



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERÍA TÉCNICA EN INFORMÁTICA**

**DE SISTEMAS**

SMARTDJ: APLICACIÓN MÓVIL PARA LA  
REPRODUCCIÓN DE ARCHIVOS MUSICALES CON  
MEZCLA PERSONALIZADA DE CANCIONES

Raúl Daniel García Molina

Septiembre 2014





**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERÍA TÉCNICA EN INFORMÁTICA**

**DE SISTEMAS**

**SMARTDJ: APLICACIÓN MÓVIL PARA LA REPRODUCCIÓN DE  
ARCHIVOS MUSICALES CON MEZCLA PERSONALIZADA DE  
CANCIONES**

**DEPARTAMENTO: LENGUAJES Y SISTEMAS INFORMÁTICOS**

**DIRECTOR DEL PROYECTO: ALBERTO SALGUERO**

**AUTOR DEL PROYECTO: RAÚL DANIEL GARCÍA MOLINA**



Agradecimientos a mi familia y pareja por su apoyo incondicional.



# Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>                        | <b>14</b> |
| 1.1. Objetivos . . . . .                      | 14        |
| 1.2. Motivación . . . . .                     | 15        |
| 1.2.1. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS | 16        |
| 1.3. Estructura de la memoria . . . . .       | 17        |
| 1.4. PLANTEAMIENTO TÉCNICO . . . . .          | 17        |
| <b>2. Planificación</b>                       | <b>20</b> |
| 2.1. Introducción . . . . .                   | 20        |
| 2.2. Desglose de la planificación . . . . .   | 20        |
| 2.3. Costes . . . . .                         | 21        |
| <b>3. Especificación de requisitos</b>        | <b>23</b> |
| 3.1. Introducción . . . . .                   | 23        |
| 3.1.1. Propósito . . . . .                    | 23        |
| 3.1.2. Ámbito . . . . .                       | 23        |
| 3.2. Descripción general . . . . .            | 24        |
| 3.2.1. Perspectiva del producto . . . . .     | 24        |
| 3.3. Catálogo de requisitos . . . . .         | 24        |
| 3.3.1. Requisitos funcionales . . . . .       | 24        |
| 3.4. Restricciones generales . . . . .        | 26        |
| 3.4.1. Requisitos de la interfaz . . . . .    | 27        |
| 3.4.2. Requisitos de eficiencia . . . . .     | 28        |
| 3.4.3. Otros requisitos . . . . .             | 28        |

---

|   |           |
|---|-----------|
| <b>4. Análisis</b>  | <b>31</b> |
| 4.1. Modelo conceptual de datos . . . . .   | 31        |
| 4.2. Modelo de casos de Uso . . . . .   | 32        |
| 4.2.1. Casos de uso referentes a la página inicial: . . . . .                             | 33        |
| 4.2.2. Casos de usos referentes a menú reproducción . . . . .                             | 39        |
| 4.2.3. Casos de uso referentes a la modificación de la lista de<br>reproducción . . . . . | 45        |
| 4.2.4. Casos de uso referentes a modificar listas de reproducción                         | 50        |
| <b>5. Diseño</b>  | <b>55</b> |
| 5.1. Arquitectura del sistema software . . . . .  | 55        |
| 5.2. Diseño de la capa de gestión de datos . . . . .                                      | 55        |
| 5.3. Diseño de la capa de dominio . . . . .   | 59        |
| 5.3.1. Clases de diseño . . . . .   | 59        |
| 5.3.2. Diseño de secuencia . . . . .  | 64        |
| 5.3.3. Diseño de la capa de presentación . . . . .  | 68        |
| <b>6. Implementación</b>  | <b>81</b> |
| 6.1. Tecnologías . . . . .  | 81        |
| 6.2. Herramientas . . . . .   | 81        |



---

|   |                |
|---|----------------|
| <b>7. Pruebas</b>   | <b>92</b>      |
| 7.1. Pruebas sobre los datos . . . . .                        | 92             |
| 7.2. Pruebas sobre la interfaz . . . . .                      | 92             |
| 7.3. Especificación del diseño de las pruebas . . . . .       | 93             |
| 7.4. Especificación de los procedimientos de prueba . . . . . | 94             |
| 7.5. Documentación de la ejecución de pruebas . . . . .       | 94             |
| <br><b>8. Conclusiones</b>                                    | <br><b>97</b>  |
| 8.1. Introducción . . . . .                                   | 97             |
| 8.2. Conocimientos adquiridos . . . . .                       | 97             |
| 8.3. Futuro del proyecto . . . . .                            | 98             |
| <br><b>A. Manual de instalación</b>                           | <br><b>104</b> |
| A.1. Aplicaciones de terceros . . . . .                       | 104            |
| <br><b>B. Manual del usuario</b>                              | <br><b>107</b> |
| B.1. Pantalla de Login . . . . .                              | 107            |
| B.2. Pantalla Registro . . . . .                              | 108            |
| B.3. Pantalla control reproducción . . . . .                  | 110            |
| B.4. Pantalla lista de canciones . . . . .                    | 111            |
| B.5. Pantalla Listas de Reproducción . . . . .                | 112            |
| B.6. Pantalla nueva lista . . . . .                           | 114            |
| B.7. Pantalla modificar contraseña . . . . .                  | 115            |
| B.8. Pantalla recuperar contraseña . . . . .                  | 117            |



## Índice de figuras

|   |    |
|---|----|
| 4.1. Modelo conceptual de datos . . . . .                             | 32 |
| 4.2. Caso de uso. Página inicial. . . . .                             | 33 |
| 4.3. Diagrama de secuencia: identificación. . . . .                   | 35 |
| 4.4. Diagrama de secuencia: Olvido contraseña . . . . .               | 36 |
| 4.5. Diagrama de secuencia. Registro usuario . . . . .                | 38 |
| 4.6. Casos de uso. Menú reproductor . . . . .                         | 39 |
| 4.7. Diagrama de secuencia. Seleccionar Lista de reproducción . . .   | 40 |
| 4.8. Diagrama de secuencia. Gestión de reproducción . . . . .         | 41 |
| 4.9. Caso de uso. Cerrar Sesión. . . . .                              | 43 |
| 4.10. Diagrama de secuencia. Cambio contraseña. . . . .               | 45 |
| 4.11. Caso de uso. Modificar lista de reproducción. . . . .           | 45 |
| 4.12. Diagrama de secuencia. Añadir canciones. . . . .                | 46 |
| 4.13. Diagrama de secuencia. Eliminar canciones. . . . .              | 48 |
| 4.14. Diagrama de secuencia. Mostrar canciones Lista. . . . .         | 49 |
| 4.15. Diagrama de secuencia. Agrupar canciones. . . . .               | 50 |
| 4.16. Caso de uso. Modificar listas de reproducción. . . . .          | 51 |
| 4.17. Diagrama de secuencia. Añadir lista de reproducción. . . . .    | 52 |
| 4.18. Diagrama de secuencia. Seleccionar lista de reproducción. . . . | 53 |
| 5.1. Modelo relacional de la base de datos. . . . .                   | 56 |
| 5.2. Base de datos del servidor. . . . .                              | 58 |

|  |     |
|--|-----|
| 5.3. Estructuras de datos. . . . .                                   | 60  |
| 5.4. Diagrama de secuencia. Identificación. . . . .                  | 65  |
| 5.5. Diagrama de secuencia. Menú de opciones. . . . .                | 66  |
| 5.6. Diagrama de secuencia. Pantalla listas de reproducción. . . . . | 67  |
| 5.7. Diagrama de secuencia. Nueva lista de reproducción. . . . .     | 68  |
| 5.8. Capa de presentación. Pantalla Login. . . . .                   | 70  |
| 5.9. Capa de presentación. Pantalla de registro. . . . .             | 71  |
| 5.10. Capa de presentación. Pantalla recuperar contraseña. . . . .   | 72  |
| 5.11. Capa de presentación. Pantalla modificar contraseña. . . . .   | 73  |
| 5.12. Capa de presentación. Control de Reproducción. . . . .         | 74  |
| 5.13. Capa de presentación. Lista de Canciones. . . . .              | 76  |
| 5.14. Capa de presentación. Listas de reproducción. . . . .          | 77  |
| 5.15. Capa de presentación. Opciones lista. . . . .                  | 78  |
| 5.16. Capa de presentación. Nueva lista. . . . .                     | 79  |
| B.1. Manual del usuario. Pantalla de inicio. . . . .                 | 108 |
| B.2. Manual de Usuario. Registro Usuario. . . . .                    | 109 |
| B.3. Manual del usuario. Pantalla control de reproducción. . . . .   | 111 |
| B.4. Manual del usuario. Pantalla lista de canciones. . . . .        | 112 |
| B.5. Manual del usuario. Pantalla Listas de reproducción. . . . .    | 114 |
| B.6. Manual de usuario. Nueva lista. . . . .                         | 115 |
| B.7. Manual de usuario. Menú control reproducción. . . . .           | 116 |
| B.8. Manual de usuario. Cambiar contraseña. . . . .                  | 117 |
| B.9. Manual de usuario. Recuperar Contraseña. . . . .                | 118 |



## 1. Introducción

Las aplicaciones que crean una lista personalizada de reproducción se basan en sugerir canciones de su propia base de datos pero no pueden funcionar sin conexión con el consecuente consumo de datos o hay que introducir parámetros adicionales como el estado de ánimo o parámetros adicionales.

La aplicación nace con el objetivo de crear un reproductor de música que sea capaz de realizar una mezcla personalizada de canciones y que se realice de forma transparente al usuario, es decir que no se aumente la complejidad de uso de la aplicación.

### 1.1. Objetivos

El objetivo del proyecto es la creación de un reproductor de música para el sistema operativo Android que tenga un modo de reproducción SmartRand que realice una mezcla de canciones cuya misión es minimizar las veces que tenemos que pasar a la canción siguiente mientras estamos reproduciendo las canciones de la lista de reproducción en modo aleatorio.

La mezcla de canciones automática se puede realizar sobre todo el conjunto de las canciones presentes en el dispositivo móvil o en un subconjunto de este mediante la realización de listas de reproducción. La lista de reproducción puede ser modificada añadiendo o eliminando Álbumes, Artistas o canciones arrastrando los elementos hacia un lado de la pantalla de forma que podemos crear una lista de reproducción fácilmente y sin dificultad.

Para llevar un control de los usuarios, al iniciar la aplicación se conecta con una base de datos en un servidor en la cual se almacenan todos los usuarios registrados. Una vez que el usuario se registra en la base de datos se crea el usuario en una base de datos local junto con la sesión de reproducción que es la que almacena los gustos de una canción en la base de datos y una lista de reproducción predeterminada.

Además desde el menú principal desde donde se controla la reproducción se puede modificar la lista de reproducción actual, avanzar o retroceder desde

la canción actual de forma táctil y seleccionar el momento actual de reproducción dentro de la canción.

Desde la pantalla para modificar la lista de reproducción actual se puede ir a la vista para visualizar las listas de reproducción mediante la cual podemos ver todas las listas de reproducción presentes en el dispositivo.

Dentro de la cual podemos añadir listas de reproducción y asignarlas a una sesión de reproducción además de seleccionar una lista de reproducción como lista actual, quedando seleccionada para la reproducción y la modificación de las canciones.

## 1.2. Motivación

Existen diversos motivos por los que he decidido realizar este proyecto en concreto.

El primero es debido al auge de los dispositivos móviles los reproductores de música han ido perdiendo peso frente a los smartphones en la reproducción de música.

El segundo es la carencia desde mi punto de vista de un reproductor simple y gratuito y enfocado a la música alojada en el propio teléfono pero que sea proactivo mejorando la selección de canciones a medida que el usuario hace uso de la aplicación.

La tercera es de diseñar el reproductor que realice una selección de canciones de forma transparente al usuario, de forma que no aumenta la complejidad de uso de la aplicación del usuario medio.

El cuarto es la de veces que con otros reproductores hay la necesidad de saltar una canción todas las veces que se escucha una canción, sin la posibilidad de eliminar la canción o de aprendizaje en las sucesivas reproducciones.

El fin último es simular la apariencia de un reproductor estándar pero que tras sucesivas reproducciones de la lista de reproducción no haya la necesidad de pasar las canciones ya que el reproductor se adapte al uso del usuario.

Existen otro tipo de reproductores que ofrecen sugerencias de canciones pero en estos casos es el usuario el que se adapta al sistema ya sea porque te mandan que canciones quieres escuchar o porque hay que introducir **parámetros adicionales**.

### 1.2.1. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

A continuación se muestran un conjunto de términos y expresiones utilizadas en el proyecto y se aparecerán a lo largo de este:

- **Android:** Es el sistema operativa de Google destinado a dispositivos móviles, smartphones y tablets. Lanzado en septiembre del año 2008 ha revolucionado el mundo de los smarphones debido a la cantidad de dispositivos que hacen uso de Android.
- **Apache:** Es un software libre. Apache es un servidor HTTP de código abierto para plataformas Linux, Windows y Mac OS X entre otros, desarrollado por Apache Software Foundation.
- **MySQL:** Es un sistema de base de datos relacional, multihilo y multi-usuario.
- **PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico y que permite la programación orientada a objetos.
- **Smartphone:** Es el término empleado para denominar a un teléfono móvil con más funcionalidades que un teléfono móvil común.
- **SDK:** Conjunto de herramientas que permiten al programador desarrollar en un sistema en concreto.
- **IDE:** Proporciona un conjunto de herramientas útiles para programar.
- **Eclipse:** IDE que hemos utilizado para desarrollar la aplicación en Android.



### 1.3. Estructura de la memoria

- Introducción: se realiza una breve descripción junto con los objetivos y alcance del proyecto.
- Planificación: análisis del tiempo estimado en la realización de todo el proceso que ha abarcado este proyecto.
- Especificación de requisitos: consta de una visión del sistema realizando una especificación del conjunto de tecnologías utilizadas en el proyecto y sus requisitos.
- Análisis: realización del análisis del sistema en el cual se definirán los requisitos funcionales, diagramas de uso, secuencia y contrato de operaciones.
- Diseño: diseño de la base de datos, clases utilizadas junto con los diagramas de secuencia.
- Implementación: descripción de los aspectos más importantes en la implementación de la aplicación.
- Pruebas: descripción de las pruebas realizadas al sistema para asegurar su corrección.
- Conclusiones: en él se recogen las conclusiones finales así como la valoración personal del proyecto.
- Apéndices: incluye el manual de instalación y puesta en funcionamiento y el manual de usuario.
- Bibliografía: el conjunto de los libros y referencias consultados durante el desarrollo del proyecto.

### 1.4. PLANTEAMIENTO TÉCNICO

En cuanto al planteamiento técnico, la idea principal es realizar una aplicación sencilla intuitiva y lo más cómoda posible ya que el hecho de pulsar en cada una de las canciones que se desea añadir a la lista de reproducción podría

ser algo tedioso, por lo tanto se decidió realizar una división de las canciones por artista, álbum o genero y poder añadirlos deslizando el elemento con un sencillo gesto.

La lista de reproducción se basa en restricciones. Al inicio todas las canciones están en la tarjeta SD y se visualizan al entrar en la biblioteca musical.

Las canciones añadidas se pueden visualizar pulsando sobre el icono música en la biblioteca musical. De igual forma podemos añadir todas las canciones a la lista de reproducción y seleccionar que canciones no queremos que aparezcan en la lista de reproducción

En cuanto a los requisitos necesarios para el usuario tan solo es necesario disponer de un dispositivo móvil Android, instalar la aplicación y disponer de conexión a Internet. Por otro lado, para el almacenamiento de los datos será necesario un servidor, independiente del sistema operativo que emplee, en el que almacenar la base de datos. También debe ser capaz de interpretar PHP.

Para el desarrollo del proyecto se ha empleado un equipo con el sistema operativo Windows 8, trabajando con el entorno de desarrollo Eclipse y el kit de desarrollo de software, o SDK, de Android. Para la base de datos se ha empleado un hosting del proveedor hostinger ([www.hostinger.es](http://www.hostinger.es))



## 2. Planificación

### 2.1. Introducción

Para el desarrollo del proyecto se ha establecido un periodo de tiempo total de 10 meses. En la siguiente sección se encuentra descrito el conjunto de periodos de tiempo que se ha empleado durante la realización de las diferentes etapas de las que se compone el proyecto.

De igual forma el desglose del tiempo empleado se ha reflejado en el diagrama de Gantt incluido a continuación de la descripción en el cual se puede apreciar de forma gráfica la planificación empleada.

Para la implementación de las diferentes funcionalidades de conforman el sistema, hemos seguido el desarrollo evolutivo, de forma que el producto final es el resultado de sucesivas iteraciones, cada una de las cuales añade funcionalidad al producto.

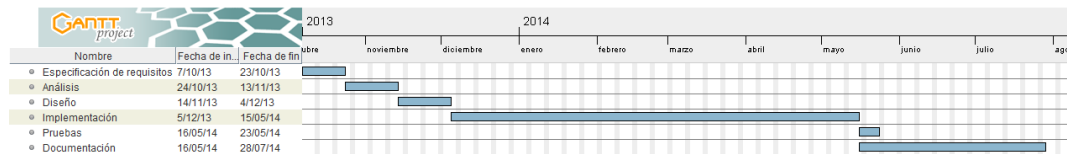
Estas iteraciones son realizadas hasta llegar a un prototipo suficientemente depurado y que tenga las características suficientes para un producto final

### 2.2. Desglose de la planificación

- Desarrollo:

- Especificación de requisitos(7/10/2013 - 23/10/2013): estudio de los requisitos del sistema.
- Análisis (24/10/2013- 13/11/2013: Elaboración del diagrama de clases conceptuales y modelo del sistema.
- Diseño (14/11/2013 - 4/12/2013): Se diseña la aplicación de acuerdo al análisis y a los requisitos especificados.
- Implementación (5/12/2013 -15/05/2014): Se implementan todos los requisitos especificados en el sistema.
- Pruebas (16/05/2014 - 23/05/2014): Probar toda la funcionalidad del sistema corrigiendo los errores encontrados.

Documentación: (23/05/2014 - 28/07/2014): Realización de la memoria para documentar todo el proceso.



### 2.3. Costes

Para la realización del calculo de los costes de personal se obtiene una media del sueldo aproximado Programador y a continuación se indica la correspondencia para el total del tiempo en el que se ha desarrollado el proyecto. Indicando un coste medio de 21.000€ anuales por programador realizamos la tabla correspondiente.

| Cantidad | Descripción            | Coste unitario | Coste total |
|----------|------------------------|----------------|-------------|
| 1        | Analista/Desarrollador | 21.000€ / año  | 17.000      |



## 3. Especificación de requisitos

A continuación se detallan los requisitos de la aplicación.

### 3.1. Introducción

En primer lugar se definen determinadas características del proyecto, tales como el ámbito y una serie de definiciones.

#### 3.1.1. Propósito

La finalidad es especificar los requerimientos que es necesario que cumpla la aplicación. El objetivo es detallar las funcionalidades y requisitos para la correcta implementación posterior.

#### 3.1.2. Ámbito

La aplicación consiste en la elaboración de un reproductor de música para el sistema operativo Android, el cual almacenará datos estadísticos relativos a la reproducción de los archivos de audio para poder ofrecer mezcla de canciones personalizadas. Además dispone de un sistema alojado en un servidor que almacenará información de los usuarios.

El reproductor permitirá la creación de listas de reproducción inteligentes mediante la creación de restricciones, evitando la aparición de canciones pertenecientes a un determinado artista, álbum, o canciones en particular.

Seleccionando la función de mezcla inteligente el algoritmo calculará una secuencia de canciones que se irá creando dinámicamente.

El objetivo de la aplicación es crear una mezcla de canciones con las restricciones que seleccionemos respecto al artista, el álbum y las canciones la cual minimice la probabilidad de que el usuario detenga la reproducción de la canción actualmente en reproducción para continuar con la siguiente.

## 3.2. Descripción general

En el siguiente apartado se procederá a la especificación del proyecto, en la cual se enumerarán los requisitos del sistema y su funcionalidad.

### 3.2.1. Perspectiva del producto

El desarrollo de la aplicación esta dividida en dos partes, el desarrollo del reproductor de música en el sistema operativo para móviles Android android y por otro lado, la parte del servidor que maneja una base de datos con los usuarios y las diferentes funciones necesarios para la gestión de los usuarios y los datos asociados a estos.

Para la aplicación Android se emplea el SDK eclipse junto con ADT (Android developers tools) utilizando el lenguaje de programación Java. En cuanto al lado del servidor, se utilizará el servidor Apache y una base de datos MySQL.

Para la comunicación de la aplicación con el servidor se empleará PHP debido a su fácil integración.

La aplicación se debe de mostrar igual en diferentes dispositivos y tiene que ser compatible con el mayor número de versiones de android.

## 3.3. Catálogo de requisitos

A continuación se especifican los requisitos de la aplicación.

### 3.3.1. Requisitos funcionales

Las requisitos funcionales de la aplicación son las siguientes:

- Requisitos funcionales de identificación:



- El usuario tendrá acceso al iniciando sesión como usuario registrado. Para que un usuario pueda iniciar sesión será necesario que disponga de una dirección de correo electrónico activada y una contraseña asociada al correo. En el caso que no disponga de la contraseña será necesario recuperarla a través de una funcionalidad diseñada para tal efecto.
- Requisitos funcionales de registro:
  - El usuario podrá registrarse en el sistema introduciendo una dirección de correo electrónico activada, una contraseña, un identificador de usuario y un nombre y contraseña.
- Requisitos funcionales de recuperar contraseña:
  - El usuario será capaz de recuperar el acceso a su cuenta mediante el envío de una nueva contraseña al correo electrónico asociada a la cuenta.
- Requisitos funcionales del control de reproducción:
  - El usuario podrá reproducir la canción siguiente a la actual, reproducir la canción anterior a la actual y pausar la reproducción.
  - El usuario tendrá la posibilidad de introducir sus gustos sobre la canción actual además de seleccionar el modo de reproducción, ya sea mezcla de canciones o reproducción secuencial.
  - El sistema mostrará las listas de reproducción de la sesión actual para que el usuario tenga la posibilidad de modificar la lista de reproducción desde el menú de control de reproducción.
  - Seleccionando las opciones el usuario podrá salir de la sesión actual además de cambiar la contraseña asociada a la cuenta.
- Requisitos funcionales de modificar la lista de reproducción:
  - El usuario podrá visualizar las canciones presentes en la lista de reproducción además del conjunto de canciones presentes en el dispositivo y que no se han añadido a la lista de reproducción.

- El usuario podrá organizar las canciones mostradas por Artistas, Álbumes o nombre de las canciones.
- El usuario tendrá la capacidad de añadir todas las canciones a la lista de reproducción además de eliminar todas las canciones de la lista.
- Requisitos funcionales de modificación de la lista de reproducción:
  - El usuario podrá visualizar las listas de reproducción.
  - El usuario será capaz de seleccionar una lista de reproducción para la reproducción actual
  - El usuario tendrá la capacidad de eliminar una lista de reproducción.
- Requisitos funcionales para agregar una lista de reproducción:
  - El usuario introducirá el nombre de la lista de reproducción y el nombre de la sesión a la que pertenecerá la lista de reproducción. La lista de reproducción se asignará a una sesión de reproducción por defecto en el caso de que no se introduzca ninguna lista de reproducción.
- Requisitos de rendimiento:
  - El sistema actualizará las canciones presentes en el dispositivo de forma dinámica cuando el usuario avance una canción. Esto tiene una mayor importancia en el caso que se seleccione la mezcla de canciones ya que se actualizará indefinidamente la lista de canciones en relación a las canciones que vaya devolviendo el algoritmo y que se irán agregando al conjunto de canciones pendientes por reproducir.

### 3.4. Restricciones generales

Para el correcto funcionamiento de la aplicación hay que mencionar la necesidad de tener acceso a una base de datos con una estructura establecida. Si la

estructura de la base de datos no fuera correcta, la aplicación no funcionaría correctamente o dejaría de funcionar.

Al inicio por primera vez la aplicación es necesario tener acceso a internet para el registro del usuario pero las veces siguientes no es necesario el acceso a internet pudiendo reproducir las canciones y realizar mezclas personalizadas de estas en modo offline.

#### 3.4.1. Requisitos de la interfaz

Con respecto a los requisitos de la interfaz podemos diferenciar tres tipos: interfaz de usuario, interfaz software e interfaz hardware.

En cuanto a la interfaz de usuario, el principal objetivo al desarrollarla es conseguir una interfaz sencilla intuitiva y dinámica. Puesto que la aplicación es un reproductor de audio existe un patrón básico que comparten la mayor parte de los reproductores, que consiste en la posibilidad de reproducir y pausar la reproducción, reproducir la canción siguiente a la actual y reproducir la canción anterior a la actual y la creación de listas de reproducción con la que poder escoger las canciones que queremos escuchar.

Además de los requisitos básicos, es necesario añadir a la interfaz los elementos necesarios para la correcta implementación del algoritmo que realiza la mezcla de canciones como son la posibilidad de crear sesiones de reproducción con las que podemos diferenciar el aprendizaje en diferentes circunstancias pudiendo de esta forma optimizar la reproducción de canciones en diferentes entornos y contextos.

Para ello añadimos a la posibilidad de crear listas de reproducción, la posibilidad de modificar la lista de reproducción a la seleccionada por defecto.

También es necesaria la interfaz para el registro de la aplicación la cual solicitará elementos de identificación como son el nombre, correo, nombre de usuario y clave.

Para la interfaz software, el proyecto se desarrolla haciendo uso del sistema operativo Windows 7, empleando el entorno de desarrollo Eclipse con el plugin ADT (Android developers Tools). Java es el lenguaje principal puesto

que Android hace uso de este lenguaje, pero también se emplea PHP. Por último para el sistema de gestión de base de datos se emplea MySQL.

Haciendo referencia al hardware, se ha imprescindible el uso de un smartph-  
hone con el sistema operativo Android y conexión a Internet, además de un  
servidor que tenga integrado PHP Y MySQL.

#### **3.4.2. Requisitos de eficiencia**

En la aplicación Android sólo puede estar registrado un usuario simultánea-  
mente debiendo cerrar el usuario actual para cambiar de usuario.

Debido a que la aplicación realiza una selección de la música propia del  
usuario, se va mejorando la personalización de la mezcla musical conforme  
se incrementa el número de reproducciones.

Los beneficios de este sistema es que realiza una personalización de los gustos  
del usuario basado en la experiencia y con ello incrementa la personalización  
de la mezcla musical.

Por otro lado al desinstalar la aplicación o al modificar las canciones en el  
reproductor vuelve a necesitar de la experiencia proporcionada por la re-  
producción para crear una correcta mezcla de canciones. Debido a que la  
aplicación se ejecuta en un dispositivo móvil la eficiencia de la aplicación  
dependerá de los requisitos del propio terminal, a pesar de esto, cualquier  
smartphone con Android debería de ser capaz de ejecutar la aplicación de  
forma eficiente.

#### **3.4.3. Otros requisitos**

La aplicación hace uso de una base de datos en el teléfono que almacena  
todos los datos relativos a la reproducción de las canciones y de las listas de  
reproducción.

También el servidor utiliza otra base de datos que almacena los datos perte-  
necientes a los usuarios.

Para el acceso a la base de datos alojada en el servidor se implementan funciones en PHP debido a la compatibilidad.

Uno de los aspectos importantes es la compatibilidad de la aplicación con las versiones de android, por este motivo se establece que la aplicación funcione en dispositivos con Android 3.0 o superior lo que reúne la gran mayoría de los teléfonos disponibles con Android en el mercado.



## 4. Análisis

En el siguiente apartado se procede a detallar el proceso de análisis realizado para la implementación de la aplicación, haciendo uso de la metodología UML.

### 4.1. Modelo conceptual de datos

En la siguiente figura podemos ver el diagrama que hace referencia al modelo conceptual de datos el cual utilizaremos durante la implementación del sistema.

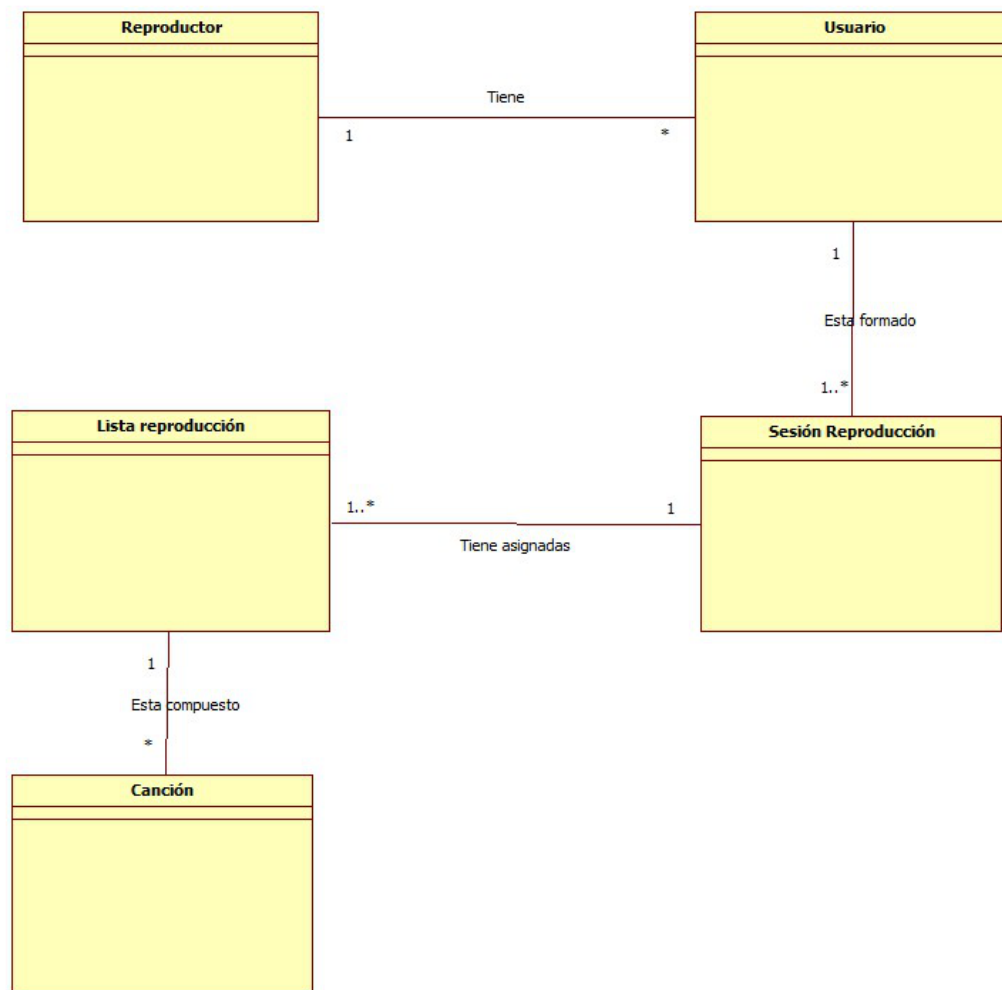


Figura 4.1: Modelo conceptual de datos

## 4.2. Modelo de casos de Uso

En la siguiente sección describiremos el conjunto de casos de usos que pertenecen a la aplicación, así como los diagramas de casos de uso y los de secuencia.



Para el análisis se hace uso de la metodología UML. Lo primero es la realización de los casos de uso. En este caso, como solo existe un usuario todas las acciones son realizadas por el mismo usuario.

#### 4.2.1. Casos de uso referentes a la página inicial:



Figura 4.2: Caso de uso. Página inicial.

- Caso de uso: Identificación de usuario.
  - Descripción: Se identifica el usuario en el sistema.
  - Actor principal: El usuario.
  - Precondición: El usuario esta registrado en el sistema y no tiene una sesión activa.
  - Postcondición: El usuario ha iniciado sesión en el sistema y podrá utilizar la funcionalidad de la aplicación
- Escenario principal:
  1. El usuario introduce su correo.
  2. El usuario introduce su contraseña
  3. El usuario pulsa iniciar sesión.

4. El sistema comprueba la dirección de correo electrónico y contraseña introducida por el usuario.
  5. Se crea una sesión para el usuario con todos los datos del perfil del usuario.
  6. Se dirige al usuario a la página inicial de la aplicación.
- Escenario secundario o de error:
    - 4b - La contraseña introducida por el usuario no guarda el formato correcto debido a su longitud.
    - 5a - La dirección de correo electrónico no se encuentra registrada en el sistema.
    - 5b - La contraseña no corresponde a la asignada a la dirección de correo electrónico.
- Se muestra un mensaje de error indicando que el correo y la contraseña no coinciden.

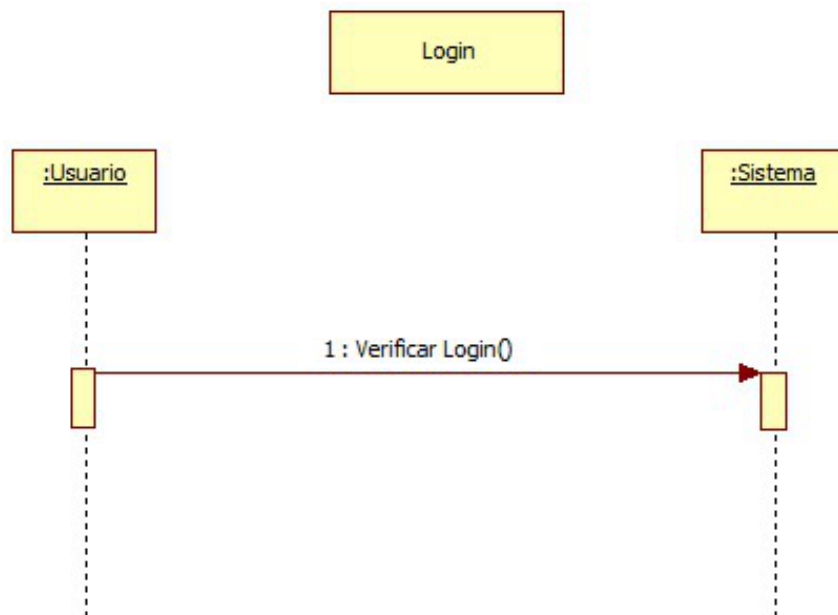


Figura 4.3: Diagrama de secuencia: identificación.

- Contrato de operación:
  - Responsabilidades: Comprueba que los datos introducidos por el usuario son correctos, realiza la identificación y permite el acceso al sistema mostrando el mensaje de error si la contraseña o el correo no son correctos.
  - Referencias cruzadas: Caso de uso identificación”.
  - Precondición:
    - El usuario se ha registrado previamente en el sistema.
    - La dirección introducida de correo electrónico tiene el formato correcto.
    - La contraseña introducida por el usuario es correcta para el correo introducido.
  - Postcondición:
    - El usuario esta registrado en el sistema y tiene asignada una sesión con todos sus datos.
- Caso de uso: Olvido contraseña
  - Descripción: El usuario no recuerda la contraseña asignada a su correo y desea entrar en el sistema.
  - Actor principal: El usuario.
  - Precondición: El usuario se encuentra dado de alta en el sistema.
  - Postcondición: El sistema envía al correo una contraseña generada aleatoriamente y la asigna al usuario.
  - Escenario principal:
    1. El usuario pulsa en la sección Olvido contraseña.
    2. Se carga la sección con el formulario para introducir el correo del usuario.
    3. El usuario introduce su dirección de correo.

4. El sistema envía un correo con la contraseña generada aleatoriamente al usuario.

- Escenario secundario o de error:
  - 4a - El usuario introduce una dirección de correo que no se encuentra en el sistema.
  - 4b - Se produce un error y no se realiza el cambio correctamente.

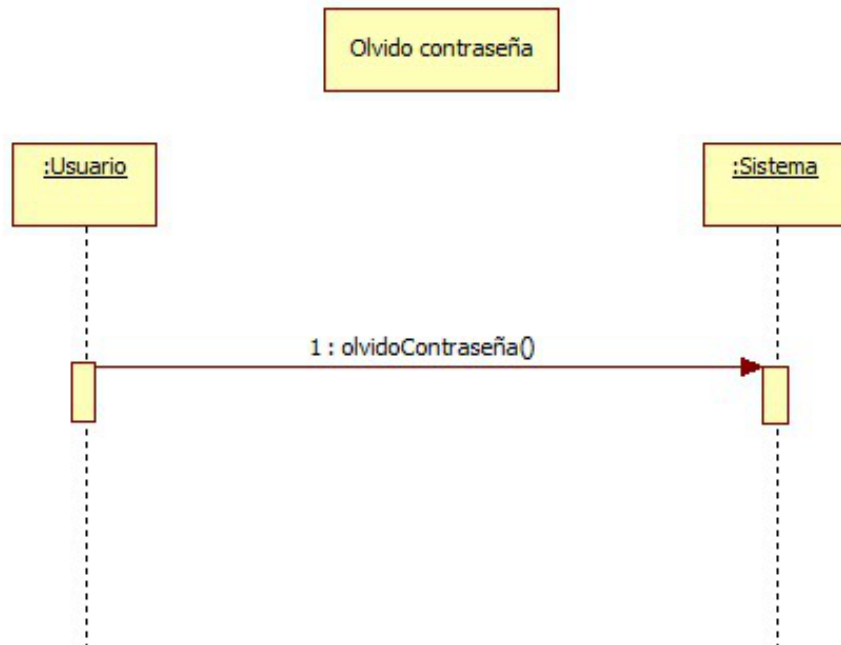


Figura 4.4: Diagrama de secuencia: Olvido contraseña

- Contrato de operación:
  - Responsabilidades: Restablecer la contraseña del usuario por una aleatoria y enviarla al usuario por correo.
  - Referencias cruzadas: Caso de uso Olvido contraseña.

- Precondición:
    - El usuario ha iniciado sesión en el sistema.
    - La dirección introducida de correo electrónico tiene el formato correcto.
  - Postcondición:
    - El sistema sustituirá la contraseña por la generada por el sistema de forma aleatoria.
- Caso de uso: Registro usuario:
- Descripción: Se realiza el registro de un nuevo usuario en el sistema.
  - Actor principal: El usuario.
  - Precondición: La dirección de correo electrónico no está en el sistema.
  - Postcondición: El usuario esta dado de alta y el usuario y contraseña están almacenados en la base de datos.
  - Escenario principal:
    1. El usuario pulsa en la sección Registrarse.
    2. El sistema muestra por pantalla el formulario de registro con su dirección de correo y su contraseña.
    3. El usuario rellena los campos del formulario que es mostrado por pantalla y selecciona registrarse.
    4. El sistema comprobación que la dirección de correo no se encuentra actualmente en el sistema.
    5. El sistema almacena los datos pertenecientes al usuario en la base de datos y envía un correo electrónico al usuario y comprueba que el correo tiene un formato válido.
  - Escenario secundario o de error:
    - 3a - El correo electrónico que el usuario ha introducido no tiene el formato correcto.
    - 3b - La contraseña que ha introducido el usuario no tiene la longitud requerida.

4a - La dirección de correo que ha introducido el usuario ya se encuentra almacenado en el sistema.

\* Se muestra el mensaje de error indicando el motivo al usuario.

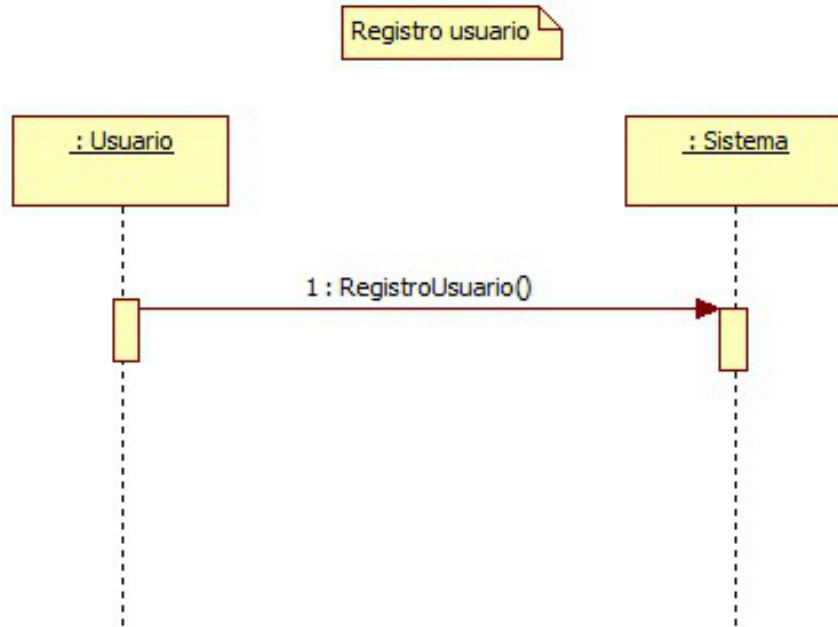


Figura 4.5: Diagrama de secuencia. Registro usuario

- Contrato de operación:
  - Responsabilidades: Dar de alta al nuevo usuario en el sistema.
  - Referencias cruzadas: Caso de uso «Alta usuario».
- Precondición:
  - No se ha dado de alta ningún usuario previamente con la dirección de correo introducida.
  - La contraseña tiene la longitud requerida.
- Postcondición:
  - El usuario esta dado de alta en el sistema

#### 4.2.2. Casos de usos referentes a menú reproducción

A continuación se muestran las especificaciones de los casos de uso correspondientes al menú reproducción.

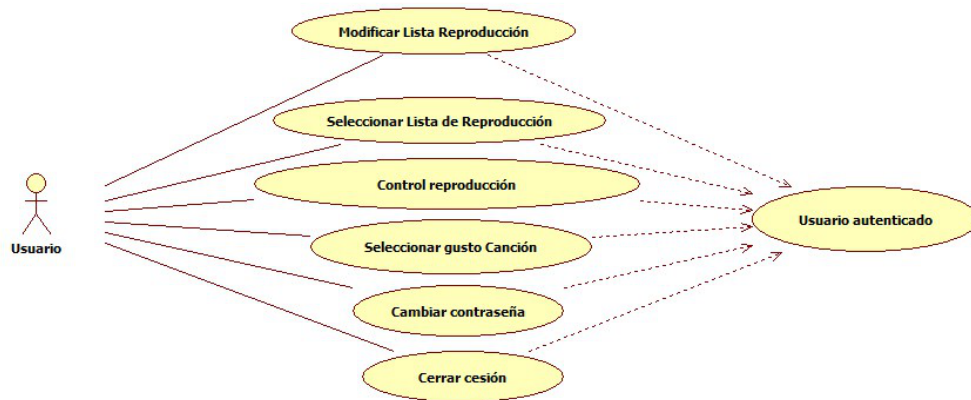


Figura 4.6: Casos de uso. Menú reproductor

- Caso de uso: Seleccionar Lista de Reproducción
  - Descripción: El usuario seleccionará la lista de reproducción del conjunto de listas de reproducción que mostrará el sistema pertenecientes a la sesión actual.
  - Actor principal: El usuario.
  - Precondición: El usuario tiene una sesión activa.
  - Postcondición: Se muestra el listado con las listas de reproducción de la sesión actual.
  - Escenario principal:
    1. El usuario selecciona la lista de reproducción del conjunto de listas que mostrará el sistema.
    2. El sistema recibe la petición y carga el listado con las listas de reproducción.

3. El usuario selecciona la lista de reproducción.
4. El sistema pausa la reproducción carga la lista de reproducción seleccionada.

■ Escenario secundario

3a - El usuario selecciona la misma lista de reproducción que se esta reproduciendo y no realiza ninguna acción.



Figura 4.7: Diagrama de secuencia. Seleccionar Lista de reproducción

■ Contrato de la operación:

- Responsabilidades: Permitir al usuario modificar la lista de reproducción actual.
- Precondición:
  - El usuario ha iniciado sesión en el sistema.
- Postcondición:



- El usuario podrá elegir la lista de reproducción actual.
- Caso de uso: Control reproducción
  - Descripción: El usuario podrá controlar la reproducción actual, iniciar/pausar, avanzar a la canción siguiente de la reproducción, reproducir la canción anterior a la actual, modificar el segundo en el que se esta reproduciendo la canción y controlar si el modo de reproducción es secuencial o en modo automático.
  - Actor principal: El usuario.
  - Precondición: El usuario tiene una sesión activa.
  - Postcondición: Se puede controlar la reproducción actual.
  - Escenario principal:
    1. El usuario selecciona la zona correspondiente al control de la reproducción.
    2. La reproducción se modifica en función de la opción seleccionada.

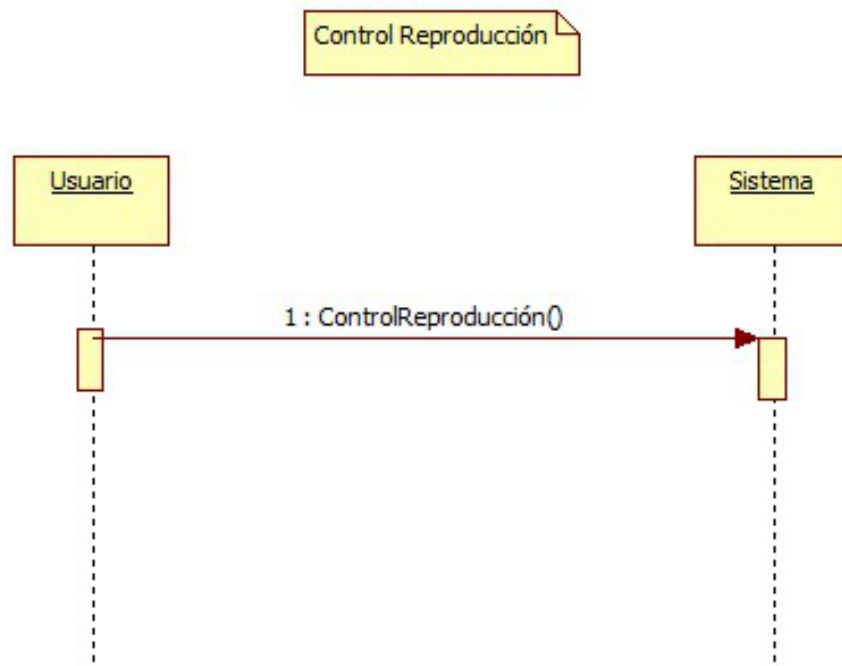


Figura 4.8: Diagrama de secuencia. Gestión de reproducción

- Contrato de operación:
  - Responsabilidades: Permitir al usuario controlar la reproducción actual.
  - Precondición:
    - El usuario está autenticado en el sistema y tiene una sesión activa.
  - Postcondición:
    - El usuario podrá controlar la reproducción de las canciones.
- Caso de uso: Seleccionar gusto de la canción
  - Descripción: El usuario va a proceder a seleccionar el gusto de la canción actual, ya sea positivo, negativo o neutro.
  - Actor principal: El usuario.
  - Precondición: El usuario esta la la pantalla menú reproducción.
  - Postcondición: El usuario selecciona el gusto y éste se almacena en la base de datos.
  - Escenario principal:
    1. El usuario selecciona si la canción actual le gusta, no le gusta, o elimina el gusto a la canción actual.
- Caso de uso: Cerrar sesión
  - Descripción: El usuario desea modificar el usuario actual de la aplicación.
  - Actor principal: El usuario.
  - Precondición: El usuario tiene una sesión activa.
  - Postcondición: El usuario cierra la sesión que tiene abierta y se abre la vista correspondiente a la página inicial.
  - Escenario principal:
    1. El usuario selecciona cerrar sesión.
    2. El sistema recibe la petición de cerrar la sesión y cierra la sesión del usuario eliminando al usuario de la sesión actual.

Se redirige al usuario a la página inicial de la aplicación.

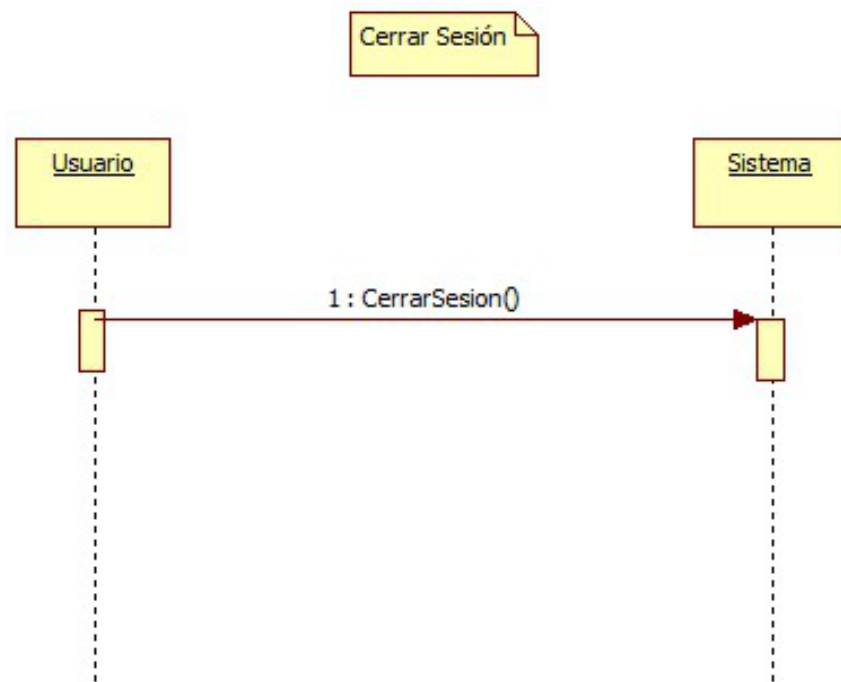


Figura 4.9: Caso de uso. Cerrar Sesión.

- Contrato de la operación:
  - Responsabilidades: Cierra la sesión del usuario y muestra la vista de la pantalla de inicio
  - Referencias cruzadas: Caso de uso «Cerrar Sesión».
  - Precondición:
    - El usuario tiene una sesión activa y ha iniciado sesión previamente en el sistema.
  - Postcondición:

- Se cerrará la sesión actual, y se muestra la pantalla de inicio en la que el usuario puede volver a registrarse si fuera necesario.
- Caso de uso: Cambiar contraseña
  - Descripción: El usuario accederá a la sección cambiar contraseña donde podrá modificar su contraseña actual por otra.
  - Actor principal: El usuario.
  - Precondición: El usuario tiene esta autenticado en el sistema.
  - Postcondición: Se carga la sección cambio de usuario.
  - Escenario principal:
    1. El usuario pulsa en cambio de usuario.
    2. El sistema recibe la petición para la modificación de la contraseña y muestra la pantalla correspondiente.
    3. El usuario introduce la contraseña nueva.
- Escenario secundario o de error:
  - 3a - El usuario no introduce ninguna contraseña y sale del caso de uso.

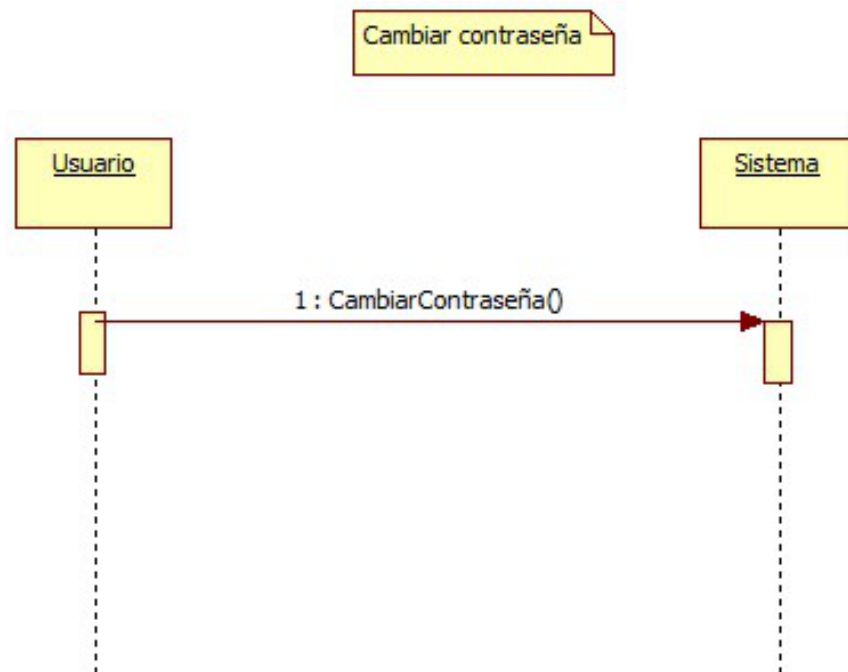


Figura 4.10: Diagrama de secuencia. Cambio contraseña.

- Contrato de la operación:
  - Responsabilidades: Permitir al usuario acceder a la sección cambiar contraseña.
  - Precondición:
    - El usuario esta autenticado en el sistema y tiene una sesión activa.
  - Postcondición:
    - El usuario modificará la contraseña actual.

#### 4.2.3. Casos de uso referentes a la modificación de la lista de reproducción

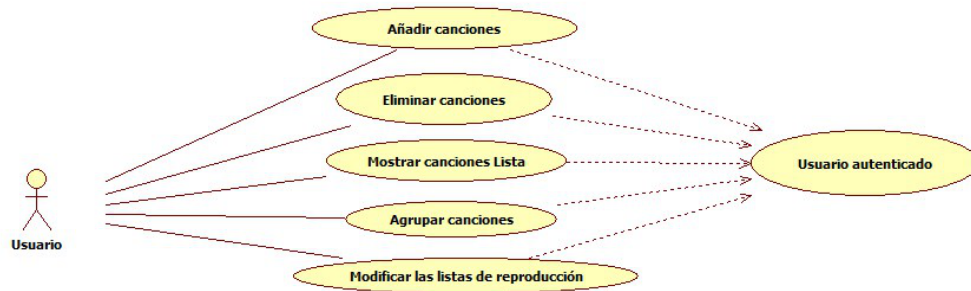


Figura 4.11: Caso de uso. Modificar lista de reproducción.

- Caso de uso: Añadir canciones
  - Actor principal: El usuario.

- Precondición: El usuario ha iniciado sesión en el sistema y está visualizando las canciones que se encuentran en memoria del dispositivo.
- Postcondición: el sistema agregará el conjunto de canciones seleccionadas a la lista de reproducción
- Escenario principal:
  1. El usuario esta visualizando la lista de canciones que no se encuentran en la lista de reproducción.
  2. El usuario selecciona la canción o canciones a añadir a la lista de reproducción.
  3. El sistema añadirá la canción o canciones a la lista de reproducción.
- Escenario secundario o de error:
  - 2a - No se ha seleccionado ninguna canción y no se ha añadido ninguna canción a la lista de reproducción.

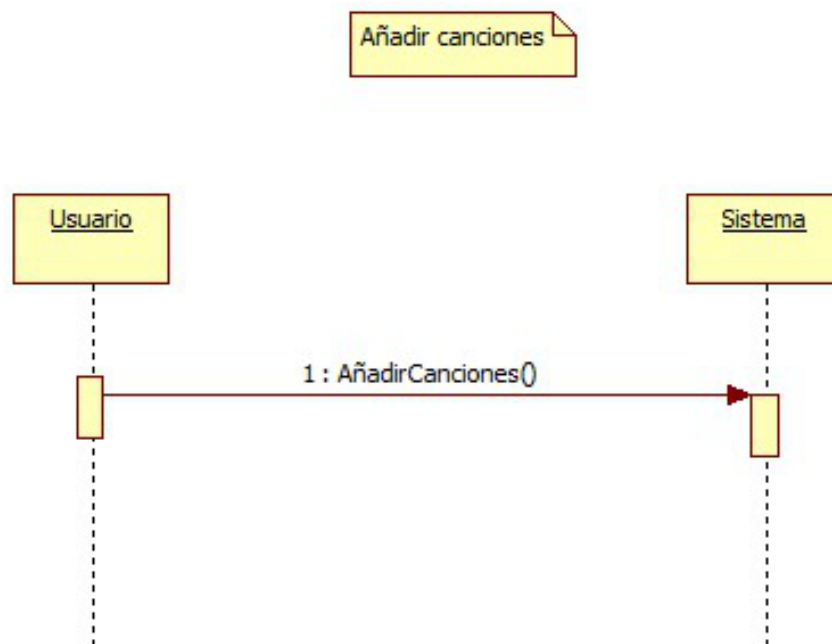


Figura 4.12: Diagrama de secuencia. Añadir canciones.

- 
- Caso de uso: Eliminar canciones
  - Actor principal: El usuario.
  - Precondición: El usuario ha iniciado sesión en el sistema y se están visualizando las canciones de la lista de reproducción.
  - Postcondición: el sistema eliminará el conjunto de canciones seleccionadas a la lista de reproducción.
  - Escenario principal:
    1. El usuario esta visualizando la lista de canciones de la lista de reproducción.
    2. El usuario selecciona la canción o canciones a eliminar de la lista de reproducción.
    3. El sistema eliminará la canción o canciones de la lista de reproducción.
  - Escenario secundario o de error:
    - 2a - No se ha seleccionado ninguna canción y no se ha eliminado ninguna canción de la lista de reproducción.

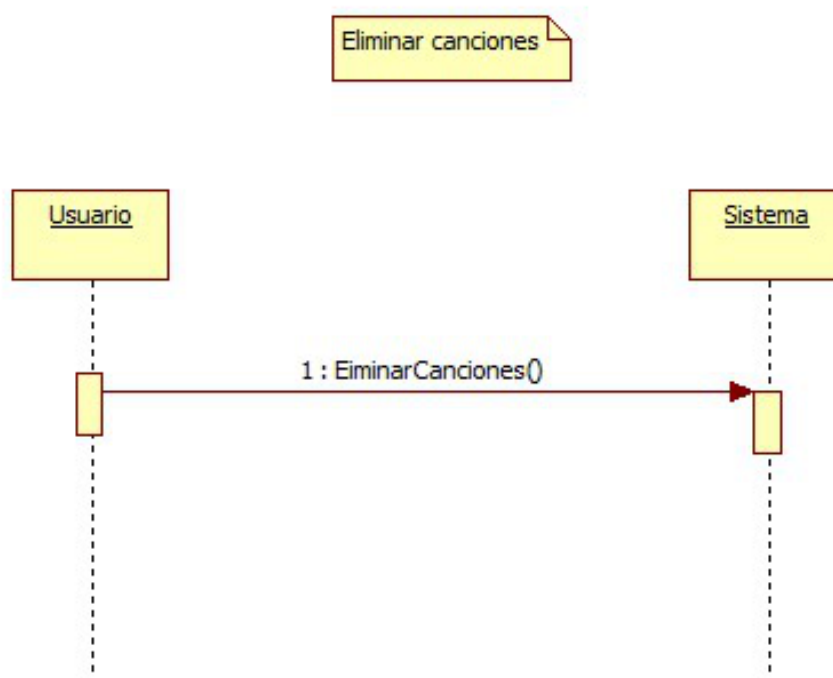


Figura 4.13: Diagrama de secuencia. Eliminar canciones.

- Caso de uso: Mostrar canciones Lista
  - Actor principal: El usuario.
  - Precondición: El usuario ha iniciado sesión en el sistema y se esta mostrando las canciones presentes en el dispositivo.
  - Postcondición: el sistema mostrará las canciones que se encuentran en la lista de reproducción.
  - Escenario principal:
    1. El usuario esta visualizando las canciones presentes en la memoria del dispositivo y no se encuentran en la lista de reproducción.
    2. El usuario selecciona mostrar la lista de reproducción.
    3. El sistema muestra las canciones que se encuentran en la lista de reproducción.



- Escenario secundario o de error:  
2a - Cuando se seleccionó el caso de uso se estaban mostrando las canciones de lista de reproducción y en consecuencia se procede a mostrar las canciones que se encuentran en la memoria del dispositivo y no están en la lista de reproducción.

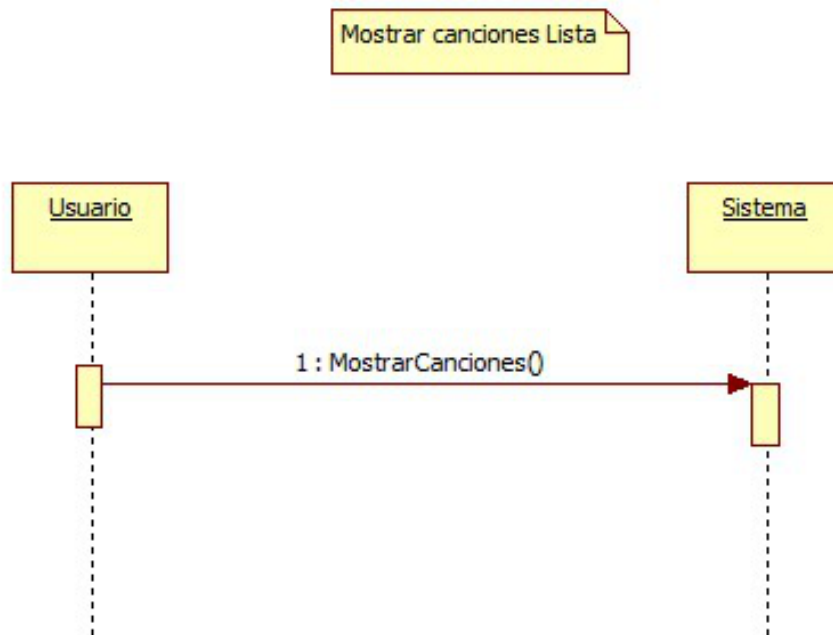


Figura 4.14: Diagrama de secuencia. Mostrar canciones Lista.

- Caso de uso: Agrupar canciones
  - Actor principal: El usuario.
  - Precondición: El usuario ha iniciado sesión en el sistema y se está mostrando una lista de canciones
  - Postcondición: El sistema agrupará las canciones por álbum o por artista.
  - Escenario principal:
    1. El usuario está visualizando una lista de canciones.

2. El usuario selecciona la agrupación por álbum o por artista.
  3. El sistema agrupa las canciones de la forma seleccionada.
- Escenario secundario o de error:
    - 3a - No hay ninguna canción en la lista de reproducción y no se agrupa ninguna canción.

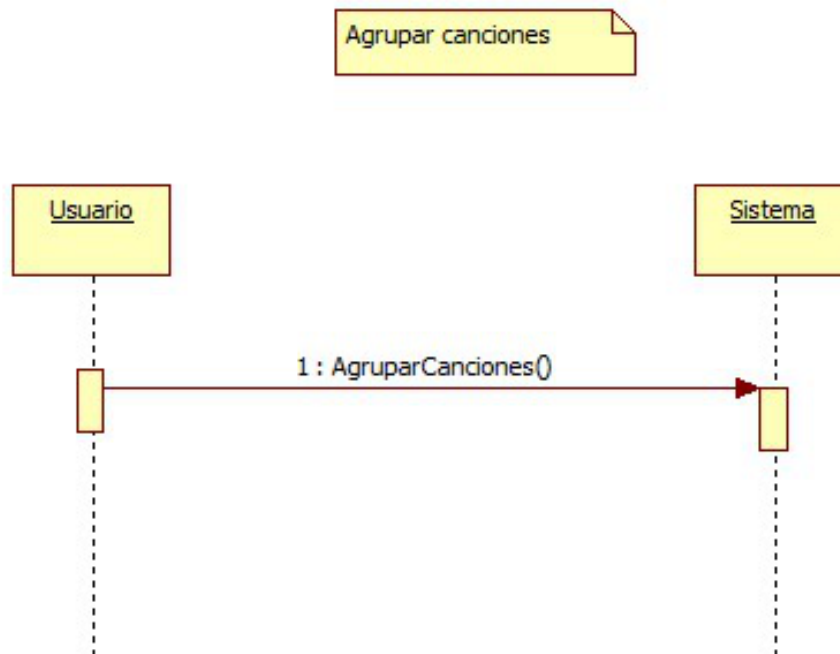


Figura 4.15: Diagrama de secuencia. Agrupar canciones.

#### 4.2.4. Casos de uso referentes a modificar listas de reproducción

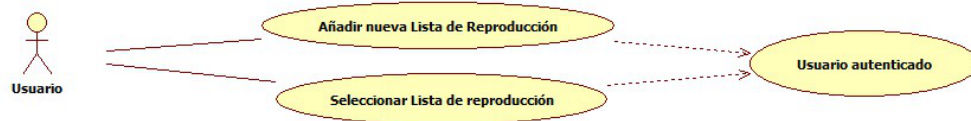


Figura 4.16: Caso de uso. Modificar listas de reproducción.

- Caso de uso: Añadir nueva lista de reproducción
  - Descripción: El usuario accederá a la sección para añadir la lista de reproducción
  - Actor principal: El usuario.
  - Precondición: El usuario tiene una sesión activa.
  - Postcondición: Se carga la sección para modificar las listas de reproducción y se añade una lista de reproducción.
  - Escenario principal:
    1. El usuario selecciona la opción que permite añadir una lista de reproducción.
    2. El sistema recibe la petición y carga la pantalla correspondiente para añadir la listas de reproducción.
    3. El usuario introduce el nombre de la lista de reproducción.
    4. El usuario introduce el nombre de la sesión en la que quiere almacenar la lista de reproducción.
    5. Se selecciona almacenar la lista de reproducción y se crea la nueva lista de reproducción.
  - Escenario secundario o de error:
    - 4a - No se introduce ningún nombre de sesión y la lista se asigna a la sesión de reproducción por defecto.
    - 5a - No se ha introducido ningún nombre para la lista de reproducción y no se almacena la lista de reproducción emitiendo una notificación al usuario.

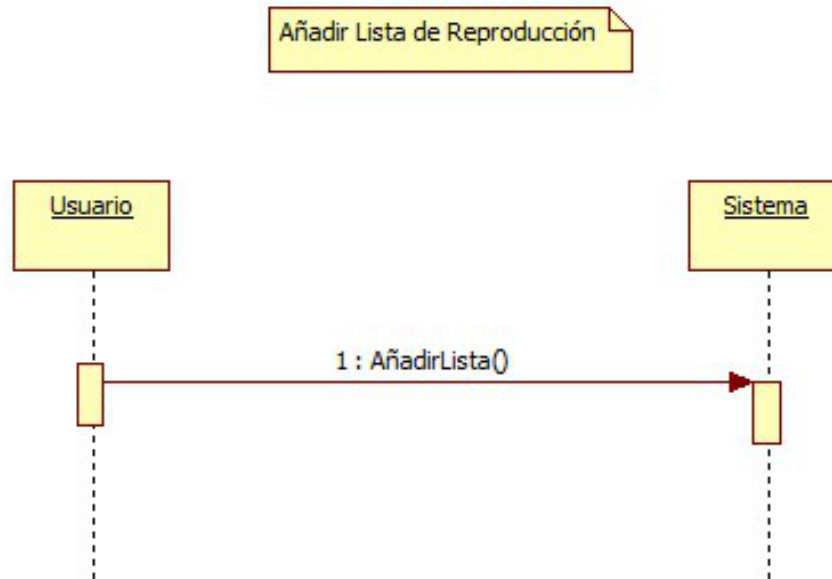


Figura 4.17: Diagrama de secuencia. Añadir lista de reproducción.

- Caso de uso: Seleccionar lista de reproducción
  - Descripción: El usuario selecciona la lista de reproducción y accede a las opciones de la lista de reproducción.
  - Actor principal: El usuario.
  - Precondición: El usuario tiene una sesión activa.
  - Postcondición: Se carga la sección seleccionar lista de reproducción para visualizarla o eliminarla.
  - Escenario principal:
    1. El usuario selecciona una lista de reproducción de las mostradas por pantalla.
    2. El sistema muestra el nombre de la lista, la sesión a la que pertenece.
    3. El usuario selecciona la opción para visualizar y reproducir las canciones.
    4. El sistema muestra las canciones de la lista de reproducción y la establece como lista de reproducción actual.

- Escenario secundario o de error:  
3a - El usuario selecciona la opción para eliminar la lista de reproducción y a continuación se muestran las listas de reproducción.

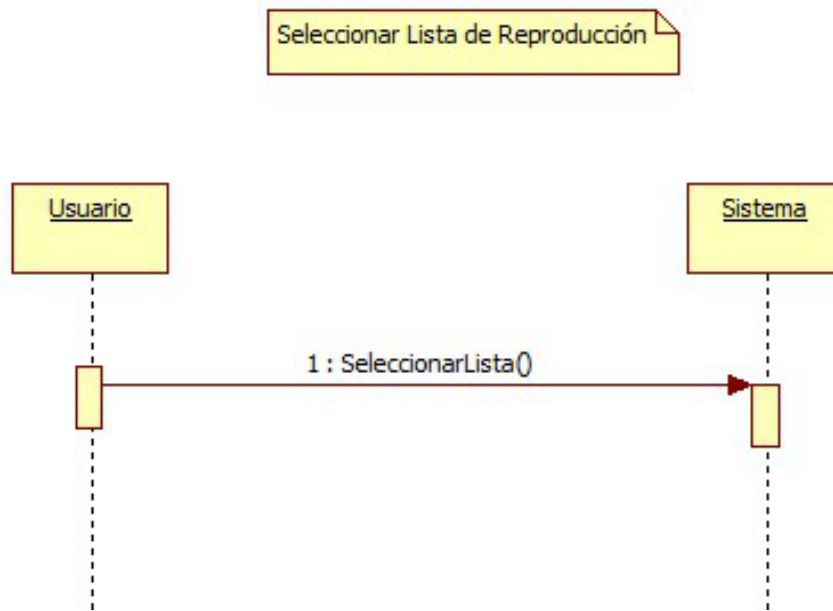


Figura 4.18: Diagrama de secuencia. Seleccionar lista de reproducción.



## 5. Diseño

### 5.1. Arquitectura del sistema software

Para la elección de la arquitectura del sistema se ha seguido el patrón arquitectónico de tres capas con el objetivo de propiciar la facilidad de mantenimiento, la reutilización de código, mejorar la cohesión y para separar correctamente la interfaz de la implementación. El cual consta de:

- Capa de gestión de datos: se encuentra en el nivel más bajo de abstracción y se encarga de interactuar con la base de datos.
- Capa de dominio: es la encargada de implementar las funcionalidades del sistema.
- Capa de presentación: es la encargada en interactuar con el usuario.

### 5.2. Diseño de la capa de gestión de datos

En la siguiente sección se procede a describir el modelo relacional empleado en el diseño de la base de datos en el dispositivo móvil.

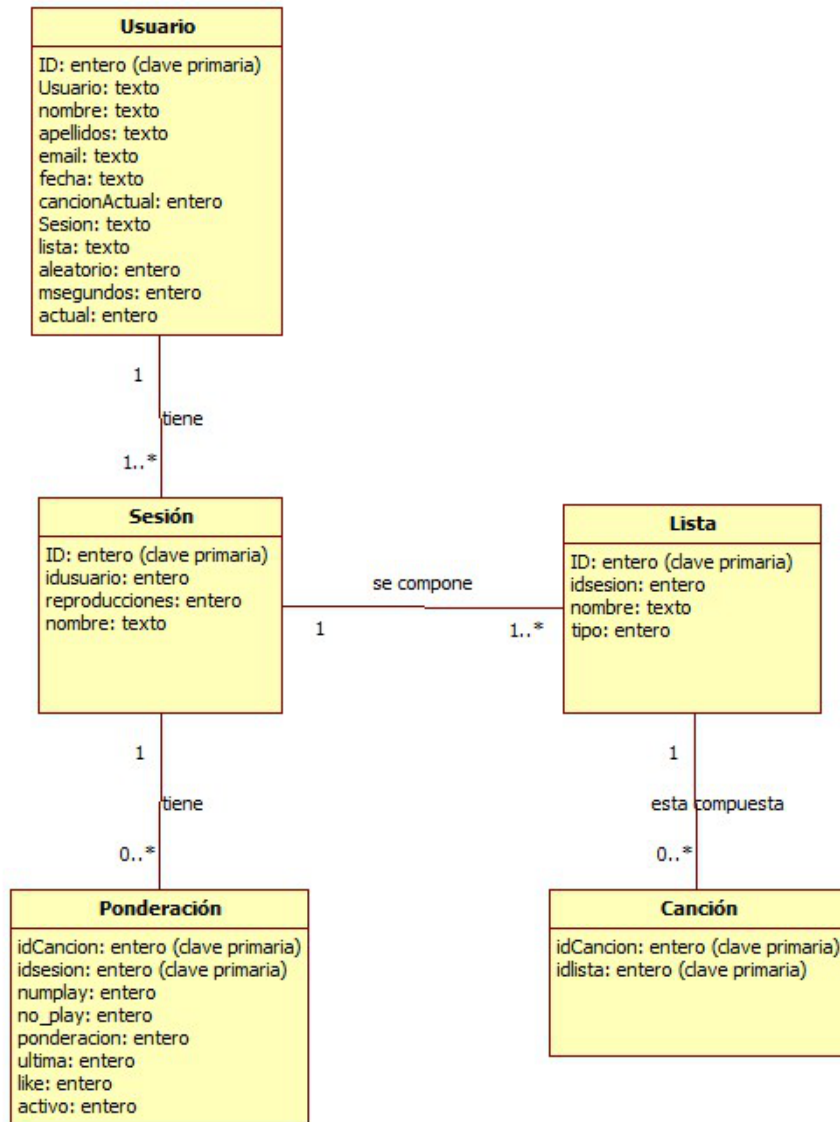


Figura 5.1: Modelo relacional de la base de datos.

En el diagrama se puede apreciar que la capa de gestión de datos esta compuesta por 5 entidades:



- Usuario, que se encarga de almacenar todos los datos relativos a los usuario almacenados en el sistema.
- Sesión, que es la que gestiona los datos de la ponderación relativos a una lista de reproducción.
- Lista, que es la encargada de almacenar una lista de reproducción.
- Canción, que almacena el identificador de cada una de las canciones que contiene una lista de reproducción.
- Ponderación, la cual almacena el conjunto de ponderaciones y datos estadísticos para poder realizar la mezcla de canciones.

La existencia de estas entidades se debe a las necesidades funcionales de la aplicación ya que al registrar un usuario en la aplicación se comprueba que el usuario es único utilizando un servicio web y las ponderaciones de las canciones son almacenadas localmente en el dispositivo para hacer posible la reproducción de canciones presentes en la memoria del dispositivo.

La realización de la mezcla musical se realiza utilizando la información previa sobre la reproducción de canciones en el dispositivo almacenando entre otros datos el número de reproducciones de cada canción así como las veces que no se ha reproducido.

Esta información se almacena en una determinada sesión, la cual almacena la información de la reproducción sobre las canciones reproducidas, las cuales están almacenadas en el dispositivo.

De forma que si queremos optimizar la reproducción en una parte del día o situación en concreto iniciamos otra sesión que almacene esta información y que posteriormente pueda realizar una mezcla de canciones óptima.

Esta serie de datos estadísticos que se utilizan para obtener la lista de reproducción se complementan con la restricción de las canciones que pueden estar en la lista de reproducción de forma que si no queremos seguir escuchando a un determinado grupo o canción podemos eliminarlo de la lista de reproducción y que la sesión siga almacenando datos para realizar mezcla de canciones.

En los dispositivos android existe una clase llamada MEDIASTORE, la cual se encarga de almacenar todos los datos relativos a los archivos multimedia, de esta forma no es necesario actualizar todos los datos de los archivos cuando se cierra la aplicación y actualizarlos con los del reproductor.

De esta forma la canción se almacena con un id único correspondiente al id de MediaStore al que hace referencia.

Para almacenar los datos en el servidor se utiliza una base de datos MySQL que tiene una entidad con los atributos necesarios para asegurar que los usuarios sean únicos así como los datos relativos a los usuarios.

| Usuario                      |
|------------------------------|
| UID: entero (clave primaria) |
| UNIQUE_ID: texto             |
| nombre: texto                |
| apellidos: texto             |
| nombreUsuario: texto         |
| email: texto                 |
| password_hash: entero        |

Figura 5.2: Base de datos del servidor.

Las bases de datos representadas anteriormente se encuentran normalizadas en la cuarta forma normal para conseguir que sea menos vulnerable a inconsistencias y anomalías.

- 1<sup>o</sup> Forma Normal: Es necesario que los atributos sean atómicos, que sus componentes no puedan ser descompuestos en elementos más pequeños.
- 2<sup>o</sup> Forma Normal: Los atributos que no son clave dependen de la clave primaria de forma completa.

- 3<sup>o</sup> Forma Normal: Si no existe ninguna dependencia funcional transitiva entre los atributos que no forman parte de una clave.
- 4<sup>o</sup> Forma Normal de Boyce-Codd: Se cumple cuando solo existen dependencias funcionales elementales que dependan de la clave primaria o de cualquier clave alternativa.

### 5.3. Diseño de la capa de dominio

#### 5.3.1. Clases de diseño

A continuación describiremos las estructuras de datos que componen el sistema, así como las relaciones entre ellas.

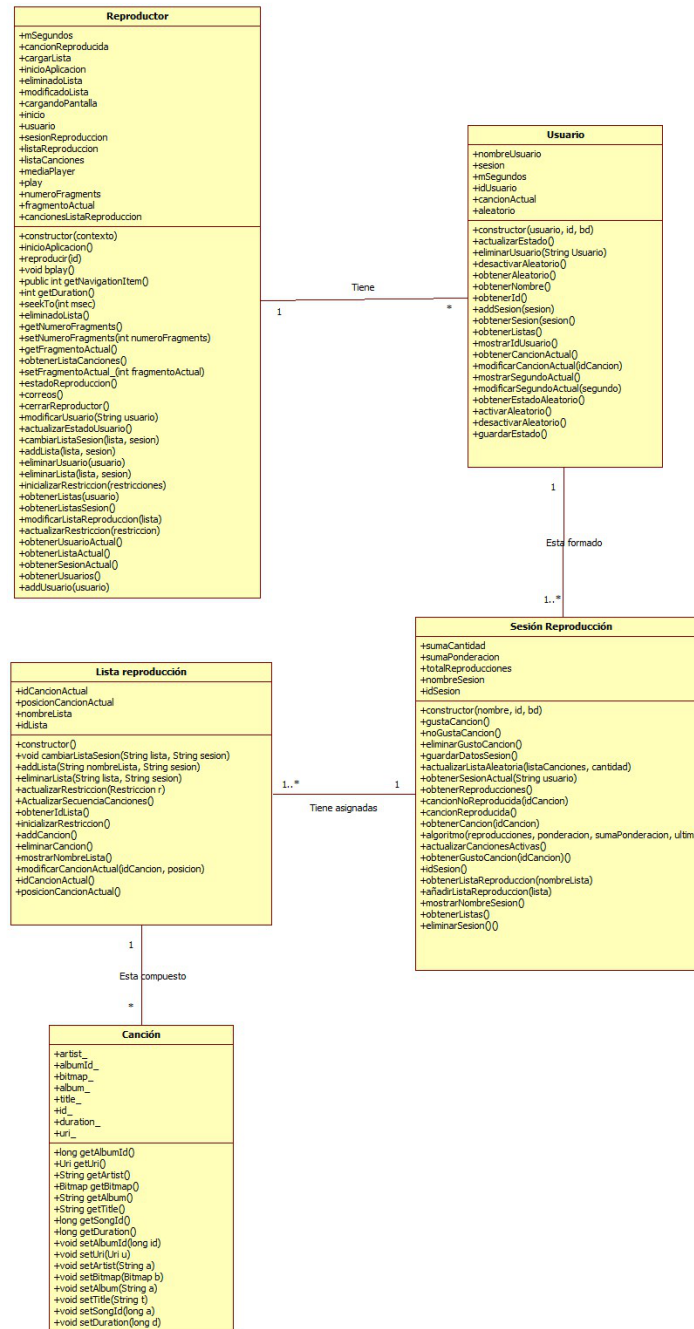


Figura 5.3: Estructuras de datos.

- Clase Reproductor:
  - constructor(contexto): Partiendo del contexto de la aplicación inicializa todos los atributos necesarios para el funcionamiento de la aplicación así como el sistema para la gestión de la base de datos.
  - inicioAplicación(): Método que devuelve verdadero si es la primera vez que se inicia la aplicación y no ha iniciado sesión un usuario en la aplicación, falso si de lo contrario hay un usuario que ha iniciado sesión en el sistema.
  - reproducir(id): Reproduce la canción presente en la memoria del dispositivo cuyo idCanción es id, preparando para ello el mediaPlayer.
  - bplay(): inicia la reproducción de la canción previamente preparada en el mediaPlayer.
  - setNumeroFragments(fragments): establece el número de canciones previamente cargadas que hay en el slider.
  - obtenerListaCanciones(): devuelve la lista de canciones que tiene la lista de reproducción actual.
  - modificarUsuario(usuario): establece el usuario como usuario actual cargando la sesión y lista de reproducción por defecto del usuario.
  - inicializarRestriccion(Restriccion restricciones): actualiza la lista de reproducción del usuario con el nuevo conjunto de canciones introducidas.
  - addUsuario(idUsuario) : Crea un nuevo usuario en el reproductor con el nombre de usuario especificado.
  - getNavigationItem(): Devuelve el elemento que corresponde en la lista desplegable con la lista de reproducción actual.
  - obtenerListasSesion(): Devuelve las listas de la sesión actual.
  - obtenerUsuarioActual(): Devuelve el usuario que se encuentra con la sesión activa en el sistema.
  - obtenerUsuarios(): Devuelve el conjunto de usuarios que ha iniciado sesión alguna vez en el sistema.

- Clase Usuario:
  - constructor(usuario,id,bd) : Crea un objeto Usuario partiendo del nombre, su id y la base de datos donde se encuentra almacenado.
  - actualizarEstado(): se actualizan parámetros de estado de la última reproducción desde la base de datos como la canción por la que se encontraba, el tiempo escuchado de la canción y el modo de reproducción.
  - eliminarUsuario(usuario): elimina el usuario especificado del reproductor.
  - obtenerNombre(): devuelve un string con el nombre de usuario.
  - obtenerId(): devuelve el id del usuario.
  - modificarCancionActual() : almacena en la base de datos la canción que se esta reproduciendo actualmente.
  - guardarEstado(nombreUsuario,cancionActual,aleatorio,mSegundos) : se almacenan en la base de datos los parámetros de estado de la reproducción.
  - activarAleatorio(): modifica el tipo de reproducción de reproducción a mezcla de canciones.
- Sesión Reproducción:
  - constructor(nombre,id,bd): Crea un objeto SesionReproducción a partir del nombre de la sesión, del id y de la base de datos donde se almacena.
  - actualizarListaAleatoria(listaCanciones,int cantidad) : Partiendo de las canciones actuales en la listaCanciones añade a la mezcla de canciones automática el número de canciones especificado en la variable cantidad, devolviendo las canciones contenidas en listaCanciones más el número de canciones especificado en cantidad.
  - gustaCancion(cancion): especificamos que la canción especificada ha sido seleccionada como que le gusta al usuario actual.
  - obtenerReproducciones(): la función devuelve el número de reproducciones totales del usuario actual en la sesión actual siendo las reproducciones totales la suma de las canciones que se han reproducido completamente y las reproducidas parcialmente.

- `cancionReproducida(cancion)`: se especifica que la canción actualmente en reproducción se ha reproducido completamente.
  - `cancionNoReproducida(cancion)`: se indica que se ha saltado a la canción siguiente a la actual interrumpiendo la reproducción de la canción actual.
  - `obtenerCancion()` : devuelve una canción utilizando el algoritmo de mezclas automáticas para calcular la próxima canción a reproducir.
  - `algoritmo(reproducciones,ponderacion,sumaPonderacion,ultima)`: función que calcula la ponderación óptima utilizando los datos del número de reproducciones totales, la ponderación actual, la suma de todas las ponderaciones de las canciones y el identificador numérico que indica el valor que tenía el total de reproducciones cuando se reprodujo la canción por última vez.
  - `actualizarCancionesActivas()`: método que actualiza las canciones presentes en la lista de reproducción con el campo activo dentro de ponderaciones.
  - `obtenerListaReproduccion(nombreLista)`: método que devuelve una lista de reproducción cuyo nombre es pasado como atributo.
  - `obtenerListas()`: devuelve todas las listas de reproducción asociadas a una determinada sesión.
- Clase Canción:
- `constructor(id,title,albumId,album, artist,bitmap,duration,uri)` : A partir del id de la canción, el título, el id del álbum, el artista, el bitmap, la duración y la dirección que hace referencia a la canción crea el objeto canción con los atributos introducidos.
  - `getAlbumId()`: Función que devuelve el id del álbum al que pertenece la canción.
  - `getUri()`: Devuelve la dirección a la que hace referencia la canción.
  - `getArtist()`: Devuelve el artista de la canción.
  - `getBitmap()`: Devuelve el bitmap de la caratula.
  - `getAlbum()`: Devuelve el nombre del álbum.

- getTitle(): Devuelve el titulo de la canción
- getSongId(): Devuelve el id de la canción
- Clase lista de reproducción
  - constructor(idLista,nombreLista,bd)
  - inicializarRestriccion() : almacena las canciones de la lista de reproducción desde la base de datos hasta la pantalla para modificar la lista de reproducción para poder visualizarlas y editarlas.
  - actualizarRestriccion(restriccion): realiza una actualización de la lista de reproducción en la base de datos desde la pantalla para modificar la lista de reproducción.
  - addCancion(cancion): añade la canción introducida como parámetro a la lista de reproducción.
  - actualizarSecuenciaCanciones(): devuelve las canciones que se encuentran en la lista de reproducción actual.

### 5.3.2. Diseño de secuencia

A continuación procedemos a describir los principales diagramas de secuencia que componen la aplicación.

A partir de los diagramas de secuencia es posible describir las interacciones entre los diferentes elementos.

Comenzaremos la descripción de los diagramas de secuencia por la identificación, cuya finalidad es la implementación de un sistema de seguridad que permita la identificación de un usuario en la aplicación.



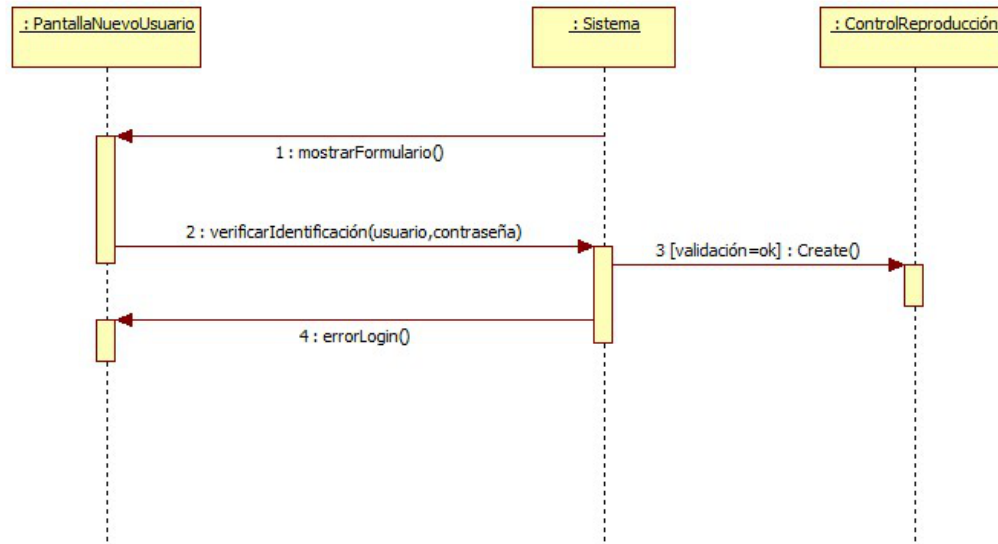


Figura 5.4: Diagrama de secuencia. Identificación.

Una vez iniciado la pantalla de control de la reproducción se puede desplegar el menú mediante el cual se accede a las opciones relativas al usuario como son cerrar la sesión o cambiar la contraseña del usuario actual.

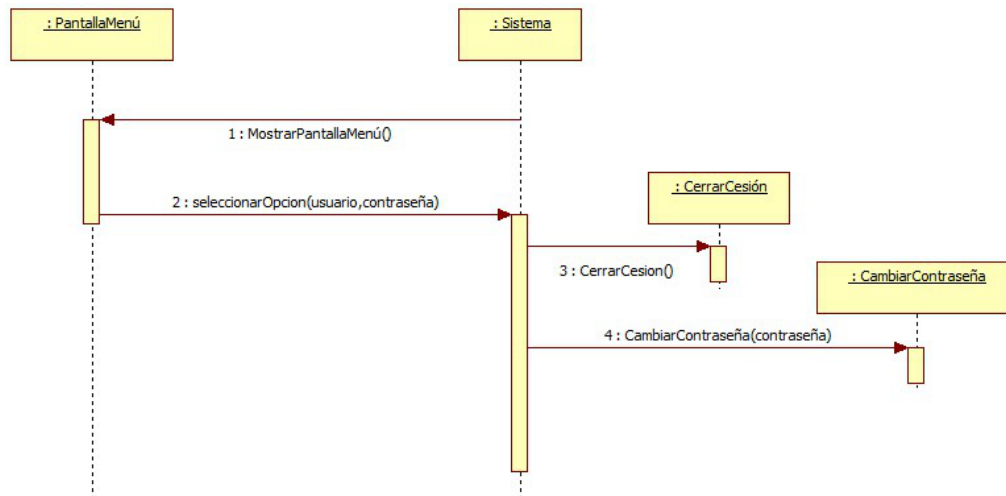


Figura 5.5: Diagrama de secuencia. Menú de opciones.

Otro diagrama de secuencia en el sistema es el de la modificación de las listas de reproducción. Para ello el usuario previamente ha iniciado sesión en el sistema selecciona la lista de reproducción y aparece una pantalla en la que se puede seleccionar si eliminar la lista de reproducción o seleccionar la lista para la reproducción.

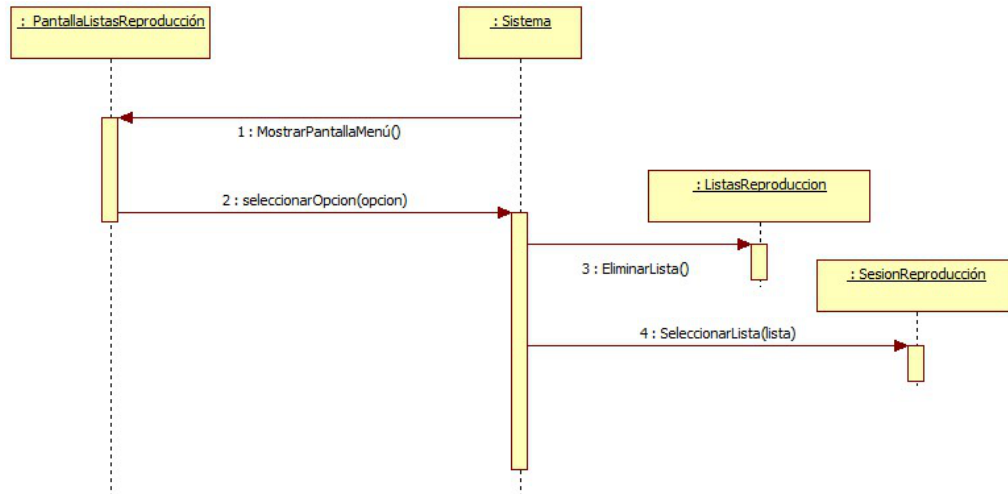


Figura 5.6: Diagrama de secuencia. Pantalla listas de reproducción.

Desde la pantalla lista de reproducción podemos acceder a la vista para añadir una nueva lista de reproducción. Cuando el sistema nos muestra la pantalla el usuario introduce el nombre de la lista de reproducción y la sesión de reproducción a la que pertenece. Si el usuario no introduce la lista de reproducción se asigna a una sesión de reproducción por defecto.

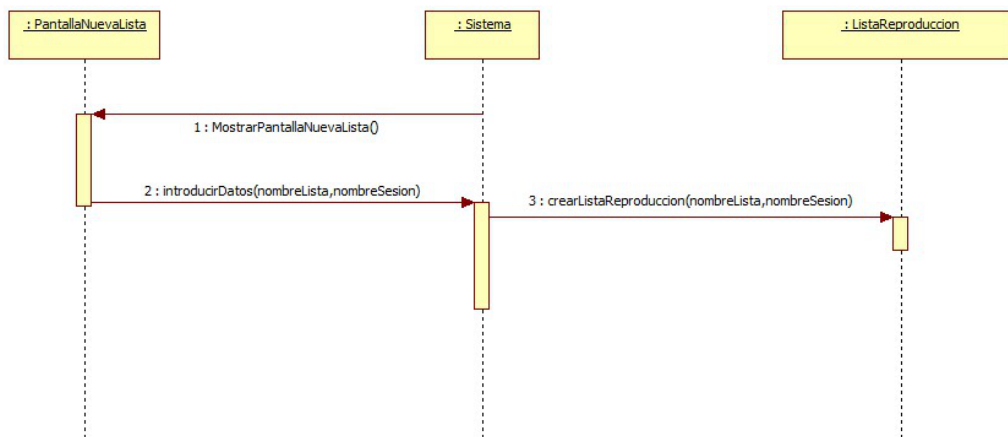


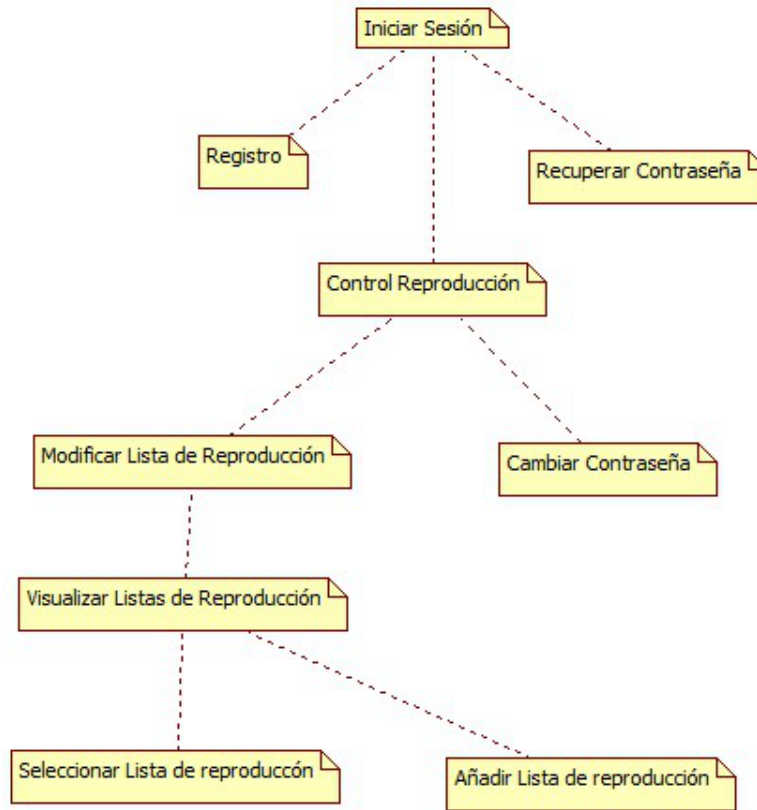
Figura 5.7: Diagrama de secuencia. Nueva lista de reproducción.

### **5.3.3. Diseño de la capa de presentación**

A continuación procedemos a la descripción de la capa de presentación. Esta capa es fundamental ya que será la única que el usuario vea y será con la que interactuará el usuario.

La finalidad de la interfaz es obtener un sistema intuitivo y práctico. Por ello se han tenido prioridad los elementos de más común acceso, dejando en un segundo plano los elementos secundarios o de características avanzadas con una menor relevancia. El objetivo ha sido hacer más accesibles las funciones más utilizadas.

- Esquema de la capa de presentación:



#### ■ Pantalla Login

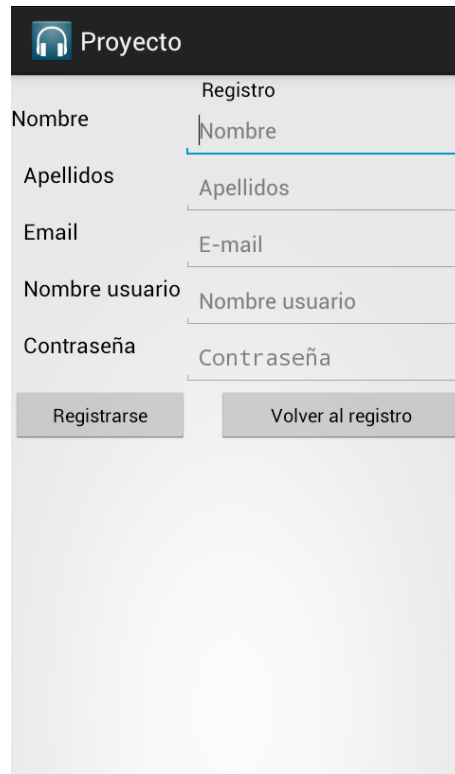
- En la zona superior izquierda de la pantalla se sitúa el logo de la aplicación.
- El email y la contraseña pueden ser introducidos desde un comboBox.
- Dispone de tres botones para iniciar sesión, dirigirse a la ventana de registro y recuperar la contraseña.

The mockup shows a mobile application interface for a project named 'Proyecto'. At the top is a dark header with a blue headset icon and the word 'Proyecto'. Below the header, the text 'Introduzca su correo y contraseña' is centered. There are two input fields: 'Email' and 'Contraseña', both with placeholder text. Below the input fields are three buttons: 'Registrarse', 'Iniciar sesión', and 'Recuperar contraseña'. At the bottom, there is a section labeled 'Usuarios:' followed by the text 'raulgmd'.

Figura 5.8: Capa de presentación. Pantalla Login.

■ Pantalla de Registro

- En esta pantalla se muestran los formularios para introducir los datos necesarios para el registro del usuario los cuales son el nombre, apellidos, email, nombre de usuario y la contraseña.
- Dispone de dos botones, uno para finalizar el registro y otro para volver a la pantalla de login.

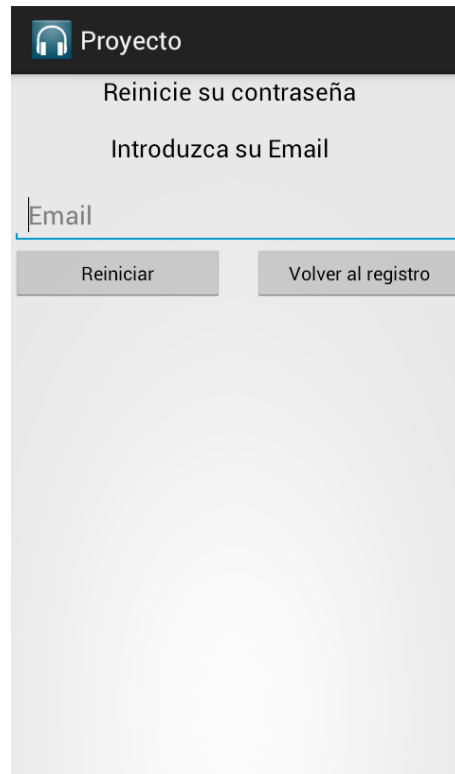


The image shows a mobile application interface for a registration form. At the top, there is a dark header bar with a blue icon of a person and the word "Proyecto" in white. Below the header, the form is organized into a table-like structure with two columns. The left column contains labels for the form fields: "Nombre", "Apellidos", "Email", "Nombre usuario", and "Contraseña". The right column contains the corresponding input fields, each with a light gray background and a thin blue border. The input fields are labeled "Registro", "Nombre", "Apellidos", "E-mail", "Nombre usuario", and "Contraseña". Below the input fields, there are two buttons: "Registrarse" and "Volver al registro". The "Registrarse" button is on the left and the "Volver al registro" button is on the right. Both buttons have a light gray background and a thin blue border.

|   | Registro       |
|---|----------------|
| Nombre                                    | Nombre         |
| Apellidos                                 | Apellidos      |
| Email                                     | E-mail         |
| Nombre usuario                            | Nombre usuario |
| Contraseña                                | Contraseña     |
| <div>Registrarse Volver al registro</div> |                |

Figura 5.9: Capa de presentación. Pantalla de registro.

- Pantalla de Recuperar Contraseña
  - En esta pantalla hay un formulario para introducir nuestro email en el cual se enviará la nueva contraseña.



The image shows a mobile application interface for password recovery. At the top, there is a dark header bar with a white headset icon and the text "Proyecto". Below the header, the text "Reinicie su contraseña" is displayed in a bold, black font. Underneath, the instruction "Introduzca su Email" is shown in a regular black font. A text input field with the placeholder "Email" is positioned below the instruction. At the bottom of the form, there are two buttons: "Reiniciar" on the left and "Volver al registro" on the right. The background of the screen is a light gray color.

Figura 5.10: Capa de presentación. Pantalla recuperar contraseña.

- Pantalla de modificación de la contraseña
  - En la pantalla para la modificación de la contraseña dispone de un campo en el que introducir la nueva contraseña. A la cual puede acceder un usuario que ha iniciado sesión dentro de la pantalla de control de la reproducción.



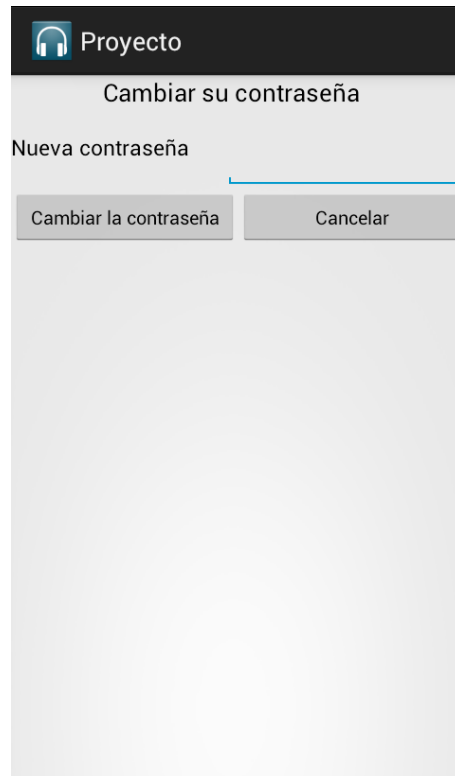


Figura 5.11: Capa de presentación. Pantalla modificar contraseña.

■ Pantalla de Control de Reproducción

- En la zona superior de la pantalla se encuentra una lista para elegir la lista de reproducción.
- A la derecha de la elección de la lista de reproducción se encuentra la opción para ir a la pantalla para modificar la lista de reproducción.
- En la parte central de la pantalla se describe la canción que se está reproduciendo actualmente mostrando el nombre de la canción, el álbum y la imagen asociada al álbum, si estuviera disponible.
- A continuación se encuentra los checkbox para seleccionar si ha gustado la canción.

- Inmediatamente después se encuentra el deslizador de tiempo para escoger el segundo que se desea reproducir de la canción.
- En la parte inferior izquierda de la pantalla, se encuentra la opción que nos permite reproducir el modo de reproducción mediante mezcla automática de canciones, o bien, la reproducción de forma secuencial de las canciones.
- A la derecha de éste, se distingue el icono que permite reproducir la canción anterior a la actual.
- El siguiente, representa la opción de pausar o continuar con la reproducción.
- Finalmente, en la parte inferior derecha se sitúa el botón para reproducir la canción siguiente a la que se está reproduciendo.

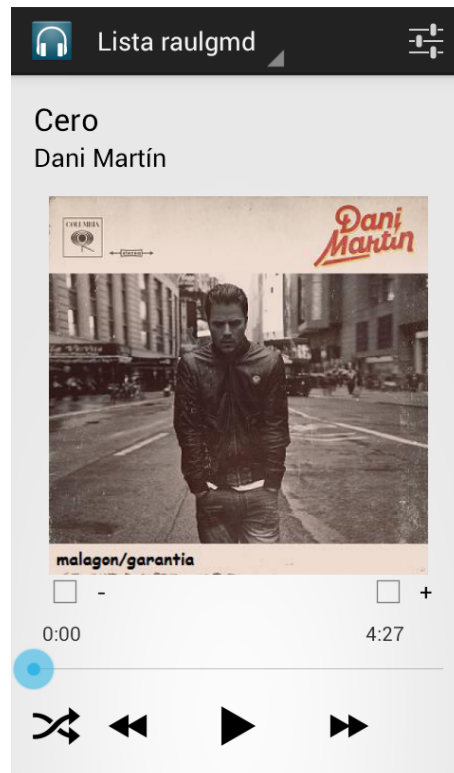


Figura 5.12: Capa de presentación. Control de Reproducción.

- Pantalla lista de canciones
  - En la parte más superior encontramos comenzando por la derecha la opción que nos dirigirá a la pantalla de acceso a las listas de reproducción.
  - A su izquierda se encuentra el icono representado por una carpeta que nos permite visualizar las canciones que están almacenadas en la memoria del dispositivo y no se encuentran en la lista actual.
  - A continuación, encontramos el icono de unos auriculares cuya funcionalidad es la de mostrar las canciones que forman parte de la lista de reproducción actual.
  - El icono representado por un símbolo + provee la funcionalidad para añadir o eliminar todas las canciones de la lista de reproducción.
  - Inmediatamente debajo del navbar figuran las tres opciones que nos permiten organizar las canciones de la lista de reproducción según el artista, el álbum al que pertenezcan o el nombre de la canción las cuales se visualizarán en la parte posterior según el criterio que hayamos seleccionado.

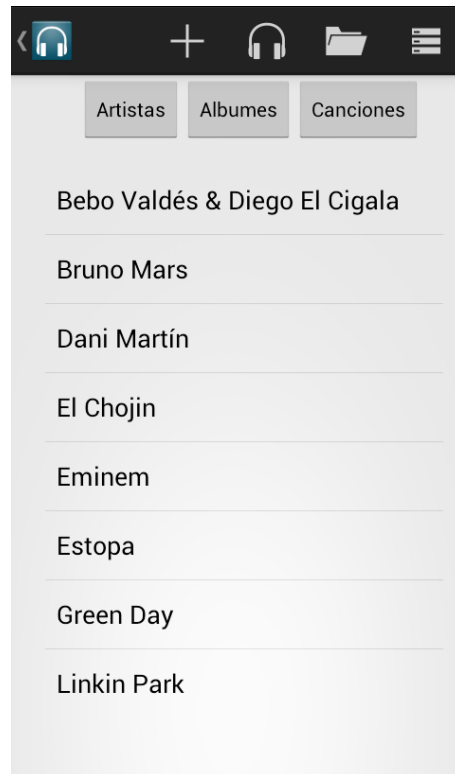


Figura 5.13: Capa de presentación. Lista de Canciones.

- Pantalla Listas de Reproducción
  - En esta ventana podemos ver dos partes bien diferenciadas la parte superior y parte inferior:
    - La parte superior compuesta por dos iconos, a la izquierda se encuentra el icono de la aplicación que nos permite retroceder en cada una de las ventanas; a la derecha de la parte superior tenemos la opción para añadir una nueva lista de reproducción, esta representada por un símbolo más.
    - La parte inferior se encuentra el conjunto de listas de reproducción. Al seleccionar una de ellas accedemos a la pantalla de opciones lista.

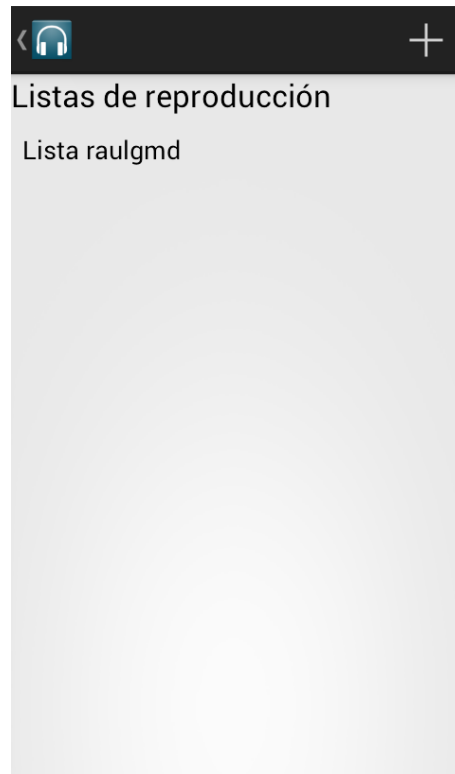


Figura 5.14: Capa de presentación. Listas de reproducción.

■ Pantalla Opciones Lista

- Esta pantalla muestra el nombre de la lista de reproducción y a continuación el nombre de la sesión a la que pertenece.
- Además, dispone de dos botones que permiten seleccionar la lista para reproducirla o eliminar la lista de reproducción.

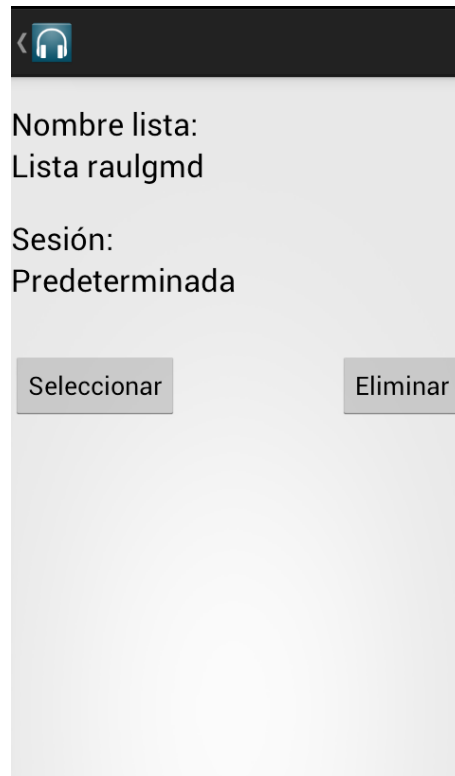
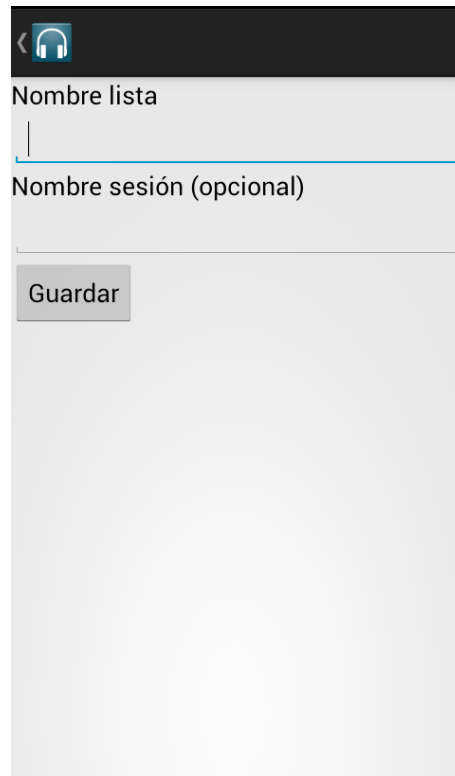


Figura 5.15: Capa de presentación. Opciones lista.

■ Pantalla Añadir Lista

- Esta ventana dispone de dos comboBox que permiten la introducción del nombre de la nueva lista de y de forma opcional la introducción del nombre de la sesión a la que pertenece, de lo contrario se asignará a la sesión por defecto.



The image shows a mobile application interface for creating a new list. At the top, there is a dark header bar with a back arrow and a headset icon. Below the header, the text "Nombre lista" is followed by a text input field with a blue underline. Underneath that, the text "Nombre sesión (opcional)" is followed by another text input field. At the bottom of the form, there is a grey button labeled "Guardar".

Figura 5.16: Capa de presentación. Nueva lista.





## 6. Implementación

### 6.1. Tecnologías

La aplicación se ha desarrollado empleando las siguientes tecnologías:

Windows 8, como sistema operativo sobre el que se ha desarrollado el proyecto.

Android, como sistema operativo del dispositivo que ejecuta la aplicación. Es el sistema operativo móvil que tiene una mayor difusión y el que dispone de un mayor número de aplicaciones disponibles y desarrolladores, lo que lo convierte en un sistema operativo con un amplio número de usuarios.

Apache como servidor web mediante el cual se proporciona acceso a la base de datos y a las funciones PHP que permiten a la aplicación comunicarse con la base de datos.

MySQL como sistema de gestión de base de datos y PHPMyAdmin para gestionar la base de datos.

### 6.2. Herramientas

En cuanto a las herramientas usadas para el desarrollo de la aplicación son las siguientes:

Para el desarrollo del código fuente de la aplicación se ha utilizado Eclipse como entorno de desarrollo integrado. Además, se ha usado el SDK de Android, el cual funciona con Eclipse a la perfección, gracias a la capacidad del entorno para permitir el uso de plugins.

Para la implementación de la aplicación Android hemos utilizado el lenguaje java junto utilizando el IDE Eclipse con Android Developers Tools.

En el desarrollo del servidor web se ha utilizado el lenguaje php junto con la tecnología JSON.

Cabe destacar la implementación del slider del control de reproducción que permite un uso intuitivo del reproductor junto con la opción de poder modificar la lista de reproducción actual en el **navBar**.

Para ello ha sido necesario heredar de `FragmentActivity` y realizar una composición de `ActionBar.OnNavigationItemSelectedListener`.

```
1
2 public boolean onNavigationItemSelected(int itemPosition, long
   itemId)
3 {
4     // le enviamos a reproductor la lista que se ha seleccionado
   boolean modificado =reproductor_.modificarListaReproduccion
   (listas.get(itemPosition));
5
6     // Cuando se modifica la lista de reproducción se cargan las
   canciones de la lista de reproducción
7
8     if (modificado)
9     {
10
11         if (reproductor_.estadoReproduccion())           reproductor_.
            bplay();
12         bplay.setImageResource(R.drawable.bplay);
13         reproductor_.setFragmentoActual_(0);
14         listaCanciones_=reproductor_.obtenerListaCanciones();
15         fragmentSize_=listaCanciones_.size();
16         reproductor_.setNumeroFragments(fragmentSize_);
17         mSectionsPagerAdapter.notifyDataSetChanged();
18         mViewPager.setCurrentItem(0);
19     }
20
21     return true;
22 }
```

Para el funcionamiento del slider es necesario implementar la clase `SectionsPagerAdapter` que hereda de `FragmentStatePagerAdapter`. Esta se encarga de cargar debidamente los fragments; los cuales son cada una de las secciones de las que se compone el slider.

Esta clase utiliza `DummySectionFragment` que es la encargada de generar cada uno de los fragments.

Los fragments son cargados según se vayan a visualizar y no se inician todos al inicio de la aplicación debido a que utilizamos la clase `FragmentManagerAdapter`.

Esta es una adaptación de la clase `FragmentManagerAdapter` la cual es más útil cuando hay un gran número de paginas, trabajando como una list view.

Cuando las paginas no son visibles al usuario, el fragment puede ser destruido, manteniendo solo el estado. Esto permite ahorrar memoria en el dispositivo comparado con `FragmentManagerAdapter` con el posible retardo al cambiar entre páginas.

```
1 public class SectionsPagerAdapter extends
   FragmentPagerAdapter
2 {
3     public SectionsPagerAdapter (FragmentManager fm)
4     {
5         super(fm);
6     }
7
8     public int getItemPosition(Object object)
9     {
10        return POSITION_NONE;
11    }
12
13    public Fragment getItem(int position)
14    {
15        Fragment fragment = new DummySectionFragment();
16        Bundle args = new Bundle();
17        args.putInt(DummySectionFragment.ARG_SECTION_NUMBER, position);
18        fragment.setArguments(args);
19        return fragment;
20    }
21
22    public int getCount()
23    {
24        return fragmentSize;
25    }
26 }
```

De igual forma ha sido necesario conocer cuando una pagina es visible a un usuario para comenzar la reproducción de esa canción y cargar el checkbox con el gusto de la canción.

Existe un caso especial cuando cargamos canciones del mix aleatorio.

Debido a la dificultad de tener cargadas un número mínimo de canciones para poder visualizarlas es necesario ir añadiendo canciones de la mezcla aleatoria conforme se van escuchando.

Para ello cuando queda una canción además de la actual se actualizan las canciones calculando 5 nuevas canciones y cargándolas apropiadamente.

También es necesario distinguir cuando se ha seleccionado un checkBox o si se ha cargado el fragment y se ha actualizado su valor. Para ello cuando se modifica manualmente un checkBox se sitúa la variable booleana actualizarCheckBox a True y esto permite que se actualice correctamente.

```
1 public static class DummySectionFragment extends Fragment
2 {
3     public static final String ARG_SECTION_NUMBER = "section_number
4         ";
5     public DummySectionFragment() { }
6
7     public void setUserVisibleHint(boolean isVisibleToUser)
8     {
9         super.setUserVisibleHint(isVisibleToUser);
10
11         if (isVisibleToUser)
12         {
13             reproductor_.setFragmentoActual_(getArguments().
14                 getInt(ARG_SECTION_NUMBER));
15
16             int gusto=reproductor_.obtenerGustoCancionActual();
17             int diferencia= listaCanciones_.size()- reproductor_.
18                 getFragmentoActual();
19
20             if (diferencia<=2 && reproductor_.obtenerAleatorio())
21                 actualizarSlider=true;
22             else
23                 actualizarSlider=false;
24
25             if (gusto==2)
26             {
27                 actualizarCheck=false;         like.setChecked(true);
28                 actualizarCheck=true;
29             }
30         }
31     }
32 }
```

```
25
26     if (gusto==1)
27     {
28         actualizarCheck=false;    like.setChecked(false);
                actualizarCheck=true;
29         actualizarCheck=false;    notLike.setChecked(false);
                actualizarCheck=true;
30     }
31
32     if (gusto==0)
33     {
34         actualizarCheck=false;    notLike.setChecked(true);
                actualizarCheck=true;
35     }
36 }
37 }
38 }
```

El método `onCreateView` es el encargado de crear cada uno de los fragments cuando lo invoque `sectionsPagerAdapter`.

En este método se carga el nombre de la canción, el nombre del álbum y la imagen del álbum si tuviera asignado alguna.

```
1  public View onCreateView(LayoutInflater inflater , ViewGroup
    container , Bundle savedInstanceState)
2  {
3  View rootView = inflater.inflate(R.layout.fragment_main_dummy,
    container , false);
4
5  TextView textView1_=(TextView) rootView.findViewById(R.id.
    textView1);
6
7  TextView textView2_=(TextView) rootView.findViewById(R.id.
    textView2);
8  ImageView imagen = (ImageView) rootView.findViewById(R.id.
    imageView);
9  Cancion cancion;
10
11 int idFragment=getArguments().getInt(ARG_SECTION_NUMBER);
12
13 if (!listaCanciones_.isEmpty())
14 {
```

```
15  cancion=listaCanciones_.get(idFragment);
16  textView1_.setText(cancion.getTitle());
17  textView2_.setText(cancion.getArtist());
18
19  Uri sArtworkUri= Uri.parse("content://media/external/audio/
    albumart");
    Uri albumArtUri=ContentUris.
    withAppendedId(sArtworkUri,cancion.getAlbumId());
20
21  Bitmap bitmap = null;
22
23  try
24  {  bitmap=MediaStore.Images.Media.getBitmap(getActivity().
    getResolver(), albumArtUri);
25  } catch (FileNotFoundException e)
26  {
27    e.printStackTrace();
28  } catch (IOException e)
29  {
30    e.printStackTrace();
31  }
32
33  if (bitmap!=null)
34  {
35    bitmap = Bitmap.createScaledBitmap(bitmap,400,400,true);
36
37    imagen.setImageBitmap(bitmap);
38  }
39  }else
40  {
41    textView1_.setText("");
42    textView2_.setText("");
43  }
44  return rootView;
45 }
```

Otro aspecto importante es la implementación de la visualización de las listas de reproducción, para el cual según la selección que se realice la aplicación es capaz de agrupar las canciones según el criterio que se le indique. Una vez seleccionado el criterio para agrupar las canciones se invoca a la clase restricciones que almacena el conjunto de las canciones utilizando el tipo de dato HashSet mediante el cual podemos buscar elementos eficientemente, ya que esta implementado como un árbol de búsqueda.

Una vez obtenido el conjunto de canciones según el criterio de selección actua-

lizamos el resultado por pantalla actualizando el adaptador y notificándole que hemos modificado el conjunto de datos.

```
public boolean onOptionsItemSelected (MenuItem item)
{
    switch (item.getItemId ())
    {
        case R.id.action_opciones:
            startActivity (new Intent (Listas.this, Opciones_lista.class));
            return true;

        case R.id.action_biblioteca:
            restricciones.visualizarBiblioteca ();
            updateData (opcion);
            return true;

        case R.id.action_opciones_mas:
            modificado=true;
            restricciones.modificarLlenoAlInicio ();
            updateData (opcion);
            return true;

        case R.id.action_biblioteca2:
            restricciones.visualizarListaReproduccion ();
            updateData (opcion);
            return true;
        case R.id.action_settings:
            startActivity (new Intent (Listas.this, Login.class));
            return true;
        case android.R.id.home:
            NavUtils.navigateUpFromSameTask (Listas.this);
            return true;
        default:
            return super.onOptionsItemSelected (item);
    }
}
```

Para finalizar es necesario comentar el algoritmo que calcula que canción se va a reproducir en un determinado momento.

Para la implementación del algoritmo nos hemos basado en un algoritmo para la simulación de la extracción sin remplazamiento, como puede ser la obtención de las cartas dentro de una baraja.

Una posible implementación sería obtener un vector de elementos booleanos

inicializado a verdadero y el elemento  $i$  del vector se corresponde con la carta  $i$  de la baraja. Si extraemos la carta  $i$  de la baraja situamos el elemento  $i$  del vector a falso.

El algoritmo consiste en calcular un número aleatorio correspondiente al número de elementos por salir, y establecer un contador que aumenta en 1 su valor cada vez que encontramos un elemento cuyo valor sea verdadero.

Cuando el contador sea igual al número aleatorio encontrado extraemos ese elemento de la baraja colocando su valor a falso.

Si modificamos este algoritmo modificando el vector por un vector de entero y situando los valores que persisten en la baraja a 100 y los eliminados a 0 nos acercamos a la implementación del algoritmo.

Nuestro algoritmo crea un vector cuyos elementos corresponden con cada una de las canciones presentes en la lista de reproducción.

Durante la reproducción de las canciones se va optimizando la ponderación de las canciones ya si no se reproduce una canción se reduce a la mitad su ponderación y si se reproduce la canción aumenta la mitad de lo que le queda. Por tanto se basa en que de forma general todas las canciones que no han sido seleccionadas como que no gustan quieren ser reproducidas en algún momento. Reduciendo o aumentado su valor vamos estimando la frecuencia con que se quiere escuchar la canción.

A la hora de obtener la canción que va a ser reproducida, obtenemos en un vector todas las ponderaciones de las canciones de la lista de reproducción.

Calculamos el total de la suma de todas las ponderaciones de este vector y calculamos un número aleatorio desde cero hasta este valor.

Vamos recorriendo el vector sumando las ponderaciones y cuando la suma sea mayor o igual que el valor obtenido en la ponderación el algoritmo devuelve esa canción como la próxima canción a reproducir. En general es un algoritmo pseudo-aleatorio basado en ponderación, cuanto mayor ponderación tenga una canción mayor probabilidad tendrá de salir del algoritmo.

Cuando se reproduce una canción, para evitar que se repitan las canción que acaban de ser reproducidas, se eliminan del algoritmo hasta que no ha pasado



la mitad de la lista de reproducción. Esto se calcula almacenando en la base de datos cual era el valor del contador general de reproducciones cuando se reprodujo cada canción.

De esta forma sabiendo el valor actual del contador de reproducciones sabemos que canciones se pueden reproducir y cuales no. Cuando el algoritmo devuelve 0 la canción no puede ser reproducida.

El método algoritmo devuelve el valor de la ponderación almacenada en la base de datos si se ha recorrido la mitad de la listas de reproducción desde que se reprodujo la canción. Se compara con el cociente entre la sumaPonderacion y ponderacion\*2 debido a que el número de reproducciones que tiene que esperar una canción para ser reproducida va en función de la ponderación que tiene la canción y de la ponderación del resto de las canciones.

En el caso que hayan 100 canciones, si cada canción tiene de ponderación 100 y la ponderación actual de la canción es 100 hay que esperar 50 reproducciones para ser reproducida. Si la ponderación fuera 50 habría que esperar 100 reproducciones y de esta forma establecemos un mecanismo mediante el cual limitar las veces que puede salir una canción.

Para finalizar queda por especificar como se implementa el gusto de la canción, si se selecciona que la canción gusta la ponderación se sitúa a 100 y no baja de 50 y si se selecciona que no gusta se sitúa la ponderación a 0.

A continuación se encuentra la implementación del algoritmo que devuelve el valor final de la ponderación:

```

1 private int algoritmo(int reproducciones, int ponderacion, int
    sumaPonderacion, int ultima)
2 {
3 // Las canciones no estan disponibles hasta que se ha recorrido
    la mitad de la lista de reproduccion (si no es la primera
    reproduccion)
4
5 if ( ponderacion==0 || ( ultima!=0 && ( (reproducciones-ultima) <
    (sumaPonderacion/(ponderacion*2)) )) )
6     {
7         return 0;
8     }
9     else
10    {

```

```
11         return ponderacion;  
12     }  
13 }
```



## 7. Pruebas

Las pruebas se pueden dividir en dos bloques: pruebas sobre los datos y pruebas sobre la interfaz.

### 7.1. Pruebas sobre los datos

Para comprobar que el reproductor almacenaba correctamente los datos se realizaron las siguientes pruebas:

- Pruebas de caja negra independientes: se realizaron pruebas de caja negra sobre aquellos objetos que hacen uso de la base de datos y no dependen de otros objetos. Para ellos se crearon usuarios, sesiones y listas de reproducción ficticias y se fueron corrigiendo los errores como el error que se producía al introducir un elemento ya existente.
- Pruebas de caja negra sobre subsistemas: se realizaron pruebas de caja negra sobre los subsistemas para confirmar su corrección. En estas pruebas se comprueba el buen funcionamiento del intercambio de información entre los distintos sistemas que lo componen.
- Pruebas de caja blanca: son las pruebas que recorren todas las posibles opciones de un módulo, para asegurarse de que todas que toda la funcionalidad esta implementada correctamente.

### 7.2. Pruebas sobre la interfaz

Sobre la interfaz se realizaron el siguiente tipo de pruebas:

- Limitación en el tipo de campo: si imitan los caracteres introducidos por el usuario en un determinado campo.
- Longitud de los campos: determinados campos como el “DNI” o el código postal tienen un número determinado de caracteres, por lo que se comprueba la longitud de la cadena introducida.

- Comprobación de campos vacíos: debido a que determinados campos pertenecientes a la base de datos no admiten valores nulos, se comprobaba que el valor de dichos campos nunca sea nulo.

### 7.3. Especificación del diseño de las pruebas

El conjunto de las pruebas realizadas a la aplicación se han realizado en 2 fases temporales.

- Durante el desarrollo de la aplicación: durante este periodo se han realizado pruebas referentes a las diferentes clases por separado. Se comprobó el buen funcionamiento de las clases que almacenan los datos relativos a la aplicación para asegurar que era correcto el tratamiento de datos como su almacenamiento posterior. Los datos introducidos durante este periodo han variado en función de los requisitos del módulo o la clase que se estuviera evaluando.
- Al concluir la aplicación: Se comenzó instalando la aplicación en diferentes móviles para comprobar el correcto funcionamiento.

Se continuó realizando pruebas iniciando la aplicación en un dispositivo que no tuviera archivos de audio. Luego se continuó con las pruebas para comprobar el buen funcionamiento del registro, introduciendo para ellos todos los datos necesarios y comprobando el envío del correo de confirmación a nuestra cuenta de email.

Seguidamente se comprobó el funcionamiento del Login. A continuación se comprobó el funcionamiento del control de reproducción, comprobando el funcionamiento del control deslizante, que se mostraba correctamente el nombre de la canción y su Artista y que se cargaba correctamente en el checkbox si la canción había sido previamente seleccionada como que era del gusto del usuario o de lo contrario que no era de su agrado. Además se comprobó que se actualizaban las canciones al modificar el modo de reproducción desde reproducción lineal a mezcla automática de canciones comprobando que la mezcla de canciones era dinámica e ilimitada.

De igual forma se comprobó el correcto funcionamiento del botón para parar la reproducción y reproducir.

A continuación se comprobó la funcionalidad de cambio de contraseña visualizando el correo de aviso en el email del usuario junto con cerrar sesión que cierra la sesión actual y envía al usuario a la pantalla de Login.

Después de comprobar la pantalla del control de reproducción se continuó con las comprobaciones de la pantalla para modificar las canciones de la lista de reproducción actual.

En ella se realizaron pruebas visualizando las canciones de la lista de reproducción o las que no se encontraban en esta.

También se agrupó las canciones por artista, álbumes o por su nombre para poder añadir las, o eliminarlas en conjunto de la lista de reproducción.

Para finalizar se añadieron todas las canciones a la lista de reproducción para posteriormente eliminarlas.

Se siguió realizando las comprobaciones de la pantalla para visualizar las listas de reproducción presentes en el dispositivo. En ella se añadieron nuevas listas de reproducción introduciendo el nombre de la lista y la sesión a la que pertenece, o introduciendo solo la sesión y dejando que el sistema la asigne a la sesión por defecto.

De igual forma se seleccionaron determinadas listas para visualizar su nombre y sesión y poder seleccionarlas como lista de reproducción actual para poder editar sus canciones y reproducirla o por otro lado eliminar esta lista de reproducción permanentemente.

## 7.4. Especificación de los procedimientos de prueba

Las pruebas fueron realizadas sobre Android 3.0, Android 4.1 y Android 4.4.

## 7.5. Documentación de la ejecución de pruebas

- Histórico de pruebas: los errores obtenidos durante la realización de las pruebas fueron los referentes errores al almacenar el estado actual de la canción debido a la complejidad de un sistema en el que están

funcionando varias funciones concurrentemente, junto a los de la correcta actualización de las canciones cuando esta activa la mezcla de canciones.

- Informe de incidentes ocurridos: no se produjeron incidentes de importancia durante la realización de las pruebas ya que todos los errores encontrados fueron subsanados correctamente.





## 8. Conclusiones

### 8.1. Introducción

La motivación de este proyecto era la de realizar una aplicación que reuniera los requisitos que eran necesarios para una aplicación que reprodujera música y que desde mi punto de vista carecían las aplicaciones que se encuentran en el mercado.

Esto se debía a que la mayor parte del tiempo reproducía música aleatoria de mi biblioteca música y en reiteradas ocasiones tenía que pasar a la canción siguiente debido esa canción no era de mi agrado en ese momento y a que las alternativas existentes de aplicaciones musicales son en streaming o hay que introducir un conjunto excesivo de parámetros para su reproducción.

De aquí surgió la idea de realizar una aplicación que siguiendo la línea de las aplicaciones actuales de reproducción fuera de forma transparente de obtener datos de reproducción del usuario y posteriormente cuando se solicitara la mezcla de canciones que utilizara esos datos para ofrecer una mezcla musical lo más satisfactoria posible. El algoritmo va aprendiendo de los gustos musicales del usuario, ya que cuando salta a la canción siguiente actualiza la ponderación de la canción y cuando se calcule la próxima canción a reproducir esa canción tendrá un menor peso dentro del algoritmo aleatorio.

Para aumentar la velocidad del aprendizaje se vio oportuno añadir la opción de indicar si la canción era o no del gusto del usuario mediante lo cual actualizar más rápidamente el algoritmo de aprendizaje.

### 8.2. Conocimientos adquiridos

Conforme realizaba el proyecto me iba cerciorando de lo importante de seguir una buena metodología de desarrollo en la ingeniería del software incluso cuando hacemos frente a un proyecto de complejidad media.

Durante la realización del proyecto he aprendido a utilizar diversas tecnologías como eclipse, las librerías de android, php, JSON y diseño de interfaces gráficas.

De la misma forma he afianzado mis conocimientos en otros ámbitos que se estudian intensamente en la carrera como son el diseño e implementación de bases de datos en MySQL y SQLite, la realización de clases mediante la programación orientada a objetos, diseño de algoritmos, generación de números pseudo-aleatorios, programación estructurada, el diseño e implementación de algoritmos de inteligencia artificial y programación concurrente debido a la simultaneidad de procesos.

También he tenido que utilizar tecnologías para la comunicación con el servidor mediante php y JSON que posibilitan la interacción con el servidor y el intercambio de información.

Una de las partes que ha necesitado mayor esfuerzo es la realización de la documentación completa ya que no había realizado nunca una documentación tan amplia y detallada.

Es de destacar la gran cantidad de tecnologías que es necesario dominar para realizar una aplicación funcional y el esfuerzo necesario pero el resultado ha sido satisfactorio. Es gratificante haber contemplado la evolución del proyecto y comprobar como con esfuerzo se transforma una idea en un proyecto y en aplicación funcional.

### **8.3. Futuro del proyecto**

La principal mejora que se le podría realizar a la aplicación sería la posibilidad de almacenar y exportar los datos almacenados en la aplicación en el servidor.

De esta forma sería posible realizar una copia de seguridad de las listas de reproducción de un usuario.

Otra posible mejora sería la exportación de los datos almacenados de la ponderación de las canciones para que al desinstalar la aplicación se pueda almacenar la información en el servidor.

Teniendo estos datos en el servidor sería posible compartir listas de reproducción con el resto de usuarios y poder recomendar música que sería del gusto del usuario y un posible enlace a la tienda música digital.

Entre las diversas mejoras que podrían realizarse a la aplicación sería la posibilidad de almacenar las listas de reproducción en el servidor y poder ser compartidas y exportadas.

Existen diversas mejoras que podrían ser aplicadas a la aplicación actual.

Otro proyecto sería la realización de aplicaciones nativas en otras plataformas como sería para iOS y Windows Phone 8. Realizando la aplicación para estas plataformas lograríamos abarcar un completísimo margen de los dispositivos móviles que existen en el mercado.





## Bibliografía y Referencias

1. Web oficial del desarrollador de Android <http://developer.android.com>.
2. Página web enfocada en encontrar soluciones a problemas de programación en diferentes lenguajes <http://stackoverflow.com/>.
3. Web en la encontrar información sobre desarrollo web <http://www.w3schools.com>.
4. Android Cookbook by Ian F. Darwin. ISBN: 978-1-4493-8841-6.
5. Applying UML and Patterns. By Craig Larman. 3<sup>a</sup> Edición. ISBN: 978-0131489066.



## A. Manual de instalación

En este apartado procederemos a detallar el manual de usuario de la aplicación para poder utilizar adecuadamente toda la funcionalidad que se encuentra implementada en el sistema.

El modo estándar de instalar una aplicación android es mediante el Play Store donde se encuentra la mayor parte de las aplicaciones dedicadas a esta plataforma.

Debido que al subir una aplicación a esta plataforma tiene un coste de 20\$ hemos decidido no subir la aplicación a la tienda de aplicaciones y que se pueda realizar la descarga desde una dirección web habilitada.

De esta forma al realizar la descarga de la aplicación se procederá a la descarga de la aplicación en el dispositivo. Una vez concluida la descarga deberemos comprobar si se encuentra activa la instalación de aplicaciones de orígenes desconocidos en nuestro dispositivo y podríamos proceder a instalar la aplicación en nuestro dispositivo. A continuación explicaremos los pasos necesarios para habilitar la opción “instalar aplicaciones de orígenes desconocidos”.

### A.1. Aplicaciones de terceros

Al contrario que ocurre con otros sistemas operativos para móviles como iOS o Windows Phone 8, en el sistema operativo Android se permite instalar aplicaciones que se encuentran fuera del Play Store que se encuentren en la memoria del dispositivo.

La opción para instalar aplicaciones de otras fuentes no viene habilitado por defecto y no es recomendable que se encuentre activado por defecto para evitar la instalación de apps que no son de confianza y que pueden realizar acciones que no deseamos en nuestro dispositivo como son el malware.

Para habilitar la instalación de software fuera de la Play Store nos dirigimos a ajustes y dentro de esta a la sección seguridad.



Dentro de seguridad hacemos click en orígenes desconocidos y dejamos señalada la opción disponible “Permitir la instalación de aplicaciones de otras fuentes diferentes a Google Play Store”.

Como hemos comentado con anterioridad después de realizar la descarga de la aplicación procedemos a abrir el gestor de archivos y nos dirigimos a la ruta en la que se encuentra el archivo que contiene la aplicación, la cual se empaqueta con una extensión de archivo .apk.

Tras ejecutar la aplicación se procederá a la instalación en nuestro dispositivo y nos pedirá que confirmemos si deseamos instalar la aplicación en nuestro dispositivo. Una vez instalada la aplicación podremos acceder a la aplicación seleccionando el icono correspondiente a la aplicación en el menú aplicaciones de nuestro dispositivo



## **B. Manual del usuario**

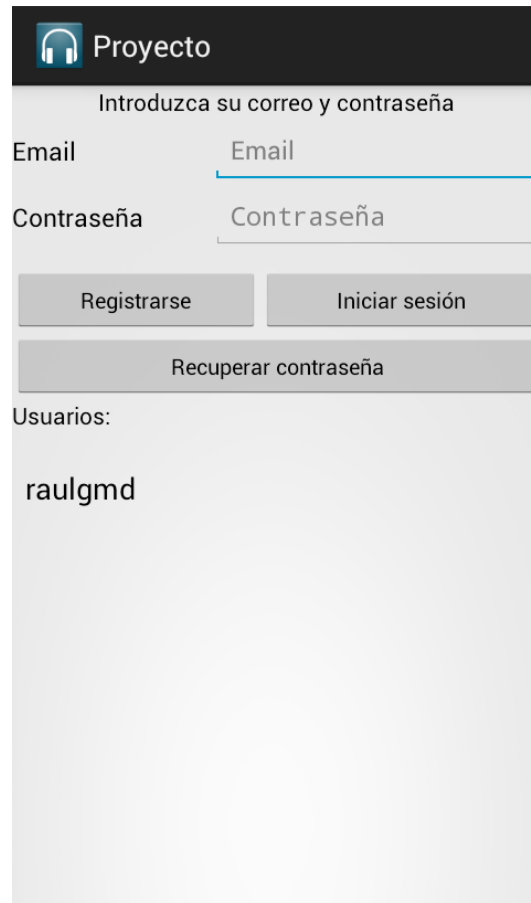
Para el correcto funcionamiento de la aplicación es necesario tener la versión 3.0 o superior del sistema operativo Android y una capacidad de almacenamiento de 2 Mb como mínimo. Como se ha explicado anteriormente, debido a que la aplicación no está disponible en la tienda de aplicaciones de Android, Google Play Store, tendremos que tener habilitada la opción de instalar aplicaciones de orígenes desconocidos.

### **B.1. Pantalla de Login**

En la pantalla inicial tenemos los formularios necesarios para la identificación del usuario, como son el email y la contraseña.

Además de los botones para iniciar sesión, registrarse en el sistema en el caso que no se hubiera registrado previamente y la opción para recuperar la contraseña.

.



Proyecto

Introduzca su correo y contraseña

Email

Contraseña

Registrarse

Iniciar sesión

Recuperar contraseña

Usuarios:

raulgmd

Figura B.1: Manual del usuario. Pantalla de inicio.

## B.2. Pantalla Registro

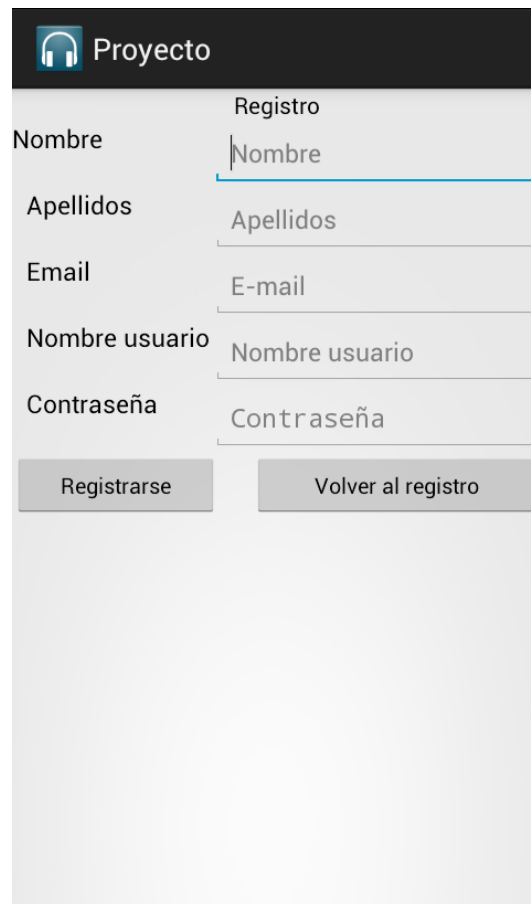
Si es la primera vez que hacemos uso de la aplicación tendremos que seleccionar la opción de registro mediante la cual se nos mostrará unos formularios con los datos necesarios para realizar el registro.

Los campos que deberemos de introducir son el nombre y apellidos, el email en el que recibiremos las notificaciones cuando nos registremos.

Hay que tener en cuenta que de no introducir correctamente el correo corre-mos el riesgo que ante un olvido de la contraseña perdamos el acceso a la cuenta y que no podríamos acceder a ella.

Si no recordamos la contraseña bastaría con seleccionar la opción de recuperar contraseña y al introducir el email la nueva contraseña sería enviada a nuestro correo.

Además de lo descrito anteriormente es necesario introducir el nombre de usuario mediante el cual seremos identificados en la aplicación y la contraseña que junto con el correo nos servirá para identificarnos en el sistema.



| Proyecto                                  |   |
|---|---|
| Registro                                  |   |
| Nombre                                    | <input type="text" value="Nombre"/>         |
| Apellidos                                 | <input type="text" value="Apellidos"/>      |
| Email                                     | <input type="text" value="E-mail"/>         |
| Nombre usuario                            | <input type="text" value="Nombre usuario"/> |
| Contraseña                                | <input type="password" value="Contraseña"/> |
| <div>Registrarse Volver al registro</div> |   |

Figura B.2: Manual de Usuario. Registro Usuario.

### B.3. Pantalla control reproducción

Al haber iniciado sesión satisfactoriamente nos encontramos con el menú de control de la reproducción, el cual tiene el conjunto de funcionalidades con más relevancia en el sistema como el control de la reproducción actual, la selección de la lista de reproducción y el control del modo de reproducción actual pudiendo cambiar el modo de reproducción de lineal a aleatorio.

Igualmente podemos visualizar los detalles de la canción actualmente en reproducción como son el nombre de la Canción, el nombre del Artista, la imagen del Álbum y el seekBar donde podemos avanzar la reproducción.

En la parte superior derecha de la pantalla nos encontramos con la opción que nos dirigirá a la modificación de la lista de reproducción actual.

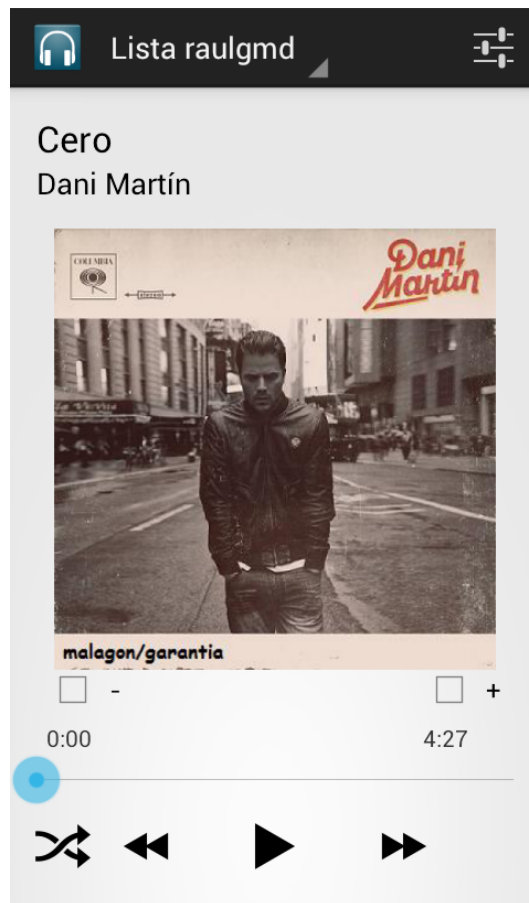


Figura B.3: Manual del usuario. Pantalla control de reproducción.

## B.4. Pantalla lista de canciones

En la pantalla de listas de canciones podemos visualizar las canciones presentes en la lista de reproducción actual pulsando en el icono de los cascos, que lo podemos encontrar en la parte superior de la pantalla, dentro del navBar.

Hecho esto podemos ver las canciones presentes en la lista de reproducción actual y mediante el símbolo + adyacente al de los cascos podemos añadir todas las canciones a la lista de reproducción o eliminar todas las canciones, según el estado actual de la lista de reproducción.

Las canciones que se muestran en la lista pueden ser agrupadas por Artistas, Álbumes o Canciones, según las preferencias del usuario.

El símbolo de la carpeta nos permite visualizar las canciones que no están presentes en la lista de reproducción actual, de forma que si estamos viendo las canciones presentes en la memoria del dispositivo y arrastramos una canción hacia un lateral, esta desaparecerá de las canciones que se muestran al pulsar el símbolo de la carpeta y aparecerá al pulsar el símbolo de los cascos.

De esta forma podemos visualizar rápidamente que canciones tenemos en la lista de reproducción actual y cuales no están incluido, mediante la selección del símbolo correspondiente.

Si estamos visualizando la lista de reproducción (habiendo pulsado el símbolo de los cascos) y estamos visualizando las canciones por Artistas (habiendo pulsado Artistas) podemos eliminar un grupo de la lista de reproducción colocando el dedo en el nombre del artista y deslizando el dedo hacia uno de los lados.

Podemos añadir el artista posteriormente pulsando el símbolo de la carpeta, y deslizando el artista hacia uno de los lados. El artista desaparecerá de la lista perteneciente a las canciones en el dispositivo pero aparecerá en las canciones presentes en la lista de reproducción.

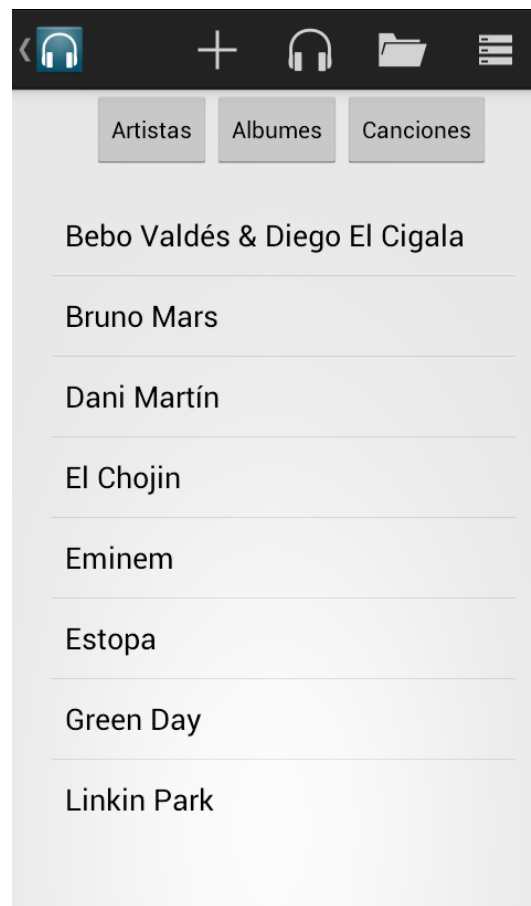


Figura B.4: Manual del usuario. Pantalla lista de canciones.

## B.5. Pantalla Listas de Reproducción

En la pantalla de gestión de las listas de reproducción podemos visualizar las diferentes listas de reproducción presentes en el dispositivo.

Podemos seleccionar una lista de reproducción y se muestra el nombre de la lista de reproducción y la sesión de reproducción a la que pertenece la lista seleccionada.



La modificación de la sesión de reproducción es una funcionalidad avanzada que no es necesario utilizar para añadir nuevas listas de reproducción debido a que de no especificarlo las listas de reproducción son asignadas a una sesión de reproducción por defecto.

Al iniciar la aplicación se crea una lista de reproducción automática en la cual el usuario puede añadir y eliminar canciones pudiendo utilizar la funcionalidad principal de la aplicación.

Para usuarios que deseen almacenar listas de reproducción específicas es posible crear nuevas listas de reproducción pulsado el símbolo  $+$  en la parte superior derecha de la pantalla.

Al añadir la lista de reproducción se introduce el nombre de la lista de reproducción y el nombre de la sesión de forma opcional, ya que se asignará a la sesión por defecto en el caso de no introducir ninguna.

La sesión de reproducción almacena todos los datos estadísticos sobre todas las canciones que pertenezcan a una lista de reproducción que estén asignadas a la lista.

De forma que las sesiones están compuestas de listas de reproducción, estas a su vez de listas de canciones y cada sesión de reproducción almacena las ponderaciones y estadísticas sobre las canciones que forman parte de ella.

De forma que si nuestros gustos musicales son diferentes por la mañana que por la noche creamos una lista de reproducción por la mañana en una sesión de reproducción diferente a la de por la noche de forma que el algoritmo pueda asignar diferentes ponderaciones a la canciones que escuchemos y realice una mezcla de canciones más exacta.

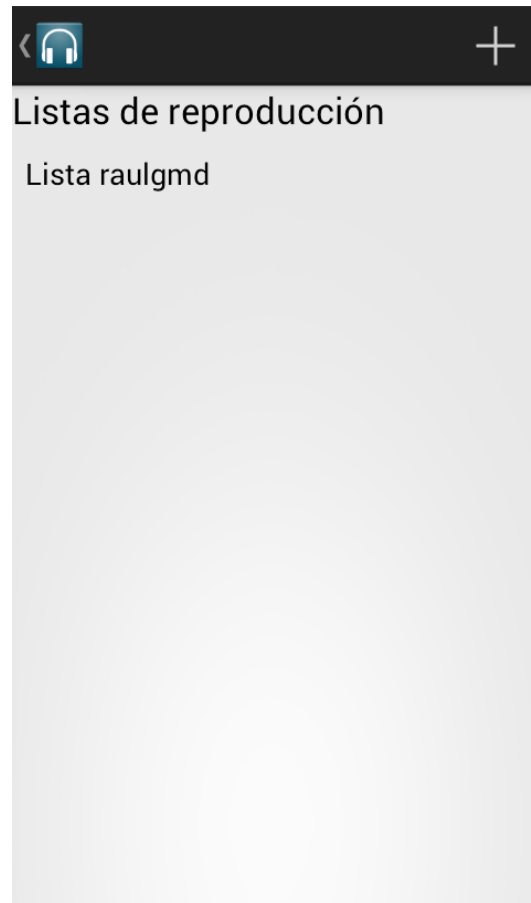
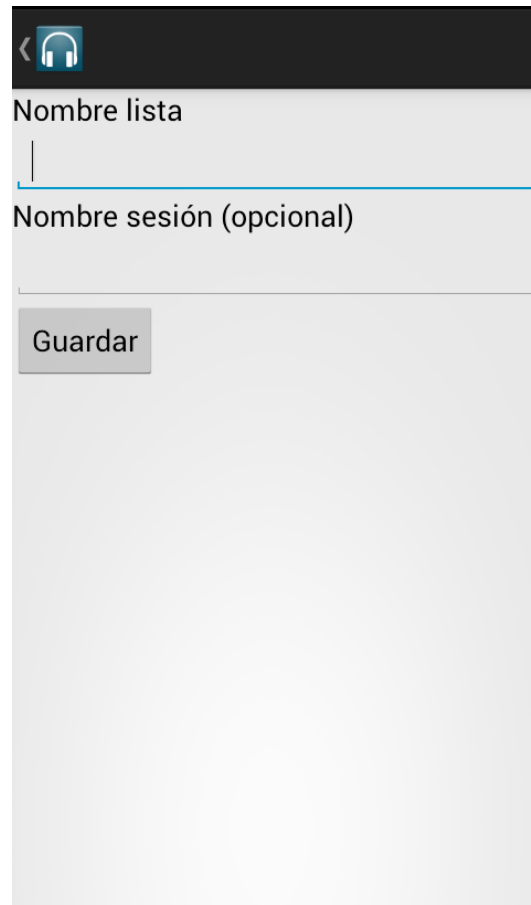



Figura B.5: Manual del usuario. Pantalla Listas de reproducción.

## B.6. Pantalla nueva lista

En la pantalla nueva lista procedemos a introducir el nombre de la lista de reproducción junto con el parámetro opcional del nombre de la sesión de reproducción.

Al seleccionar Guardar añadimos la nueva lista de reproducción al usuario actual y se muestra la ventana de listas de reproducción.



< 

Nombre lista

Nombre sesión (opcional)

Guardar

Figura B.6: Manual de usuario. Nueva lista.

## B.7. Pantalla modificar contraseña

Estando situados en la pantalla del control de reproducción es posible modificar la contraseña actual.

Los móviles android actuales tienen a eliminar el botón físico de opciones, menú y pantalla anterior. Para ello disponen de los botones de forma táctil en la pantalla.

Ya sea un móvil que dispone de los botones físicos o que las opciones se muestran en la pantalla táctil es posible seleccionar el botón de opciones que nos dará acceso a un desplegable mediante en el cual podremos acceder además de a modificar la contraseña actual a cerrar la sesión actual.

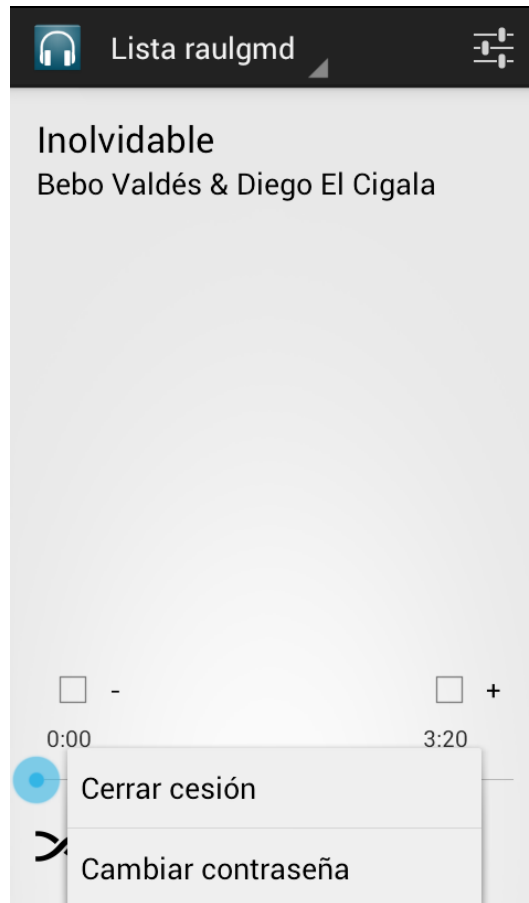
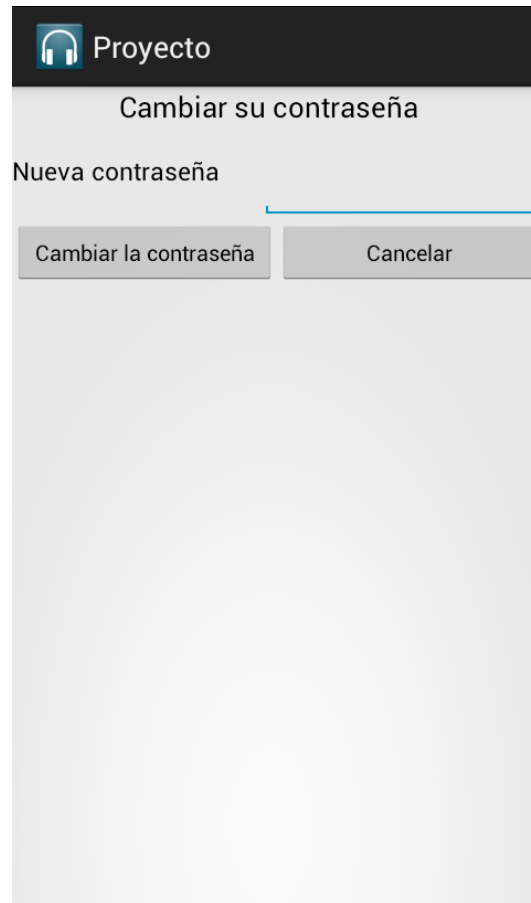


Figura B.7: Manual de usuario. Menú control reproducción.

Una vez seleccionada la opción se nos muestra la pantalla para la modificación de la contraseña en la cual introduciremos la nueva contraseña.

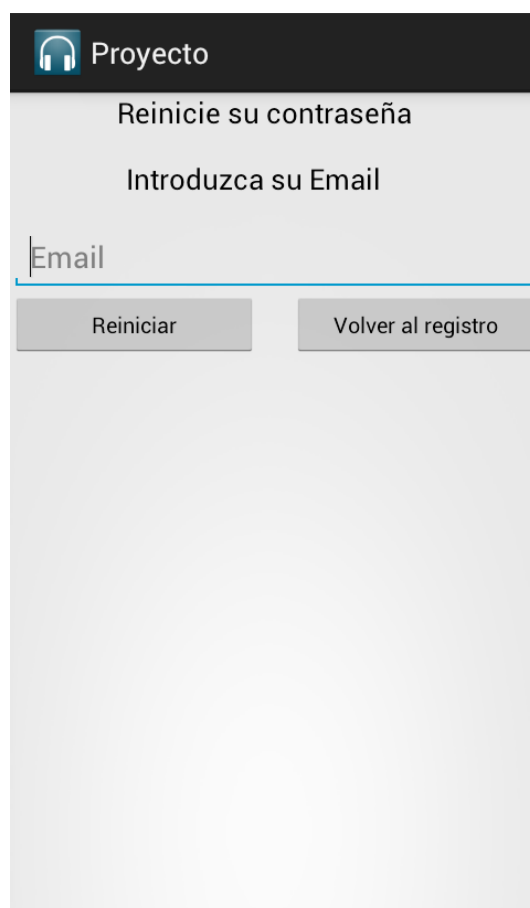


The screenshot shows a mobile application interface for changing a password. At the top, there is a dark header bar with a blue icon of a person and the word "Proyecto" in white. Below the header, the title "Cambiar su contraseña" is centered in a light gray box. Underneath the title, the text "Nueva contraseña" is displayed. A blue horizontal line indicates the input field for the new password. At the bottom of the form, there are two buttons: "Cambiar la contraseña" and "Cancelar".

Figura B.8: Manual de usuario. Cambiar contraseña.

## B.8. Pantalla recuperar contraseña

Cuando no podamos recordar el email y contraseña requeridos para entrar en el sistema seleccionaremos la opción de recuperar contraseña, mediante la cual mandaremos una petición de cambio de contraseña al servidor y se nos enviará al correo una nueva contraseña que posteriormente podremos modificar.



The screenshot shows a mobile application interface for password recovery. At the top, there is a dark header bar with a white headset icon and the text "Proyecto". Below the header, the text "Reinicie su contraseña" is displayed. Underneath, the instruction "Introduzca su Email" is shown. A text input field with the placeholder "Email" is present, with a blue underline. At the bottom, there are two buttons: "Reiniciar" and "Volver al registro".

Figura B.9: Manual de usuario. Recuperar Contraseña.