

FIRtro: guía de instalación

Contenido

- Control de cambios.....	3
1 - S.O. Linux.....	4
1.1 - Instalación nueva.....	4
1.2 - Usuarios Linux.....	4
2 - Actualizar el sistema de paquetes.....	5
3 - Paquetes de software de utilidad general.....	5
4 - Paquetes de software necesarios para FIRtro.....	5
4.1 - OpenSSH.....	5
4.2 - Python 2.7.....	5
4.3 - Jack.....	5
4.4 - Brutefir.....	6
4.5 - Ecasound.....	7
4.6 - Para máquinas con eXcritorio (Pulseaudio).....	7
4.7 - Otro software.....	7
4.8 - Página web de control remoto.....	8
4.9 - Servidor de música MPD (Music Player Daemon).....	8
4.10 - El reproductor multimedia Mplayer.....	10
4.11 - Mopidy.....	10
4.12 - Cliente Spotify de escritorio.....	10
4.13 - Carpetas de red SMB (samba).....	11
5 - Software FIRtro.....	11
5.1 - Descarga del ZIP desde GitHub.....	11
5.2 - Usuarios avanzados: clon GitHub.....	11
5.3 - Estructura del software FIRtro.....	12
5.4 - Permisos de ejecución.....	12
5.5 - Configuración inicial.....	12
6 - Arranque de FIRtro.....	12
7 - Anexo. Servicio para el arranque (daemon) /etc/init.d/firtro.....	12

Control de cambios

Fecha	Comentarios
2016-nov	Version inicial

1 S.O. Linux

Se recomienda la distro estable de **Debian**, u otra derivada de esta. Para una Raspberry Pi usaremos Raspbian.

FIRtro es un sistema en principio “black box”, una caja dedicada al filtrado de altavoces y servidor de música controlable por la LAN del usuario..

FIRtro no requiere escritorio X windows (servidor gráfico), pero si se desea tenerlo incluso una RPI moderna lo soporta sin penalizar el consumo de recursos.

En todo caso es posible dejar una máquina “headless” (sin escritorio gráfico) y usar algunas aplicaciones gráficas, como por ejemplo `qjackctl` y `python-matplotlib` sin necesidad de equipar la máquina FIRtro con un servidor gráfico de escritorio (GDM en la jerga). En este caso las aplicaciones gráficas podrán “proyectarse” en una máquina remota que se conecte a FIRtro, ver OpenSSH más abajo.

1.1 Instalación nueva

Se recomienda instalar el S.O. con un usuario personal. Por ejemplo Raspbian de Raspberry Pi se entrega con el usuario pi.

Una vez instalado el S.O., añadiremos un usuario llamado “firtro”, integrado en los grupos adecuados del S.O.:

```
sudo adduser firtro
sudo usermod -a -G cdrom,audio,video,plugdev firtro
sudo visudo
    firtro ALL=(ALL) ALL          -> Pide password
    firtro ALL=(ALL) NOPASSWD:ALL -> No pide password
sudo nano /etc/hosts
    127.0.0.1 firtro
sudo nano /etc/hostname
    firtro
```

Añadimos el directorio de scripts de FIRtro al PATH del usuario firtro:

```
echo export "PATH=$PATH:$HOME/bin" >> /home/firtro/.bashrc
```

Otras variables de entorno:

```
echo "# resaltado en color de codigo fuente con less" >> .bashrc
echo "# less source highlighting" >> .bashrc
echo "export LESSOPEN='| /usr/share/source-highlight/src-hilite-lesspipe.sh"
%s'" >> .bashrc
echo "export LESS=' -R '" >> .bashrc

echo "# esto evita la aparicion de archivos .pyc .pyo" >> .bashrc
echo "export PYTHONDONTWRITEBYTECODE=1" >> .bashrc
```

Ajuste para que el editor **nano** use 4 espacios al tabular (standard de Python):

```
echo "set tabsize 4" >> .nanorc
echo "set tabstospaces" >> .nanorc
```

1.2 Usuarios Linux

Podemos considerar instalar FIRtro en una máquina Linux ya existente. En el apartado instalación del **software de FIRtro** se indica cómo.

Añadimos el directorio de scripts de FIRtro al PATH del usuario existente:

```
echo "export PATH=$PATH:/home/firtro/bin" >> /home/miusurioexistente/.bashrc
```

2 Actualizar el sistema de paquetes

Antes de empezar a instalar software, incluso en una instalación reciente del S.O., deberemos poner al día la base de datos de paquetes software:

```
sudo apt-get update
```

3 Paquetes de software de utilidad general

- Un monitor del sistema: `htop`
- Un gestor de archivos: `mc`
- La utilidad de descompresión `unzip`.
- El gestor de copias de seguridad `rsync`
- El entorno interactivo Python `ipython`
- La utilidad de automontaje de unidades USB `usbmount`

```
sudo apt-get install htop mc unzip rsync ipython usbmount
```

4 Paquetes de software necesarios para FIRtro

4.1 OpenSSH

```
sudo apt-get install openssh-server # habitualmente ya incluido
```

Probemos que accedemos desde nuestro PC de trabajo, por ejemplo un portátil:

```
ssh -X myUser@FIRtroIPAddress
```

NOTA:

Conviene comprobar que cuando accedemos a FIRtro desde otra máquina, que es lo más usual, dicha máquina sea capaz de proporcionar un X server para que las aplicaciones gráficas de FIRtro puedan visualizarse en ella. Por ejemplo `qjackctl` o las gráficas de un ecualizador paramétrico.

Desde un portátil Linux basta conectarse a FIRtro con `ssh -X firtro@IP_FIRtro`

Desde un Mac OS X, previamente deberemos instalar XQuartz.

En entornos Windows se puede resolver configurando PuTTY para hacer X-Forwarding

4.2 Python 2.7

Viene de fábrica

4.3 Jack

Jack es el core de conectividad de flujos de audio en FIRtro.

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

(i) Al instalar JACK nos preguntará si debe activarse el “real time”, responderemos afirmativamente.

Para poder usar JACK con privilegios “real time audio” añadimos al usuario `firtro` (o el que usemos) al grupo audio:

sudo adduser firtro audio

NOTA para máquinas sin servidor X (sin escritorio):

Se prefiere JACK2 (jackd2) por la mejora de uso de su módulo de audio por la red. Pero JACK2 requiere que el perfil de la sesión desde la que se arranca jackd indique a DBUS que se dispone de un X display. En una máquina sin servidor X obtendremos un error al intentar arrancar Jack2:

```
Failed to connect to session bus for device reservation Unable to autolaunch a
dbus-daemon without a $DISPLAY for X11
```

En una máquina sin servidor X será necesario ajustar lo siguiente:

1.- Declarar variables de entorno del usuario firtro:

```
echo "export DBUS_SESSION_BUS_ADDRESS=unix:path=/run/dbus/system_bus_socket" >>
/home/firtro/.profile
```

2.- Editar: sudo nano etc/dbus-1/system.conf insertando en <busconfig>...</busconfig> lo siguiente:

```
<policy user="firtro">
<allow own="org.freedesktop.ReserveDevice1.Audio0"/>
<allow own="org.freedesktop.ReserveDevice1.Audio1"/>
<allow own="org.freedesktop.ReserveDevice1.Audio2"/>
<allow own="org.freedesktop.ReserveDevice1.Audio3"/>
</policy>
```

(nota: Audio0....Audio4 cubre hasta cuatro posibles tarjetas automáticamente descubiertas por el servicio DBUS)

3.- Reiniciar el servicio:

```
sudo service dbus restart
```

4.- Iniciar un nuevo login del usuario firtro y comprobar que jackd arranca:

```
firtro@firtro:~ $ jackd -dalsa &
```

```
firtro@firtro:~ $ killall jackd # para interrumpirlo
```

4.4 Brutefir

Brutefir es el convolver que posibilita el filtrado FIR para el corte de vías (xover), para DRC, el ajuste de niveles y delays y la ecualización básica de loudness y "room curves".

sudo apt-get install brutefir

(i) Brutefir es el proceso más exigente con el PC, se recomienda desactivar el swap de memoria a disco. Ver la guía del usuario: herramientas y troubleshooting.

```
nano /etc/rc.local
swapoff -a
```

Editamos **/home/firtro/.brutefir_defaults** incluyendo lo siguiente:

```
# modulo "cli" para puerto tct/ip para consultas y cambios al vuelo
# y modulo "eq" para el EQ del preamplificador de FIRtro
logic:
```

```
"cli" { port: 3000; },
```

```
"eq" {
  #debug_dump_filter: "/home/firtro/.brutefir/rendered-%d";
  {
    coeff: "c_eq0", "c_eq1";
    bands: 18.0, 20.0, 22.4, 25.0, 28.0, 32.0, 36.0, 40.0, 44.8, 50.0, 56.0,
63.0, 71.0, 80.0, 90.0, 100.0, 112.0, 126.0, 142.0, 160.0, 180.0, 200.0, 224.0,
250.0, 280.0, 320.0, 360.0, 400.0, 448.0, 500.0, 560.0, 630.0, 710.0, 800.0,
900.0, 1000.0, 1120.0, 1260.0, 1420.0, 1600.0, 1800.0, 2000.0, 2240.0, 2500.0,
```

```
2800.0, 3200.0, 3600.0, 4000.0, 4480.0, 5000.0, 5600.0, 6300.0, 7100.0, 8000.0,
9000.0, 10000.0, 11200.0, 12600.0, 14200.0, 16000.0, 18000.0, 20000.0, 22040.0;
};
```

4.5 Ecasound

Ecasound proporciona la facilidad de EQ paramétrico en FIRtro.

```
sudo apt-get install ecasound ecatoools python-ecasound
```

(i) En entornos Desktop p.ej. Gnome, el script `initfirtro.py` se ejecutará en un "non-login shell", tal que al intentar lanzar `ecasound` se produce un error: "TERM environment variable not set.", y `ecasound` no arranca. El error es capturable si redirigimos a un archivo los errores `stderr` de Ecasound. Se soluciona incluyendo lo siguiente en `~/ .bashrc` (`.bashrc` se toma en cuenta en un non login shell).

```
echo 'export TERM=xterm-256color' >> /home/firtro/.bashrc
```

Software accesorio para los scripts relacionados con Ecasound:

```
sudo apt-get install python-scipy python-matplotlib ladspa-sdk
```

Plugins de filtros paramétricos de Fons Adriaensen:

```
sudo apt-get install fil-plugins
```

4.6 Para máquinas con eXcritorio (Pulseaudio)

Pulseaudio es el estándar de gestión del audio en escritorios Linux.

El script de inicio de FIRtro se ocupa de que Pulseaudio se inhíba en el uso de nuestra tarjeta de audio que será tomada por Jack.

Lo siguiente nos servirá para que el audio que se manda a Pulseaudio, p.ej. un video de YouTube, llegue a JACK que es el core de FIRtro:

```
sudo apt-get install pulseaudio-module-jack
```

4.7 Otro software

Sox es la navaja suiza del audio:

```
sudo apt-get install sox libsox-dev libsoxr-dev libsoxr-lsr0
```

Complemento para lectura resaltada de códigos de programa (como Python):

```
sudo apt-get install source-highlight
```

Complemento para los programas que usan Jack:

```
sudo apt-get install libjack-jackd2-dev (o libjack-dev si usamos JACK1)
```

Complementos de Python, y el compilador `gcc` de apoyo:

```
sudo apt-get install python-numpy python-dev gcc
```

Y la librería Python para interactuar con Jack:

Opc.1)

```
sudo apt-get install python-pip # pip es el gestor de paquetes de python
```

```
sudo pip install py-jack
```

Opc.2)

```
wget http://sourceforge.net/projects/py-jack/files/py-jack/0.5.2/pyjack-0.5.2.tar.gz
```

Descomprimir y ver el README para su instalación (sudo)

4.8 Página web de control remoto

Instalar el servidor web apache2:

```
sudo apt-get install apache2
```

Instalar el módulo php para apache:

```
sudo apt-get install libapache2-mod-php
```

Si lo anterior no funciona, buscar la versión más alta de la distribución que será la que instalaremos, ejemplo:

```
aptitude search libapache2-mod-php
```

```
p  libapache2-mod-php          - server-side, HTML-embedded ...
p  libapache2-mod-php:i386     -
p  libapache2-mod-php7.0      - server-side, HTML-embedded ... <<<<----ESTA
p  libapache2-mod-php7.0:i386
```

Preparar el archivo que cargará el sitio web:

```
sudo nano /etc/apache2/sites-available/FIRtro.conf
```

```
DocumentRoot /home/firtro/www
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory /home/firtro/www/>
    # Apache ver < 2.4
    #Options Indexes FollowSymLinks MultiViews All
    # Apache ver >=2.4
    Options All Indexes FollowSymLinks MultiViews
    # Apache version < 2.4
    #AllowOverride None
    # Apache version < 2.4
    #Order allow,deny
    # Apache version >=2.4
    Require all granted
    allow from all
</Directory>
```

Activar el sitio web:

```
sudo a2ensite FIRtro.conf
```

```
sudo a2dissite 000-default.conf
```

```
sudo service apache2 reload
```

4.9 Servidor de música MPD (Music Player Daemon)

En FIRtro, el servicio MPD se configura para ser ejecutado “per user”.

Algunas distribuciones Linux instalarán el servicio (demonio) MPD en modo “wide system”, es decir al arranque de la máquina y a cargo de un usuario llamado `mpd`. FIRtro se ha diseñado para que el servicio MPD corra a cargo del usuario `firtro`.

Instalamos MPD y desactivamos el servicio “system wide” que viene de fábrica:

```
sudo apt-get install mpd mpc
```

```
# Desactivamos el servicio “system wide” al arranque:
```

```
sudo service mpd stop
```

```
sudo update-rc.d mpd remove
```

```
# En sistemas modernos con systemd necesitaremos también desactivar el socket  
# de autoarranque de mpd bajo demanda de un posible cliente (p.ej gmpc)
```



```
# ver con sudo systemctl list-unit-files | grep mpd
sudo systemctl stop mpd.service
sudo systemctl disable mpd.service
sudo systemctl stop mpd.socket
sudo systemctl disable mpd.socket
```

La configuración del servicio MPD está definida en el archivo **/home/firtro/.mpdconf** que viene con el software de FIRtro. Aquí se declara principalmente:

La localización de la biblioteca musical:

```
music_directory      "/home/firtro/music"
```

La localización de los archivos de "playlist":

```
playlist_directory   "/home/firtro/playlists"
```

La salida del audio desde aquí al servidor de audio (ejemplo):

```
audio_output {
    type                "jack"
    name                "mpd_jack"
    client_name         "mpd_jack"
    format              "*:*:2"
    #destination_ports  "brutefir:input-0,brutefir:input-1"
    destination_ports   "ecasound:in_1,ecasound:in_2"
    enabled             "yes"
    always_on           "yes"
}
```

Si ya dispusiéramos de otras carpetas con archivos musicales, deberemos enlazarlas a **/home/firtro/music** para que MPD las incorpore en su base de datos, ejemplo:

```
ln -s /path/to/MyMusic /home/firtro/music
```

Si nuestra biblioteca musical estuviera localizada en un disco de red, deberemos tener el recurso montado (ver más abajo Carpetas de Red) y enlazarlo también, ejemplo:

```
ln -s /mnt/nas_music /home/firtro/music/nas_music
```

Si optamos por alguna de las soluciones mencionadas que en definitiva introducen un symlink dentro del **music_directory** de mpd, deberemos habilitar el seguimiento de symlinks externos en **.mpdconf**:

```
follow_outside_symlinks "yes"
```

```
# Preparamos un directorio para uso interno del
# servidor MPD que será iniciado por el usuario firtro:
mkdir -p /home/firtro/.mpd
```

Preparamos el directorio de playlists de MPD:

Configuración habitual:

```
mkdir /home/firtro/.mpd/playlists
```

Configuración alternativa para usuarios avanzados:

También podemos compartir las playlists entre varios servidores MPD que pudiéramos tener en otros PCs (al margen de FIRtro), enlazando a un directorio de "playlists" alojado en nuestro disco de red de música. Para esto usaremos otro enlace:

```
ln -s /home/firtro/.mpd/playlists /mnt/nas_music/mpd_playlists
```

Si queremos que MPD actualice la biblioteca musical ahora:

```
# arrancamos el servidor (advertirá que la salida de JACK no está disponible)
mpd
```

```
# actualizamos la BD
mpc update
```

NOTA posible error confuso:

Se ha detectado que al iniciar MPD pudiera aparecer un error como:

```
bind to '0.0.0.0:6600' failed: Address already in use
```

Sin embargo, el problema puede ser debido a que el directorio decarado en `playlist_directory` en `.mpdconf` no está disponible. Una vez corregido este problema, desaparece el mensaje de error mencionado. Entendemos que es un bug de MPD.

NOTA para otros usuarios en un sistema de escritorio:

Es posible que un usuario existente distinto de `firtro` pueda ejecutar `FIRtro` en su sesión, y también MPD.

Para ello el usuario debe disponer en su “home” de una copia del archivo `.mpdconf` de `firtro`, modificada apuntando a su propio directorio:

```
cp /home/firtro/.mpdconf /home/MiUsuario/.mpdconf
# modificamos el archivo reapuntando a nuestro 'home'
sed -i -e 's/\\firtro/\\MyUsuario/g' /home/MiUsuario/.mpdconf
# preparamos nuestro directorio para ser usado por mpd
mkdir -p /home/MyUsuario/.mpd
```

4.10 El reproductor multimedia Mplayer

Mplayer permite reproducir discos usando el lector CD de la máquina. También permite la recepción de canales DVB-T (TDT) equipando la máquina con una sintonizadora compatible.

(i) NOTA: También es posible reproducir CDs y Unidades USB automáticamente con el servidor de audio MPD.

Ver la [Guía de instalación mplayer CD DVB](#) y la [Guía del Usuario](#) para más información.

4.11 Mopidy

Es de interés para conectar con el servicio Spotify de internet, Mopidy es manejable con cualquier cliente compatible con MPD. Solo es utilizable en el caso de disponer de cuenta premium en Spotify.

PENDIENTE DESARROLLAR

4.12 Cliente Spotify de escritorio

Ver la información en:

```
https://www.spotify.com/es/download/other/
https://www.spotify.com/es/download/linux/
```

Que se reduce a:

```
# 1. Add the Spotify repository signing key to be able to verify downloaded
packages
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
BBEBDCB318AD50EC6865090613B00F1FD2C19886
# 2. Add the Spotify repository
echo deb http://repository.spotify.com stable non-free | sudo tee
/etc/apt/sources.list.d/spotify.list
# 3. Update list of available packages
sudo apt-get update
# 4. Install Spotify
sudo apt-get install spotify-client
```

4.13 Carpetas de red SMB (samba)

```
sudo apt-get install cifs-utils
```

```
sudo nano /etc/fstab
```

```
//192.168.1.90/media/music /mnt/nas_music cifs
credentials=/root/.smbpasswd,sec=ntlm 0 0
```

Las credenciales las guardamos en /root

```
sudo su
cd
echo username=mywindowsusername > .smbpasswd    (the NAS user)
echo password=mywindowpassword >> .smbpasswd
chmod 600 .smbpasswd
exit
sudo mkdir /mnt/nas_music
sudo mount -a
```

Ahora tendremos acceso al disco en red.

5 Software FIRtro

El software FIRtro reside en GitHub, que es una plataforma standard para repositorio y desarrollo colaborativo de software.

El software FIRtro puede descargarse por dos métodos:

5.1 Descarga del ZIP desde GitHub

```
cd /home/firtro
```

Bajamos y descomprimos el .zip de la versión actual de FIRtro:

```
wget https://github.com/AudioHumLab/FIRtro/archive/master.zip
```

```
unzip master.zip
```

Se habrá creado el directorio FIRtro-master conteniendo todo lo necesario. Lo copiamos en el directorio firtro:

```
cp FIRtro-master/* /home/firtro/
```

5.2 Usuarios avanzados: clon GitHub

El sistema git está disponible en Linux, si fuera necesario instalarlo:

```
sudo apt-get install git
```

Clonamos el sistema FIRtro:

```
git clone https://github.com/AudioHumLab/FIRtro.git
```

5.3 Estructura del software FIRtro

Se organiza en carpetas:

— audio	archivos de configuración, de estado, plantillas del sistema.
— bin	scripts python del sistema y del usuario
— doc	documentación
— lspk	archivos de filtros de los altavoces
— www	sitio web de la página de control

5.4 Permisos de ejecución

NOTA: si se usa Windows en el proceso de descompresión del archivo zip se perderán los permisos.

Los scripts contenidos en la carpeta ~/bin deben aparecer con permisos de ejecución:

```
ls -l /home/firtro/bin/*  
  
-rwxrwxr-x 1 firtro firtro 497 oct 16 10:31 basepaths.py  
...  
... etc ...
```

También deben aparecer con permisos de ejecución todos los archivos contenidos en ~/www (la página web).

En el caso que fuera necesario ajustarlos:

```
chmod 775 /home/firtro/bin/*  
  
chmod -R 775 /home/firtro/www
```

Y si FIRtro ha sido instalado en una máquina ya existente con otros usuarios que quieran ejecutar FIRtro, ajustaremos los permisos del fichero de estado:

```
chmod 666 /home/firtro/audio/status
```

5.5 Configuración inicial

El sistema se suministra configurado con JACK trabajando sobre el backend “dummy” que es ficticio, es decir no se usa una tarjeta de sonido real. Ver la guía del usuario para configurar Jack sobre el backend “alsa” para usar la tarjeta ALSA física de nuestra elección.

6 Arranque de FIRtro

El arranque de FIRtro se detalla en la Guía del usuario.

7 Anexo. Servicio para el arranque (daemon) /etc/init.d/firtro

```
#!/bin/sh  
### BEGIN INIT INFO  
# Provides:          FIRtro  
# Required-Start:    $all  
# Required-Stop:     $all  
# Default-Start:     2 3 4 5  
# Default-Stop:      0 1 6  
# Short-Description: Start FIRtro at boot time  
# Description:       MPD-Jack-Brutefir  
### END INIT INFO  
  
# /etc/init.d/firtro  
# Some things that run always  
touch /var/lock/firtro  
  
# Carry out specific functions when asked to by the system  
case "$1" in  
    start)  
        tput clear  
        #Cambiamos permisos para poder acceder a la pantalla principal desde el  
usuario firtro  
        chown firtro /dev/tty1  
        #Para poder monitorizar la salida con un tail desde el usuario firtro  
        chown firtro /dev/vcs1  
        echo "Iniciando script FIRtro..."  
        su -c "/home/firtro/bin/initfirtro.py all" -l firtro  
        ;;  
    stop)  
        echo "Deteniendo script FIRtro..."  
        su -c "python /home/firtro/bin/stopfirtro.py all" -l firtro  
        #Desmontamos share de red  
        /usr/local/bin/umount_cifs  
        ;;  
    *)  
        ;;  
esac
```

```
*)
    echo "Uso: /etc/init.d/firtro {start|stop}"
    exit 1
;;
esac
exit 0
```