per a cada cançó:

- **Primera línia:** **UID** +1 (Nombre)
- **Segona línia:** **Títol** (PARAULES).
- **Tercera línia:** **Artista** (PARAULES)
- **Quarta línia:**
 - **Any** (Nombre).
 - **Durada** (Nombre = segons).
 - **Gènere** (Paraula $\in \{\text{jazz, blues, pop, rock, soul, funk, country, latin, reggae, hip-hop, folk, easy, altre}\}$).
 - **Humor** (Paraula $\in \{\text{trist, calmat, alegre, energetic}\}$).
 - **Reproduccions** (Nombre = nombre de vegades que la cançó ha estat reproduïda).

Següent línia:

- **MTP[RECENTS]** (Paraula).
- **Nombre** de cançons recentment escoltades (Nombre).

Següent línia:

- **UIDs** dintre de la pila de recents (Nombres). Direcció de lectura: **el primer número és el cim de la pila.**

Següent línia:

- MTP [PENDENTS] (Paraula).
- Nombre de cançons en la cua de pendants (Nombre).

Següent línia:

- UUIDs dintre de la cua de pendants (Nombres). Direcció de lectura: el primer número és el primer element de la cua.

Següents línies:

- Si hi ha més text que segueix a les línies anteriors no ha de ser llegit per MTP.

6 Criteris d'avaluació i procediment d'entrega

Per a l'entrega de la pràctica, cal que es compleixin els punts que es detallen a continuació:

1. Cal que el codi estigui ben documentat, amb la descripció de cada classe i l'especificació dels mètodes, accions i funcions.
2. Es valorarà el bon disseny i la claredat del codi: noms de variables i mètodes ben triats, codi ben indentat, ús de constants, etc.
3. Els estudiants de GEINF i GDDV hauran d'implementar les classes que encapsulin, respectivament, la taula dinàmica, la cua i la pila que es demanen en el programa. Per altra banda, els estudiants de GEB hauran d'utilitzar les STL corresponents: vector, queue i stack.
4. Cal tenir el programa acabat i entregat a l'ACME com a molt tard el 05-06-2025.
5. Cal que el codi que s'entregui compili correctament a l'ACME. Si no es satisfà aquesta condició la pràctica no es corregirà i la nota serà un 0. El nombre d'errors de l'ACME no penalitzen. Tingueu però en compte que l'ACME no ha de ser utilitzat com una plataforma de testeig.
6. També cal entregar a través de la tasca "Pràctica final" del Moodle (Bloc 3) un document en PDF (no més d'una pàgina i el més curt millor) on es descrigui l'estat de la pràctica que s'entrega, indicant si està tot fet i funciona correctament, o si algunes comandes no acaben de funcionar, o si falta fer alguna comanda, etc.
7. Passat el termini d'entrega, hi haurà un període d'entregues presencials on cada alumne haurà de presentar la seva pràctica individualment.

7 Exemples proporcionats

Juntament amb aquest enunciat, es proporcionen els fitxers següents:

cataleg<i>.mtp Fitxer com el que s'ha d'escrit a la Secció 5, que conté un catàleg i un estat del gestor per ser carregat quan executem el programa.

exemple<i>_in.txt Fitxers que conté només l'entrada de l'exemple d'execució *i*, tal com l'entra l'ACME.

exemple<i>_out.txt Fitxer que conté només la sortida de l'exemple d'execució *i*, tal com l'espera l'ACME.

exemple<i>_in_out.txt Fitxer que conté l'entrada i la sortida de l'exemple d'execució *i*, tal com ho veureu al vostre ordinador si executeu el programa i aneu introduint les entrades intercalades amb la sortida.