	Nombre: Juan Diego Orrillo		Grupo: ASIR 2
	Fecha: Dic 24 Ene 25	Curso 2024/2025	Módulo Profesional: SRI
Trabajo Investigación – UA7 Servicio de Audio			
Normas generales: <ul style="list-style-type: none"> La ausencia de los pasos y/o explicaciones necesarias en la resolución de los ejercicios se penalizará hasta con un 100%, aunque el resultado sea correcto. 			

PUNTUACIÓN:

Ejercicio	RA	% de la RA en el total	CE	% del CE en su RA	Calificación del ejercicio sobre 10
1	7	5%	b)	15%	10
2	7	5%	d)	10%	10
3	7	5%	g)	15%	10

Todas las documentaciones deberán llevar Introducción, Desarrollo y Conclusiones.

Contenido

EJERCICIO 1.....	3
Introducción	3
Desarrollo Linux.....	3
Desarrollo Windows	5
Conclusiones.....	8
EJERCICIO 2.....	8
Introducción	8
Desarrollo	8
Conclusiones.....	12
EJERCICIO 3.....	12
Introducción	12
Desarrollo	12
Conclusiones.....	15

EJERCICIO 1

Se ha instalado y configurado un servidor de distribución de audio en Linux Server y en Windows Server. Explicar paso a paso las diferentes acciones realizadas para poner a punto el servicio.

Introducción

El objetivo de este ejercicio es implementar un servidor de distribución de audio utilizando Icecast, tanto en **Ubuntu Desktop** como en **Windows Server**. Este servidor permitirá la **transmisión de contenido de audio** a los clientes mediante **streaming en tiempo real y bajo demanda**. Para ello, se seguirá los pasos necesarios para **instalar y configurar** el servidor en ambos sistemas operativos.

Desarrollo Linux

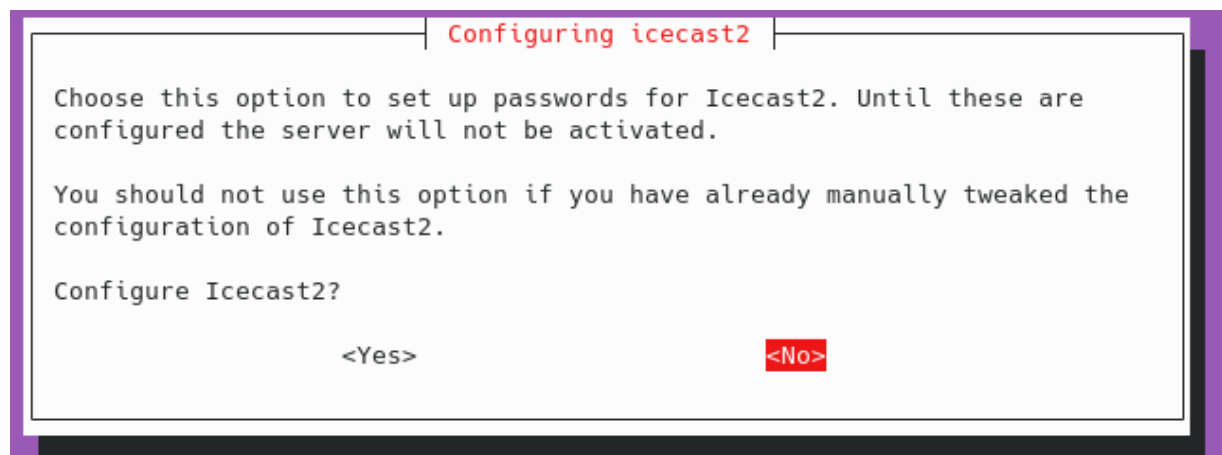
Iniciamos sesión y posteriormente **actualizamos los repositorios**

```
root@asir217-VirtualBox: /home/asir217
root@asir217-VirtualBox:/home/asir217# apt update && apt upgrade -y
Hit:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
```

Instalamos el programa necesario

```
root@asir217-VirtualBox:/home/asir217# apt install icecast2
Reading package lists... Done
Building dependency tree... Done
```

Y prefiero configurar luego icecast2




Vamos a hacer un `systemctl enable icecast`, un `start` y por último vemos el estado.

```

root@asir217-VirtualBox:/home/asir217# systemctl enable icecast2
icecast2.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable icecast2
root@asir217-VirtualBox:/home/asir217# systemctl start icecast2
root@asir217-VirtualBox:/home/asir217# systemctl status icecast2
● icecast2.service - LSB: Icecast2 streaming media server
   Loaded: loaded (/etc/init.d/icecast2; generated)
   Active: active (running) since Mon 2024-12-09 21:56:17 CET; 4s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 3271 ExecStart=/etc/init.d/icecast2 start (code=exited, status=0/S>
    Tasks: 4 (limit: 4608)
   Memory: 4.4M
      CPU: 39ms
   CGroup: /system.slice/icecast2.service
           └─3276 /usr/bin/icecast2 -b -c /etc/icecast2/icecast.xml

```



Editamos el **archivo de configuración de Icecast** que está en la ruta especificada en la captura

```

root@asir217-VirtualBox:/home/asir217# nano /etc/icecast2/icecast.xml

```

Y adentro vamos a configurar la locación, correo del admin

```

(server_version.xml). -->
<location>Spain, Cantabria, Santander</location>
<admin>adminasir2@localhost</admin>

```

La autenticación:

```

<authentication>
  <!-- Sources log in with username 'source' -->
  <source-password>Root12345</source-password>
  <!-- Relays log in with username 'relay' -->
  <relay-password>Root123456</relay-password>

  <!-- Admin logs in with the username given below -->
  <admin-user>admin</admin-user>
  <admin-password>iesalisal2024</admin-password>
</authentication>

```

También es necesario cambiar la ip por defecto

```

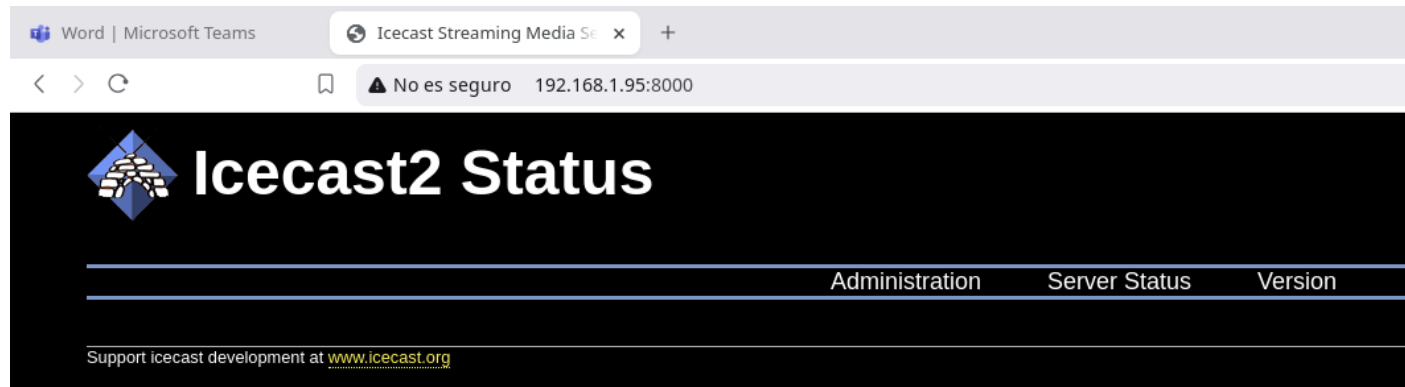
<hostname>localhost</hostname>

<!-- You may have multiple <listen-socket> element
<listen-socket>
  <port>8000</port>
  <bind-address>192.168.1.95</bind-address>
  <!-- <shoutcast-mount>/stream</shoutcast-mount>
</listen-socket>

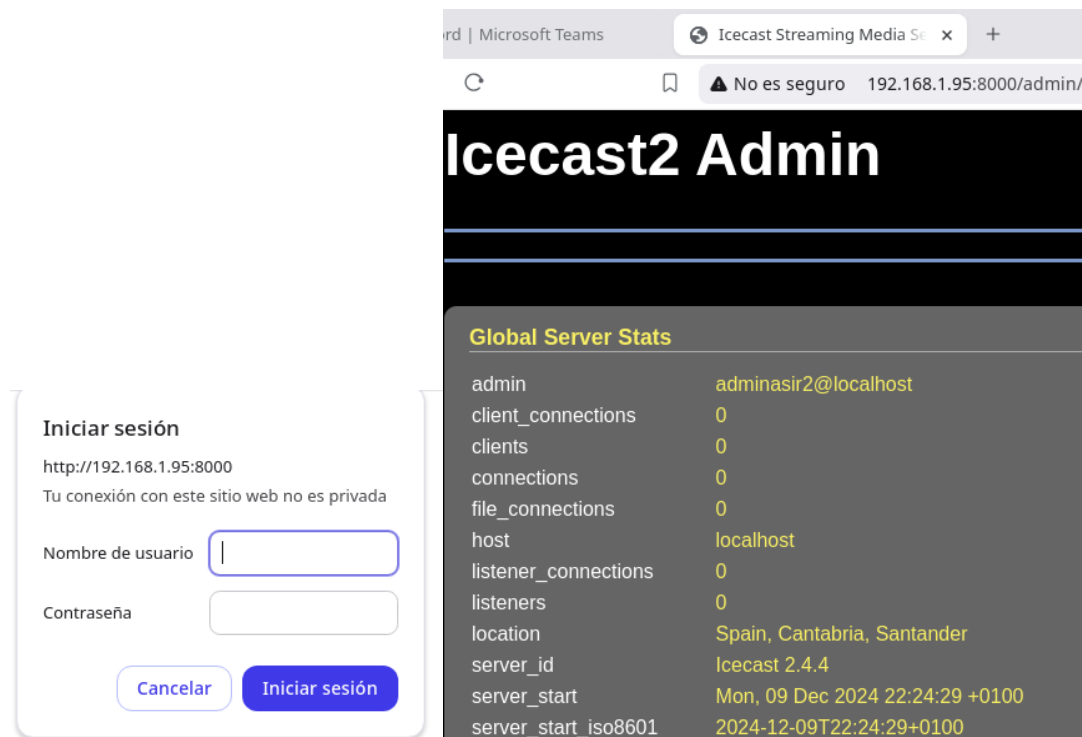
```

Y ya estaría, si queremos probarlo simplemente debemos asegurarnos de probar esto en una entorno en la que la máquina cliente y el servidor estén en la misma red o también podríamos probar esto con una ip pública en el archivo de configuración de icecast2, también necesitaríamos introducir este link en nuestro buscador : <http://192.168.1.95:8000/>

Y nos aparecería esto:

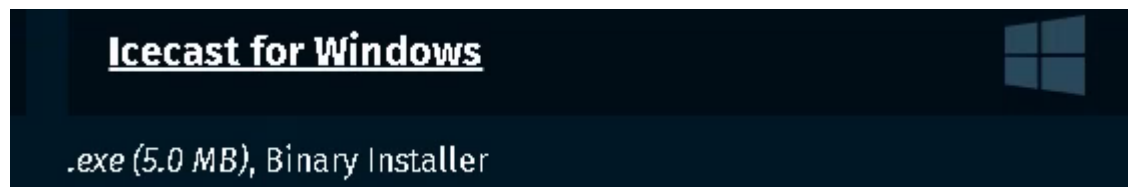


Y si presionamos en **administration**, lo que nos enviará a un formulario pequeño de inicio de sesión, debemos introducir nuestras credenciales (las que antes puse en el archivo de configuración) y nos saldrán datos acerca del servidor

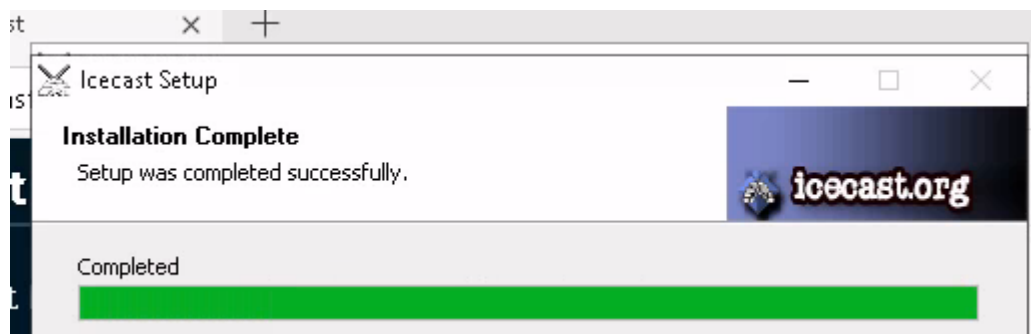


Desarrollo Windows

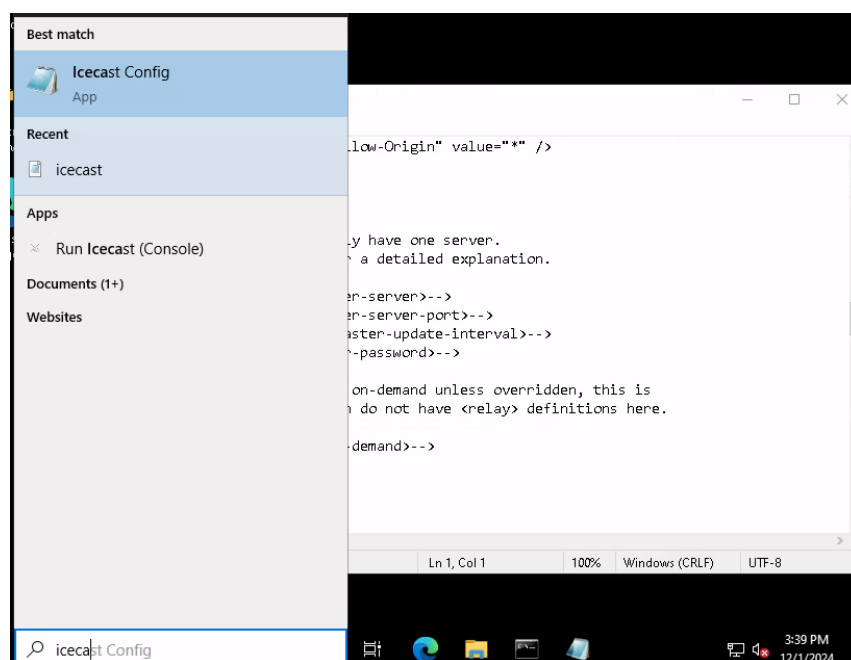
Primero debo descargar Icecast desde la página oficial <https://icecast.org/download>



Lo instalamos



Buscamos **Icecast config** y al abrirlo nos podemos dar cuenta de que es el mismo **archivo de configuración** que en el de Ubuntu



Configuramos la **autenticación**

```
<!-- Admin logs in with the username given below -->
<admin-user>admin</admin-user>
<admin-password>Iesalisa12024</admin-password>
</authentication>
```

Y aquí está el **puerto necesario ACTIVO** y la **ip privada**

```
<!-- You may have multiple <listener> elements -->
<listen-socket>
  <port>8000</port>
  <bind-address>192.168.1.96</bind-address>
  <!-- <shoutcast-mount>/stream</shoutcast-mount> -->
</listen-socket>
```

Y para guardar presionamos **control + S**

Y por último vamos a buscar **Run icecast**, lo **ejecutamos** y ya estaría.

Mejor coincidencia



Run Icecast (Console)

Aplicación

The screenshot shows a Windows Server virtual machine running Icecast2. The browser window displays the Icecast2 Status page at the URL 192.168.1.96:8000. The page has a dark background with the Icecast2 logo and the title 'Icecast2 Status'. Below the title are three tabs: 'Administration', 'Server Status', and 'Version'. The console window shows the Icecast2 startup logs, including warnings about configuration defaults. The command prompt window shows the command 'http://localhost:8000'.

Conclusiones.

En conclusión, este ejercicio me permitió comprender el proceso de implementación de un **servidor de distribución de audio** utilizando **Icecast** en **dos sistemas operativos diferentes**, Ubuntu Desktop y **Windows Server**. A través de la **instalación y configuración** de **Icecast** en ambas plataformas, aprendí sobre los ajustes necesarios, como la **configuración de contraseñas** para el **acceso y administración**, así como la **modificación del archivo de configuración** para asegurar la **conectividad adecuada**.

EJERCICIO 2

Se han reconocido y utilizado formatos de audio digital. Analizar e investigar los archivos de audio digital mas conocidos. Verificar su funcionamiento con un reproductor y realizar pruebas de inserción al servidor de audio implementado en el ejercicio anterior en una de los 2 servidores implementados.

Introducción

En este ejercicio, el objetivo es analizar los formatos de audio digital más reconocidos, probar su funcionamiento en un reproductor de audio y verificar su compatibilidad al insertarlos en el servidor de distribución de audio implementado previamente. Este análisis permitirá determinar las ventajas y desventajas de cada formato en términos de calidad, compresión y compatibilidad.

Desarrollo

Reconocimiento de formatos de audio digital

Realicé una investigación sobre los formatos más utilizados y sus características:



MP3 (MPEG Audio Layer III): Un formato ampliamente compatible, que ofrece compresión con pérdida y es ideal para streaming debido a su bajo tamaño

AAC (Advanced Audio Coding): Ofrece mayor calidad que MP3 a la misma tasa de bits; utilizado por plataformas como YouTube y Apple Music.

OGG Vorbis: Código abierto y con buena calidad; se encuentra en reproductores como VLC.

FLAC (Free Lossless Audio Codec): Ideal para audiófilos; conserva toda la calidad original del audio, pero requiere más espacio y ancho de banda.



Ahora, para hacer la **prueba de reproducción local** en Ubuntu Desktop necesitaré la instalación de un reproductor básico como lo puede ser **VLC**, usaré esta herramienta con **Icecast2**

Lo instalamos

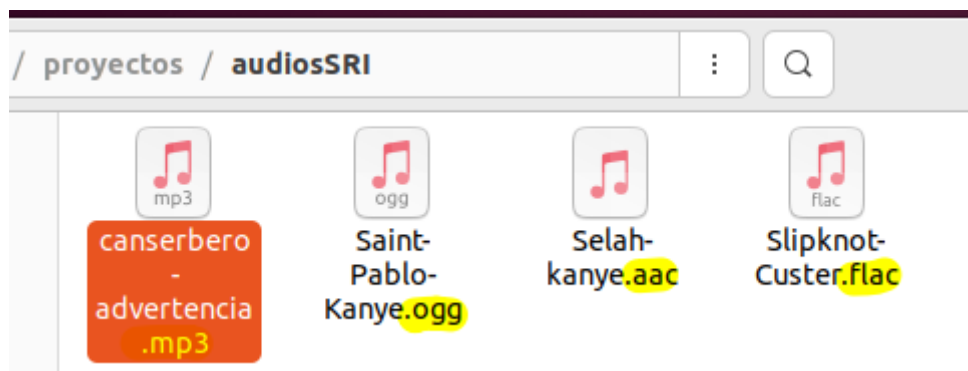

```
root@asir217-VirtualBox: /home/asir217

root@asir217-VirtualBox:/home/asir217# apt install vlc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

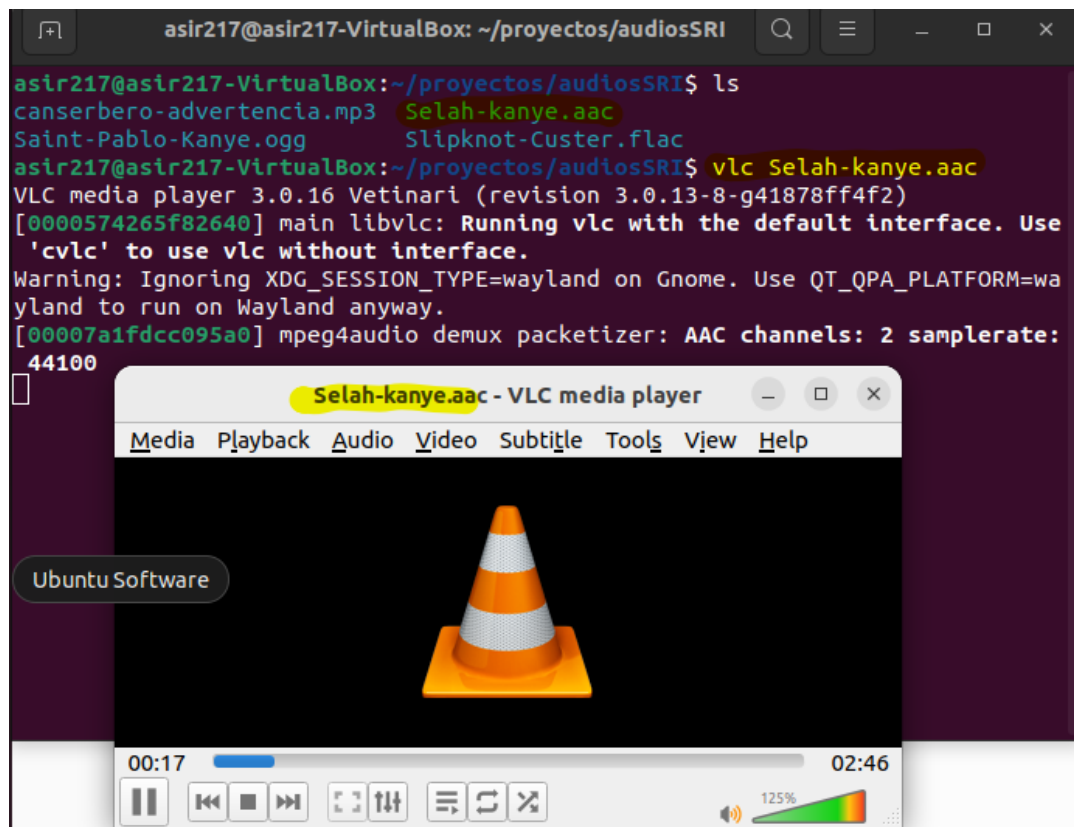
Hago un directorio para los audios

```
root@asir217-VirtualBox:/home/asir217/proyectos# mkdir audiosSRI
root@asir217-VirtualBox:/home/asir217/proyectos#
```

Ahora debo descargar varios formatos de archivos de audio en fuentes confiables porque no quiero un troyanos en mis máquinas y al final estos serían los audios en cuestión:



Los probamos de esta manera:



Cabe recalcar que lo hice con todos los audios y me funcionaron todos.

Posteriormente a todo esto, debo crear una carpeta para alojar tanto los archivos de audio como (más adelante) el archivo xml del feed RSS y también le daremos los permisos adecuados

```
root@asir217-VirtualBox:/var# cd
root@asir217-VirtualBox:~# mkdir -p /var/www/audio
root@asir217-VirtualBox:~# chown -R www-data:www-data /var/www
root@asir217-VirtualBox:~# chmod -R 755 /var/www
root@asir217-VirtualBox:~#
```

Y copio los audios hacia esa carpeta recién creada

```
root@asir217-VirtualBox:/home/asir217/proyectos/audiosSRI# cp *.mp3 *.ogg *.aac *.flac /var/www/audio
root@asir217-VirtualBox:/home/asir217/proyectos/audiosSRI# cd /var/www/audio
root@asir217-VirtualBox:/var/www/audio# ls
canserbero-advertencia.mp3 Saint-Pablo-Kanye.ogg Selah-kanye.aac Slipknot-Custer.flac
root@asir217-VirtualBox:/var/www/audio#
```

Vamos al archivo de configuración para asegurarnos de que **Icecast** pueda acceder a estos archivos y ponerlos a disposición para el streaming.

```
root@asir217-VirtualBox:/var/www/audio# nano /etc/icecast2/icecast.xml
```

```

</security>

<!-- PUNTOS DE MONTAJE DE AUDIOS -->
<mount>
  <mount-name>/canserbero-advertencia.mp3</mount-name>
  <file>/var/www/audio/canserbero-advertencia.mp3</file>
  <type>audio/mpeg</type>
</mount>

<mount>
  <mount-name>/Saint-Pablo-Kanye.ogg</mount-name>
  <file>/var/www/audio/Saint-Pablo-Kanye.ogg</file>
  <type>audio/ogg</type>
</mount>

<mount>
  <mount-name>/Selah-kanye.aac</mount-name>
  <file>/var/www/audio/Selah-kanye.aac</file>
  <type>audio/aac</type>
</mount>

<mount>
  <mount-name>/Slipknot-Custer.flac</mount-name>
  <file>/var/www/audio/Slipknot-Custer.flac</file>
  <type>audio/flac</type>
</mount>
</icecast>

```

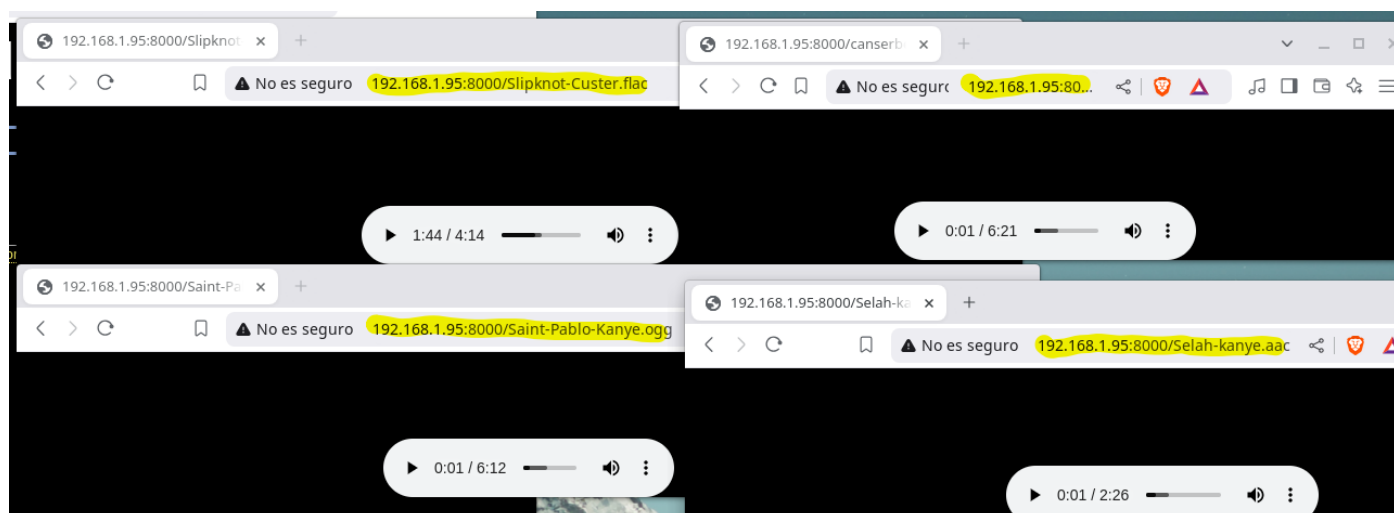
Y esto no me funcionó debido a que busqué esta ruta:

<http://192.168.1.95:8000/Slipknot-Custer.flac>

Y me salió un error 404, me di cuenta de que es porque en el archivo de configuración el WEBROOT es

/usr/share/icecast2/web y no **/var/www/audio** para solucionar esto simplemente copié los archivos que había dentro de “audios” y los copié dentro de **/usr/share/icecast2/web**, luego reinicié el icecast

Aquí podemos ver el funcionamiento de los audios en el servidor



Conclusiones.

En este ejercicio, se analizaron los formatos de audio digital más conocidos, como **MP3, AAC, OGG Vorbis y FLAC**, evaluando su calidad, compresión y compatibilidad en un reproductor local con VLC. Tras **instalar y configurar Icecast2**, se realizaron **pruebas de inserción de los archivos de audio en el servidor**, aunque inicialmente se presentó un problema con la ruta de acceso, que se solucionó copiando los archivos a la ubicación correcta en el servidor. Al corregir este detalle, todos los formatos de audio funcionaron correctamente, lo que destacó la importancia de una configuración precisa para el streaming y me permitió comprender mejor la relación entre los formatos y la transmisión de audio en servidores.

EJERCICIO 3

Se han utilizado técnicas de sindicación y suscripción de audio. En uno de los 2 servidores implementado en el ejercicio 1 generar canales RSS o feeds para la sindicación y suscripción de seguidores al canal.

Introducción

El objetivo de este ejercicio es implementar un sistema de sindicación y suscripción de contenido de audio utilizando canales RSS en el servidor configurado previamente con Icecast. A través de este sistema, los usuarios podrán acceder y suscribirse al contenido ofrecido, recibiendo actualizaciones automáticas cada vez que se agregue nuevo material al canal. Este ejercicio permite comprender cómo los feeds RSS pueden integrarse en un entorno de distribución de audio.

Desarrollo

Primero que nada inicié creando un archivo que es el que define el canal RSS, este archivo es el que contiene información básica del canal como lo puede ser el **título, descripción, enlace y una lista de elementos que representan los audios disponibles**.

Tengo que meterme en esta ruta:

```
root@asir217-VirtualBox:~# cd /usr/share/icecast2/web/  
root@asir217-VirtualBox:/usr/share/icecast2/web#
```

Crear el archivo necesario que se llamará feed.rss

```
root@asir217-VirtualBox:/usr/share/icecast2/web# nano feed.rss
```

Y este será su contenido, que como ya mencioné antes sólo es información básica del canal:

```
root@asir217-VirtualBox: /usr/share/icecast2/web
GNU nano 6.2 feed.rss
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>Canal de Audio ASIR</title>
    <link>http://192.168.1.95:8000/</link>
    <description>Canal de distribución de audio de ASIR 2</description>
    <language>es-es</language>
    <item>
      <title>Slipknot - Custer FLAC</title>
      <link>http://192.168.1.95:8000/Slipknot-Custer.flac</link>
      <description>Audio en formato FLAC de alta calidad.</description>
      <pubDate>Mon, 24 Dec 2024 12:00:00 +0000</pubDate>
    </item>
    <item>
      <title>Selah-kanye AAC</title>
      <link>http://192.168.1.95:8000/Selah-kanye.aac</link>
      <description>Archivo de audio de prueba en formato .aac.</description>
      <pubDate>Tue, 25 Dec 2024 12:00:00 +0000</pubDate>
    </item>
    <item>
      <title>Saint Pablo Kanye OGG</title>
      <link>http://192.168.1.95:8000/Saint-Pablo-Kanye.ogg</link>
      <description>Archivo de audio de prueba en formato .ogg.</description>
      <pubDate>Tue, 25 Dec 2024 12:00:00 +0000</pubDate>
    </item>
    <item>
      <title>Canserbero Advertencia MP3</title>
      <link>http://192.168.1.95:8000/canserbero-advertencia.mp3</link>
      <description>Archivo de audio de prueba en formato MP3.</description>
      <pubDate>Tue, 25 Dec 2024 12:00:00 +0000</pubDate>
    </item>
  </channel>
```

Como último paso en esta configuración sólo nos queda reiniciar el icecast2

```
t@asir217-VirtualBox:/usr/share/icecast2/web# sudo systemctl restart icecast2
t@asir217-VirtualBox:/usr/share/icecast2/web#
```

Ahora pasamos a la fase de **pruebas** de las configuraciones que hemos hecho antes.

Simularé que soy un cliente y entraré a : <http://192.168.1.95:8000/feed.rss> después de que entre se me va a descargar un archivo

Historial de descargas recientes



feed.rss

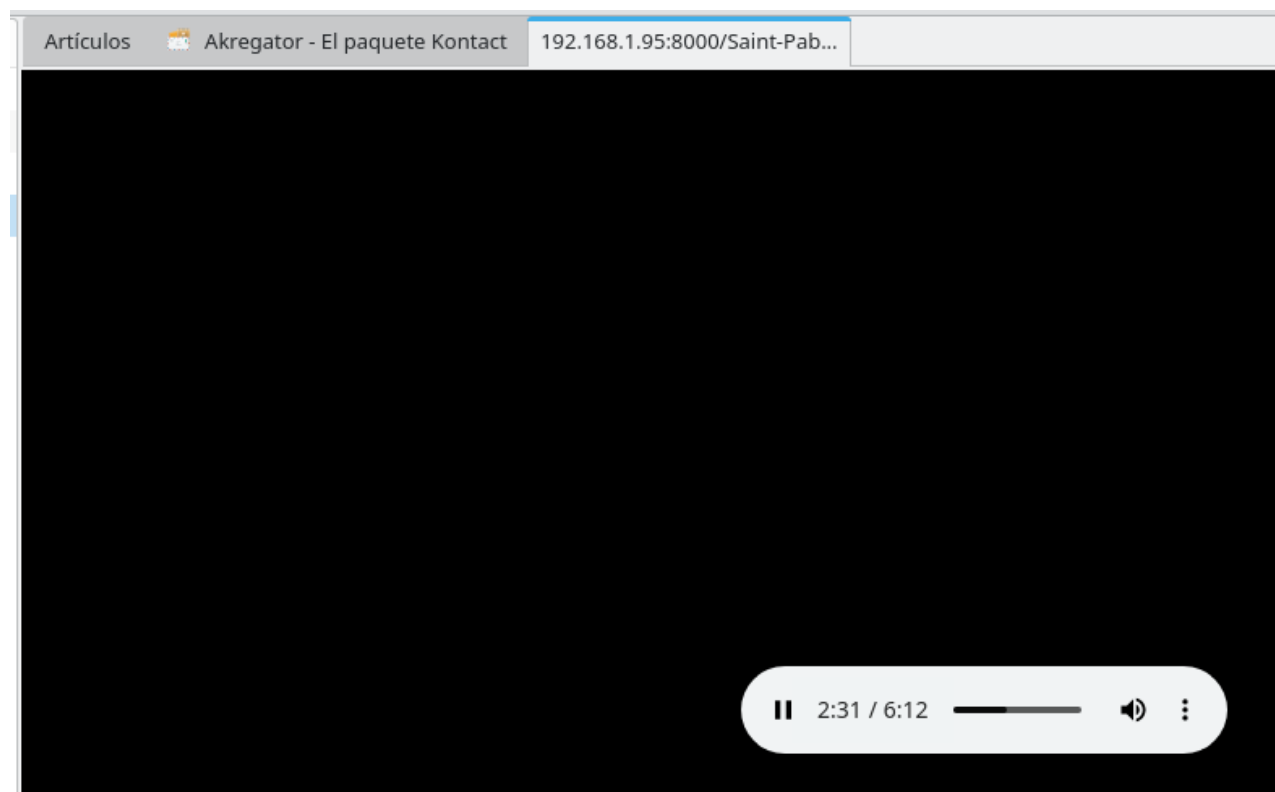
1.333 B • Hace 8 minutos

Debido a que uso debian dicho archivo se me abrirá con una aplicación que ya por defecto tengo instalada, llamada **AKREGATOR**, ella nos va a permitir ver el sitio RSS.

Lo abrimos y podremos ver esto:



Desde Akregator, seleccioné uno de los audios listados y lo reproduje. Verifiqué que el streaming funcionaba correctamente, confirmando que la sindicación y suscripción estaban operativas.



Conclusiones.

Este ejercicio me permitió aprender cómo crear e implementar un canal RSS para distribuir contenido de audio, lo que amplía las posibilidades de compartir recursos con los usuarios de manera eficiente. La utilización de Icecast y la creación de un archivo XML estructurado fueron clave para lograr la sindicación. Además, comprendí la importancia de los estándares RSS en la automatización y accesibilidad de los contenidos digitales