MusicShare

Proyecto Final – Sistemas Multimedia

[Seleccione la fecha]

[Escriba el nombre de la compañía]

Jesús Cuenca López

**Contenido**

[1. Objetivos de la Aplicación 3](#_Toc358584889)

[2. Diseño General 4](#_Toc358584890)

[3. Diseño del Lado Cliente 5](#_Toc358584891)

[4. Diseño del Lado Servidor 10](#_Toc358584892)

[5. Diseño de las Conexiones 12](#_Toc358584893)

[6. Problemas y Dificultades encontrados 14](#_Toc358584894)

[7. Conclusión 15](#_Toc358584895)

# Objetivos de la Aplicación

Ante la libertad que se nos ha dado en este proyecto, y sabiendo que debíamos hacer algo relacionado con los sistemas Multimedia, hemos decidido desarrollar una aplicación que permita la retransmisión de música entre equipos conectados a la misma red local.

De esta idea ha surgido nuestra aplicación, **MusicShare**, que permitirá al Cliente navegar por los archivos del Servidor, escuchar las canciones que tenga almacenadas el mismo, y hasta crear sus propias listas de reproducción con la música del Servidor.

Antes de empezar a diseñar la aplicación, la idea era poder transmitir música entre un PC a un dispositivo móvil (inicialmente, Android) pero, una vez que empezamos a crear el Servidor, decidimos que finalmente sería una aplicación para PC debido a la incompatibilidad entre la librería de audio usada y el sistema operativo móvil. Aun así, no descartamos la idea de poder rediseñarlo, algún día, para dispositivos móviles.

# Diseño General

Por su propia naturaleza, nuestra aplicación necesita dos agentes ejecutándose cooperativamente: el lado Servidor, que mandará la información de la música y los datos que puede compartir, así como también enviará la propia música; y el lado Cliente, que recibirá dicha información, y la representará de forma adecuada y fácil de usar para el usuario, y reproducirá la música que le llegue del servidor.

Así, en principio, nuestros agentes estarán constituidos por una interfaz fácil y cómoda de utilizar para el usuario, una interfaz común entre ambos para que puedan comunicarse entre ellos, y una forma de retransmitir y recibir la música que el usuario pida.

Tal y como está diseñada la aplicación, el Cliente necesitará la dirección IP del Servidor para realizar la conexión. Sin embargo, esta forma de conectarse no sería demasiado cómoda por el usuario, ya que debería estar cerca del Servidor para comprobar cuál es su dirección IP para después introducirla en el Cliente. Por ello, el diseño incluye también una forma de conectar Cliente y Servidor de forma automática, a través de una inundación UDP.

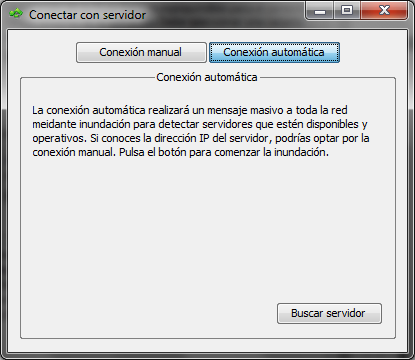
# Diseño del Lado Cliente

El Cliente será el agente con el que esté más relacionado el usuario, ya que será la piedra angular con la que éste interactuará. Por ello, la interfaz debe ser lo más simple y cómoda posible.

Lo primero que deberá hacer el usuario será conectarse con el servidor. Para ello, la aplicación cuenta con dos posibilidades:



La conexión manual permite al usuario conectarse al Servidor a través de la dirección IP del mismo. Es la forma más directa y más segura (sólo aquellos que conozcan su dirección IP podrán conectarse), sin embargo, no es la más cómoda, ya que no todos los usuarios conocerán o sabrán cómo obtener la dirección IP de su equipo. Por ello se ha creado la segunda opción:

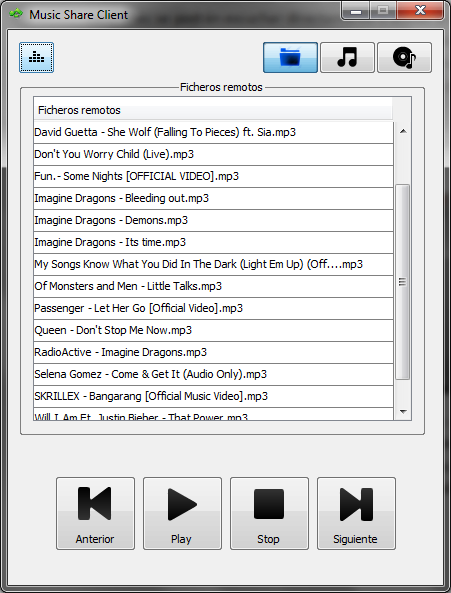


Con la Conexión Automática o Directa, el usuario podrá conectarse al Servidor con tan solo pulsar un botón: será la apliación Cliente la que se encargará de realizar una inundación UDP por la red local para encontrar Servidores disponibles escuchando en la misma. En cuanto un Servidor responda, el Cliente podrá iniciar la conexión, y mostrar al usuario la interfaz principal. Si en treinta segundos no se ha obtenido respuesta de ningún Servidor, se dará la búsqueda como fallida, y se avisará al usuario de que no existen Servidores disponibles.

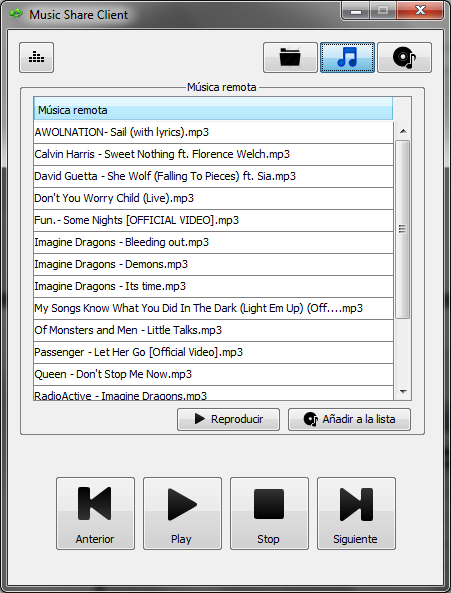
Por otro lado, la interfaz principal de la aplicación está basada en nuestros objetivos para la aplicación, ésta tendrá tres partes bien diferenciadas:

* **Exploración** por los archivos del Servidor, desde donde solamente se podrán reproducir o añadir a la lista de reproducción las canciones que se encuentren, además de navegar por las carpetas a las que el Servidor tenga permiso.
* **Lista de música del Servidor**, en la que aparecerán todas las canciones encontradas recursivamente desde la raíz establecida por el Servidor. Estas canciones se podrán escuchar directamente o añadir a la lista de reproducción, y es una forma fácil de encontrar directamente la música.
* **Lista de reproducción**, desde donde se ejecuta la funcionalidad básica del Servidor, y que permitirá al usuario mantener una lista de canciones preferidas, por ejemplo.

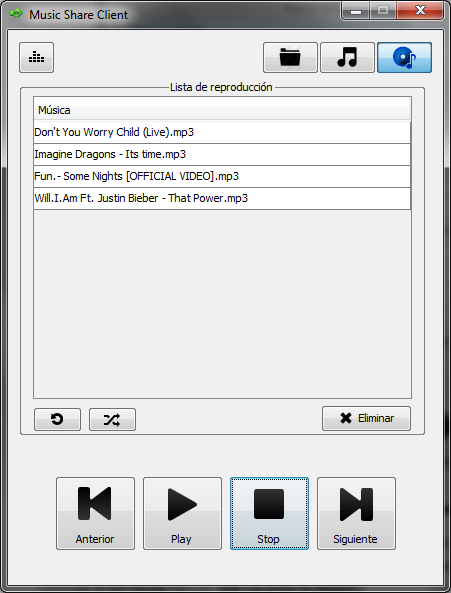
Atendiendo a estas tres opciones, la interfaz de la aplicación Cliente tendrá tres pestañas en las que se podrá conseguir cada una de ellas:



La pestaña “Archivos” permitirá el primero de los objetivos: que el usuario navegue por los directorios escogidos por el Servidor en la interfaz del Cliente. Así, se podrá entrar en carpetas y reproducir la música encontrada en ellos, hasta el nivel máximo permitido por el Servidor (para que no tengamos acceso a todo el sistema de archivos del mismo).



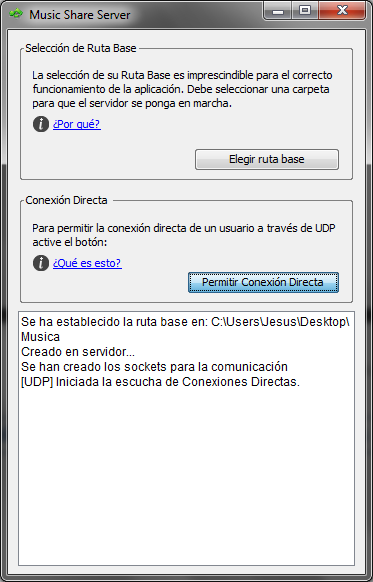
En la segunda ventana, “Música”, el usuario podrá visualizar toda la música a la que el Servidor nos da acceso, de forma que escoger las canciones a escuchar será más cómoda si no recuerdas en qué carpeta la dejaste.



Y, por último, la pestaña de “Lista de reproducción” presentará al usuario su lista actual de música que tiene almacenada. La mayor parte de la aplicación trabajará con dicha lista de reproducción, ya que los botones inferiores funcionan sobre ella y, en el lado Servidor, por simplicidad, hemos considerado que no se puede reproducir una canción si no está añadida a la lista.

# Diseño del Lado Servidor

El servidor trabajará cooperativamente con el cliente, pero su interfaz gráfica es mucho más simple, ya que las acciones que el usuario podrá realizar sobre el Servidor son más reducidas que con el cliente:



En el Servidor, el usuario sólo podrá modificar cuestiones sobre seguridad básica de la aplicación: la selección de la ruta base del Servidor, así como permitir las conexiones directas o no.

El Servidor no se pondrá en marcha hasta que el usuario no **seleccione una ruta base** para la aplicación. Esta ruta será el directorio a partir del cual el Cliente tendrá acceso dentro de la aplicación. Por ejemplo, si se selecciona como ruta base la raíz de directorios del sistema, el Cliente podrá visualizar todo el sistema de archivos de la máquina servidora. De esta forma, el Servidor puede decidir a qué carpetas tendrá acceso el Cliente antes de que este pueda conectarse. Se puede seleccionar como ruta base la carpeta en la que el usuario guarda toda su música, o un directorio en el que no tenga archivos privados o que no desea que el Cliente vea (aunque el Cliente no podrá visualizar el contenido de dichos ficheros, a no ser que sean archivos de audio).

El Servidor también podrá permitir o no el uso de la conexión directa del Cliente. La conexión directa permitirá al Cliente conectarse al Servidor sin conocer la dirección IP del mismo, utilizando una inundación UDP a través de la red local para conseguir la conexión. Si el Servidor no habilita la conexión directa, el Cliente no podrá conectarse a través de este método, y tendrá que introducir su dirección IP para poder iniciar la aplicación.

# Diseño de las Conexiones

Nuestra aplicación está basada en las conexiones establecidas entre Cliente y Servidor a lo largo de su ejecución. En total, MusicShare mantiene tres conexiones TCP más la UDP utilizada para la inundación que el Cliente realiza para encontrar al Servidor.

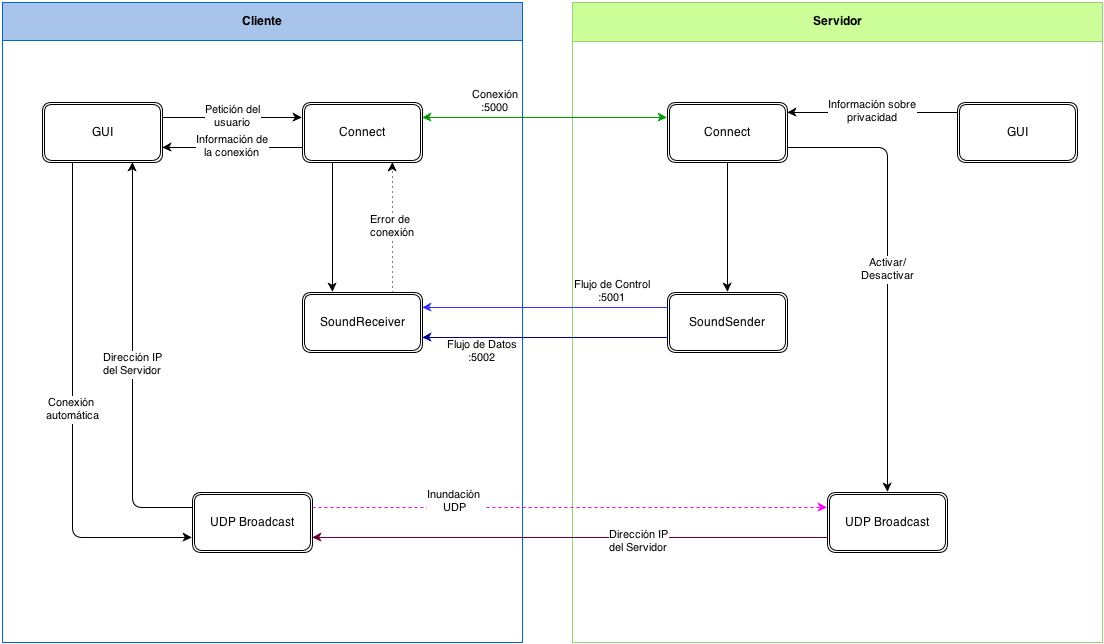
Estas conexiones están resumidas en el siguiente diagrama, en el que se pueden ver las cases que interaccionan entre sí tanto en el lado Cliente como en el lado Servidor.

Como se puede ver, es la clase Connect la que inicia y mantiene la conexión principal, por la que se transmiten las acciones que el usuario realiza en el Cliente al Servidor para que éste realice las operaciones pertinentes.

Por otro lado, las clases SoundSender y SoundReceiver son las encargadas de enviar y recibir, respectivamente, tanto los datos en crudo de la canción, como la información de control (tamaño del archivo, tamaño de los segmentos,…) de la misma.

Por su parte, a través de la interfaz del Cliente, el usuario podrá iniciar la inundación UDP para realizar la búsqueda del Servidor en la red local, a la que éste contestará si así lo ha decidido el usuario en la propia interfaz del Servidor.

Debido de que las conexiones que mantienen SoundServer y SoundReceiver son TCP, éste último podrá informar en cualquier momento al usuario de que la conexión se ha caído a lo largo de la ejecución de la aplicación, en caso de que haya algún problema en la misma o en el Servidor.



# Problemas y Dificultades encontrados

# Conclusión

Durante la implementación de este proyecto hemos tenido que pelearnos con datos de archivos de música en crudo, solventar algunos problemas entre conexiones, tanto TCP como UDP, e implementar la forma de navegar por los archivos del disco duro, tratarlos y enviarlos. Sin embargo, ver cómo tu aplicación reproduce música de forma remota es una sensación muy satisfactoria que compensa las dificultades encontradas.

En conclusión, este proyecto ha resultado ser una forma bastante gratificante de poner en práctica los conocimientos adquiridos en clase, a pesar de que nos ha dado algún que otro quebradero de cabeza.