HAZTE UNA RADIO EN CASA
Mini manual de instrucciones

- **Bases Generales** - Introducción sobre el concepto de webradio
- **SoftWare** - Breve enumeración genérica de software utilizado para realizar una radio vía web
- **IceCast** - El servidor

LAS BASES

Realizar una radio on line no es una tarea dificilísima, pero tampoco es muy fácil.
Es necesario comprender básicamente algunos conceptos y la instalación de algunos programas, y a continuación un poco de experimentación. Casi como cualquier otra cosa tech ;))
Podemos simplificar el esquema de una Radio On Line de este modo:
Fuente de música -> Encoder -> Streamer -> Servidor -> Cliente

Fuente de música
Cualquier fuente de música. Para hacerla más fácil, pensémosla como un mixer [mezclador] al cual están conectadas algunas cosas, desde una lectora cd a un mini disc, pasando por el micrófono.

Encoder
El sonido que llega desde la Fuente de música, tiene que ser transformado en formato digital para poder ser transmitido en algo que "pueda ir en la red". El Encoder, sustancialmente, hace esto.
El Encoder es una computadora a la cual está conectada el mezclador que citamos, y que recoge el sonido de la tarjeta/placa de audio y lo transforma en un archivo mp3 (un formato digital comprimido para el sonido).
Herramientas de software: ***MuSe***, darkice, liveice, xmms, winamp

Streamer
Una vez que se obtuvo el formato digital que corresponde al sonido de la Fuente (el mp3 que citábamos antes), el streamer se encarga de transformarlo en una serie de paquetes de datos y pasarlo al servidor.
Siendo digital, el sonido ahora no es algo continuado, sino algo más humilde, es decir, constituido por pequeñas partes (bytes) que pueden ser enviados "en red", pieza por pieza, y ser reconstruido en forma siempre idéntica a sí mismo.
Para esto, el formato mp3 (o el ogg.vorbis) nos ayuda bastante, dado que el formato mp3 es prácticamente una serie de datos en bruto, sin encabezados ni cierre, como sucede con otros formatos de archivos. Por lo tanto, también si se recibe solamente una parte de un archivo, esa pieza aún resulta "audible".
Y es esta propiedad lo que permite el stream, es decir, la reproducción continua del mp3 que envían a la red.
El encoder y el streamer son la computadora, que toma el sonido y lo transforma en algo que puede ser enviado a la red.
Herramientas de software: ***MuSe***, darkice, liveice, xmms plugin, winamp plugin

Servidor
Este es el nodo de distribución de la señal de la radio on line. En la práctica, el servidor recibe la señal de una fuente (la radio desde el streamer) y lo redistribuye (sirve precisamente a los que se lo demandan).

Obviamente es posible para un servidor servir a diferentes streams, es decir, dos radios pueden utilizar el mismo servidor para hacer streaming, pero esto también limitará la banda a disposición tanto para uno como para el otro.
Más abajo se pueden ver las aclaraciones sobre los formatos, sobre la banda, sobre la calidad del sonido y otras cuestiones más cualitativas que relativas a la realización material de una radio on line.
La misma computadora puede hacer encoding y streaming, pero por lo general es otra computadora la que se encarga solamente de distribuir la señal, preferentemente conectada con banda ancha.
Herramientas de software: ***IceCast***, shoutcast

Cliente
Finalmente alguien escuchará la radio, ¿o no?
Bien, esto será el cliente, el oyente.
Herramientas de software: Xmms, mpg123, mp3blaster, winamp

SOFTWARE - INFORMACIÓN GENERAL

MuSe
Es un software desarrollado por Jaromil y muchas otras personas del grupo de operadores italianos y de otros países. Fue ideado sustancialmente como un **software multiusos para realizar una radio** en una sola computadora, aparte de un servidor. Permite mezclar piezas de sonido, voces y otras entradas de audio, guardar en el disco el mp3 que se está transmitiendo, conectarse a un servidor para hacer stream, y mucho más. Está todavía en fase de desarrollo y por lo cual, algunas veces tiene problemas, pero seguramente es el más simpático e interesante software para hacer streaming disponible.
http://muse.dyne.org

Darkice
Es un software bastante simple para hacer stream. Toma el sonido que llega de la tarjeta/placa de audio y lo transmite a un servidor. Permite hacer pocas cosas más, aparte de la regulación del volumen de la computadora para ajustar el volumen del stream.
http://darkice.sourceforge.net

Liveice
Software distribuido por el grupo de icecast y que tiene las mismas funciones que darkice. Es un poco más difícil de configurar.
http://star.arm.ac.uk/~spm/software/liveice.html

Xmms
Software similar al Winamp pero para Linux, que prácticamente todos conocen. Es un programa para reproducir música en diversos formatos. Es posible agregar un plugin que permite conectarlo directamente con liveice para hacer streaming de la música que se está escuchando.

Winamp
Software windows típico para escuchar piezas musicales en diversos formatos. Es posible agregar un plugin, con las adecuadas búsquedas on line, para liveice o para shoutcast que permite hacer streaming desde una computadora con windows.

mpg123 o mp3blaster
Reproductor de mp3 para linux en consola.

IceCast
Icecast es EL servidor para hacer streaming de audio. No hace nada más que tomar un stream y retransmitirlo a todos los que lo demandan. La versión 1 del servidor soporta solamente el stream mp3, mientras que la versión 2 soporta también el stream ogg-vorbis. Es simple para configurar y usar.
http://www.icecast.org

Shoutcast
Shoutcast es un servidor propietario para hacer streaming de audio. No lo conozco bien, y no pienso comenzar a conocerlo ahora.

ICECAST

.oO ICECAST Oo.

Este programa es un servidor, es decir, un programa que una vez ejecutado abre dos "puertos virtuales" que pone a la espera (los predeterminados son 8000/8001) para permitir a las personas conectarse y escucharnos. ;-)
Veamos cómo se hace.

.oO INSTALEMOS ICECAST Oo.

Descargar el código fuente de icecast, guardarlo (como usuario root) en /usr/src/ e descomprímanlo.

```
# mv icecast* /usr/src/
# cd /usr/src/
# tar -xvzf icecast-1.3.12.tar.gz
```

Esto creará la carpeta con el código fuente del programa.

```
# cd icecast-1.3.12
# less README
# less INSTALL
```

como está descripto en los archivos README e INSTALL, colocamos en la línea de comando

```
# ./configure
# make
# make install
```

En la documentación dice que si pasamos la opción --with-crypt a "./configure" podremos utilizar la posibilidad de una clave encriptada para garantizar mayor seguridad.
Otra posibilidad ofrecida para usar es --with-libwrap, para usar un tcp wrapper para seleccionar a quienes tienen acceso al servidor. Otro método (built in [incorporado] en esta versión) es el uso de las acl (access list [listados de acceso]) del servidor. Por ahora lo dejamos de lado.

.oO CONFIGUREMOSLO Oo.

Una vez instalado icecast en /usr/local/icecast, entramos en la dir con los principales archivos de configuración, la dir conf.

```
# cd /usr/local/icecast/
# cd conf/
```

Renombramos los archivos presentes en este directorio quitando el .dist qu está demás.

```
# mv icecast.conf.dist icecast.conf
```

Editamos icecast.conf, el archivo principal para nuestro objetivo.

```
# vi icecast.conf
```

Si tienen problemas con el inglés, sepan que las opciones principales que hay que tener en cuenta son:

```
port
server_name
```

Con estas dos opciones se decide sobre cuál puerto funcionará icecast, y tendrán que colocar como server_name el número del ip público o, si lo tienen, un dominio registrado.
El archivo de configuración está realmente bien explicado, no obstante, sepan que estos dos parámetros son los fundamentales para sobrevivir.
Demos una ojeada más detallada al archivo de configuración. ione.

```
location
rp_email
server_url
```

Estos parámetros son informaciones que icecast brindará a los diversos clientes que se conectarán, elijan lo que quieran decir, a mí me gusta Just west of Mars [el oeste de Marte] prescindiendo del lugar donde esté y, generalmente, los dejo tal como están, ya que son parámetros no fundamental para el funcionamiento.

max_clients, etc., etc., permite decidir cuántos pueden conectarse al servidor en calidad de clientes, fuentes y administradores.

```
clients --> quienes escuchan
sources --> quien/es proporcionan música
admins  --> quien/es administran el servidor
```

Como pueden observar, puede haber muchos oyentes (obvio), varias fuentes musicales (menos obvio) y varios administradores contemporáneamente.
Sobre los oyentes no agrego nada más, sobre las fuentes (MuSE, Multiple Streaming Engine es un ejemplo), se puede decir que es mejor tener dos, una principal y otra de soporte. Pero esto lo vamos a ver después, por ahora nos conformamos con un encoder.
Aquí se decide cuántas conexiones y de qué tipo (client, source, admin) están dispuestos a aceptar.
Dejemos de lado la sección Stream Meta Data, que no es importante en particular.

La opción

```
mount_fallback 1
```

es aquella que permite, en el caso de que tengan dos fuentes funcionando, hacer "saltar" a los clientes conectados a la fuente 1 sobre la fuente 2, en el caso de que la fuente 1 se caiga imprevistamente y viceversa... diría muy útil, siempre por el discurso de las caídas.

encoder_password <-- la clave que tendrá que proporcionar "MuSE" para poder conectarse como fuente al servidor. Los programas que envían música al icecast se llaman encoders, porque se encargan de tomar los que es reproducido en el escritorio, "codificarlo" y enviarlo a un servidor icecast que lo difundirá al mundo

admin_password <-- la clave para administrar en un primer nivel icecast directamente desde la consola (luego lo explicaré mejor)

oper_password <-- segundo nivel de administración del servidor desde la consola

La sección Directory servers ofrece la posibilidad de hacer aparecer sobre los servidores que están indicados allí, la presencia de nuestro servidor icecast para darle mayor visibilidad.
Otra opción importante a conocer es el alias.

alias petardo http://malasystem.com:8000

una línea como esta en el archivo de configuración significa que en la dirección http://mio.server.taz:8000/petardo se puede escuchar la música que proviene de http://malasystem.com:8000 directamente desde nuestro server. Esto no significa que si nadie está escuchando en ese mount point se consumirá ancho de banda para tenerlo activo, ya que se activa solamente bajo demanda de un cliente o forzándolo con el comando relay que veremos después. ;)
Un detalle a tener en cuenta porque podría hacernos perder tiempo: cuando introduzcamos un nuevo alias en la línea que pasamos en consola no debemos poner la / delante del nuevo mount point, sino que si queremos tener éxito al poner un alias tenemos que indicar en /petardo nuestro mount point... ahora lo sabemos. :)

Otra opción que hay que observar es

templatedir /usr/local/icecast/templates/

Este comando permite a icecast usar plantillas html para facilitar la administración vía web del servidor. el problema es que CUALQUIERA puede conectarse vía web y cambiar las opciones del servidor icecast, visto que no usamos (al menos por ahora) ni acl, ni tcp wrapper. Y esto diría que está mal.
Pongamos delante un simpático candado (#) agregando en el archivo también la variable

http_admin 0 <-- configurada en uno (en default sin estar en el archivo de configuración) es posible entrar en icecast vía web, configurada en cero, no.

y terminamos el juego :)

¿Cómo hago para cambiar las opciones "al vuelo"?

.oO ¡Y AHORA USEMOSLO! Oo.

Cuando ejecutamos icecast sin ningún parámetro, queda en primer plano la terminal con una consola "admin" abierta y lista para usar. Si se lanza con la opción –b, la misma queda en background [fondo] y la única manera para tener una consola sobre la cual operar es hacer un telnet mi.servidor.taz 8000 (si ese es nuestro puerto con icecast en espera).

telnet mio.server.taz 8000

```
Trying 0.0.0.0              <-- el simpático output [salida] del telnet
Connected to 0.0.0.0        <-- ¡lo mismo!
Escape character is '^]'    <-- ctrl + ] abre la consola de telnet.. :)
ADMIN password              <-- AQUÍ ESCRIBIMOS NUESTRA CLAVE ADMIN y pulsamos Enter
                            <-- se envía dos veces
ok                          <-- le ha gustado la clave
->                          <-- consola ICECAST!! :)
```

demos inmediatamente un bonito comando:

```
-> help
```

TRAMM! todos los comandos de consola listos para ser aprendidos.
por cada uno, se ofrece una pequeña descripción, para cada uno hay una posibilidad de aclaraciones digitando:

```
-> help nombre_comando_que_quiero_aprender
```

apenas logueados echamos un vistazo veloz a la situación:

```
-> list
```

Nadie está conectado todavía, ni fuentes, ni clientes, solamente hay un admin: nosotros.
Poniendo la clave de oper podemos acceder a los comandos alias, set y otros.
¡Convirtámonos en oper!

```
> oper password
You are now an icecast operator
```

¿Con qué configuración funciona icecast?

```
> set
Current settings
       encoder_password:     hackme
        client_password:     hackme
         admin_password:     hackme
          oper_password:     hackme
             touch_freq:5
        client_timeout:      30
           max_clients:      90
  max_clients_per_source:    90
           max_sources:      5
            max_admins:      3
               reverse:  1
              location:  Just west of Mars
              rp_email:  kirk@enterprise.space
     transparent_proxy:      0
             stats_log:  stats.log
         statshtml_log:      stats.html
            stats_time: 60
            acl_policy:  1
              throttle:  10.000000
           kick_relays: 10
          kick_clients: 0
           status_time:      120
      logfiledebuglevel:     0
```

```
      consoledebuglevel:      0
            server_url: http://rmi.homeip.net
         use_meta_data:      0
         streamurllock:      0
            streamurl: http://rmi.homeip.net
    streamtitletemplate:      %s
          nametemplate:      %s
          desctemplate:      %s
             logfile:    icecast.log
           accessfile:   access.log
            usagefile:   usage.log
            staticdir:    /usr/share/icecast/static/prova.mp3
  default_source_options:    istphecyocdrlumnagbUMDRWCT
       mount_fallback:      1
      force_servername:      0
         resolv_type:      1
          http_admin:      1
    relay_reconnect_max:    -1
   relay_reconnect_time:    90
         sleep_ratio: 0.100000
End of settings
```

Si ya hay un encoder instalado (en breve una sección INSTALEMOS MuSE), ejecútenlo brindando las informaciones para conectarse.
introducimos el comando sources en consola:

```
[Id: 1] [Sock: 12] [Time of connect: 17/Apr/2002:12:03:09] [IP: 192.168.x.x] [Host:
vega.mala.taz]
[State: 1] [Type: encoder] [Proto: x-audiocast] [Clients: 0] [Dumpfile/fd: (null)/-1]
[Priority: 0]
[Song Title: ] [Song URL: http://www.spaziopetardo.it] [Stream Message: (null)] [Song
Length:
-1 bytes] [Stream Name: radiomozzarellainternational] [Stream Genre: icecast]
[Stream Bitrate: 24000] [Stream URL: http://muse.dyne.org] [Mountpoint: /petardo]
[Description: streaming with MuSE] [MBytes read: 0] [MBytes written: 0] [Client
connections: 0]
[Connected for: 4 minutes and 20 seconds]
End of source listing (1 listed)
```

Fundamentales los campos:

[Id: num]	con el número indicato en este campo, identificamos la fuente (funciona del mismo modo para los clientes, los admins y los operators)
[Type: encoder,pulled relay ..]	el tipo de fuente conectada. puede ser un encoder u otro servidor icecast peticionado con el comando relay, por ejemplo.
[Clients: num]	¿cuantos oyentes hay en este momento?
[Stream Bitrate: num]	¿a cuánto está haciendo streaming la fuente? aconsejo ajustar el stream a 24 Kbit/s para permitir una mayor accesibilidad a quienes tengan conexiones lentas para escuchar :)
[Mountpoint: name]	el mount point sobre el cual esta fuente está

funcionando. cuantas más fuentes, más mount points, en modo que podamos tener más posibilidades de escuchar para ofrecer o simplemente habilitando la opción mount_fallback para evitar el riesgo que cuando se caiga una fuente, se caigan también todos nuestros oyentes, los cuales en esta forma saltan al segundo, o bien permitirnos el lujo de desplazarnos de una fuente a otra a nuestros clientes según la necesidad ("y ahora, línea al estudio en Roma." :)

ahora veremos cuando llega un cliente para escucharnos:

```
-> listeners
Listing listeners
[Host: 213.140.9.154] [Mountpoint /petardo] [Id: 488] [Connected for: 1 hours, 13
minutes and 43 seconds] [Source Id: 491] [Bytes written: 13162855] [Errors: 0] [User
agent: icecast/1.3.12] [Type: client]
End of listener listing (1 listed)
```

los campos principales:

[HOST: IP or NAME]	quien sea, ¿o el mítico primer oyente?
[Mountpoint name]	¿qué estás escuchando?
[Source Id: num]	¿quién y qué fuente está asociada a ese mount point?
[Type: client,puller]	es un simple cliente o un servidor que me está "relayando", es decir, que me ha tomado como su fuente?

Resumiendo:

```
-> list
Listing connections:
[Id: 490] [Host: localhost] [Type: admin] [Connected for: 36 minutes]
[Id: 491] [Host: vega.mala.taz] [Type: source] [Connected for: 1 hours, 17 minutes and
55 seconds]
[Id: 488] [Host: 213.140.9.154] [Type: client] [Connected for: 34 minutes and 27
seconds]
End of list listing (3 listed)
```

en este momento hay un admin en consola. Si ejecutamos icecast sin la opción -b y luego telneteamos el servidor tenemos como resultado dos, pero no es un intruso ;), es solamente la consola que queda abierta cuando ejecutamos icecast y nos agregamos con telnet.
Como verán los [ID: num] identifican todas las tipologías de "presencias" en el servidor.
Cada tanto, el servidor por sí solo ofrece un resumen veloz de la situación:

```
-> [17/Apr/2002:13:27:46] [Bandwidth: 0.016667MB/s] [Sources: 1] [Clients: 3]
[Admins: 1] [Uptime: 2 hours, 10 minutes]
```

Juguemos un poco con nuestro servidor. :)
ahora tenemos una fuente (la nuestra), ¿y si quisiéramos tener otra? ya desde la configuración de lanzamiento podremos tener varios alias preconfigurados, sin embargo estos no estarán en uso hasta que no sean peticionados (no consumen banda :), un modo

para activarlos es que un cliente los demande (poniendo el mount point del alias en la dirección del servidor), otro modo es demandar el servidor que querramos volver a ejecutar con el comando "relay". O bien que otro encoder se conecte sobre otro mount point.

-> relay rmi.homeip.net:30000
relay added

dentro de un rato tendremos dos fuentes.
con las dos fuentes pueden desplazar desde una a otra a todos los clientes

-> select -a Id source1 Id source2

y desplazamos a todos los oyentes desde la primera fuente hacia la segunda.
o bien

-> select Id client Id source2

y desplazo a un solo oyente a una nueva fuente.
Hey!, y en este punto entra tu curiosidad yganas de jugar, ¿o no?

Los comandos necesarios para sobrevivir son:

list	<-- situación general (cuántos admin, sources, listeners)
sources	<-- quien está haciendo streaming (el encoder)
listeners	<-- quién y qué está escuchando
select {-a|idC idS}	<-- desplaza a todos los clientes de la fuente 1 a la fuente 2, o bien desplaza el cliente [idC] desde la fuente 1 a la fuente 2 [idS]
alias {list|add|del}	<-- agrega una fuente (no un encoder, sino otro servidor icecast que querramos sostener)
relay {push|pull}i	<-- abre o cierra la conexión con un servidor con el que querramos tener un alias
oper {password}	<-- para un cierto grupo de comandos es necesario brindar una segunda clave
set	<-- ¿con qué configuración estoy haciendo las cosas?