

DOCUMENTACIÓN MUSICLIC

DAVID SÁNCHEZ
MORALES

En este texto se describe el proceso de diseño e implementación de la Aplicación para computadora “MusiClick” la cual tiene como objetivo la creación de listas de reproducción de música para escuchar las canciones de tus artistas favoritos.

Contenido

1. Introducción.	2
1.1. Los Géneros.	2
1.2. Los Artistas.	2
1.3. Las Canciones.	2
1.4. Las Listas de Reproducción.	2
1.5. Los Usuarios y las Sesiones.	3
1.6. El Manager.	3
2. Fase de Diseño.	4
2.1. Distribución del Código.	4
2.2. Modelo Entidad-Relación en la Base de Datos.	5
2.3. Diseño de Pantallas.	6
3. Fase de Desarrollo.	11
3.1. Filtrado de recursos mediante uso de expresiones regulares en lenguaje Java.	11
3.2. Administración de la información.	11
3.3. Utilización de elementos multimedia.	11
3.4. Utilización de Eventos y Disparadores en la BBDD.	12
4. Enlaces y más documentación.	13

1.Introducción.

La aplicación de computadora “MusiClick” se trata de un software que permite reproducir una cola de reproducción de canciones que el usuario decida, pudiendo incluso personalizar y crear sus propias listas de reproducción. Además, se ofrece información sobre los artistas, los discos y las canciones.

Además, también se ha desarrollado un pequeño software de gestión de los géneros, artistas, canciones y listas de reproducción predeterminadas.

1.1. Los Géneros.

En MusiClick tenemos una serie de Géneros que nos permiten catalogar de forma eficiente nuestras canciones. Estos géneros pueden crearse, editarse y eliminarse desde el Manager.

Como mínimo ha de existir un género, y cuando se elimina un género de una canción que estaba asociado al mismo, a esta se le atribuye otro género.

1.2. Los Artistas.

Los artistas son los creadores de las canciones. Se ha planteado la aplicación de tal forma que una canción solo pueda pertenecer a un solo artista, de modo que para crear colaboraciones, ha de crearse un artista colaboración con el nombre de todos los participantes.

Si se elimina un artista, se eliminarán también todas las canciones que ha compuesto.

1.3. Las Canciones.

Las canciones son la clase principal del programa. Solo pueden pertenecer a un solo disco y a un solo artista. Las canciones pueden estar en muchas listas de reproducción (solo una vez por lista) ya sea en las predeterminadas del sistema o las creadas por los usuarios.

Al eliminar las canciones, no se eliminarán ni los discos que las contienen ni las listas.

1.4. Las Listas de Reproducción.

Las listas de reproducción contienen canciones en su interior, sin límite alguno. Existen varias listas de reproducción por defecto visibles para todos los usuarios y estas son creadas/modificadas/eliminadas por el administrador de la aplicación mediante el Manager.

Los usuarios podrán crear sus propias listas de reproducción para posteriormente poder reproducirlas. Estas listas serán visibles a los demás usuarios si las crean como públicas o solo serán visibles para ellos mismos si las crean como privadas. Pueden tener tantas listas de reproducción como quieran con tantas canciones como deseen.

1.5. Los Usuarios y las Sesiones.

En MusiClick, los usuarios deberán crearse cuentas individuales. Estas cuentas, necesitarán de un nombre de usuario y mail únicos, así como también de una contraseña.

Los usuarios necesitan validar su email para confirmar su cuenta y posteriormente poder acceder a la opción de recuperación de contraseña.

Un usuario no podrá logearse/conectarse si ya se encuentra conectado, para eso, existen las “Sesiones”, que determinan los usuarios conectados.

1.6. El Manager.

Se trata de una aplicación que se encarga de gestionar los datos de la base de datos de la aplicación. Solo será accesible por el usuario administrador y tendrá acceso a todos los recursos de MusiClick. Esta aplicación, permite crear, modificar y eliminar los géneros, artistas, canciones y listas de reproducción por defecto, no pudiendo editar en ningún caso, las creadas por los usuarios.

2.Fase de Diseño.

Para el diseño e implementación de esta aplicación de tipología Standalone y con características CRUD (CREATE, READ, UPDATE, DELETE), se ha utilizado el lenguaje Java ayudándonos del entorno IDE Eclipse en un proyecto basado en Maven y JavaFX para el diseño de interfaces apoyado en Scene Builder.

Para el diseño del Modelo de Entidad-Relación utilizado en la base de datos hemos utilizado PHPMyAdmin con el objetivo de hacer una aplicación que pueda conectarse a un servidor remoto y obtener y registrar datos en el mismo.

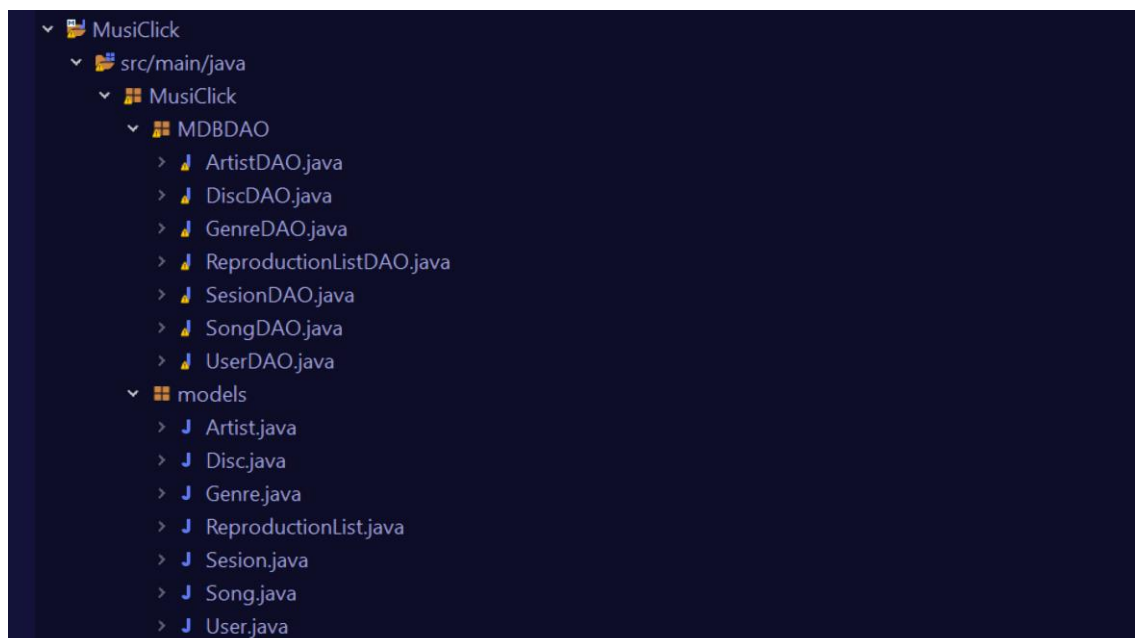
2.1. Distribución del Código.

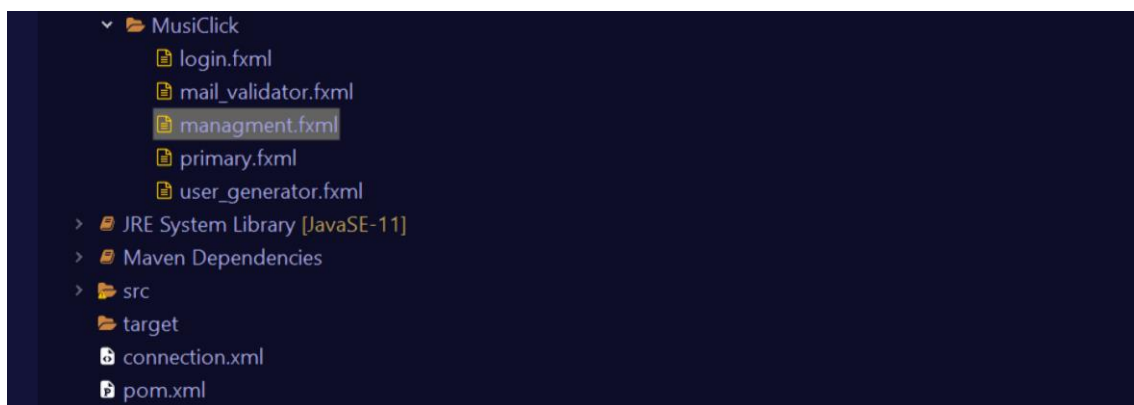
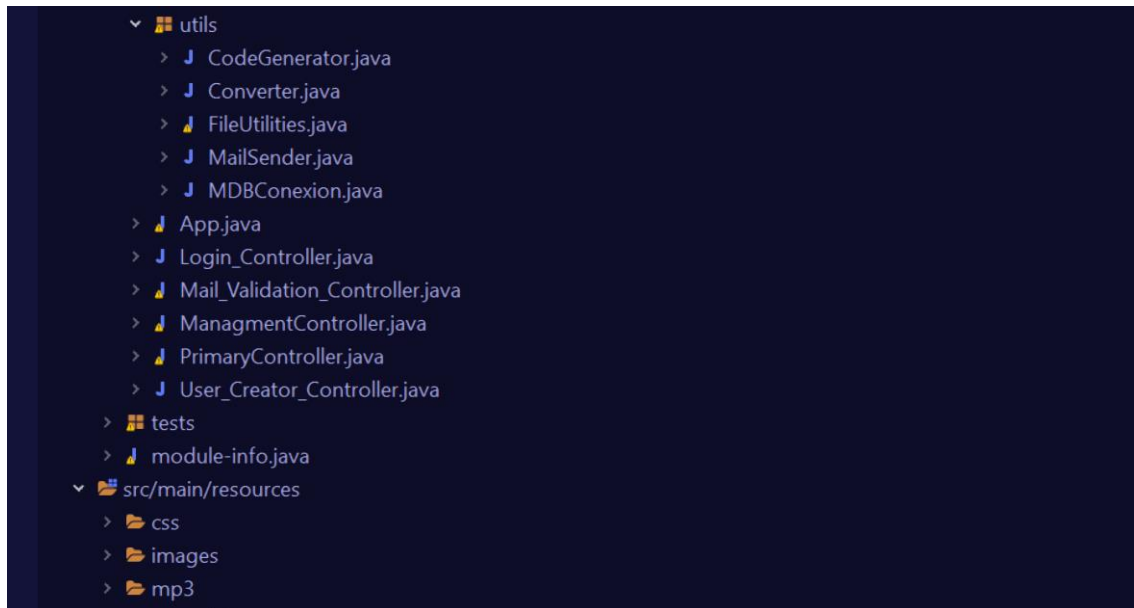
Se ha utilizado como carpeta raíz para los Controllers la carpeta padre “MusiClick”, para los modelos, la carpeta padre “models”, y para los DAO que gestionan la petición al servidor, se han paqueteado en la carpeta MDBDAO.

Además, hemos utilizado otro package para funcionalidades extra como el grabado y eliminación de archivos, así como también la conexión a la base de datos. Este package se llama utils, y alberga las clases FileUtilities, MDBConexion (para la BBDD), MailSender (para el envío de emails, el Converter (convertidor de lista a ObservableList) y el CodeGenerator.

Con respecto a las Vistas, las cuales se han creado mediante elementos FXML, se han guardado en el package resources, en una carpeta llamada “MusiClick”.

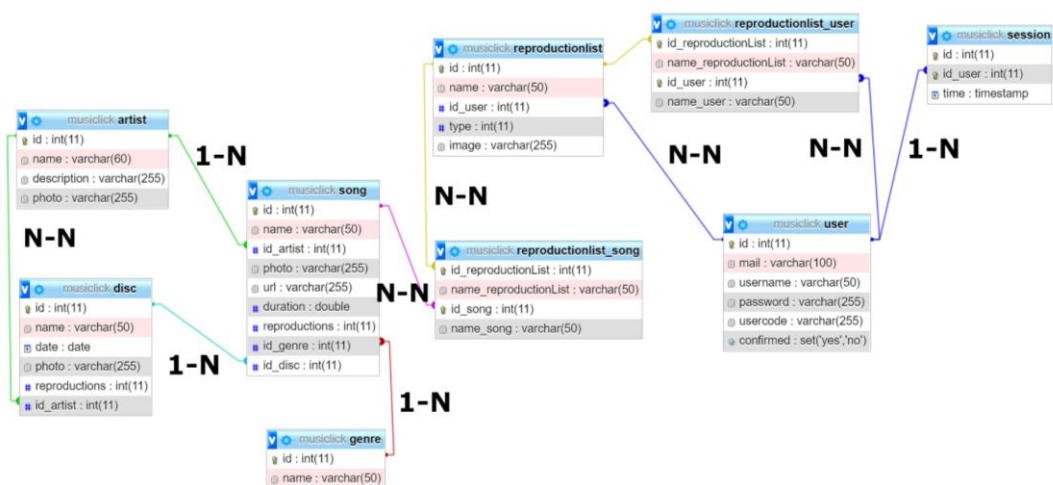
En resources también se encuentran los archivos locales como imágenes o sonidos.





2.2. Modelo Entidad-Relación en la Base de Datos.

El esquema utilizado en la base de datos es el siguiente:



Con respecto a los archivos multimedia, cabe destacar que en la bbdd se almacenan como Strings que albergan la ubicación relativa real del archivo con respecto a la máquina y el usuario que está utilizando la aplicación. Esto se debe a que los archivos multimedia se guardan en la carpeta resources del proyecto. Esto se hace para evitar sobrecarga de datos en la bbdd y mayor rapidez de lectura de los recursos.

La información para la conexión con la base de datos se encuentra en un archivo xml que contiene la información del servidor, base de datos específica a utilizar (en nuestro caso, musiclick), usuario (root) y password (ninguna).

2.3. Diseño de Pantallas.

La aplicación se iniciará como una ventana inicial, a la que se le ha llamado como “login”.

Login

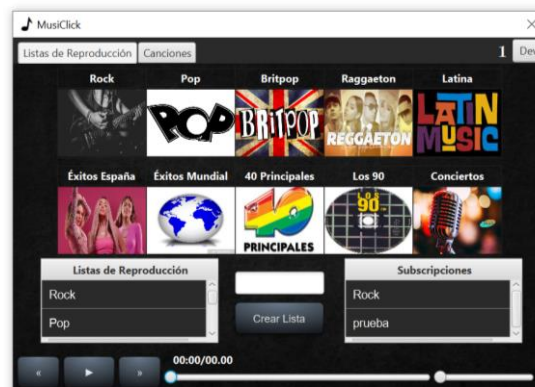
Creación de Usuario

Validación de Correo Electrónico

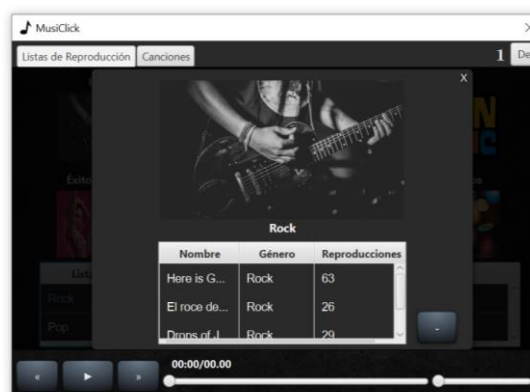
En esta ventana el usuario tendrá la opción de logearse o registrarse.

Cuando el usuario inicia sesión, entrará en la ventana que hemos llamado como “primary”, la cual tiene dos pestañas principales:

- **La pestaña de listas de reproducción:** Es primera y tiene un filtro de listas de reproducción, así como información sobre todas las listas visibles para el usuario y las listas a las que se ha suscrito. Además, también puede acceder a las listas por defecto principales desde el acceso rápido creado en la parte superior.



- **Panel de Lista de Reproducción:** Sirve para ver información sobre una lista de reproducción. También puedes acceder a las canciones de la lista.



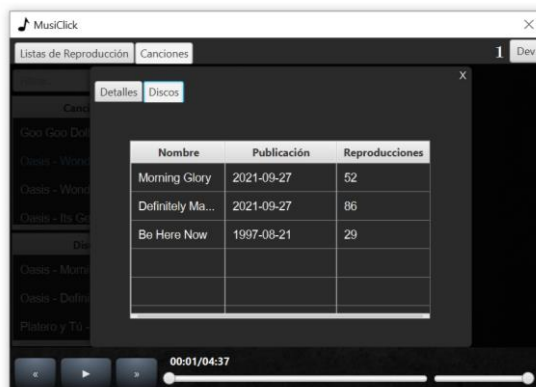
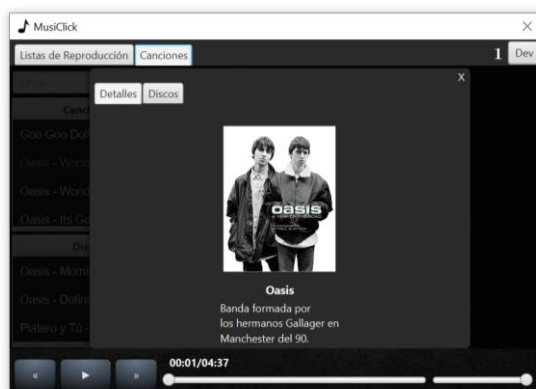
- **Panel de creación Lista de Reproducción:** Tras abrir este panel, el usuario conectado podrá crear su propia lista de reproducción, modificando campos como imagen, nombre y canciones que contendrá la lista. Cuando un usuario crea una lista, se suscribe automáticamente a dicha lista y no podrá cancelar la suscripción a menos que la borre.



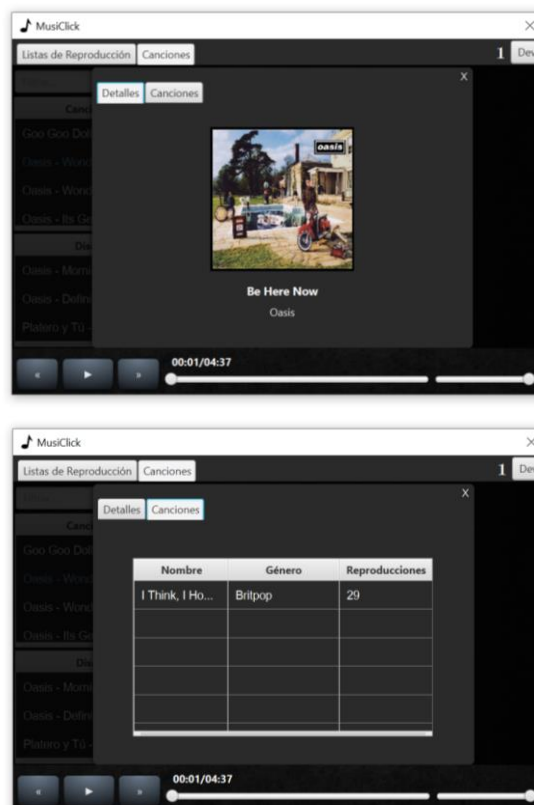
- **Pestaña de Filtro o búsqueda:** Sirve para buscar canciones o discos en el reproductor, pudiendo, además, ver información adicional de los mismos como sus autores, su disco (si lo tiene) o sus reproducciones reales.



- **Panel de Artista:** Muestra información sobre un artista en concreto, como sus discos.



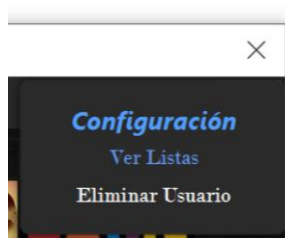
- **Panel de Disco:** En este panel podremos ver información sobre un disco en concreto, como su artista principal y sus canciones principales, pudiendo ver también las reproducciones de las mismas.



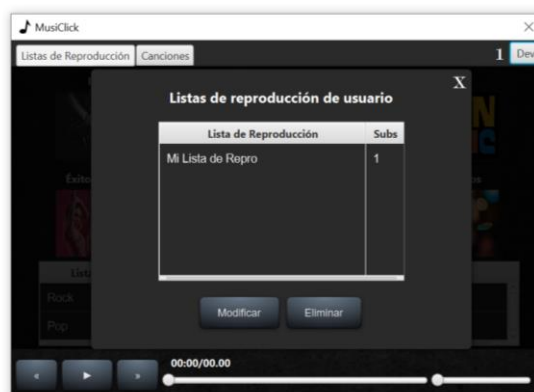
- **Reproductor:** Se encuentra en la parte inferior de la ventana. Siempre es visible para el usuario y sirve para gestionar el reproductor, pudiendo pausar/reanudar la canción en reproducción o saltar a una anterior o siguiente con los botones correspondientes. También puede realizar saltos a tiempos concretos de reproducción con el slider inferior central para ir a un tiempo concreto de reproducción. Con el slider de la derecha podrá ajustar el volumen.



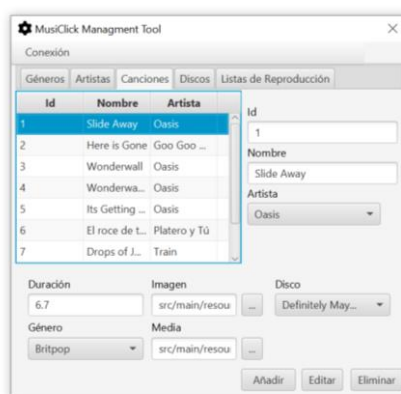
- **Panel de opciones del usuario:** Panel de control de Usuario. Desde aquí, el usuario podrá eliminar su usuario o ver sus listas de reproducción creadas.



- **Panel de Listas de Reproducción del Usuario:** Desde este panel, el usuario podrá gestionar sus listas de reproducción, ya sea eliminándolas o modificándolas, en cuyo caso se le redirigirá a la ventana de modificación/creación de lista.



- **Manager o Software de Administración:** Desde este programa, el administrador podrá añadir, modificar y eliminar registros de la base de datos. Podrá modificar géneros, artistas, canciones y listas de reproducción por defecto. El guardado se hace tras pulsar la opción guardar a la bbdd de forma única y directa.



3. Fase de Desarrollo.

3.1. Filtrado de recursos mediante uso de expresiones regulares en lenguaje Java.

Se han utilizado bastantes expresiones regulares en varios apartados del código.

Destacar, el del uso de estas en el filtrado de búsqueda de Listas, Discos, Canciones y Artistas por nombre en la ventana principal de la aplicación.

Las Expresiones regulares también han sido utilizadas, por ejemplo, para cotejar datos alfanuméricos válidos y no válidos.

3.2. Administración de la información.

Para la administración de la información, normalmente se han utilizado clases estáticas y públicas con todos los datos cotejados en varias clases.

Normalmente, la información es una variable “public static” que se trata de una lista, alcanzable desde la mayoría de los controladores que la comparten.

Todo lo que se crea, edita o modifica en la lista, se ve repercutido en todos los controladores del código sin necesidad de pedir nuevamente información, pues ya disponen de ella con el puntero a esa lista.

Para la carga y sobreescritura de esta información se han utilizado los DAOs, los cuales son extensiones de las clases que manipulan. Estos DAOs tienen varias funcionalidades, como cargar/descargar datos en la base de datos y devolver las peticiones de información de los controladores de la aplicación.

3.3. Utilización de elementos multimedia.

Debido a la naturaleza principal de la aplicación que se ha desarrollado, era indispensable la utilización de recursos multimedia.

Estos recursos se pueden cargar desde el explorador de archivos de Windows gracias a la clase FileChooser, la cual nos permite cargar archivos al programa y posteriormente, mediante un método de la clase FileUtilities, se podrá guardar en la carpeta resources el archivo solicitado, al cual se le pondrá un nombre concreto en función del tipo de recurso, su funcionalidad y el id de la clase que lo esté utilizando.

3.4. Utilización de Eventos y Disparadores en la BBDD.

Se ha requerido la utilización de eventos y disparadores en la BBDD para la gestión de los usuarios y las sesiones, pues, como se ha comentado anteriormente, no puede haber logeado simultáneamente el mismo usuario desde varios equipos diferentes, por lo que, para tratar este problema, se ha utilizado una cuenta atrás del tiempo de logeo de los usuarios y se envía una actualización desde el programa principal cada varios segundos.

El servidor comprueba los usuarios que llevan demasiado tiempo sin recibir información actualizada de sesión y borra aquellos que considere que no están en uso, dando la oportunidad nuevamente de logearse.

4. Enlaces y más documentación.

- Enlace a GitHub: [DavidSM24/MusiClick: Aplicación para escuchar música en computadoras. \(github.com\)](https://github.com/DavidSM24/MusiClick).