



Laboratorio de Sistemas Operativos y Redes



Proyecto: mStream

Alumnos: Rosato Jimena, Sanchez Dario

Introducción

Mstream¹ es un servidor personal de música en streaming. Se puede usar para emitir y difundir una biblioteca musical personal y ser escuchada en cualquier dispositivo en cualquier lugar.

- Éste tiene soporte web, de escritorio y mobile, con distintos alcances en cada plataforma.

Instalación

Aclaración: Nosotros ya contabamos con node/npm en nuestras máquinas, pero de no contar con esta herramienta, se deberá instalarla para usar la aplicación. Mas info: <https://nodejs.org/>

Hay varias formas de instalación posibles. Descargando los binarios, por medio de docker, o por la que optamos en este informe, que fue descargar el código fuente desde github.

Clonamos el repositorio:

```
$ git clone https://github.com/IrosTheBeggarmStream.git
```

Ingresamos a la carpeta que se creó:

```
$ cd mStream
```

Dentro de la carpeta ejecutamos:

```
$ npm install
```

¹ <https://www.mstream.io/>

Esto es así ya que el autor utiliza npm como sistema de gestión de paquetes².

Luego para facilitar el uso del programa se ejecuta

\$ sudo npm link

Este comandol crea un link simbólico es la carpeta global, así puede hacerse llamado a mStream desde cualquier directorio, también puede omitirse este paso y ejecutar **npm start** desde la carpeta contenedora, pero es mucho más sencillo poder acceder en cualquier momento a mstream con el comando

\$ mstream

La ventaja de mStream es que no tiene ninguna dependencia y ya con esto podemos hacer uso de todas sus funcionalidades.

Alternativas de instalación:

Docker:

<https://github.com/linuxserver/docker-mstream>

Binarios (vienen con una interfaz gráfica y no hay que hacer configuraciones en la consola):

<https://github.com/IrosTheBegggar/mStream/releases/tag/v4.2.1>

Se descarga la versión de acuerdo al sistema operativo. En el caso de linux, una vez descargado y descomprimido, se ejecuta el archivo mStreamExpress desde una consola con:

\$./mStreamExpress

² <https://www.npmjs.com/>

Uso

Por defecto mStream se levanta en el puerto 3000 del localhost. Esto puede ser cambiado mediante algunas configuraciones. Además de otras características que pueden ser cambiadas.

Las configuraciones de mStream están dadas por un archivo json, que en caso de que no exista, se puede crear en el momento. Este puede hacerse a mano pero mStream ofrece un configurador que corre por terminal al cual podemos acceder con el comando:

```
$ mstream --wizard "archivoDeConfiguracion.json"
```

Hay ciertas configuraciones que no son accesibles por medio del wizard y tienen que ser escritas dentro del archivo json, pero aun así es muy cómodo configurarlo.

Algunos argumentos de configuración son:

- **port**: Define el puerto donde se va a levantar mStream, por defecto lo hará en el 3000 si no está definido.
- **writeLogs**: Booleano que determina si se van a escribir logs o no.
- **scanOptions** (opciones ligadas al escaneo de canciones y sincronización):
 - **scanInterval**: Determina cada cuanto se escanean los archivos en busca de nuevas canciones, se representa en horas.

- **saveInterval**: El intervalo de tiempo en el que se sincroniza la base de datos con los nuevos archivos escaneados.
- **folders** (opciones ligadas a la organización de archivos en carpetas):
 - **nombreDeCarpeta**: Determina un nombre de carpeta para el sistema y se le provee de una ruta en los archivos locales, ejemplo:


```
“folders”:{
    “Queen”: “media/music/Queen”
  }
```
- **users** (Creacion de usuarios con música personalizada):
 - **nombreDeUsuario**: Se determina un nombre de usuario y se le provee una password y opcionalmente unas carpetas para el mismo, de agregarse un usuario la función de stream público cambia y se requiere iniciar sesión si o si. Ejemplos:


```
“users”: {
  “dario”: {
    “password”: ”ponganPersianasEnLa331”
    “vpaths”: “Queen”(Del ejemplo de carpetas)}
  }
}
```

Ahora un ejemplo de usuario con varias carpetas:

```
“usuarioDesconocido”{
  “password”: “somePassword”,
  “vpaths”: [“music”, “themes”](varias carpetas musicales)
}
```
- **ssl**: Soporte ssl, solo hay que poner la key y la certificación en las variables “key” y “cert”.
- **lastFM**: Pueden agregarse credenciales de lastFM a los usuarios, solo hay que agregar el usuario y password de

lastFM en los atributos lastfm-user y lastfm-password en la porción de json del usuario en cuestión

- **storage**: Se pueden definir los directorios en los que se guardan el arte de los álbums, la base de datos y los logs con las variables “albumArtDirectory”, “dbDirectory” y “logsDirectory” respectivamente.

Dejamos un json de ejemplo, este es el que usamos nosotros:

```
{
"scanOptions": {
  "pause": 0,
  "scanInterval": 24,
  "bootScanDelay": 3
},
"secret":
"83e24ff7220ed69d55788388f580098db908729dfedec38043fe
c68b3deee7f7242e8475996c28ea9857d03172a28d8c",
"users": {},
"folders": {
"musica": {
  "root": "/home/Dario/Música"
}
},
"tunnel": true,
"storage": {},
"port": 3000
}
```

Para levantar el mstream con la configuración establecida hay que agregarle la opción -j:

mstream -j archivoDeConfiguracion.json

Una vez levantado el servidor de música se puede ver un explorador de archivos en el que, si hay canciones, podremos encolarlas en una lista de reproducción rápida.

Esta misma se puede compartir a través de un link con un periodo de vida en días.

Podemos elegir descargar la lista de reproducción y que se guarde en nuestro dispositivo como también elegir crear una lista de reproducción que queda guardada en la sesión del servidor. De esta manera podemos crear algunas listas para compartir en el momento u otras que además de compartirlas serán alojadas en el servidor permanentemente.

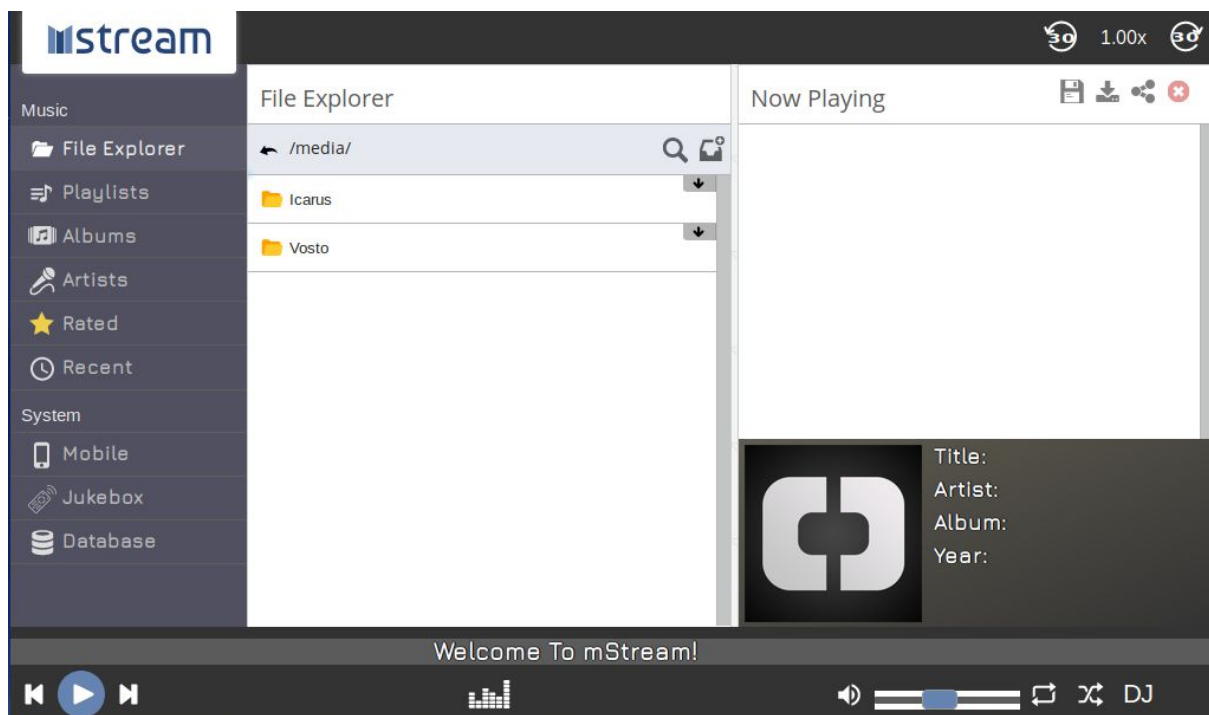
Por último también podemos elegir limpiar la lista de reproducción, y si esta está alojada en el servidor podremos borrarla del mismo.

El reproductor de música por su parte cuenta con las opciones básicas esperadas, play/pausa, información de la canción con título, artista y álbum, siguiente canción o anterior, controlador de volumen, repetir canción, aleatorio y un indicador numérico que se titula “playback”. (Creemos que sirve para modificar la velocidad de reproducción de la música, pero por lo visto no funciona ya que no hay efecto alguno al querer modificarlo.)

Como dato curioso se pueden agregar las mismas canciones las veces que se quiera a una lista, pudiendo tener varias veces una misma canción en una sola lista de reproducción. También cuenta con una opción DJ que selecciona canciones

aleatorias y crea una lista de reproducción con ellas, y como adorno hay un visualizador gráfico, el cual podemos seleccionar para que reproduzca patrones de colores mientras suena la música.

Podemos encontrar en el lado izquierdo un menú de selección en el que se encuentran los siguientes elementos:



- **File explorer:** Podremos indagar en la estructura del servidor en busca de carpetas y canciones, pudiendo elegir de a una cuales queremos escuchar, creando así una lista de reproducción. Cuenta además con un botón de agregado rápido, el cual agrega todas las canciones presentes en ese directorio a una lista. De igual manera podremos eliminar cualquiera de estas canciones del servidor.
- **Playlists:** Aquí se encuentran las listas de reproducción creadas por el usuario y que son guardadas en el servidor

. Estas se pueden borrar y compartir con unos botones presentes en la lista de las mismas.

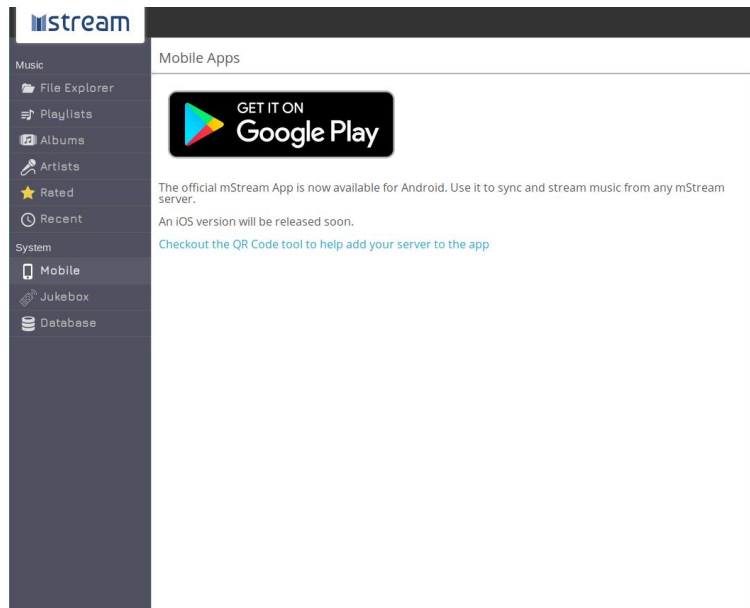
Si queremos agregar canciones a la misma, se la puede elegir para reproducir y ahí seleccionar canciones del administrador de archivos, de esta manera se agregarán.

- **Albums:** mStream escanea los archivos como vimos previamente y de encontrar información de álbums, se crea un álbum del artista en el que se encuentran todas las canciones del mismo, que estén en el servidor.
- **Artist:** De igual manera el servidor agrupa las canciones de un mismo artista, si es que hay alguna canción que le provea esta información.
- **Rated:** mStream cuenta también con una función de valoración y en esta sección encontramos las canciones valoradas de mayor a menor rating. Esta calificación se da en forma de estrellas con un máximo de 5, pudiendo elegir calificaciones intermedias como por ejemplo 4.5.
- **Recent:** Aquí se encuentran las últimas canciones agregadas al servidor. Una forma muy útil de mostrar novedades en el mismo, nos permite elegir qué cantidad de canciones queremos ver, por ejemplo las últimas 10, 50 o 100 canciones.

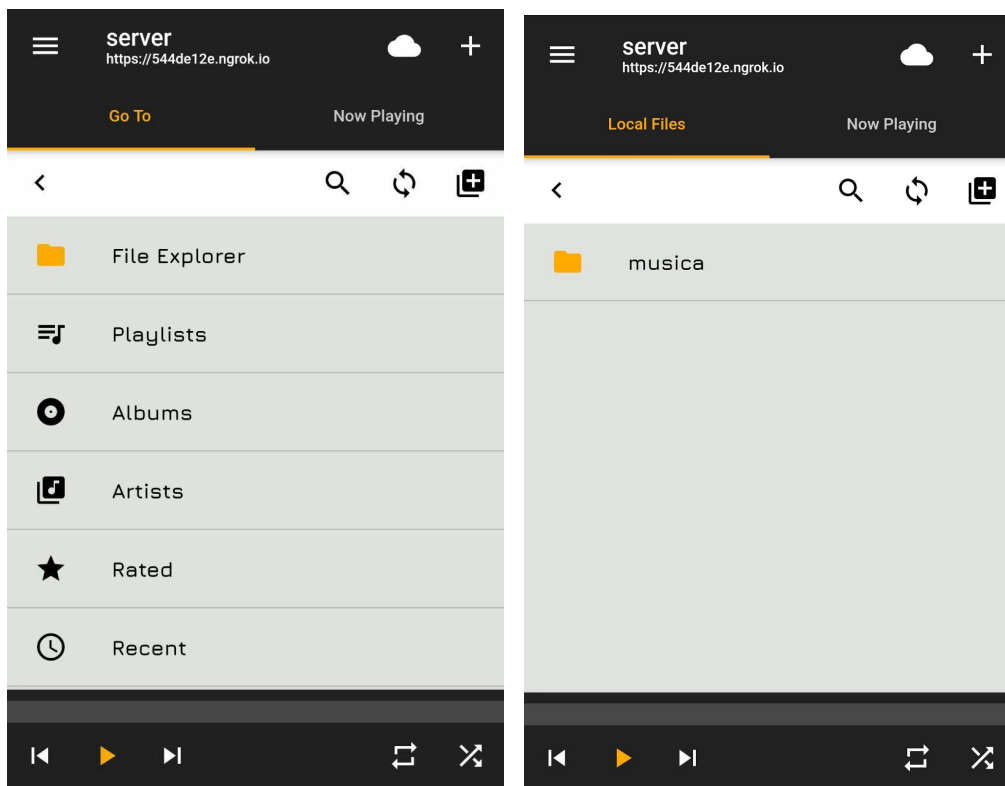
Luego encontramos un grupo de 3 opciones con el título de system, que nos permiten 3 utilidades muy interesantes:

- **Mobile:** Es un apartado que nos informa de la existencia de la app en la playstore de google, además de esto nos permite crear un código QR con el que difundiremos nuestro servidor, una vez escaneado en la aplicación

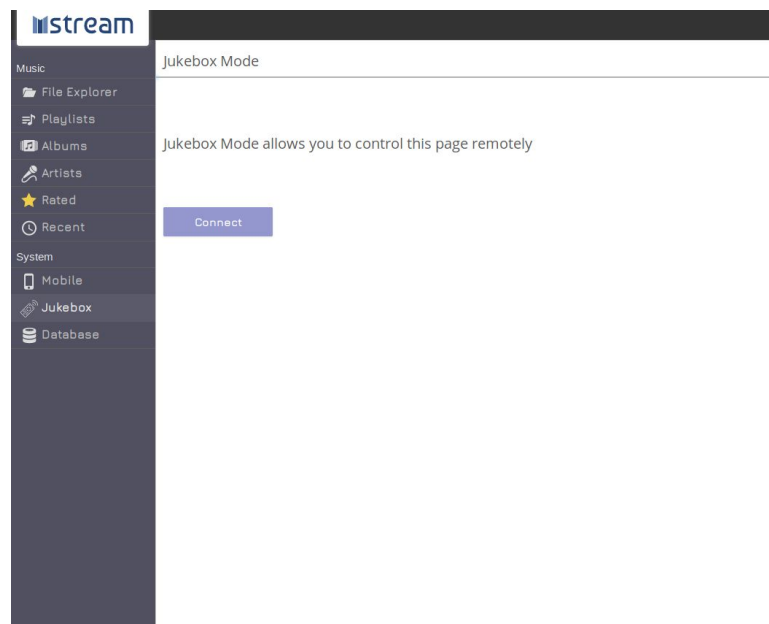
tendremos acceso a todo el catálogo del servidor en nuestro celular.

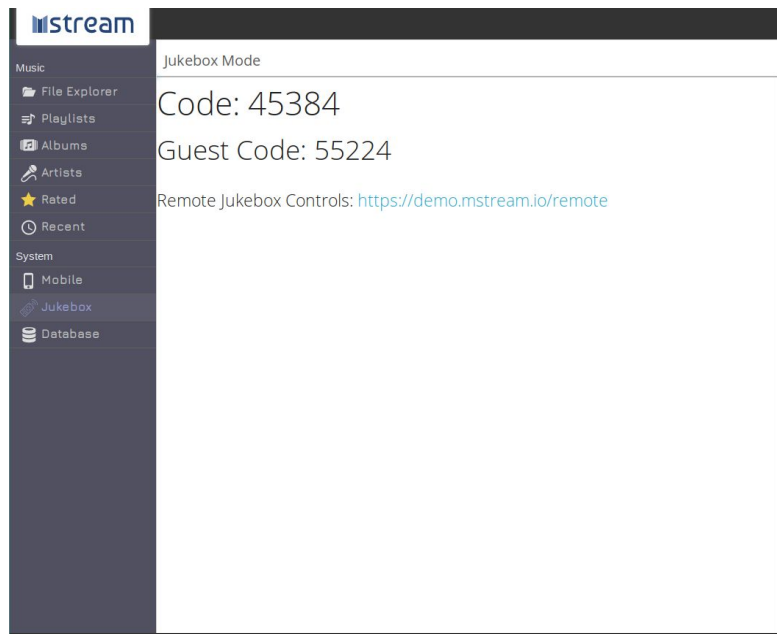


Versión mobile para Android

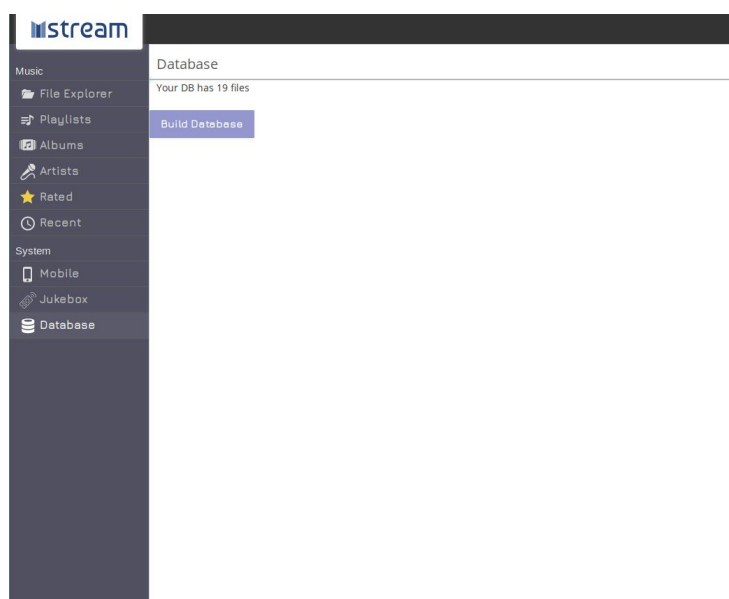


- **Jukebox:** Aquí se nos permite generar un código de identificación y de validación el cual podremos introducir en cualquier dispositivo, esto hará que el servidor se convierta en una especie de gramola. Es decir, desde los dispositivos conectados se podrán elegir canciones en el servidor para que este las reproduzca en una lista de reproducción propia del servidor.





- **Database:** Aquí podremos elegir la opción de sincronizar manualmente la base de datos del servidor, permitiendo que cambios como creación de listas de reproducción, lectura de nuevos archivos, generación de albums o calificaciones nuevas, entre otros, sean introducidos en el momento a la información del servidor y tener así un repertorio actualizado para los clientes del servidor.



Problemas

No nos encontramos con ningún problema durante la instalación ni en el uso de la aplicación.

Nos llamo la atención el botón para cambiar la velocidad de reproducción, que se mostraba, pero parecía que no estar implementado todavía.

Y quizás el único 'inconveniente' a destacar es que el archivo ejecutable de la versión express, no se ejecuta al hacerle doble click. Sino que hay que ejecutarlo por terminal con el comando **./mStreamExpress** (esto no está explicado en la página de github, y podría requerir una búsqueda adicional de información para algunos usuarios)

Conclusiones

A pesar de que el proyecto está en una etapa beta, es una aplicación muy interesante.

Quizás se podría facilitar un tutorial de instalación, configuración y uso para el usuario común.

Si el usuario eligiese pagar por el servidor que los desarrolladores ofrecen, ya tiene asegurado su propio servicio de streaming de música, y totalmente configurable por el mismo.

Consideramos que a futuro, y con más desarrollo va a aumentar su potencial, agregando más características y mejorando las que ahora ya tiene.