



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## Ofrecer live stream de video y audio (44413 lectures)

Per Joan Ginard, [lumen](http://www.bulma.net) (<http://www.bulma.net>)

Creado el 21/04/2006 19:03 modificado el 21/04/2006 19:03

*Desde la versión 2.2 los servidores icecast2 soportan el stream de vídeo en formato ogg theora. Este apunte nos podrá sobre la pista si queremos ofrecer un servicio de vídeo y audio en directo.*

## Este artículo no tiene poderes de supervaca

Este artículo describe las directrices generales para poner en marcha estos servicios. Por supuesto que no comento todas las opciones de comando posibles ni configuraciones existentes. Sí indico algunos enlaces interesantes. Comenta tus dudas más abajo, pero recordad: *man* es tu amigo e *info* también.

## Las partes

Un sistema de vídeo y audio en directo consta de tres elementos:

- **La fuente de datos:** recoge de algún dispositivo de vídeo o fichero la información del vídeo y audio. Icecast lo llama sources.
- **El servidor de stream:** recibe los datos ya procesados desde la fuente de datos y los ofrece públicamente a quien quiera recibirlos.
- **La aplicación cliente:** se conecta al servidor y muestra el stream a los humanos.

## El servidor de stream

El servidor de stream puede manejar varios streams de diferentes tipos a la vez. Cada uno de los streams posibles tiene una url diferente.

Icecast llama a cada stream un *mountpoint*. Icecast permite definir infinidad de parámetros para cada *mountpoint*. Incluso no hace falta definir parámetros para cada *mountpoint* que queramos ofrecer, sino que estos se creen así como se van añadiendo nuevas fuentes de datos.

Los *sources* se conectarán al servidor icecast2. Este los autenticará y si el *source* quiere emitir por un *mountpoint* que tiene definidos el servidor, se le aplicarán los parámetros que el servidor defina. En caso de que el servidor no tenga definido ese *mountpoint*, lo creará (se puede restringir), con los parámetros que la *source* defina más los valores por defecto.

Para instalar icecast2 (>2.2) en debian, lo tenis a golpe de aptitude. Tranquilos, incluso los más estables [tenéis la versión 2.2](#)<sup>(1)</sup>



, aunque puede que sí tengáis problemas para conseguir las libtheora (libtheora-dev si lo compilais).

Para los que no usáis debian supongo que ya conocéis vuestros propios métodos. Hay [svn](#)<sup>(2)</sup> y [rmp's](#).<sup>(3)</sup>

La [configuración del servidor](#)<sup>(4)</sup> en `/etc/icecast2/icecast.xml` ya está correcta para poder hacer las pruebas en local.

Es buenísimo porque además de guardar todo en logs, ofrece una interfaz web con datos de estado actualizados sobre cada *mountpoint* (máximo de clientes, número de clientes actual, formato en el que estáis emitiendo. Incluso lo podéis personalizar si sabéis algo de xsl.

podéis indicarle al servidor que guarde todo lo que se emita en cualquier *mountpoint* en ficheros locales.

## La fuente de datos

Hay [muchas aplicaciones](#)<sup>(5)</sup> que pueden hacer de *source* para un servidor icecast. Fuentes de audio hay infinidad (de hecho hasta el [mpd](#)<sup>(6)</sup> lo soporta[rá] a partir de la versión que la versión [0.11.2](#)<sup>(7)</sup>) y también el xmms tiene plugis para ello. También os puede resultar útil el ices2, que por defecto captura cualquier sonido que salga por la tarjeta de audio.

Fuentes de vídeo no hay tantas, pero las suficientes. Una de ellas es [freej](#).<sup>(8)</sup> pero incluso el [vlc lo soporta](#)<sup>(9)</sup>. Lo bueno del freej son dos cosas:

1. Soporta a la perfección dispositivos v4l.
  2. Trata a cada fuente fichero como una capa, de manera que puede aplicar efectos entre ellas (transparencias, rotaciones, diferencias...)
  3. Transcodea a ogg theora.
- 
1. es fantástico si queremos usar cámara de vídeo o la señal de una capturadora de vídeo como fuente.
  2. es genial porque así nos será muy fácil incrustar una marca de agua al vídeo o incluso escribir los subtítulos, rotar la imagen, aplicarle efectos dinámicos...
  3. es estupendo, porque normalmente las fuentes suelen venir en crudo, con resoluciones altas o calidades bestiales. Esa señal no la podemos enviar directamente a los clientes (pesa mucho). Así que freej la puede cambiar a un formato más ligero, a la vez que lo emite al servidor.

El vídeo que se envía al servidor icecast puede ir acompañado de audio (por supuesto en ogg vorbis). Freej recoge el audio por defecto desde la entrada que tengamos seleccionada en el mixer. Eso nos permite enviar música *de fondo* cuando no queremos que se oiga el sonido del micro.

Os recomiendo la versión de debian unstable, está en pleno desarrollo todavía. Hay manuales y todas las opciones están ya comentadas en su *man*. Básicamente necesita la url donde se encuentra el servidor icecast (incluido el *mountpoint* por donde queremos que salga) el password para tener permisos en ese servidor y los parámetros sobre la calidad del vídeo y audio que queremos mandar al servidor.

Freej guarda por defecto todo el vídeo que se emita al servidor en un fichero local con la calidad que lo estáis emitiendo, no la calidad de la fuente (es decir, después de transcodear). Por eso el fichero local incluye el audio que grabéis de la tarjeta de audio.

## La aplicación cliente

Las personas en sus casas pueden usar cualquier [reproductor que soporte el stream de vídeo ogg theora](#)<sup>(10)</sup>. Recordad que para los VLC se puede instalar en el innombrable (y está comprobado que funciona con este sistema).



Para abrir el stream tan sólo tienen que indicarle al reproductor la url del servidor icecast (y el password si no teneis si es un stream protegido).

## Notas importantes

1. Si estáis pensando en montarlo para unas conferencias, es muy interesante ofrecer tanto el stream de vídeo (audio i vídeo) y por separado un stream de audio. Ya he comentado que para el stream de audio los parámetros son los mismos y que podéis usar ices2.
2. La fuente del stream y el servidor icecast pueden estar en la misma máquina y pueden no estarlo. Eso os puede venir bien si allí donde tenéis la fuente de datos, no hay una conexión con suficiente subida para soportar muchos clientes. En ese caso, podéis montar el servidor icecast en un lugar donde sí haya suficiente ancho de banda y hacerle llegar la señal del stream por una conexión lenta. Un stream de vídeo de calidad modesta puede subir fácilmente a los 25 Kb/s. Para cada cliente nuevo que tengáis necesitareis 25Kb más de subida.
3. Para que a la gente se le abra el reproductor cuando clique la url del stream en vuestra web, tenéis que crear un fichero de texto con la url del stream y con extensión (el fichero) `.m3u`. En la web ponéis un enlace al fichero y listos.
4. No olvidéis enviar el audio en mono, no en estéreo.
5. Montadlo un número de días mayor o igual a 2 antes de cuando lo necesitéis. Hacerlo todo con prisas puede resultar un poco estresante ([comprobado](#)<sup>(11)</sup>)

Si te has perdido, pregunta:

---

### Lista de enlaces de este artículo:

1. <http://breu.bulma.net/?l6192>
  2. <http://www.icecast.org/svn.php>
  3. <http://www.icecast.org/download.php>
  4. <http://breu.bulma.net/?l6193>
  5. <http://www.icecast.org/3rdparty.php>
  6. <http://www.musicpd.org/info.shtml>
  7. <http://mpd.wikia.com/wiki/Configuration>
  8. <http://freej.dyne.org/>
  9. <http://forum.videolan.org/viewtopic.php?t=14476>
  10. <http://wiki.v2v.cc/cgi-bin/trac.cgi/wiki/OggTheoraPlayer>
  11. [http://mnm.uib.es/gallery2/v/Jornades\\_Bulma/Dia+0/dsc00971.jpg.html](http://mnm.uib.es/gallery2/v/Jornades_Bulma/Dia+0/dsc00971.jpg.html)
- 

E-mail del autor: [lumen\\_ARROBA\\_bulma.net](mailto:lumen_ARROBA_bulma.net)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2290>