

eSpotyfai

Projecte final



**Disseny i programació orientat a
objectes**

Grup: B9

Samuel Tavares	- Is30659
Grabriel Cammany	- Is30652
Jorge Melguizo	- Is29135
Iván Romero	- Is31013
Carla Urrea	- Is30627

Índex

1. Especificacions del projecte
2. Disseny de la interfície gràfica
 - 3.1. Servidor
 - 3.2. Client:
3. Model de disseny
 - 3.1. Diagrama de classes
 - 4.1.1 Client
 - 4.1.2 Servidor
 - 3.2. Descripció
 - 4.2.1. Client
 - 4.2.2. Servidor
4. Metodologia de desenvolupament
5. Dedicació
6. Conclusions
7. Bibliografia

1. Especificacions del projecte

El projecte escollit ha estat l'eSpotyfai. Com es pot intuir mitjançant el propi nom, es tracta de la realització d'una plataforma similar al Spotify real.

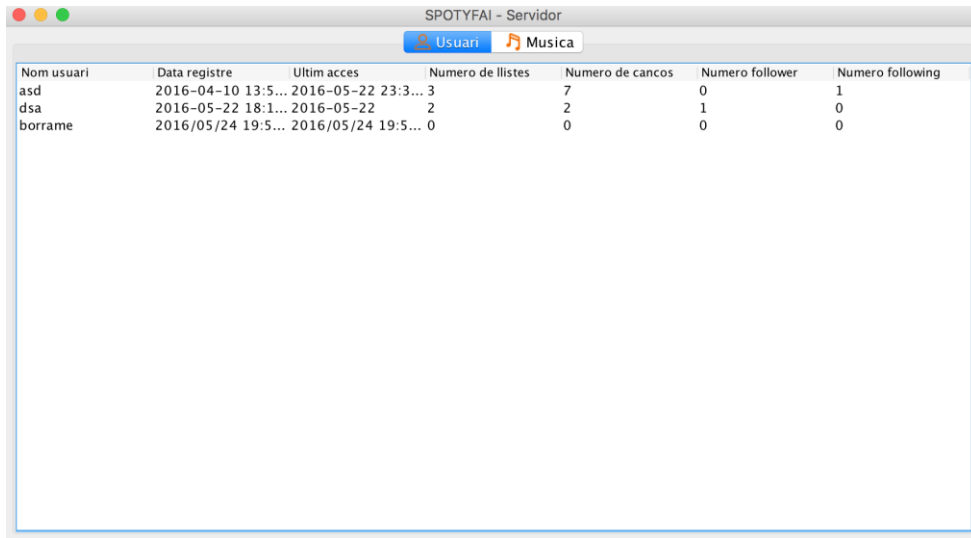
Consisteix en la realització d'un programa que permeti als diferents usuaris escoltar la música emmagatzemada en un servidor centralitzat. Per tant, l'arquitectura del projecte serà client-servidor.

El client disposarà d'una interfície que li permetrà realitzar diferents tasques. Per una banda, a l'inici, disposarà de l'opció d'accedir com un client ja registrat o registrar-se en cas que sigui la primera vegada que l'utilitza. Per altra banda, una vegada accedeix a la plataforma, el client podrà navegar per les diferents opcions mitjançant una interfície de reproducció. Podrà escoltar i pausar les diferents cançons disponibles, controlar el volum del so, crear llistes tant públiques com privades i afegir o eliminar música de les llistes. A més, podrà interactuar socialment ja que disposarà de l'opció de fer following a altres usuaris, escoltar les llistes públiques dels usuaris que segueix i votar les cançons disponibles al sistema.

El servidor es centrarà en la gestió de la base de dades i dels arxius de música. Cada vegada que l'usuari es registri, voti una cançó, faci following a un usuari etc. el servidor anirà actualitzant la base de dades. A més, quan l'usuari demani una cançó per reproduir-la, el servidor s'encarregarà d'enviar l'arxiu mp3 al client. També cal destacar que mitjançant la interfície del servidor podrem visualitzar els usuaris registrats, la música disponible al sistema i unes estadístiques formades amb les 10 cançons més escoltades pels diferents usuaris.

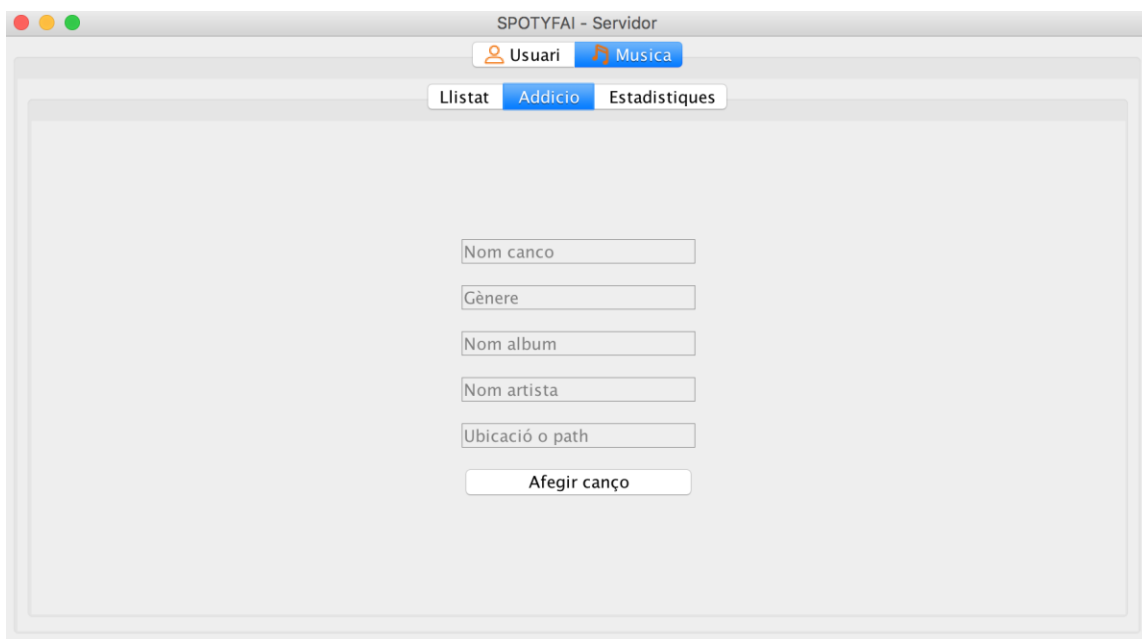
2. Disseny de la interfície gràfica

2.1. Servidor



Nom usuari	Data registre	Ultim acces	Numero de llistes	Numero de cancos	Numero follower	Numero following
asd	2016-04-10 13:5...	2016-05-22 23:3...	3	7	0	1
dsa	2016-05-22 18:1...	2016-05-22	2	2	1	0
borrame	2016/05/24 19:5...	2016/05/24 19:5...	0	0	0	0

Aquesta vista es un JFrame amb un JTabbedPane amb dues opcions (usuari i musica). En l'opció d'usuari hi ha un JScrollPane on dins l'inserim una DefaultTableModel amb l'informació dels usuaris.



SPOTYFAI - Servidor

Usuari Musica

Llistat Addicio Estadistiques

Nom canco

Gènere

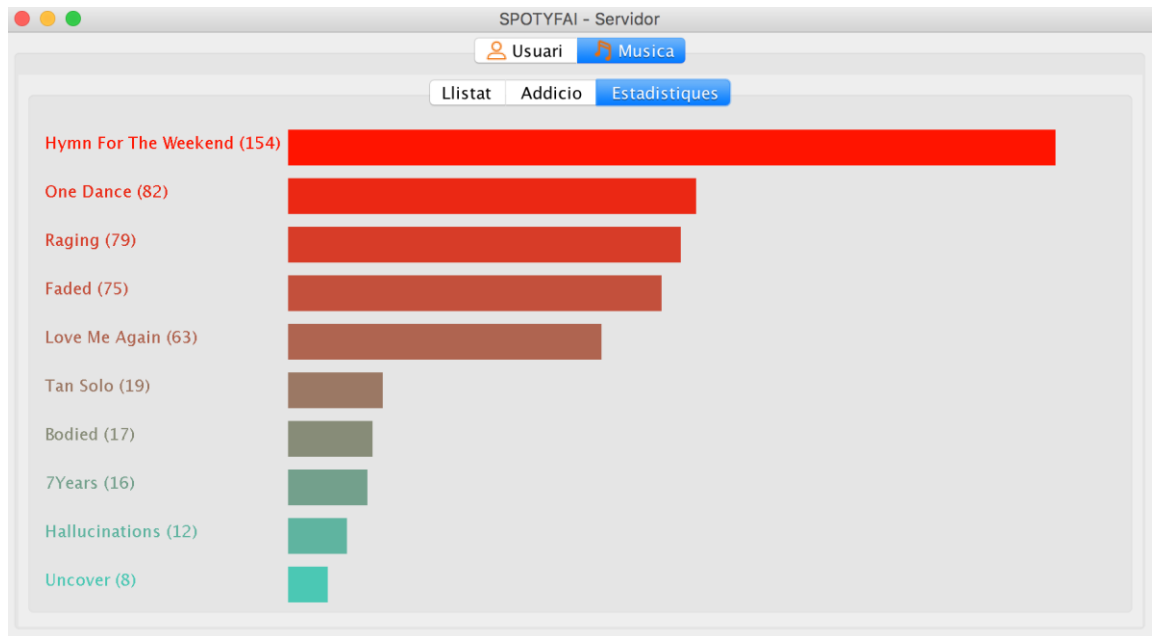
Nom album

Nom artista

Ubicació o path

Afegir canço

Dins de l'opció de música trobem un altre JTabbedPane amb tres opcions. L'opció de llistat conté una taula com l'anterior. En l'opció d'addició hi ha un JPanel amb un MigLayout i 5 JTextField amb un JButton final per afegir la canço.



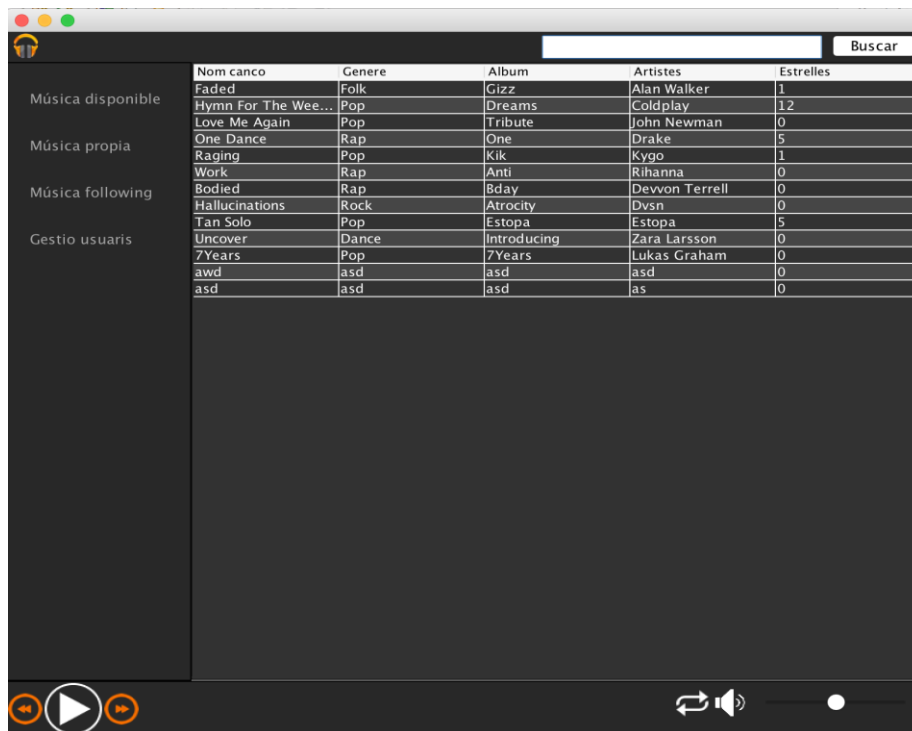
A l'opció d'estadístiques tenim un JPanel on es pinta els gràfics amb la utilització de l'eina *Graphics* sino que ho fem mitjançant les eines gràfiques que ofereix Swing/AWT com els rectangles.

2.2. Client:



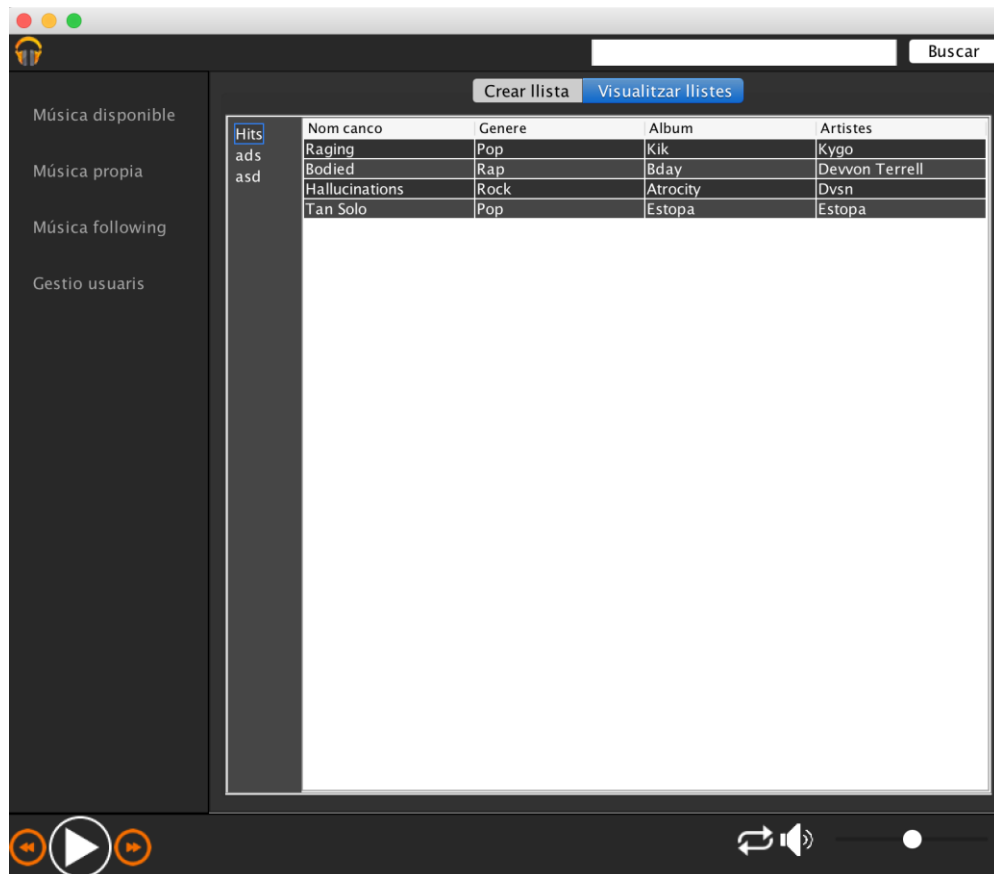
Aquesta opció es la mateixa que en la finestra de login. Es un JFrame on te tenim una ImageIcon posada dins d'un JLabel i li assignem com ContentPane del JFrame. La resta es un JPanel amb MigLayout i l'assignem al centre un JTextField, un JPasswordField i un JButton.

Per aquesta finestra hem emprat un JFrame on li hem introduït un JPanel amb BorderLayout. En el nord li hem assignat un altra JPanel amb BorderLayout on a l'esquerra li hem posat el logo amb un ImageIcon dins d'un JLabel i a la dreta un JTextField i un JButton. A l'esquerra hem posat 4 JLabels amb un action listener. En el sud hem inserit un JPanel amb un altre BorderLayout on a l'esquerra li hem posat un JPanel amb MigLayout amb els tres botons que son 3 ImageIcon i a la dreta un Image Icon amb un JSlider pel volum. En el centre li hem posat un JScrollPane amb un TableModel amb l'informació de les cançons.



Per la resta de finestres canviem el JScrollPane del centre. En aquest cas li hem afegit un JTabbedPane on a l'opció de visualitzar llistes hi ha un JPanel amb BorderLayout on a l'esquerra li hem posat un JList i al centre un TableModel.

Les TableModel estan renderitzades per canviar-li el color



3.2. Descripció

4.2.1. Client

Del client, a la part de la vista, hem de destacar sobretot al classe **FinestraReproduccio**. Aquesta es la que conté tots el procediments i components per tractar amb la part gràfica de la part del client. Per altra banda, **FinestraLogin** i **FinestraRegistre** s'encarreguen de la part gràfica de la finestra de login o registre.

A la part del model, una de les classes més important es **Reproductor**, ja que és l'encarregada de la lògica del reproductor de música del nostre projecte. També trobem d'altres com **Canco**, **Llista** i **User** utilitzades per tractar amb les diferents unitats d'informació.

Respecte a la part dels controladors, un dels més importants es **ControladorReproductor** el qual comunica **FinestraReproduccio** i **Reproductor**. També s'inclouen altre controlador com, per exemple **ControladorFinestres** o **ControladorBotons** encarregats de gestionar tasques més específiques com per exemple el clic dret a les llistes o la recuperació de les dades quan l'usuari fa *follow* a un altre.

Per últim, a la part de *Network* té un gran pes en aquest projecte ja que l'usuari no emmagatzema dades i les demana periòdicament al servidor. Per aquesta tasta tenim **InfoServidor** que controla i estableix la connexió per *Socket* amb el servidor.

4.2.2. Servidor

A la part gràfica del servidor, la classe **FinestraServidor**, permet la creació de la finestra principal del servidor i dels seus diferents apartats com Musica o Usuari, amb un afegit que és que es poden reproduir les cançons des del mateix servidor.

A la part de model del servidor trobem classe, anàlogament al client, com **User**, **Canco**, **Llista** per l'estructuració de informació. Moltes d'aquestes estan *serialitzades* per poder ser enviades per Sockets. A més, tenim la classe **Sessio** que permet al servidor emmagatzemar informació necessària de les diferents sessions que estan actives concurrentment. Per últim, També trobem la classe **Reproductor** ja que el servidor també disposa de l'opció de reproduir cançons i de la classe **Query** encarregada únicament de contenir els diferents formats de peticions que es poden realitzar a la base de dades.

Respecte a la part de controladors, destaquem la classe **SocketController**, aquesta s'encarrega d'escoltar les peticions que els clients li fan i també de respondre-li aquestes peticions, tant quan el client demana les llistes, que li respon tornant-li els id de les cançons corresponents, com quan el client vol reproduir alguna cançó des de la seva finestra, que en aquest cas el servidor li respon passant-li l'arxiu de la cançó.

Per últim, de la part de *Network* del servidor destaquem la classe **ConnectorDB**, és la classe la qual s'encarrega de fer totes les peticions a la base de dades, tant com per a comprovar si un usuari que intenta fer login existeix o no, com també per a demanar les cançons, llistes, followings, etc.. A més, per permetre la connexió amb els diferents clients també disposa de la classe **MessageService**.

4. Metodologia de desenvolupament

En primer lloc, el primer pas que vam realitzar a l'hora de començar amb el desenvolupament del projecte va ser la **lectura de les especificacions** i el **disseny** general del nostre projecte. En aquesta primera fase, no vam centrar-nos en el repartiment de les tasques ja que el millor era que totes participessin de igual forma aportant diferents opinions i idees. Durant aquesta fase vam prendre diferents decisions a l'hora de facilitar la codificació del projecte.

Posteriorment, vam enfonsar-nos en una segona fase dedicada a preparar les diferents eines que utilitzaríem al llarg del projecte. Per exemple, la creació del repositori a Github, la connexió i configuració del servidor extern i la creació de la base de dades.

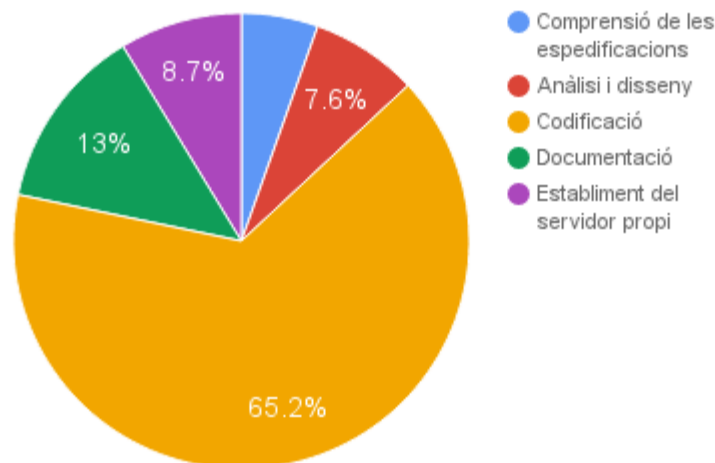
Una vegada tot preparat per iniciar el projecte vam decidir realitzar un repartiment de les tasques entre els diferents integrant. Cadascun es dedicava especialment a una part específica com per exemple la base de dades, les connexions i la gestió de les dades entre servidor-client, les interfícies, etc. Però, a mesura que avançàvem al projecte cada vegada anàvem oblidant aquestes especificacions i intentàvem col·laborar i entendre totalment el nostre codi per poder ajudar als altres integrant del grup si sorgia algun problema. A més, en l'etapa final de la realització del projecte, va ser quan més units vam treballar per intentar compensar la falta de implicació d'alguns integrants.

5. Dedicació

Podem dividir les hores dedicades en el projecte en diferents grups segons les tasques realitzades:

1. Comprensió de les especificacions: aproximadament 5 hores
2. Anàlisi i disseny: aproximadament 7 hores
3. Establiment del servidor propi: aproximadament 8 hores
4. Codificació: aproximadament 60 hores
5. Documentació: aproximadament 12 hores

Dedicació



Com s'aprecia en el gràfic anterior, la majoria del nostre temps dedicat al projecte ha estat invertit en la codificació ja que ha sigut la part més costosa, que més temps requeria i que sobretot on més problemes i dificultats ens hem trobat.

A més, de forma optativa, el nostre grup va decidir optar per emmagatzemar la nostra base de dades en un servidor extern. Aquesta part també ha requerit una part important del nostre temps, sobretot pels problemes de connexió però que, a la llarga, ens ha estalviat temps.

6. Conclusions

Aquest treball ha suposat un primer aproximament al que realment ens trobarem en un futur pròxim a l'hora de treballa amb un grup de persones sobre un mateix projecte.

Durant aquest anys la majoria de pràctiques que hem realitzat han sigut en parelles i han suposat una carrega de treball equitativa. En canvi, amb aquest projecte, la carrega de treball era considerablement superior però hi havien més persones implicades. Més treball, més persones, més opinions diferents i poc temps disponible han sigut factors que han desembocat en algunes situacions d'estres i petits conflictes en el grup. Però, passar per aquestes situacions, ens ha ajudat a comprendre que, en un grup de cinc persones, ser individualista és un greu problema. La clau ha estat en superar aquestes situacions, realitzar una càrrega de treball justa per tots els integrants i sobretot, mantenir una bona comunicació entre nosaltres. Hem experimentat en primera persona que es essencial mantenir a la resta del grup de treball informats de les tasques que hem realitzat i comentar detalladament les nostres modificacions al codi per evitar confusions.

Després de la realització completa del projecte, tots hem compartit l'opinió que no vam dedicar temps suficient a l'hora del disseny del programari inicialment. Creiem que ha sigut el nostre gran error com a grup. Moltes de les dificultats que hem superat en aquest mesos han estat ocasionades per un mal disseny de classes, confusions y falta d'estructuració. Si tornessim a realitzar un projecte d'aquestes magnituds tenim clar que dedicariem el temps necessari a la realització d'un diagrama de classes clar i estructurat abans de començar la progrmació.

Valorem molt la nostra decisió d'utilitzar un servei de hosting de repositoris com Github. A l'inici vam tenir alguns problemes ja que, per nosaltres, era la primera vegada que l'utilitzavem. Una vegada vam començar a dominar-lo i entendre els diferents conflictes aquesta eina ens va facilitar molt les tasques. Podem dir que ara ens sentim prou segurs amb aquesta plataforma i apostem completament per utilitzar-la en qualsevol projecte futur.

En resum, la realització d'aquest projecte final no només ens ha aportat coneixements a l'hora de programar, ens ha apropat a la realitat, als problemes que tindrem en un futur. Ens quedem amb l'ensenyança que els problemes del grup, es resolen en grup.

7. Bibliografia

Tutoriales point [en línia] [Data de consulta: 5 abril 2016]. Disponible a:

http://www.tutorialspoint.com/json/json_java_example.htm

Ejemplos TIW [en línia] [Data de consulta: 8 abril 2016]. Disponible a:

<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>

Línea de código [en línia] [Data de consulta 15 abril 2016]. Disponible a:

<http://lineadecodigo.com/java/reproducir-musica-por-java/>

Tutorial point [en línia] [Data de consulta 12 abril 2016]. Disponible a:

<http://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm>

Mig Layout [en línia][Data de consulta 20 abril 2016]. Disponible a:

<http://www.miglayout.com/cheatsheet.html>

Serialización de objetos [en línia][Data de consulta 1 Maig 2016]. Disponible a:

http://chuwiki.chuidiang.org/index.php?title=Serializaci%C3%B3n_de_objetos_en_java

How to use sliders? [en línia] [Data: 20 de Maig] Disponible a :

<https://docs.oracle.com/javase/tutorial/uiswing/components/slider.html>

Cargando imágenes desde Jva [en línia] [Data: 20 de Maig] Disponible a:

<https://lefunes.wordpress.com/2009/01/29/cargando-imagenes-desde-java/>