

# **Disseny i programació orientats a objectes**

**Pràctica del segon semestre**

***Espotifai***

**Departament d'Enginyeria  
La Salle – Universitat Ramon Llull  
24 febrer de 2023**

# Índex

<b>1</b>	<b>Descripció general .....</b>	<b>3</b>
<b>2</b>	<b>Especificació de requeriments .....</b>	<b>4</b>
2.1	Configuració .....	4
2.2	Registrar-se i iniciar sessió .....	4
2.3	Tancar sessió i eliminar el compte .....	5
2.4	Gestió de música .....	5
2.4.1	Veure música disponible .....	5
2.4.2	Afegir música .....	6
2.4.3	Eliminar música.....	6
2.4.4	Estadístiques musicals .....	6
2.5	Reproductor de música .....	7
2.6	Gestió de llistes de reproducció .....	7
<b>3</b>	<b>Consideracions .....</b>	<b>8</b>

# 1 Descripció general

Després de nombrosos estudis de la màxima serietat científica, el departament de Desenvolupament i Recerca pels Projectes de DPOO ha descobert, de forma inesperada, que a alguns estudiants els agrada la música. Amb ànims renovats, s'han posat a treballar en un programa que permeti emmagatzemar música en local i organitzar-la en llistes de reproducció. Això sí, sempre respectant els drets d'autor de tota la música que el sistema faci servir.

L'arquitectura del sistema *Espotifai* seguirà el model d'una arquitectura per capes ([Layered Architecture](#)) tal com es pot veure en la representació de la Figure 1. En aquest diagrama es pot veure com el sistema estarà implementat segons una arquitectura de tres capes, tot incorporant el patró de disseny *Model-View-Controller* (MVC). Les vistes i els controladors d'aquestes vistes residiran dins la capa de presentació, mentre que la capa de negoci farà de model. L'objectiu d'aquest patró és mantenir les vistes totalment desacoblades del model, de forma que el model pugui treballar amb total independència de la implementació emprada per representar les vistes.

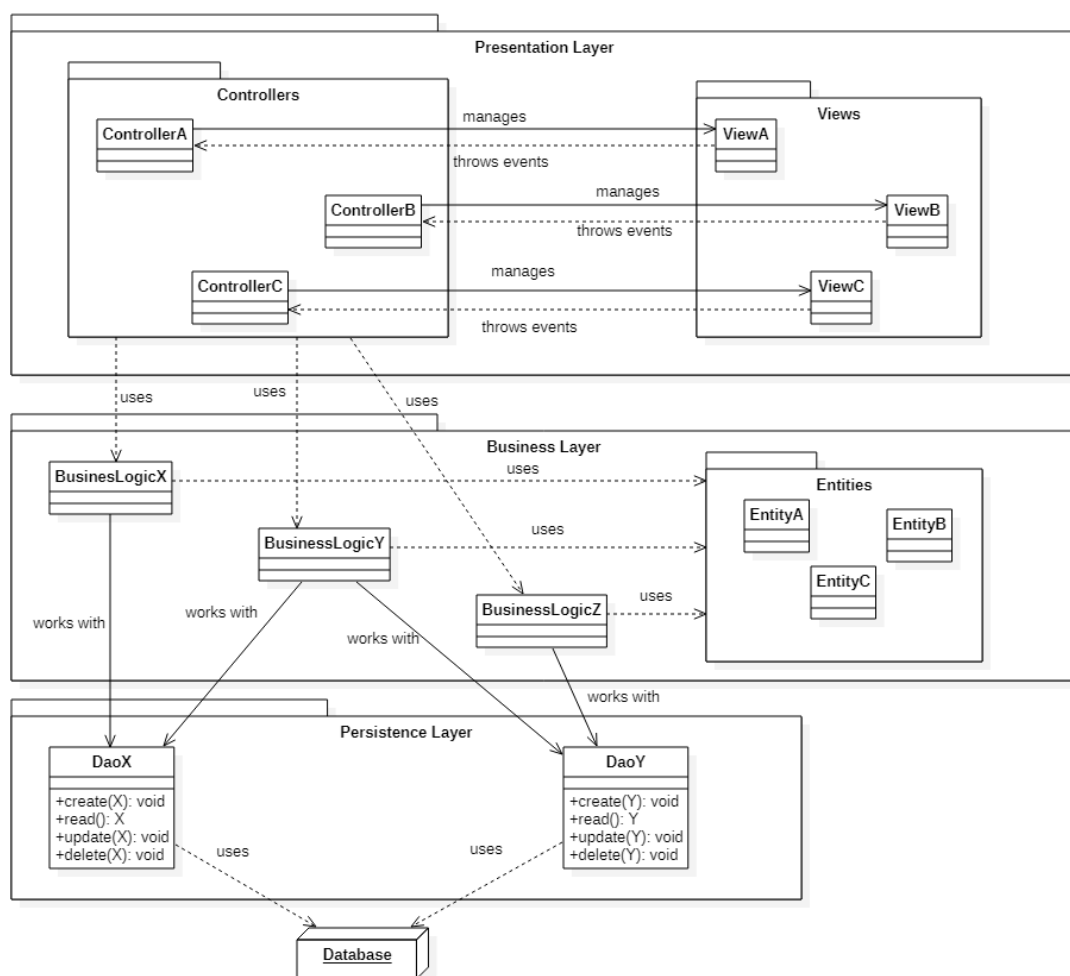


Figure 1: Representació d'una arquitectura per capes

Els apartats següents descriuen els detalls i el comportament que s'espera de les funcionalitats concretes que ha d'implementar cadascuna de les parts identificades en aquest escenari.

## 2 Especificació de requeriments

Aquest apartat recull el funcionament i el comportament que s'espera de l'aplicació *Espotifai* per tal de satisfer les especificacions requerides.

### 2.1 Configuració

En el moment d'engegar el programa aquest haurà de llegir un fitxer de configuració anomenat *config.json* ubicat en el directori arrel del projecte on hi haurà la següent informació:

- Port de connexió amb la Base de dades
- Adreça IP del servidor de la base de dades (normalment "localhost")
- Nom de la Base de dades
- Usuari d'accés a la Base de dades
- Contrasenya d'accés a la Base de dades

### 2.2 Registrar-se i iniciar sessió

Quan un usuari executi el programa, aquest es podrà registrar o bé iniciar sessió. Opcionalment també es poden implementar eines per recuperar la contrasenya, si l'usuari l'ha oblidat o iniciar sessió a través de comptes de xarxes socials.

Per tal de registrar-se al sistema, l'usuari haurà d'introduir la següent informació en un formulari (el format del qual és totalment lliure):

- **Nom d'usuari** (ha de ser únic en el sistema)
- **Correu** (el format del correu ha de ser correcte i ser únic en el sistema)
- **Contrasenya** (la contrasenya pot contenir caràcters alfanumèrics incloent minúscules i majúscules així com tot tipus de caràcters especials. Cal que la contrasenya tingui com a mínim una longitud de 8 caràcters així com contingui com a mínim majúscules, minúscules i valors numèrics. No obstant, es recomana la lectura de [la política de contrasenyes del MIT](#) com exemple de quines restriccions s'apliquen en entorns reals.)
- **Confirmació de contrasenya** (el contingut d'aquest camp ha de coincidir amb el contingut del camp "contrasenya")

Quan l'usuari cliqui a trametre el formulari, caldrà comprovar, que les dades del formulari compleixin els requisits explicats anteriorment i, en cas que no sigui així, es mostrarà un missatge a la interfície gràfica del programa explicant quin o quins han estat els problemes. En cas que tot sigui correcte es procedirà a registrar l'usuari en el sistema. En cas de registre satisfactori, l'usuari s'autenticarà automàticament al sistema i se li mostrarà la pantalla principal.

Per tal d'iniciar sessió al sistema l'usuari haurà d'introduir el seu nom d'usuari o el correu amb el qual es va registrar en el seu moment i la contrasenya associada. És important remarcar que s'ha de poder accedir tant amb l'usuari com amb el correu.

Un cop es trameti aquest formulari, es validarà i garantirà accés si les credencials són correctes i, en cas contrari, mostrarà un error genèric (per exemple, "Les credencials introduïdes són incorrectes").

## 2.3 Tancar sessió i eliminar el compte

Des de qualsevol pantalla de l'aplicació, un cop s'ha iniciat la sessió, els usuaris han de poder tancar la sessió. Un cop es tanca la sessió, l'aplicació retornarà a l'usuari a la pantalla per registrar-se o iniciar sessió. Aquesta opció ha d'estar sempre disponible per a l'usuari.

Segons el [RGPD](#) tot usuari té dret a ser oblidat, per aquest motiu, aplicació també ha de disposar d'una opció que permeti a un usuari eliminar el seu compte i esborrar d'aquesta forma tota la seva informació. Això implica esborrar tota informació d'aquest usuari de qualsevol element del sistema (base de dades, fitxers, memòria RAM del programa, ...).

Opcionalment, es poden agrupar aquestes dues funcionalitats en un panell de gestió del compte de l'usuari que sempre ha de ser accessible des de qualsevol punt del programa.

## 2.4 Gestió de música

A partir del moment que un usuari s'ha autenticat en el sistema, el programa ha de mostrar diversos components per a poder gestionar la música que s'emmagatzemarà en el sistema.

És important tenir en compte que tota la música serà pública, és a dir, qualsevol cançó que publiqui un usuari haurà de ser visible per a qualsevol usuari registrat del sistema.

### 2.4.1 Veure música disponible

El programa ha de permetre al seu usuari veure un llistat de totes les cançons que actualment hi ha al sistema. Aquest llistat s'haurà de mostrar actualitzat en temps real, de forma que serveixi per veure els canvis que es produeixen durant l'execució del programa.

Totes les cançons s'hauran de mostrar mitjançant el component gràfic JTable de la llibreria AWT/Swing. Per cada cançó caldrà mostrar com a mínim, el seu títol, gènere, àlbum, autor i el nom de l'usuari propietari.

El programa també haurà d'incloure un cercador que permeti filtrar la informació que es mostra sobre les cançons a partir d'un text.

Opcionalment, també es pot implementar controls per poder ordenar la informació mostrada segons diferents criteris (alfabètic ascendent, alfabètic descendent, etc.).

Quan l'usuari faci clic en una fila de la JTable, caldrà mostrar una nova vista amb la informació completa de la cançó triada. En aquest cas, caldrà mostrar tota la informació que ja es mostrava en la vista anterior, i addicionalment, caldrà mostrar la durada total de la cançó en el format <minuts:segons>, la lletra de la cançó, en cas d'estar disponible, un botó per poder reproduir la cançó, i l'opció d'afegir la cançó en una llista de reproducció ja existent.

Per obtenir la lletra d'una cançó, podeu fer servir la següent [API<sup>1</sup> REST<sup>2</sup>](#) :

<https://balandrau.salle.url.edu/dpoo/lyrics/{artist}/{song}>

Aquesta API retorna una representació JSON d'un objecte amb un sol camp. Aquest camp sempre serà de tipus String, però es dirà "lyrics" si conté la lletra, o "error" si aquesta no s'ha pogut trobar.

---

<sup>1</sup> Application Programming Interface

<sup>2</sup> REpresentational State Transfer

### 2.4.2 Afegir música

Per tal d'afegir una cançó es demanarà a l'usuari el títol, el gènere, l'àlbum, l'autor i un fitxer amb la cançó a afegir. Si les dades són correctes, llavors el programa ha de procedir a emmagatzemar aquestes dades. Quan la cançó ha estat afegida, aquesta és immediatament disponible per a tots els usuaris mostrant-ne com a propietari a l'usuari que l'ha afegit al sistema.

### 2.4.3 Eliminar música

El programa ha disposar de controls per poder eliminar les cançons que estan emmagatzemades al sistema. Una cançó només pot ser esborrada pel seu propietari. Quan l'usuari decideixi esborrar una cançó, l'acció haurà de ser confirmada d'alguna forma pel mateix usuari. La idea és que no sigui fàcil poder esborrar una cançó per error.

Quan una cançó s'esborra del sistema, també s'haurà d'esborrar de qualsevol llista de reproducció de la que formés part. Tingueu en compte que no es podrà eliminar una cançó que està sent reproduïda actualment.

### 2.4.4 Estadístiques musicals

El gràfic d'estadístiques musicals consistirà en un gràfic de barres on l'eix de les y mostrarà els diferents gèneres musicals que existeixen en el programa, i l'eix de les x mostrarà el nombre de cançons que formen part de cada un dels gèneres determinat en l'eix de les y. Tota la informació s'haurà de mostrar actualitzada en temps real.

En la Figure 2 es pot veure un exemple d'un gràfic de barres per representar les estadístiques musicals. En l'eix horitzontal, podem veure el nombre total de cançons que hi ha actualment en cada un dels gèneres musicals que es mostren en l'eix vertical. En aquest exemple els eixos estan escalats fins a 60, ja que el gènere musical amb més cançons és el Heavy Metal amb un total de 53 cançons. En l'eix vertical s'indiquen els noms de cada un dels gèneres musicals existents en el sistema. En cas que es consideri necessari per tal que la informació sigui llegible, es pot limitar el gràfic a un màxim dels 10 gèneres més importants del sistema.

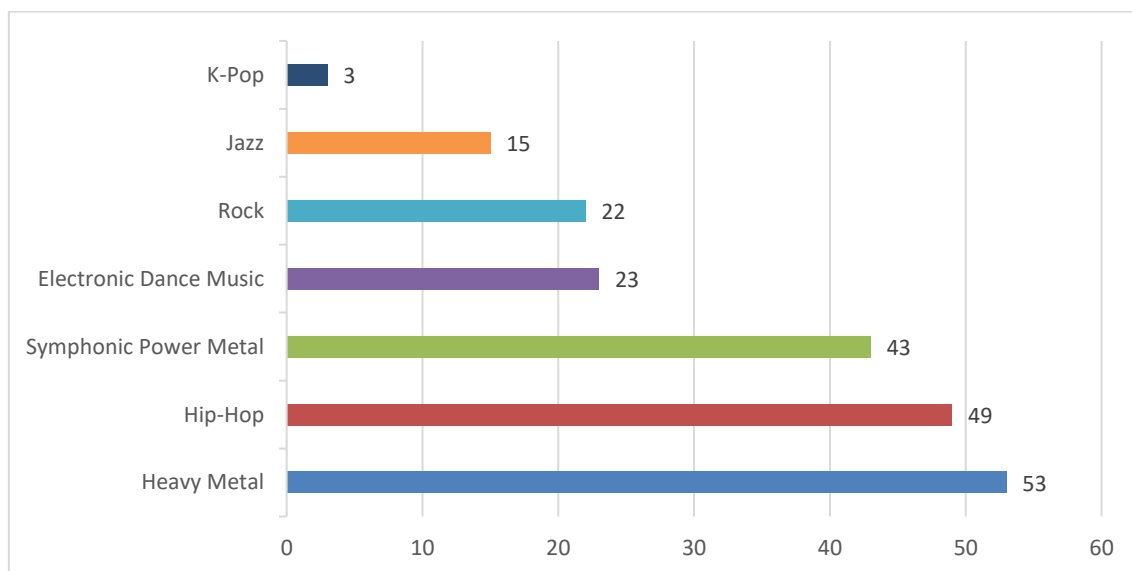


Figure 2: Exemple de gràfic d'estadístiques musicals

Els gràfic hauran de ser generat des del programa, utilitzant les eines de dibuix que ofereix la llibreria AWT/SWING de Java i escalat de forma conseqüent en funció dels valors mínim i màxim dels eixos. No es poden utilitzar llibreries externes per dibuixar el gràfic.

## 2.5 Reproductor de música

Des del moment en que l'usuari s'autentiqui en el sistema, el programa ha de mostrar en tot moment un conjunt de controls que permetin escoltar la cançó seleccionada. Aquests controls de reproducció de música han de permetre:

1. **Reproduir/Pausar/Parar.** Permet iniciar, pausar o aturar la reproducció d'una cançó.
2. **Anterior/Següent.** Permet avançar o retrocedir en les cançons d'una llista. No ha d'estar actiu quan es reprodueix una cançó des del llistat general de cançons descrit en l'apartat 2.4.1.
3. **Repetició individual.** Permet activar un mode en què la cançó que s'està reproduint torni a començar automàticament un cop acabi.
4. **Repetició global.** Arribat al final d'una llista de reproducció, aquesta torna a començar per la primera cançó d'aquesta llista. Només ha d'estar actiu quan es reprodueix una llista.

També caldrà mostrar visualment, el nom de la cançó que s'està reproduint, el temps transcorregut des de l'inici de la cançó, el temps total de la cançó i una barra de progrés que indiqui visualment quin percentatge de la cançó s'ha reproduït ja.

## 2.6 Gestió de llistes de reproducció

Des d'aquesta secció l'usuari ha de poder crear llistes de música i modificar-les afegint i eliminant cançons, així com visualitzar les llistes ja creades i les cançons contingudes en cada una d'elles.

Tot usuari haurà de disposar de dos espais separats, on pugui distingir entre les llistes de reproducció que ha creat ell i les llistes de reproducció que han creat la resta d'usuaris del sistema.

L'usuari pot, en qualsevol moment, visualitzar la informació d'una llista de reproducció. En aquest cas, haurà de poder veure el títol de la llista, així com un llistat de totes les cançons que s'han afegit en aquesta. Addicionalment, si la llista de reproducció l'ha creat el mateix usuari, aquest haurà de poder afegir-hi cançons del llistat de música disponible al sistema, reordenar les cançons ja existents a la llista, així com eliminar cançons de la llista. Recordeu que si un usuari elimina una cançó utilitzant les eines explicades a l'apartat, 2.4.3, aquesta cançó també s'haurà d'eliminar de la llista de reproducció.

La creació d'una llista ha de demanar a l'usuari el nom de la llista. A partir d'aquest moment, l'usuari que ha creat la llista ha poder veure-la en el llistat de llistes pròpies.

Totes les llistes de reproducció són públiques, és a dir, quan un usuari creï una llista nova, aquesta serà visible també per a la resta d'usuaris del sistema.

### 3 Consideracions

De forma addicional a les especificacions anteriors, cal tenir en compte les restriccions següents.

#### Estructures de dades

No és necessari que codifiqueu les estructures de dades, com per exemple una llista dinàmica. Podeu usar les classes del paquet `java.util` de l'API de Java, com les classes `LinkedList<E>`, `ArrayList<E>`, `HashMap<K,V>` i/o d'altres.

D'aquesta manera obtindreu, "gratuïtament", estructures de dades amb funcionalitats interessants, lliures d'errors i ràpides, amb el que agilitzareu molt el procés de codificació de la pràctica.

#### Dibuix de gràfiques

Com ja s'ha comentat, per tal de dibuixar les gràfiques no es pot fer ús de cap llibreria externa al SDK de Java (JDK). Cal utilitzar els mecanismes que ofereix la llibreria AWT/SWING de Java i escalar els eixos en funció dels valors màxim i mínim del conjunt de dades que es representen.

#### Disseny gràfic

L'aspecte visual del programa és totalment lliure, sigueu creatius. Investigueu el potencial que us ofereix Java, utilitzeu menús, separadors, desplegable, colors, creeu imatges personalitzades pels botons, etcètera. Tot disseny és implementable.

En programació, la clau sovint resideix en l'abstracció, pel que podeu intentar generalitzar components amb l'objectiu de reutilitzar-los.

#### Procés de desenvolupament

Abans de posar les mans sobre el teclat i començar a codificar, penseu. Analitzeu què cal implementar i dissenyeu quines classes i quines relacions tindrà el sistema, agafeu paper i llapis o utilitzeu l'eina StarUML per tal de crear un diagrama de classes UML base.

Segurament aquest diagrama no serà el definitiu, i anirà canviant a mida que avanceu el desenvolupament, però d'aquesta manera assentareu unes bases sòlides pel desenvolupament del sistema i us estalviareu hores de codificació i molts maldecaps.

El desenvolupament d'un projecte informàtic real típicament comprèn les fases: presa de requeriments, especificació, anàlisi, disseny, implementació, proves i implantació. Podeu pensar que les dues fases inicials d'aquesta seqüència ja han estat realitzades per l'equip docent i que ara us passem el relleu per tal que acabeu el sistema.

**El no compliment d'algun dels punts especificats en aquest enunciat suposarà la no acceptació de la pràctica i la devolució de la mateixa.**  
**La detecció de copia comportarà suspendre tots els integrants dels grups implicats, tant els que han copiat com els que han estat copiats, i perdre l'opció de presentar-se a la pròxima convocatòria ordinària. (veure normativa de la universitat)**