

# **Disseny i programació orientats a objectes**

**Pràctica del segon semestre**

***eSpotyfai***

**Departament d'Enginyeria  
La Salle – Universitat Ramon Llull  
5 d'abril de 2016**

# Índex

<b>1</b>	<b>eSpotyfaí .....</b>	<b>3</b>
<b>2</b>	<b>Especificació de requeriments.....</b>	<b>3</b>
2.1	Servidor .....	3
2.1.1	Configuració .....	3
2.1.2	Connexions .....	4
2.1.3	Interfície gràfica.....	5
2.1.4	Gestionar música .....	5
2.1.5	Gestió usuaris .....	5
2.1.6	Base de dades.....	6
2.2	Client.....	6
2.2.1	Configuració .....	6
2.2.2	Registre i autenticació .....	6
2.2.3	Música disponible.....	7
2.2.4	Llistes de música pròpies.....	7
2.2.5	Follow .....	8
2.2.6	Llistes de música <i>following</i> .....	8
2.2.7	Reproductor de música .....	8
2.2.8	Gestió fitxers locals.....	8
<b>3</b>	<b>Consideracions .....</b>	<b>9</b>

## 1 eSpotyfai

*eSpotyfai* neix amb la idea d'oferir una plataforma perquè els usuaris puguin escoltar la seva música preferida allà on vulguin. Aquesta música serà emmagatzemada en un servidor i servida als clients de forma legal segons la legislació vigent sobre els drets d'autors i de les discografies del país o països afectats.

L'arquitectura de la plataforma *eSpotyfai* segueix un esquema de disseny client-servidor (observeu l'esquema següent). El servidor emmagatzema les dades dels usuaris (les seves credencials d'accés i les llistes de reproducció) i tota la música disponible del sistema. Pel que fa al client, és el programa que permet als usuaris accedir al sistema, llistar i escoltar la música disponible, així com d'altres funcionalitats com votar cançons i altres de tipus xarxa social.

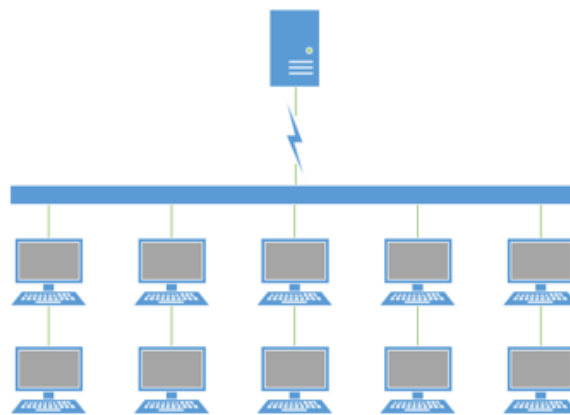


Figura 1. Representació de l'arquitectura client-servidor d'eSpotyfai.

## 2 Especificació de requeriments

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

Tot seguit es pot veure la separació entre les aplicacions client i servidor (expressades a la figura 1), i en cada secció el recull de les funcionalitats que ha de presentar cada programa.

### 2.1 Servidor

El servidor és l'encarregat de gestionar les connexions dels clients i oferir els recursos perquè aquests puguin recuperar la informació necessària per al funcionament normal de l'aplicació client. El programa servidor ha d'emmagatzemar la informació referent a tots els clients o usuaris i a tota la música disponible al sistema.

#### 2.1.1 Configuració

En el moment d'engegar el servidor aquest haurà de llegir un fitxer de configuració anomenat *config.json* ubicat al directori arrel del projecte Servidor. En aquest hi constarà la informació:

- Port de connexió amb la Base de dades
- Direcció 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

- Port de comunicació on escoltar les peticions dels programes Client

### 2.1.2 Connexions

El servidor ha de ser capaç d'atendre les connexions dels clients que es vulguin connectar al sistema. Al servidor es podran connectar múltiples clients al mateix temps i aquest ha de ser capaç d'oferir els mateixos serveis a cadascun d'ells de forma equitativa.

Durant la connexió o sessió, el client podrà escoltar música, crear llistes de reproducció de música, agregar cançons a les seves llistes, etcètera. Cada acció d'aquestes ha de comprendre una comunicació específica amb el servidor, doncs aquest haurà d'actualitzar la informació del client a la base de dades del servidor perquè aquesta estigui disponible per la pròxima sessió.

El sistema ha d'acceptar que un client es connecti sense ser usuari del sistema només en el cas en que es vulgui registrar. Per al registre d'un usuari el programa client enviarà al servidor la informació que l'usuari hagi facilitat mitjançant un formulari de registre. Aquesta informació serà processada per validar que és correcte i en cas afirmatiu serà emmagatzemada a la base de dades. Cal tenir en compte que els noms d'usuari han de ser únics i que la paraula de pas o contrasenya ha de tenir una longitud mínima de 6 caràcters i presentar lletres i números.

Quan un usuari registrat es connecta, s'ha de permetre que aquest accedeixi a tota la seva informació. Les accions que ha de poder dur a terme, entre d'altres, són:

1. Obtenir el llistat de totes les cançons del sistema.
2. Consultar i modificar llistes de reproducció de música.
3. Escoltar qualsevol cançó del sistema, ja sigui d'una llista de reproducció pròpia o bé d'algun dels seus *followings* (altres usuaris que segueix l'usuari).
4. Votar qualsevol cançó del sistema.
5. Seguir a altres usuaris (fer un *follow*) o bé deixar-los de seguir (fer un *unfollow*).

El servidor ha d'oferir al client les eines perquè aquest es pugui descarregar una cançó per posteriorment reproduir-la de forma local a l'ordinador del client. Quant un client vulgui descarregar-se una cançó, el servidor enviarà el fitxer de música cap al client conjuntament amb tota la informació d'aquesta cançó: el nom, l'artista, l'àlbum i el gènere.

És important notar el fet de que el programa servidor és l'únic programa que té accés a la base de dades, que emmagatzema i té accés als fitxers de totes les cançons i que la lògica de funcionament del sistema comprèn l'intercanvi d'informació entre els programes servidor i client mitjançant la comunicació TCP amb *sockets*. Aquestes característiques es poden apreciar en el següent diagrama de desplegament UML.

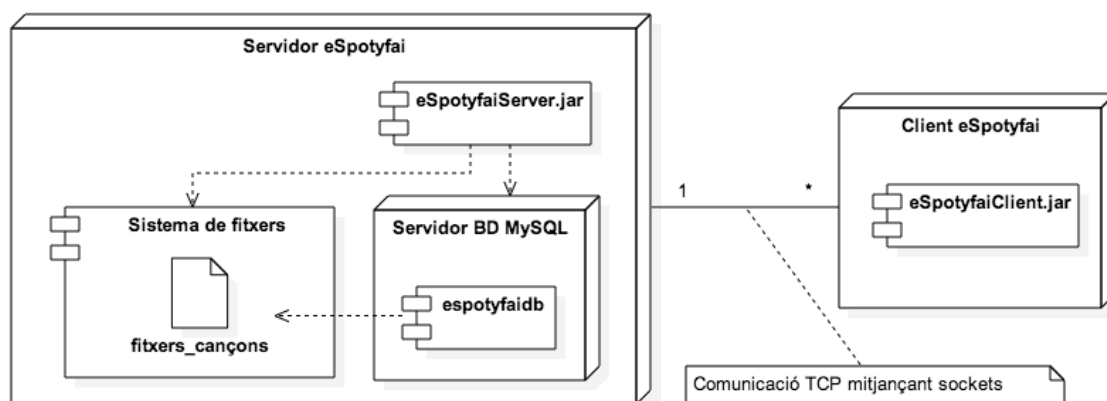


Figura 2. Diagrama de desplegament UML d'eSpotyfai.

### 2.1.3 Interfície gràfica

El servidor, a part d'atendre de forma indefinida les peticions dels clients explicades anteriorment, ha de disposar d'una interfície gràfica que permetin executar funcionalitats per:

1. Gestionar la música
2. Gestionar els usuaris

A continuació s'expliquen les característiques de cadascuna d'aquestes seccions així com les funcionalitats que han d'oferir.

### 2.1.4 Gestionar música

Aquesta secció és orientada a llistar, afegir, eliminar i veure estadístiques sobre les cançons del sistema. És l'opció que permet gestionar la música que s'ofereix als clients.

1. **Llistar i eliminar música.** La interfície gràfica ha de permetre visualitzar un llistat on hi figurin totes les cançons disponibles en el sistema. Les cançons s'han de mostrar en format taula (utilitzant el component *JTable*) i per cadascuna s'ha de mostrar: el nom, el gènere, l'àlbum, l'artista i s'ha de permetre la seva reproducció. L'eliminació de les cançons també s'ha de poder realitzar des d'aquesta mateixa secció, concretament, fent clic amb el botó dret sobre la fila de la cançó que es vulgui esborrar, moment en el qual es demanarà la confirmació de la seva eliminació a l'usuari. La cançó únicament s'haurà d'esborrar de la base de dades, no del sistema de fitxers o disc.
2. **Addició de música.** Per tal d'afegir una cançó es demanarà a l'usuari el nom, el gènere, l'àlbum, l'artista i la ubicació o *path* del fitxer dins del sistema de fitxers o disc del servidor. Si les dades són correctes, llavors el programa ha de procedir a emmagatzemar aquestes dades a la base de dades relacional MySQL. Un cop la cançó ha estat afegida, aquesta és immediatament disponible per a tots els usuaris i el número d'estrelles i de vots és igual a 0. Es recomana crear un directori a la mateixa arrel del projecte Eclipse del servidor on emmagatzemar els fitxers de les cançons.
3. **Estadístiques.** L'opció d'estadístiques ha de permetre visualitzar el top 10 de les cançons que tenen el major nombre de reproduccions. Pel que fa a l'organització dels eixos, l'eix de les 'Y' ha de representar el nombre de reproduccions i l'eix de les 'X' per cada increment una cançó diferent amb el seu nom. El gràfic pot ser de línies o bé de barres i haurà de ser generat des del programa, utilitzant les eines de dibuix que ofereix la llibreria AWT/SWING de Java. No es poden utilitzar llibreries externes de Java per aquest propòsit.

### 2.1.5 Gestió usuaris

Aquesta secció del servidor ha de permetre visualitzar els usuaris registrats al sistema. Aquests s'han de mostrar mitjançant una taula (component *JTable*), i per a cadascun d'ells s'ha de mostrar: el nom d'usuari, la data del registre, la data del darrer accés al sistema, el número de llistes de cançons, el número total de cançons que té a les seves llistes i el número de *followers* (usuaris que el segueixen) i *following* (usuaris que segueix).

Com es veurà més endavant, un usuari podrà seguir a altres usuaris (*follow*) i veure les seves llistes de reproducció públiques. Des d'aquesta secció s'han de poder visualitzar tots els seguidors o *followers* d'un determinat usuari. Per a fer-ho caldrà fer clic amb el botó esquerra sobre una fila, moment en el qual ha d'aparèixer un menú contextual amb l'opció de veure els seus *followers*. Si se selecciona l'opció, apareixerà una segona taula en la mateixa vista on es

vegi el llistat dels seus seguidors. Per a cadascun es mostrarà la mateixa informació que a la taula anterior. En aquest cas no tindrà efecte fer clic damunt de les files.

Des d'aquesta interfície també s'ha de poder eliminar un usuari. Per a fer-ho es podrà fer clic amb el botó dret a qualsevol de les dues taules anteriors (usuaris o *followers*) moment en el qual ha d'aparèixer un menú contextual amb l'opció d'eliminar l'usuari. Cal tenir en compte que l'eliminació d'un usuari ha de comportar l'eliminació de tota la seva informació de la base de dades.

### 2.1.6 Base de dades

La informació referent als usuaris, a les llistes de reproducció dels usuaris i a les cançons disponibles en el sistema s'ha d'emmagatzemar en una base de dades relacional MySQL.

Les dades associades a un usuari són, com a mínim: el *nickname*, la paraula de pas, la data del registre i la data del darrer accés al sistema (s'ha d'actualitzar per cada accés que faci l'usuari al sistema). També s'ha de poder guardar de cada usuari les llistes de reproducció que posseeix, les cançons que hi ha a cada llista així com les relacions entre els usuaris (*followers* i *following*). De cada cançó s'ha de guardar, el nom, el gènere, l'àlbum, l'artista, la ubicació del fitxer dins el disc del servidor i el número d'estrelles (resultat del còmput de les votacions dels usuaris).

## 2.2 Client

L'aplicació client és el punt de connexió de cada un dels usuaris al sistema amb el servidor del sistema eSpotifyai. Aquesta és l'encarregada d'oferir a l'usuari final les possibilitats de registrar-se, autenticar-se, escoltar música, crear llistes de reproducció, així com altres funcionalitats.

### 2.2.1 Configuració

En el moment d'engegar-se, el programa client haurà de llegir un fitxer de configuració anomenat *config.json* on es podrà consultar: (1) la direcció IP i (2) el número de port de connexió amb el servidor.

Un cop configurat apareixerà una interfície gràfica amb opcions per:

1. Registrar-se o autenticar-se
2. Visualitzar la música disponible en el sistema
3. Visualitzar les llistes de música pròpies
4. Seguir (*follow*) o deixar de seguir a altres usuaris (*unfollow*)
5. Visualitzar les llistes de música dels usuaris que se segueix
6. Escoltar música

Aquestes funcionalitats s'expliquen en els apartats següents.

### 2.2.2 Registre i autenticació

L'aplicació del client ha de permetre l'autenticació de l'usuari, o en cas que aquest no tingui un compte creat, haurà de facilitar l'opció perquè aquest es registri.

En cas de seleccionar *nou registre*, el programa ha de mostrar un formulari, esperar a que l'usuari introdueixi les dades, notificar les dades del nou registre al servidor i esperar resposta. Cal considerar que la resposta pot ser satisfactòria (no hi ha cap usuari amb el mateix *nickname* donat d'alta i la paraula de pas té 6 o més caràcters i conté lletres i números) o

insatisfactòria. Si la resposta ha estat satisfactòria, llavors l'usuari formarà part dels usuaris del sistema i a partir d'aquest punt podrà usar les diferents opcions del programa i podrà tancar la sessió.

Si l'usuari disposa d'un *nickname* i d'una paraula de pas (perquè anteriorment s'havia registrat), llavors seleccionarà l'opció *accedir*, la qual demanarà les credencials i procedirà a l'autenticació mitjançant la comunicació amb el servidor. Un cop s'hagi autenticat satisfactòriament, llavors l'usuari podrà executar les diferents opcions del programa (explicades a continuació).

### 2.2.3 Música disponible

Un cop l'usuari ha estat autenticat, la interfície gràfica del programa client ha de permetre accedir a una secció des d'on es pugui visualitzar la música disponible al sistema. Aquest llistat de cançons ha d'estar en tot moment actualitzat vers la música disponible al servidor.

Les cançons s'han de mostrar en format taula (usant el component *JTable*) i per cadascuna s'ha de mostrar: el nom, el gènere, l'àlbum, l'artista, el número d'estrelles (de 0 a 5) i el número de reproduccions. Des d'aquesta secció s'ha de permetre la reproducció de qualsevol cançó i s'ha de poder afegir qualsevol cançó a les llistes de música de l'usuari fent clic al botó dret sobre la cançó que es vulgui afegir. Quan es faci clic amb el botó dret apareixerà un menú contextual amb l'opció per l'addició a alguna de les llistes de reproducció actuals de l'usuari.

### 2.2.4 Llistes de música pròpies

Des d'aquesta secció l'usuari ha de poder:

1. Crear llistes de música
2. Visualitzar les llistes i la música que aquestes contenen
3. Eliminar i modificar les llistes
4. Votar cançons

Cal destacar que a una llista de reproducció s'hi afegeixen i eliminen cançons a partir de l'opció anterior. Per tant l'addició es fa des de la secció "Música disponible". En canvi l'eliminació es fa des de la secció explicada en aquest apartat, "Llistes de música".

La creació d'una llista de música ha de demanar a l'usuari el nom de la llista i si aquesta és pública o no. El fet d'indicar si és pública determina si aquesta llista serà accessible (en mode lectura) pels *followers* de l'usuari en qüestió. Un cop indicada aquesta informació, el programa ha de procedir a la seva creació i conseqüent actualització del llistat de llistes de música.

La visualització de les cançons de cada llista s'ha de realitzar en format taula, anàlogament a la taula que s'ha explicat a l'apartat anterior.

L'eliminació de les cançons de les llistes s'ha de poder realitzar mitjançant una de les opcions d'un menú contextual. Concretament, un menú que aparegui quan es faci clic amb el botó dret sobre la fila de la cançó corresponent de la taula del llistat de cançons.

Les cançons llistades en cada llista de música s'han de poder votar. La votació es farà amb 5 opcions del menú contextual (1, 2, 3, 4 i 5 estrelles) quan es faci clic sobre la fila de la cançó amb el botó dret.

És important observar el fet de que qualsevol modificació realitzada a les llistes de música de l'usuari s'ha de comunicar al servidor, donat que són dades vinculades al compte d'usuari i cal emmagatzemar-les a la base de dades del servidor en tot moment.

### **2.2.5 Follow**

La interfície del programa client ha d'oferir una secció on es puguin veure els usuaris que s'estan seguint, amb opcions per a deixar-los de seguir i també permeti cercar usuaris donats d'alta en el sistema i fer-se'n seguidor (*follow*).

Quan un usuari es fa seguidor (*follower*) d'un altre, aquest veu les llistes de reproducció públiques de l'altre. I evidentment, quan deixa de ser-ho, deixa de veure les seves llistes. Val a dir que tant el fet de fer-se seguidor com deixar de ser-ho no és recíproc i que les llistes de música públiques dels usuaris que s'estan seguint són accessibles des de la secció que s'explica a continuació.

### **2.2.6 Llistes de música *following***

Aquesta secció del programa client és molt semblant a "Música disponible" però mostrant les llistes de tots els usuaris que es segueixen i amb menys opcions.

El llistat de les llistes de música *following* han de permetre saber a quin usuari pertanyen, i han d'estar en tot moment actualitzades vers les llistes de reproducció dels usuaris corresponents.

Les cançons s'han de mostrar en format taula (usant el component *JTable*) i per cadascuna s'ha de mostrar: el nom, el gènere, l'àlbum, l'artista, el número d'estrelles (de 0 a 5), el número de reproduccions i s'ha de permetre la seva puntuació i reproducció. En canvi, no es podran eliminar ni afegir altres cançons en elles.

### **2.2.7 Reproductor de música**

Des de les seccions "Música disponible", "Llistes de música pròpies" i "Llistes de música *following*" s'ha de visualitzar sempre un conjunt de controls que permetin escoltar la cançó seleccionada. Aquest reproductor de música ha d'oferir controls per:

1. Reproduir/Parar. Permet aturar o posar en marxa la reproducció d'una cançó.
2. Anterior/Següent. Permet avançar o retrocedir en les cançons d'una llista.
3. Repetició individual. Permet activar un mode en què la cançó que s'està reproduint en acabar torni a començar.
4. Repetició global. Arribat al final d'una llista torna a començar per la primera cançó d'aquesta llista.

### **2.2.8 Gestió fitxers locals**

Quan finalitza la reproducció d'una cançó, ja sigui des de la secció "Música disponible", "Llistes de música pròpies" o "Llistes de música *following*" el fitxer de la cançó s'ha d'eliminar immediatament del disc del client sense deixar rastre.



### 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>`, `Vector<E>` 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

Per tal de dibuixar les gràfiques no es pot fer ús de cap llibreria externa al SDK de Java (JDK). Cal utilitzar algun dels 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 tan del programa servidor com client és totalment lliure, sigueu creatius. Investigueu el potencial que us ofereix Java, utilitzeu menús, separadors, desplegable, colors, creeu imatges personalitzades pels botons, etcètera.

#### 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, que continueu amb l'anàlisi, disseny, etcètera.

**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 el 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)**

