

# UNIDAD 7 – AUDIO STREAMING

## Índice:

0. Introducción:	2
PROCEDIMIENTO	3
Listado de servidores de streaming	3
1. Realiza un análisis previo del software servidor elegido (plataformas, última versión, libre/proprietaria, funcionalidad, plugins ...),	4
Plataformas:	4
- Windows:	4
- Linux:	5
- macOS	5
Última versión:	6
Licencia:	6
Funcionalidades:	6
Plugins:	6
2. Instalación y configuración del servidor asignado para abordar el problema planteado.	7
Configuración de AWS:	7
Ampache:	8
Introducir música en Ampache:	13
Usuarios	19
Azuracast:	20
Introducir canciones en AzureCast	24
Añadir "Radio Station" a Ampache:	28
3. Instalación y configuración de clientes específicos (No navegador, apps, apks, etc).	29
S.O. de escritorio:	29
Android	31
4. Analiza la posibilidad de difusión de la Radio del instituto desde la plataforma: "Shoutcast" o de cualquier otra plataforma. Implementad una prueba simulada, analizando las posibilidades de conexión de los clientes, costes derivados, etc...	36

## 0. Introducción:

### Instala y configura un Servidor de Streaming de Audio

SRI P1 - Unidad de Trabajo 7

#### SERVICIOS DE RED E INTERNET

Tomando como orientación el esquema de laboratorio virtual propuesto, implementar uno o varios servicios de streaming de audio para el subdominio

**2asirXX.institutodh.net** (en mi caso, usaré **im07.institutodh.net**) que trate de resolver las siguientes cuestiones:

1. Sirva de soporte a un **proyecto de Radio** de los alumnos de 2ºASIR de forma que el alumnado cree los distintos programas en formato .MP3 y estén disponible para su reproducción para toda la comunidad escolar. (AzuraCast, [LibreTime](#), Zeno Tools, ...)
  2. Se creará **una sección musical/de audio abierta** también a la comunidad educativa, los profes (prof1 y prof2) podrán realizar aportes musicales/de audio. Se procurará ofrecer canciones sueltas y estudiar la posibilidad de ofrecer playlists.
  3. De otra parte, el administrador ofrecerá una **biblioteca de audiolibros** en MP3 para los usuarios registrados previamente.
- Intentaremos que los clientes puedan acceder desde:
    - El navegador de sus equipos.
    - Desde la aplicación VLC.
    - Desde cualquier otro software cliente de tu Móvil/Tablet.

## PROCEDIMIENTO

Realizar:

1. Realiza un análisis previo del software servidor elegido (plataformas, última versión, libre/propietaria, funcionalidad, plugins ...),
2. Instalación y configuración del servidor asignado para abordar el problema planteado.
3. Instalación y configuración de clientes específicos (No navegador, apps, apks, etc).
4. Analiza la posibilidad de difusión de la Radio del instituto desde la plataforma: "Shoutcast" o de cualquier otra plataforma. Implementad una prueba simulada, analizando las posibilidades de conexión de los clientes, costes derivados, etc...
5. Prepara una presentación y una demo en directo.
6. Se valorará el uso de **DNS**, **docker**, **certbot** y **AWS/Azure** para el despliegue del servicio de streaming.

Listado de servidores de streaming de audio propuestos: Puedes elegirlo también de entre los propuestos en el siguiente enlace:

<https://selfhosted.libhunt.com/airsonic-alternatives> (Koel, Modipy, mStream, AzuraCast, Airsonic, Airtime etc...)

- Opción 1.- Servidor **Ampache**
- Opción 2.- Servidor **FunkWhale**
- Opción 3.- **Navidrome + Zeno Tools**
- Opción 4. Servidor **Koel Community**
- Opción 5. **mStream**
- Opción 6. Servidor **AzuraCast** (sólo streaming)
- Opción 7. Servidor **Icecast** (sólo streaming) / cliente IceS2

Fuentes de documentación recomendadas:

1. La web oficial del desarrollador de cada servidor.
2. La web del Instituto Abierto de Cataluña: [Actividades streaming audio](#)

1. Realiza un análisis previo del software servidor elegido (plataformas, última versión, libre/propietaria, funcionalidad, plugins ...),

## Ampache:

### Plataformas:

#### - Windows:

“Windows Installation Guide”<sup>1</sup>:

Para instalar Ampache en Windows necesitas un entorno **WAMP** (Apache, MySQL y PHP), preferiblemente mediante **Bitnami**. También requiere **Composer** para gestionar dependencias, **Git** para descargar los archivos y un navegador para completar la configuración inicial y la base de datos.

Windows Server IIS (Server 2008R2)<sup>2</sup>: Esto es un Server basado en Windows 7, completamente obsoleto.

Para Windows Server necesitas **IIS** con el módulo **URL Rewrite**, **PHP** y **MySQL/MariaDB**. Utiliza **Git** y **Composer** para las dependencias. Es vital configurar permisos de lectura al usuario **IUSR** en las carpetas de medios y definir reglas de reescritura para la API.

---

<sup>1</sup> <https://ampache.org/docs/installation/windows-installation-guide/>

<sup>2</sup> <https://ampache.org/docs/installation/guides/windows-installation-on-iis7.5-from-he99/>

## - Linux:

### Ubuntu 22.04<sup>3</sup>:

Para instalar Ampache en Ubuntu necesitas un entorno **LAMP** (Apache, MariaDB y PHP 8.0). Requiere instalar **FFmpeg** para transcodificación, configurar límites en php.ini y usar **Certbot** para HTTPS. Finalmente, descargas el paquete, enlazas el directorio /public y completas el asistente web.

### Resumen de dependencias clave:

- **Servidor:** Apache con a2enmod rewrite.
- **Base de datos:** MariaDB (asegurada con mysql\_secure\_installation).
- **PHP 8.0:** Extensiones como php-xml, php-mysql y php-gd.
- **Multimedia:** ffmpeg, flac y lame para el streaming.

### OpenWRT<sup>4</sup>:

Si te sientes muy aventurero, también puedes probar a instalarlo en un router al que le hayas instalado OpenWRT.

## - macOS<sup>5</sup>:

Para instalar Ampache en macOS, necesitas un servidor web (**Apache/Nginx**), **PHP 7.2+** y **MySQL/MariaDB**. Extrae los archivos en tu directorio web, renombra el archivo de configuración, crea la base de datos manualmente y ajusta los permisos de carpeta para permitir el acceso al servidor.

### Puntos clave para macOS:

- **Base de datos:** Se debe crear la DB ampache vía terminal antes de iniciar.
- **Configuración:** Es necesario renombrar manualmente config.inc.php.dist a config.inc.php.
- **Permisos:** Usa chown y chmod para que el usuario del servidor (normalmente \_www en macOS) tenga acceso.

---

<sup>3</sup> <https://ampache.org/docs/installation/install-ampache-on-ubuntu2204/>

<sup>4</sup> <https://www.youtube.com/watch?v=F-JwWWn-jc4>

<sup>5</sup> <https://ipv6.rs/tutorial/macOS/Ampache/>

## Última versión:

Según la página oficial de Ampache en GitHub<sup>6</sup>, la última es la 7.8.0.

## Licencia:

Volviendo a su página oficial en GitHub, “*ampache is licensed under the **GNU Affero General Public License v3.0***”<sup>7</sup>.

## Funcionalidades:

Según la propia página de Ampache<sup>8</sup>:

- **Colección de música**<sup>9</sup>: Explora y gestiona tu colección musical a través de una interfaz web sencilla. Sincroniza catálogos locales y remotos en una colección única y consistente.
- **Streaming de música**<sup>10</sup>: Transmite tu música a tu reproductor preferido, contrólala con Localplay o escúchala directamente en la página web con el reproductor HTML5.
- **Código abierto**<sup>11</sup>: Completamente gratuito y de código abierto desde 2001, bajo licencia AGPLv3. Participa y toma el control.
- **Música en todas partes**<sup>12</sup>: Escucha tu música desde tu teléfono, tableta o televisor. En casa, en el trabajo o de vacaciones: ¡lleva Ampache a todas partes usando un cliente compatible!

## Plugins:

Vienen instalados todos por defecto<sup>13</sup>: Admin → Manage Plugins. Aunque si quieres puedes crear el tuyo propio<sup>14</sup>. Hay bastantes opciones: para búsquedas en Amazon, mostrar localización con Google Maps, redes sociales como Tuiteer, mostrar metada y análisis con MusicBraizn o Google Analytics, etc.

---

<sup>6</sup> <https://github.com/ampache/ampache/releases>

<sup>7</sup> <https://github.com/ampache/ampache/blob/develop/LICENSE.md>

<sup>8</sup> <https://ampache.org/>

<sup>9</sup> <https://ampache.org/docs/installation/catalog/>

<sup>10</sup> <https://ampache.org/docs/configuration/>

<sup>11</sup> <https://github.com/ampache/ampache>

<sup>12</sup> <https://ampache.org/docs/clients/>

<sup>13</sup> <https://ampache.org/docs/plugins/>

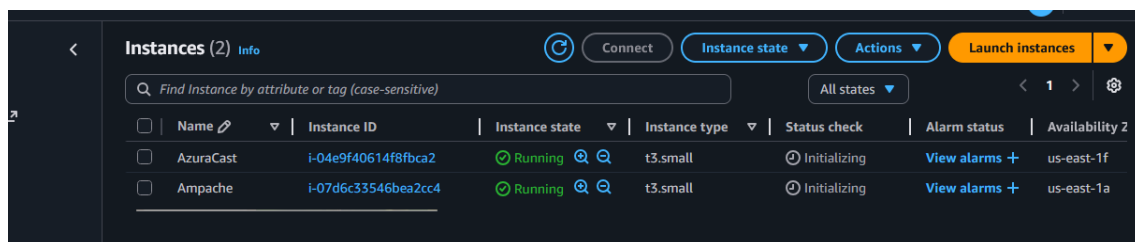
<sup>14</sup> <https://ampache.org/docs/plugins/writing-plugins/>

## 2. Instalación y configuración del servidor asignado para abordar el problema planteado.

En esta práctica vamos a usar dos servidores diferentes, uno para [Ampache](#) y otro para [AzuraCast](#). Esto es debido a que Ampache no tiene un servicio de streaming de audio en “tiempo real”, y necesita conectar con un servidor de streaming, en este caso, AzuraCast. Aprovechando que ambos tienen contenedores de Docker oficiales, vamos a usar este método. Cada uno estará montado en una instancia EC2 de AWS, con su propia IP elástica.

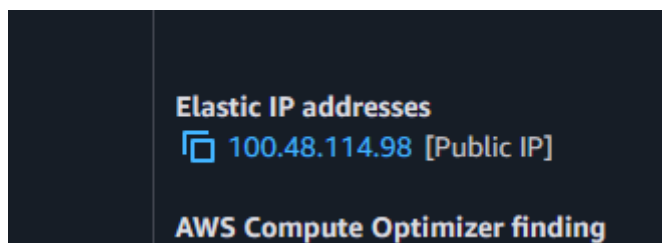
### Configuración de AWS:

Creamos una instancia para cada uno de los servicios:

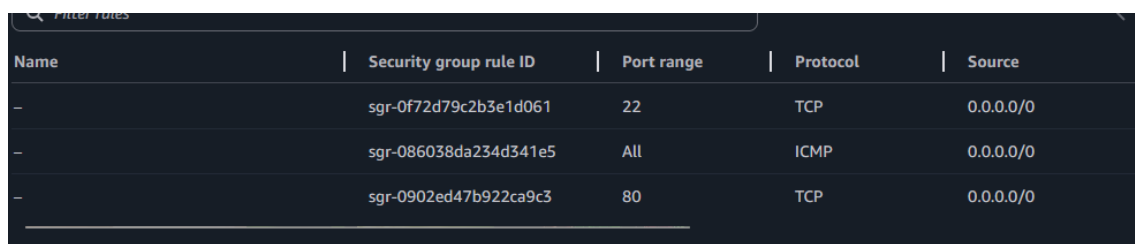


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
AzuraCast	i-04e9f40614f8fbca2	Running	t3.small	Initializing	View alarms +	us-east-1f
Ampache	i-07d6c33546bea2cc4	Running	t3.small	Initializing	View alarms +	us-east-1a

Cada uno tendrá una IP elástica, en el caso de Ampache, que será el principal, reusaremos el de la práctica anterior porque así aprovechamos el nombre del dominio que está asociado.



En cada instancia, en Security, abrimos los puertos necesarios de entrada. Para Ampache solo es necesario abrir el 80, pero para AzureCast se abren el 80 y el 443.



Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0f72d79c2b3e1d061	22	TCP	0.0.0.0/0
-	sgr-086038da234d341e5	All	ICMP	0.0.0.0/0
-	sgr-0902ed47b922ca9c3	80	TCP	0.0.0.0/0



## Ampache:

En la instancia EC2, modificamos `/etc/hosts` y `hostname` con el nombre del dominio y máquina. En este caso:

Im07.institutodh.net im07

```
root@ip-172-31-22-47: /ampache
GNU nano 7.2
127.0.0.1 localhost
127.0.1.1 im07.institutodh.net im07
::1 localhost ip6-localhost ip6-loopback

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
```

Después instalamos Docker y el plugin de Docker Compose:

```
apt update && apt install docker.io -y && apt install docker-compose-v2 -y
```

Luego, siguiendo las recomendaciones de las guías oficiales<sup>1516</sup>:

- Creamos un directorio para ampache y nos posicionamos en el:

```
mkdir /ampache
```

```
cd /ampache
```

- Creamos directorios para más adelante mapear las partes más importantes del contenedor:

```
mkdir -p data/{config,log,media,mysql}
```

- Le damos permisos a esos directorios al usuario "www-data":

```
chown www-data:www-data ./data/config -R
```

```
chown www-data:www-data ./data/log -R
```

```
chown www-data:www-data ./data/media -R
```

```
chown www-data:www-data ./data/mysql -R
```

<sup>15</sup> <https://ampache.org/docker/>

<sup>16</sup> <https://github.com/ampache/ampache-docker/blob/master/docker-compose.yml>

Creamos un docker-compose.yml

name: ampache

services:

ampache:

image: ampache/ampache:latest

container\_name: ampache

restart: unless-stopped

ports:

- 80:80

volumes:

- ./data/config:/var/www/config

- ./data/log:/var/log/ampache

- ./data/media:/media

- ./data/mysql:/var/lib/mysql

environment:

MYSQL\_USER: \${MYSQL\_USER-admin}

MYSQL\_PASS: \${MYSQL\_PASS-\*\*Random\*\*}

DB\_NAME: \${DB\_NAME-}

DB\_HOST: \${DB\_HOST-}

DB\_USER: \${DB\_USER-}

DB\_PASSWORD: \${DB\_PASSWORD-}

DB\_PORT: \${DB\_PORT-}

DISABLE\_INOTIFYWAIT\_CLEAN: \${DISABLE\_INOTIFYWAIT\_CLEAN-0}

FORCE\_INSTALL: \${FORCE\_INSTALL-0}

AMPACHE\_DB\_USER: \${AMPACHE\_DB\_USER-}

AMPACHE\_DB\_PASSWORD: \${AMPACHE\_DB\_PASSWORD-}

AMPACHE\_ADMIN\_USER: \${AMPACHE\_ADMIN\_USER-}

AMPACHE\_ADMIN\_PASSWORD: \${AMPACHE\_ADMIN\_PASSWORD-}

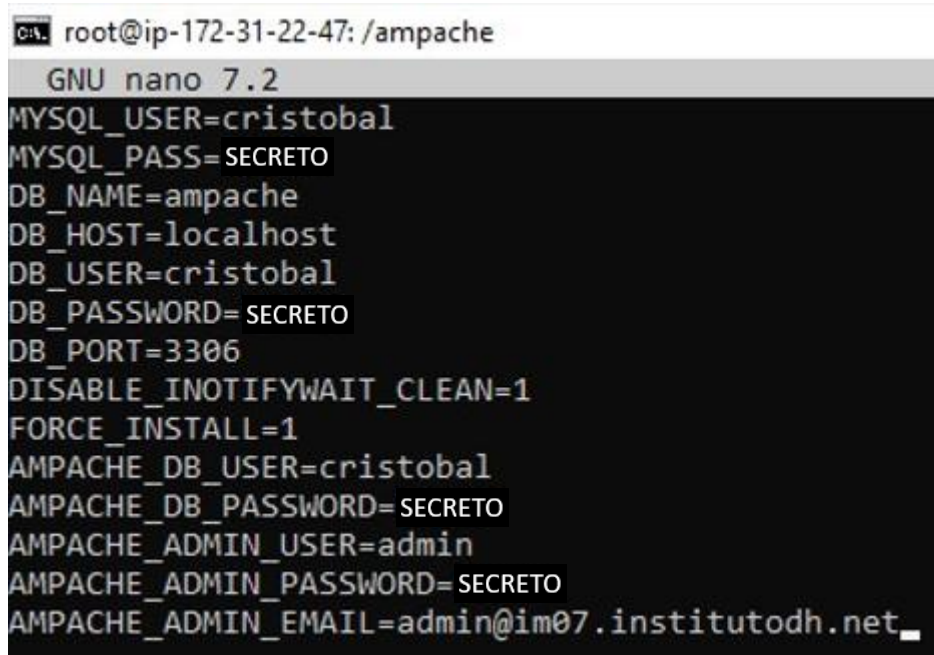
AMPACHE\_ADMIN\_EMAIL: \${AMPACHE\_ADMIN\_EMAIL-}

```
root@ip-172-31-22-47: /ampache
GNU nano 7.2
name: ampache

services:
  ampache:
    image: ampache/ampache:latest
    container_name: ampache
    restart: unless-stopped
    ports:
      - 80:80
    volumes:
      - ./data/config:/var/www/config
      - ./data/log:/var/log/ampache
      - ./data/media:/media
      - ./data/mysql:/var/lib/mysql
    environment:
      MYSQL_USER: ${MYSQL_USER-admin}
      MYSQL_PASS: ${MYSQL_PASS-**Random**}
      DB_NAME: ${DB_NAME-}
      DB_HOST: ${DB_HOST-}
      DB_USER: ${DB_USER-}
      DB_PASSWORD: ${DB_PASSWORD-}
      DB_PORT: ${DB_PORT-}
      DISABLE_INOTIFYWAIT_CLEAN: ${DISABLE_INOTIFYWAIT_CLEAN-0}
      FORCE_INSTALL: ${FORCE_INSTALL-0}
      AMPACHE_DB_USER: ${AMPACHE_DB_USER-}
      AMPACHE_DB_PASSWORD: ${AMPACHE_DB_PASSWORD-}
      AMPACHE_ADMIN_USER: ${AMPACHE_ADMIN_USER-}
      AMPACHE_ADMIN_PASSWORD: ${AMPACHE_ADMIN_PASSWORD-}
      AMPACHE_ADMIN_EMAIL: ${AMPACHE_ADMIN_EMAIL-}
```

Ahora creamos un archivo “.env” en el mismo directorio que el anterior, donde pondremos la información que usaran las variables que hemos visto en el docker-compose.yml.

```
MYSQL_USER=cristobal
MYSQL_PASS=12345
DB_NAME=ampache
DB_HOST=localhost
DB_USER=cristobal
DB_PASSWORD=*****
DB_PORT=3306
DISABLE_INOTIFYWAIT_CLEAN=1
FORCE_INSTALL=1
AMPACHE_DB_USER=cristobal
AMPACHE_DB_PASSWORD=*****
AMPACHE_ADMIN_USER=admin
AMPACHE_ADMIN_PASSWORD=*****
AMPACHE_ADMIN_EMAIL=admin@im07.institutodh.net
```



The screenshot shows a terminal window with the prompt 'root@ip-172-31-22-47: /ampache'. The terminal is running the GNU nano 7.2 text editor. The file being edited contains the following environment variables:

```
GNU nano 7.2
MYSQL_USER=cristobal
MYSQL_PASS=SECRETO
DB_NAME=ampache
DB_HOST=localhost
DB_USER=cristobal
DB_PASSWORD=SECRETO
DB_PORT=3306
DISABLE_INOTIFYWAIT_CLEAN=1
FORCE_INSTALL=1
AMPACHE_DB_USER=cristobal
AMPACHE_DB_PASSWORD=SECRETO
AMPACHE_ADMIN_USER=admin
AMPACHE_ADMIN_PASSWORD=SECRETO
AMPACHE_ADMIN_EMAIL=admin@im07.institutodh.net_
```

Después solo hay que levantar el contenedor:  
docker compose up -d

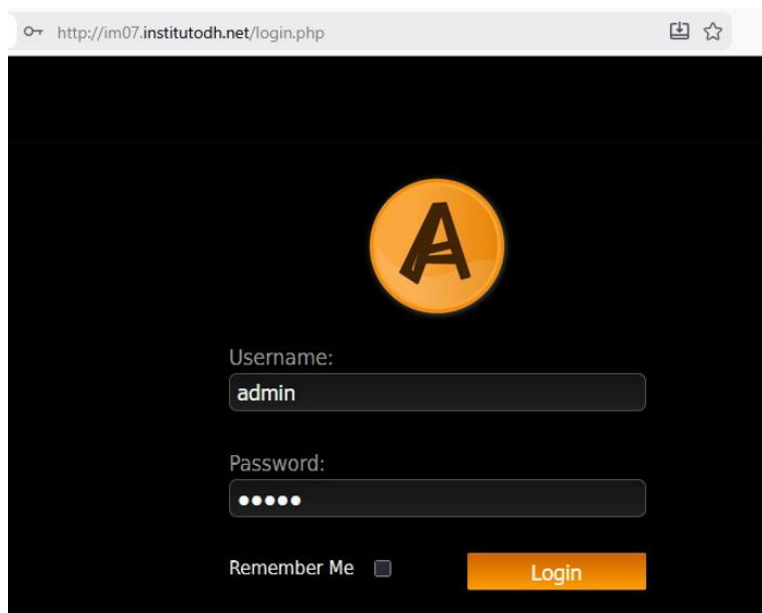
```
root@ip-172-31-22-47:/ampache# docker compose up -d
[+] Running 9/9
  ⬢ ampache Pulled
    ⬢ e4daac338336 Pull complete
    ⬢ 6023ce430f50 Pull complete
    ⬢ 5fd23dcaae62 Pull complete
    ⬢ ff24b1fe359d Pull complete
    ⬢ 92f58f4a84be Pull complete
    ⬢ 6054750f9a51 Pull complete
    ⬢ f5c22d5bc598 Pull complete
    ⬢ 2123da8849ef Pull complete
[+] Running 2/2
  ⬢ Network ampache_default Created
  ⬢ Container ampache Started
```

**IMPORTANTE:** recuerda abrir el puerto 80 en AWS. Es necesario para poder acceder al servicio.

Ahora podremos entrar desde nuestro navegador web:

<http://im07.institutodh.net>

Ahora usamos la credencial que establecimos en el archivo “.env”.



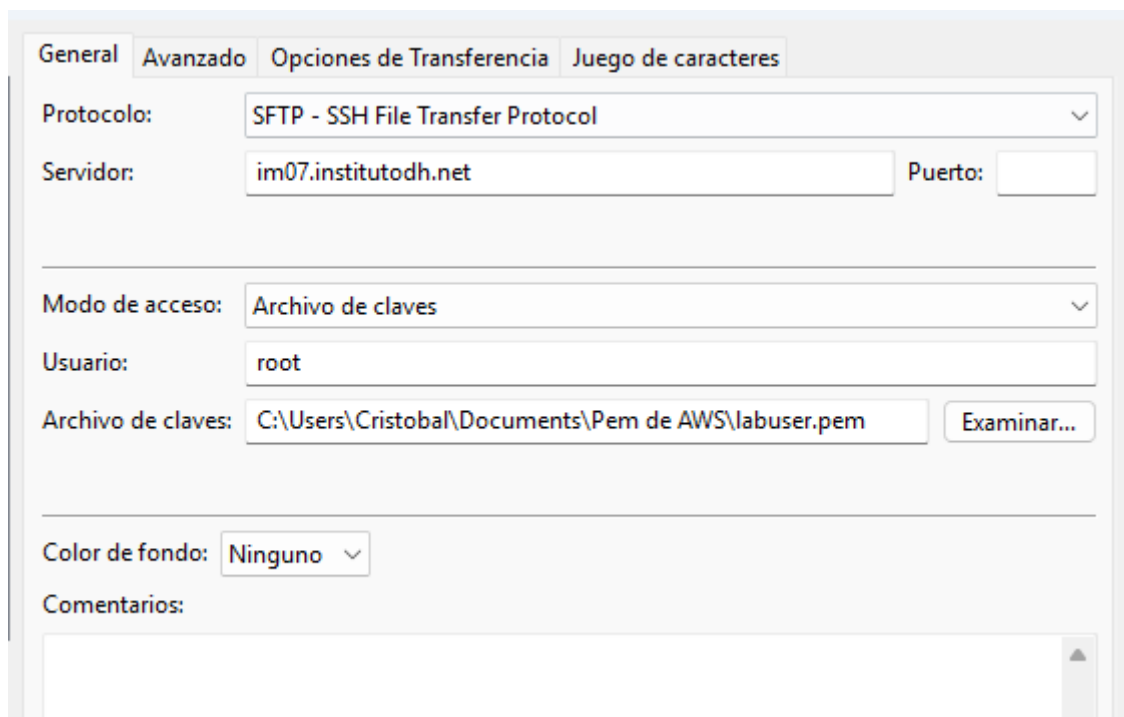
## Introducir música en Ampache:

Para subir música al servicio de Ampache tenemos dos maneras, usando FTP (recomendado para subir grandes cantidades de archivos) o configurando el servicio para permitir subir los archivos desde el navegador web.

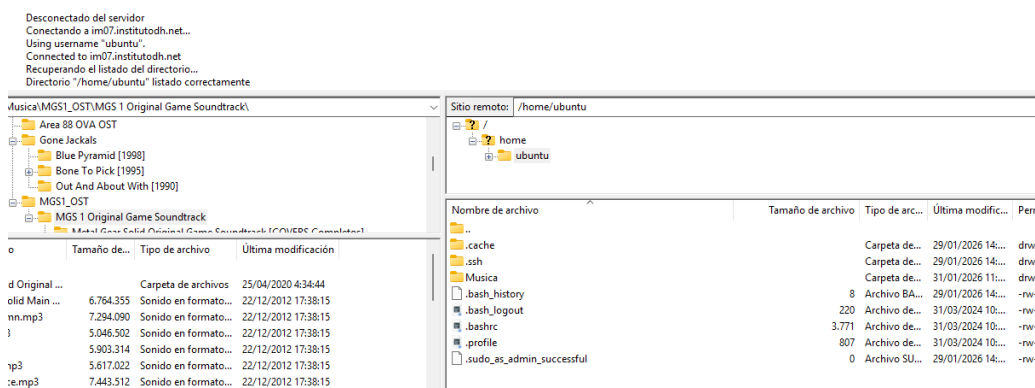
- Mediante FTP/SFTP:

Para nuestro caso usamos Filezilla. Como tenemos que usar un archivo .pem para conectarnos a la instancia, debemos configurar la conexión de esta manera.

Abrimos al “Gestor de Sitios” e introducimos los datos necesarios. El importante en este caso es el “Archivo de Claves”.



Una vez conectados podremos pasar la música a Ampache (ten en cuenta que por temas de permisos no se puede pasar directamente a /ampache/data/media).



Ahora debemos ir a “Add Catalog” y establecer nombre, que sea “local” y el directorio donde se aloja.

Los directorios de los catálogos deben pertenecer a “www-data”.

```
AudioLibro Gone_Jackals upload_catalogs
root@ip-172-31-22-47:/home/ubuntu# ls -l /ampache/data/media/
total 12
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan 31 11:57 AudioLibro
drwxr-xr-x 3 root root 4096 Jan 31 10:29 Gone_Jackals
drwxr-xr-x 3 www-data www-data 4096 Jan 31 18:24 upload_catalogs
root@ip-172-31-22-47:/home/ubuntu#
```

Los catálogos se pueden actualizar, limpiar, descargar carátulas, etc.

Catalogs

Item Count: 3

Name	Path	Last Verify	Last Add	Last Clean	Actions
Audiolibros	/media/AudioLibro	Never	1/31/26, 11:58 AM	1/31/26, 10:30 PM	<div>Add</div> <div>Add</div> <div>Verify</div> <div>Clean</div> <div>Update</div> <div>Gather Art</div> <div>Import</div> <div>Update File Tags</div> <div>Garbage Collection</div> <div>Delete</div>
Gone Jackals	/media/Gone_Jackals	Never	1/31/26, 10:30 PM	1/31/26, 10:30 PM	
Upload Catalogs	/media/upload_catalogs	Never	1/31/26, 10:30 PM	1/31/26, 10:30 PM	

Item Count: 3

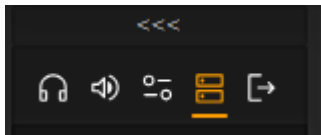
- Configurando el servidor:

Primero debemos crear un directorio dentro de “media” que será donde subamos los archivos. Recuerda que debe pertenecer a “www-data”.

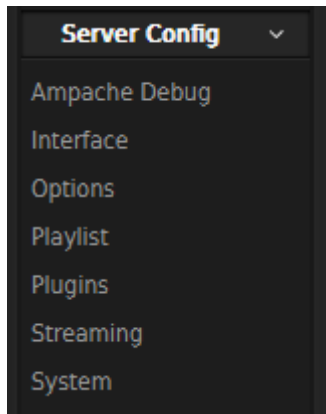
/ampache/data/media/

```
drwxr-xr-x 3 root root 4096 Jan 31 10:29 Gone_Jackals
drwxr-xr-x 3 www-data www-data 4096 Jan 31 18:24 upload_catalogs
root@ip-172-31-22-47: /home/ubuntu# |
```

Ahora, desde la cuenta de administrador, nos vamos a Admin



Y en configuraciones del servidor nos vamos a “System”.



Y nos vamos a la sección de “Uploads”: Habilitamos las secciones necesarias. En nuestro caso en la opción “Upload Access Level” hemos seleccionado porque se es el nivel más bajo que va a poder subir música (lo usaremos para prof1 y prof2).

Upload	
Allow user uploads	On ▾
Allow users to edit uploaded songs	On ▾
Allow users to remove uploaded songs	On ▾
Consider the user sender as the track's artist	Off ▾
Create a subdirectory per user	On ▾
Post-upload script (current directory = upload target directory)	<input type="text"/>
Rename uploaded file according to catalog pattern	Off ▾
Upload Access Level	Content Manager ▾

[Update Preferences](#)



Después nos vamos a “Add Catalog”, porque hay que crear un catalogo para subidas. El nombre que queramos, en “Cataloge Type” elegimos “local” y en “Path” la ruta relativa del directorio que creamos previamente.

In the form below enter either a local path (i.e. /data/music) or the URL to a remote Ampache in

Catalog Name:

Catalog Type:

Filename Pattern:

Folder Pattern:   
(no leading or ending '/')

Gather Art: ☒

Gather Art: ☒

Build Playlists from Playlist Files. (m3u, m3u8, asx, pls, xspf): ☐

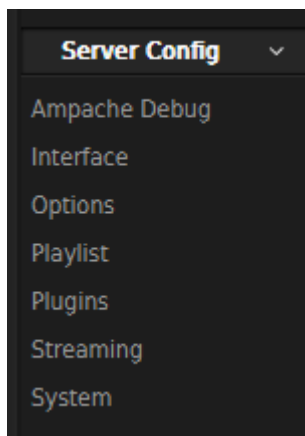
Catalog Media Type:

Path:

Una vez creado nos aparecerá junto con el resto de los catálogos:

Name	Path	Last Verify	Last Add	Last Clean	Actions
Audiolibros	/media/AudioLibro	Never	1/31/26, 11:58 AM	1/31/26, 10:30 PM	Add <input type="button" value="Go"/>
Gone Jackals	/media/Gone_Jackals	Never	1/31/26, 10:30 PM	1/31/26, 10:30 PM	Add <input type="button" value="Go"/>
Upload Catalogs	/media/upload_catalogs	Never	1/31/26, 10:30 PM	1/31/26, 10:30 PM	Add <input type="button" value="Go"/>

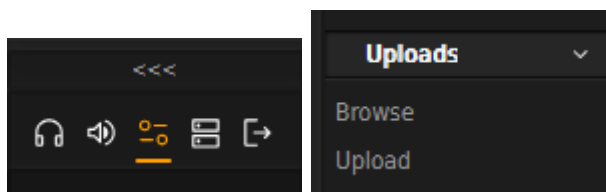
Ahora vamos a “Options” y nos dirigimos a la sección de “Upload”.



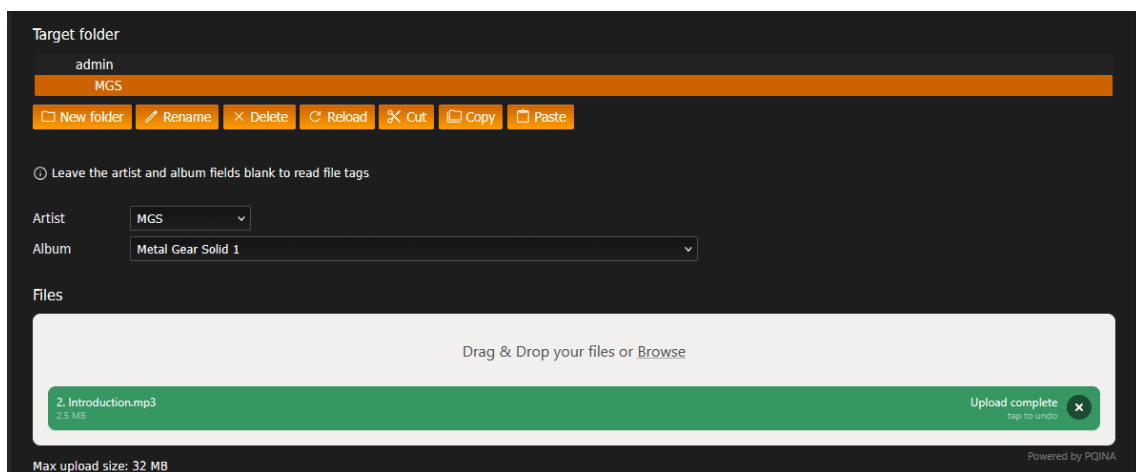
Ahí establecemos el catálogo para las subidas. Marca la casilla si quieres que se aplique para todo el mundo.



Ahora nos vamos a “Preferences” y a “Upload”.



Podremos subir archivos, establecer artista, album, crear carpetas, etc.



En “Browse” veremos que está el archivo y el álbum. También aparecerán para el resto de los usuarios.

Metal Gear Solid Main Theme

Rika Muranaka

Metal Gear Solid Original Game Soundtrack

1998

Soundtrack

2:43

☆

☆

☆

☆

☆

♡

📄

⬇

✎

Introduction

Rika Muranaka

Metal Gear Solid Original Game Soundtrack

1998

Soundtrack

0:57

☆

☆

☆

☆

☆

♡

📄

⬇

✎


Item Count: 3

View

Albums

Item Count: 1

View

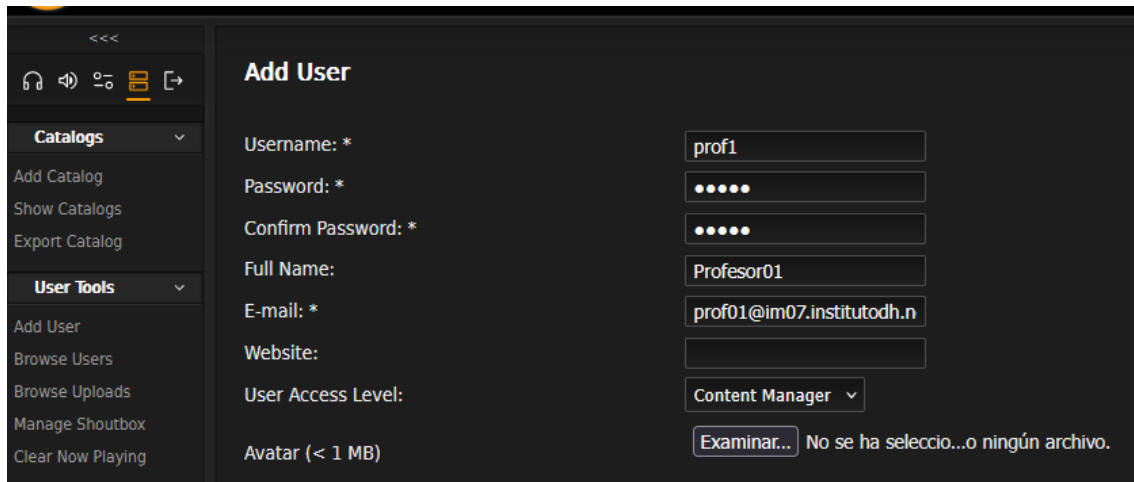
Art	Album	Album Artist	Songs	Year	Genres	Rating	Actions
	Metal Gear Solid 1	MGS	1		Soundtrack	<div><div></div><div>☆</div><div>☆</div><div>☆</div><div>☆</div><div>☆</div></div> ♡	<div><div></div><div>📄</div><div>✎</div></div>

Item Count: 1

View

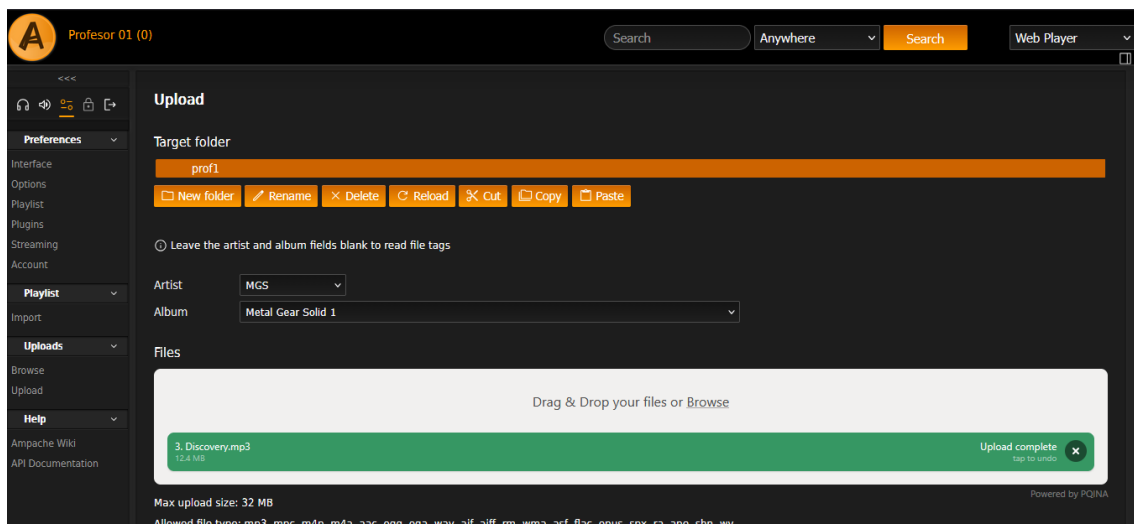
## Usuarios

Admin → Add user: Importante establecer el “User Access Level”.



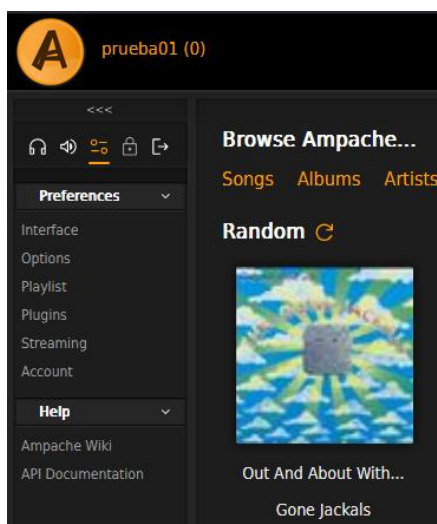
The screenshot shows the 'Add User' form in the Ampache web interface. The form is titled 'Add User' and contains several input fields for user information. The fields are: Username (\*), Password (\*), Confirm Password (\*), Full Name, E-mail (\*), Website, User Access Level, and Avatar (< 1 MB). The Username field is filled with 'prof1'. The Password and Confirm Password fields are filled with six dots. The Full Name field is filled with 'Profesor01'. The E-mail field is filled with 'prof01@im07.institutodh.n'. The User Access Level dropdown menu is set to 'Content Manager'. The Avatar field has a button labeled 'Examinar...' and a message 'No se ha seleccio...o ningún archivo.'.

Vamos a comprobar que puede subir archivos:



The screenshot shows the 'Upload' page in the Ampache web interface. The page is titled 'Upload' and contains a 'Target folder' dropdown menu set to 'prof1'. Below the dropdown menu are buttons for 'New folder', 'Rename', 'Delete', 'Reload', 'Cut', 'Copy', and 'Paste'. There is a checkbox labeled 'Leave the artist and album fields blank to read file tags'. The Artist dropdown menu is set to 'MGS' and the Album dropdown menu is set to 'Metal Gear Solid 1'. Below these fields is a 'Files' section with a message 'Drag & Drop your files or Browse'. A file named '3\_Discovery.mp3' (12.4 MB) is shown in a green bar with a button labeled 'Upload complete' and a close button. At the bottom, there is a note 'Max upload size: 32 MB' and a list of allowed file types: mp3, mpc, m4p, m4a, aac, ogg, oga, wav, aif, aiff, rm, wma, asf, flac, opus, spx, ra, ape, shn, wv.

El usuario prueba01, que es solo “user” no puede subir nada.



The screenshot shows the 'Browse Ampache...' page in the Ampache web interface. The page is titled 'Browse Ampache...' and contains tabs for 'Songs', 'Albums', and 'Artists'. Below the tabs is a 'Random' button with a circular arrow icon. A music player is visible, showing a cover image and the title 'Out And About With...'. Below the player is the text 'Gone Jackals'.

## Azuracast

En esta ocasión no hemos tenido que establecer ninguna configuración en el archivo `/etc/hosts`.

En AWS recuerda abrir los puertos 80 y 443.

Name	Security group rule ID	Port range	Protocol	Source
–	sgr-07c139fbb1471443b	All	ICMP	0.0.0.0/0
–	sgr-029f193e0c28d44b0	80	TCP	0.0.0.0/0
–	sgr-0c3f3bda257fa9382	443	TCP	0.0.0.0/0
–	sgr-0a431fa4326db9e09	22	TCP	0.0.0.0/0

Seguimos la guía oficial<sup>17</sup>:

```
mkdir -p /var/azuracast
```

```
cd /var/azuracast
```

```
curl -fsSL https://raw.githubusercontent.com/AzuraCast/AzuraCast/main/docker.sh >
docker.sh
```

```
chmod a+x docker.sh
```

```
yes 'Y' | ./docker.sh setup-release
```

*"Your current release channel is 'Rolling Release'. Switch to 'Stable' release channel?  
[y/N] You should update the Docker Utility Script after changing release channels.  
Automatically update it now? [Y/n] New Docker utility script downloaded.*

*You can now re-run any previous command with the updated utility script."*

```
./docker.sh install
```

- Durante la instalación te saldrán varias opciones:

Elige idioma (usamos English porque hay más guías).

Customize ports: En nuestro caso no.

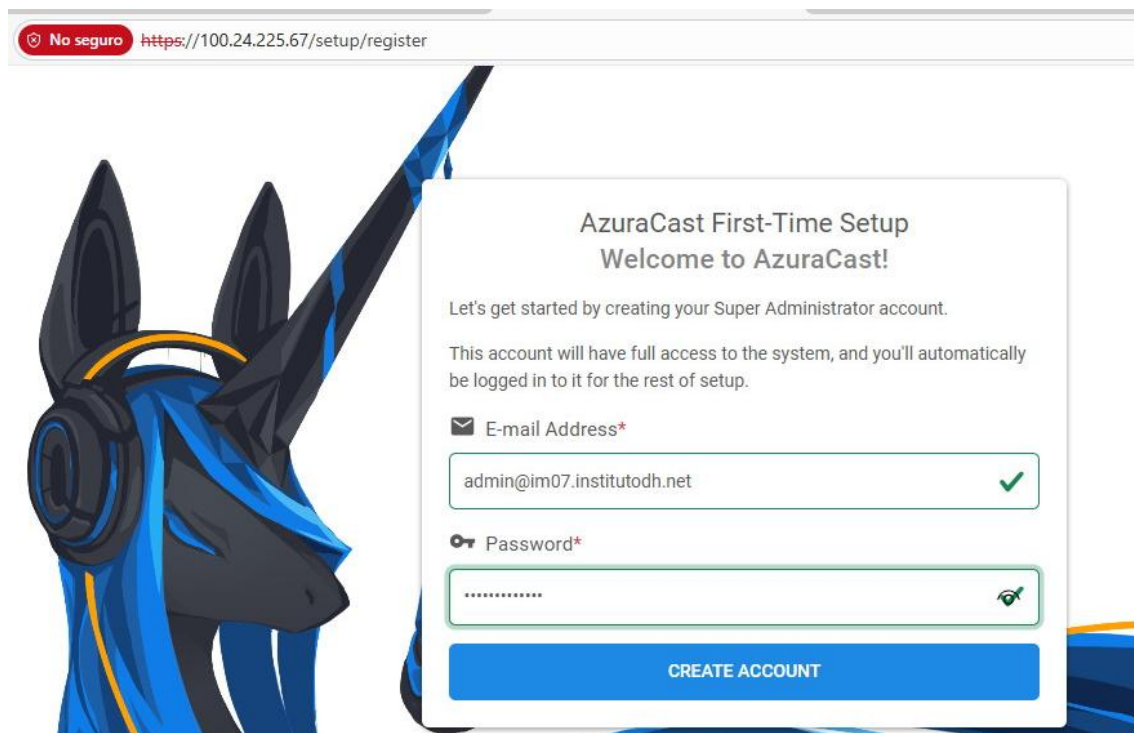
Enable web based docker image updates: En nuestro caso no.

Automatically block common bots and crawlers (yes/no) yes

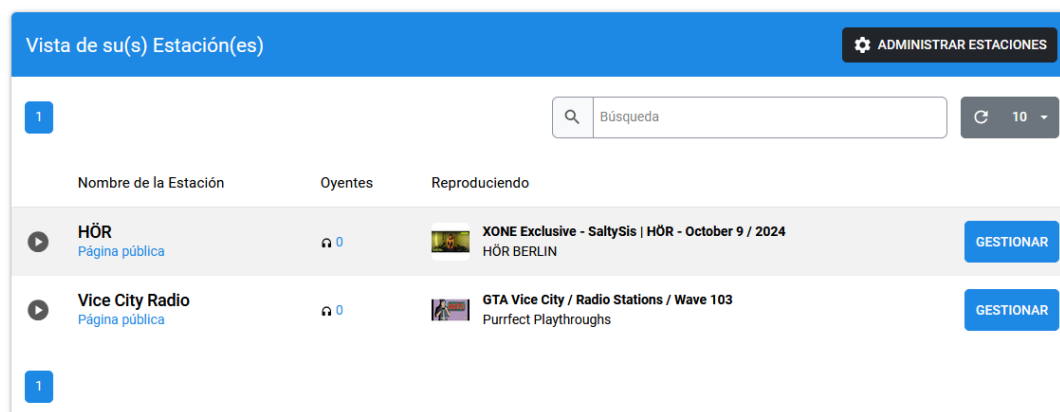
<sup>17</sup> <https://www.azuracast.com/docs/getting-started/installation/docker/>

Una vez terminado, nos dirigimos a:

<https://100.24.225.67> (en este caso es HTTPS). Y nos registramos.



Al principio no hay ninguna radio creada. Para crearlas le damos a “Administrar Estaciones”. Y luego “Añadir Estación”.



En “Perfil” establecemos los campos que se ven en la imagen. Ten en cuenta que más adelante se pueden modificar.

Añadir Estación

Perfil

Emitiendo

AutoDJ

HLS

Pedidos de Canciones

Streamers/DJs

Administración

Nombre\*

Mojinos Escozios

Descripción

Recopilatorio de los mejores temas de los Mojinos Escozios

Género

parody rock

URL del sitio web

https://mojinosescozios.com

Zona Horaria

Madrid (UTC+1)

Número de Canciones Visibles Recientes

Reproducidas

Avanzado

☐ Desactivado
☐ 1
☒ 5
☐ 10
☐ 15
☐ Personalizado

Stub de URL

Avanzado

mojinosescozios

Importante activar como “Página Pública”. Las otras dos opciones según las necesidades.

## Páginas Públicas

☒ Activar Páginas Públicas

Mostrar la estación en páginas públicas y resultados generales de la API.

## Habilitar Streaming Bajo Demanda

☒ Habilitar Streaming Bajo Demanda

Si está habilitado, la música de la listas de reproducción con streaming bajo demanda habilitado, estarán disponible para transmitir y descargar a través de una página pública especializada.

☒ Habilitar Descargas en la Página Bajo Demanda

Si está habilitado, un botón de descarga también estará presente en la página pública "On Demand".

CERRAR

GUARDAR LOS CAMBIOS

IceCast 2.4 viene activado por defecto. En esta pestaña también se pueden bloquear (o permitir) direcciones IPs, puertos, numero de usuarios, etc. También prohibir países. También permite hacer la configuración por XML o JSON.

Añadir Estación

Perfil

Emitiendo

AutoDJ

HLS

Pedidos de Canciones

Streamers/DJs

Administración

**Servicio Local de Radiodifusión\***

☒ Usa Icecast 2.4 en este servidor.

☐ No utilice un servicio de radiodifusión local.

Este software se utiliza tradicionalmente para entregar su emisión a sus oyentes. Puede emitir de forma remota o a través de HLS si este servicio está deshabilitado.

El servicio de AutoDJ viene activado por defecto y será el encargado de administrar las playlist durante la transmisión.

Añadir Estación

Perfil

Emitiendo

AutoDJ

HLS

Pedidos de Canciones

Streamers/DJs

Administración

**Servicio de AutoDJ\***

☒ Usar Liquidsoap en este servidor.

☐ No utilice un servicio de AutoDJ.

Este software cambia constantemente las listas de reproducción de música y se reproduce cuando no hay otra fuente de radio disponible.

**Método de Crossfade**

**Duración de crossfade (segundos)**

Habilitar las peticiones de canciones.

Añadir Estación

Perfil

Emitiendo

AutoDJ

HLS

Pedidos de Canciones

Streamers/DJs

Administración

**Pedidos de Canciones**

Algunos proveedores de licencias de streaming pueden tener reglas específicas con respecto a las solicitudes de canciones. Revisa tus regulaciones locales para más información.

☒ Permitir solicitudes de canciones

Permite que los oyentes soliciten una canción para reproducirla en tu estación. Solo se pueden solicitar las canciones que ya están en tus listas de reproducción.

Retraso Mínimo de Solicitud (Minutos)

Tiempo de espera antes de pedir un nuevo título (minutos)

Si las peticiones están habilitadas, esto especifica el retraso mínimo (en minutos) entre una solicitud que se envía y se reproduce. Si se establece en cero, se aplica un retraso menor de 15 segundos para prevenir saturaciones de solicitudes.

Esto especifica el tiempo mínimo (en minutos) entre una canción que se reproduce en la radio y que vuelve a estar disponible para solicitarse de nuevo. Establecer en 0 para ningún umbral.

CERRAR

GUARDAR LOS CAMBIOS



## Establecer directorios y habilitar la transmisión.

Perfil
Emitiendo
AutoDJ
HLS
Pedidos de Canciones
Streamers/DJs
Administración

☒ Habilitar Transmisión  
Si está desactivado, la estación no transmitirá ni reproducirá aleatoriamente su AutoDJ.

Directorio Base de la Estación A avanzado  
  
El directorio principal donde se almacenan los archivos de configuración y la lista de reproducción de la estación. Déjelo en blanco para usar el directorio predeterminado.

Maximum Bitrate  
  
The maximum bitrate in which the station allowed to broadcast at, in Kbps. 0 for unlimited.

Maximum Mounts  
  
The maximum number of mount points allowed. 0 for unlimited.

Maximum HLS Streams  
  
The maximum number of HLS streams allowed. 0 for unlimited.

Ubicación de Almacenamiento de los Medios

Ubicación del Almacenamiento de Las Grabaciones en Vivo

Ubicación del Almacenamiento de Podcasts

## Introducir canciones en AzureCast

Cada radio tiene su directorio de almacenamiento o pueden compartirlo (aunque no encuentro la manera de incluir más de un directorio). Nos vamos a Medios → Archivos de Música.

Mojinos Escozios  
10:12:53 a. m. CET

**Iniciar Estación**  
¿Listo para empezar a emitir? Haz click para iniciar tu estación.

Overview

Edit Station Settings

Páginas Públicas

Medios

Archivos de música

Canciones duplicadas

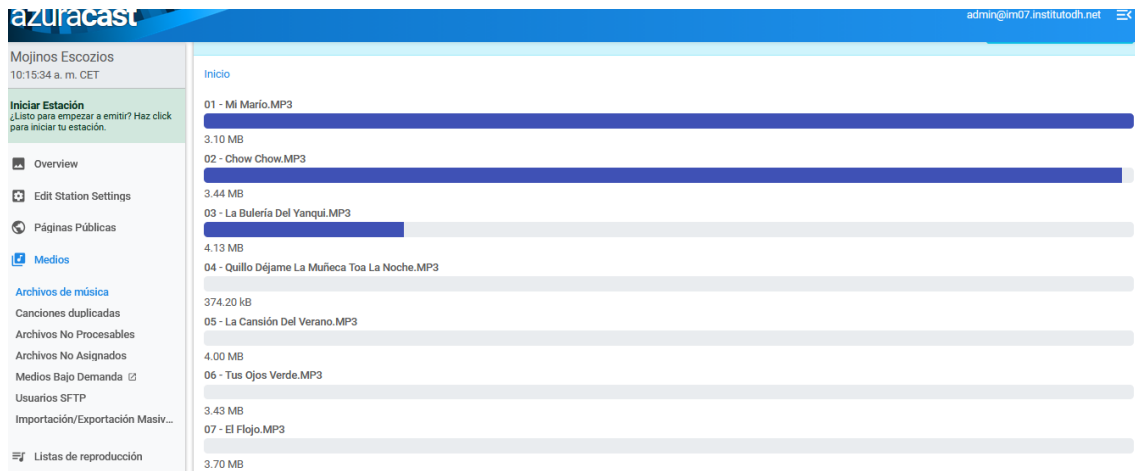
Archivos No Procesables

Archivos No Asignados

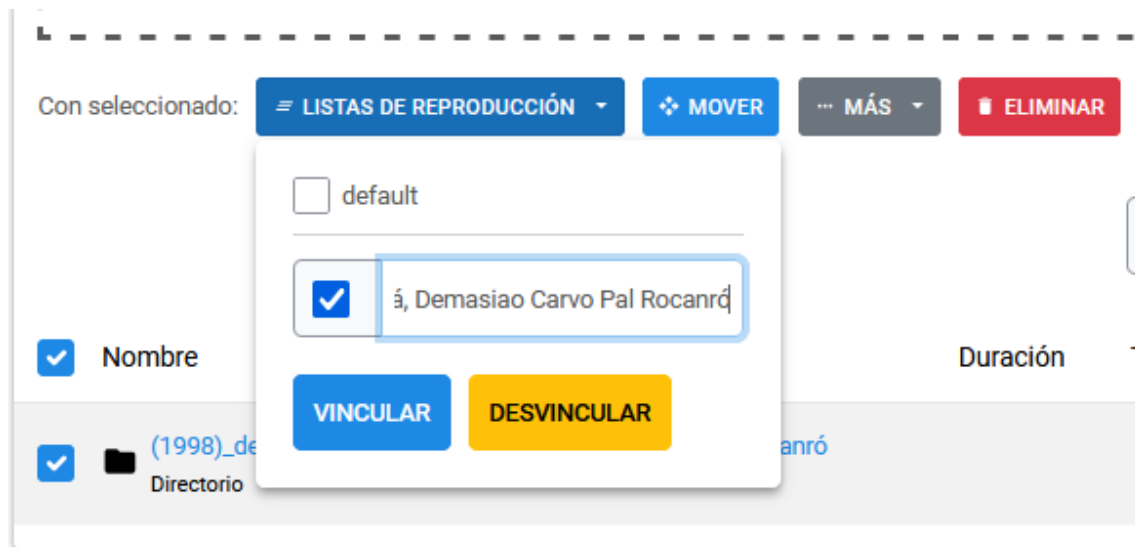
Medios Bajo Demanda ☒

Usuarios SFTP

Y arrastramos los archivos:



Después lo seleccionamos y lo metemos en una lista de reproducción. Puede ser Default o una nueva.





En la parte de la derecha vemos controles básicos para parar, iniciar y reiniciar la radiodifusión.

Servicio de Radiodifusión  
Icecast

En Ejecución

Administración	Nombre de Usuario: admin Contraseña: hccr3zQE	
Puerto	8020	
Fuente	Nombre de Usuario: source Contraseña: ZqcFQCMz	
Relé	Nombre de Usuario: relay Contraseña: m3RznM47	

... OCULTAR CREDENCIALES
REINICIAR
DETENER

Servicio de AutoDJ  
Liquidsoap

En Ejecución

REINICIAR
DETENER

En emisiones tenemos la URL Streaming que usaremos para ligar la transmisión a la Radio Station de Ampache (y otros clientes). En páginas publicas los diferentes enlaces para los servicios que ofrece AzuraCast a los usuarios.

Emisiones

Streams Locales	Oyentes
<div> /radio.mp3 (192kbps MP3)  <a href="https://100.24.225.67/listen/mojinosescozios/radio.mp3">https://100.24.225.67/listen/mojinosescozios/radio.mp3</a> </div> <div> 1 1  1 Únicos </div>	

DESCARGAR PLS
DESCARGAR M3U

Páginas Públicas

Habilitado

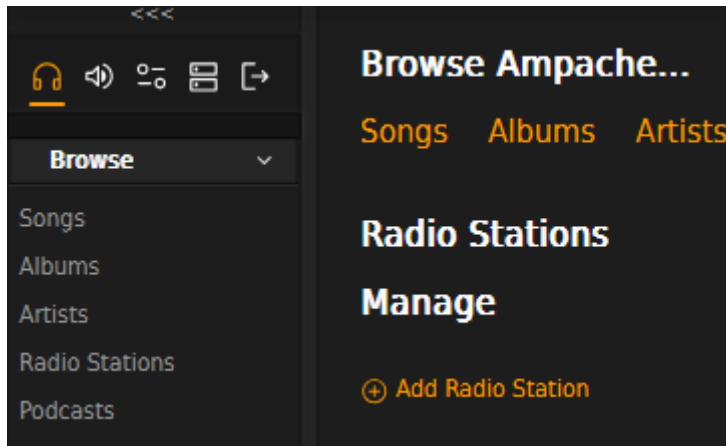
Página pública	<a href="https://100.24.225.67/public/mojinosescozios">https://100.24.225.67/public/mojinosescozios</a>
Medios Bajo Demanda	<a href="https://100.24.225.67/public/mojinosescozios/ondemand">https://100.24.225.67/public/mojinosescozios/ondemand</a>
Podcasts	<a href="https://100.24.225.67/public/mojinosescozios/podcasts">https://100.24.225.67/public/mojinosescozios/podcasts</a>
Programar	<a href="https://100.24.225.67/public/mojinosescozios/schedule">https://100.24.225.67/public/mojinosescozios/schedule</a>

< > INSERTAR WIDGETS
EDITAR MARCA
X DESHABILITAR

## Añadir “Radio Station” a Ampache:

Una vez configurada la radiodifusión en Azuracast, vamos a configurar Ampache para que pueda “emitir” esta radio.

Home → Radio Stations → Add Radio Stations



Nombre el que queramos, la web igual, la URL Stream es muy importante (ponemos el enlace acabado en este caso en .mp3 que vimos antes), el codec igual que la extensión del enlace de streaming. Y la estación no hay porque crear una propia, pero queda más ordenado así.

### Add Radio Station

Name	Mojinos Escozios
Website	.67/public/mojinosescozios
Stream URL	nojinosescozios/radio.mp3
Codec	mp3
Catalog	Mojinos Escozios ▼

Name	Website	Codec	Rating	Action
GTA Vice City Radio Stations	https://100.24.225.67/public/vicacityradio	mp3	☆☆☆☆☆	🗑️ ⚙️ ✖️
Mojinos Escozios	https://100.24.225.67/public/mojinosescozios	mp3	☆☆☆☆☆	🗑️ ⚙️ ✖️

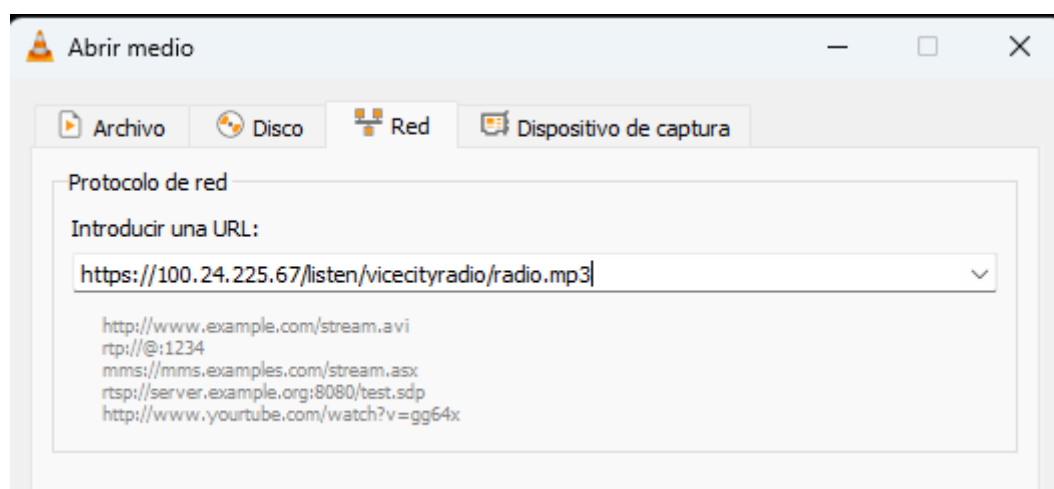
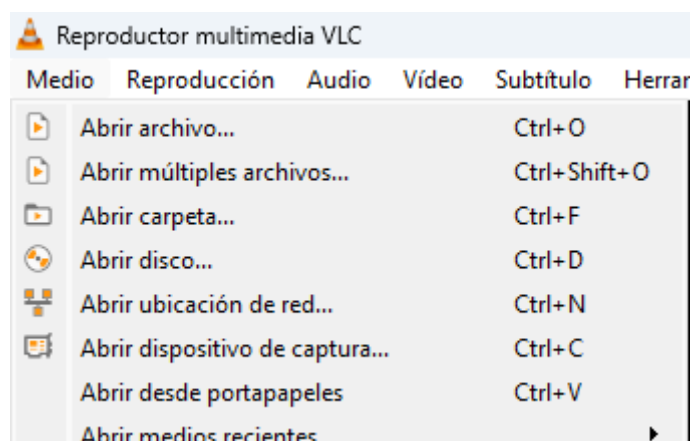
### 3. Instalación y configuración de clientes específicos (No navegador, apps, apks, etc).

S.O. de escritorio:

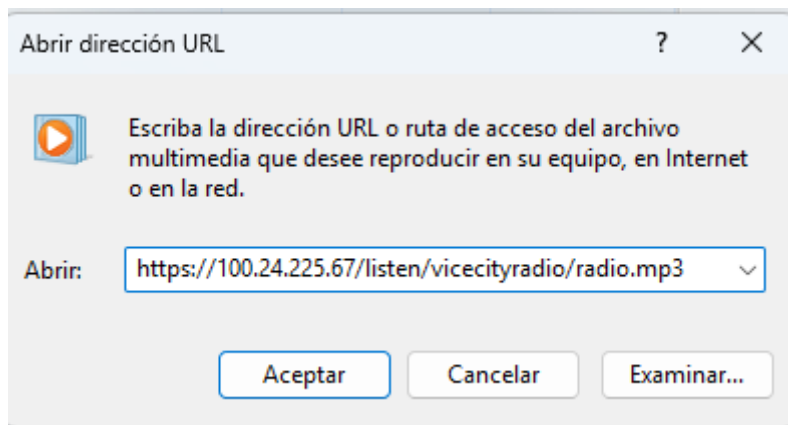
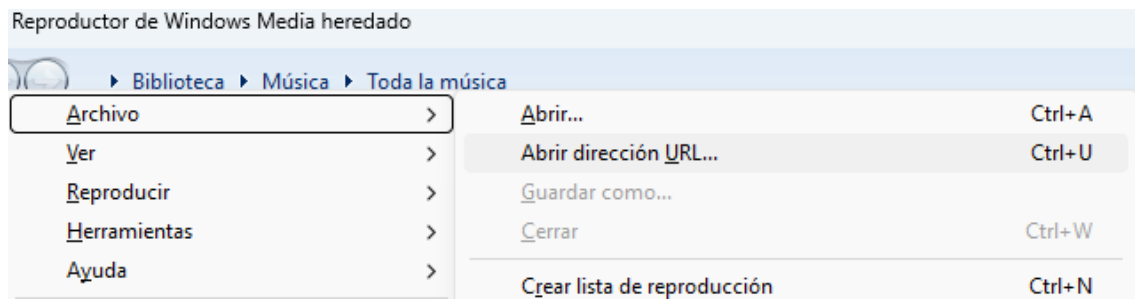
La mayoría de los reproductores multimedia tienen esta opción:

- VLC:

Abrir ubicación de red:



- Windows Media Player:

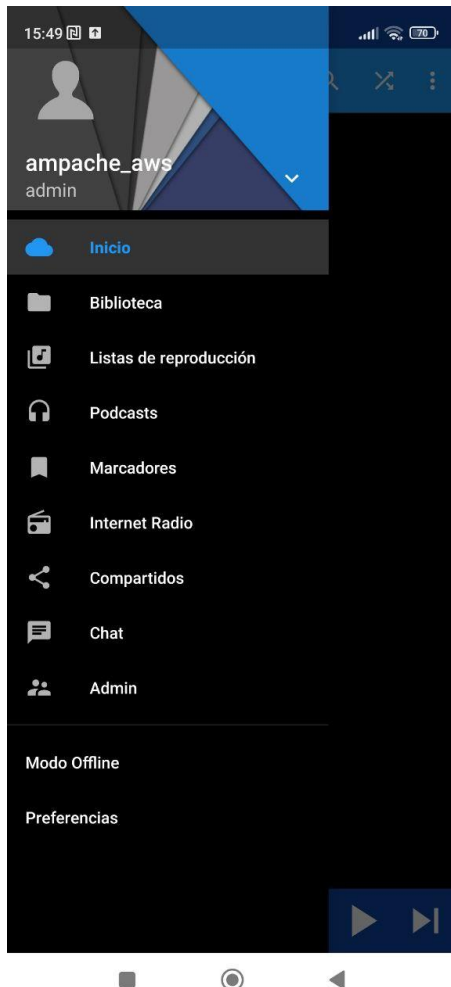


## Android

- DSub2000:

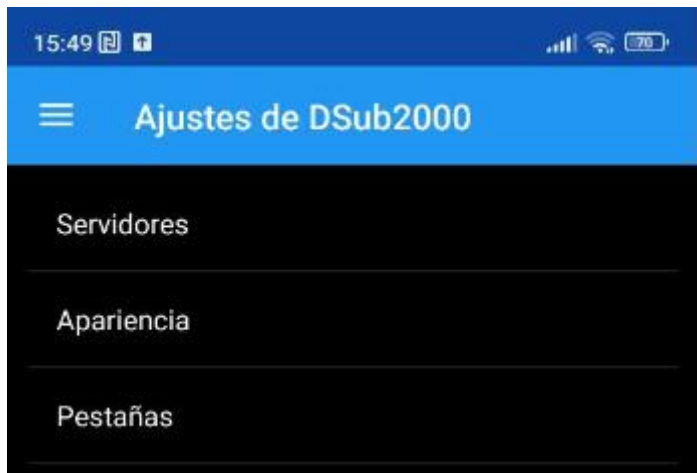


Nos vamos a Preferences

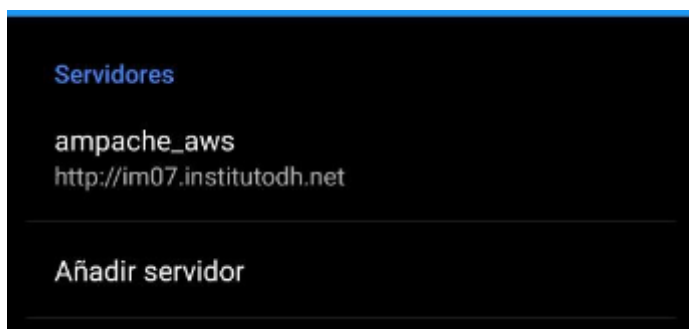




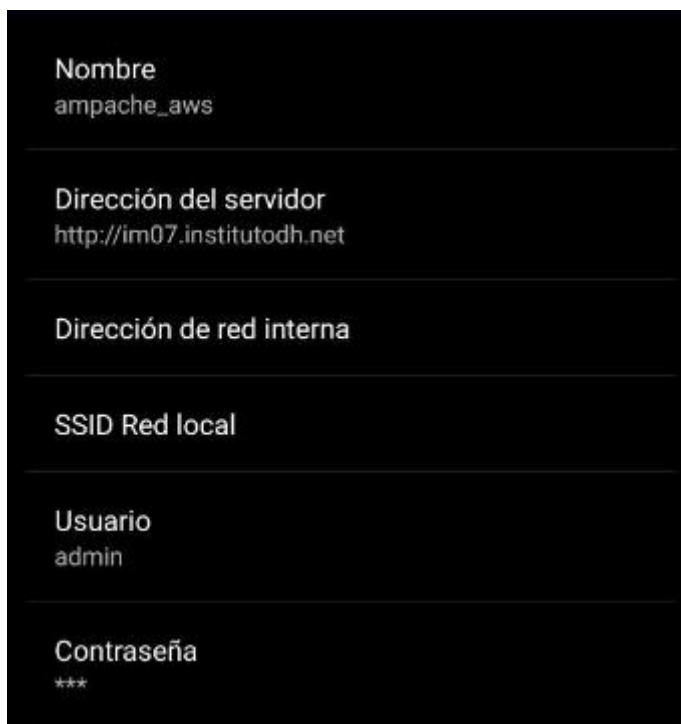
## Servidores



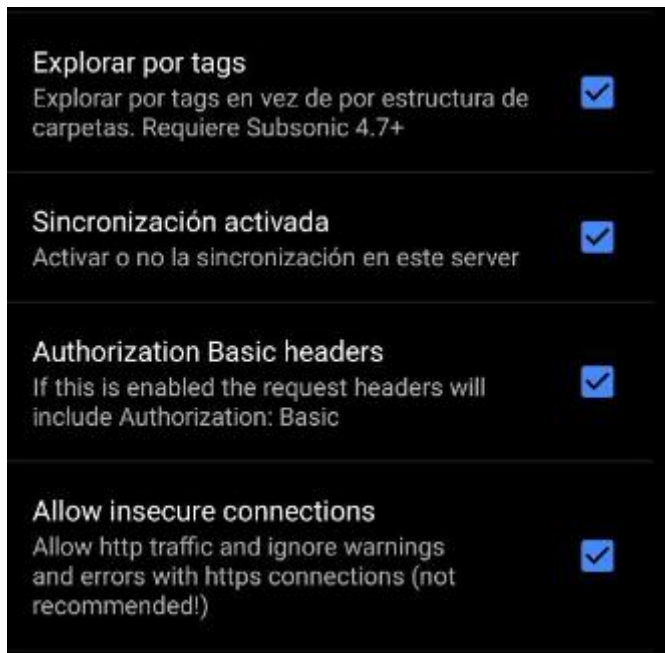
## Añadir Servidor



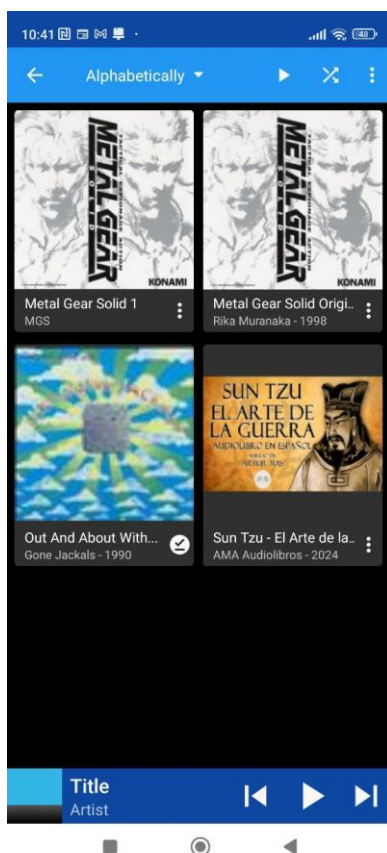
## Sus datos



Y en este caso permitir la conexión no segura.



Una vez terminado, nos saldrán los álbumes y canciones en la sección correspondiente. Si obtiene las Radio Stations, pero no las reproduce. Por eso es necesaria una segunda app.

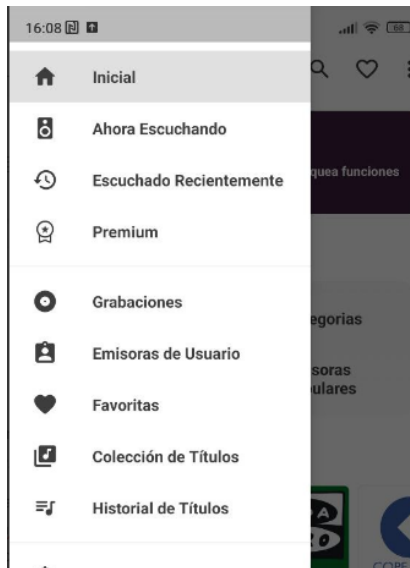


## - VRadio:



Como el anterior tiene un soporte malo para las emisoras de radios, se ha tenido que recurrir a una segunda app.

## Emisoras de usuario



Crear una emisora.



Introducir los datos y guardar.

←

Editar Emisora

🔍

❤️

⋮

Nombre

📁

HÖR

Flujo

🔗

https://100.24.225.67/listen/horberlin/radio.mp3

Agregar flujo adicional

🌐

Sitio de Web

🖼️

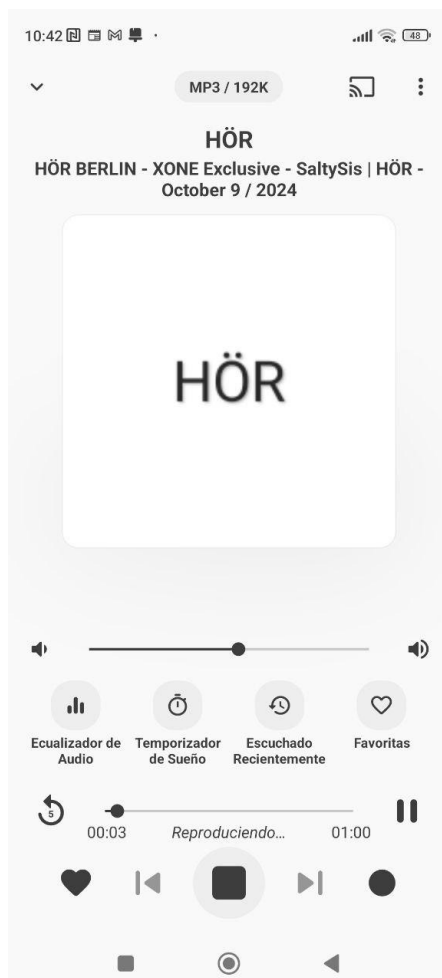
HÖR

User-Agent (HTTP Header)

💾

Guardar

Una vez terminado nos saldrá en la sección de Emisoras de Usuario.



4. Analiza la posibilidad de difusión de la Radio del instituto desde la plataforma: “Shoutcast” o de cualquier otra plataforma. Implementad una prueba simulada, analizando las posibilidades de conexión de los clientes, costes derivados, etc...

**ShoutCast Basic** no tiene coste<sup>18</sup>, a cambio de que el usuario gestione su propio servidor. La ausencia de un Docker oficial no es un problema real, ya que la instalación sigue siendo relativamente sencilla<sup>19</sup>.

---

<sup>18</sup> <https://www.shoutcast.com/pricing>

<sup>19</sup> <https://en.ubunlog.com/how-to-install-shoutcast-on-ubuntu/>