

Unidad 7



Servicios de
audio y vídeo
en streaming

Servicios de
red e internet



Índice



7.1. Funcionamiento del servicio de audio streaming

7.1.1. Formatos de audio

7.2. Servidores de streaming de audio

7.2.1. Instalación del servidor de streaming de audio

7.3. Sindicación y suscripción de audio: podcast

7.3.1. Temáticas y plataformas

7.4. Formatos de imagen

7.5. Formatos de video y códecs

7.6. Las videoconferencias

7.7. Servidores de video y videoconferencia

7.8. Sindicación y suscripción de video



Introducción

En la actualidad, las plataformas de streaming como Twitch son de los principales portales web con más uso en el mundo.

El streaming como servicio permite tanto la compartición de audio como de video de manera que el usuario pueda tener cierto contenido disponible en la web o la aplicación de manera persistente o volátil, depende.

En esta unidad veremos las principales funciones de estos servidores y como configurarlos en ambos casos, solo con audio o con audio y vídeo, hablando de manera más específica de las videoconferencias y su funcionamiento.

Por último, en ambos casos se verá como la sindicación de contenidos está cobrando hoy en día también una importancia absoluta.

Al finalizar esta unidad

- + Seremos capaces de describir la funcionalidad del servicio de audio y vídeo.
- + Sabremos instalar y configurar un servidor de distribución de audio y vídeo.
- + Conoceremos como configurar el cliente para el acceso al servidor de audio y vídeo.
- + Podremos reconocer y utilizar los formatos de audio y vídeo digital.
- + Sabremos usar herramientas de reproducción de audio y vídeo en el cliente.
- + Conoceremos como emplear servicios de audio y vídeo a través de un navegador web.
- + Podremos usar técnicas de sindicación y suscripción de audio.
- + Seremos capaces de elaborar documentación relativa a la instalación y administración del servidor de audio y vídeo.



7.1.

Funcionamiento del servicio de audio streaming

El servicio de audio en *streaming* se basa en poder enviar sin interrupciones, al menos las menos posibles, audio desde un servidor hacia clientes en tiempo real. Los clientes pueden generar las solicitudes de audio desde una aplicación creada para esto o desde navegadores accediendo a los diferentes portales. Como en oda comunicación, existe el concepto de emisor y receptor, pero en este caso lo normal es que haya un único emisor y varios receptores simultáneos. Los receptores son todos los usuarios conectados al servidor y escuchando la retransmisión en vivo.

Las transmisiones se pueden realizar de dos modos:

- > **Directo:** es como la radio convencional y de toda la vida, se retransmite contenido en vídeo ya sea con micrófonos o ficheros de audio creados anteriormente.
- > **Diferido:** se transmiten solo ficheros de audio normalmente almacenados en *playlist* o listas de reproducción. Se pueden configurar dichas reproducciones para que se retransmita la información aleatoriamente, con repeticiones, saltada, etc.

Indiferente a de qué modo se haga, la información se envía en forma de paquetes de bits que el cliente reproduce en su destino. De esto modo, el cliente no tiene por qué almacenar la información para poder reproducirla, lo hace a medida que le va llegando.

Este servicio suele ser usado sobre todo por:

- > Empresas que se dedica a la retransmisión de eventos.
- > Canales de televisión y radio.
- > Productoras audiovisuales.
- > Profesionales de retransmisión de contenido, los llamados *streamers*.

7.1.1. Formatos de audio

Cuando se almacena el audio que se va a transmitir en el servidor, normalmente se encuentra en formato comprimido para no consumir mucho ancho de banda a la hora de transmitirlo. Los formatos más comunes de uso de audio son:

- > **MP3.** Su nombre viene derivado de *MPEG-1 Audio Layer III* o *MPEG-2 Audio Layer III*. Su función se basa en la aplicación de un algoritmo de compresión de audio con pérdidas. En este se aplican técnicas de muestreo para eliminar frecuencias que el cerebro desecha, aunque se reciba por el oído. Gracias a todo esto, los ficheros de audio en MP3 ocupan mucho menos que un fichero con toda la información real, aunque parte de la información sea para desear.
- > **OGG.** Este formato de compresión es de código abierto y de tipo contendido. Además de audio, también puede almacenar vídeo, texto y metadatos. Aunque no elimina tanta información innecesaria como MP3, la información obtiene una mayor calidad. Como contiene demás datos, también se usa para incluir subtítulos a algunos vídeos.

7.2.

Servidores de streaming de audio

Los servidores de *streaming* de audio trabajan con un proceso del sistema que se encuentran a la espera de distintas solicitudes de paquetes de difusión de la información. Los servidores se basan para la transmisión de la información en el protocolo RTSP, al igual que pasaba con VoIP. Muchos de ellos se encuentran integrados en un servicio web, por lo que, si se quiere solicitar el audio, se hará mediante solicitudes http.

En el siguiente diagrama podemos ver una descripción gráfica del funcionamiento de un servidor de *streaming* de audio:

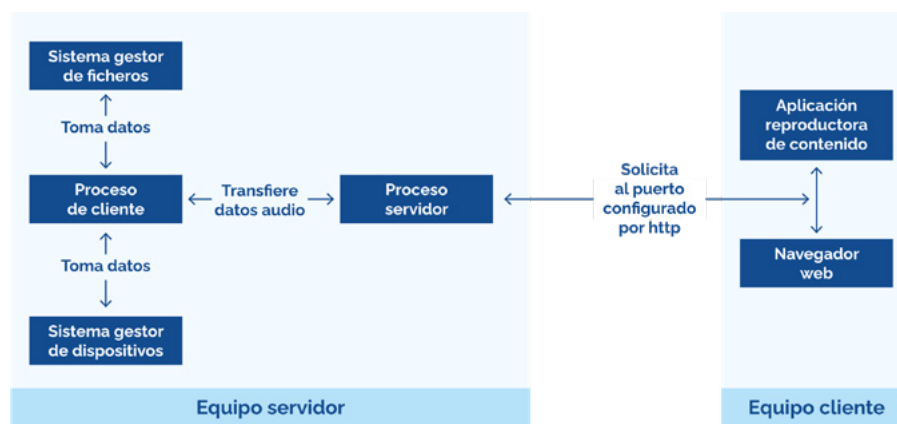


Imagen 1. Funcionamiento de un servidor *streaming* de audio

7.2.1. Instalación del servidor de streaming de audio

Vamos a comenzar a trabajar con un servidor de *streaming* de audio. En Ubuntu, como solución a esto proponemos *Icecast2*, ya que es sencillo y existe una gran cantidad de documentación en la red. Pero, aunque instalemos dicho servicio necesitamos de otro paquete adicional que nos permita emitir lo que deseamos, para lo que usaremos *ices2*. Por último, también deberemos instalar un paquete de herramientas que permiten usar ficheros *.ogg*.

Los pasos que debemos de seguir son los siguientes:

1. Actualizamos los repositorios y paquetes de Ubuntu.
2. Instalamos los siguientes paquetes: *icecast2*, *ices2* y *vorbis-tools*.
 - » Se nos pedirá configurar *IceCast2*, indicamos que no para configurarlo posteriormente de manera manual.

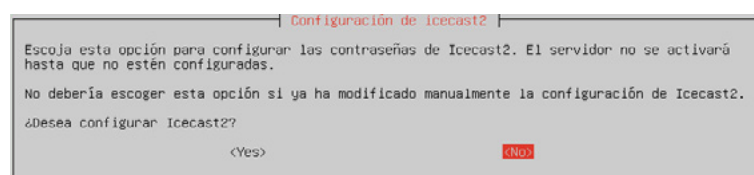


Imagen 1. Instalación de servicio de *streaming* de audio 1



- Al instalarse los paquetes anteriores, se generan los siguientes ficheros:

» `/etc/icecast2/icecast.xml`: el principal fichero de configuración del servicio *Icecast*.

```
GNU nano 6.2 /etc/icecast2/icecast.xml
<icecast>
  <!-- location and admin are two arbitrary strings that are e.g. visible
        on the server info page of the icecast web interface
        (server_version.xml). -->
  <location>Earth</location>
  <admin>icemaster@localhost</admin>

  <!-- IMPORTANT!
        Especially for inexperienced users:
        Start out by ONLY changing all passwords and restarting Icecast.
        For detailed setup instructions please refer to the documentation.
        It's also available here: http://icecast.org/docs/
  -->

  <limits>
    <clients>100</clients>
    <sources>2</sources>
    <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>
```

Imagen 2. Instalación de servicio de *streaming* de audio 2

» `/etc/default/icecast2`: es el fichero que almacena donde se encuentra el principal fichero de configuración de *Icecast* además de algún parámetro concreto.

```
GNU nano 6.2 /etc/default/icecast2
# 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
```

Imagen 3. Instalación de servicio de *streaming* de audio 3

» `/usr/share/doc/ices2/examples`: este directorio contiene distintos ficheros que sirven de ejemplos de configuración de *Ices2*. Para cada tipo de entrada contenido tenemos un fichero distinto:

```
root@universae:~# ls /usr/share/doc/ices2/examples/
ices-alsa.xml  ices-init.example  ices-oss.xml  ices-playlist.xml  ices-roar.xml
root@universae:~#
```

Imagen 4. Instalación de servicio de *streaming* de audio 4

- + `ices-playlist.xml`: se pueden configurar listas de ficheros `.mp3` o `.ogg`.
- + `ices-alsa.xml`: para configurar la entrada de micrófono.
- + `ices-oss.xml`: para configurar la entrada mediante una lista de ficheros `.oss`.



4. Pasamos a configurar el fichero de configuración anteriormente creado:
 - » En la directiva `<authentication>`, indicamos la contraseña de administrador del servicio y las contraseñas para el acceso a los puntos de montaje.

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

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

Imagen 5. Configuración de servicio de *streaming* de audio 1

- » La directiva `<hostname>` se usa para indicar la dirección del servidor de *streaming* de audio, por defecto es `localhost` y en nuestro caso también.

```
<hostname>localhost</hostname>
```

Imagen 6. Configuración de servicio de *streaming* de audio 2

5. El fichero que tenemos que configurar ahora es `/etc/default/icecast2`, en dicho fichero se va a habilitar el servicio de *streaming* de audio añadiendo la directiva `ENABLE=TRUE`.

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

ENABLE=true
```

Imagen 7. Configuración de servicio de *streaming* de audio 3

6. Reiniciamos el servicio y comprobamos que todo funciona correctamente.

```
root@universae:~# systemctl restart icecast2
root@universae:~# systemctl status icecast2.service
* icecast2.service - LSB: Icecast2 streaming media server
   Loaded: loaded (/etc/init.d/icecast2; generated)
   Active: active (exited) since Wed 2022-09-14 06:18:40 UTC; 8s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 2293 ExecStart=/etc/init.d/icecast2 start (code=exited, status=0/SUCCESS)
    CPU: 17ms

sep 14 06:18:39 universae systemd[1]: Starting LSB: Icecast2 streaming media server...
sep 14 06:18:40 universae icecast2[2293]: * Starting streaming media server: icecast2
sep 14 06:18:40 universae icecast2[2293]: ...done.
sep 14 06:18:40 universae systemd[1]: Started LSB: Icecast2 streaming media server.
sep 14 06:10:40 universae icecast2[2290]: /etc/icecast2/icecast.xml:40: parser error : Opening and
sep 14 06:10:40 universae icecast2[2290]: <admin-password>Universae</admin-password>
sep 14 06:18:40 universae icecast2[2298]:
lines 1-14/14 (END)
```

Imagen 8. Configuración de servicio de *streaming* de audio 4

7. El siguiente paso viene dado por la configuración de algún contenido en concreto para *Icecast2*, es decir, para *ices2*. Lo que se va a proponer en el ejemplo es la emisión en diferido de unos ficheros de audio .ogg, por lo que usaremos el fichero *ices-playlist.xml*:

- » Se realiza una copia de seguridad de los ficheros de ejemplo de *ices2* en el directorio (que habrá que crear), */etc/ices2*:

```
root@universae:~# mkdir /etc/ices2
root@universae:~# cp /usr/share/doc/ices2/examples/* /etc/ices2/
root@universae:~# mkdir /var/log/ices
root@universae:~# ls /etc/ices2/
ices-alsa.xml ices.init.example ices-oss.xml ices-playlist.xml ices-roar.xml
root@universae:~#
```

Imagen 9. Configuración de servicio de streaming de audio 5

- » Se trabaja en este momento en este nuevo directorio creado.
- » Se otorgan a dicho directorio los permisos necesarios para la ejecución de los ficheros necesarios.
- » En el fichero */etc/ices2/ices-playlist.xml*, se editan los siguientes puntos:
 - + El parámetro *<background>* se coloca en uno para que se ejecute siempre el servicio en segundo plano.

```
<?xml version="1.0"?>
<ices>
  <!-- run in background -->
  <background>1</background>
  <!-- where logs, etc go. -->
```

Imagen 10. Configuración de servicio de streaming de audio 6

- + La sección *<metadata>* indica los valores que se establecen para buscadores, como nombre, genero, etc.

```
<stream>
  <!-- metadata used for stream listing (not currently used) -->
  <metadata>
    <name>Playlist de Universae</name>
    <genre>Playlist de Universae</genre>
    <description>Las canciones más escuchadas por los docentes</description>
  </metadata>
```

Imagen 11. Configuración de servicio de streaming de audio 7

- + La sección *<input>* será la que se editará, en el parámetro *file* para indicar de que fichero se deben de coger los ficheros de audio que se van a reproducir. Los demás los podemos dejar por defecto ya que son parámetros de reproducción.

```
<input>
  <module>playlist</module>
  <param name="type">basic</param>
  <param name="file">/etc/ices2/canciones.txt</param>
  <!-- random play -->
  <param name="random">0</param>
  <!-- if the playlist get updated that start at the beginning -->
  <param name="restart-after-reread">0</param>
  <!-- if set to 1, plays once through, then exits. -->
  <param name="once">0</param>
</input>
```

Imagen 12. Configuración de servicio de streaming de audio 8

- + La sección *<instance>* será en la que se deban rellenar los datos del servidor tal y como hiciémos en el primer fichero.

```
<instance>
  <!-- Server details:
  You define hostname and port for the server here, along with
  the source password and mountpoint. -->
  <hostname>localhost</hostname>
  <port>8000</port>
  <password>Universae</password>
  <mount>/streaming</mount>
  <yp>1</yp> <!-- allow stream to be advertised on YP, default 0 -->
```

Imagen 13. Configuración de servicio de streaming de audio 9

- » Por último, en el fichero que se ha creado para la reproducción de los audios, se indica por cada línea, cada una de las ubicaciones de los distintos audios.

```
GNU nano 6.2 /etc/ices2/canciones.txt *
/musica/Always_Bon_Jovi.ogg
/musica/I_dont_wanna_miss_a_thing_Aerosmith.ogg
/musica/Heaven_Bryan_Adams.ogg
```

Imagen 14. Configuración de servicio de streaming de audio 10

8. Reiniciamos los servicios

9. Para ejecutar el servicio lanzamos el comando siguiente:

ices2 ruta_fichero.xml &

10. Los clientes se podrán conectar con los distintos reproductores de audio que elijan usando la siguiente URL:

http://dirección_servidor/punto_montaje

7.3.

Sindicación y suscripción de audio: podcast

Las suscripciones *Podcast* se basan en adquirir cierto contenido multimedia que el usuario puede reproducir en cualquier momento. El servicio de *podcast* permite que haya listas de contenido actualizado o actualizable por un propietario de dicho contenido que lo difunde a sus suscriptores para que puedan, en diferido, disfrutar de todo el contenido.

La siguiente ilustración nos permite ver cuál es el funcionamiento de estos servicios:



Imagen 15. Funcionamiento del servicio de *podcast*

Si tenemos que decir una diferencia entre este servicio y el de *streaming* es el flujo de transmisión de la información. Mientras que en los *podcasts* la información se transmite en diferido y se puede reproducir varias veces, pararla, volver atrás, etc.; en el servicio de *streaming* no se puede volver a reproducir el contenido que se ha visualizado, independientemente de si es en directo o no la emisión.

Por lo general, los *podcasts* suelen organizarse por temáticas, para hacer más fácil la elección al usuario.

Los ficheros de *podcast* se almacenan por lo general en formato *.mp3*, pero en la actualidad existen también *podcast* con formatos de video.

7.3.1. Temáticas y plataformas

Las tres temáticas principales de los *podcasts* son las siguientes:

- > **Entretenimiento:** los contenidos pueden ser humorísticos, de divulgación, de turismo, de ocio, de videojuegos, etc.
- > **Información:** los contenidos que se emiten suelen ser a nivel informativo actual.
- > **Formativos:** los contenidos emitidos se basan en fines educativos y de aprendizaje. Existen varias ramas y especialidades dentro de esto para los distintos conocimientos que queramos adquirir. Muchas veces se tratan de tutoriales o audioguías.

Aunque son muchas las plataformas de emisión de *podcast* hoy en día priman dos, *Apple Music* y *Spotify*.



7.4.

Formatos de imagen

Los formatos más comunes son:

Formato pixelado

Es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color, denominada matriz, que se puede visualizar en un monitor, papel u otro dispositivo de representación que, al contrario que las vectoriales, pierden calidad al ser aumentadas. Los formatos más utilizados son:

Formato píxel	
GIF	Permite animaciones y transparencias. Se sirve de 256 colores.
JPG/JPEG	Es el que suele usar la fotografía (además de RAW, usado por las cámaras profesionales) y cuenta con millones de colores. Suele ser compatible con todos los dispositivos. Es posible comprimirlo estableciendo el nivel de calidad. Ocupa menos espacio, pero no guardan una copia perfecta de la imagen. Se usa mucho en web, por el menor peso y tamaño del archivo.
BMP	Se trata del formato propio de Paint (Microsoft). Se sirve de formatos de gran tamaño.
TIFF	Generalmente se utiliza para impresiones de gran formato por su eficiencia con imágenes de grandes resoluciones. Es usado para el guardar imágenes en las que aparecen muchos colores y detalles.
PNG	Este tipo de formato libre permite transparencias, es decir, incluye transparencias sin perder calidad
WEBP	Desarrollado por Google con el fin de mejorar la calidad y reducir el tamaño para aumentar la velocidad de carga de cada web.
PSD	Nativo de Photoshop. Mantiene las capas del proyecto.
RAW	Es el "bruto" de las cámaras réflex

Formatos vectoriales

Para el diseño de logotipos o imágenes corporativas es necesario servirse de diversas resoluciones dentro de una misma imagen. Cuando se requiere de una calidad más grande por el tamaño de la impresión, se debe servir de este tipo de formatos.

Algunos de los formatos más usados son:

- > **Formato AI:** el formato de referencia, de Adobe Illustrator.
- > **Formato SVG:** muy utilizado actualmente, de W3C. Está optimizado como lenguaje de programación.
- > **Formato EPS:** el más común, y, por tanto, el más compatible con otros programas. Es de Adobe.
- > **Formato PDF:** usado como formato de documento, también se puede usar como imagen vectorial.

Formatos de propietario

Este tipo de formato se utilizan en programas de diseño que permiten la edición original para sus posibles modificaciones.

Formatos propietarios	
AI	Illustrator
PSD	Photoshop
CDR	Corel Draw
XCF	Gimp

Aplicaciones de tratamiento de imágenes como Paint, Photoshop o Gimp permiten guardar los archivos en diversos formatos y resoluciones.



7.5.

Formatos de vídeo y códecs

Los formatos de vídeo son contenedores que incluyen archivos de vídeo, audio, texto e imágenes, que al juntar todos estos archivos nos da un resultado con gran cantidad de información y aquí es donde entran en juego los códecs.

Un códec es el código utilizado para diseñar los datos de un archivo de audio o vídeo de tal manera que se pueda utilizar para reproducir, editar o cambiar a otros códecs (transcodificación).

Los códecs se utilizan para organizar los datos de los medios, pero esos datos se mantienen dentro de un contenedor. Hay muchos tipos diferentes de códecs de audio y vídeo, y cada uno tiene sus propias ventajas.

Gracias a ellos se comprime y descomprime un archivo. Esta compresión puede ser con pérdida o sin pérdida, sobre todo en audio, ya que en vídeo un formato sin pérdida pesaría demasiado para uso convencional.

Aunque hayan sido ya vistos, vamos a explicar con mayor detalle los formatos más utilizados:

MP4

Este tipo de formato de vídeo es uno de los formatos más utilizados en la actualidad, especialmente cuando se trata de compartir contenido en línea. De hecho, YouTube recomienda cargar archivos en formato .mp4 para obtener la mejor calidad de vídeo.

Además de los datos de vídeo y audio, también se puede usar para almacenar cosas como subtítulos e imágenes fijas. Por lo general, se combina con H.264 o H.265.

Los vídeos que usan el contenedor .MP4 pueden tener tamaños de archivo relativamente pequeños mientras conservan una alta calidad.

AVI

Uno de los tipos de formato de vídeo estándar más antiguos y universalmente aceptados es .AVI. Puede utilizar una enorme variedad de códecs, lo que da como resultado una gran variedad de configuraciones de archivo diferentes. Si bien los vídeos .AVI se pueden reproducir en una amplia gama de reproductores, los tamaños de archivo tienden a ser grandes, lo que los hace menos ideales para la transmisión o descarga. Es una excelente opción para los vídeos que planeas almacenar en una computadora.

Microsoft desarrolló y lanzó esto con Windows 3.1 hace mucho tiempo.



MKV

El formato de video profesional MKV es un formato abierto y cuyo nombre completo es *Matroska*.

MKV surgió a finales de 2002 y se ha popularizado gracias a que dentro de un único archivo MKV podemos almacenar, junto con el canal de audio, varios canales o pistas de audio y varias pistas de subtítulos.

Igual que MP4, proporciona muy buena calidad de audio y video en poco espacio

FLV

Dentro de los tipos de formato de video que existen, FLV fue creado para el reproductor Flash de Adobe, los videos .FLV fueron extremadamente comunes durante varios años gracias a su tamaño de archivo muy pequeño y una amplia gama de complementos de navegador y reproductores de video Flash de terceros. Recientemente, ha habido una disminución significativa en los videos Flash.

MOV

Dentro de los tipos de formato de video para Apple, es el mejor. Apple desarrolló el contenedor .MOV para usar con su reproductor QuickTime. Los videos que usan .MOV generalmente tienen una calidad muy alta, pero también tamaños de archivo bastante grandes. Los videos de QuickTime no tienen tanta compatibilidad con los reproductores que no son de QuickTime, aunque hay reproductores de terceros que los leerán.

WMV

Los videos de Windows Media tienden a tener el tamaño de archivo más pequeño, lo que los convierte en una buena opción si necesitas enviarlos por correo electrónico u otros métodos con límites de tamaño de archivo. Sin embargo, esto viene con la compensación de tener una caída significativa en la calidad. Un uso común de .WMV es enviar vistas previas de videos por correo electrónico a los clientes.

H.264

Más que un formato, es el códec de video más usado, conocido como MPEG-4, H.264 usa compresión con pérdida de uso común en la actualidad. El códec es ampliamente compatible y se utiliza en la producción, publicación y distribución de video. Muchas videocámaras y DSLR graban en H.264. Es el estándar para discos Blu-ray, así como para muchos servidores de video web. H.264 es más eficiente para la compresión que MPEG-2 y normalmente ofrece una mejor calidad de video con la misma tasa de bits.

Este códec se utiliza a menudo con contenedores .mp4 y .mov.

Así pues, algunos de los códec más usados serían:

Códec más utilizados	
DIVX	Códec extendido en la red que ofrece alta calidad, además tiene una versión profesional de pago.
XDIV	Códec opensource que compite en calidad con DivX
MPEG	Los códec que se asocian a los formatos mpg, DivX es un derivado de Mpeg-4



7.6.

Las videoconferencias

Las videoconferencias son uno de los mayores medios de uso del *streaming* de video y audio. Las transmisiones en tiempo real de video y audio entre dos personas siempre tienen un emisor y varios receptores, como los *webinars* o un emisor y un receptor, una videollamada al uso.

Tenemos dos modos de hacer videoconferencias dependiendo del medio o del dispositivo que usemos:

- > **Usando aplicaciones de escritorio:** se usan aplicaciones instaladas en el equipo como puede ser *Microsoft teams*.
- > **Usando dispositivos profesionales:** se usan ciertos dispositivos diseñados para esta función. - Dichos dispositivos se conectarán a red mediante el protocolo IP, pero usarán por lo general el protocolo SIP explicado en la unidad anterior y códecs de audio y vídeo estándares. Sony destaca como fabricante de dichos dispositivos.

7.7.

Servidores de vídeo y videoconferencia

Los servicios de video en *streaming* tienen como función principal que el usuario reciba contenido de vídeo, ya sea en directorio o que el usuario pueda acceder a una colección de vídeos para verlos en cualquier momento.

Como *software* libre para este servicio lo más implementado es *Streama*. Esta plataforma de código abierto tipo OTT proporciona contenido de vídeo en *streaming* y se basa en Java. Además, se pueden acceder a los materiales mediante un navegador web.

Para instalar *Streama* en Ubuntu necesitamos tener instalado el entorno de ejecución en Java. El fabricante aconseja que se tenga la versión 8, por lo que el paquete que debemos de instalar es *openjdk-8-jre*.

Si ya hemos instalado el entorno de trabajo, lo recomendable es crear dos directorios para el almacenamiento de los datos de *Streama*, por lo que usaríamos los siguientes comandos:

```
mkdir /data  
mkdir /data/streama
```

El propietario de estos directorios debe ser el usuario que vaya a ejecutar *Streama*, por lo que habrá que usar el comando *chown* para cambiar dicho propietario.



Es momento de instalar *Streama*:

1. Nos dirigimos al directorio que acabamos de crear con el comando `cd`.
2. En *GitHub*, vemos cual es la ultima versión disponible de *Streama* desde la página web <https://github.com/streamaserver/streama/releases>.
3. Usamos el siguiente comando para descargar en dicho directorio la última versión:

`wget dirección_descarga`

```
root@universae:/data/streama# wget https://github.com/streamaserver/streama/releases/download/v1.10.4/streama-1.10.4.jar
--2022-09-14 10:07:48-- https://github.com/streamaserver/streama/releases/download/v1.10.4/streama-1.10.4.jar
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/39890823/c5be662c-a27b-49d7-aa97-b7c256e0251e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220914%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220914T1007482&X-Amz-Expires=300&X-Amz-Signature=75511f30a907a500a3e4dc17fe17d4a59e39cadc1629eae0d39436239200407a&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=39890823&response-content-disposition=attachment%3B%20filename%3Dstreama-1.10.4.jar&response-content-type=application%2Foctet-stream [following]
--2022-09-14 10:07:48-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/39890823/c5be662c-a27b-49d7-aa97-b7c256e0251e?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220914%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220914T1007482&X-Amz-Expires=300&X-Amz-Signature=75511f30a907a500a3e4dc17fe17d4a59e39cadc1629eae0d39436239200407a&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=39890823&response-content-disposition=attachment%3B%20filename%3Dstreama-1.10.4.jar&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 103082468 (98M) [application/octet-stream]
Saving to: 'streama-1.10.4.jar'

streama-1.10.4.jar      100%[=====>] 98,31M  3,15MB/s  in 27s

2022-09-14 10:08:16 (3,59 MB/s) - 'streama-1.10.4.jar' saved [103082468/103082468]

root@universae:/data/streama# _
```

Imagen 15. Servicio de streaming de video 1

4. Cuando ya se ha descargado, se cambian los permisos del fichero para que se pueda ejecutar.
5. Por último, se ejecuta el fichero de Java con el comando:

`java -jar fichero.jar`

```
root@universae:/data/streama# chmod +x streama-1.10.4.jar
root@universae:/data/streama# java -jar streama-1.10.4.jar
INFO streama.Application - Starting Application on universae with PID 5292 (/data/streama/streama-1.10.4.jar started by root in /data/streama)
DEBUG streama.Application - Running with Spring Boot v1.4.4.RELEASE, Spring v4.3.6.RELEASE
INFO streama.Application - The following profiles are active: production

Configuring Spring Security Core ...
... finished configuring Spring Security Core

Configuring Spring Security LDAP ...
... finished configuring Spring Security LDAP

INFO grails.app.services.streama.MigrationService - SQL Executed, 0 rows Updated
INFO streama.Application - Started Application in 27.716 seconds (JVM running for 30.1)
Grails application running at http://localhost:8080 in environment: production
```

Imagen 16. Servicio de streaming de video 2

Se puede acceder al servicio mediante el navegador usando la URL `http://dirección_servidor:8080`.



7.8.

Sindicación y suscripción de video

Como pasa con la sindicación de audio, el video permite que se use el *streaming* almacenado en un servidor para que este se pueda consumir desde distintas localizaciones. Con esto lo que se consigue es que una web de videos pueda verse alimentada por contenidos que se alojan en distintas webs.

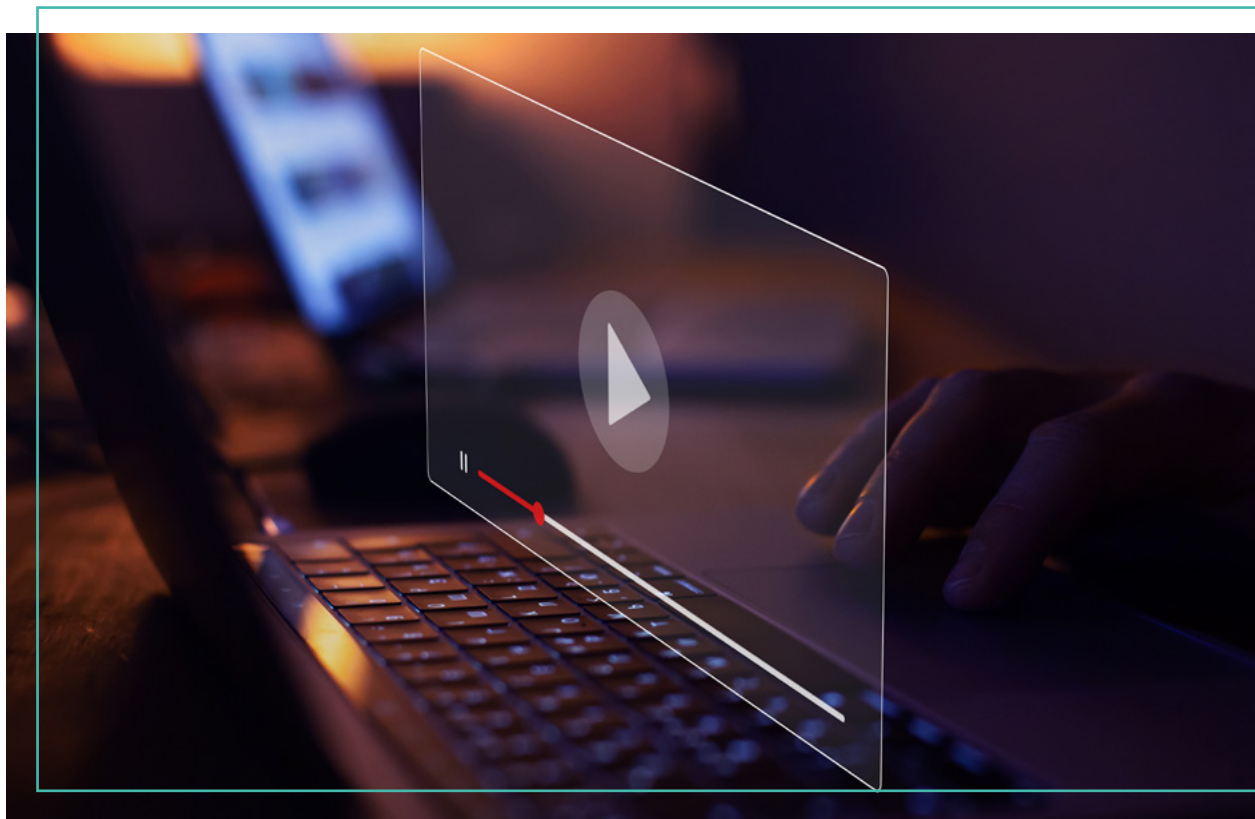
Este servicio tiene la finalidad de presentar la información de manera cómoda para que el usuario pueda acceder a la que desea de manera sencilla en cada momento.

Existen multitud de páginas que hoy en día se dedican únicamente a recopilar videos para luego mostrarlos como plataforma, ejemplo de esto es *YouTube*.

Usar los videos de *YouTube* de manera que se puedan presentar en nuestra página web es uno de los más valiosos recursos que puede usar una empresa relacionada con el mundo audiovisual, destacando aquí dos modos para realizar esto:

- > **Compartirlo en redes sociales.** *YouTube* permite que sus videos se compartan en las distintas redes sociales.
- > **Incrustando o embebiendo el código en una web.** Mediante el lenguaje de etiquetado HTML se puede incrustar el código que muestra el video en una web.

Se pueden usar las dos alternativas al mismo tiempo, ya que más que excluyentes, son complementarias.





 www.universae.com

