



Práctica 2: proyecto de aplicación telemática – Home Droid Streamer

Alberto Mateos Checa

Implementación de sistema de streaming de archivos mp3 desde PC a teléfono móvil con sistema operativo Android.

1. Introducción.-

El nombre del sistema implementado es **Home Droid Streamer (Hodroist)** y permite a un usuario escuchar en un teléfono móvil con sistema operativo Android música en formato mp3 que se encuentre almacenada en un ordenador, realizando un streaming de ésta a través de Internet.

Además, el usuario puede visualizar en su teléfono la información relativa a los archivos mp3 (artista, título, álbum y año) que se encuentra en los tags ID3 así como la imagen (artwork) asociada a cada uno.

Por último, hay que decir que la totalidad del proyecto ha sido realizada en inglés con la intención de, en un futuro, mejorar la aplicación e intentar ofrecer el servicio a usuarios de forma real. En cualquier caso, el código del sistema sería liberado mediante el alojamiento del mismo en Google Code.

2. Partes.-

Hodroist consta de 3 subsistemas o aplicaciones claramente diferenciadas: servidor de registro y directorio, servidor de streaming y cliente Android.

2.1 Servidor de registro y directorio.-

El servidor de registro y directorio consta de una serie de **páginas web** y una base de datos **MySQL**. Este servidor es accesible a través de Internet aunque en este caso ha sido implementado de modo local usando MAMP.

De esta forma, el usuario puede crear una cuenta a través de la web principal del sistema introduciendo nombre de usuario, email de contacto y contraseña. Esta información es almacenada en la base de datos MySQL. Además, el servidor admite consultas para almacenar la dirección IP del ordenador de un usuario y para obtener dicha IP almacenada.

Es por ello que este servidor sirve como nodo intermedio entre el servidor de streaming y el cliente Android, permitiendo la interconexión de ambos sin necesidad de que el cliente conozca previamente la dirección IP del servidor de streaming.

Base de datos MySQL.

Tal y como se ha comentado anteriormente, los datos de los usuarios serán almacenados en una base de datos MySQL. Para ello se dispone de una tabla con los siguientes campos:

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	id	bigint(7)			No	None	auto_increment
<input type="checkbox"/>	username	char(100)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	password	char(100)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	email	char(100)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	server_ip	char(15)	latin1_swedish_ci		No	None	

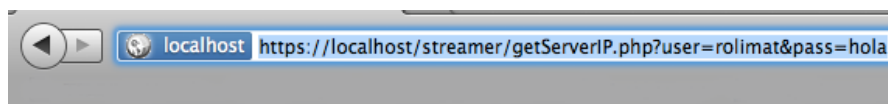
Como puede verse, para cada usuario se van a tener los campos id, username, password, email y server_ip. Es en este último en el que se almacenará la dirección IP del ordenador donde se ejecuta la aplicación que realizará el streaming.

Por motivos de seguridad, se almacena la huella MD5 de la contraseña en lugar de la propia contraseña.

Archivos web.

Los archivos web que se encuentran en la carpeta accesible del servidor web son los siguientes:

- **db.php:** contiene las instrucciones necesarias para establecer la conexión con la base de datos. Es incluido en aquellos archivos .php que realizan operaciones con la base de datos.
- **functions.php:** contiene una serie definiciones de funciones php que permiten entre otras cosas agregar usuarios a la base de datos o comprobar si existe un usuario.
- **getServerIP.php:** devuelve en formato **JSON** la dirección IP del servidor de streaming de un usuario. Para ello es necesario pasarle, en forma de parámetros, el nombre de usuario y contraseña a través de la URL.

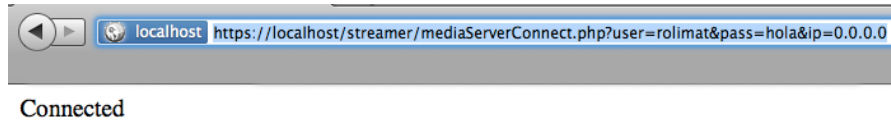


```
[{"ip": "192.168.1.66"}]
```

- **index.html:** se trata de una página web a través de la cual se introducen los datos del usuario para crear una cuenta.



- **mediaServerConnect.php**: gestiona el almacenamiento de la dirección IP del servidor de streaming relativo a un usuario en la base de datos. Para ello, es necesario pasarle mediante parámetros en la URL el nombre de usuario, la contraseña y la dirección IP que se quiere almacenar. Devuelve el resultado de la operación en formato texto. Este resultado será mostrado posteriormente en la GUI de la aplicación del servidor de streaming.



- **register.php**: gestiona el registro de usuarios para la creación de cuentas tomando los datos necesarios del formulario de la web index.html.

Además, también se dispone de los archivos **style.css**, **webTop.php** y **webBottom.php**, todos ellos necesarios para proporcionar el aspecto visual de la web index.html y de la respuesta al registro que proporciona register.php.

Por último, hay que destacar que el servidor web Apache utilizado permite la consulta de webs cifradas a través de https (SSL) tal y como se muestra en la siguiente captura:



Así, las comunicaciones que se producen entre cualquiera de las aplicaciones y éste servidor y la creación de cuentas mediante la interfaz web están cifradas. Para ello, puesto que el certificado es auto-firmado y no ha sido expedido por una autoridad certificadora autorizada, las aplicaciones aceptan el certificado del servidor sin comprobarlo, es decir, no hacen uso de ninguna clave ya que no es posible utilizar claves para aceptar certificados obtenidos mediante peticiones https.

2.2 Servidor de streaming.-

Es la aplicación que se ejecuta en el ordenador desde el que se realizará el streaming de la música hacia el móvil Android. El programa ha sido realizado en Java y las clases que intervienen en el funcionamiento del mismo no serán explicadas aquí puesto que se entregan junto con este documento y están debidamente comentadas.

Interfaz de usuario.

Para ello, en el mismo directorio desde donde se ejecuta la aplicación debe de tenerse otro directorio llamado **Music**, que será en el que se almacenará la música en formato mp3 que se enviará al móvil.

El aspecto de la aplicación es el siguiente:



Como puede comprobarse, se pueden diferenciar 3 zonas en la interfaz de usuario. En la zona superior se encuentra el menú de la aplicación que permite salir de la misma (File → Exit), realizar una conexión o desconexión con el servidor de directorio (a través de Connection) y obtener información del programa (Help → About).

Más abajo, en la zona central, se visualizan las canciones que han sido escaneadas de la carpeta Music, mostrando la información del archivo extraída de sus etiquetas ID3.

Por último, en la zona inferior, se tiene la barra de status, la cual muestra información acerca de la conexión o no conexión de la aplicación con el servidor de directorio.

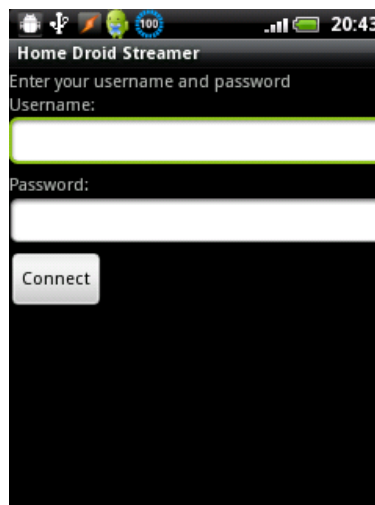
2.3 Cliente Android.-

Es una aplicación desarrollada para móviles con sistema operativo Android que permite la conexión con el servidor de streaming y la reproducción de audio procedente de dicho servidor. Además, permite elegir la canción a reproducir de entre los archivos disponibles para el streaming y muestra la información de cada uno así como la imagen asociada.

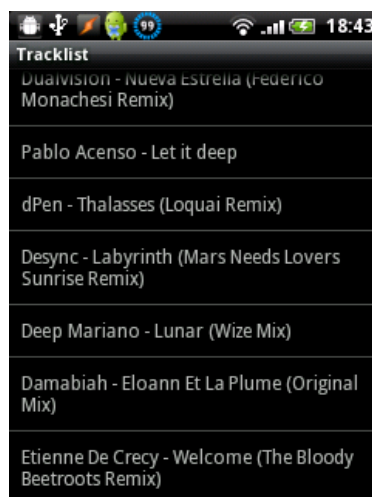
Interfaz de usuario.

La aplicación está formada por 3 pantallas o activities:

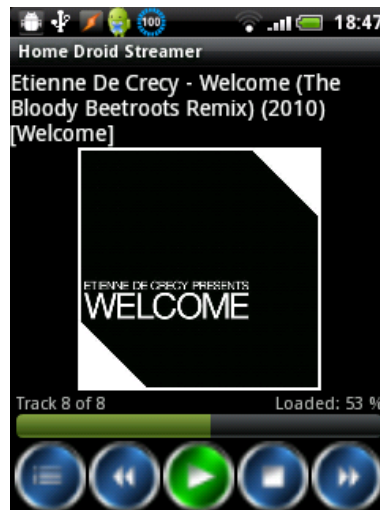
- **Login:** es la pantalla que se muestra al ejecutar la aplicación. Permite al usuario introducir su nombre y contraseña para realizar la conexión con el servidor de directorio y, posteriormente, con el servidor de streaming.



- **Tracklist:** muestra la lista de canciones disponibles para su reproducción. Al pulsar sobre una de ellas se procede a la reproducción de la misma.



- **Player:** reproduce la canción seleccionada y muestra información sobre la misma y su imagen asociada. Además, incluye los controles necesarios para la reproducción de las canciones y para volver a ver la lista de canciones (botón de la izquierda).

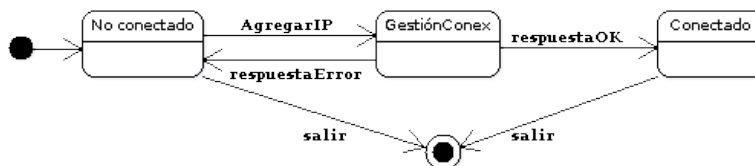


3. Comunicaciones y protocolos.-

3.1 Servidor directorio – servidor streaming.-

El servidor directorio y el servidor de streaming están comunicados a través de Internet mediante una conexión SSL. De esta forma, el servidor de streaming realiza peticiones de tipo GET HTTPS al servidor de directorio, en concreto al programa mediaServerConnect.php. Mediante esta petición el servidor de streaming almacena su dirección IP en la base de datos. Para la conexión HTTPS se utiliza un certificado auto-firmado, por lo que el servidor de streaming ha sido programado de forma que acepta cualquier certificado.

La comunicación realizada puede representarse a través del siguiente diagrama de estado:



Como puede verse en el diagrama anterior, el programa se inicia en el estado “No conectado”. Tras enviar su dirección IP al servidor de directorio pasa al estado “GestiónConexión”. En caso de que la dirección se almacene en la base de datos de forma correcta se pasa al estado “Conectado”. En caso de error se vuelve al estado “No conectado”. Además, puede salirse de la aplicación en cualquier momento a través del menú haciendo click en File → Exit.

El siguiente diagrama ejemplifica la comunicación que se produce entre ambos sistemas:



3.2 Cliente Android – servidor directorio.-

De la misma forma que en el caso anterior, la comunicación entre el cliente Android y el servidor directorio se produce mediante peticiones GET HTTPS que realiza el primero al segundo. En este caso, la petición se realiza al programa `getServerIP.php` con el fin de obtener la dirección IP del servidor de streaming, que debe de haber sido almacenada anteriormente.

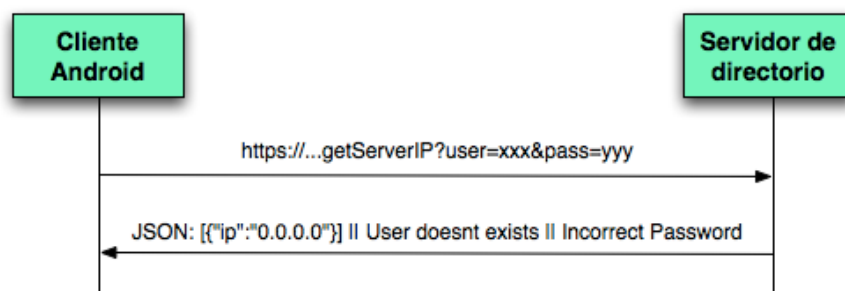
El diagrama de estado de la comunicación es el siguiente:



Al iniciar la aplicación se accede al estado “Login”. Tras introducir los datos oportunos se realiza la petición de para obtener la IP del servidor de streaming, pasando al estado “GestiónConexión”. Si se obtiene una respuesta satisfactoria se pasa al estado “Conectado”. En caso contrario se vuelve al estado anterior.

Para finalizar la ejecución del programa hay que dejar de visualizarlo pulsando el botón con el dibujo de la casa del móvil. Posteriormente será el Android el que se encargue de cerrar la aplicación y liberar los recursos cuando se necesiten éstos. Éste es el método habitual que utiliza Android para finalizar los procesos y programas abiertos.

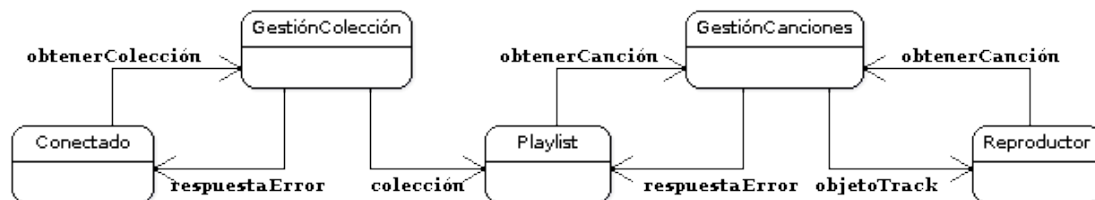
El siguiente diagrama ejemplifica la comunicación que se produce entre ambos sistemas:



3.3 Servidor streaming – cliente Android.-

La comunicación entre servidor de streaming y cliente Android se produce mediante dos métodos. El primero de ellos se basa en la creación de un socket TCP en el puerto 4444 por parte del servidor de streaming al que el cliente Android realiza peticiones enviando mensajes de texto.

La comunicación realizada puede representarse a través del siguiente diagrama de estado:

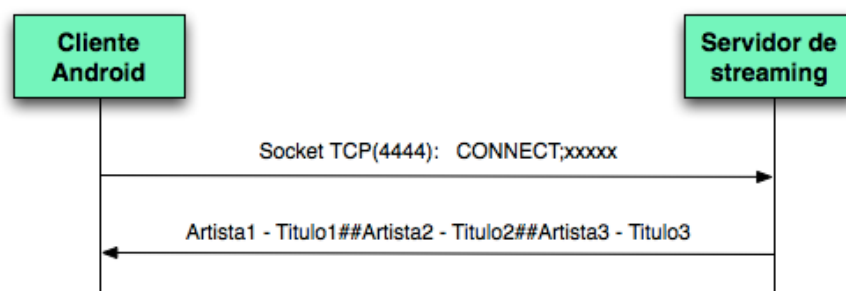


Partiendo de que la aplicación Android se encuentra en estado “Conectado”, algo que ocurre tras realizar la conexión con el servidor de directorio, se pasa al estado “GestiónColección” al solicitar el cliente Android al PC la colección de canciones. En caso de error se vuelve al estado “Conectado” mientras que en caso de éxito se pasa al estado “Playlist”.

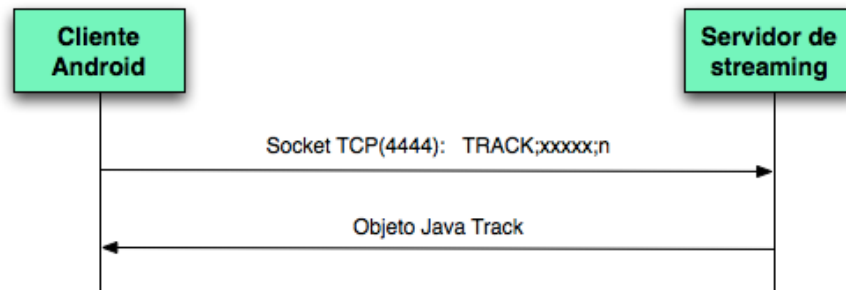
De la misma forma, al realizar una petición de una canción para ser reproducida en el cliente Android, se pasa al estado “GestiónCanciones”. Si la canción se obtiene satisfactoriamente se pasa al estado “Reproductor”. Una vez en este estado se pueden realizar más peticiones de canciones, pasando momentáneamente de nuevo al estado “GestiónCanciones”.

Los mensajes que puede enviar el cliente Android a través del socket tcp al servidor son dos:

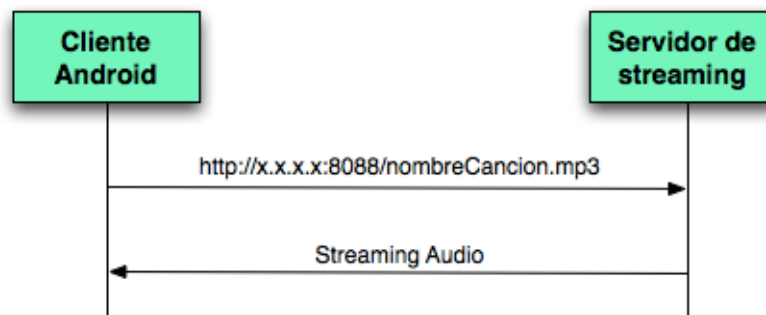
- **Mensaje de conexión:** el mensaje tiene la estructura **CONNECT;password** donde password es la contraseña del usuario. En caso de que la contraseña sea la correcta el servidor de streaming responde enviando en un string la lista de canciones, separadas cada una por el símbolo ##.



- **Obtención de canción:** el mensaje tiene la estructura **TRACK;password;NúmeroCanción** donde NúmeroCanción es la posición en la colección de la canción que se desea obtener. El servidor responde en este caso enviando el objeto Track correspondiente. Es por ello que la clase Track es serializable.



El segundo de los métodos de comunicación utilizados es el protocolo HTTP. En este caso, el servidor de streaming sirve los archivos mp3 a partir de un servidor web que crea en el puerto 8088. Por tanto, para reproducir las canciones, el cliente Android realiza peticiones GET a dicho servidor gracias a la función de reproducción de archivos que se descargan progresivamente de que dispone la clase MediaPlayer que incluye Android.



4. Demostración del funcionamiento.-

A continuación se va a mostrar paso por paso un ejemplo de uso del sistema para reproducir las canciones almacenadas en el ordenador del usuario en el móvil Android. Los pasos a realizar son los siguientes:

1. Para poder hacer uso del sistema, el usuario debe de crear una cuenta a través de la web del proyecto insertando los datos que se piden:



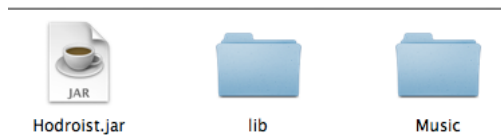
2. Una vez introducidos los datos se obtiene lo siguiente:



3. Cuando el registro se produce de forma exitosa, los datos introducidos se almacenan en la base de datos de la siguiente forma:

			id	username	password	email	server_ip
<input type="checkbox"/>			16	usuario1	122b738600a0f74f7c331c0ef59bc34c	usuario1@ugr.es	
<input type="checkbox"/>			17	usuario2	2fb6c8d2f3842a5ceaa9bf320e649ff0	usuario2@ugr.es	
<input type="checkbox"/>			18	usuario3	5a54c609c08a0ab3f7f8eef1365bfda6	usuario3@ugr.es	
<input type="checkbox"/>			19	rolimat	4d186321c1a7f0f354b297e8914ab240	rolimat@correo.ugr.es	
<input type="checkbox"/>			20	usuario4	0ddd0fbf933b170eb6d90987a67d0a5d	usuario4@ugr.es	
<input type="checkbox"/>			21	usuario5	0b65933df3421cf1bdf4ff082ffc8901	usuario5@ugr.es	

4. A continuación, el usuario deberá de crear una carpeta llamada **Music** en el directorio desde el que se ejecutará el archivo ejecutable .jar de la aplicación para el ordenador:

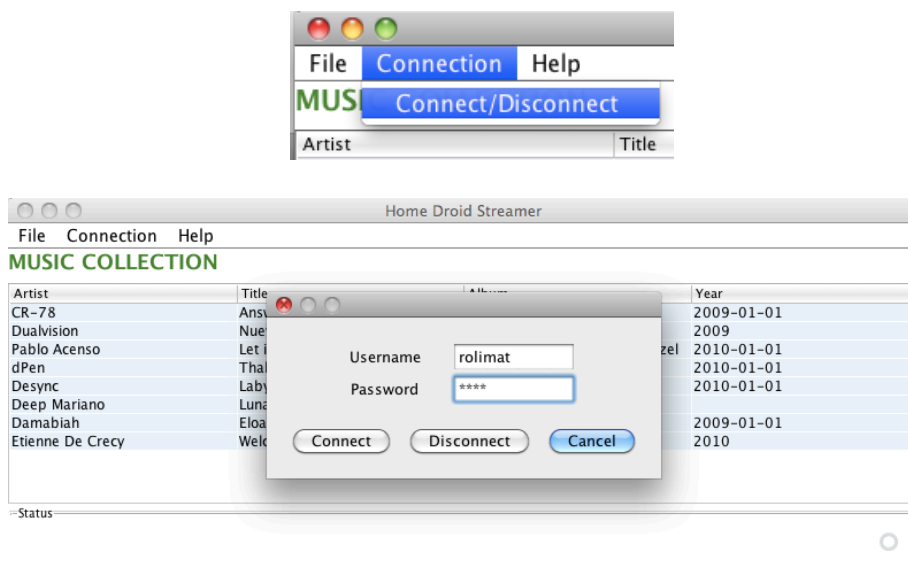


- Posteriormente se ejecuta la aplicación en el PC mediante el comando **java -jar Hodroist.jar**, abriéndose la interfaz del programa:



Como puede verse, al abrir la aplicación, se escanea la carpeta Music y se muestran todas las canciones disponibles en la interfaz del programa.

- Lo siguiente será realizar la conexión de la aplicación con la base de datos, enviando la dirección IP del ordenador para ser almacenada:

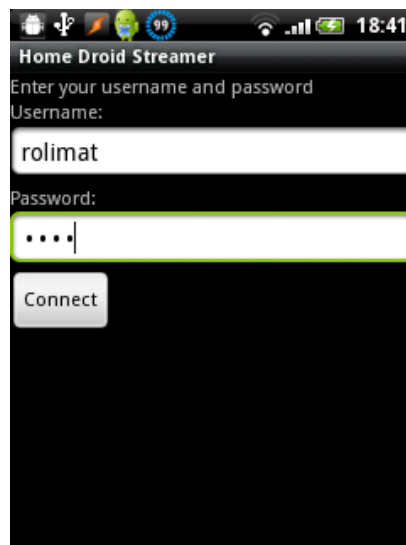


Al pulsar el botón “Connect” la dirección IP del ordenador se envía a la base de datos. Posteriormente se crea el socket TCP para escuchar las peticiones del cliente Android y se inicia el servidor web.

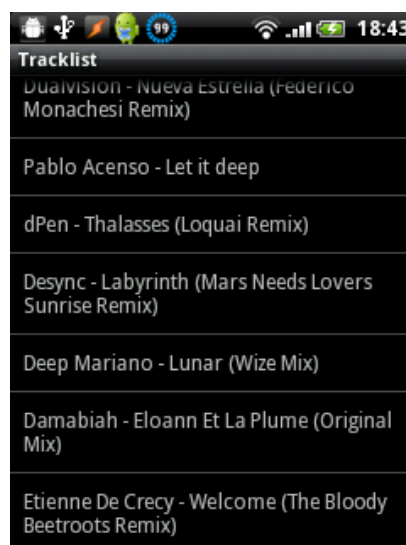
La base de datos tendrá ahora el siguiente aspecto:

		id	username	password	email	server_ip
<input type="checkbox"/>			16	usuario1	122b738600a0f74f7c331c0ef59bc34c	usuario1@ugr.es
<input type="checkbox"/>			17	usuario2	2fb6c8d2f3842a5ceaa9bf320e649ff0	usuario2@ugr.es
<input type="checkbox"/>			18	usuario3	5a54c609c08a0ab3f7f8eef1365bfda6	usuario3@ugr.es
<input type="checkbox"/>			19	rolimat	4d186321c1a7f0f354b297e8914ab240	rolimat@correo.ugr.es 192.168.1.66
<input type="checkbox"/>			20	usuario4	0ddd0bf933b170eb6d90987a67d0a5d	usuario4@ugr.es
<input type="checkbox"/>			21	usuario5	0b65933df3421cf1bdf4ff082ffc8901	usuario5@ugr.es

7. Tras instalar la aplicación cliente Android en el teléfono, se ejecuta la misma. En la pantalla que se muestra han de introducirse el nombre de usuario y la contraseña:



8. Una vez que se verifican el nombre de usuario y contraseña se obtiene del servidor de directorio la dirección IP del servidor de streaming. Posteriormente se realiza una petición al servidor de streaming para obtener la lista de canciones. Dicha lista se muestra en la pantalla que se abre:



9. En la pantalla anterior se puede hacer click sobre cualquiera de las canciones que se quiera escuchar, abriéndose el reproductor:



10. En la pantalla del reproductor se puede navegar por las diferentes canciones haciendo uso de los botones de la parte inferior. Al pulsar el botón de la izquierda se vuelve a la pantalla "Tracklist" en la que se muestra la lista de canciones disponibles.

5. Librerías utilizadas.-

Para la implementación de las aplicaciones ha sido necesario utilizar las siguientes librerías:

Servidor de streaming:

- Apache HttpClient: permite realizar peticiones de tipo http. Está compuesta a su vez de varias librerías de las que depende.
- JAudioTagger: permite la lectura de los tags ID3 de los archivos mp3 así como de la imagen asociada (artwork) a los mismos.
- SimpleWebServer: ejecuta un servidor web a partir de la creación de un objeto.

Cliente Android:

- Json-simple: permite la lectura y extracción de datos incluidos en el formato JSON.