



ESCUELA SUPERIOR POLITÉCNICA DEL
LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN

**“ESTUDIO COMPARATIVO DE DOS SISTEMAS DE
TRANSMISIÓN AUDIO STREAMING”**

TESINA DE SEMINARIO

Previo a la obtención del título de:

INGENIERO EN TELEMÁTICA

PRESENTADO POR:

SANTIAGO VINICIO MORA TINOCO

CHRISTIAN ISMAEL NARANJO MARIÑO

Guayaquil – Ecuador

2014

AGRADECIMIENTO

A la Ing. Patricia Chávez por ser más que una profesora una guía en nuestra formación profesional y contribuir con el desarrollo de este trabajo y a todas aquellas personas que contribuyeron para realización de este trabajo nuestros más sinceros agradecimientos.

DEDICATORIA

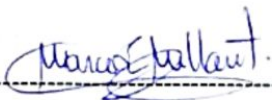
A mis padres por ser pilar fundamental para mi desarrollo como persona e inculcarme valores éticos bien definidos, a mis profesores por ser parte esencial en mi formación profesional y a mi familia en general por brindarme todo el apoyo que necesite para cumplir mis metas.

Santiago Mora

A mis padres por el apoyo y comprensión que siempre me han brindado, a mis hermanos que me impulsan a seguir adelante y a las personas que han estado conmigo en esta etapa importante de mi vida.

Christian Naranjo

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in blue ink, reading "Marcos Millán", written over a horizontal dashed line.

Ing. Marcos Millán
PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in blue ink, reading "Patricia Chávez", written over a horizontal dashed line.

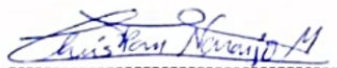
Ing. Patricia Chávez.,MSEE
PROFESORA DELEGADA POR LA UNIDAD ACADÉMICA

RESUMEN

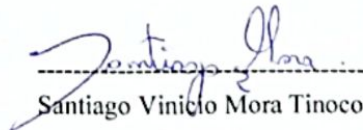
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)



Christian Ismael Naranjo Mariño.



Santiago Vinicio Mora Tinoco.

RESUMEN

El presente trabajo trata sobre el manejo e implementación de dos sistemas para la transmisión en vivo de audio, para dar a conocer su funcionamiento y composición de los mismos. Además se detallan los pasos a seguir para el diseño de ambos sistemas de transmisión en vivo de audio.

Para la implementación se utilizó herramientas de software y conocimientos básicos de redes de datos, estadística y del sistema operativo Linux. De igual manera se definieron las variables a evaluar y con esto proceder a realizar las respectivas pruebas en cada uno de los sistemas mediante el uso del software Jperft.

Luego de las pruebas, se efectuó un análisis estadístico de las muestras para evaluar las variables que influyen en el comportamiento de un sistema de transmisión en vivo de audio, con dicho análisis se realizó un estudio comparativo de ambos sistemas evaluados por cada variable.

ÍNDICE GENERAL

RESUMEN.....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS	xv
ABREVIATURAS	xvi
INTRODUCCIÓN.....	xviii

CAPÍTULO 1

Generalidades

1.1. Antecedentes.....	2
1.2. Justificación.....	3
1.3. Objetivos.....	4
1.3.1 Objetivo General.....	4

1.3.2 Objetivos Específicos.....	5
1.4 Metodologías.....	5
1.5 Resultados Esperados.....	6
1.6 Alcance.....	7
1.7 Limitaciones.....	7

CAPÍTULO 2

Tecnología Streaming

2.1 Definición de Streaming.....	9
2.2 Características.....	9
2.3 Aplicaciones y Usos.....	10
2.4 Protocolos.....	14
2.4.1 HTTP.....	14
2.4.2 RTSP.....	15
2.4.3 RTP.....	16
2.4.4 RTCP.....	17

2.4.5 TCP.....	18
2.4.6 UDP.....	19
2.5 Arquitecturas.....	20
2.5.1 Arquitectura Cliente-Servidor.....	21
2.5.2 Arquitectura sin Servidor.....	26
2.5.3 Arquitectura sin Cliente.....	27
2.6 Modalidades.....	28
2.6.1 En directo.....	28
2.6.2 Bajo Demanda.....	29

CAPÍTULO 3

Implementación de un Sistema de Audio Streaming

3.1 Servidor Streaming.....	32
3.1.1 Shoutcast Server.....	32
3.1.2 Icecast Server.....	33
3.2 Fuentes.....	35
3.3 Clientes.....	39

CAPÍTULO 4

Realización de Pruebas y Obtención de Resultados

4.1 Pruebas realizadas en Sistema 1 (Servidor Icecast 2 – Fuente Mixxx).....	44
4.2 Pruebas realizadas en Sistema 2 (Servidor Icecast 2 – Fuente IDJC).....	49

CONCLUSIONES Y RECOMENDACIONES.....54

BIBLIOGRAFÍA.....57

ANEXO A: Diseño del Primer Sistema de Transmisión de Audio Streaming

ANEXO B: Diseño del Segundo Sistema de Transmisión de Audio Streaming

ÍNDICE DE FIGURAS

Figura 1.1	Metodología usada en la realización de este trabajo.....	6
Figura2.1	Comparación del sistema tradicional y un sistema de streaming.....	10
Figura 2.2	Aplicaciones que usan streaming.....	13
Figura 2.3	Protocolos usados en un sistema streaming.....	19
Figura 2.4	Arquitectura básica de un sistema streaming.....	20
Figura 2.5	Componentes de la Arquitectura Streaming Cliente-Servidor..	25
Figura 2.6	Arquitectura Streaming Sin Servidor.....	26
Figura 2.7	Arquitectura Streaming Sin Cliente.....	27
Figura 4.1	Sistema 1: Datos estadísticos de la variable transferencia.....	47
Figura 4.2	Sistema 1: Curva de la Normal de la variable transferencia...	48
Figura 4.3	Sistema 1: Datos estadísticos de la variable ancho de banda.....	49
Figura 4.4	Sistema 1: Curva de la Normal de la variable ancho de banda.....	50

Figura 4.5	Sistema 1: : Datos estadísticos de la variable variación de retardo	50
Figura 4.6	Sistema 1: : Curva de la Normal de la variable variación de retardo	51
Figura 4.7	Sistema 2: Datos estadísticos de la variable transferencia....	52
Figura 4.8	Sistema 2: Curva de la Normal de la variable transferencia...	53
Figura 4.9	Sistema 2: Datos estadísticos de la variable ancho de banda.....	53
Figura 4.10	Sistema 2: Curva de la Normal de la variable ancho de banda.....	54
Figura 4.11	Sistema 2: : Datos estadísticos de la variable variación de retardo	55
Figura 4.12	Sistema 2: : Curva de la Normal de la variable variación de retardo	55
Figura A.1	Instalación icecast2	
Figura A.2	Modificación del archivo Icecast.xml	
Figura A.3	Contenido del archivo Icecast.xml	
Figura A.4	Archivo Icecast.xml con las modificaciones realizada.	

- Figura A.5 Modificación del archivo icecast2
- Figura A.6 Permitir la habilitación del servicio Icecast2
- Figura A.7 Activación de icecast2
- Figura A.8 Servicio icecast2 activo
- Figura A.9 Instalación de Mixxx
- Figura A.10 Agregación de los repositorios
- Figura A.11 Actualización de la Instalación de Mixxx
- Figura A.12 Instalación de los codecs de audio para Mixxx
- Figura A.13 Instalación de paquetes Mixxx
- Figura A.14 Comprobación de la instalación de Mixxx
- Figura A.15 Configuración de Mixxx
- Figura A.16 Ingreso de datos en la configuración de Mixxx
- Figura A.17 Activar la emisión en vivo de Mixxx
- Figura A.18 Vista de la interfaz gráfica del programa Mixxx emitiendo
- Figura B.1 Edición del archivo limits.config
- Figura B.2 Agregación del texto al archivo limits.config

Figura B.3 Inclusión del usuario al grupo de audio

Figura B.4 Creación de archivo para Jack

Figura B.5 Instalación de QJACK

Figura B.6 Instalación de IDJC

Figura B.7 Interfaz gráfica de IDJC

Figura B.8 Configuración del servidor para IDJC

Figura B.9 Configuración de los formatos para IDJC

Figura B.10 Configuración de la información para cliente de IDJC

Figura B.11 Activar la emisión en vivo para IDJC

ÍNDICE DE TABLAS

Tabla I	Media obtenida de las 3 variables con muestras del Sistema 1	
	52
Tabla II	Media obtenida de las 3 variables con muestras del Sistema 2	
	56

ABREVIATURAS

AAC: Del inglés (Advanced Audio Coding) Protocolo de Codificación de audio avanzado.

ALSA: Del inglés (Advanced Linux Sound Architecture) Arquitectura avanzada de sonido de Linux.

AMR: Del inglés (Adaptive Multi-Rate) Codec de Audio Multi-Tasa Ajustable.

FTP: Del inglés (File Transfer Protocol) Protocolo de Transferencia de Archivo.

HTTP: Del inglés (Hypertext Transfer Protocol) Protocolo de Transferencia de Hipertexto.

MIDI: Del inglés (Musical Instrument Digital Interface) Interfaz digital de instrumentos musicales.

MMS: Del inglés (Microsoft Media Server) Servidor de multimedia de Microsoft.

MP3: Del inglés (Moving Pictures ExpertsGroup - MPEG-1) Protocolo de Codificación de Audio Digital.

Ogg Vorbis: Códec de Audio Digital que utiliza contenedor ogg.

OSS: Del inglés (Open Sound System) Protocolo de Sistema de audio libre.

Real Audio: Formato de audio propietario desarrollado por RealNetworks.

RTCP: Del inglés (Real time Control Protocol) Protocolo de Control en Tiempo Real.

RTMP: Del inglés (Real Time Messaging Protocol) Protocolo de mensajería en tiempo real.

RTP: Del inglés (Real-time Transport Protocol) Protocolo de Transporte de Tiempo Real.

RTSP: Del inglés (Real Time Transmisión en vivo Protocol) Protocolo de Flujo en Tiempo Real.

TCP: Del inglés (Transmission Control Protocol) Protocolo de Control de Transmisión.

UDP: Del inglés (User Datagram Protocol) Protocolo de Datagrama de Usuario.

WAV: Del inglés (Waveform Audio File Format) Formato para almacenar sonido en archivos desarrollado por Microsoft e IBM.

WMA: Del inglés (Windows Media Audio) Tecnología de compresión de archivos de audio desarrollada por Microsoft.

INTRODUCCIÓN

Todo sistema de audio transmisión en vivo trabaja principalmente con una arquitectura basada en servidor, fuentes y clientes, dentro de los mismos existen infinidad de estos tres componentes; con lo cual se podrá obtener variedad de sistemas de transmisión en vivo con diferentes servidores, clientes y fuentes.

Los primeros capítulos del presente trabajo explican conceptos básicos y esenciales para entender el funcionamiento y composición de la tecnología transmisión en vivo. La siguiente parte será la explicación de su implementación y los pasos que se siguieron para la misma, en la cual también se explicará los diversos tipos de componentes que podremos usar para aquello.

Y en el capítulo final se realizara las respectivas pruebas de rendimiento y eficiencia necesarias para analizar y visualizar el comportamiento de estos sistemas.

CAPÍTULO 1

Generalidades

En este capítulo se dará a conocer todas las pautas para la realización de nuestro trabajo tales como que queremos lograr con el mismo, que dificultades se presentaron durante su desarrollo y el porqué de la realización de esta tesina.

1.1 Antecedentes

El requerimiento del ser humano de tener a su disposición numerosa información tal como periódicos, revistas, fotos, radio, videos, música, etc., sumado a la constante evolución de las tecnologías ha generado a los medios de comunicación la necesidad de encontrar el/los medios necesarios para poder satisfacer las necesidades de sus clientes, y facilitar la tarea de acceder a la información. Este proyecto busca explicar de manera detallada el proceso de implementación y realización de pruebas de rendimiento y eficiencia de un sistema de transmisión en vivo de audio, por la cual enunciaremos los conceptos, funcionamientos, ventajas y desventajas que obtenemos al usar dicho sistema.

La aparición de la transmisión en vivo como tecnología se remonta al año 1995 para la transmisión de audio, y poco tiempo después para la transmisión de video, permite acceder a contenido audiovisual sin tener la necesidad de descargarlo previamente. Para poder realizar dicho tipo de transmisión, fue necesaria la creación de algoritmos de compresión, también conocidos como “codecs”, los mismos que codifican los datos de audio y video para mejorar la calidad de transmisión y tratar de disminuir el consumo de recursos.

La constante búsqueda de alternativas para la libre emisión de información multimedia sumado al ofrecimiento de proveedores de servicios de Banda Ancha a bajo costo ha permitido el crecimiento del uso de transmisión en vivo como tecnología de transmisión de señales multimedia. A pesar del crecimiento de usuarios a nivel mundial con acceso a Banda Ancha, la transmisión en vivo no ha sido explotado con nuevos servicios, pudiendo ser un factor el desconocimiento de tecnologías de bajo costo, es por esto que usualmente se implementa en aplicaciones libres como Linux. Cabe resaltar que dicha falta de crecimiento de la transmisión en vivo se debe a varias circunstancias tales como problemas con derechos de autor, muchos software y licencias propietarias impiden que nuevos usuarios puedan implementar esta tecnología, además de que en ciertos casos se prefiere las redes P2P ya que potencian la descarga.

1.2 Justificación

La justificación de un sistema de transmisión en vivo de audio se puede dar por los siguientes escenarios:

1. No tener disponible un dispositivo de radio.
2. Desconocer la frecuencia con la que dicha radio emite en un área desconocida.

3. Estar fuera de los límites de emisión (ciudad, país) de una estación de radio requerida.
4. Expandir a nivel geográfico el alcance que posee actualmente la radio implementada.

Por los escenarios antes mencionados es ineludible la implementación de un sistema de transmisión en vivo de audio para satisfacer las necesidades de información del usuario que posea conexión a la red, y así conseguir un nuevo medio de transmisión.

Debido a esto, es necesario analizar las herramientas necesarias para realizar la implementación de un sistema de transmisión en vivo de audio y en base a la evaluación determinar la mejor elección.

1.3 Objetivos

En este subcapítulo se tratarán los diversos objetivos planteados en la realización de nuestro trabajo estos se detallan a continuación:

1.3.1 Objetivo General

- Implementar un sistema de transmisión en vivo de audio mediante el uso de distintos software para realizar transmisiones en vivo en la red.

1.3.2 Objetivos Específicos

1. Implementar dos sistemas de transmisión en vivo de audio para realizar transmisiones en vivo en la red.
2. Recopilar datos de las variables evaluadas en pruebas efectuadas sobre ambos sistemas de transmisión en vivo de audio
3. Evaluar los resultados obtenidos en cada sistema de transmisión en vivo de audio, utilizando medidas estadísticas.

1.4 Metodologías

El método analítico es el método más apropiado para realizar nuestro estudio, ya que nos permite presentar tanto información cualitativa como cuantitativa.

Además nuestra investigación se respaldara con la técnica de pruebas de rendimiento y eficiencia que nos permite realizar un estudio de desempeño comparativo. La misma nos facilita medir el rendimiento de un sistema, y luego de compararlo con algún parámetro, poder realizar el respectivo análisis comparativo de software para un sistema de transmisión en vivo de audio.

Al hacer esto podremos comparar los beneficios y/o perjuicios que tendremos al utilizar uno u otro software, para poder estimar el rendimiento del sistema. La metodología es observada en la Figura 1.1

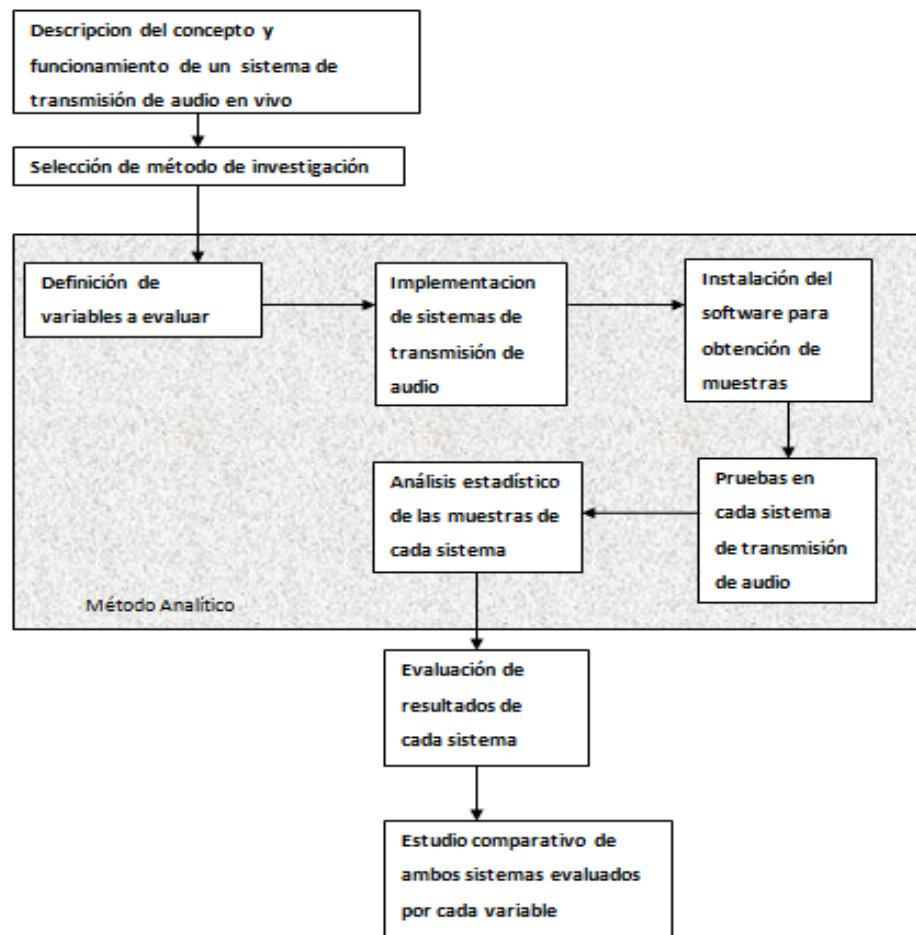


Figura 1.1 Metodología usada en la realización de este trabajo

1.5 Resultados Esperados

Demostrar de manera objetiva los beneficios de utilizar la tecnología de transmisión en vivo para la transmisión de audio.

1.6 Alcance

Implementar dos sistemas de transmisión en vivo de audio, a los cuales se les realizara diferentes pruebas de rendimiento que nos permitirán conocer las falencias que podrían presentar dichos sistemas.

Transmitir nuestra radio a través de la red para que el oyente pueda acceder a la misma, para lo cual utilizaremos una IP (privada) brindada por el proveedor de servicio de Internet, dependiendo del entorno de transmisión.

1.7 Limitaciones

No podremos transmitir a un número excesivo de oyentes, dado que no se cuenta con un ancho de banda ni servidor que nos permita realizar transmisiones a gran escala.

No podremos implementar el sistema de audio transmisión en vivo con determinados software tales como Shoutcast Server y Muse, debido a costos de licencia que están fuera de nuestro alcance.

CAPÍTULO 2

TECNOLOGIA STREAMING

Este capítulo trata principalmente de familiarizarnos con los conceptos y características principales que tienen que ver con la tecnología streaming y dar a conocer los aspectos más importantes de la composición y funcionamiento de la misma.

2.1 Definición de Streaming

Detalla una metodología que permite realizar un mejor control del contenido multimedia que puede ser audio o video y poder transmitirlo a través de la red en tiempo real sin que el usuario tenga que antes llevar a cabo la descarga de dichos contenidos.

Esta tecnología realiza la descarga de los datos a medida que se los va transmitiendo al usuario lo que agiliza en proceso de intercambio de información con este. Con esto se evitaría el lento proceso de primero descargar el archivo y luego reproducirlo además permite el ahorro de espacio en nuestros dispositivos o unidades de almacenamiento.

2.2 Características

No se requiere la descarga total del archivo multimedia en una unidad de disco local para poder reproducir dicho archivo. También puede adaptarse según los requerimientos de ancho de banda, además no es tan necesario la utilización de la web ya que los datos pueden ser transmitidos por redes móviles o intranets privadas.

Controlar el flujo de usuarios que se puede tener al transmitir la información, la misma que se va guardando en un búfer de

almacenamiento que permite una reproducción continua de los archivos, como se puede apreciar en la Figura 2.1.

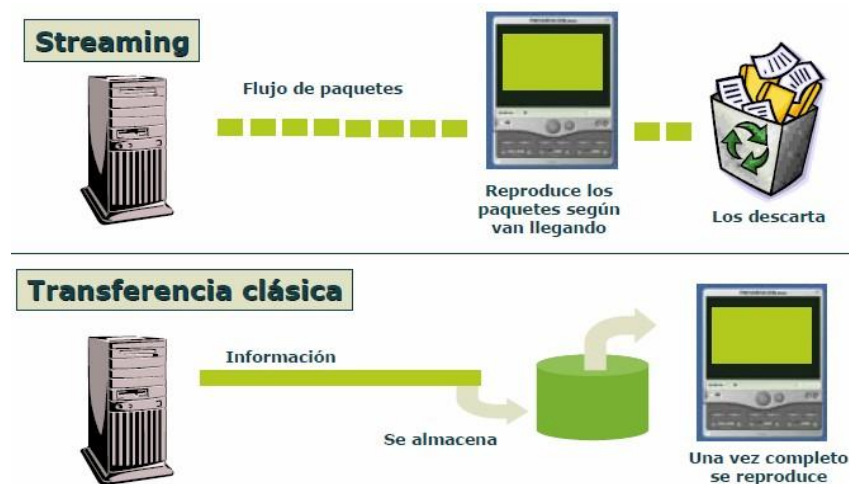


Figura 2.1 Comparación del sistema tradicional y un sistema de streaming [1]

2.3 Aplicaciones y usos

Este tipo de tecnología permite la utilización y desarrollo de un sin número de nuevos servicios los cuales se puede observar en la Figura 2.2, además de mejorar la calidad y desempeño de algunas aplicaciones antiguas y facilitar el uso de las mismas.

Entre algunas de las aplicaciones de la transmisión en vivo tenemos:

- a. La radio por internet es uno de los principales servicios del uso del transmisión en vivo ya que permite que las radios puedan

realizar sus emisiones por la red y poder llegar a millones de usuarios a los que no podrían llegar debido a que las señales que ellos emiten son muy condicionadas por el alcance de la mismas y las condiciones climáticas y geográficas que afectan la propagación de dicha señal. Con la transmisión en vivo se soluciona dicho problemas y permite también a millones de personas realizar sus propias producciones y compartirlas por la red.

- b. La televisión por internet al igual que la radio es otro de los servicios del transmisión en vivo pero este aun con más acogida debido a que se realizan reproducciones de video ya que la mayoría de los por lo regular buscan darse a conocer y obtener seguidores y ganar fama esto ha servido para que grandes figuras den a conocer sus talentos. Además de esto permite una información continua de acontecimientos importantes a nivel mundial y que sean difundidos de manera global en el instante en que sucedan.
- c. Grandes empresas usan la transmisión en vivo para realizar sus sesiones o reuniones con todos socios en las distintas ciudades del mundo en donde se encuentre la empresa.

- d. El servicio de transmisión en vivo nos brindara la suficiente confiable para que los médicos de forma remota realicen consultas entre sí y con los pacientes, así como dar consejos durante la operación. Esto es un gran beneficio para todos los centros médicos ubicados en zonas escasamente pobladas. Las consultas y el contacto con los pacientes a través de Internet ahorra tiempo y los gastos de traslado del paciente a un centro urbano para el tratamiento.
- e. Es un gran aporte para el ámbito de la educación ya que a través de esta tecnología se puede implementar clases en línea de esta manera el estudiante podrá recibir su clase a pasar del distanciamiento que tenga a un centro educativo y a cualquier hora del día, además con esto se lograra que las clases sean más interactivas, se podrá adaptar enlaces web hacia páginas de interés, la visualización de videos y demás metodologías para lograr mayor entendimiento por parte del alumnado.
- f. La transmisión en vivo de video también puede ayudar a controlar ubicaciones remotas. Esto es mucho más beneficioso que las actuales cámaras de seguridad normales, debido a que esta tecnología de transmisión en vivo no necesita un circuito cerrado. El video se puede monitorear desde cualquier parte del

mundo, siempre y cuando exista una conexión a Internet cercana.

- g. Transmisión en vivo de tipo multimedia, presenta a todo profesional de la comunicación toda una nueva manera de entregar información, mensajes, y capacitarse. Al aprovechar el Internet, los costos de distribución puede ser mucho más bajo que los medios de comunicación tradicionales. La difusión por Internet con éxito tendrá que armar un polivalente y creativo equipo para producir contenido multimedia en transmisión en vivo de alta calidad. La audiencia Web es implacable, así que el contenido tiene que ser convincente para recibir la pena de visión cifras que darán un retorno de la inversión en transmisión

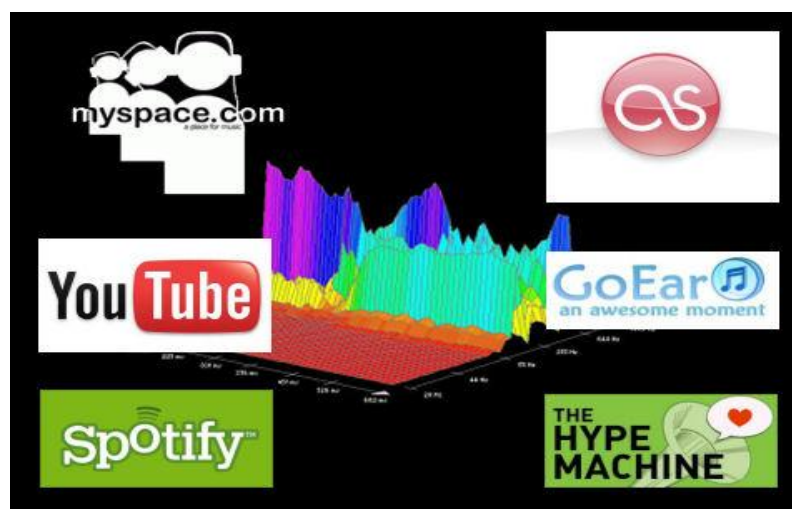


Figura 2.2 Aplicaciones que usan streaming [2]

2.4 Protocolos

Entre los protocolos utilizados por la tecnología de transmisión en vivo tenemos HTTP, RTSP, RTP, RTCP, TCP, UDP como se visualiza en la Figura 2.3

2.4.1 HTTP

Este protocolo me permite el envío de datos desde un servidor web en respuesta a un evento además de establecer la sintaxis y semántica que usaran los distintos elementos de software pertenecientes a la arquitectura web para comunicarse. Permite la comunicación a nivel de aplicación entre el navegador y el servidor web, en los sistemas de información hipermedia distribuidos y colaborativos. Estas comunicaciones constan de peticiones y respuestas.

HTTP es usado como un protocolo genérico para la comunicación entre agentes, proxy, puerta de enlace y de otros sistemas de Internet, incluido soporte para protocolos SMTP, FTP, Gopher y Wais. De esta manera HTTP permite el de acceso básico al contenido multimedia procedente de una amplia gama de aplicaciones.

Una característica de HTTP es que permite que los sistemas se construyan independientemente de la información que se transfiere HTTP se basa en sencillas operaciones de solicitud / respuesta. En la cual un cliente comienza estableciendo una conexión con un servidor y realiza el envío de un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado.

Debido a que el protocolo es sin estado y sin conexión, el servidor web se libera de la sobrecarga de mantener una conexión con el navegador. Esta característica permite a los recursos del servidor de conexión poder manejar muchos usuarios simultáneamente. [13] [14] [15]

2.4.2 RTSP

Es un protocolo de flujo de datos realizados en tiempo real que permite establecer como se hará el envío de información entre el cliente y el servidor. Trabaja al nivel de aplicación controlando la transferencia de datos multimedia a través de ello establece y maneja uno o varios flujos sincronizados de medios continuos en donde cada flujo puede residir en un servidor distinto.

La funcionalidad de RTSP permite a un usuario manejar el flujo de los medios de comunicación a través de comandos:

1. Pausa y reanudación;
2. Rebobinado y avance rápido;
3. Reposicionamiento de la reproducción.

RTSP permite el establecimiento de múltiples sesiones, y se puede seleccionar los distintos protocolos de transporte a utilizar como UDP, RDP y TCP, además de proporcionar un medio para que esté pueda seleccionar un mecanismo de distribución basado en RTP pero siendo independiente del mecanismo de transporte usado para la transmisión continua de los datos. [8] [16] [17] [18] [19]

2.4.3 RTP

Es un protocolo que trata de brindar servicios de transporte a las aplicaciones que requieran transmisiones en tiempo real mediante el uso de redes de paquetes. Este contribuye con la introducción de marcas de tiempo y números de secuencia que permitirán controlar el flujo, la clasificación e identificación de los paquetes que se vayan perdiendo. Trabaja especialmente sobre el protocolo de transporte UDP.

Este protocolo no me brinda confiabilidad en la entrega de los datos ni me asegura que una adecuada calidad de servicio sobre la red. RTP permite asignar a cada fuente su propio flujo de paquetes RTP independiente es decir en el caso de una videoconferencia podría asignar un flujo solo para audio y otro solo para video o ambos en uno solo. **[8] [16] [17] [18] [19]**

2.4.4 RTCP

Controla el transporte de los datos en tiempo real, además se encarga de los procesos de comunicación e información para el correcto control del flujo de datos de RTP.

Al trabajar en conjunto con RTP se logra dar mecanismos de retroalimentación a cada participante en una sesión de RTP con esto se consigue tener un control de la calidad de distribución de los datos, comprobar que los codificadores sean adaptables y diagnosticar cualquier fallo producido en la transmisión.

Los mensajes en esta sesión incluyen informes de recepción, incluyendo el número de paquetes perdidos estos pueden ser enviados tanto por el receptor como por el emisor. **[13] [18] [19]**

Sus paquetes no transportan datos multimedia, sino que trabaja con RTP para el transporte y empaquetado de los datos.

2.4.5 TCP

Es un protocolo destinado al envío seguro y protección contra los errores en la transmisión de paquetes punto a punto es decir proporcionar una conexión y comunicación fiable.

TCP trata en lo posible de prestar un servicio libre de errores para ello agrega una serie de funciones que le permiten realizar una comunicación sin pérdida de paquetes y sin duplicaciones.

Entre los servicios que provee TCP tenemos los siguientes:
Transfiere los datos a través de un canal, brinda confiabilidad en el envío de datos, controla el flujo de paquetes y el tráfico que puede soportar un canal, multiplexación, conexiones lógicas y full dúplex

Las conexiones TCP se componen de tres etapas las cuales son el establecimiento de la conexión, la transferencia de los datos y la finalización de dicha conexión. . [13] [14] [17]

2.4.6 UDP

Es un protocolo de transporte que básicamente ofrece servicios de aplicaciones de usuario entre los cuales se destaca el servicio de envío de datagramas a través de la red sin que sea necesario establecer antes una conexión esto lleva consigo a que no se garantice la entrega de dichos paquetes además de no realizar corrección de errores ni control de flujo como lo hace TCP pero por el contrario permite que las entregas de paquetes de datos desde el servidor a quien reproduce el archivo se hagan con una velocidad mucho mayor. [13] [14] [17]

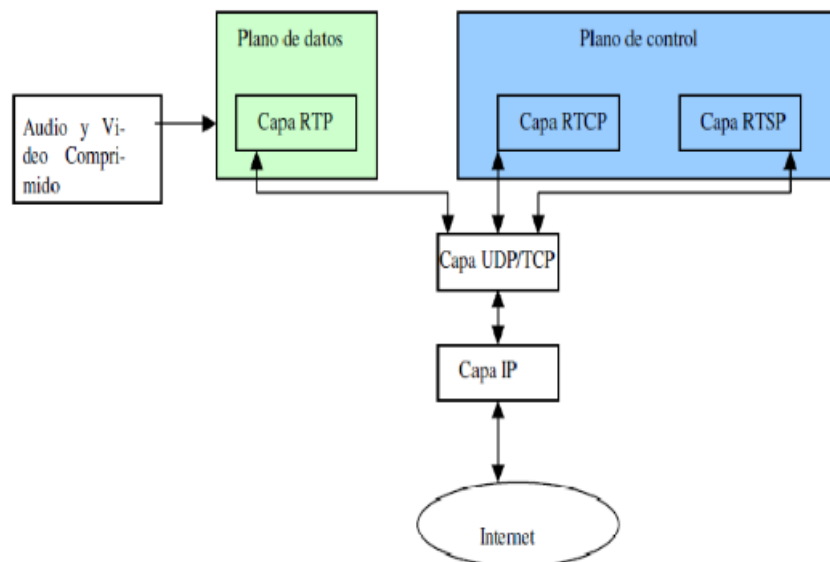


Figura 2.3 Protocolos usados en un sistema streaming [3]

2.5 Arquitecturas

La arquitectura nos permite definir la estructura de un sistema, cómo está compuesto y de qué manera trabaja tal como se muestra en la Figura 2.4.

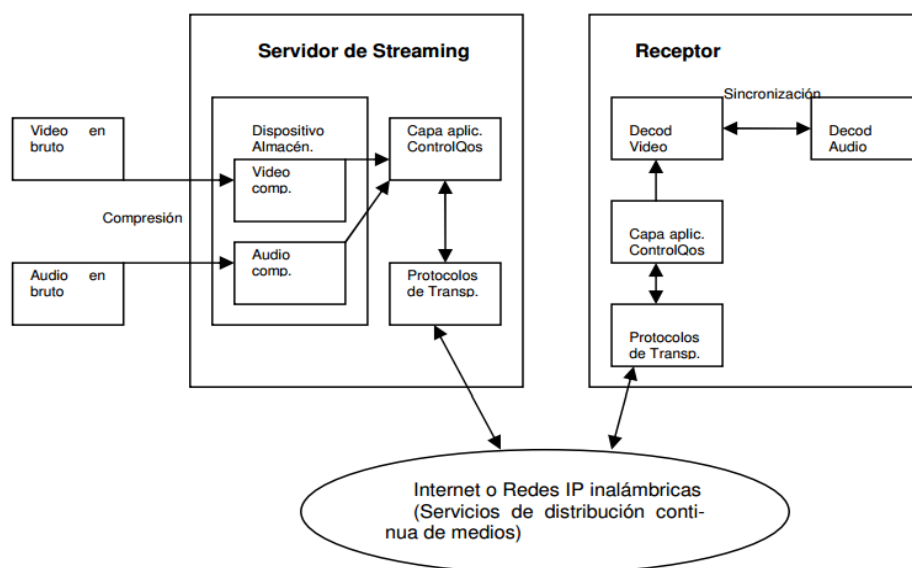


Figura 2.4 Arquitectura básica de un sistema streaming [3]

La tecnología de transmisión en vivo trabaja con diferentes arquitecturas tales como: Arquitectura de Transmisión en vivo Cliente-Servidor, Arquitectura de Transmisión en vivo sin servidor, Arquitectura de Transmisión en vivo sin cliente.

2.5.1 Arquitectura Cliente-Servidor

La transmisión en vivo tiene como principal forma de funcionamiento que los datos en crudo enviados ya sea desde una cámara o micrófono son previamente comprimidos y depositados en un servidor de almacenamiento que posee un software que le permite manejar la entrega en tiempo real.

Una vez almacenada esta información el cliente podrá realizar una solicitud de acceso a los datos, el canal de distribución conecta al cliente con el servidor y este recoge los datos almacenados, la capa de aplicación adecua el flujo de bits y los debidos requerimientos de QoS para que los protocolos de transporte transformen dicho flujo comprimido en paquetes algunos de los cuales podrían ir siendo descartados o llegando con retrasos considerables debido a la congestión del canal después de esto serán transmitidos sobre internet o redes IP.

La arquitectura cliente-servidor trabaja con 4 bloques principales que son:

“Captura y Codificación”; Se toman las señales de audio y video transmitidas en un formato determinado y se las transforma en archivos de tipo transmisión en vivo es decir este material

pasa de una etapa analógica a una de digitalización realizado esto los datos son almacenados en los dispositivos. Esto produce que el gran consumo de ancho de banda que provocan los datos enviados en bruto a través del sistema se reduzca considerablemente, y para permitir que el flujo de datos sea aún mejor se realiza una compresión de los mismos.

Esta compresión de archivos es determinada por un algoritmo que se establece en un software, este es denominado como códec y dicho procedimiento se realiza por unos servidores de codificación.

Un inconveniente al comprimir los datos es que algunas veces se produce una pérdida de paquetes esto podría afectar la calidad de visualización o reproducción del archivo aunque no a gran escala debido a que esta calidad de transmisión depende también en si del contenido ,la resolución , que composición tienen dichos paquetes y el algoritmo a utilizarse por ellos.

Lo que se desea con esta etapa es que se reduzca la velocidad de datos para que coincida con el ancho de banda disponible y no se produzcan consecutivamente estas dificultades en la transmisión.

“Servicio”; Los archivos ya una vez codificados se cargan en un servidor para la entrega a la red. El servidor siempre pasa pendiente de las peticiones que llegue a realizar el usuario, si notifica una petición busca en el directorio apropiado el archivo que fue solicitado y si es encontrado es entregado a dicho usuario usando los protocolos de red apropiados en la transmisión de red que se esté usando.

El servidor es la parte fundamental de esta etapa de la arquitectura ya que es una aplicación de software que procesa los datos multimedia controla en tiempo real el envío de información y demás funciones interactivas estas acciones tienen ciertas restricciones temporales que son realizadas para impedir fallas en el caso de transmisión de video los jerkiness y en audio los pop y con esto se permitirá que suministre de manera apropiada los servicios de audio y video en modo sincronizado.

“Distribución y Entrega”; La distribución es sobretodo un sistema de conmutación de paquetes de datos asíncronos .En esta fase se comprueba la conectividad IP este activa entre el servidor y el cliente eso establece que existe transmisión de paquetes entre ambos.

La red debe de brindar soporte para la distribución de contenido multimedia de buena calidad cuando cuenta con una velocidad de transmisión alta y un tráfico de red descongestionado.

Se implementa el uso del QoS y adaptación al estado de la red que me permiten reducir la pérdida de paquetes y retardos además de esto es necesario el uso de otras aplicaciones como replicación y un almacenamiento en cache para así tratar de disminuir el tráfico en la red.

“Reproductor de Transmisión en vivo”; Un reproductor de tipo transmisión en vivo trabaja de manera autónoma que soporte los formatos de los archivos de transmisión en vivo este recibe la información y la almacena temporalmente para que el usuario los pueda visualizar en el tamaño de ventana que el prefiera por lo general soporta las funciones de reproducir, pausar, rebobinar hacia adelante y hacia atrás.

El usuario puede usar este reproductor como un plug-in en un navegador web y esto lo han hecho un sin número de portales web que ofrecen un contenido específico a través de estos reproductores incorporados en los mismos aquí los archivos son procesados por partes y presentados sin necesidad de ser descargados

“**Reproductor Media**”; Este software es el pilar fundamental para procesar el flujo de datos multimedia que ingresan en las máquinas de destino y posee el decodificador que permite al cliente visualizar la información de manera eficiente y adecuada.

Además de encargarse de realizar la petición de los datos al servidor y una vez recibidos poder ser reproducidos para visualización del cliente. La composición de esta arquitectura es visualizada en la Figura 2.5. [10] [11] [13] [18] [19]

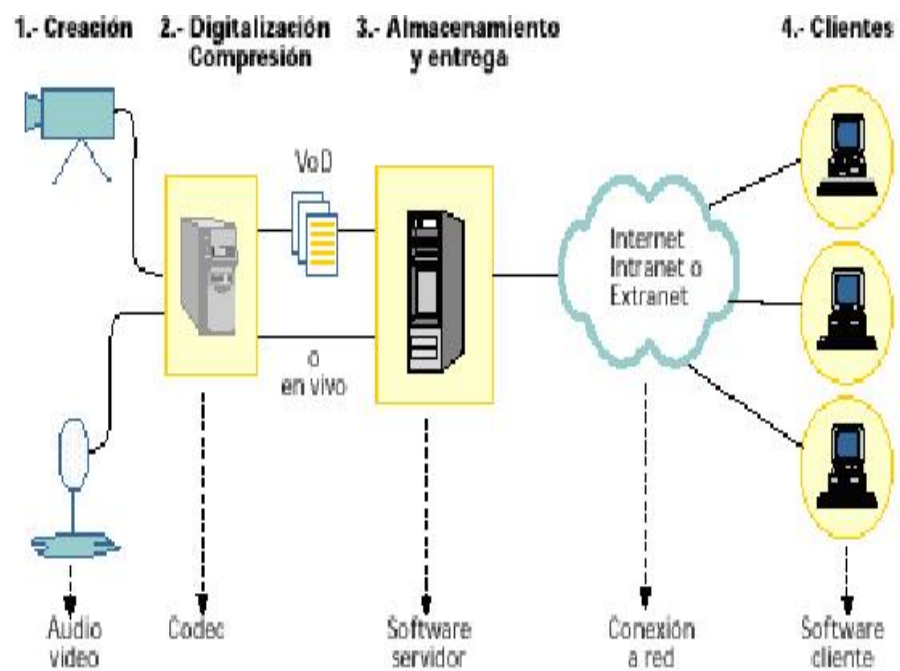


Figura 2.5 Componentes de la Arquitectura Streaming Cliente-Servidor [5]

2.5.2 Arquitectura sin servidor

Como su nombre mismo lo dice es una arquitectura que trabaja sin el uso de servidor alguno ya sea de audio o de video en vez de este usa un servidor de tipo web que realizara algunas de las funciones del servidor multimedia a este servicio se lo conoce como pseudo-streaming o arranque rápido.

En esta arquitectura se obtiene los recursos en los equipos de usuarios finales para el almacenamiento de datos distribuidos y la entrega de datos multimedia. Por ello cuando se añaden nuevos huéspedes para el sistema, estos añaden recursos, además de una carga de trabajo para el sistema esto se puede visualizar en la Figura 2.6.

Entre las características que puede darnos el servidor web contamos con las siguientes que son: Transferencia completa del archivo, no temporizado, no existe un control sobre el flujo de la información, los datos se almacenan en el cliente, usa el protocolo HTTP. **[6] [10]**

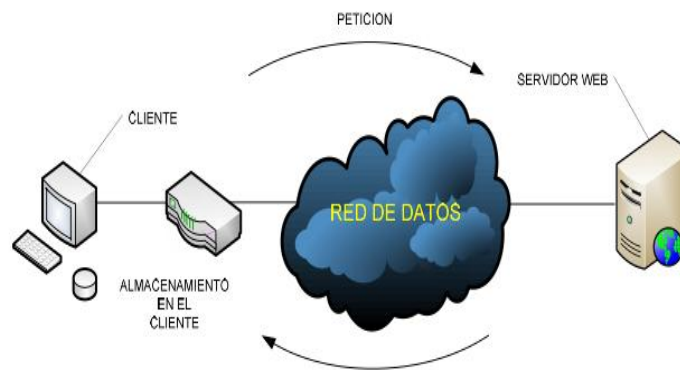


Figura 2.6 Arquitectura Streaming Sin Servidor. [6]

2.5.3 Arquitectura sin Cliente

En este tipo de arquitectura no existe un programa cliente. Simula el funcionamiento de un servicio bajo demanda con flujos de datos en directo. Para visualizar se utiliza un plugin como se muestra en la Figura 2.7. Es frecuentemente usado por las páginas web debido a que brinda la visualización en línea de cualquier tipo de contenido un ejemplo muy evidente de esto es YouTube que brinda al usuario un gran repositorio de videos y que pueden ser vistos sin necesidad de descargarlos.

Esta arquitectura se caracteriza porque la página web puede realizar la función de un servidor de almacenamiento de contenido en memorias temporales. [6] [10]



Figura 2.7 Arquitectura Streaming Sin Cliente. [7]

2.6 Modalidades

Entre las modalidades que se implementan en un sistema transmisión en vivo tenemos la modalidad en directo y bajo demanda.

2.6.1 En Directo

Realiza una transmisión de multidifusión de la información, es decir es enviada a un sin número de destinos de manera simultánea. El flujo de contenido multimedia es procesado en tiempo real una misma información puede estar siendo transmitida a varios usuarios pero estos no podrán realizar tareas de pausa, avance o retroceso sobre la transmisión. Este tipo de

modalidad trabaja con 2 categorías definidas que trabajan según el tipo de señal y transmisión que se realice.

Por su tipo de señal podemos tener Información en vivo cuya información es transmitida justo en el momento que el cliente realice la producción de la misma y también se posee Información almacenada la cual después de ser reproducida la información esta es almacenada en dispositivos para estar disponible en el momento que sea requerida.

Por su tipo de transmisión podemos tener Unicast en donde los paquetes de datos son enviados a un único cliente conectado al sistema y Multicast donde los paquetes de datos son enviados a un grupo de clientes conectados al sistema.

Si lo que se requiere es brindar un servicio de manera eficiente es preferible realizar una difusión del contenido de manera Multicast.

2.6.2 Bajo Demanda

Esta modalidad permite al cliente administrar el contenido este solicita que le sean enviados los datos del archivo que desea reproducir luego la matriz de almacenamiento entrega dicha información a los servidores.

Se trabaja con una transmisión unicast para permitir un envío de datos específico a cada usuario y tenga la posibilidad de tener interacción con la información es decir pausar, avanzar y retroceder la misma. Una desventaja muy notoria de este servicio es que al ser una transmisión de tipo unicast es que se realiza un consumo elevado de ancho de banda puesto que cada cliente tiene una conexión propia con el servidor.

Por ello es recomendable que para que este sistema sea más fiable si la carga se la distribuye en varios servidores pequeños, en vez de un gran servidor de procesador múltiple.

CAPITULO 3

IMPLEMENTACION DE UN SISTEMA DE AUDIO STREAMING

Este capítulo nos describirá una variedad de software y herramientas existentes para poder implementar un sistema de transmisión en vivo de audio, dentro de las cuales se incluyen aquellas aplicaciones que usamos para el presente trabajo de grado.

3.1 Servidor Streaming

Entre los servidores que podemos usar en un sistema de transmisión en vivo de audio contamos con Shoutcast Server e Icecast Server.

3.1.1. Shoutcast Server

Shoutcast es un servidor de transmisión en vivo, propietario multiplataforma que nos permite alojar una emisora de radio y a su vez transmitir audio online con eficacia (difusión por Internet). Es potente y estable, lo que significa que puede lanzar todo su tráfico de radio en Internet sin tener accidentes u otros problemas.

El software permite que el contenido de audio digital, principalmente en formato AAC o MP3, que se emitirá desde y hacia el software reproductor de medios, de esta manera nos permita la creación de Radio de Internet o de DJ para difundir audio transmisión en vivo.

Una de las ventajas de usar Shoutcast es que el formato de salida es compatible con varios clientes, incluyendo Winamp, iTunes, XMMS y VLC Media Player. Si el cliente no admite el protocolo de Shoutcast, a continuación, el servidor Shoutcast envía el stream sin los metadatos lo que le permite ser escuchado en clientes como Windows Media Player. **[14]**

Entre otras ventajas de Shoutcast tenemos:

1. Es ampliamente utilizado en sistemas de audio transmisión en vivo
2. Disfruta de apoyo popular
3. Una gran cantidad de información para solucionar posibles problemas de configuración.

3.1.2. Icecast Server

Icecast es un servidor libre que fue desarrollada por la Fundación Xiph, el mismo que nos permite transmitir archivos de audio mediante la red, gracias al uso de la tecnología de transmisión en vivo.

Icecast Server en su segunda versión tiene la característica de basarse en el formato libre Ogg Vorbis, mientras que en su versión precedente permite utilizar archivos MP3, pero para poderlos usar se requiere de un pago de licencia ya que como sabemos MP3 es un formato propietario.

Entre las ventajas que nos presenta Icecast Server tenemos su flexibilidad, ya que permite al administrador del servidor tener varios orígenes del archivo de audio que va a

reproducirse, sean estas fuentes de reproducción (playlists) previamente establecidas o a su vez reproducir datos de la salida de una tarjeta de audio específica (como el caso de un micrófono). Posee la arquitectura “cliente-servidor”, la misma que permite que se originen contenidos desde varias ubicaciones, que se distribuirán a los clientes a través de un único servidor de Icecast.

Icecast tiene puntos de montaje, lo que significa que transfiere automáticamente oyentes del Auto DJ para la transmisión en vivo sin tener que apagar manualmente cualquier cosa dentro o fuera. Es así que una vez conectado el servidor todos los oyentes se transfieren desde el autodj para la transmisión en vivo y luego de vuelta otra vez cuando haya terminado.

Todos los paquetes de hosting Icecast incluirán libre Auto DJ. La característica de auto dj le permitirá subir archivos mp3 al sistema y poder multiplicar las listas de reproducción seleccionadas y transmitir 24 horas al día, los 7 días de la semana sin tener el ordenador encendido todo el tiempo.**[14]**

3.2 Fuentes

Entre algunas fuentes que permiten realizar una transmisión en vivo de audio tenemos:

“**Muse**” es una aplicación para la mezcla, codificación y audio transmisión en vivo a través de la red. Puede mezclar hasta 6 flujos de bits de audio codificados (a partir de archivos o en la red, sean estos mp3, ogg), más una señal de entrada de una tarjeta de sonido, la transmisión en vivo resultante puede reproducirse localmente en la tarjeta de sonido y / o codificada a diferentes tasas de bits, grabarlos en disco duro y / o ser transmitidos a la red. Cuando se envía a un servidor, el audio resultante se puede escuchar a través de la red por un gran número de reproductores disponibles en diferentes sistemas operativos. Para ser administrada, Muse ofrece interfaces gráficas y una interfaz de línea de comandos documentados al estilo de Unix. [4]

“**DarkIce**” es un software de transmisión de audio en vivo. Graba audio desde una interfaz de audio (por ejemplo, tarjeta de sonido), la codifica y la envía a un servidor de transmisión en vivo.

Darklce puede grabar desde dispositivos de audio OSS, dispositivos de audio ALSA, interfaz de audio de Solaris, fuentes de Jack, uLaw entrada de audio a través de una interfaz en serie, CoreAudio (sucursal darklce-macosx), fuentes de PulseAudio.

Darklce puede enviar los archivos codificados de los siguientes servidores de transmisión en vivo: Shoutcast, Icecast 1.3.xy 2.x, Darwin Transmisión en vivo Server. Darklce se puede ejecutar en los siguientes sistemas operativos tales como FreeBSD, Linux, MacOS X, NetBSD/OpenBSD, Sun Solaris, GNU/Hurd.

Darklce actualmente está siendo probado con las siguientes librerías de audio: GCC 4.4.2, lame 3.98.2, libvorbis 1.2.3, libogg 1.1.4, FAAC 1.28, Twolame 0.3.12, libaacplus versión 2.0.0 o mayor, Jack 0.116.2, libopus 0.9.14. **[20]**

“IceGenerator” es un generador de transmisión en vivo dirigido para Icecast/Shoutcast. Este software se ejecuta como un daemon, el cual posee tres procesos “hijos” que manejan todas las funciones del software. Un cuarto proceso “hijo” (opcional) se preocupa por la comunicación telnet.

La idea principal de este programa es la automatización de radio en el servidor Icecast/Shoutcast. Si tiene muchos procesos en el servidor y el ahorro de CPU es nuestra prioridad, podemos

utilizar IceGenerator. Obviamente archivos mp3/ogg serán remuestreados para permitir transmisión en vivo dependiendo de la capacidad de ancho de banda.

Entre las características de IceGenerator tenemos: Permite colocar automáticamente en directorios la lista de reproducción completa de mp3, o en una tabla MySQL, no permite la repetición de muestreo, por lo que mejora el uso de recursos del CPU, los archivos MP3s/OGGs se reproducen aleatoriamente o en orden alfabético, sin repetir dos veces una canción, podemos recorrer toda la lista de reproducción hasta el final, así como la particularidad que cuando terminamos de utilizar IceGenerator, la última canción se convierte en la primera en ser reproducida la próxima vez que se ejecute. [21]

Entre los requerimientos de IceGenerator para poder ser utilizado tenemos Linux 2.x/FreeBSD/NetBSD/MacOS/Solaris/OpenBSD, instalar previamente libshout2, instalar id3lib para soportar ID3, tener configurado como servidor transmisión en vivo Icecast o Shoutcast y además tener una buena conexión de banda ancha.

“**ices2**” es un software fuente para un servidor en un sistema de transmisión en vivo de audio. El propósito de este software fuente es proporcionar una transmisión de audio a un servidor de

audio transmisión en vivo de tal manera que uno o más oyentes pueden acceder a la transmisión. Con esta configuración, este software fuente nos permite situarnos de forma remota desde el servidor Icecast.

Ices2 lee los archivos de audio desde un origen y los envía a uno o varios archivos o servidores Icecast. Antes de que los archivos de audio sean enviados, Ices2 realiza un procedimiento de remuestreo para generar una transmisión en vivo adecuada dependiendo de los requisitos de ancho de banda.

Un tipo de entrada de audio adicional que posee Ices2 es Roar/RoarAudio, el mismo que es un moderno sistema de sonido para múltiples sistemas operativos. Ices2 no es una aplicación gráfica, su propósito es alimentar la transmisión de un servidor de transmisión en vivo Icecast. Sin embargo necesita tener los siguientes complementos: libogg, libvorbis, libxml2, libshout 2. Actualmente Ices2 soporta las siguientes plataformas Unix: Linux (la mayoría de las distribuciones incluidas Debian y RedHat), FreeBSD, OpenBSD, Solaris. **[23]**

3.3 Clientes

Entre algunos de los clientes que permiten captar una transmisión en vivo de audio tenemos:

“VLC Media Player” es un software reproductor multimedia libre y de código abierto desarrollado por el proyecto VideoLAN. Tiene la característica de ser un software multiplataforma con versiones disponibles para numerosos sistemas operativos como iOS, Microsoft Windows, BSD, BeOS, GNU/Linux, Android y Mac OS entre otros, además de reproducir innumerables formatos de video sin tener la necesidad de instalar códec externos.

VLC es un reproductor de audio y video que puede ser usado como software cliente en un sistema de transmisión de audio en vivo para que el usuario pueda captar los datos transmitidos desde un servidor, y así poder escuchar la transmisión de audio en vivo. VLC Media Player es capaz de leer una gran variedad de archivos dependiendo del sistema operativo. VLC Media Player puede leer varios formatos dependiendo del sistema operativo en el cual este siendo ejecutado.

Entre algunos de los protocolos de red que puede leer VLC Media Player tenemos HTTP, FTP, UDP, RTP (unicast o multicast), MMS, RTSP y RTMP.

Entre algunos formatos de audio de entrada que pueden ser leídos por VLC Media Player son MP3, AAC, AC3, AMR, RealAudio, Ogg Vorbis y WMA.

VLC Media Player tiene la ventaja de poder recodificar numerosos formatos, esto dependiendo del sistema operativo que estemos usando. Entre los protocolos de transmisión en vivo que soporta VLC Media Player tenemos UDP, HTTP, RTP, RTSP y MMS. [22]

“Zinf” es un reproductor de audio para Linux y Win32 simple pero potente que puede ser usado como software cliente en un sistema de transmisión en vivo de audio para que el usuario pueda captar los datos transmitidos desde un servidor, y así poder escuchar la transmisión de audio transmisión en vivo. Zinf se puede ejecutar en Windows 95/98/Me/NT/2000/XP

Zinf tiene la característica de poder reproducir MP3 con el decodificador de audio Xing, reproduce formatos Ogg/Vorbis, WAV, reproducción de audio CD, reproducción de transmisión en vivo HTTP con servidores Icecast y Shoutcast, transmisión en vivo RTP unicast y multicast.

Además permite la transmisión de títulos (estilos de Icecast y Shoutcast), soporta listas de reproducción PLS/M3U, tiene integrado buscador de música y lista de reproducción, Interfaz de usuario

temático, búsqueda de metadatos a través de MusicBrainz, gestor de descargas. [25]

“**Mixxx**” es un software de libre distribución para disc jockeys que permite reproducir archivos de audio y realizar mezclas de los mismos. Posee una interfaz sencilla e intuitiva que permite que este programa pueda ser usado fácilmente por novatos como por usuarios avanzados. Está disponible para Mac OSX, Linux y Windows.

Mixxx es actualmente el único software gratuito que soporta el control de vinilo multiplataforma, además cuenta con los más avanzados controladores MIDI/HID a través de su innovador motor de scripting basado en JavaScript.

Entre las principales características de Mixxx tenemos la visualización de doble audio, la misma que nos permite visualizar el desplazamiento en tiempo real de la forma de la onda de el o los archivos que están siendo reproducidos, a la vez que nos muestra ritmos y puntos de referencia de la pista, junto con una visión general de forma de onda de todo el archivo de audio para la búsqueda rápida de determinado punto de la pista. Posee además controles avanzados como cambio de reproducción con estiramiento de tiempo, y los segmentos del ritmo del bucle. La integración de iTunes en Mixxx le permite al usuario usar sus listas de reproducción y canciones desde iTunes, Traktor y

Rhythmbox. La detección y sincronización de BPM en Mixxx nos permite detectar y sincronizar de forma instantánea el tiempo de sus canciones.

Posee la funcionalidad de transmisión en vivo lo que nos permite iniciar una estación de radio y transmitir tus mezclas en directo a través de Internet usando la tecnología de audio transmisión en vivo. Posee un motor de mezcla avanzado que soporta varios formatos de archivo, estantes de ecualización personalizada, grabación, etc. Cuenta con la función de mezcla automática que le permite al usuario crear una lista de reproducción rápida y dejar que el software se haga cargo mediante la funcionalidad de auto DJ. [24]

“IDJC” Internet DJ Console es un software libre que podemos encontrar en los repositorios de la mayoría de las distribuciones de GNU/Linux existentes. Es una herramienta poderosa, flexible y dinámica que puede ser usada como fuente en un sistema transmisión en vivo de audio, ya que nos permite crear programación rad' Internet.

Así mismo, IDJC puede ser usado como cliente en un sistema de audio transmisión en vivo, ya que se comporta como un cliente gráfico para Icecast y Shoutcast, permite configurar micrófonos, establecer los jingles y archivos de audio que serán reproducidos. [26]

CAPITULO 4

Realización de Pruebas y Obtención de Resultados

En este capítulo se mostraran los análisis estadísticos realizados a través de muestras tomadas durante el funcionamiento de cada uno de los sistemas audio transmisión en vivo, para ello se escogió tres variables las cuales son transferencia, ancho de banda y variación de retardo. Estos datos fueron tomados utilizando la herramienta de software Jperf con la cual podremos simular una conexión cliente-servidor y visualizar los valores de estas

variables durante la transmisión y con esto llevar a cabo nuestros cálculos. El número de muestras por variable que se tomó para el análisis fue obtenido a través de la siguiente fórmula:

$$n = \frac{Z^2 p \cdot q}{e^2} \quad [1,4]$$

n = Tamaño de la muestra

Z = Valor correspondiente al nivel de confianza

pq = Varianza de la población

e = Error muestral

De esta ecuación tomamos como valor $Z = 1.96$, $pq = 0.25$ y un error $e = 5\%$.

Se obtuvo un valor $n=386$

4.1 Pruebas realizadas en Sistema 1 (Servidor Icecast 2 – Fuente Mixxx)

Los resultados de las medias de las variables son vistos en la Tabla I.

Transferencia

En la Fig. 4.1 se muestran los datos estadísticos de esta variable.

Estadísticas descriptivas: transferencia(KB)						
Variable	Media	Varianza	CoefVar	Mínimo	Máximo	Moda
transferencia (KB)	317,94	29,56	1,71	286,00	378,00	317

Figura 4.1 Sistema 1: Datos estadísticos de transferencia

Se muestra claramente algunos datos estadísticos de las muestras tomadas de la variable transferencia, un valor importante a observar es la media que nos permite establecer el tamaño de datos recibidos por un usuario y verificar la calidad de transmisión que en este caso el valor medio fue de 317,94 KB esto lo apreciamos con mayor facilidad en la siguiente Figura 4.2

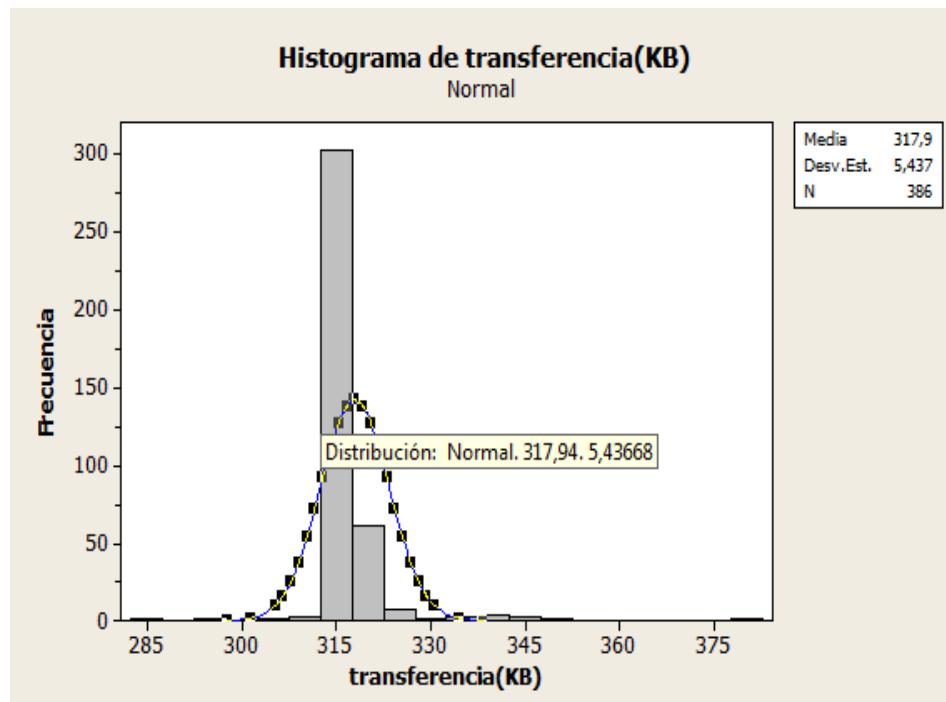


Figura 4.2 Sistema 1: Curva de la Normal de transferencia

Ancho de Banda

En la Fig. 4.3 se muestran los datos estadísticos de esta variable.

Estadísticas descriptivas: ancho de banda(Kb/s)						
Variable	Media	Varianza	CoefVar	Mínimo	Máximo	Moda
ancho de banda(Kb/s)	2618,7	14906,2	4,66	2340,0	3611,0	2599

Figura 4.3 Sistema 1: Datos estadísticos variable ancho de banda

Se obtuvieron algunos datos estadísticos sobre la variable ancho de banda tales como media, varianza, su máximo y su mínimo valor, pero en nuestro trabajo se analizará sobre todo el valor de la media que se obtuvo a través de las muestras sobre las pruebas realizadas en este sistema.

Se identifican algunos datos estadísticos de las muestras tomadas de la variable ancho de banda ya que es nuestro factor limitante al realizar una transmisión de audio transmisión en vivo, dependiendo de su valor el cliente podrá tener una mejor o peor recepción de la información. Para este caso el valor medio fue de 2618,7 Kb/s. Como se muestra en la siguiente Figura 4.4

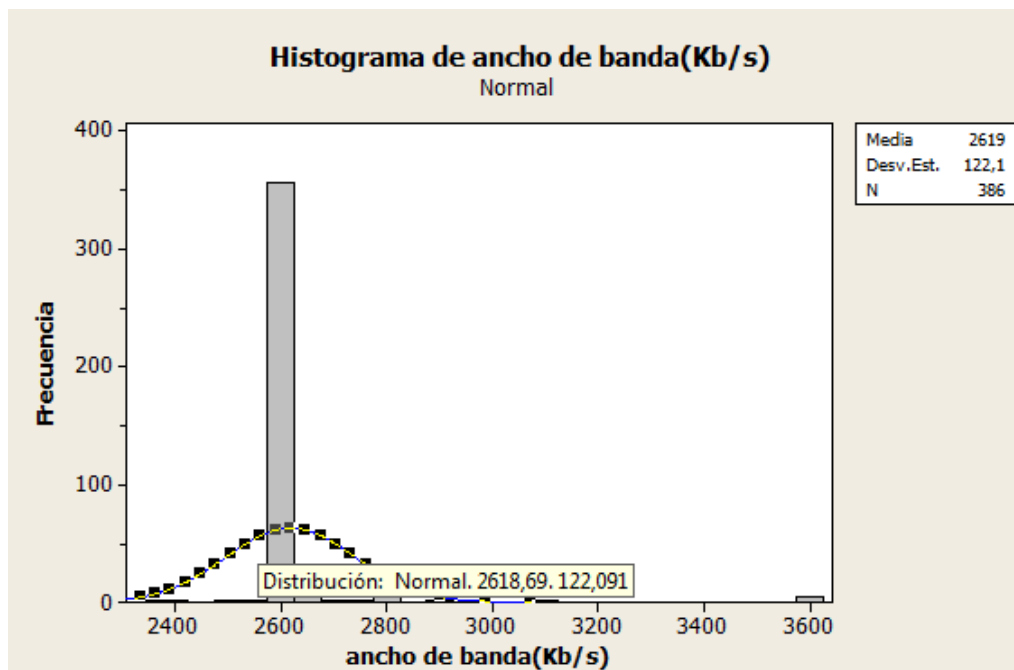


Figura 4.4 Sistema 1: Curva de la Normal variable ancho de banda

Variación de Retardo

En la Fig. 4.5 se muestran los datos estadísticos de esta variable.

Estadísticas descriptivas: variación del retardo (ms)						
Variable	Media	Varianza	CoefVar	Mínimo	Máximo	Moda
variación del retardo (ms)	0,7422	1,1895	146,95	0,0900	9,8500	0,2

Figura 4.5 Sistema 1: Datos estadísticos de variación de retardo

Se recopilan algunos datos estadísticos de las muestras tomadas de la variable variación de retardo que analiza primordialmente el retardo en el tiempo que demora en llegar un paquete cuando una red se encuentra

congestionada por diversos motivos, esto afecta notablemente en el sistema audio transmisión en vivo puesto que los paquetes llegaran a diferentes velocidades y consigo provocara que la trasmisión sea entrecortada, por ello si se desea tener una transmisión con calidad este valor debe ser lo más bajo posible. En nuestro caso el valor medio de la variación de retardo fue de 0,7422 ms como se describe mejor en la siguiente Figura 4.6.

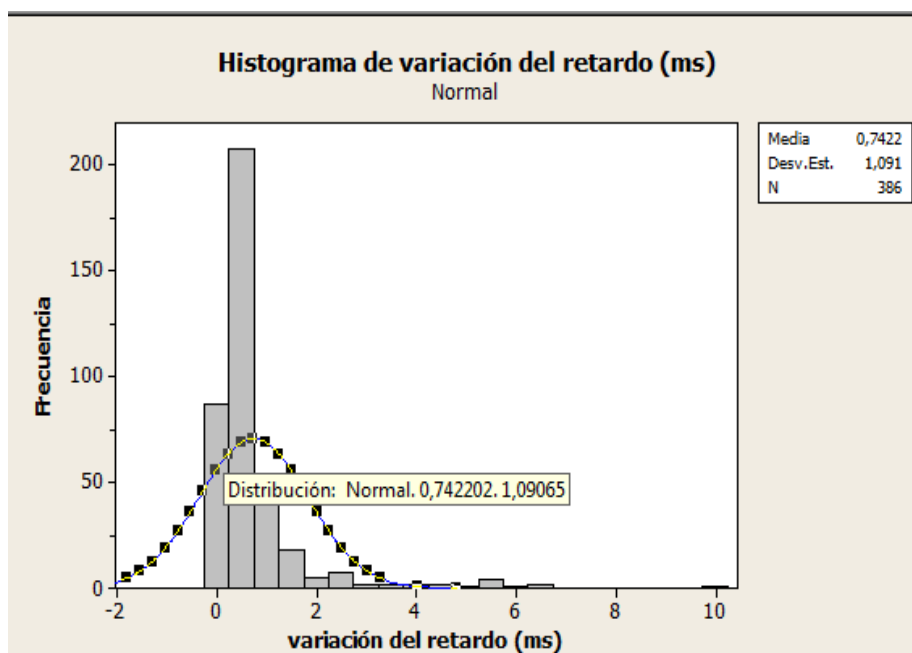


Figura 4.6 Sistema 1: Curva de la Normal de variación de retardo.

Tabla I Media obtenida de las 3 variables con muestras del Sistema 1

Variables	Media
Transferencia	317,94 KB
Ancho de Banda	2618,7 Kb/s
Variación de retardo	0,7422 ms

4.2 Pruebas realizadas en Sistema 2 (Servidor Icecast 2 – Fuente IDJC)

Los resultados de las medias de las variables son vistos en la Tabla II.

Transferencia

En la Fig. 4.7 se muestran los datos estadísticos de esta variable.

Estadísticas descriptivas: transferencia(KB)							
Variable	Media	Varianza	CoefVar	Mínimo	Mediana	Máximo	Moda
transferencia(KB)	317,18	13,24	1,15	287,00	317,00	347,00	317

Figura 4.7 Sistema 2: Datos estadísticos de transferencia

El valor medio obtenido de la variable transferencia con muestras del Sistema 2 es de 317,18 KB y se lo visualiza mejor en la Figura 4.8.

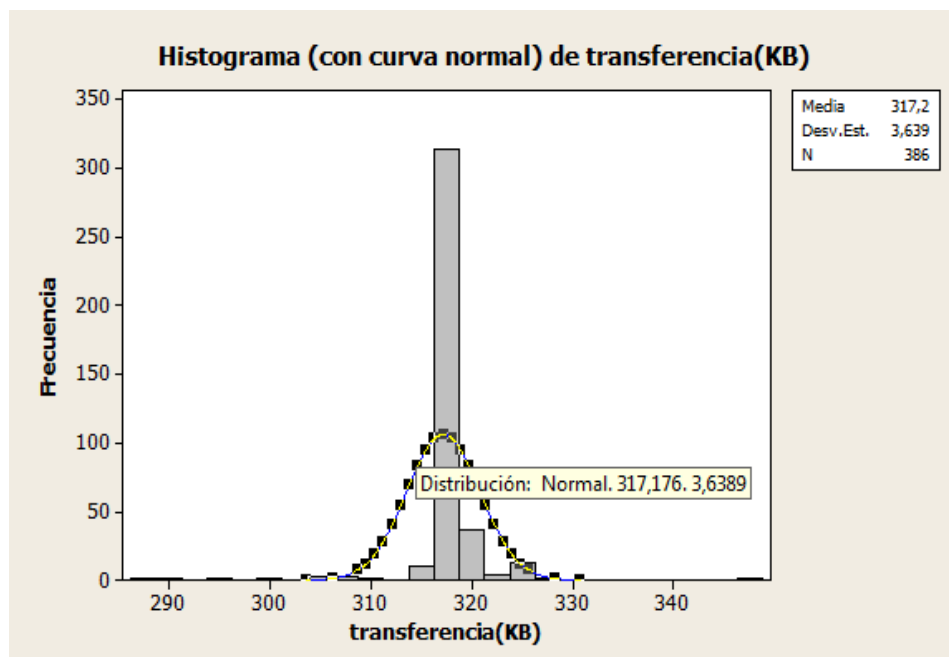


Figura 4.8 Sistema 2: Curva de la Normal de transferencia

Ancho de Banda

Los valores estadísticos de esta variable se aprecian en la Figura 4.9

Estadísticas descriptivas: ancho de banda(Kb/s)

Variable	Media	Varianza	CoefVar	Mínimo	Mediana	Máximo	Moda
ancho de banda(Kb/s)	2599,8	889,0	1,15	2352,0	2599,0	2846,0	2599

Figura 4.9 Sistema 2: Datos estadísticos de ancho de banda

Se observa en la Figura 4.9. El valor medio obtenido de la variable ancho de banda con muestras del Sistema 2 es de 2599,8 Kb/s y se lo visualiza en la Figura 4.10

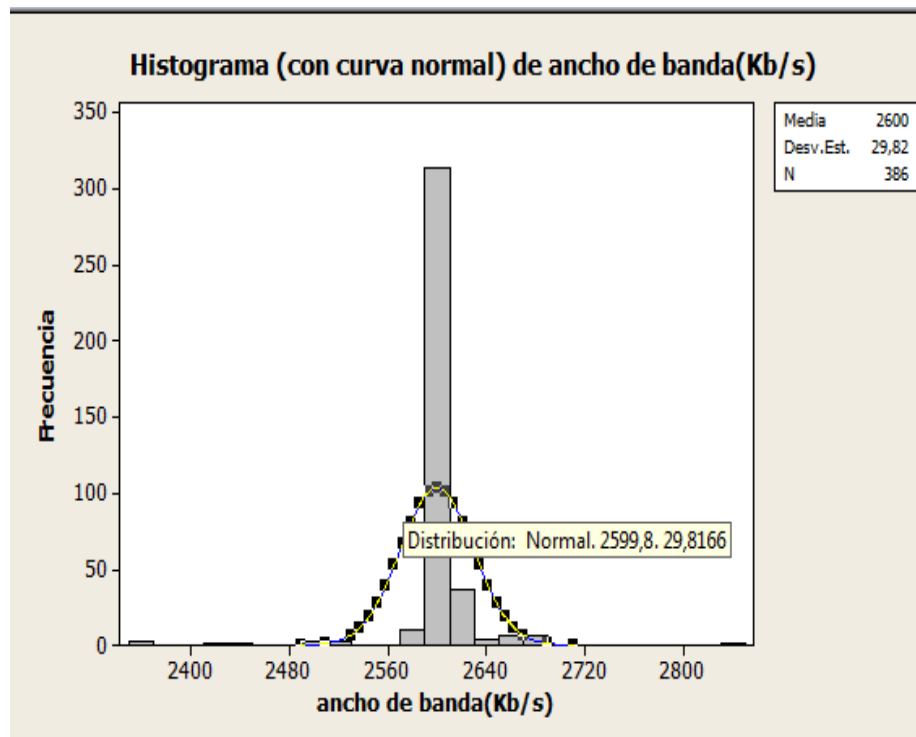


Figura 4.10 Sistema 2:

Curva de la Normal de la variable ancho de banda

Variación de Retardo

Los valores estadísticos de esta variable se aprecian en la Figura 4.11

Estadísticas descriptivas: Variación de retardo (ms)							
Variable	Media	Varianza	CoefVar	Mínimo	Mediana	Máximo	Moda
Variación de retardo (ms)	0,4670	0,5014	151,61	0,0600	0,2700	6,5000	0,24

Figura 4.11 Sistema 2: Datos estadísticos de variación de retardo

El valor medio obtenido de la variable variación de retardo con muestras del Sistema 2 es de 0.467 ms y se lo visualiza en la Figura 4.12.

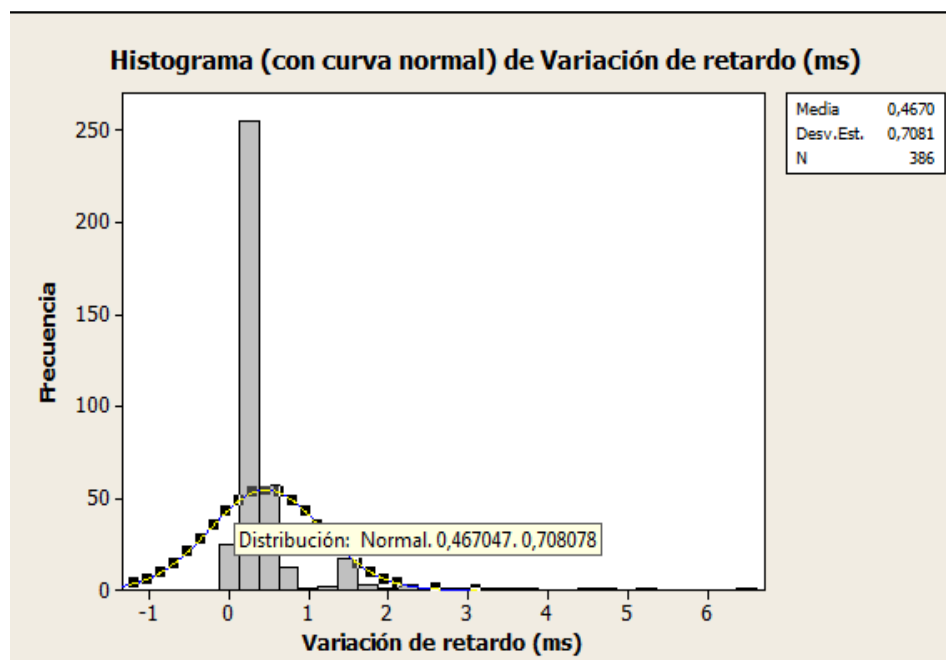


Figura 4.12 Sistema 2: Curva de la Normal de variación de retardo

Tabla II Media obtenida de las 3 variables con muestras del Sistema 2

Variables	Media
Transferencia	317,18 KB
Ancho de Banda	2599,8 Kb/s
Variación de retardo	0,467 ms

CONCLUSIONES Y RECOMENDACIONES

Luego de todo el trabajo realizado en el presente estudio comparativo de dos sistemas de transmisión de audio en vivo, podemos concluir lo siguiente:

1. Durante la implementación de ambos sistemas se determinó que el Sistema 2 presento una mayor dificultad durante la instalación, debido a que requirió una mayor investigación para solucionar los problemas de compatibilidad entre la librería de audio JACK y la versión de software IDJC.
2. Al finalizar las pruebas y analizar los resultados obtenidos con el software Jperf sobre los muestreos de cada sistema, el Sistema 1 presento un valor medio de transferencia mayor en un 0.24% al valor medio de transferencia obtenido en el Sistema 2, valores que pueden ser visualizados en la Tabla I y Tabla II. Esto nos permite concluir que en el Sistema 1 existirá una menor congestión en la red, para un mismo número de usuarios en ambos sistemas, pero esto no representa un factor determinante.
3. Al finalizar las pruebas y analizar los resultados obtenidos con el software Jperf sobre los muestreos de cada sistema, el Sistema 1 presenta un valor medio de ancho de banda mayor en un 0.73% al

valor medio de ancho de banda obtenido en el Sistema 2, valores que pueden ser visualizados en la Tabla I y Tabla II. Esto nos permite concluir que en el Sistema 2 existirá un menor consumo de ancho de banda en el servidor, lo que se reflejara en una mejor conectividad con el cliente, pero esto no representa un factor determinante.

4. Al finalizar las pruebas y analizar los resultados obtenidos con el software Jperf sobre los muestreos de cada sistema, el Sistema 1 presento un valor medio de variación de retardo mayor en un 58.93% al valor medio de variación de retardo obtenido en el Sistema 2, valores que pueden ser visualizados en la Tabla I y Tabla II .Esto nos permite concluir que en el Sistema 2 existirá mejor calidad en la transmisión, por lo tanto será menos probable que se interrumpa la transmisión.

Después de la realización de este trabajo se puede recomendar lo siguiente:

1. Al analizar la calidad de la transmisión de audio en vivo, se recomienda comprender que no existe un valor idóneo para el ancho de banda del usuario (cliente) que sea el adecuado para todos los servidores, a pesar de tener equipos con recursos similares. El resultado que tiene el ancho de banda en la calidad de la transmisión depende de los recursos disponibles momentáneamente cuando se

procesan las peticiones de conexión. El número de peticiones que podría procesar un servidor sin afectar la calidad de la transmisión de audio en vivo varía según el número de usuarios que estén recibiendo contenido actualmente desde dicho servidor.

2. Verificar la arquitectura del sistema operativo en el que se instalaran las aplicaciones IDJC y Mixxx, para no tener problemas de incompatibilidad en versiones de 32 y 64 bits.

BIBLIOGRAFIA

[1] Liliana Patricia Londoño Pinilla (2013, Marzo 23), Software II [online], Disponible en: tallersemestrei.blogspot.com

[2] Anónimo (2009, Marzo 14), Es el streaming la alternativa definitiva para la música [online], Disponible en: <http://notedetengas.es/es-el-streaming-la-alternativa-definitiva-para-la-musica/>

[3] Robalino Romero Geoconda Vanessa, Rodríguez Cabay Adriana Gabriela, Estudio Comparativo de Aplicaciones Software para Video Streaming y Pasarelas en Servicios de Videoconferencia para la ESPOCH, Escuela Superior Politécnica de Chimborazo, Riobamba -Ecuador, 2011, pag 46, 53.

[4] Anónimo (2013, Marzo 15), Linux-Muse [online], Disponible en: <http://linux.downloadatoz.com/muse-streamer>.

[5] Anónimo (2013, Mayo 9), Integración de los Servidores de Media en las Redes Empresariales [online], Disponible en: http://www.betsime.disaic.cu/secciones/tec_ef_06.htm.

[6] Guevara Jiménez Juan Xavier, LlumiQuinga LlumiQuinga Braulio Alonso, Análisis de Factibilidad para la Implementación del Servicio Streaming en el

ámbito del E-Learning en la Escuela Politécnica Nacional, Agosto 2011, pag 42.

[7] Anónimo (2013, Mayo 28), Streaming [online], Disponible en: <http://informatica.iessanclemente.net/>

[8] David Mateos Costilla, Samuel Reaño Montoro, Streaming de Audio/Video: Protocolo RTSP, 2007-2008

[9] Marcela Barria, Eduardo Hales, Tecnología Streaming, Universidad Técnica Federico Santa María Departamento De Electrónica, 26 de Agosto de 2013.

[10] Francisco José Suarez Alonso, Tecnologías de Streaming, Universidad de Oviedo Área de Arquitectura y Tecnología de Computadores, 2010-2011.

[11] Martínez Sánchez Emmanuel, Pérez Rojas Carlos Eduardo, Aplicación Triple Play en internet por Wimax-Estacion de Radio, Escuela Superior de Ingeniería Mecánica y Eléctrica, México- 2008.

[12] Anónimo (2012, Febrero 13), Transmitir radio en vivo con Icecast2 desde Linux Ubuntu [online], Disponible en: <http://robersoft.blogcindario.com/2012/02/00046-transmitir-radio-en-vivo-con-icecast2-desde-linux-ubuntu.html>.

[13] Granja Requelme Eduardo Reinaldo, Vega Monge Pablo Javier, Diseño e Implementación De Un Sistema De Transmisión de Audio y Video Por Internet Para La Ex-Facultad De Ingeniería Eléctrica, Quito, Agosto 2004.

[14] Anónimo,[online], Disponible en <http://www.wikipedia.org/>,

[15] González Escobar Elisa, Protocolo HTTP, Universidad Tecnológica de Tulancingo , Julio 2013.

[16] Cesar Henrique Pereira Ribeiro, Novas Propostas para Protocolos de Streaming, Escola de Engenharia – Universidade Federal Fluminense , Brazil, fecha de consulta junio 2013.

[17] Juan Pablo Quintero Ortiz, Cristian Andrey Castro Serna, Evaluación De Servidores De Streaming De Video Orientado A Dispositivos Móviles, Universidad De Antioquia Facultad De Ingeniería Departamento De Ingeniería Electrónica, Medellín 2006.

[18] Davis Austerberry, The Technology of Video & Audio Streaming Second Edition, 2005

[19] Daniel Rijo Sciara, Fundamentos de Video Streaming, Instituto de Ingeniería Eléctrica Facultad de Ingeniería Universidad de la República, Montevideo – Uruguay, Diciembre de 2004

[20] Federico (2007, Octubre 12), Una radio en Linux [online], Disponible en <http://federicosayd.wordpress.com/2007/10/12/una-radio-en-linux>.

[21] Tony (2003, Julio), IceGenerator [online], Disponible en:
<http://www.becrux.com/index.php?page=projects&name=icegenerator>,.

[22] Anónimo (2003 , Junio 16), VLC playback Features [online], Disponible en : <http://www.videolan.org/vlc/features.html>.

[23] Anónimo (2004, Junio 7), Ices [online], Disponible en:
<http://www.icecast.org/docs/ices-2.0.2/basic.html>,

[24] Anónimo (2013, Mayo 9), Mixxx User Manual [online], Disponible en:
<http://mixxx.org/manual/latest>,

[25] gk, mayhemchaos (2013,Abril 23), Zinf [online], Disponible en :
<http://zinf.sourceforge.net>.

[26] Stephen Fairchild (2007, Julio 22), IDJC [online] Disponible en :
<http://idjc.sourceforge.net>.

ANEXOS

ANEXO A

Diseño del Primer Sistema de Transmisión de Audio Streaming

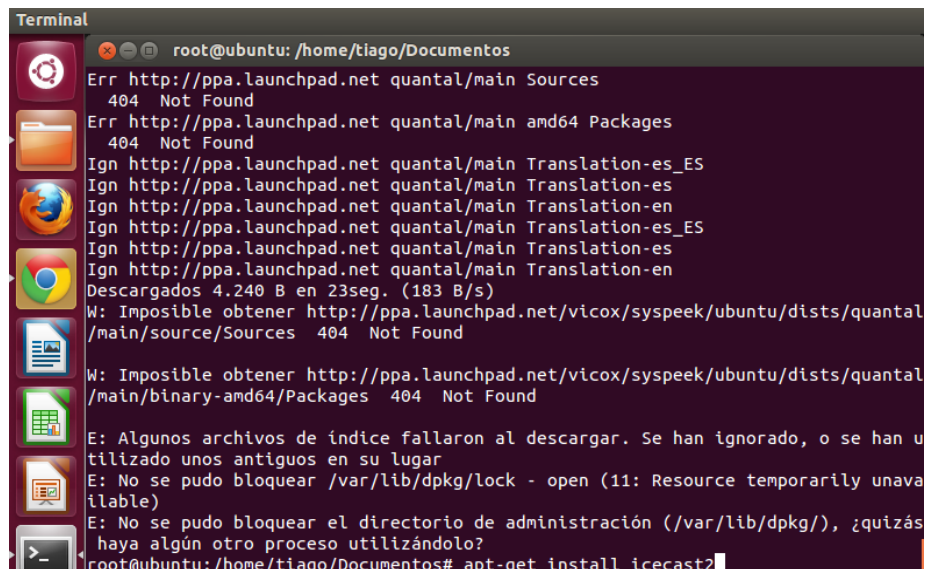
En este capítulo describiremos paso a paso el procedimiento a seguir para el diseño de los sistemas de audio transmisión en vivo utilizados en este trabajo.

Este sistema será diseñado usando como servidor icecast2 y como fuente el cliente mixxx.

PASO 1 INSTALACION DE ICECAST2

Procedemos a instalar icecast2. Para ello abrimos una terminal y escribimos el siguiente comando.

sudo apt-get install icecast2 como se observa en la Figura A.1

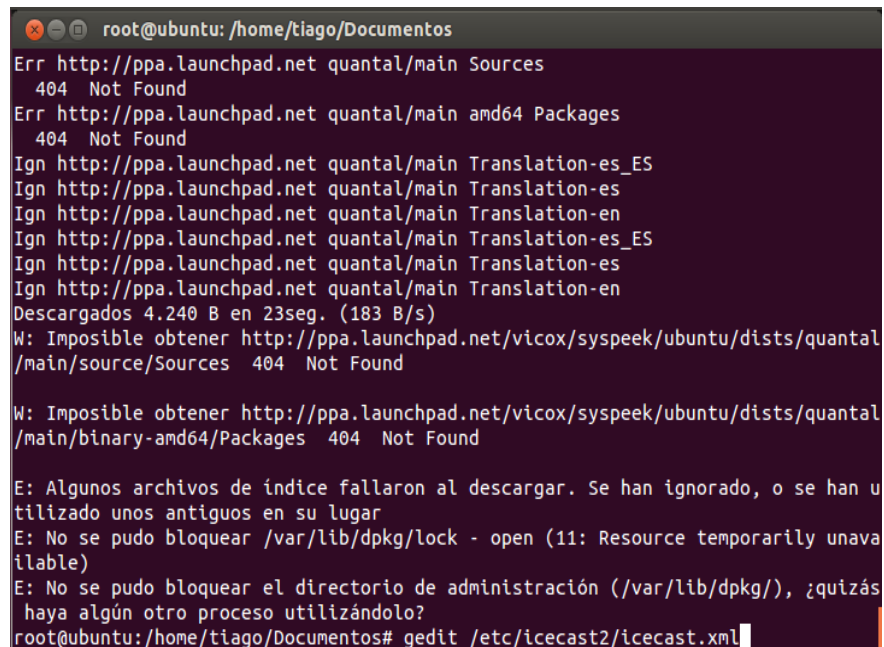


```
Terminal
root@ubuntu: /home/tiago/Documentos
Err http://ppa.launchpad.net quantal/main Sources
404 Not Found
Err http://ppa.launchpad.net quantal/main amd64 Packages
404 Not Found
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Descargados 4.240 B en 23seg. (183 B/s)
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal
/main/source/Sources 404 Not Found
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal
/main/binary-amd64/Packages 404 Not Found
E: Algunos archivos de índice fallaron al descargar. Se han ignorado, o se han u
tilizado unos antiguos en su lugar
E: No se pudo bloquear /var/lib/dpkg/lock - open (11: Resource temporarily unava
ilable)
E: No se pudo bloquear el directorio de administración (/var/lib/dpkg/), ¿quizás
haya algún otro proceso utilizándolo?
root@ubuntu: /home/tiago/Documentos# apt-get install icecast2
```

Figura A.1 Instalación icecast2

PASO 2 CONFIGURACION DE ICECAST2

Icecast2 tiene ubicados sus archivos de configuración en /etc/icecast2. Debemos configurar las contraseñas tanto para las fuentes como para el administrador, además de configurar el nombre del servidor, editando el archivo /etc/icecast2/icecast.xml como se observa en la Figura A.2



```
root@ubuntu: /home/tiago/Documentos
Err http://ppa.launchpad.net quantal/main Sources
  404 Not Found
Err http://ppa.launchpad.net quantal/main amd64 Packages
  404 Not Found
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Descargados 4.240 B en 23seg. (183 B/s)
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal
/main/source/Sources 404 Not Found

W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal
/main/binary-amd64/Packages 404 Not Found

E: Algunos archivos de índice fallaron al descargar. Se han ignorado, o se han u
tilizado unos antiguos en su lugar
E: No se pudo bloquear /var/lib/dpkg/lock - open (11: Resource temporarily unava
ilable)
E: No se pudo bloquear el directorio de administración (/var/lib/dpkg/), ¿quizás
haya algún otro proceso utilizándolo?
root@ubuntu: /home/tiago/Documentos# gedit /etc/icecast2/icecast.xml
```

Figura A.2 Modificación del archivo icecast.xml

Se observan las modificaciones que se realiza al archivo en la Figura A.3 y Figura A.4

```
icecast.xml x
<!-- You may have multiple <listener> elements -->
<hostname>localhost</hostname>
<listen-socket>
  <port>8000</port>
  <!-- <bind-address>127.0.0.1</bind-address> -->
  <!-- <shoutcast-mount>/stream</shoutcast-mount> -->
</listen-socket>
<!--
<listen-socket>
  <port>8001</port>
</listen-socket>
-->

<!--<master-server>127.0.0.1</master-server>-->
<!--<master-server-port>8001</master-server-port>-->
<!--<master-update-interval>120</master-update-interval>-->
<!--<master-password>hackme</master-password>-->

<!-- setting this makes all relays on-demand unless overridden, this is
useful for master relays which do not have <relay> definitions here.
The default is 0 -->
<!--<relays-on-demand>1</relays-on-demand>-->

<!--
<relay>
  <server>127.0.0.1</server>
  <port>8001</port>
  <mount>/example.ogg</mount>
  <local-mount>/different.ogg</local-mount>
  <on-demand>0</on-demand>

  <relay-shoutcast-metadata>0</relay-shoutcast-metadata>
</relay>
-->
```

Figura A.3 Contenido del archivo Icecast.xml

```
icecast.xml x
<icecast>
  <limits>
    <clients>5</clients>
    <sources>2</sources>
    <threadpool>5</threadpool>
    <queue-size>524288</queue-size>
    <client-timeout>30</client-timeout>
    <header-timeout>15</header-timeout>
    <source-timeout>10</source-timeout>
    <!-- If enabled, this will provide a burst of data when a client
first connects, thereby significantly reducing the startup
time for listeners that do substantial buffering. However,
it also significantly increases latency between the source
client and listening client. For low-latency setups, you
might want to disable this. -->
    <burst-on-connect>1</burst-on-connect>
    <!-- same as burst-on-connect, but this allows for being more
specific on how much to burst. Most people won't need to
change from the default 64k. Applies to all mountpoints -->
    <burst-size>65535</burst-size>
  </limits>

  <authentication>
    <!-- Sources log in with username 'source' -->
    <source-password>tiago16</source-password>
    <!-- Relays log in username 'relay' -->
    <relay-password>tiago16</relay-password>

    <!-- Admin logs in with the username given below -->
    <admin-user>admin</admin-user>
    <admin-password>tiago16</admin-password>
  </authentication>

  <!-- set the mountpoint for a shoutcast source to use, the default if not
specified is /stream but you can change it here if an alternative is
```

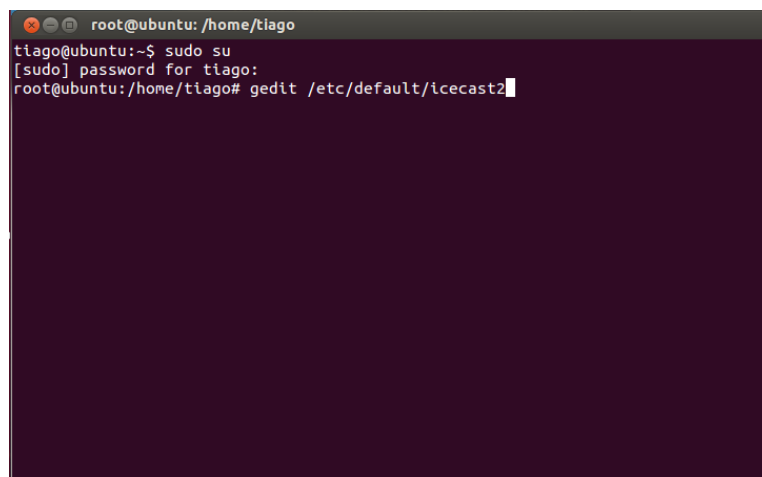
Figura A.4 Archivo Icecast.xml con las modificaciones realizadas

PASO 3 HABILITAR EL SERVICIO

Para poder habilitar el servicio de Iccast2 editamos el archivo `/etc/default/icecast2` poniendo la variable `ENABLE` a `true` que por defecto viene inhabilitada.

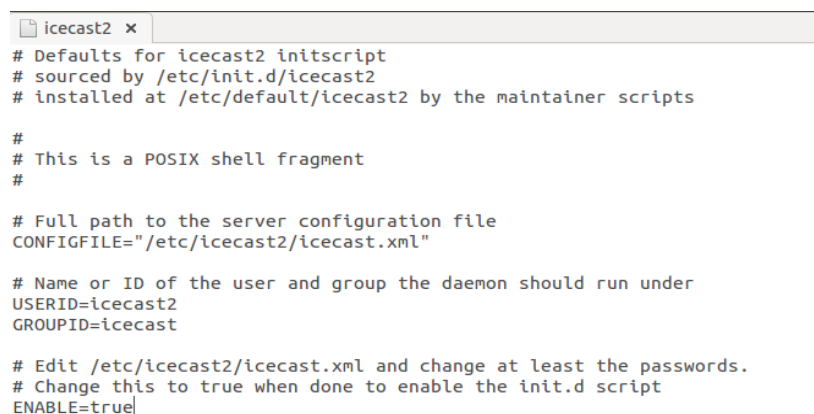
sudo gedit /etc/default/icecast2

Se observa el comando a ejecutar y las modificaciones que se realiza al archivo en la Figura A.5 y Figura A.6



```
root@ubuntu: /home/tiago
tiago@ubuntu:~$ sudo su
[sudo] password for tiago:
root@ubuntu:/home/tiago# gedit /etc/default/icecast2
```

Figura A.5 Modificación del archivo Icecast2



```
icecast2 x
# Defaults for icecast2 initscript
# sourced by /etc/init.d/icecast2
# installed at /etc/default/icecast2 by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Full path to the server configuration file
CONFIGFILE="/etc/icecast2/icecast.xml"

# Name or ID of the user and group the daemon should run under
USERID=icecast2
GROUPID=icecast

# Edit /etc/icecast2/icecast.xml and change at least the passwords.
# Change this to true when done to enable the init.d script
ENABLE=true|
```

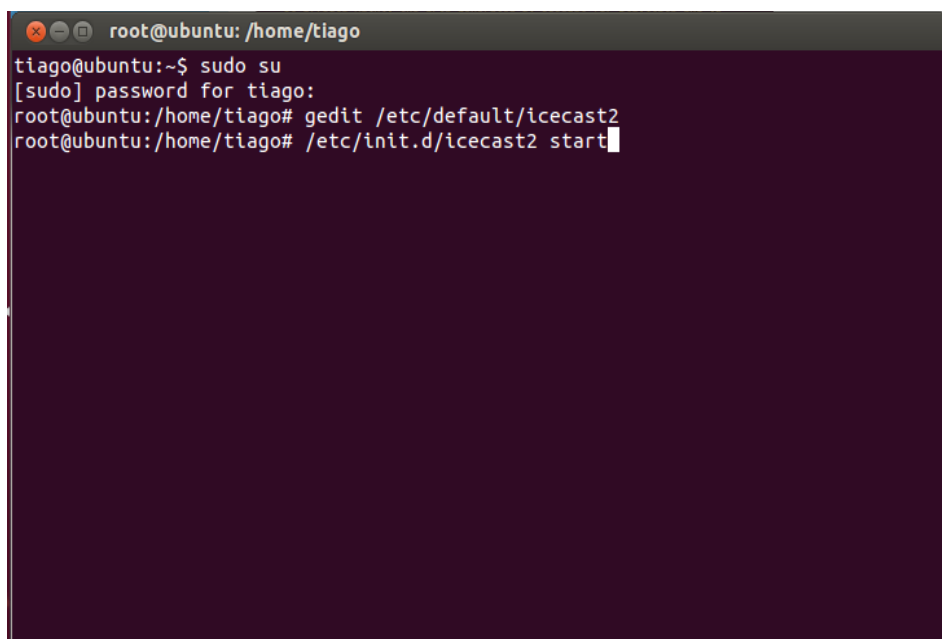
Figura A.6 Permitir la habilitación del servicio Icecast2

PASO 4 ACTIVAR EL SERVICIO DE ICECAST

Icecast2 viene con un servicio montado en `/etc/init.d/icecast2` (el cual acabamos de habilitar).

Lo activamos escribiendo el siguiente comando en una terminal:

sudo /etc/init.d/icecast2 start como se observa en la Figura A.7

A terminal window with a dark purple background. The title bar shows 'root@ubuntu: /home/tiago'. The terminal content is as follows:

```
tiago@ubuntu:~$ sudo su
[sudo] password for tiago:
root@ubuntu:/home/tiago# gedit /etc/default/icecast2
root@ubuntu:/home/tiago# /etc/init.d/icecast2 start
```

Figura A.7 Activación del Icecast2

PASO 5 COMPROBAR EL FUNCIONAMIENTO DE ICECAST2

Para comprobar el funcionamiento de Icecast2 nos vamos a la barra de direcciones del navegador web y escribimos `http://localhost:8000`.

Si todo fue correcto veremos la página de Icecast2, desde la cual

tenemos acceso a administración, listas de reproducción, conexiones clientes. Se visualiza el servicio activo en la Figura A.8

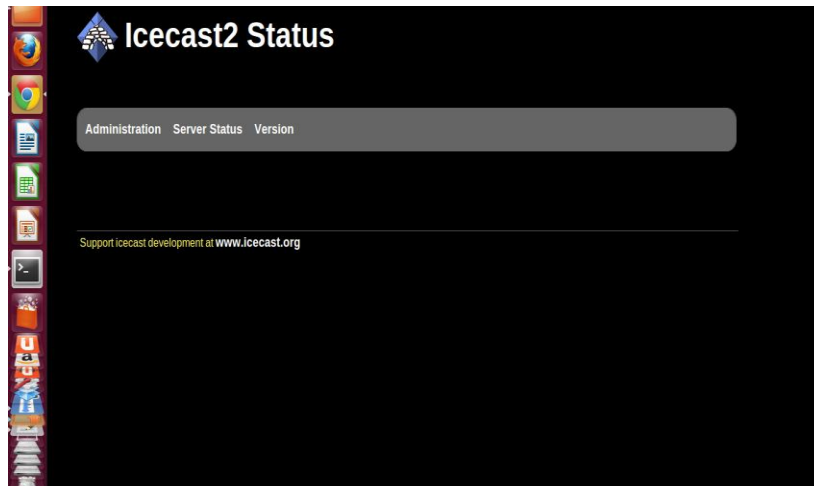


Figura A.8 Servicio Icecast2 activo

Instalación de Mixxx

PASO 1

Instalamos Mixxx escribiendo el siguiente comando desde una terminal.

sudo apt-get install mixxx como se observa en la Figura A.9

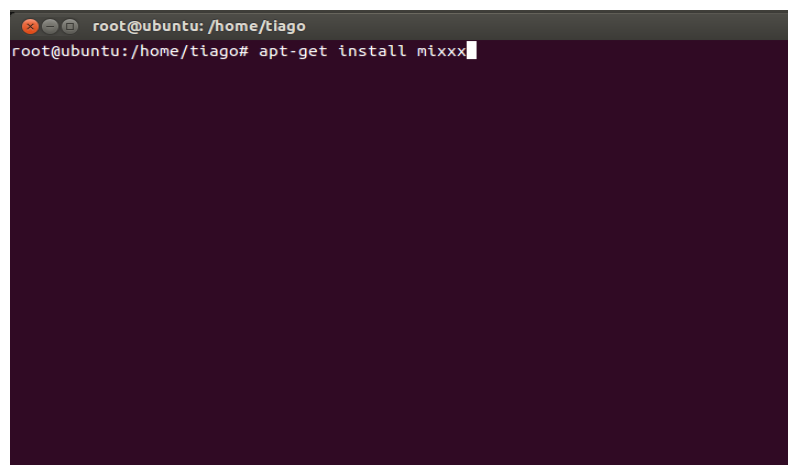
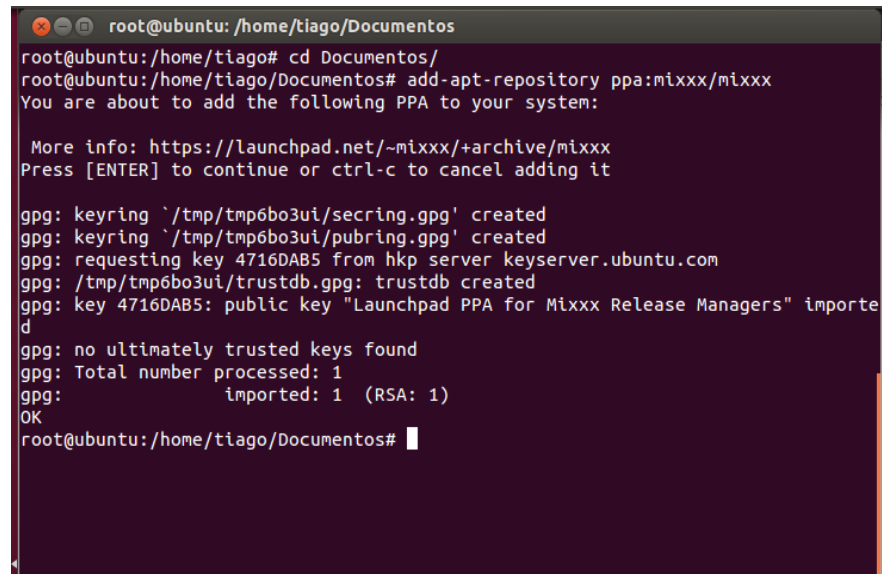


Figura A.9 Instalación de Mixxx

PASO 2

Para que exista el correcto funcionamiento de Mixxx debemos añadir los repositorios.

sudo add-apt-repository ppa:mixxx/mixxx como se observa en la Figura A.10



```
root@ubuntu: /home/tiago/Documentos
root@ubuntu:/home/tiago# cd Documentos/
root@ubuntu:/home/tiago/Documentos# add-apt-repository ppa:mixxx/mixxx
You are about to add the following PPA to your system:

More info: https://launchpad.net/~mixxx/+archive/mixxx
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmp6bo3ui/secring.gpg' created
gpg: keyring `/tmp/tmp6bo3ui/pubring.gpg' created
gpg: requesting key 4716DAB5 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmp6bo3ui/trustdb.gpg: trustdb created
gpg: key 4716DAB5: public key "Launchpad PPA for Mixxx Release Managers" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:         imported: 1 (RSA: 1)
OK
root@ubuntu:/home/tiago/Documentos#
```

Figura A.10 Agregación de los Repositorios

PASO 3

Actualizamos la instalación escribiendo el siguiente comando desde una terminal.

sudo apt-get update como se observa en la Figura A.11

```
Terminal
root@ubuntu: /home/tiago/Documentos
gpg: keyring `/tmp/tmp6bo3ui/secring.gpg' created
gpg: keyring `/tmp/tmp6bo3ui/pubring.gpg' created
gpg: requesting key 4716DAB5 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmp6bo3ui/trustdb.gpg: trustdb created
gpg: key 4716DAB5: public key "Launchpad PPA for Mixxx Release Managers" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:      imported: 1 (RSA: 1)
OK
root@ubuntu:/home/tiago/Documentos# apt-get update
Ign http://dl.google.com stable InRelease
Des:1 http://dl.google.com stable Release.gpg [198 B]
Ign http://ppa.launchpad.net quantal InRelease
Des:2 http://dl.google.com stable Release [1.347 B]
Ign http://archive.ubuntu.com quantal InRelease
Ign http://archive.canonical.com quantal InRelease
Ign http://security.ubuntu.com quantal-security InRelease
Ign http://linux.dropbox.com precise InRelease
Des:3 http://dl.google.com stable/main amd64 Packages [1.205 B]
Ign http://ppa.launchpad.net quantal InRelease
Ign http://archive.ubuntu.com quantal-updates InRelease
Des:4 http://linux.dropbox.com precise Release.gpg [489 B]
Obj http://archive.canonical.com quantal Release.gpg
```

Figura A.11 Actualización de la Instalación de Mixxx

PASO 4

Instalamos los codecs de audio necesarios para Mixxx con el siguiente comando.

sudo apt-get install mixxx libportaudio2 como se observa en la Figura A.12

```
Terminal
root@ubuntu: /home/tiago/Documentos
Err http://ppa.launchpad.net quantal/main Sources
404 Not Found
Err http://ppa.launchpad.net quantal/main amd64 Packages
404 Not Found
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-en
Descargados 4.240 B en 22seg. (189 B/s)
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal/main/source/Sources 404 Not Found
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal/main/binary-amd64/Packages 404 Not Found
E: Algunos archivos de índice fallaron al descargar. Se han ignorado, o se han utilizado unos antiguos en su lugar
E: No se pudo bloquear /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)
E: No se pudo bloquear el directorio de administración (/var/lib/dpkg/), ¿quizás haya algún otro proceso utilizándolo?
root@ubuntu:/home/tiago/Documentos# apt-get install mixxx libportaudio2
```

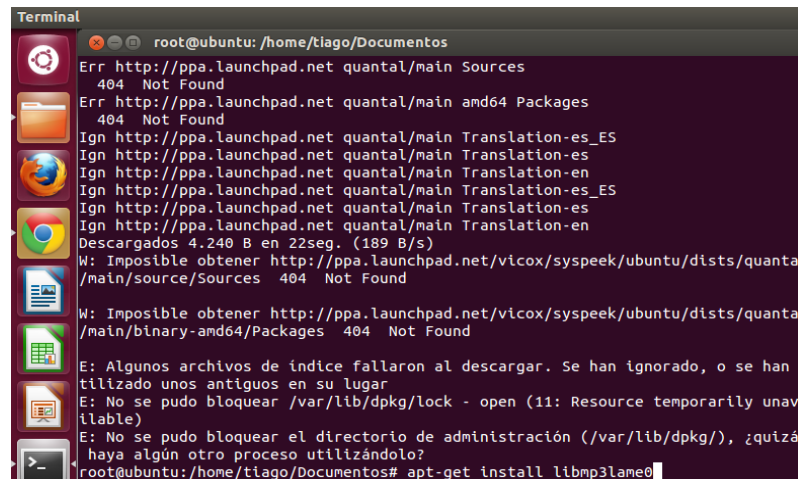
Figura A.12 Instalación de los codecs de audio para Mixxx

PASO 5

En sistemas operativos Ubuntu y GNU / Linux, el audio transmisión en vivo con formato MP3 se puede activar mediante la instalación del paquete libmp3lame. Dependiendo de la distribución de Linux el paquete puede ser ligeramente diferente llamado como lame.

Escribimos el siguiente comando desde una terminal:

sudo apt-get install libmp3lame0 como se observa en la Figura A.13



```
Terminal
root@ubuntu: /home/tiago/Documentos
Err http://ppa.launchpad.net quantal/main Sources
404 Not Found
Err http://ppa.launchpad.net quantal/main amd64 Packages
404 Not Found
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Ign http://ppa.launchpad.net quantal/main Translation-es_ES
Ign http://ppa.launchpad.net quantal/main Translation-es
Ign http://ppa.launchpad.net quantal/main Translation-en
Descargados 4.240 B en 22seg. (189 B/s)
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal/main/source/Sources 404 Not Found
W: Imposible obtener http://ppa.launchpad.net/vicox/syspeek/ubuntu/dists/quantal/main/binary-amd64/Packages 404 Not Found
E: Algunos archivos de índice fallaron al descargar. Se han ignorado, o se han utilizado unos antiguos en su lugar
E: No se pudo bloquear /var/lib/dpkg/lock - open (11: Resource temporarily unavailable)
E: No se pudo bloquear el directorio de administración (/var/lib/dpkg/), ¿quizás haya algún otro proceso utilizándolo?
root@ubuntu: /home/tiago/Documentos# apt-get install libmp3lame0
```

Figura A.13 Instalación de paquetes Mixxx

PASO 6

Una vez instalado el programa se puede encontrar en, Aplicaciones -> sonido y vídeo -> Mixxx o en buscar colocamos Mixxx como se ve en la Figura A.14

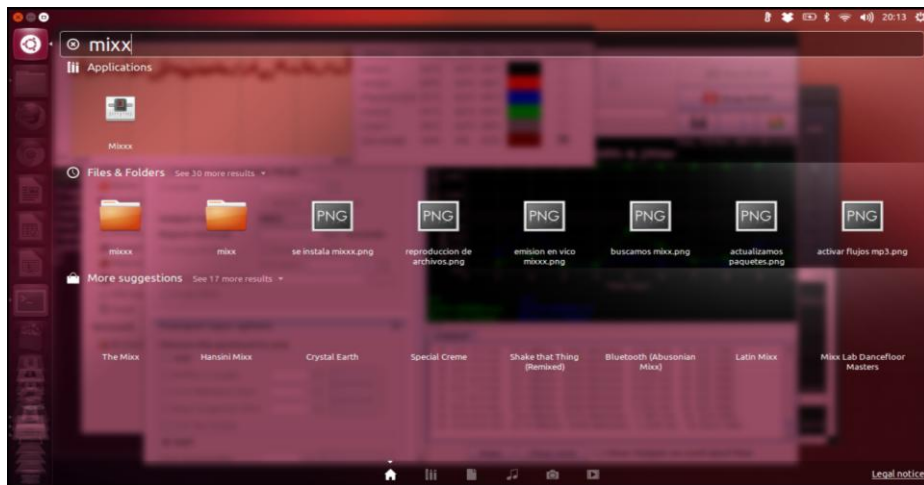


Figura A.14 Comprobación de la instalación de Mixxx

PASO 7

Una vez ejecutado Mixxx nos dirigimos a las opciones del Menú y seleccionamos:

Opciones -> Preferencias y la configuramos tal como se muestra en la Figura A.15 a continuación:

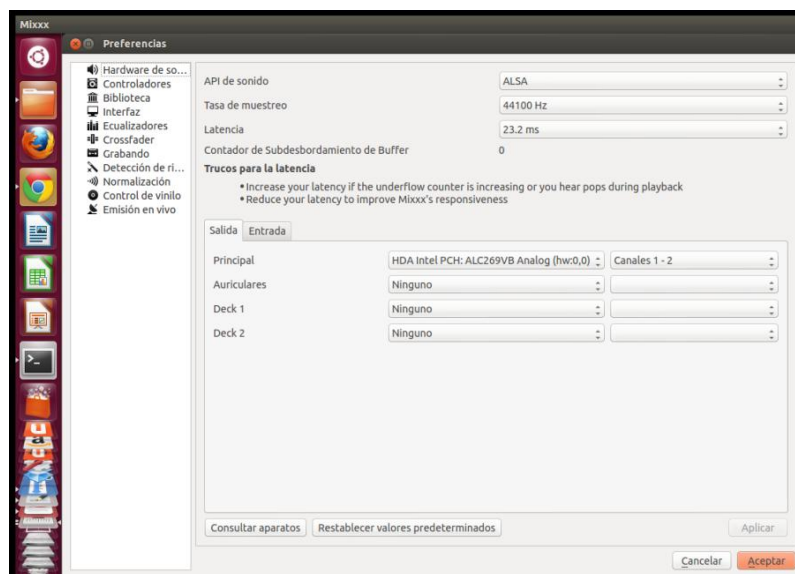


Figura A.15 Configuración de Mixxx

PASO 8

Dentro de Preferencias seleccionamos Emisión en Vivo para llenar los datos del tipo de servidor: southcast o icecast, así como configurar parámetros como el punto de montaje, servidor, formato, encoding, nombre de la estación, metadatos, etc.

Estos datos son los mismos que colocamos al configurar `/etc/icecast2/icecast.xml` como se observa en la Figura A.16

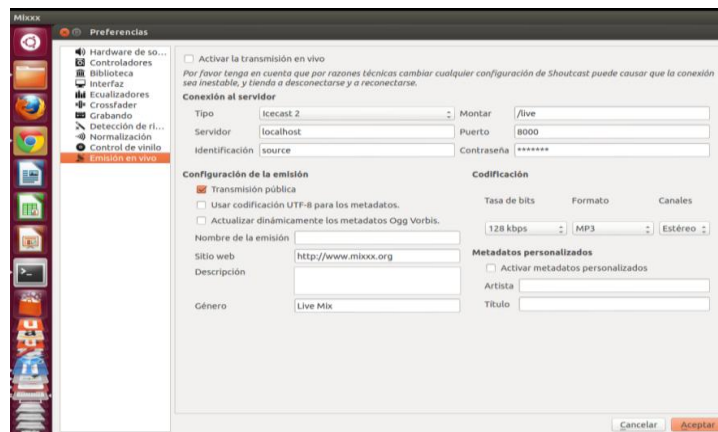


Figura A.16 Ingreso de datos en la configuración de Mixxx

PASO 9

Por último tenemos que habilitar la transmisión en vivo, Esto lo hacemos en Opciones -> Activar la transmisión en vivo, para que se emita lo que se está tocando. Se visualiza en la Figura A.17 y después se observa la interfaz gráfica del programa la Figura A.18

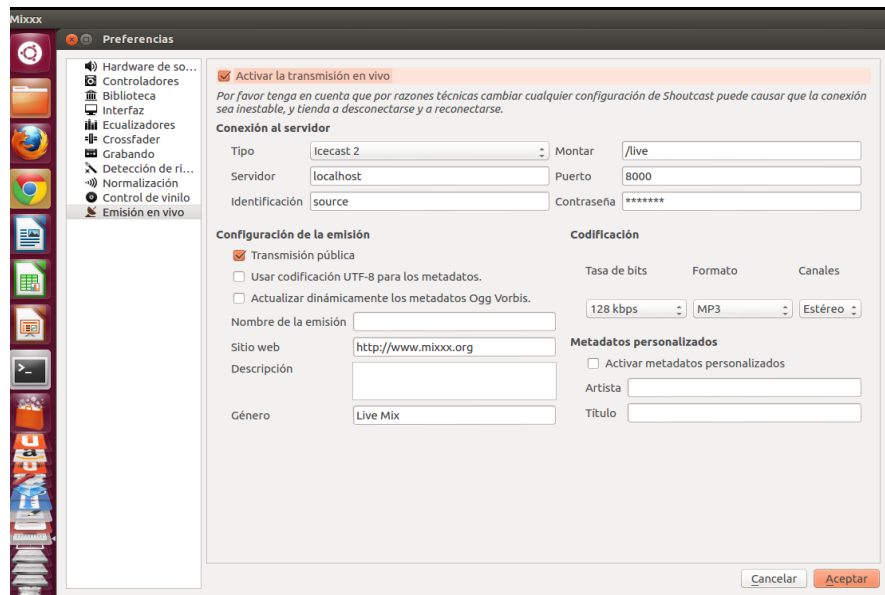


Figura A.17 Activar la emisión en vivo de Mixxx

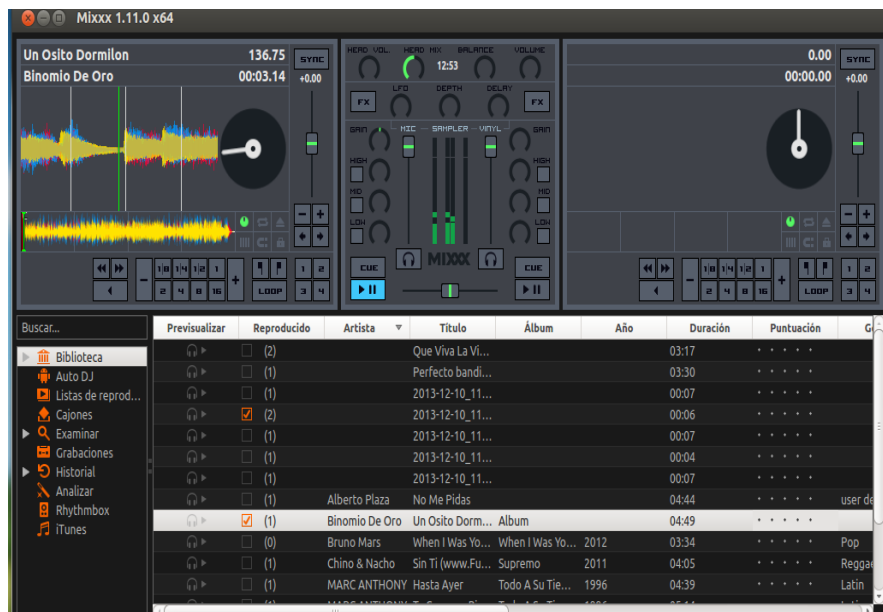


Figura A.18 Vista de la interfaz gráfica del programa Mixxx emitiendo

ANEXO B

Diseño del Segundo Sistema de Transmisión de Audio Streaming

Este sistema será diseñado usando como servidor icecast2 y como fuente el cliente IDJC.

PASO 1

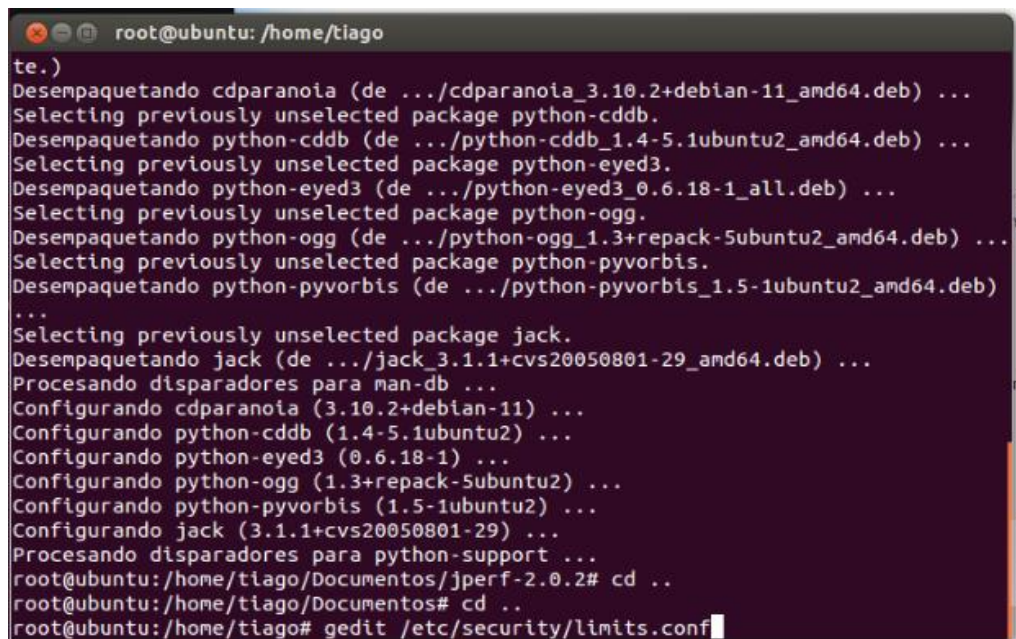
Instalamos Jack desde una terminal con el siguiente comando:

```
sudo apt-get install jackd2
```

Paso 2

Editamos el archivo `/etc/security/limits.conf`

```
sudo gedit /etc/security/limits.conf como se observa en la Figura B.1
```



```
root@ubuntu: /home/tiago
te.)
Desempaquetando cdparanoia (de ../cdparanoia_3.10.2+debian-11_amd64.deb) ...
Selecting previously unselected package python-cddb.
Desempaquetando python-cddb (de ../python-cddb_1.4-5.1ubuntu2_amd64.deb) ...
Selecting previously unselected package python-eyed3.
Desempaquetando python-eyed3 (de ../python-eyed3_0.6.18-1_all.deb) ...
Selecting previously unselected package python-ogg.
Desempaquetando python-ogg (de ../python-ogg_1.3+repack-5ubuntu2_amd64.deb) ...
Selecting previously unselected package python-pyvorbis.
Desempaquetando python-pyvorbis (de ../python-pyvorbis_1.5-1ubuntu2_amd64.deb)
...
Selecting previously unselected package jack.
Desempaquetando jack (de ../jack_3.1.1+cvcs20050801-29_amd64.deb) ...
Procesando disparadores para man-db ...
Configurando cdparanoia (3.10.2+debian-11) ...
Configurando python-cddb (1.4-5.1ubuntu2) ...
Configurando python-eyed3 (0.6.18-1) ...
Configurando python-ogg (1.3+repack-5ubuntu2) ...
Configurando python-pyvorbis (1.5-1ubuntu2) ...
Configurando jack (3.1.1+cvcs20050801-29) ...
Procesando disparadores para python-support ...
root@ubuntu:/home/tiago/Documentos/jperf-2.0.2# cd ..
root@ubuntu:/home/tiago/Documentos# cd ..
root@ubuntu:/home/tiago# gedit /etc/security/limits.conf
```

Figura B.1 Edición del archivo limits.config

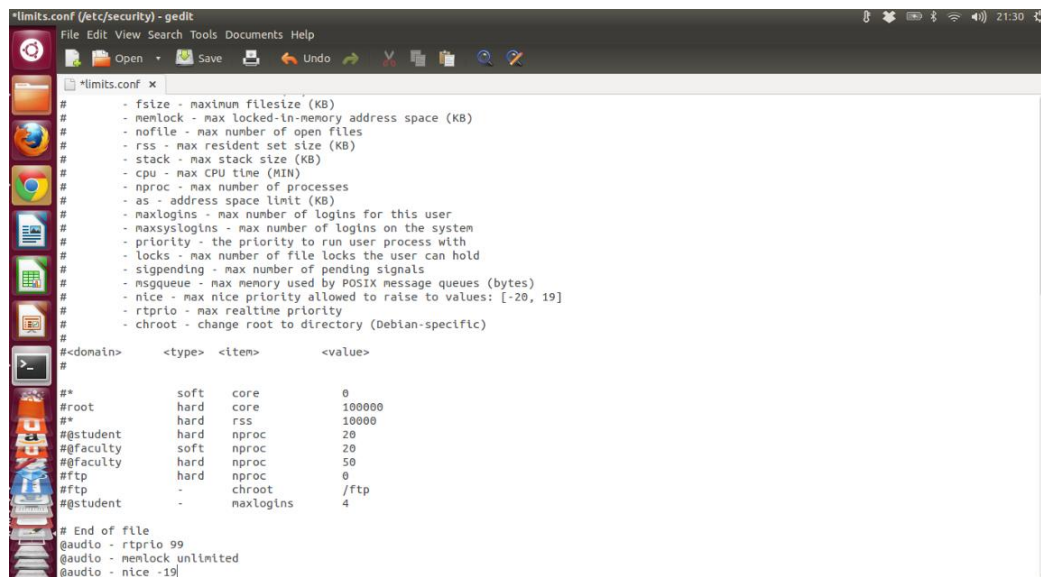
Nos vamos al final del archivo e incluimos las siguientes líneas.

```
@audio - rtprio 99
```

```
@audio - memlock unlimited
```

```
@audio - nice -19
```

Como se observa en la Figura B.2



```
*limits.conf (/etc/security) - gedit
File Edit View Search Tools Documents Help
*limits.conf x
# - fsize - maximum filesize (KB)
# - memlock - max locked-in-memory address space (KB)
# - nofile - max number of open files
# - rss - max resident set size (KB)
# - stack - max stack size (KB)
# - cpu - max CPU time (MIN)
# - nproc - max number of processes
# - as - address space limit (KB)
# - maxlogins - max number of logins for this user
# - maxsyslogins - max number of logins on the system
# - priority - the priority to run user process with
# - locks - max number of file locks the user can hold
# - stipending - max number of pending signals
# - msgqueue - max memory used by POSIX message queues (bytes)
# - nice - max nice priority allowed to raise to values: [-20, 19]
# - rtprio - max realtime priority
# - chroot - change root to directory (Debian-specific)
#
##<domain>      <type>  <item>      <value>
#
##
#*              soft    core        0
#root           hard    core        100000
#*              hard    rss         10000
#@student       hard    nproc       20
#@faculty       soft    nproc       20
#@faculty       hard    nproc       50
#ftp            -       nproc       0
#ftp            -       chroot      /ftp
#@student       -       maxlogins   4
# End of file
@audio - rtprio 99
@audio - memlock unlimited
@audio - nice -19
```

Figura B.2 Agregación del texto al archivo limits.config

Guardamos y cerramos el archivo

Paso 3

Incluimos a nuestro usuario en el grupo de audio con el siguiente comando

```
sudo usermod -a -G audio $(whoami) como se observa en la Figura
```

B.3

```
root@ubuntu: /home/tiago
Desempaquetando cdparanoia (de ../cdparanoia_3.10.2+debian-11_amd64.deb) ...
Selecting previously unselected package python-cddb.
Desempaquetando python-cddb (de ../python-cddb_1.4-5.1ubuntu2_amd64.deb) ...
Selecting previously unselected package python-eyed3.
Desempaquetando python-eyed3 (de ../python-eyed3_0.6.18-1_all.deb) ...
Selecting previously unselected package python-ogg.
Desempaquetando python-ogg (de ../python-ogg_1.3+repack-5ubuntu2_amd64.deb) ...
Selecting previously unselected package python-pyvorbis.
Desempaquetando python-pyvorbis (de ../python-pyvorbis_1.5-1ubuntu2_amd64.deb)
...
Selecting previously unselected package jack.
Desempaquetando jack (de ../jack_3.1.1+cvcs20050801-29_amd64.deb) ...
Procesando disparadores para man-db ...
Configurando cdparanoia (3.10.2+debian-11) ...
Configurando python-cddb (1.4-5.1ubuntu2) ...
Configurando python-eyed3 (0.6.18-1) ...
Configurando python-ogg (1.3+repack-5ubuntu2) ...
Configurando python-pyvorbis (1.5-1ubuntu2) ...
Configurando jack (3.1.1+cvcs20050801-29) ...
Procesando disparadores para python-support ...
root@ubuntu:/home/tiago/Documentos/jperf-2.0.2# cd ..
root@ubuntu:/home/tiago/Documentos# cd ..
root@ubuntu:/home/tiago# gedit /etc/security/limits.conf
root@ubuntu:/home/tiago# usermod -a -G audio $(whoami)
```

Figura B.3 Inclusión del usuario al grupo de audio

Paso 4

Creamos un archivo de configuración básico para que el servidor Jack este siempre disponible para cualquier aplicación que lo solicite.

Desde una terminal ejecutamos el siguiente comando:

echo "/usr/bin/jackd -d alsa -r 44100 -p 256" > ~/.jackdrc como se

observa en la Figura B.4

```
root@ubuntu: /home/tiago
tiago@ubuntu:~$ sudo su
[sudo] password for tiago:
root@ubuntu:/home/tiago# echo "/usr/bin/jackd -d alsa -r 44100 -p 256" > ~/.jack
drc
```

Figura B.4 Creación de archivo para Jack

Paso 5

Instalamos QJACK desde una terminal con el siguiente comando:

sudo apt-get install qjackctl como se observa en la Figura B.5

```
root@ubuntu: /home/tiago
tiago@ubuntu:~$ sudo su
[sudo] password for tiago:
root@ubuntu:/home/tiago# echo "/usr/bin/jackd -d alsa -r 44100 -p 256" > ~/.jack
drc
root@ubuntu:/home/tiago# apt-get install qjackctl
```

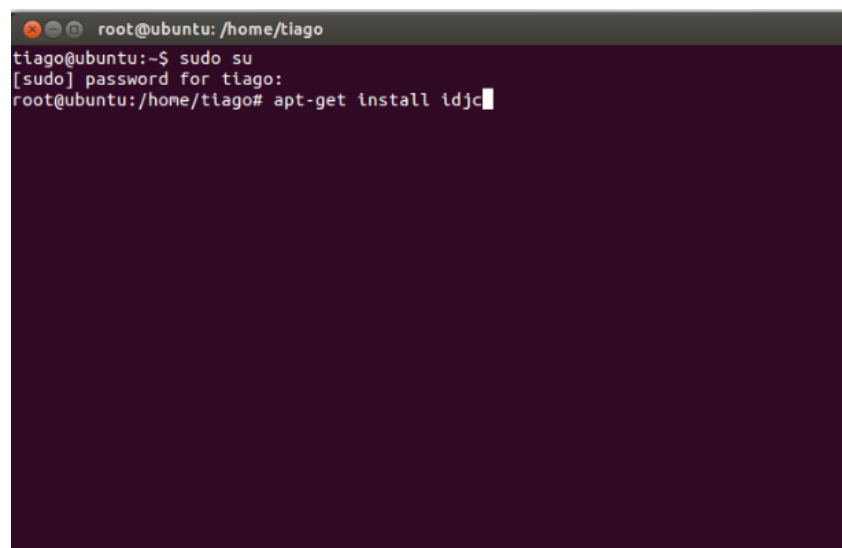
Figura B.5 Instalación de QJACK

Luego de la instalación lo podremos encontrar en Aplicaciones -> QjackCtl

Paso 6

Instalamos IDJC (Internet DJ Console) desde una terminal con el siguiente comando

sudo apt-get install idjc como se observa en la Figura B.6

A terminal window with a dark purple background. The title bar shows 'root@ubuntu: /home/tiago'. The terminal text shows a user named 'tiago' at 'ubuntu' running 'sudo su', followed by a password prompt '[sudo] password for tiago:' which is obscured. The prompt then changes to 'root@ubuntu: /home/tiago#' and the user enters the command 'apt-get install idjc' with a cursor at the end.

```
root@ubuntu: /home/tiago
tiago@ubuntu:~$ sudo su
[sudo] password for tiago:
root@ubuntu: /home/tiago# apt-get install idjc
```

Figura B.6 Instalación de IDJC

Luego de la instalación lo podremos encontrar en Aplicaciones -> Internet DJ Console

Paso 7

Configuramos IDJC desde el botón que se indica en la imagen (Open various application windows) tenemos acceso a varias ventanas de

configuración de parámetros (Preferences, Output, Jingles, Profiles).
Seleccionamos Output (salida), como se ve podemos configurar hasta 6 servidores stream. Esto se visualiza en la Figura B.7

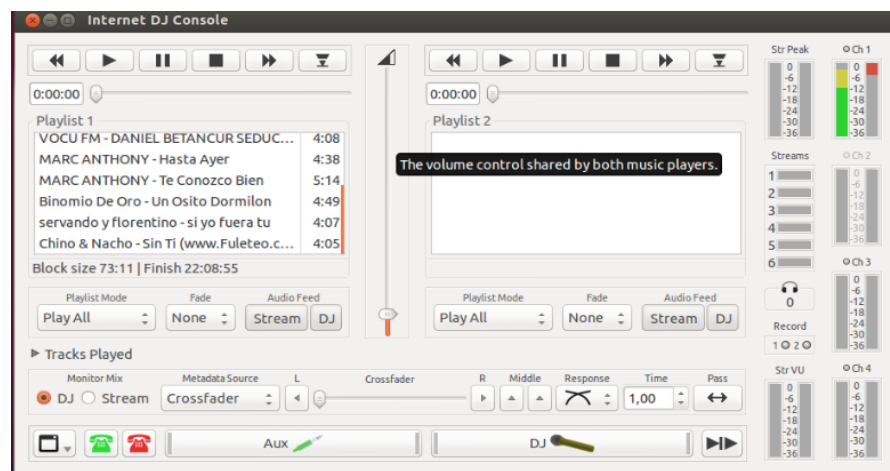


Figura B.7 Interfaz gráfica de IDJC

Procedemos a desplegar > Configuration para configurar la salida de audio hacia el servidor Icecast2.

Paso 8

Configuramos la Pestaña Connection

Hacemos clic en Nuevo e introducimos los parámetros que se indican en la siguiente imagen.

Server type: Icecast2 Master

Hostname: ponemos la URL desde donde se realiza la emisión (en nuestro caso localhost)

Port number: el puerto (en este caso 8000)

Mount Point: /listen (ponemos el punto de montaje /listen)

Logging name: source

Password: (ponemos el mismo password que configuramos en el archivo /etc/icecast2/icecast.xml para las fuentes -sources-)

Esta configuración se observa en la Figura B.8

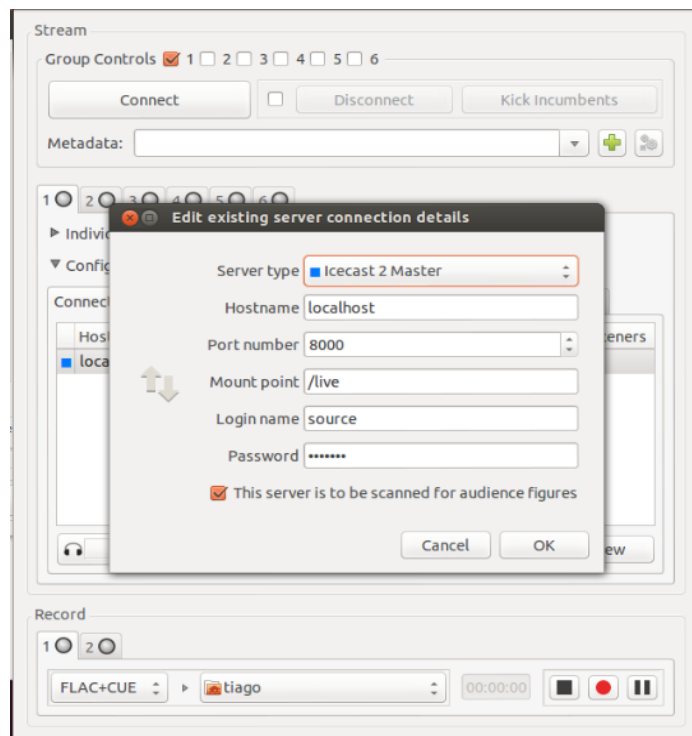


Figura B.8 Configuración del servidor para IDJC

Paso 9

Configuramos la pestaña Format

Sample rate: se nos mostrarán varias opciones, incluso dejar que sea JACK quién lo defina.

Mp3/ogg: indicamos la familia de formatos de salida.

Ogg/vorbis , OggFLAC , Speex Speech: son formatos transmisión en vivo de emisión y grabación.

Bitrate: indicamos el bitrate de salida.

Esta configuración se observa en la Figura B.9

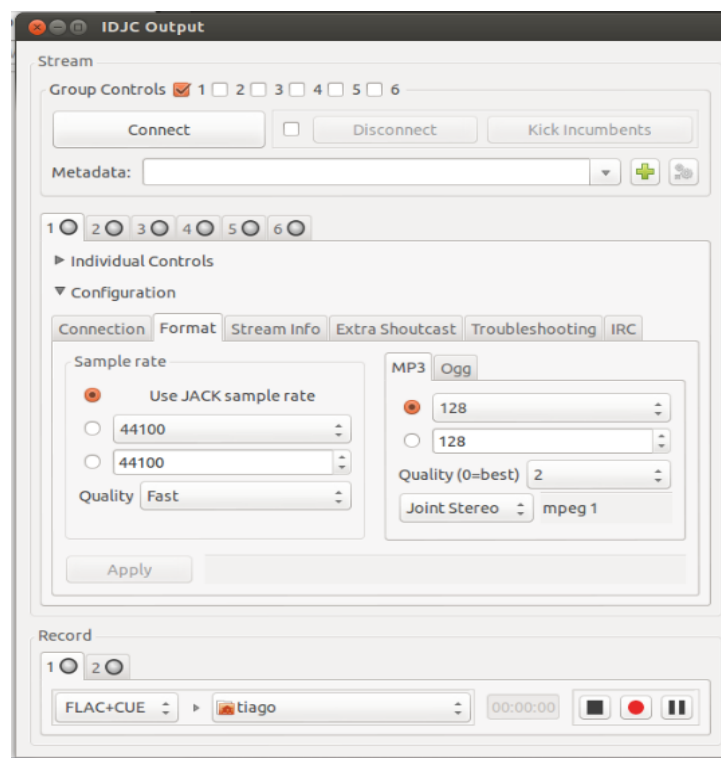


Figura B.9 Configuración de los formatos para IDJC

Paso 10

Configuramos la pestaña Stream Info

En esta pestaña configuramos los datos que proporcionaremos a los clientes durante la emisión.

Listen URL: /listen (debemos poner el punto de montaje “Mount Point” que hemos introducido en la Pestaña Connection)

Esta configuración se observa en la Figura B.10

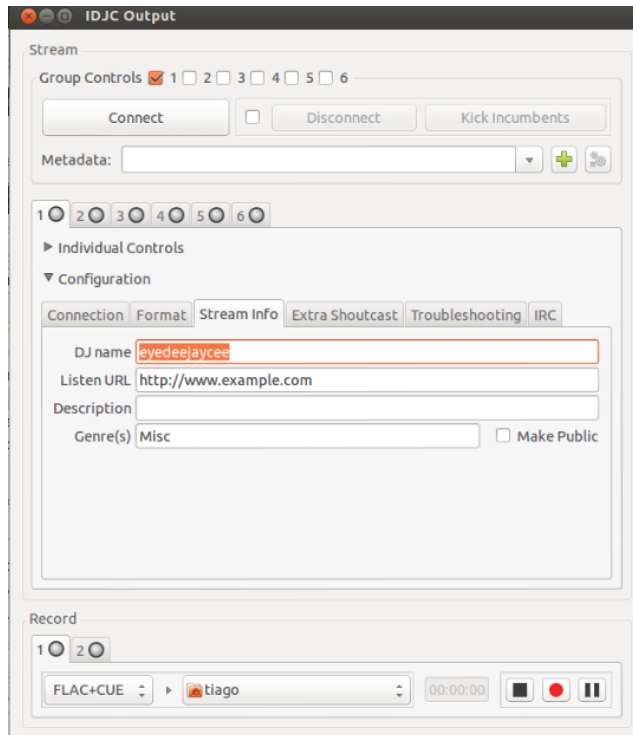


Figura B.10 Configuración de la información para cliente para IDJC

Paso 11

Desplegamos la pestaña > Individual Controls

Metadata (Format string): %S (asi podremos transmitir la información sobre la canción que se está reproduciendo)

Con esto hemos configurado los parámetros esenciales para comenzar a emitir. Ahora vamos a activar/detener la salida hacia el servidor IceCast2. Podemos activarla desde el botón que nos muestra la URL y punto de montaje -> (localhost: 8000/live); y también desde el botón Connect. Si hemos configurado correctamente los parámetros anteriores, en cuanto lo activemos veremos que su color cambia a verde lo que significa que la transmisión ha empezado. Esto se visualiza en la Figura B.11

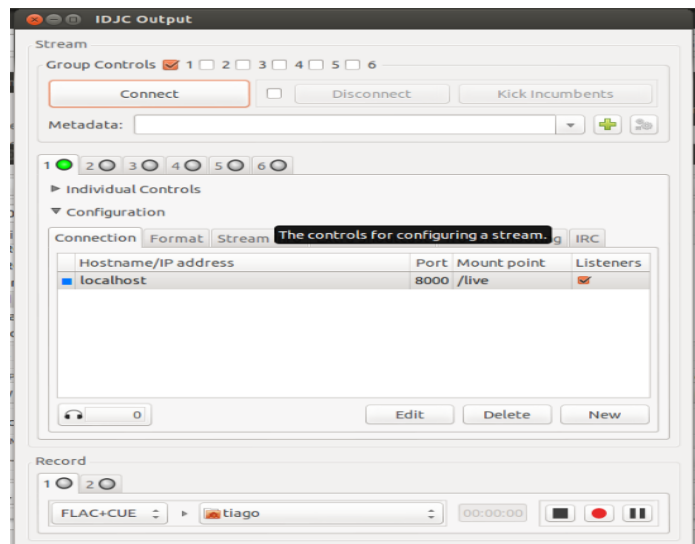


Figura B.11 Activar la emisión en vivo para IDJC