

# Instalación de Servidores de Video en Debian

## Sumario

- 1 Introducción a los servicios de video
- 2 Formatos de video
  - ◆ 2.1 ¿Qué características distinguimos en un video?
  - ◆ 2.2 Tipos de formatos
  - ◆ 2.3 Compresión de video
- 3 Instalación de Darwin Streaming Server
  - ◆ 3.1 Pasos para la instalación de Darwin Streaming Server en Debian
  - ◆ 3.2 Preparar los videos para Darwin Streaming Server
    - ◇ 3.2.1 Hacer las playlists de videos en Darwin
  - ◆ 3.3 Conectarse con un cliente a Darwin Streaming Server
  - ◆ 3.4 Manual en PDF del Administrador de Darwin Streaming Server
- 4 Instalación de RED5 Streaming Server
  - ◆ 4.1 Pasos para una instalación rápida de Red5 1.0.1
  - ◆ 4.2 Instalación manual de RED5 versión 1.0.1
  - ◆ 4.3 Gestionar el servidor
  - ◆ 4.4 Instalación de videos y aplicaciones demo en el servidor
  - ◆ 4.5 Conectarnos con un cliente al servidor RED5
  - ◆ 4.6 Ejemplo de página web con reproductor JWPLAYER y servidor RED5 con RTMP
  - ◆ 4.7 Creación de Playlist en el lado del servidor con RED5

## Introducción a los servicios de video

Vamos a ver en esta sección cómo instalar un servidor de video para poder hacer streaming.

Al igual que comentábamos en los servidores de audio, tenemos dos métodos o formas de proporcionar un contenido de video a nuestros clientes:

- Un primer método consiste en colgar el video en un servidor web.
- Un segundo método consistiría en proporcionar ese video a través de un servidor de video.

Con el primer método, el cliente tendrá que esperar a que el video se descargue completamente para poder visualizarlo. Tiene la ventaja de la facilidad de su distribución, ya que no es necesario instalar componentes adicionales, salvo el servidor web.

Con el segundo método, el cliente podrá moverse a cualquier parte del video sin tener que esperar la descarga total del mismo, ya que el propio servidor de streaming se encarga de proporcionar la parte del video solicitada en cada instante. Tiene el inconveniente de tener que instalar el nuevo servicio, pero las ventajas de uso con respecto al método tradicional compensan con creces ese handicap.

## Formatos de video

Un video es una secuencia de imágenes que proporcionan la sensación de movimiento. Cada secuencia se denomina **fotograma** o **frame**.

### ¿Qué características distinguimos en un video?

- **Framerate (fotogramas por segundo)**: nos indica cuantos fotogramas por segundo se están reproduciendo en el video. Se utilizan las siglas fps. Cuanto mayor es el número de fotogramas por segundo, mayor será la calidad del video. También se puede expresar el valor de los fotogramas en Hz(Hertzios), siendo el estándar para Europa (PAL 25 Hz) y para Estados Unidos NTSC (29,97Hz).
- **Resolución**: se expresa en pixels, y nos indica el tamaño del video (ancho x alto). Por ejemplo los DVD tienen una resolución de 720x480. Los videos de alta definición **720p** (1280x720) y los **fullhd 1080p** (1920x1080).
- **Bitrate**: este mismo término lo empleamos en los servidores de audio, y nos indica la cantidad de bits por segundo que se están transmitiendo por segundo.

## Tipos de formatos

El formato de video indica el modo en el que los vídeos guardan los datos de un archivo de vídeo con el fin de que puedan ser interpretados por el ordenador. Normalmente, un vídeo es una colección de imágenes acompañada de sonido; la información de uno y otro tipo se suele grabar en pistas separadas que luego se coordinan para su ejecución simultánea.

Para transformar la información analógica de las imágenes en digital se usan los **códec** (acrónimo de codificador/decodificador). En muchos casos estas utilidades analizan los fotogramas y emplean algoritmos para comprimir sus datos. La compresión puede ser temporal, en la que se analiza un fotograma y se guarda la diferencia entre éste y el que le precede, o espacial, en la que se eliminan los datos de los píxeles que no cambian en cada fotograma.

Existen **múltiples formatos de vídeo** de gran implantación:

- QuickTime Movie (MOV).
- AVI
- El estándar MPEG.
- Flash Video.
- MKV (Matroska Video)
- Real Video.
- OGG u OGV.

**El formato QuickTime Movie (MOV)**, creado por Apple, es multiplataforma y en sus versiones más recientes permite interactuar con películas en 3D y realidad virtual. El AVI (Audio Video Interleaved, audio vídeo intercalado) es un formato también multiplataforma. Tanto \*.avi como \*.mov son **contenedores de audio y vídeo** con lo que son formatos de archivo. A este archivo habría que especificarle el tipo de video o audio que está conteniendo y que puede ser sin compresión o con la compresión soportada por dicho fichero como pueden ser para los \*.avi el divx, Dv-pal, etc y para \*.mov el sorenson, H264, etc.

**El formato correspondiente al estándar MPEG** (Moving Pictures Experts Group) produce una compresión de los datos con una pequeña pérdida de la calidad; desde su creación, se ha definido el MPEG-1, utilizado en CD-ROM y Vídeo CD, el MPEG-2, usado en los DVD de Vídeo y la televisión digital, y el MPEG-4, que se emplea para transmitir vídeo e imágenes en ancho de banda reducido; es un formato adecuado para distribuir multimedia en la Web. El formato MPEG4 es la base de actuales formatos como el divx xvid o el H264 siendo este último (H264) un codec tan potente que soporta vídeos de gran formato y calidad en alta definición con anchos de banda muy reducidos.

**El formato MKV (Matroska Video)**, como os imagináis debe su nombre a las famosas muñecas Matroska , ya que su filosofía se debe a ellas. Es de código abierto y es capaz de contener en un único archivo, video, sonido y subtítulos. Además permite la creación de menús, selección de idioma de audio y la elección de los subtítulos. Emplea la extensión ".MKV".

**El formato OGG u OGV** es el empleado y desarrollado por Xiph.org. Es de código abierto y su homólogo en audio lo hemos utilizado en el servidor de streaming Icecast de la unidad anterior.

El desarrollo de Internet ha propiciado formatos que permiten visualizar vídeos a través de la red, sin tener que descargar previamente el archivo completo; para esta finalidad, la empresa RealNetworks ha establecido RealVideo y Microsoft su formato de vídeo correspondiente al Windows Media Player (Reproductor de Windows Media). Éstos formatos tenían un gran futuro, pero hoy en día se están utilizando nuevos sistemas de streaming y por lo tanto están cada vez más en desuso.



Youtube es uno de los sistemas más importantes en la transmisión de video en streaming, y aquí os cito los formatos conocidos que admite Youtube, para que os hagáis una idea hacia dónde van las tendencias actuales:

- **Archivos WebM:** códec de vídeo Vp8 y códec de audio Vorbis (usado en HTML5).
- **Archivos .MPEG4, 3GPP y MOV:** compatibles normalmente con códec de vídeo H.264 y MPEG4 y códec de audio AAC
- **.AVI:** el formato de un gran número de cámaras (el códec de vídeo es MJPEG y el códec de audio es PCM)
- **.MPEGPS:** compatible normalmente con códec de vídeo MPEG2 y códec de audio MP2
- **.WMV**
- **.FLV:** códec de vídeo Adobe-FLV1 y códec de audio MP3

## Compresión de video

Cuando hablamos de audio hablamos de la compresión, como la capacidad de reducir el tamaño del fichero de audio con la menor pérdida de calidad posible. Formatos como MP3, FLAC, OGG, etc.. nos permiten realizar esa compresión con diferentes calidades y tamaños.

En el apartado del video también disponemos de compresión, y en este caso es muchísimo más importante, ya que por lo general un video ocupa muchísimo más tamaño que un fichero de audio, y sabiendo además que un video a parte de imagen, lleva audio incorporado (estéreo, multicanal, etc..)

Para comprimir o descomprimir un video necesitamos los CODECS (codificador/decodificador). Es muy importante distinguir entre el formato contenedor (AVI, MKV, MOV, OGV, etc..) y el codec utilizado en el video que está dentro de ese fichero contenedor.

Muchas veces puede ocurrir que tenéis un video .avi o cualquier otro formato y no lo podéis reproducir, ya que no está instalado el CODEC necesario para su decodificación en el sistema.

### Lista de CODECS más utilizados:

- **MPEG:** Hoy en día es de los más utilizados. Ha pasado por diferentes versiones (MPEG-1, MPEG-2, hasta MPEG-4 o H.264. Éste último codec se usa mucho en páginas web y dispositivos móviles. Los archivos suelen tener extensiones como MP4, m4a, m4v, etc. Suele usar el audio en formato AAC.
- **DivX:** muy usado debido a la compresión que realiza y que reduce el tamaño de los archivos, facilitando su distribución a través de la red.
- **Adobe Flash:** se ha convertido en un formato muy utilizado para la reproducción de video en internet. Youtube lo usa a la hora de transmitir sus videos en la red, aunque se están estandarizando hacia otros sistemas como HTML5, de tal manera que el video sea reproducido por el propio navegador sin la necesidad de instalar codecs adicionales.
- **DV:** utilizado por las cámaras de video digitales.
- **WMV:** es el que utiliza Microsoft con su sistema operativo Windows.

# Instalación de Darwin Streaming Server

Darwin Streaming Server es la versión OpenSource del servidor de streaming Apple Quicktime. Este servidor es el encargado de atender las peticiones de nuestros clientes y facilitar los videos usando el protocolo RTSP y codificados en formatos como MP4, MOV, MPEG2 o MP3.

[Página oficial de Darwin Streaming Server](#)

## Pasos para la instalación de Darwin Streaming Server en Debian

Vamos a realizar la instalación de la última versión disponible de Darwin Streaming Server:

Podeis copiar este script y ejecutarlo directamente o seguir los pasos de la instalación.

Al final de la instalación os va a pedir un usuario para administrar Darwin y una contraseña.

```
#!/bin/bash
# Instalamos los paquetes necesarios para compilar fuentes.
apt-get install build-essential wget

# Añadimos el grupo qtss
addgroup --system qtss

# Añadimos el usuario qtss en el grupo qtss
adduser --system --no-create-home --ingroup qtss qtss

# Nos descargamos el fichero fuente:
wget http://dss.macosforge.org/downloads/DarwinStreamingSrvr6.0.3-Source.tar

# Si no funciona la página de macosforge, descargar las fuentes desde aquí:
wget http://public.itmatrix.eu/Darwin_Streamr_6.0.3/DarwinStreamingSrvr6.0.3-Source.tar

# Descomprimos el fichero:
tar -xvf DarwinStreamingSrvr6.0.3-Source.tar

# Tenemos que aplicar unos patch para que funcione correctamente en Linux:
wget http://public.itmatrix.eu/Darwin_Streamr_6.0.3/Darwin_6.0.3_patches.zip

# Descomprimos los patches:
unzip Darwin_6.0.3_patches.zip

# Ejecutamos los patches:
patch -p0 < Patches/dss-6.0.3.patch
patch -p0 < Patches/dss-hh-20080728-1.patch

# Compilamos los binarios.
cd DarwinStreamingSrvr6.0.3-Source
./Buildit

# Reemplazamos el script de instalación con el nuevo proporcionado:
mv Install Install.orig
cp ../Patches/Install .

# Le damos permiso de ejecución:
chmod 755 Install

# Instalamos el sistema.
./Install
```

**Para comprobar si está funcionando, lo podremos hacer con:**

```
ps aux | grep treaming | grep -v grep
```

Administración de Darwin Streaming Server:


Para gestionar nuestro servidor de video accederemos a la dirección IP del mismo y al puerto puerto 1220.

`http://IP_del_servidor:1220`

Si el servidor estuviera parado se puede arrancar con:

`/usr/local/sbin/streamingadminserver.pl`

Página de acceso a la administración web del servidor Darwin Streaming Server:



### Log In to Darwin Streaming Server

Log in with an administrative username and password.

User Name:

Password:

Si nos hemos olvidado de la contraseña de acceso al servidor, la podemos modificar desde la línea de comandos con:

`qtpasswd administrator`

Una vez dentro seguiremos el asistente para terminar de configurar nuestro servidor:



### Setup Assistant MP3 Broadcast Password

The MP3 Broadcast password is required in order to receive MP3 broadcast streams.

New Password:

Re-enter New Password:



## Setup Assistant Secure Administration

This will enable encryption between the administration server and the web client. This feature requires that you have a valid streaming server SSL certificate installed.

☐ Secure Administration (SSL)

Prev

Cancel

Next



## Setup Assistant Media Folder

This is the folder or directory in which your media is stored.

Media Folder:

/usr/local/movies/

Prev

Cancel

Next



## Setup Assistant Streaming on Port 80

This allows you to stream through firewalls.

☐ Streaming on Port 80


Prev

Cancel

Finish


Disable Server

Server is Running



QuickTime

- Main
- Connected Users
- Relay Status
- General Settings
- Port Settings
- Relay Settings
- Log Settings
- Playlists
- Error Log
- Access History
- Log Out



Server Snapshot

Server: 192.168.1.103

Status: Started Tue, 8. May 2012 11:48:31

Current Time On Server: Tue, 8. May 2012 12:02:23

Up Time: 13 min 52 sec

DNS Name (default): 192.168.1.103

Server Version: 6.0.3

Server API Version: 5.0

CPU Load: 6.00%

Current # of Connections: 0

Current Throughput: 0 bps

Total Bytes Served: 0 Bytes

Total Connections Served: 0

## Preparar los videos para Darwin Streaming Server

Este servidor de video necesita que los videos que le pasemos estén en formato MOV, MP4 o 3GP, pero además tienen que estar preparados para hacer streaming. Ese proceso es lo que se conoce como **hint**, y con ello se consigue que se puedan transmitir utilizando el protocolo **RTSP**. Se puede realizar el hint de los archivos utilizando Quicktime Pro o bien mediante alguna herramienta disponible en el paquete MPEG4IP.

Si no hacemos el "**hint**" de los videos, éstos se podrán transmitir igualmente, pero necesitarán unos cuantos segundos más antes de comenzar la reproducción de los mismos, con lo que el funcionamiento es el mismo que con un servidor web Apache, y por lo tanto es inútil usar Darwin Streaming Server.

Si queremos codificar los videos, lo podemos realizar gratuitamente sin usar Quicktime Pro (de pago), empleando ffmpeg y mpeg4ip. ffmpeg tiene más características que QuickTime Pro y además soporta scripting, con lo que se podría programar para que realice las codificaciones de los videos durante la noche. Se recomienda usar un servidor independiente para esta tarea de codificación ya que consume muchos recursos.

Más información de cómo preparar los videos para Darwin Streaming Server en:

[http://soundscreen.com/streaming/compress\\_hint.html](http://soundscreen.com/streaming/compress_hint.html) <http://blogs.informedia.in/general/howto-hint-3gp-files-streaming-darwin-streaming-server/>

## Hacer las playlists de videos en Darwin

Si queremos crear una **playlist** de videos en Darwin, nos conectaremos via web a la dirección **IP** del servidor **puerto 1220** y en la sección de **Playlist**, crearemos la lista.

**IMPORTANTE:** Cuando creamos la lista de **ficheros** éstos **deben tener las mismas** características. Es decir no podemos poner un video que tenga una resolución determinada y en la misma lista otro video con una resolución mayor. Esta restricción se debe a que el reproductor de Quicktime no se adapta a diferentes resoluciones a la hora de reproducir videos en streaming.

Hay que recordar que la playlist se reproducirá el número de veces que indiquemos o indefinidamente (sequentially looped).

## Conectarse con un cliente a Darwin Streaming Server

Para probar el funcionamiento de Darwin Streaming Server tendremos que conectarnos con un cliente para ver si todo funciona correctamente. Podemos utilizar VLC para comprobar si funciona, aunque sólo reproducirá correctamente los videos en formato H.264 y algunos MP4, por lo que si queréis comprobar cualquier video, es mejor instalar Quicktime.

Previamente, desde la interfaz web tendremos que **crear una playlist** de videos o audio que queremos transmitir a los clientes. Disponemos de algunos videos y audios de ejemplo en la carpeta /usr/local/movies de nuestro servidor.

Para probar el **funcionamiento del playlist de videos** abriremos un reproductor con soporte RTSP, en este caso con Quicktime no hay ningún problema, y nos conectamos a la dirección, siguiente:

```
rtsp://IP_del_servidor/punto_montaje.sdp
```

Fijarse que hemos escrito **rtsp** en lugar de **http**.

También podríamos conectarnos y **visualizar alguno de los videos** disponibles en la carpeta de videos, directamente con:

```
rtsp://IP_del_servidor/nombre_del_fichero.mp4
```



**Si lo que queremos es escuchar el playlist de audio MP3** nos tendremos que conectar (en este caso VLC sí que reproduce correctamente los MP3) a la siguiente dirección:

```
http://IP_del_servidor:8000/punto_montaje (sin .sdp)
```

## Manual en PDF del Administrador de Darwin Streaming Server





**PDF con información adicional de configuración y ejemplos** para crear páginas web, que muestren videos obtenidos de Darwin Streaming Server:

Pulsa en el siguiente enlace para descargarte el fichero PDF: [Archivo:Manual-Darwin-Streaming-Server-Guide.pdf](#)

## Instalación de RED5 Streaming Server

RED5 es la versión gratuita del servidor de Flash Video de Adobe.

La página oficial de RED5 se encuentra en: <http://www.red5.org/>



Algunas de las **características** que soporta RED5 son:

- Streaming de Audio/Video (FLV y MP3)
- Grabación de streams de clientes (sólo FLV)
- Shared Objects (permiten la creación de aplicaciones que se comuniquen con otras en tiempo real)
- Publicación de Streams en Directo (H264 en directo ya soportado)
- Remoting ( AMF0/ AMF3)

### Pasos para una instalación rápida de Red5 1.0.1

```
# La verdad que es muy sencillo de instalar en Debian ya que sólo tenemos que ejecutar:
# Primero actualizaremos el sistema.
apt-get clean
apt-get update
apt-get upgrade

# A continuación instalamos el paquete red5-server.
apt-get install red5-server

# Eso sí, tendremos que esperar un rato por que ocupa unos cuantos MB.
```

Una vez instalado el paquete base, tendremos que instalar las demos y aplicaciones, ya que este tipo de instalación no las incorpora. Para ello haremos lo siguiente:

```
wget http://www.red5.org/downloads/red5/1_0_1/red5-1.0.1.zip
mkdir red5
cd red5

# Si nos da algún error con la instrucción siguiente, instalamos primero unzip
# apt-get install unzip
unzip ../red5-1.0.1.zip

cd red5-server-1.0
cp -R webapps/root/demos /var/lib/red5/webapps/root/demos
cp -R webapps/installer /var/lib/red5/webapps/installer

find /var/lib/red5/webapps/ -type d -exec chown _red5 {} \;
cd ..rm -rf red5*service red5-server restart
```

A partir de ahora podremos acceder a la IP del servidor con <http://x.x.x.x:5080/> y podremos instalar la demo que queramos.

Para nuestro caso en particular, instalaremos la demo **oflaDemo** (openFlashDemo), que nos permitirá emitir en streaming nuestros videos en formato Flash.

Para ello tendremos que copiar los videos que queramos emitir (codec Flash Video o FLV o H264-MPEG4) a la carpeta:  
`/var/lib/red5/webapps/oflaDemo/streams`

**Videos descargados de Youtube con extensión FLV o MP4 los reproduce sin ningún tipo de problema y en además en Full HD 1920x1080.**

## Instalación manual de RED5 versión 1.0.1

Versiones instaladas en este tutorial:

- Red5 1.0.1
- OpenJDK 1.7.0\_03
- Ant 1.9.0

Última versión Debian requerida (5 por lo menos), que podremos comprobar con cualquiera de los siguientes comandos:

```
cat /etc/debian_version
cat /proc/version
uname -a
cat /etc/issue
cat /etc/debian_version
```

### Proceso de Instalación:

```
# Editar el fichero /etc/apt/sources.list y añadir:
# deb http://ftp.us.debian.org/debian stable main non-free contrib

# Actualizamos las apt:
apt-get update

# Instalamos Subversion:
apt-get -y install subversion

# Instalamos los paquetes necesarios para compilar:
apt-get -y install dpkg-dev debhelper dh-make devscripts fakeroot

# Instalamos openJDK:
apt-get -y install openjdk-7-jre
apt-get -y install openjdk-7-jdk
java -version

# Instalamos Apache Ant (http://ant.apache.org/):
wget http://www.apache.org/dist/ant/binaries/apache-ant-1.9.0-bin.tar.gz
tar zxvf apache-ant-1.9.0-bin.tar.gz
mv apache-ant-1.9.0 /usr/local/ant
export ANT_HOME=/usr/local/ant
/usr/local/ant/bin/ant -version

# Descargamos la última versión de Red5:
svn co http://red5.googlecode.com/svn/java/server/tags/1_0/ red5

# Ejecutamos Ant para compilar Red5:
cd red5
/usr/local/ant/bin/ant

# Nos tocará esperar un buen rato a que resuelve dependencias y compile la aplicación.
# Aproximadamente unos 5-6 minutos.

# Una vez compilada la aplicación instalaremos Red5:
mkdir /usr/local/red5
cp -R dist/* /usr/local/red5/
cd /usr/local/red5
chmod 755 red5.sh

# Arrancamos Red5 en Background:
cd /usr/local/red5
./red5.sh >> red5.log &

# Comprobamos los puertos si son correctos (puerto 5080):
netstat -ant

# Instalamos nuestras aplicaciones en /usr/local/red5/webapps .
# Nos conectamos con un navegador a:
http://IP_SERVER:5080
```

```

# Y desde allí podemos instalar OflaDemo.
# Una vez instalada la aplicación hay que reiniciar el servidor. Véase #### MÁS INFORMACIÓN ####.

# Una vez instalada para probarla podemos editar la demo de ejemplo con:
nano /usr/local/red5/webapps/oflaDemo/index.html

Y ponemos la IP correspondiente a nuestro servidor en las secciones del script:
rtmp://x.x.x.x/oflaDemo y rtmpt://x.x.x.x/oflaDemo

# Para probar las demos vamos a:
http://IP_SERVER:5080/oflaDemo/index.html

#####
#### MÁS INFORMACIÓN ####
#####

# Las webapps están en:
/usr/local/red5/webapps

# La página web por defecto está en:
/usr/local/red5/webapps/root

# Los videos para streaming se copiarán a:
/usr/local/red5/webapps/oflaDemo/streams

# Cada vez que añadamos apps a Red5 en /usr/local/red5/webapps necesitamos reiniciarlo:

# Parar Red5:
cd /usr/local/red5
./red5-shutdown.sh

# O hacemos un kill a los procesos de Red5:
ps aux | grep red5
kill $process_id

# Arrancar Red5:
cd /usr/local/red5
./red5.sh >> red5.log &

```

## Gestionar el servidor

El servidor de RED5 lo podemos administrar via web, para ello tendremos que conectarnos con un navegador a la dirección IP del servidor y puerto 5080

```
http://IP_DEL_SERVIDOR:5080
```

**La página de acceso a RED5 muestra este contenido:**



This page is used to test the proper operation of the Red5 server after it has been installed. If you can read this page it means that the Red5 server installed at this site is

If you would like to watch an instructional video on Red5 and have access to YouTube, [click here](#)

If you prefer to jump right in, follow these steps:

- Debian "red5-server" package doesn't contains /demos or /installer webapps. Please note that /installer and /demos webapps are for demonstration purpose only and installed on a public red5 server (no security at all in those applications).

Furthermore, they contains non-free stuffs :

- \*.swf binary-only (Flash Player) files
- \*.flv movies trailers like Avatar / Toy Story 3

As root, you can install /demos (SWF files) and /installer webapp, by issuing :

```
wget http://www.red5.org/downloads/0_9/red5-0.9.1.tar.gz
tar xvfz red5-*.tar.gz
cd red5-*
cp -R webapps/root/demos /var/lib/red5/webapps/root/demos
cp -R webapps/installer /var/lib/red5/webapps/installer
find /var/lib/red5/webapps/ -type d -exec chown _red5 {} \;
cd ..
rm -rf red5-*
```

*Excerpt from /usr/share/doc/red5-server/README.Debian*

- **Install** a ready-made application
- Launch a **demo**

Further information about Red5 can be found at any of these links: [official homepage](#) | [wiki](#) | [old homepage](#)

Be sure to subscribe [the mailing list](#) to get help or to be notified about new versions. You can also use the mailing list to ask questions about Red5.

## Comandos de reinicio y parada del servidor:

```
service red5-server start stop status restart
```

## Video en Youtube con introducción al manejo del servidor (Inglés):

En la página de RED5 hay mucha información y documentación para configurar este servidor. Con RED5 se pueden crear aplicaciones Flash como servicios de videoconferencia vía web, aplicaciones Flash, etc.. Pero eso ya se sale del objetivo de este tutorial.

<http://wiki.red5.org/wiki/Documentation>

## Instalación de videos y aplicaciones demo en el servidor

Para acceder a videos de demostración de RED5, necesitamos bajarnos el fichero fuente del mismo. Para ello haremos los pasos siguientes:

```
# Nos descargamos el código fuente de RED5.
wget http://red5.org/downloads/red5/1_0/red5-1.0.0-RC1.tar.gz

# Lo descomprimos
tar xvfz red5-*.tar.gz

# Accedemos a la carpeta
cd red5-*

# Copiamos la carpeta de demos y la carpeta installer a /var/lib/red5/webapps...
cp -R webapps/root/demos /var/lib/red5/webapps/root/demos
cp -R webapps/installer /var/lib/red5/webapps/installer

# Asignamos como propietario de esos directorios a red5
find /var/lib/red5/webapps/ -type d -exec chown _red5 {} \;

# Si queremos podremos borrar el paquete que hemos descargado para aprovechar recursos.
cd ..
rm -rf red5-*

# Una vez hecho ésto tendremos que reiniciar de nuevo el servidor RED5.
service red5-server restart

# A partir de ahora podemos acceder a la dirección:
http://IP_DEL_SERVIDOR:5080/installer/

# Y desde allí instalar algunas de las aplicaciones Flash incorporadas en la demo.

# Una vez instaladas, podemos probarlas desde:
http://IP_DEL_SERVIDOR:5080/demos/

# Si queremos colocar más archivos en el servidor de RED5, simplemente tendremos que copiar
# los nuevos ficheros FLV a la carpeta: /var/lib/red5/webapps/oflaDemo/streams/
cp *.flv /var/lib/red5/webapps/oflaDemo/streams/
```

Una vez que tenemos el installer y las demos instaladas, lo que tenemos que hacer es acceder al *installer* e instalar las aplicaciones que queramos probar. Luego desde la página de demos podemos comprobar el correcto funcionamiento de RED5.

## Conectarnos con un cliente al servidor RED5

Si queremos visualizar los videos que hemos insertado en la carpeta de streams de RED5, tendremos que crear una página web con algún reproductor como jwplayer que soporte la reproducción del formato FLV con el protocolo RTMP.

Necesitáis descargaros los siguientes ficheros, necesarios para el reproductor:

- jwplayer.js
- jwplayer.html5.js

- jwplayer.flash.swf

Esos ficheros los teneis que poner en la misma carpeta dónde tengáis la página .html con el reproductor.

En la página oficial del reproductor JWPLAYER tenéis la descarga de los ficheros (jwplayer.js y player.swf):  
<http://www.longtailvideo.com/players/jw-flv-player/>

Aquí tenéis un **ejemplo de Página HTML** que incluye el código necesario para reproducir un video alojado en vuestro servidor RED5. Es necesario además que tengáis la aplicación oflaDemo instalada en vuestro servidor RED5 (sección Installer).

Si queremos alojar vuestros propios videos FLV o MP4, tendreis que enviarlos previamente al directorio de RED5:

**/var/lib/red5/webapps/oflaDemo/streams**

## Ejemplo de página web con reproductor JWPLAYER y servidor RED5 con RTMP

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejemplo de Video Avatar desde RED5 en Streaming</title>
<meta charset="utf-8">
<script type="text/javascript" src="jwplayer.js"></script>
</head>
<body>
<h1>Reproduccion de Video desde RED5 Streaming Server usando RTMP.</h1>
<div id="reproductor">Cargando video...</div>

<script type='text/javascript'>
jwplayer('reproductor').setup({
'file': 'rtmp://192.168.1.24/oflaDemo/firework.mp4',
'controlbar': 'bottom',
'width': '848',
'height': '360'
});
</script>

<body>
</html>
```

Esta página la podéis colocar en la raíz de vuestro servidor web y mostrar el video de ejemplo para comprobar si funciona. Tendréis que modificar los datos referentes a la IP y el nombre del video que queréis reproducir.

Más información sobre configurar el reproductor en:

<http://www.longtailvideo.com/support/jw-player/28854/using-rtmp-streaming/>

## Creación de Playlist en el lado del servidor con RED5

Necesitamos descargarnos una aplicación para instalar en el servidor. Dicha aplicación se encuentra en  
<http://red5.electroteque.org/dev/demos/playlistTest.zip>

Puedes descargarla desde aquí también: [http://web.archive.org/web/\\*/http://red5.electroteque.org/dev/demos/playlistTest.zip](http://web.archive.org/web/*/http://red5.electroteque.org/dev/demos/playlistTest.zip)

Copia local [Archivo:PlaylistTest.zip](#)

```
wget http://red5.electroteque.org/dev/demos/playlistTest.zip

#Una vez descargada, descomprimos el zip.
unzip playlistTest.zip

# Si no disponemos de unzip lo instalamos con:
apt-get install unzip

# Accedemos a la carpeta playlistTest
```

```

cd playlistTest

# Ejecutamos ant
ant

# Cuando termine de compilar vamos a la carpeta dist
cd dist

# Nos copiamos el fichero playlistTest.war al directorio de webapps de RED5
cp playlistTest.war /var/lib/red5/webapps/

# Reiniciamos el servidor y ya tenemos disponible un nuevo punto llamado playlistTest
service red5-server restart

# La ruta de ese nuevo punto de montaje es:
cd /var/lib/red5/webapps/playlistTest/streams

# Si accedéis veréis que contiene algunos ficheros de muestra.

# La configuración de esa aplicación se encuentra en el fichero:
nano /var/lib/red5/webapps/playlistTest/WEB-INF/red5-web.properties

# Aquí se muestran algunos parámetros de ese fichero de configuración:
webapp.contextPath=/playlistTest
webapp.virtualHosts=*, localhost, localhost:8088, 127.0.0.1:8088

playlist.repeat=true
playlist.path=streams/
playlist.pattern=*.flv
playlist.streamName=webtv
playlist.runOnStart=true

```

Todos los ficheros FLV que metamos en playlistTest serán reproducidos en un carrusel por RED5. **/var/lib/red5/webapps/playlistTest/streams**

Ejemplo de página web que reproduce los videos FLV que están en la carpeta streams de playlistTest a modo de carrusel:

```
{ Pendiente de completar.. }
```

Server Side Playlists: <http://wiki.red5.org/wiki/Examples/ServerSidePlaylists>

Más información: <http://fms.denniehoopingarner.com/newapp.html>

--Veiga (discusión) 14:28 13 may 2013 (CEST)