

## Streaming de vídeo a través de Internet

Existen diferentes maneras para conseguir transmitir un vídeo a través de la red a muchos usuarios a la vez. Una de las opciones más razonables –y la única cuando queremos emitir un evento en directo- es el *streaming*. Típicamente, para poder ver un vídeo en el ordenador se necesita que el archivo se encuentre en el equipo. El vídeo, sin embargo, es un contenido que requiere archivos muy grandes, por lo que la descarga del mismo desde la red puede hacerse tediosa, e inviable cuando se precisa de la visión en tiempo real. Con el *streaming* este problema se soluciona: el usuario –cliente- simplemente hace clic sobre un link en una página Web y el vídeo comienza a reproducirse desde otro ordenador –servidor- sin necesidad de que éste sea el del usuario.

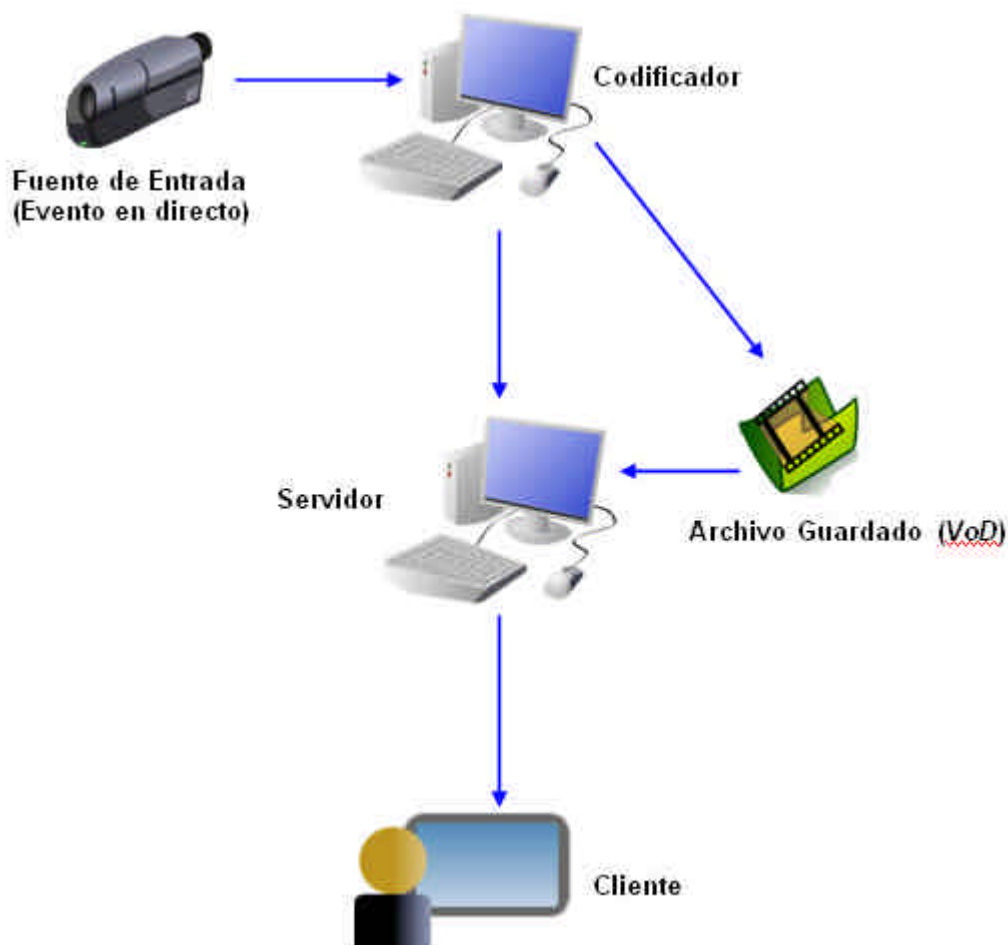
A través de este monográfico se pretende realizar una introducción práctica al *streaming*, explorando las diferentes opciones que existen en la actualidad y explicando el funcionamiento de las más importantes. Para el mismo se han utilizado las versiones gratuitas de los programas necesarios para realizar la emisión y recepción de vídeo en tiempo real a través de la red; posteriormente se hará referencia a las limitaciones que esto conlleva.

Existen diferentes maneras para conseguir transmitir un vídeo a través de la red a muchos usuarios a la vez. Una de las opciones más razonables –y la única cuando queremos emitir un evento en directo- es el *streaming*. Típicamente, para poder ver un vídeo en el ordenador se necesita que el archivo se encuentre en el equipo. El vídeo, sin embargo, es un contenido que requiere archivos muy grandes, por lo que la descarga del mismo desde la red puede hacerse tediosa, e inviable cuando se precisa de la visión en tiempo real. Con el *streaming* este problema se soluciona: el usuario –cliente- simplemente hace clic sobre un link en una página Web y el vídeo comienza a reproducirse desde otro ordenador –servidor- sin necesidad de que éste sea el del usuario.

Un ejemplo de este tipo de transmisión sería la página de vídeo bajo demanda *YouTube*: cuando vemos un vídeo a través de esta popular página éste no se encuentra en nuestro ordenador, sino que se recibe en tiempo real desde un servidor remoto. En este caso el vídeo está previamente guardado, pero permite aclarar los conceptos que nos llevarán al *streaming* como sistema de emisión de eventos en tiempo real.

A través de este tutorial se pretende realizar una introducción práctica al *streaming*, explorando las diferentes opciones que existen en la actualidad y explicando el funcionamiento de las más importantes. Para el mismo se han utilizado las versiones gratuitas de los programas necesarios para realizar la emisión y recepción de vídeo en tiempo real a través de la red; posteriormente se hará referencia a las limitaciones que esto conlleva.

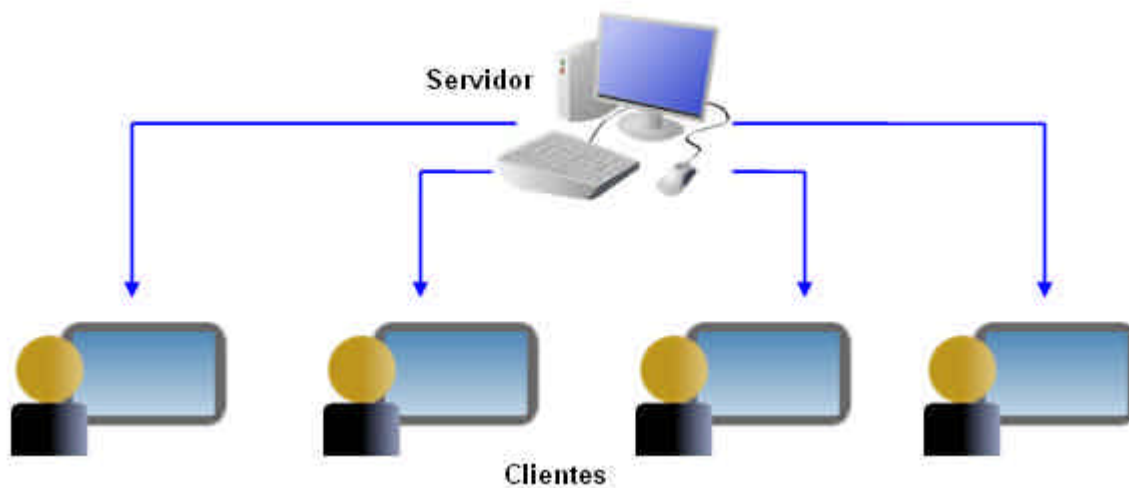
En primer lugar se deben conocer qué elementos son necesarios para realizar este proceso y qué opciones de transmisión existen en la actualidad. Dos de estos elementos han sido introducidos ya anteriormente: el **cliente** y el **servidor**, es decir, el usuario que quiere ver el vídeo y la máquina que los posee y que controla el acceso de los usuarios. Además, se precisa de una **cámara** que registre el evento y de un equipo con una tarjeta capturadora que **codifique** el vídeo y lo mande al servidor. Típicamente, el equipo que codifica el vídeo y el servidor es el mismo, aunque es posible hacerlo por separado. Si la transmisión, en lugar de hacerse en directo, se realiza a través de un archivo pre-grabado el sistema se denomina **Vídeo bajo demanda (VoD)**.



**Figura 1 – Elementos básicos de un sistema de transmisión streaming**

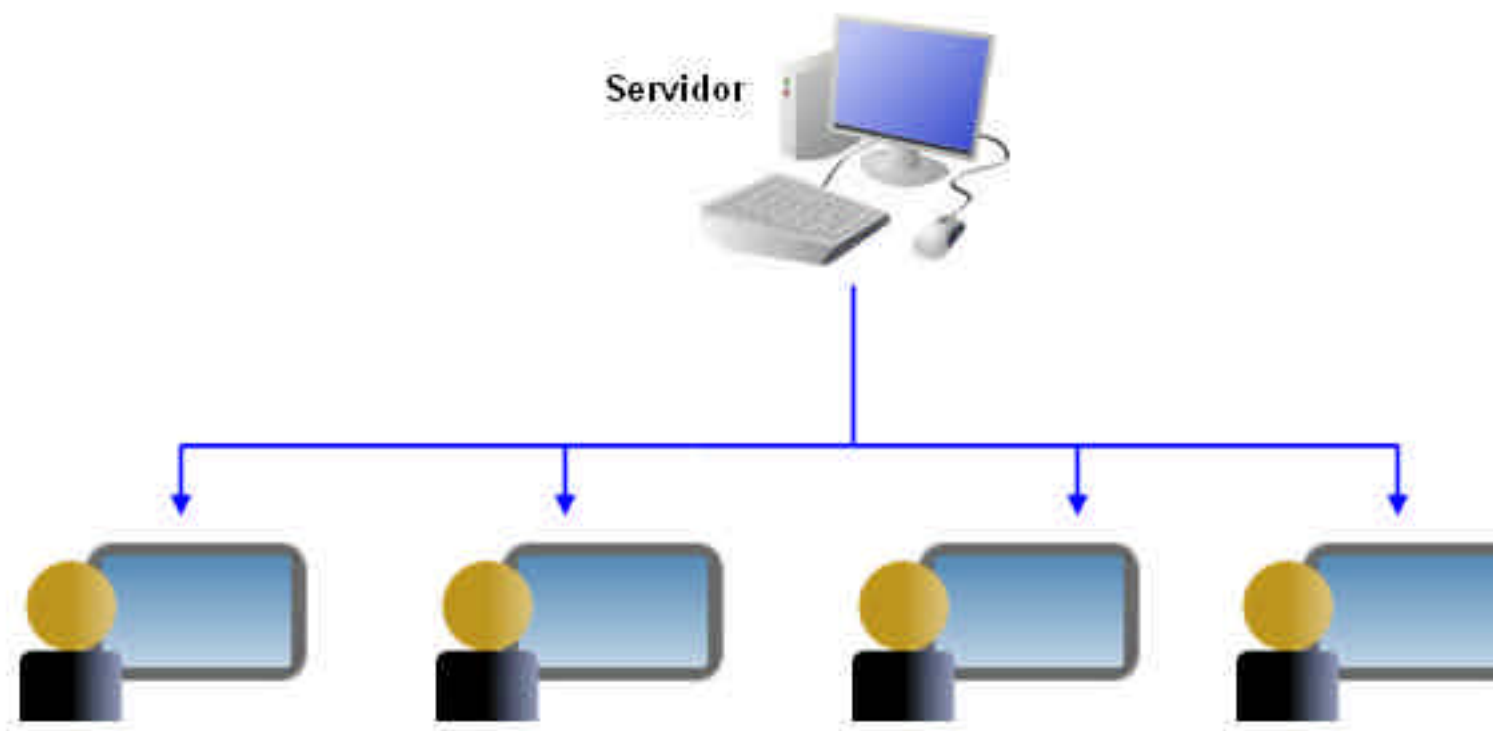
La forma en la que se envía la información a diferentes usuarios permite hacer una clasificación de las posibilidades que ofrece el servicio. Si cada usuario que accede al vídeo obtiene un flujo de datos independiente el proceso se denomina **Unicast**. Si, por el contrario, el mismo flujo de datos se utiliza para servir el vídeo a varios usuarios se dice que se está haciendo **Multicast**. Es importante conocer en qué condiciones se quiere emitir el contenido multimedia, puesto que definirá el ancho de banda necesario en el servidor para que la transmisión sea correcta. En un escenario de unicast, el método más simple de realizar *streaming*, el ancho de banda necesario aumenta linealmente en relación al número de usuarios; a mayor número de usuarios, más ancho de banda consumido. En multicast, sin embargo, este problema se solventa enviando un único flujo de datos al que se conectan los usuarios, por lo que reduce drásticamente el ancho de banda necesario para la transmisión del contenido.

## Unicasting



*Figura 2 – Ejemplo de red Unicast*

### Multicasting



*Figura 3 – Ejemplo de red Multicast*

De forma general, se precisarán tres programas software para realizar el streaming: el vídeo ha de capturarse a través de un programa productor que envía el resultado a otro programa que administra el servidor. Por último, el cliente debe poseer un reproductor compatible con el formato de archivo utilizado. Existen varias soluciones a nivel software que pasaremos a estudiar a continuación. Las más importantes son las ofrecidas a través de Real Media, Windows Media y Flash. Además, se analizarán las características básicas del servidor Darwin de Apple y se ofrecerán los conceptos básicos de la solución a través de VideoLAN, una plataforma gratuita para realizar streaming.

Para este tutorial se ha utilizado una tarjeta capturadora ViewCast Osprey 200 y una cámara JVC GR-DVP9E conectada a través de la salida de vídeo compuesto. Las capturas de pantalla pueden variar en función del hardware y de la versión del programa utilizado.

## Real Media

La primera opción que analizaremos en este tutorial es la que ofrece Real Media ([www.realnetworks.com](http://www.realnetworks.com)). Como se ha indicado anteriormente, el primer paso para la realización del streaming es la codificación de vídeo desde una o varias cámaras. La captura puede hacerse de manera analógica a través de una tarjeta capturadora o mediante un puerto FireWire (DV) si la cámara lo permite –en los últimos modelos de cámaras con disco duro ni siquiera es necesario disponer de un puerto DV, el USB es suficiente-. El programa que realiza la captura y la envía al servidor es, en este caso, el **Real Producer Basic** ([www.realnetworks.com/products/producer/basic.html](http://www.realnetworks.com/products/producer/basic.html)). La versión de prueba gratuita en inglés permite hacer unicast con tres bitrates de codificación diferentes, ofreciendo la posibilidad de realizar multicast si se adquiere la versión completa. El servidor que propone Real Networks es el **Helix Server** ([www.realnetworks.com/products/media\\_delivery.html](http://www.realnetworks.com/products/media_delivery.html)), con una licencia de prueba de hasta 5 conexiones simultáneas, y el reproductor necesario en el cliente es el **Real One** ([http://spain.real.com/freeplayer\\_r1p.html](http://spain.real.com/freeplayer_r1p.html)), descargable de forma gratuita. Puesto que el servidor tiene que estar listo para que el Real Producer se comuniquen con él, comenzaremos analizando la configuración básica del Helix Server.

## Helix Server / Administración de Contenidos

### **Requisitos Mínimos**

- Procesador Pentium 4 2.4 GHz
- 512 MB de RAM por procesador
- 100 MB de espacio en disco duro + espacio para los archivos + Log

### **Requisitos Recomendados**

- Procesador Pentium 4 o equivalente, 2.4 GHz o superior
- 1 GB de RAM por procesador
- Conexión de Internet de Alta Velocidad

El programa de instalación de Helix Server nos guiará en el proceso de configuración básica del servidor. En primera instancia pedirá la ubicación del archivo de licencia, que Real Networks envía al correo tras un sencillo proceso de registro. El archivo básico, como indicábamos antes, permite la conexión simultánea de 5 clientes a través de

archivos codificados mediante Real Media. Otras licencias de pago permiten trabajar con multicast y otros formatos de vídeo. La siguiente tabla clarifica las posibilidades que ofrece la licencia gratuita de prueba:

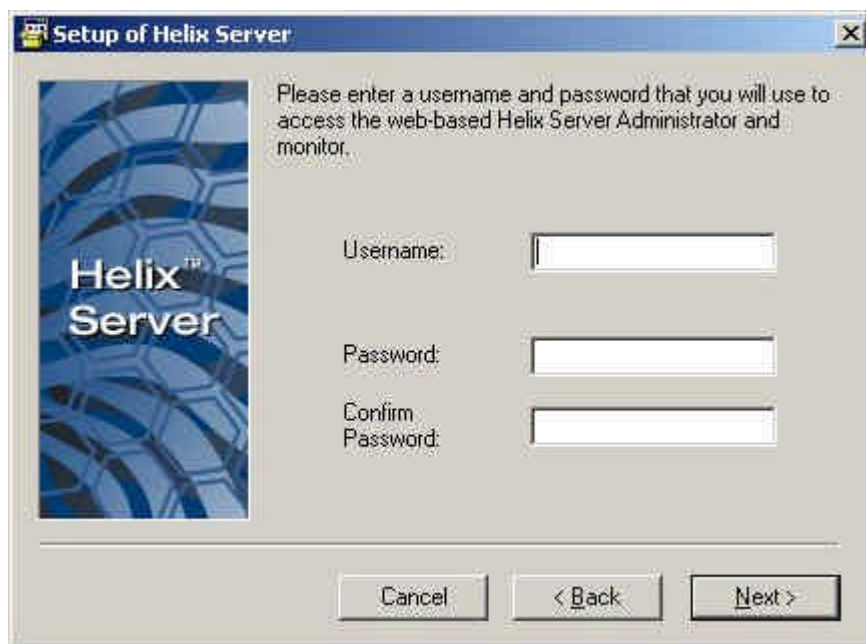
**Capacidad**

Máximo Número de Conexiones 5

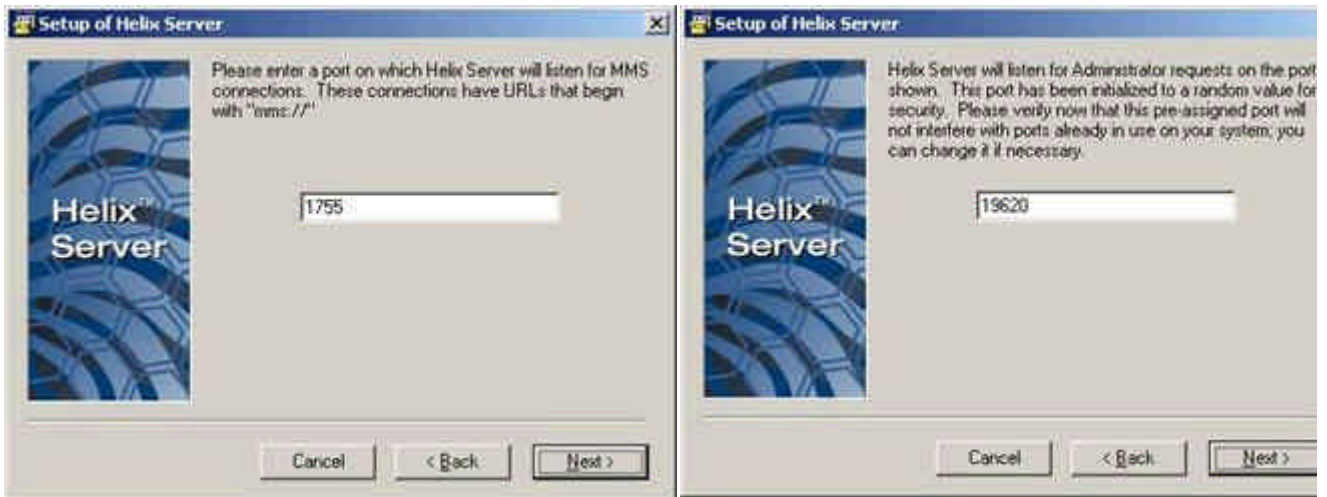
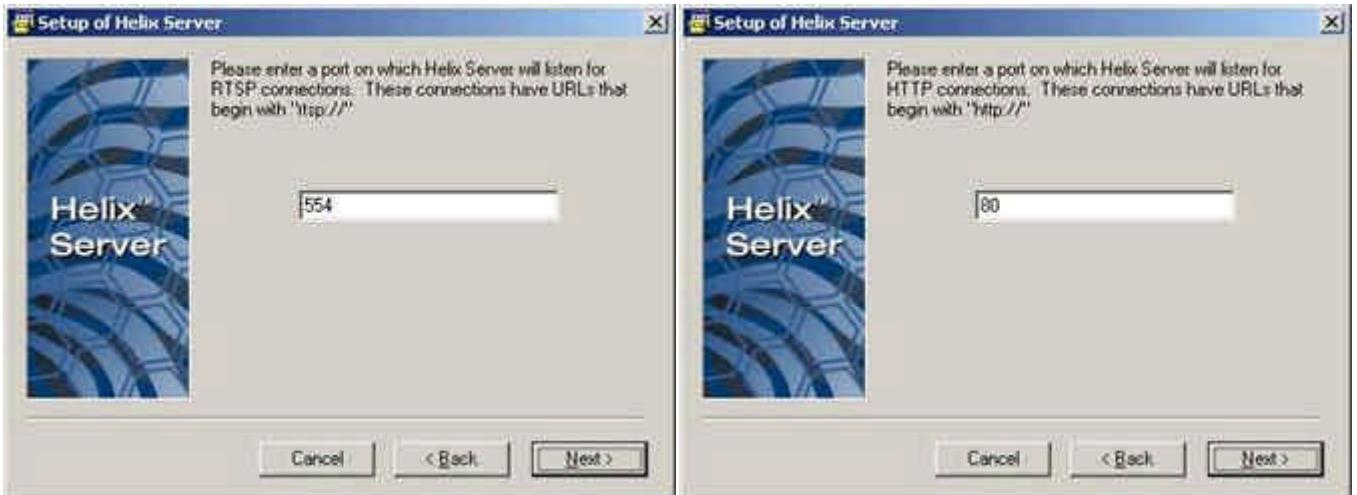
**Tipos de Archivo Multimedia**

3GPP Packetization	No
MultiRate Container	Sí
RealAudio	Sí
RealVideo	Sí
RealFlash	No
RealPix	Sí
RealEvents	Sí
RealText	Sí
MPEG 4	No
QuickTime	No
Windows Media	No

El siguiente paso en la instalación consiste en proponer un nombre de usuario y password para la administración del servidor. Debemos recordar estos datos, puesto que, al margen de poder gestionar el servidor, serán necesarios para la comunicación entre el Real Producer y el servidor.



Las siguientes pantallas permiten la configuración de los puertos que se utilizarán en la comunicación. Salvo que haya una causa justificada para cambiarlos se recomienda usar los valores por defecto, habilitando las conexiones a través de los mismos en el Firewall – si se dispone de alguno- .

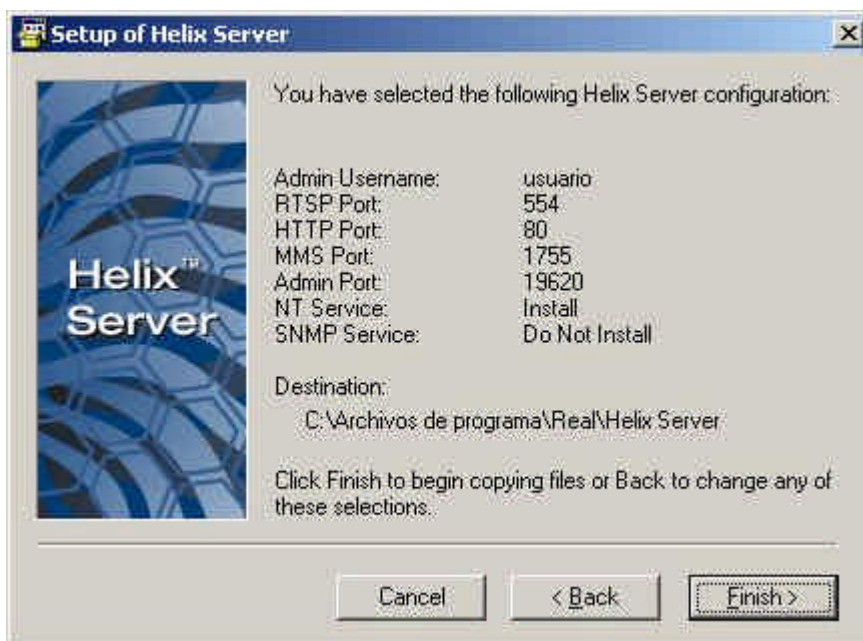


Además, configuraremos el servidor para que se inicie al encender el ordenador, de forma que esté siempre funcionando y tanto el Real Producer como los clientes puedan siempre comunicarse con él. Para ello, dejaremos marcada la opción **Install Helix Server as an NT Service** en el cuadro de diálogo siguiente.





Por último, una pantalla realiza un resumen de la configuración que acabamos de realizar antes de instalar. Pulsamos **Finish** y dejamos que el programa se instale en la ubicación que le hayamos propuesto al inicio.

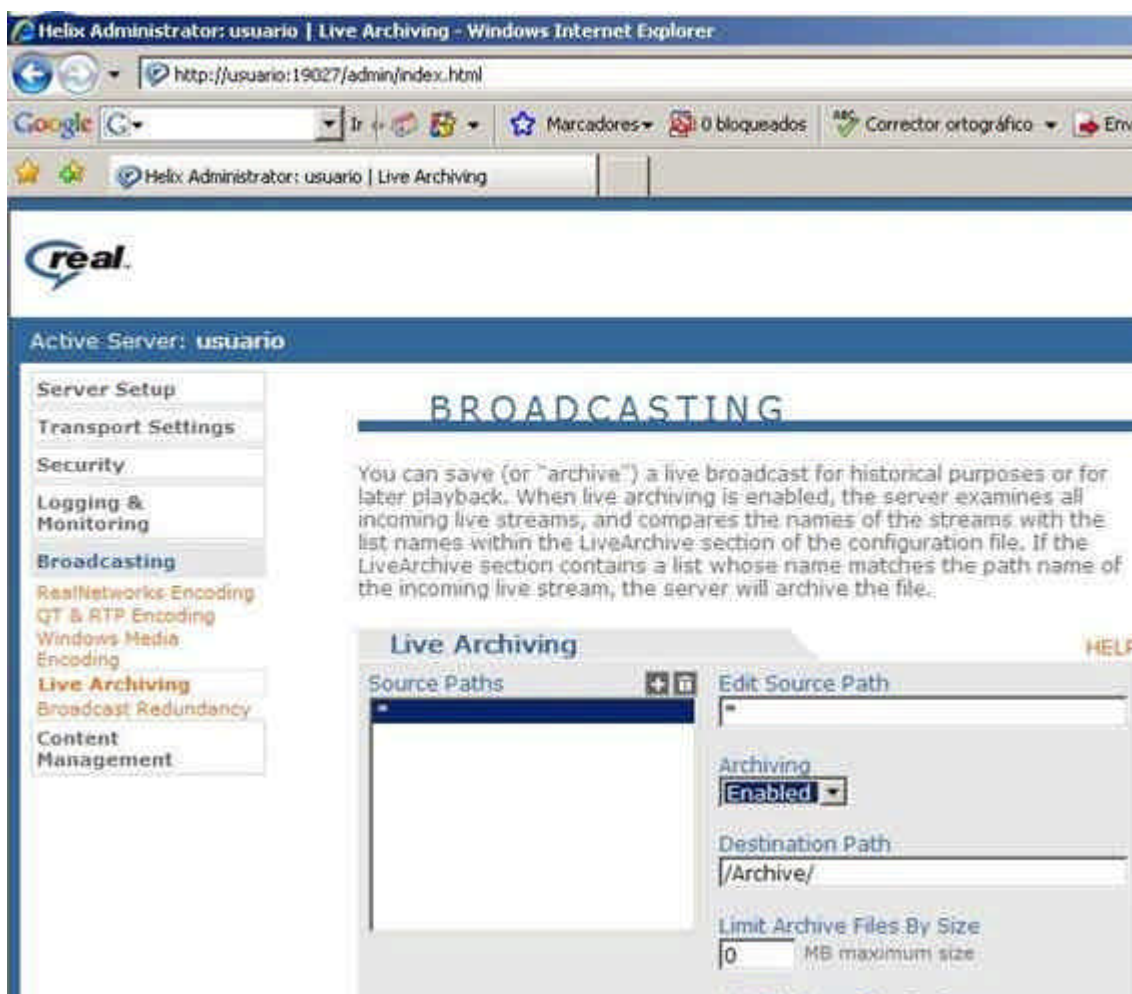


Una vez instalado, el servidor se encontrará ya en funcionamiento tras reiniciar el sistema –aunque hay accesos directos desde el grupo de programas creado para poder arrancarlo sin reiniciar-. Si pulsamos sobre el icono de **Helix Server Administrator** (habrá un acceso directo en el escritorio si no se ha especificado lo contrario en la instalación) se abre una página Web que permite la gestión del servidor una vez hayamos metido el usuario y contraseña que introdujimos en la instalación.

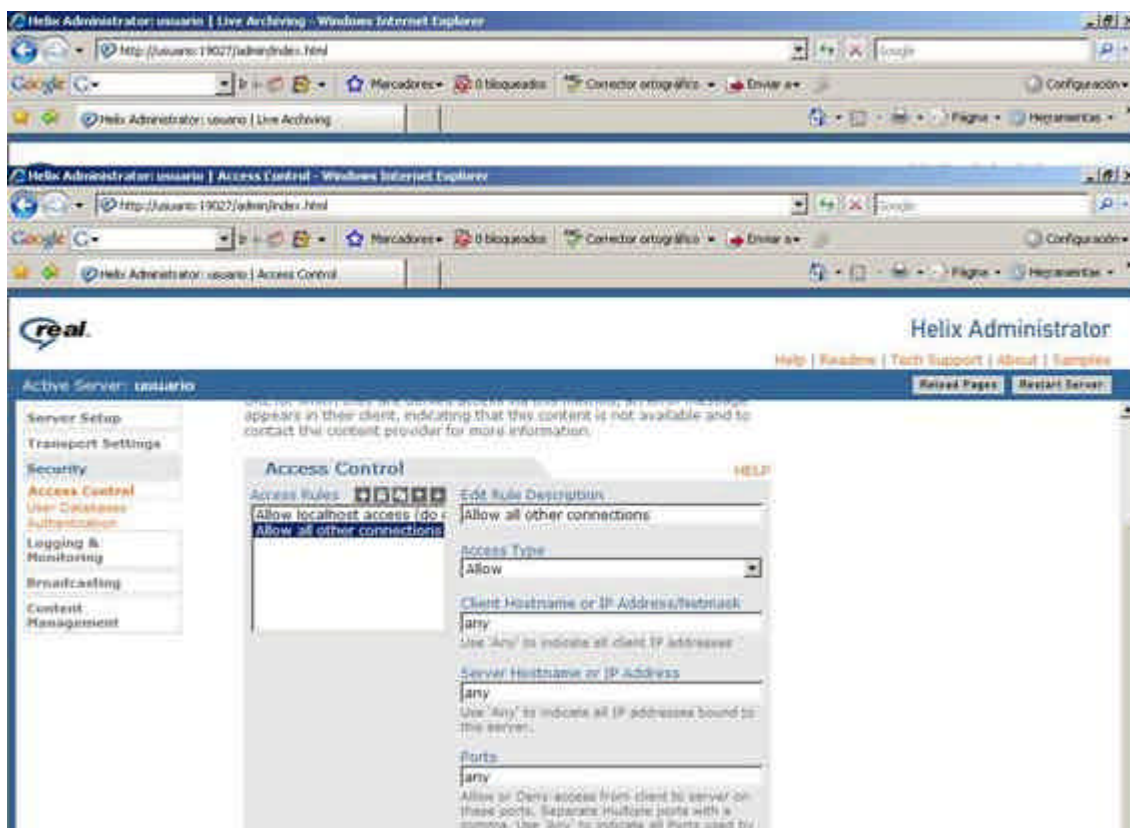


En principio no debemos cambiar nada para que el servidor funcione, pero habría que preguntarse qué queremos hacer con los archivos que se emitan en tiempo real. Si preferimos que, al margen de que el cliente pueda verlos en directo, el servidor los guarde para un visionado posterior, hay que especificárselo al servidor a través de la opción **Live Archiving**, en la pestaña **Broadcasting**, activando la opción **Archiving -> Enabled**.





El acceso a los contenidos del servidor se realiza, por defecto, sin ningún tipo de control; cualquiera que sepa la dirección del servidor puede conectarse y visualizar el vídeo –pero no administrarlo-. Si se desea variar esta configuración, para solicitar una contraseña o limitar el uso a determinados equipos, puede hacerse desde **Security -> Access Control -> Access Rules -> Allow all other Connections**, modificando la pestaña para permitir/denegar las conexiones de determinadas direcciones IP o a través de determinados puertos.



El servidor no precisa de más cambios para trabajar en su modo más básico. Para funciones más avanzadas se recomienda acudir a su completo manual en línea.

(<http://service.real.com/help/library/guides/helixuniversalserver/realsrvr.htm?page=htmlfiles/splitting.htm>)

A continuación se procede a explicar el proceso de codificación del contenido multimedia mediante el programa Real Producer Basic y la comunicación entre ambos programas.

## Real Producer Basic / Codificación de Contenidos

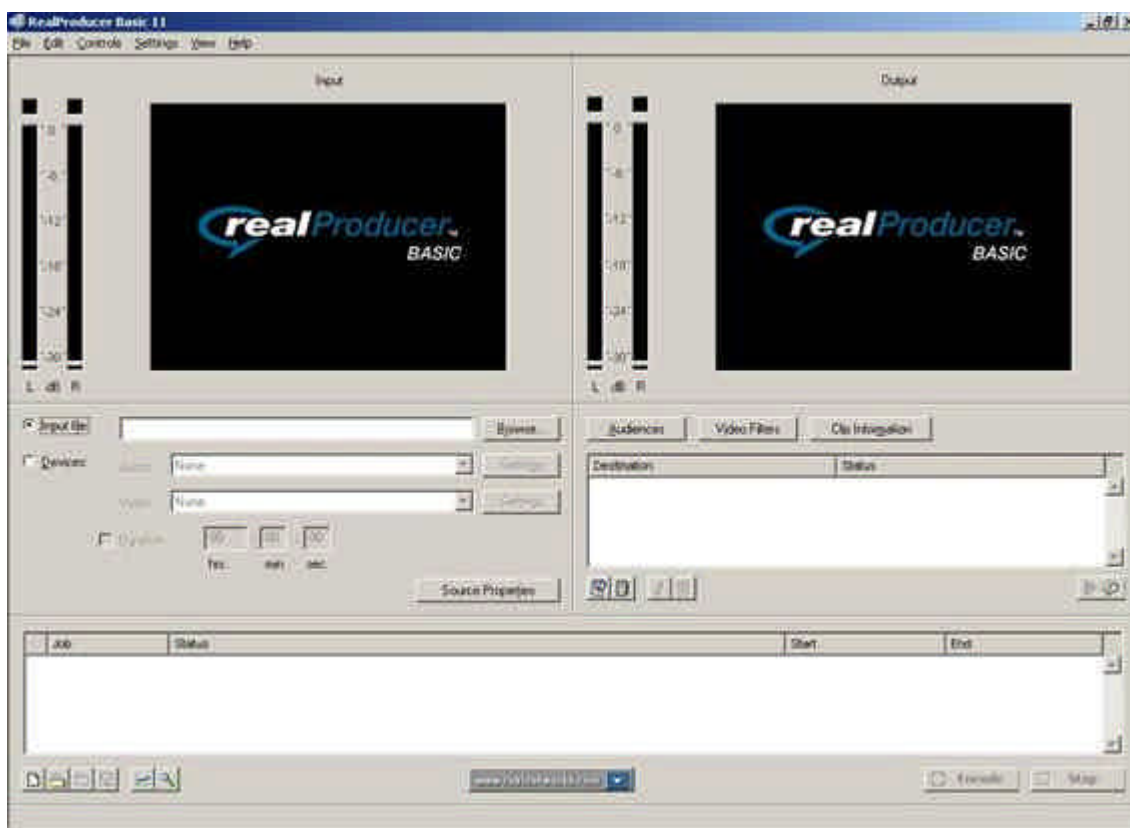
### **Requisitos Mínimos**

- Procesador 500 MHz
- 128 MB de RAM
- Tarjeta de sonido compatible con Windows
- Tarjeta Gráfica compatible con Windows
- Windows 2000
- IE 5 o posterior, FireFox

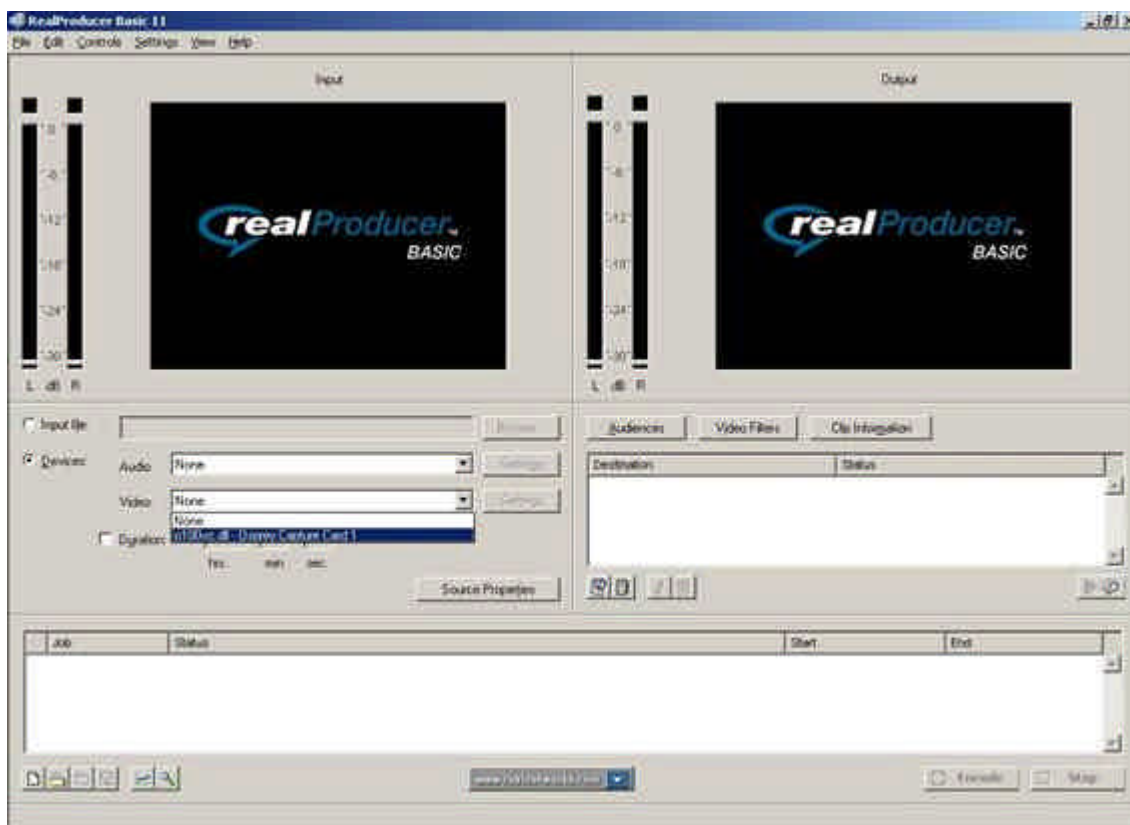
### **Requisitos Recomendados**

- Procesador Pentium 4 o equivalente, 2.4 GHz o superior
- 512 MB de RAM, 1 GB con Windows Vista
- Conexión de Internet de Alta Velocidad
- Tarjeta de video Osprey 230, Osprey 500
- Tarjeta de sonido Osprey 230, SounBlaster Live o AWE 64
- Windows XP SP2 o Vista

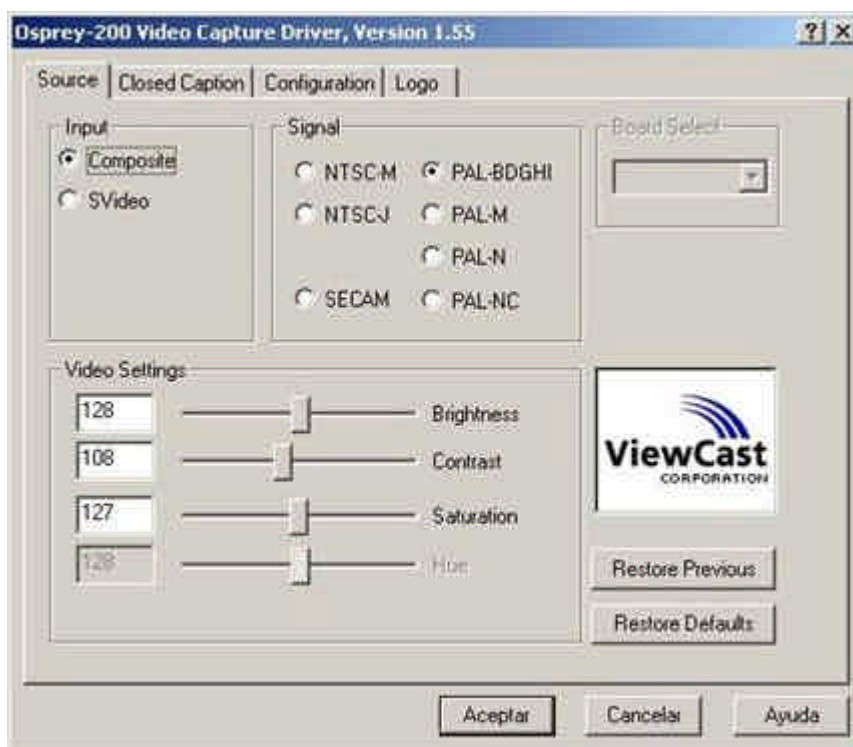
Tras instalar y ejecutar el programa, la pantalla que veremos al abrir por primera vez el programa será la siguiente:



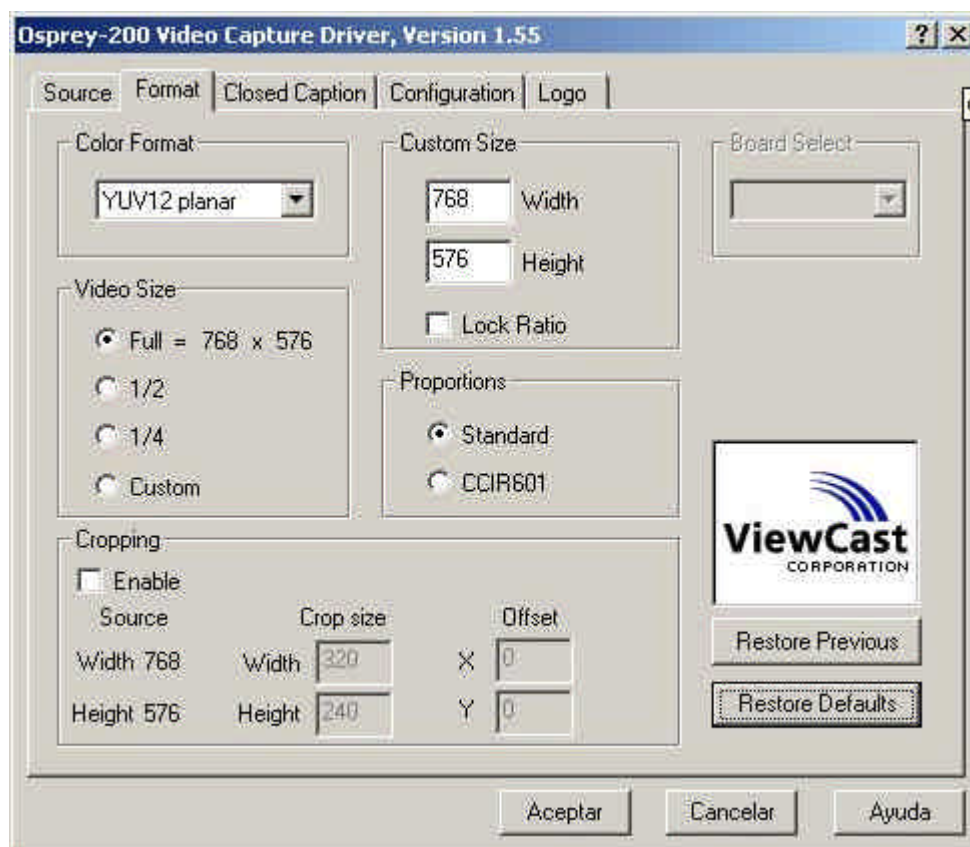
En la parte izquierda se verá el vídeo de entrada que le llega al programa, mientras que en la ventana derecha aparecerá el vídeo de salida que se va a mandar al servidor. La parte inferior de la pantalla servirá para funciones de monitorización de los trabajos realizados. Si el archivo que queremos transmitir se encuentra ya grabado en el ordenador podemos enviarlo seleccionando la opción **Input File** e introduciendo la ruta del archivo. Por el contrario, si deseamos enviar la señal de una cámara es necesario especificar al programa las fuentes de captura de audio y vídeo desde la opción **Devices**.



Una vez hecho esto podemos variar las preferencias de captura de ambas fuentes pulsando en **Settings**. Es importante cambiar, sobre todo, el formato de captura de la fuente de vídeo, pues viene por defecto en NTSC, el sistema americano. Pulsaremos, por tanto en **Settings -> Vfw Video Source** y cambiaremos el sistema a PAL-BDGHI. Si este paso se realizó en la fase de instalación de los drivers de la tarjeta podemos saltárnoslo.

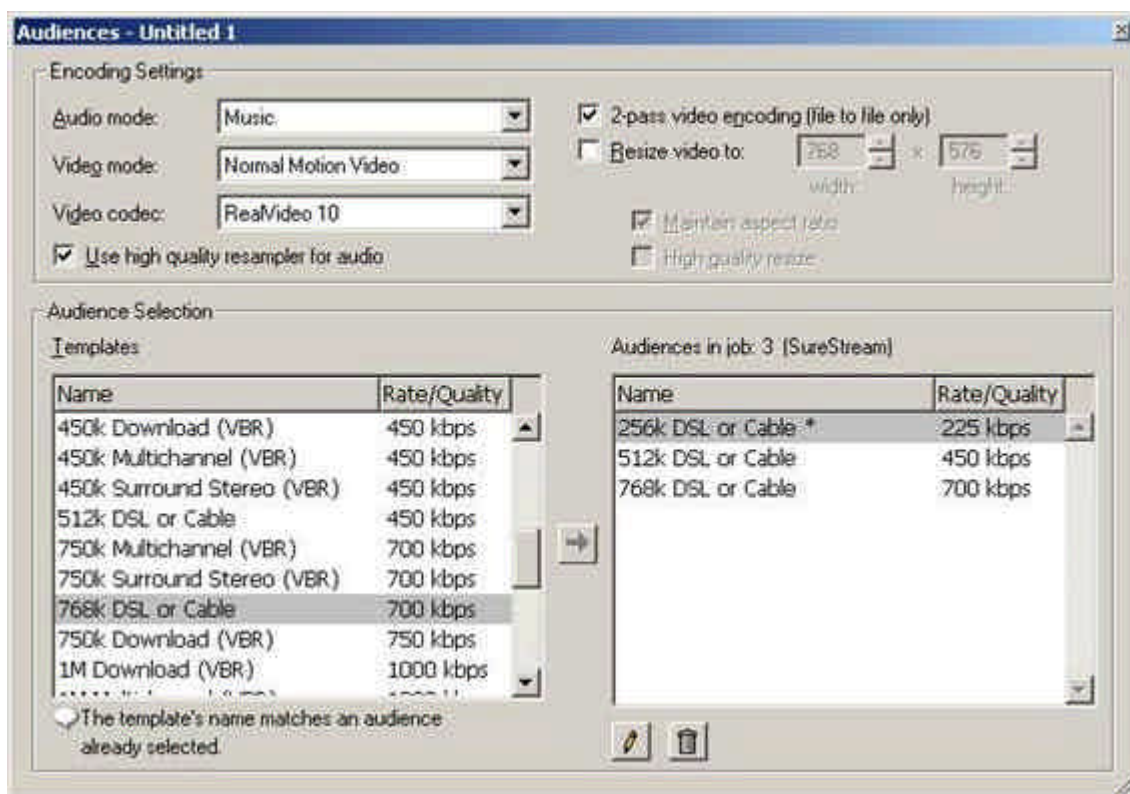


En función del tipo de audiencia que esperemos tener podemos variar también la resolución a la que se va a capturar el vídeo. Una resolución mayor consumirá también mayor ancho de banda, pero revelará detalles más finos de la imagen. Todo dependerá del número de usuarios que esperemos tener, del ancho de banda que tengamos a disposición en el servidor y de la red a través de la cual llegue el vídeo al cliente. Para modificarlo, desde el dispositivo de captura de vídeo iremos a **Settings -> VFW Format Source -> Video Size** .



Al margen de la resolución con la que se capture el vídeo, la velocidad de codificación también afectará al usuario final. Este programa permite utilizar varios bitrates simultáneos de codificación (*SureStream*) para que, en función del ancho de banda que posea el cliente, el vídeo se reproduzca con mejor o peor calidad. Por defecto, el programa codifica a una solo bitrate de 256kbps, pero puede modificarse pulsando **Audiences** en la parte derecha de la pantalla.



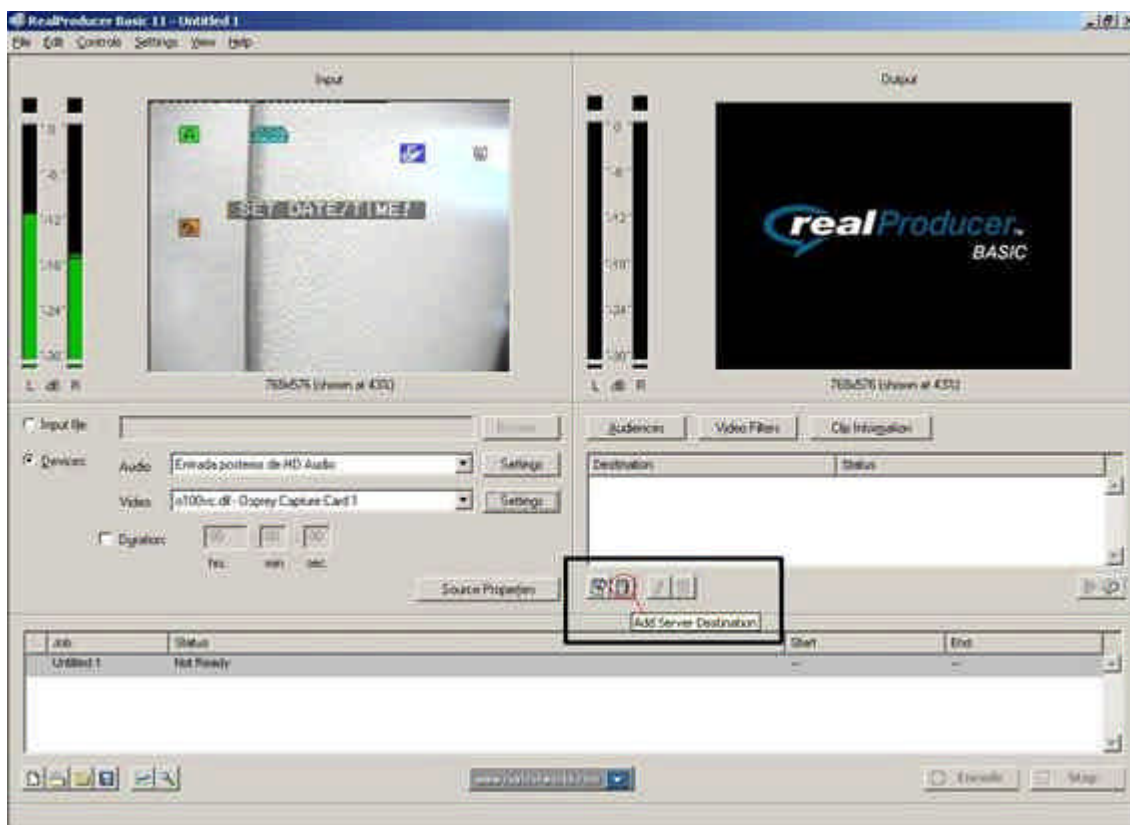


En nuestro ejemplo, el contenido multimedia se codificará a tres bitrates diferentes, 256kbps, 512kbps y 768kbps. El emparejamiento en función del ancho de banda que disponga el cliente lo realiza de forma automática el programa de la siguiente manera:

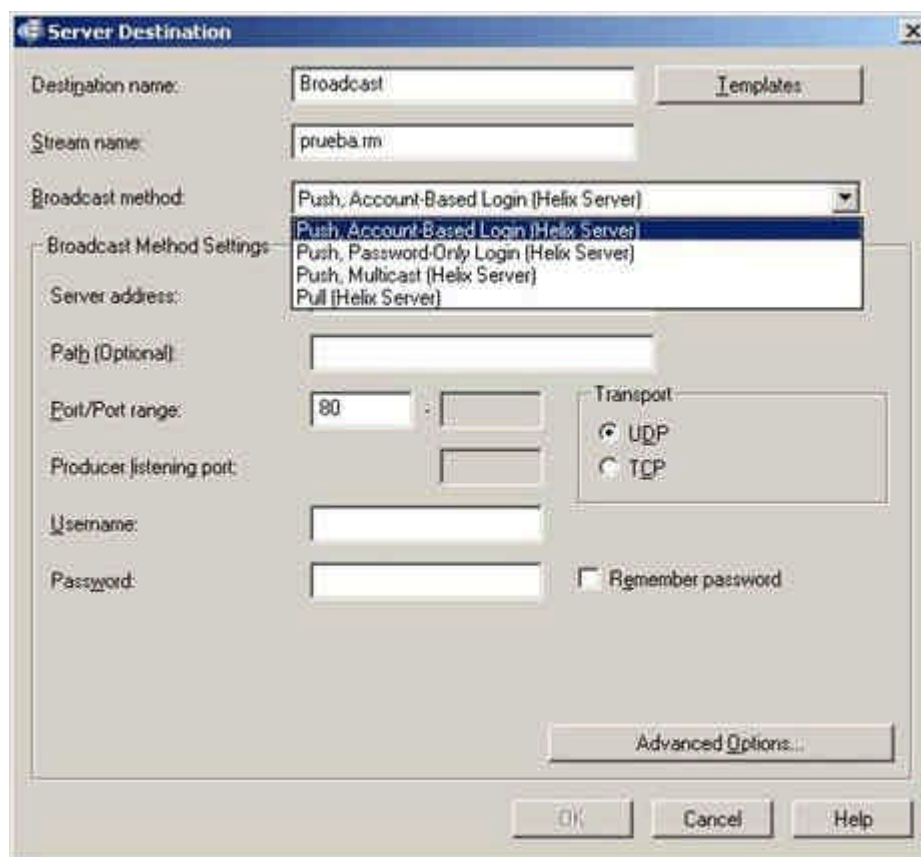
- **0 – 450 kbps**
  - o Ancho de banda de vídeo = 180.9 kbps
  - o Ancho de banda de audio = 44.1 kbps
- **450 – 700 kbps**
  - o Ancho de banda de vídeo = 353.5 kbps
  - o Ancho de banda de audio = 96.5 kbps
- **700 y bitrates superiores**
  - o Ancho de banda de vídeo = 603.5 kbps
  - o Ancho de banda de audio = 96.5 kbps

Una vez se han definido las fuentes de captura y los formatos de codificación del contenido multimedia hay que especificar cómo se realizará la conexión con el servidor de streaming. Para ello pulsamos sobre **Add Server Destination** en la parte inferior derecha de la pantalla, como se indica a continuación:

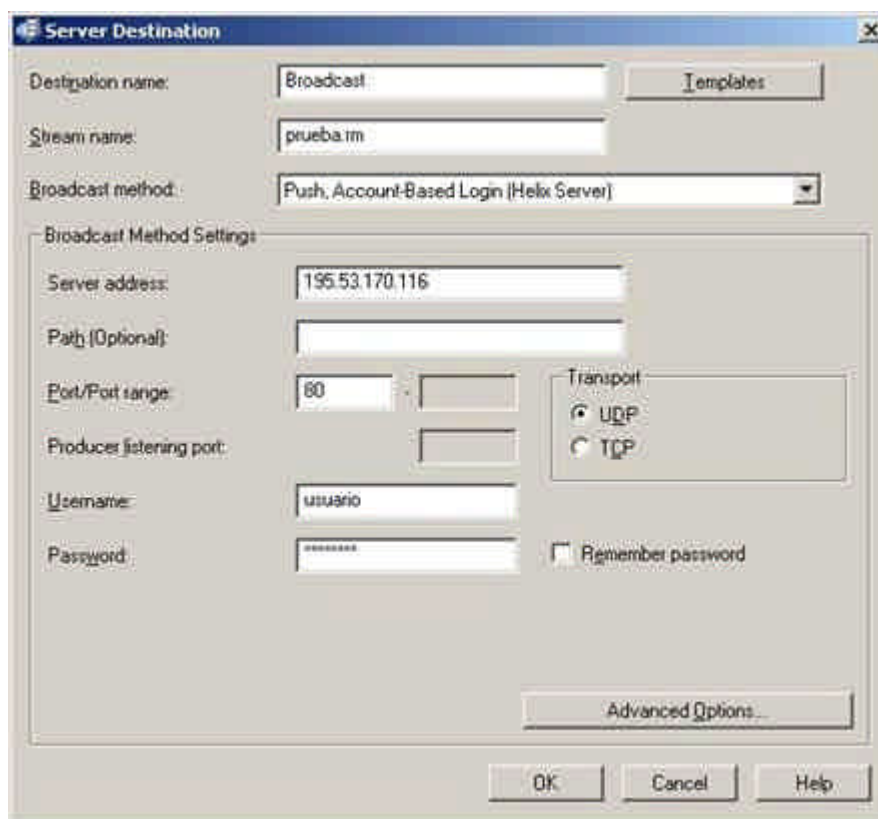




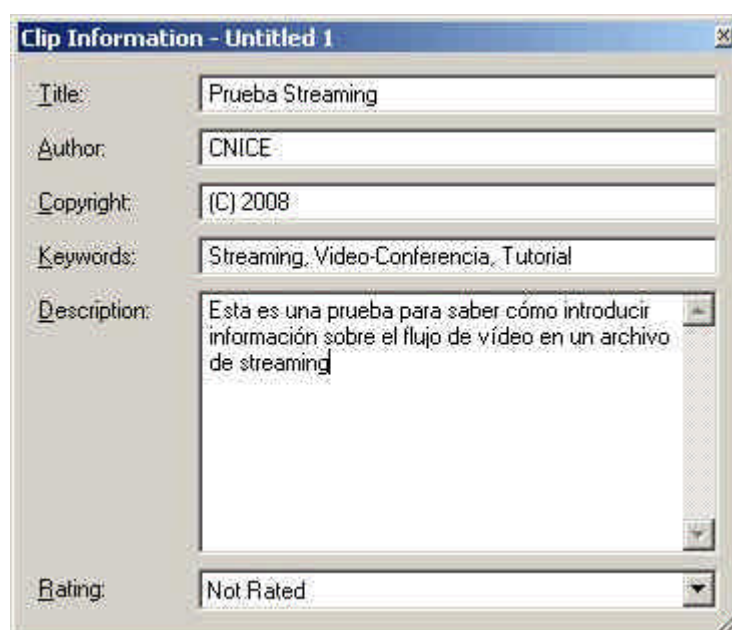
Aparecerá entonces un cuadro de diálogo que permitirá configurar la conexión. En primer lugar se especifica el nombre del trabajo y del stream de video –a nuestra elección, simplemente definirá el nombre del trabajo, así que sería bueno emplear un nombre que identifique de manera unívoca el mismo-, así como el tipo de conexión que vamos a emplear. Existen dos formas bien diferenciadas de hacer broadcast –difusión de vídeo bajo demanda por streaming-. Si se utiliza el método **Push** el flujo de vídeo está siempre enviándose al servidor, aún cuando no hay usuarios que soliciten su visionado. A través del método **Pull** podemos conseguir que no se empiece a codificar el vídeo hasta que no haya un usuario que solicite el contenido. En este tutorial se manejará el método Push, pues permite afrontar en la mayoría de los casos las aplicaciones más importantes de streaming. A través de este método se consigue, si hemos activado la opción de “archivar” en el servidor, que un usuario pueda ver posteriormente un vídeo aunque no hubiera clientes solicitándolo mientras se codificaba.



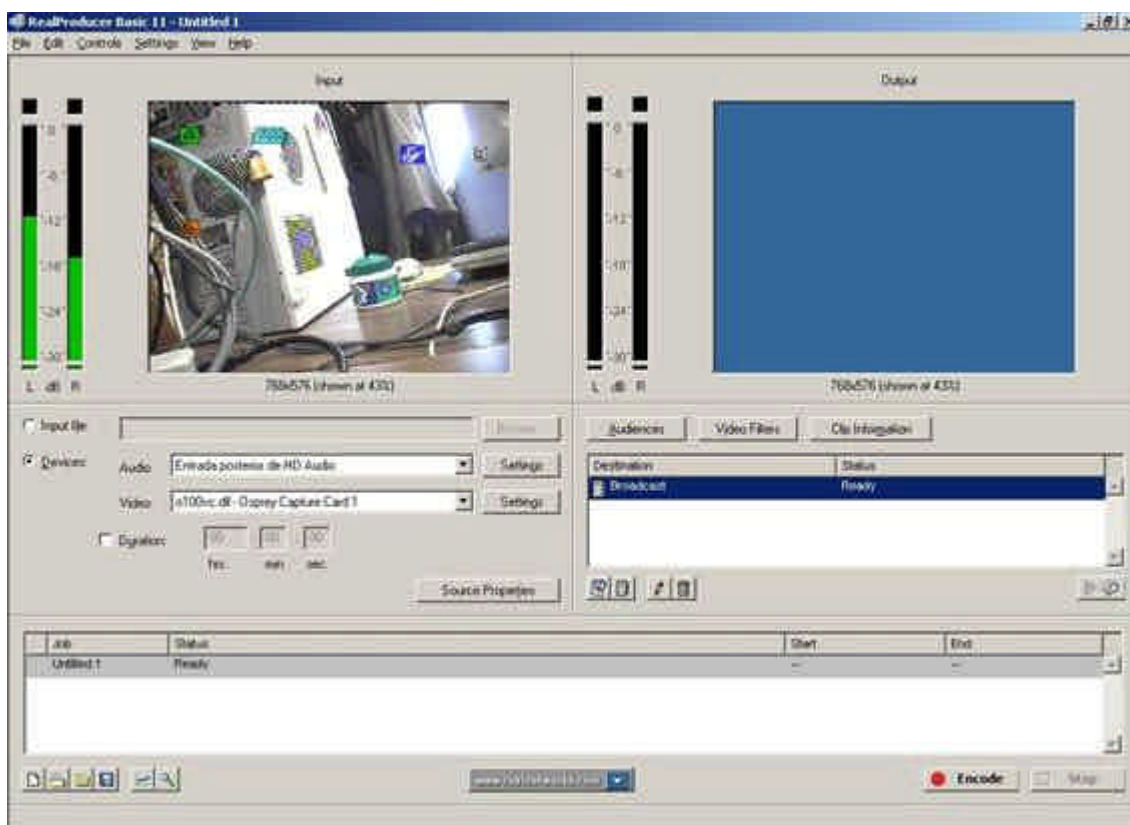
Habremos de indicar, también, la dirección IP que identifica al servidor (típicamente el mismo equipo donde corremos el Real Producer) y el nombre de usuario y contraseña que se introdujeron en el proceso de instalación de Helix Server. El puerto de comunicación (80, por defecto) y el protocolo (UDP) no se modificarán a no ser que se haya hecho lo mismo en el proceso de instalación de Helix Server.



Antes de empezar la codificación podemos, de manera opcional, definir la información básica del clip multimedia que vamos a enviar. Simplemente hay que pinchar sobre **Clip Information**, debajo de la pantalla de vídeo de la derecha.



Así, una vez definidos los formatos de codificación y la comunicación con el servidor, la pantalla del programa mostrará un aspecto similar al siguiente:



Para empezar a codificar sólo tenemos que pulsar sobre **Encode**, en la parte inferior derecha de la pantalla, pinchando en **Stop** cuando hayamos terminado. Automáticamente se enviará la información al servidor Helix tal y como se haya detallado, estando disponible para que los clientes puedan verla con el programa **Real One**.

En caso de precisar más información sobre el Real Producer, o los modos de configuración para hacer *Push Broadcasting* o *Multicast* se recomienda acudir al manual en línea ofrecido por Real Networks.

( <http://service.real.com/help/library/guides/RealProducer10/producer.htm> )

**Real One / Acceso a los contenidos**

### **Requisitos Mínimos**

- Procesador 700 MHz
- 256 MB de RAM
- 60 MB de disco duro
- Tarjeta de sonido de 16-bit y altavoces
- Tarjeta Gráfica Super VGA (1024 x 768, 65k colores)
- Windows XP con Service Pack 2, <st1:place w:st="on">Vista
- IE 5 o posterior, FireFox

## **Requisitos Recomendados**

- Procesador 1 GHz o superior
- 512 MB de RAM, 1 GB con Windows Vista
- Conexión de Internet de Alta Velocidad
- Tarjeta de video/audio Full-Dúplex
- Grabadoras de CD y DVD
- Reproductor de DVD y software
- 4GB de espacio en disco duro
- Windows Media Player 9.0 o posterior
- Quick Time 6.4 o posterior
- Adobe Flash player 8 o posterior

Aunque el usuario puede acceder al streaming a través del propio reproductor de Real Networks –Real Placer / Real One-, la situación típica será el acceso a través de un link en una página Web que abrirá automáticamente el programa.

De forma general, indicamos a continuación las URL típicas de acceso a los contenidos del servidor. Para el streaming en vivo, los contenidos se encuentran bajo el protocolo de Real Media **rtsp**, dentro de la carpeta **broadcast** (**rtsp://direcciónIP/broadcast/nombre\_de\_archivo.rm**). Por ejemplo, para un servidor cuya dirección IP fuera 195.53.170.116 y que estuviera emitiendo un flujo de vídeo de nombre *prueba.rm* la dirección URL sería la siguiente:

***rtsp://195.53.170.116/broadcast/prueba.rm***

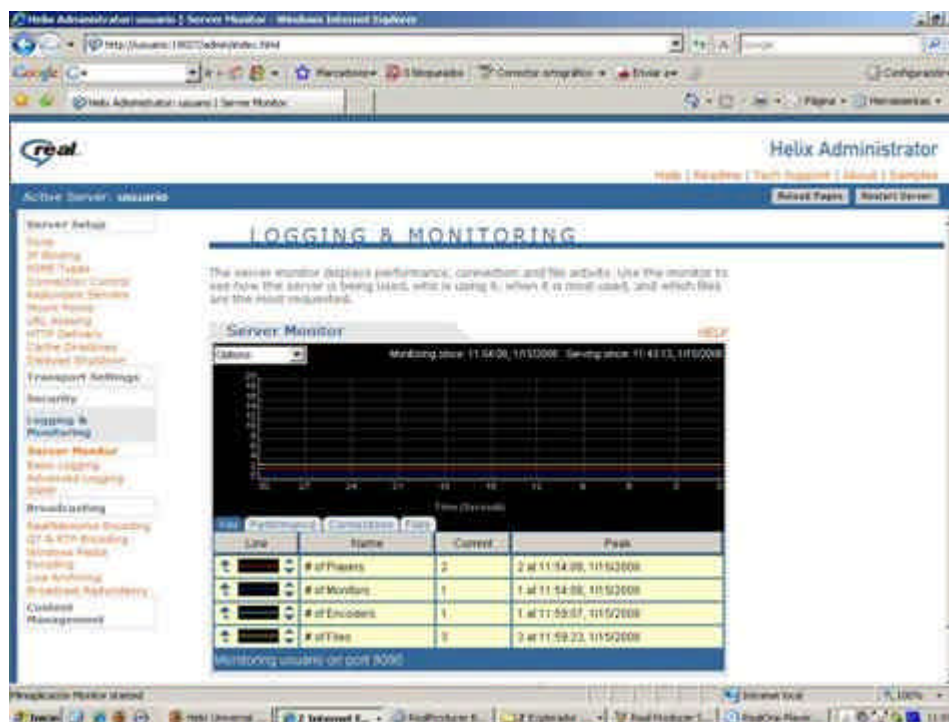
Si se pretende acceder a un contenido archivado la URL es ligeramente diferente. El protocolo de acceso es, en este caso, **http**, y la carpeta donde se almacena el contenido en el servidor por defecto es **ramgen/Archive** ([http://direcciónIP/ramgen/Archive/nombre\\_de\\_archivo.rm](http://direcciónIP/ramgen/Archive/nombre_de_archivo.rm)). De nuevo, para un servidor cuya dirección IP sea 195.53.170.116 y un archivo almacenado de nombre *prueba.rm* la dirección URL de acceso es:

**<http://195.53.170.116/ramgen/Archive/prueba.rm>**

En la pantalla de administración del Helix Server es posible asignar *alias* a las URL, de forma que se pueda acceder a las mismas usando nombres de archivo más cortos, o almacenar los archivos en carpetas distintas a las mencionadas anteriormente.

Antes de pasar al siguiente sistema de transmisión de vídeo en tiempo real a través de la red se procede a realizar algunas consideraciones en cuanto al ancho de banda necesario para ofrecer un servicio de broadcasting a través de unicast, sea cual sea el sistema elegido.

Helix Server propone un sistema de monitorización del servidor en tiempo real a través de su página de administración. Para entrar en él simplemente hay que ir a **Logging <spanclass=Spelle>and Monitoring -> Server Monitor** e introducir el usuario y contraseña de administrador. Aparecerá entonces una pantalla que indicará a lo largo del tiempo, entre otras cosas, el número de clientes conectados, el número de archivos que se están viendo, si hay programas codificando vídeo en directo y el ancho de banda y memoria consumidos, como aparece en la siguiente figura:



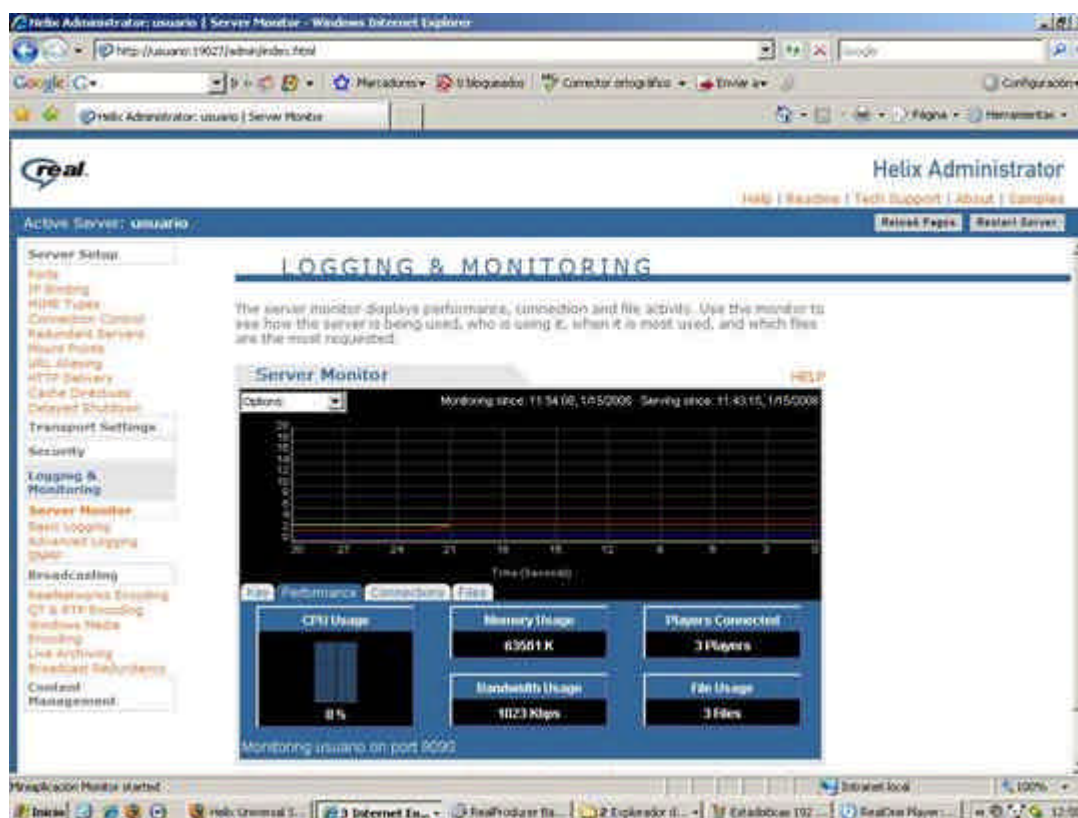
Es importante conocer el ancho de banda que posee el servidor, pues influirá en el número de clientes y codificadores con los que podremos trabajar simultáneamente. Manejar contenido multimedia –vídeo, especialmente-, consume mucho ancho de banda, por lo que hemos de estar atentos a este parámetro. Es responsabilidad del administrador del servidor y de la persona que codifica el evento el establecimiento de las reglas de juego en la comunicación entre cliente y servidor.

Por ejemplo, si estamos emitiendo en tiempo real un evento a una resolución de 768x576 píxeles y 768kbps y se conectan dos clientes al servidor a 700kbps para ver el evento, el ancho de banda consumido en el servidor ronda los 2000kbps. Es fácil extrapolar el resultado cuando el número de clientes aumenta.





El ancho de banda es un recurso limitado, así que hay que explorar las necesidades del vídeo que queremos transmitir. Si en el vídeo no hay mucho movimiento, es mejor codificar con un bitrate medio-bajo, puesto que la calidad de vídeo seguirá siendo aceptable y en el mismo ancho de banda se podrán conectar un mayor número de clientes. Por el contrario, si el movimiento y los detalles finos de la imagen son lo importante, será necesario codificar el archivo a velocidades más altas, aunque en una conexión lenta el movimiento puede resentirse. Para una sesión de videoconferencia donde la cámara está generalmente fija o su movimiento es irrelevante para la información que queremos captar, si el codificador trabaja a 512kbps y los dos clientes conectados lo hacen a velocidades inferiores a 450kbps (referencia en la página 14), el ancho de banda necesario disminuye hasta poco más de 1000kbps, prácticamente la mitad que en el caso anterior. En la siguiente captura de pantalla puede verse este último caso, y también la memoria consumida en el servidor al administrar el flujo de datos, similar a la del ejemplo anterior:



Una práctica interesante cuando el ancho de banda es limitado y el vídeo no es una característica importante en la transmisión es codificar éste a un bitrate bajo y dejar el audio con una calidad superior. El audio consume mucho menos recursos que el vídeo, pero una bajada de calidad en este aspecto es mucho más incómoda para el usuario final.

## Microsoft Windows Media

Al igual que en la solución aportada por Real Media, son necesarios tres programas para el streaming de vídeo utilizando Windows Media: un codificador de vídeo, un servidor y un reproductor, el que utilizará el cliente para la recepción del contenido. La ventaja de esta solución sobre la de Real Media es, sobre todo, funcional: el reproductor que ha de utilizar el cliente viene de serie instalada en cualquier edición de Windows, por lo que éste no tiene que descargar ningún programa. El servidor, sin embargo, debe tener instalada la versión **Enterprise** del **Windows Media Server 2003** –en su última versión-. Se aconseja esta versión en lugar de la **Standard Edition** porque esta última sólo soporta unicast, mientras que la **Enterprise Edition** se maneja sin dificultad también con multicast.

Antes de continuar hemos de analizar las diferencias con la solución anterior. En la tecnología *Real Media* se utiliza para la administración del servidor una página Web, mientras que la solución *Windows Media* analizada utiliza un sistema operativo que actúa como servidor –Windows Media Server-. Si bien la solución basada en Web es más asequible y puede ejecutarse en múltiples sistemas operativos y equipos, la opción a través de un sistema operativo dedicado permite realizar configuraciones más avanzadas y controlar el tráfico de forma más interesante. Aún así, ambas soluciones son válidas en la mayoría de los escenarios.

### **Windows Server 2003 Enterprise Edition / Administración de Contenidos**

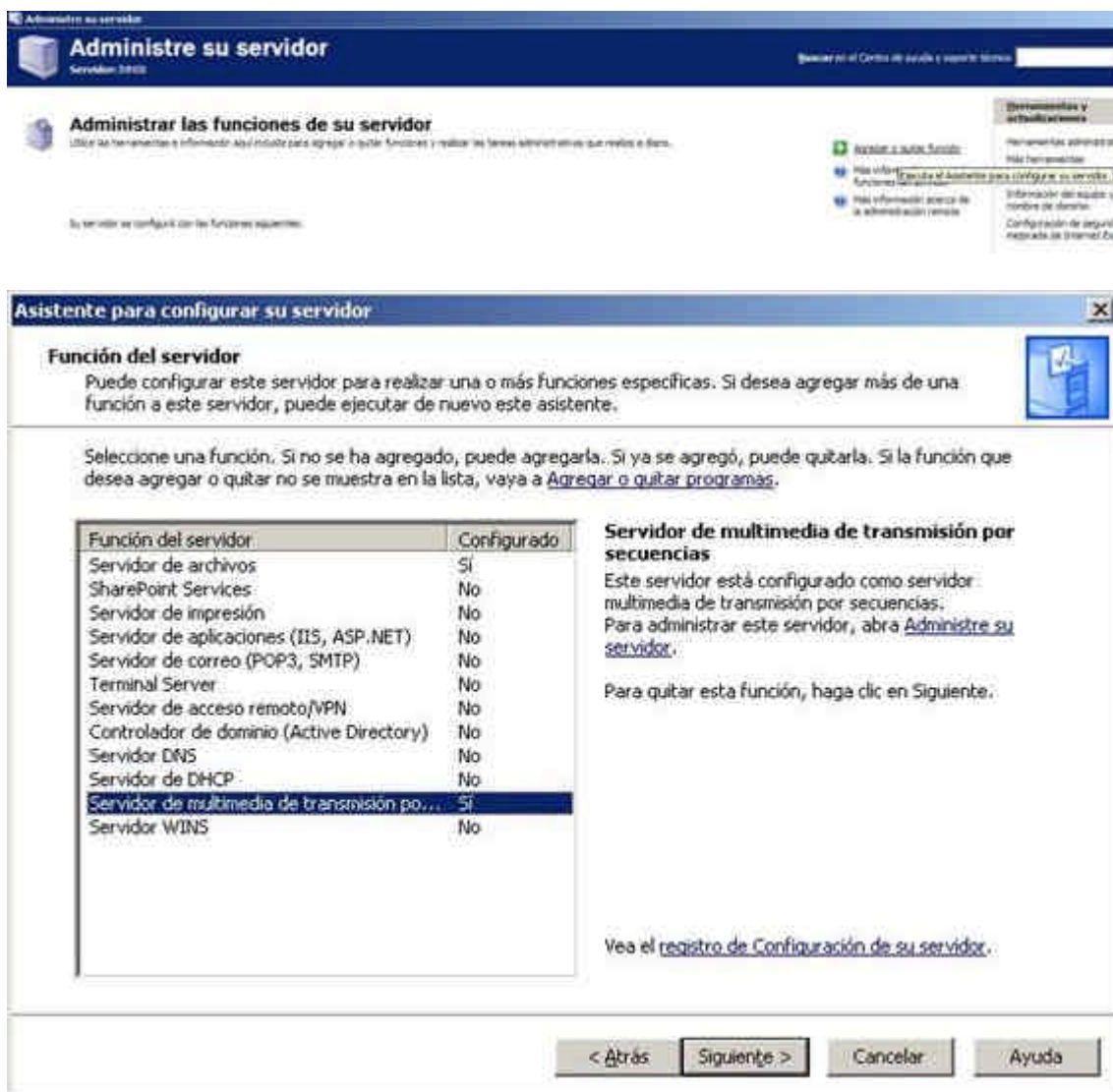
#### **Requisitos Mínimos**

- Procesador 133 MHz
- 128 MB de RAM
- 1.2 GB de espacio libre en disco duro

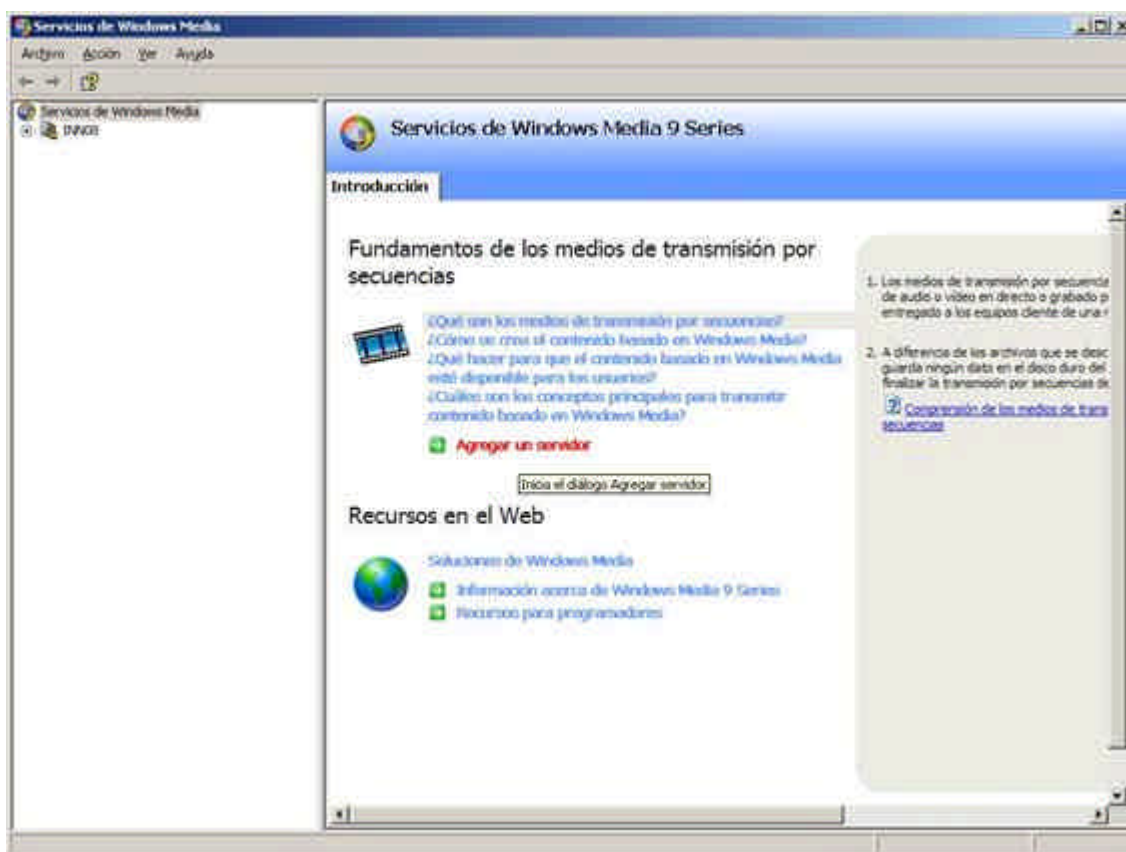
#### **Requisitos Recomendados**

- Procesador 733 MHz o superior
- 256 MB – 64 GB de RAM
- Soporte hasta 8 procesadores simultáneos
- 2.9 GB de espacio libre en disco duro

Una vez instalado el **Windows Media Server 2003 Enterprise Edition** aparece un asistente que nos permite definir que tipo de servidor queremos crear. Puesto que pretendemos realizar streaming, pulsaremos sobre **Agregar o quitar función** en la pestaña **Administrar las funciones de su servidor** e indicaremos que queremos que nuestro servidor sea un **servidor de multimedia de transmisión por secuencias**, y seguiremos el asistente de configuración, como indican las figuras siguientes:



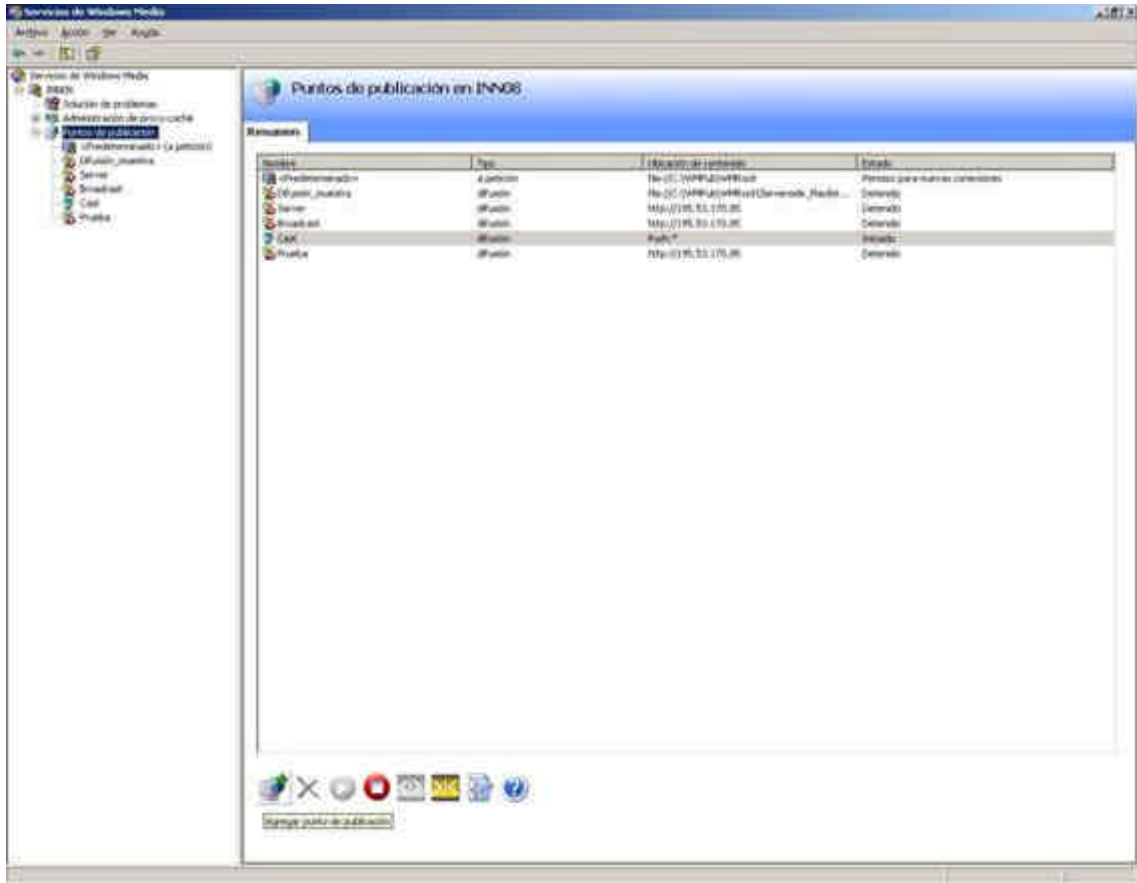
Para añadir un servidor, si no se ha hecho desde el asistente de configuración anterior, hemos de ir a la opción **Agregar un servidor**, como indica la siguiente figura:



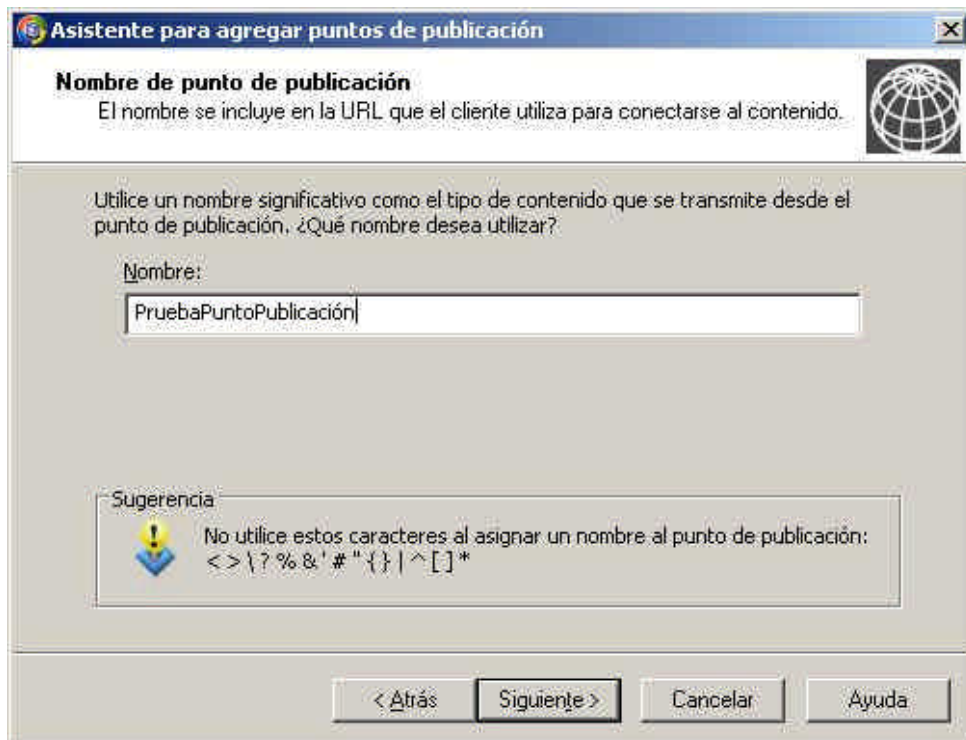
Nos pedirá entonces el nombre la dirección IP donde se encuentra el servidor –en este caso la del equipo- y creará un acceso en la parte izquierda de la pantalla que apuntará a éste.



Para poder emitir un evento dentro del servidor ha de existir un punto de publicación que esté configurado para responder ante los clientes que quieran acceder al flujo de datos. Para crear uno pulsamos sobre **Agregar Punto de Publicación**, en la parte inferior de la pantalla cuando en el menú de la izquierda se selecciona *Puntos de Publicación*.



Se inicia así un asistente que nos permitirá crear un punto de publicación. En primer lugar, se nos pide que pongamos el nombre del punto de publicación. Hemos de recordar este nombre para pasárselo al codificador y, posteriormente, al cliente.





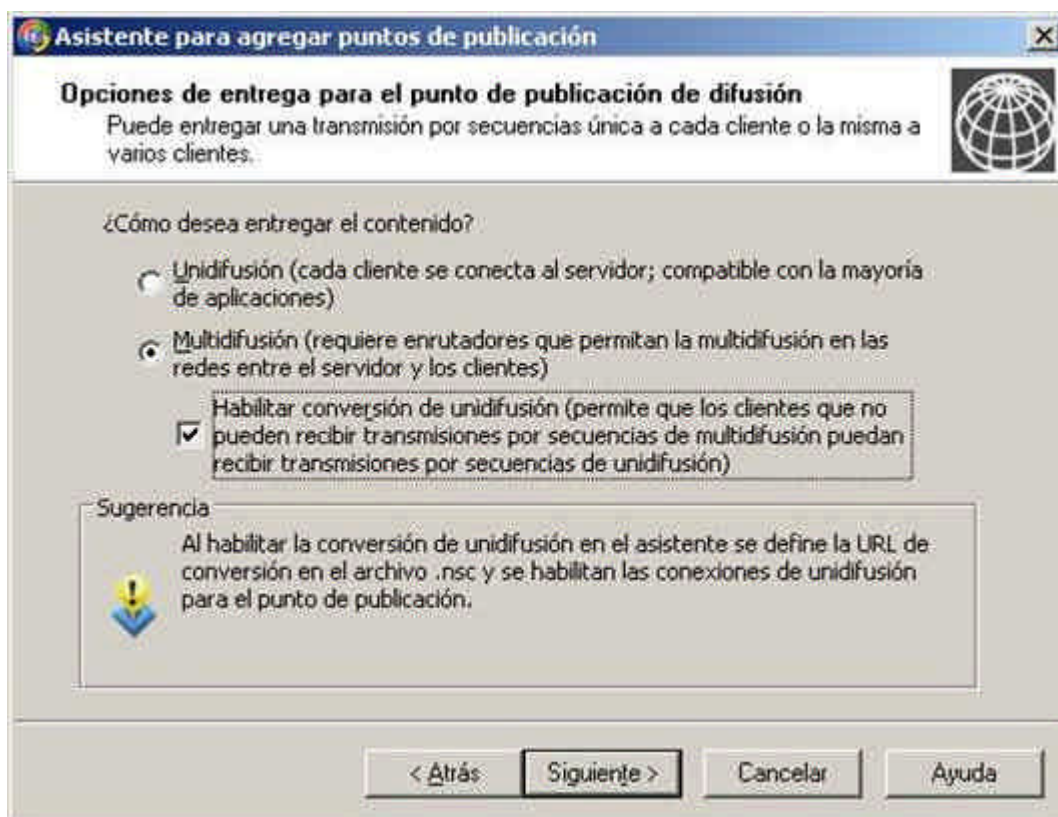
Si continuamos el asistente, el siguiente cuadro de diálogo nos permitirá definir el tipo de contenido que vamos a emitir. Puesto que en el caso propuesto transmitimos desde la cámara de vídeo pulsamos sobre la opción **Codificador (una transmisión por secuencias en directo)** y hacemos clic en **Siguiente**.



La siguiente pantalla define si será el servidor el que controle la reproducción o será el propio cliente el que podrá avanzar o retroceder la visualización a su antojo. Si la emisión es en directo el cliente no tiene acceso a este control, así que el asistente no nos deja cambiar nada. Pulsamos **Siguiente**.

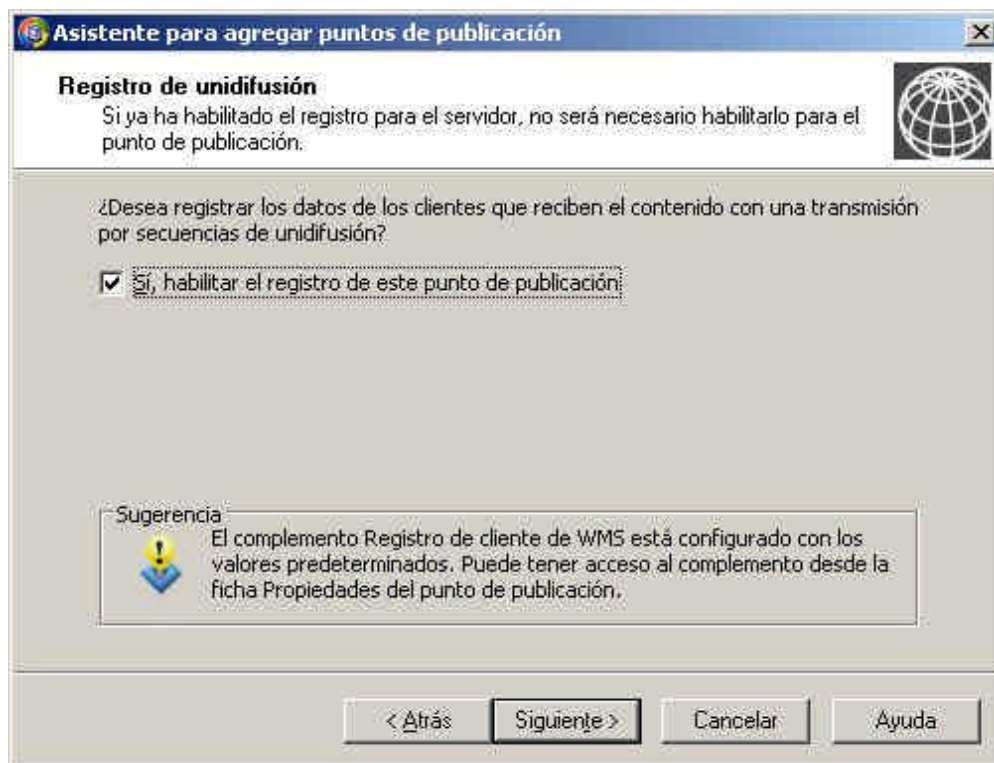


En el siguiente cuadro de diálogo podemos establecer si queremos hacer Unicast o Multicast, si la red lo permite. La opción más recomendable para ahorrar ancho de banda es seleccionar la opción de multidifusión con apoyo de unicast si la red no permite multicast, como aparece en la siguiente captura de pantalla:

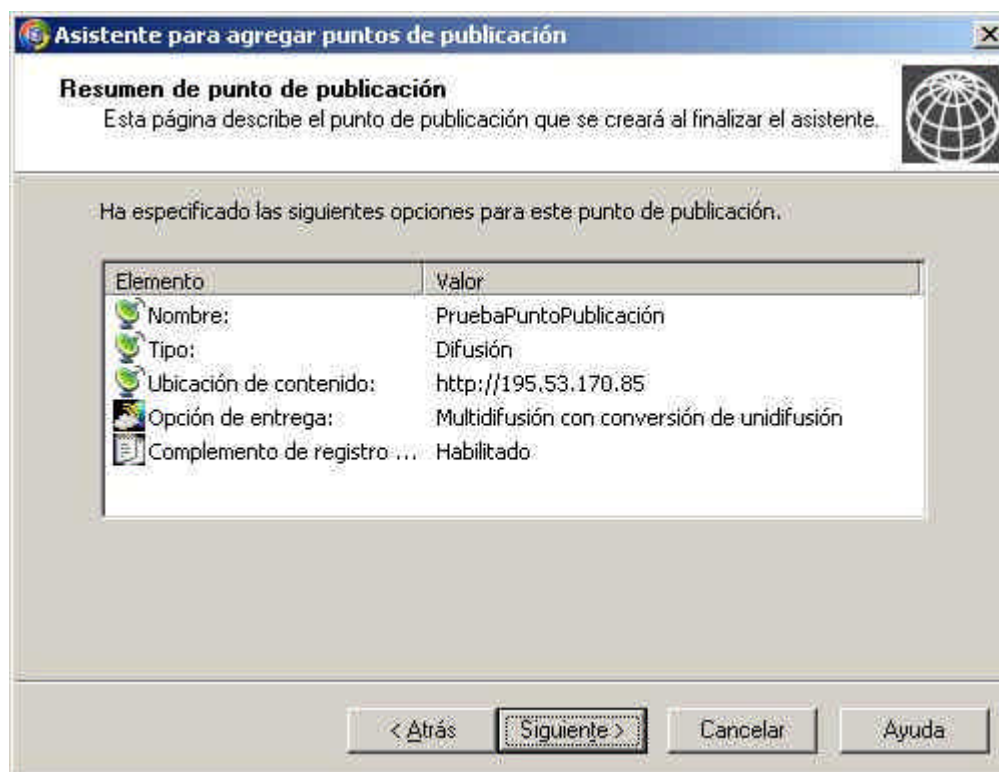


El paso que sigue consiste en especificar desde dónde se va a capturar la secuencia. Si se trata de un archivo en el servidor especificaremos la ruta al mismo, y si se trata de un codificador indicaremos su dirección. Cuando hablemos del codificador indicaremos un método para que sea el mismo programa el que gestione los puntos de publicación, con lo que la gestión es aún más fácil si definimos el punto de publicación como **Push**:\*.

En la siguiente pantalla del asistente se puede activar el registro de los datos de los clientes que accedan al streaming, para monitorizar el ancho de banda consumido y restringir el acceso a determinados clientes. Si queremos hacerlo, activamos la casilla y pulsamos **Siguiente**.

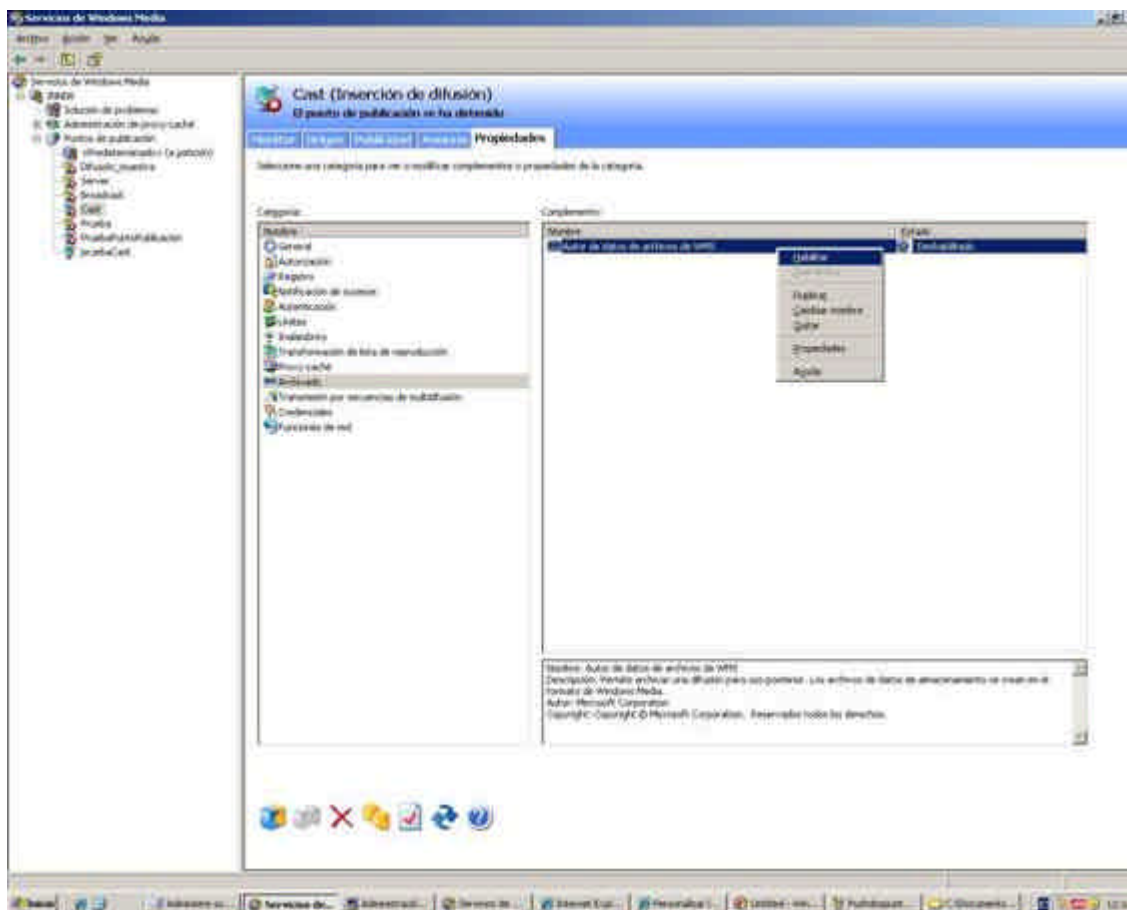


Por último, una pantalla resume la configuración que acabamos de realizar, pudiendo comprobar que es correcta:

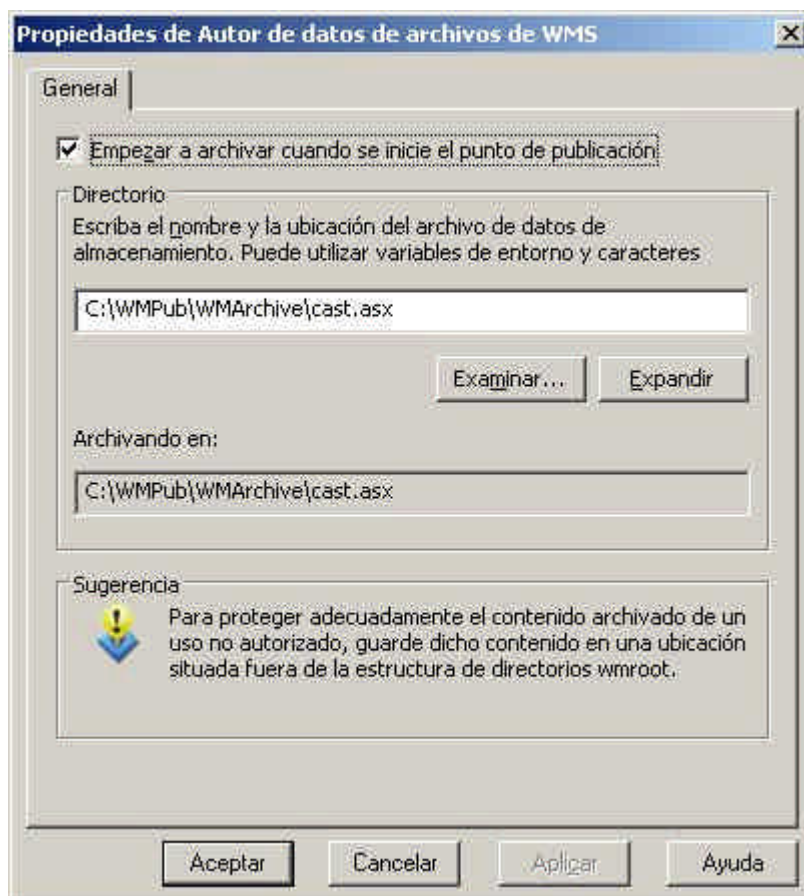
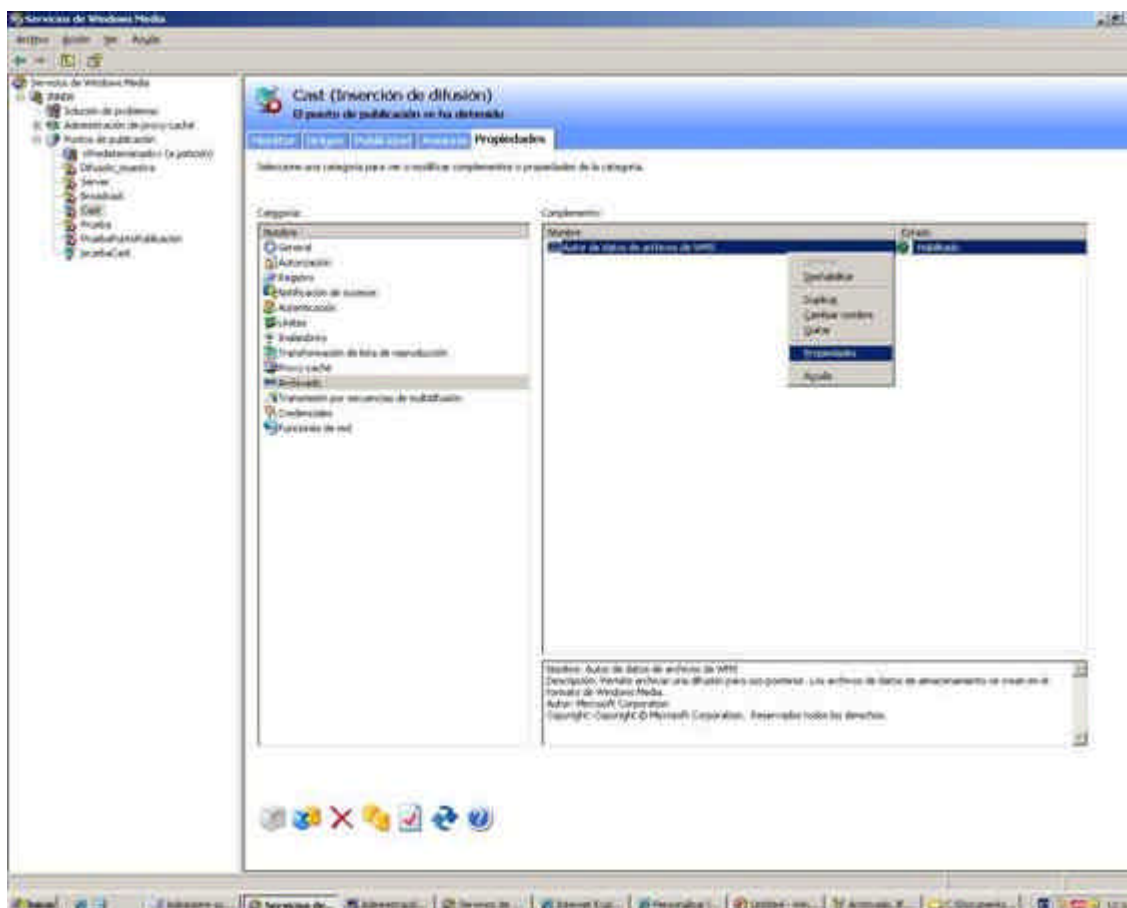


El asistente termina, y nos pregunta si queremos crear un anuncio para el contenido unicast, de forma que se cree un archivo **.asx** que redirija al Reproductor de Windows Media hasta el contenido del punto de publicación. Podemos realizar este paso desde aquí, o desmarcar la opción y activarla posteriormente como se indica a continuación, a

través de la opción **Nombre\_Servidor** **Puntos de Publicación** **Nombre\_punto\_publicación** **Archivado** **Autor de Datos de Archivos WMS** **Habilitar**. Pulsando con el botón derecho sobre esta última opción y haciendo clic en **Propiedades** se puede especificar en qué directorio y qué nombre de archivo tendrá el contenido que se guardará. Así, cuando el cliente acceda al punto de publicación si se está emitiendo en directo recibirá la señal en directo, pero podrá recibir la última emisión archivada si no se está codificando y emitiendo señal. Las siguientes imágenes clarifican el proceso de activación del archivado y el cuadro de diálogo de propiedades que permite especificar la ruta del mismo:

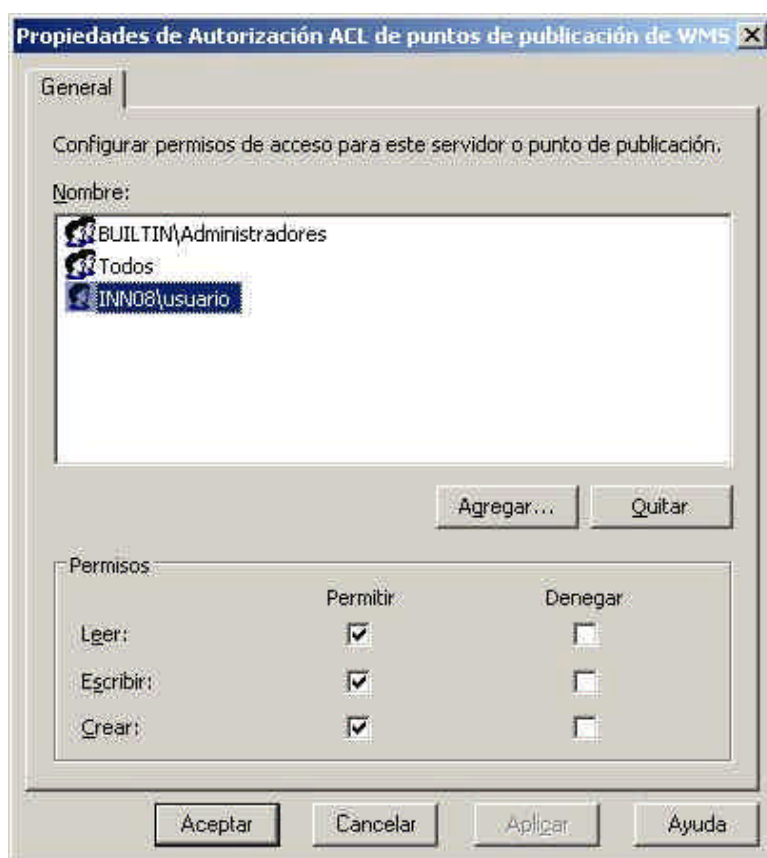
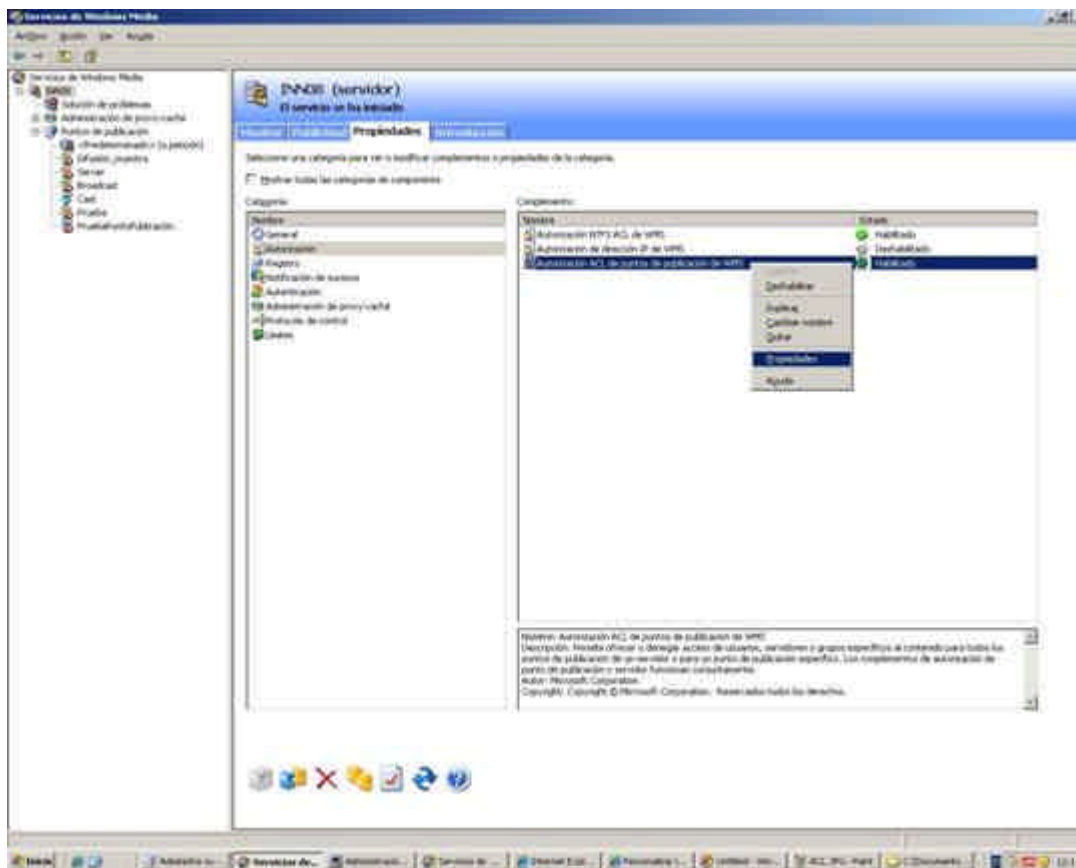




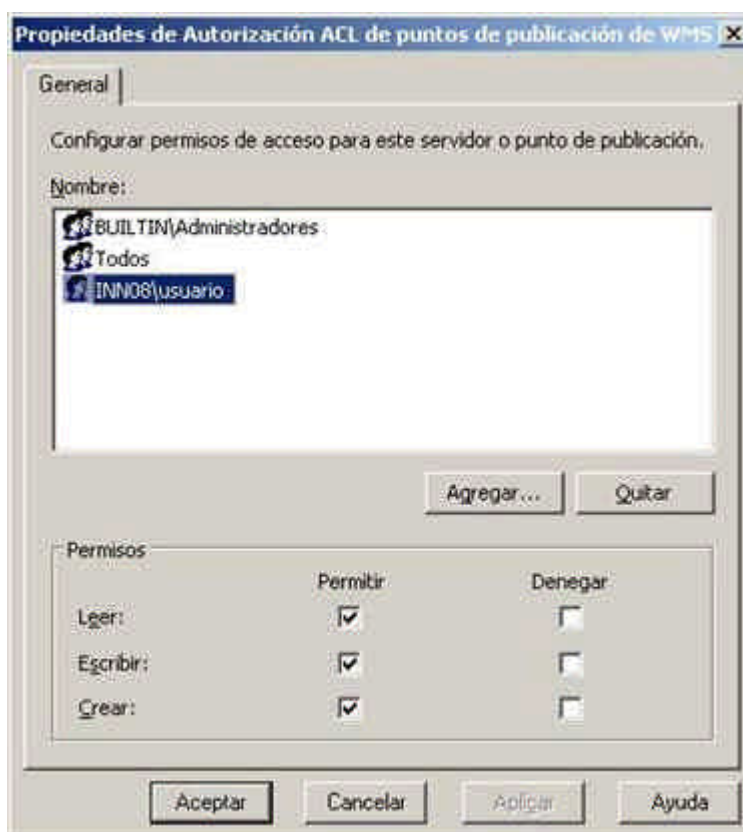
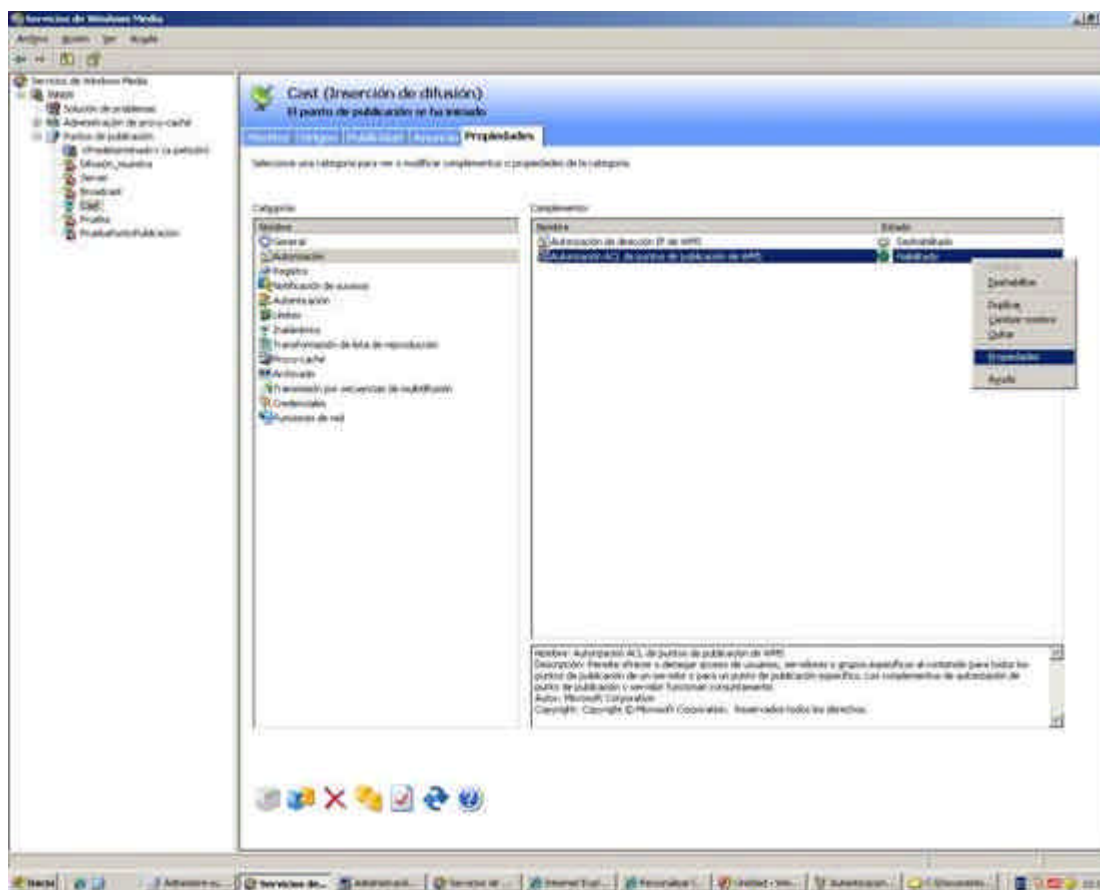








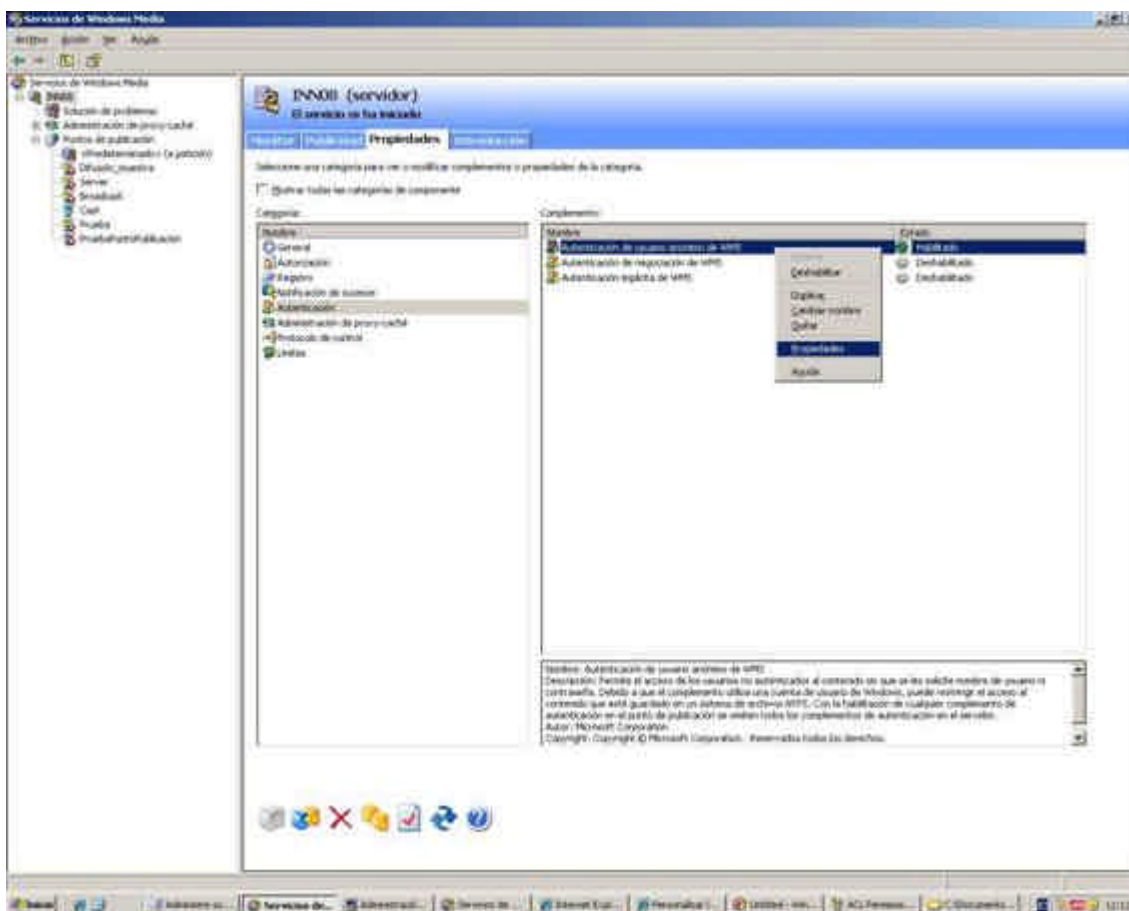
**Configuración de Permisos a nivel de Servidor**

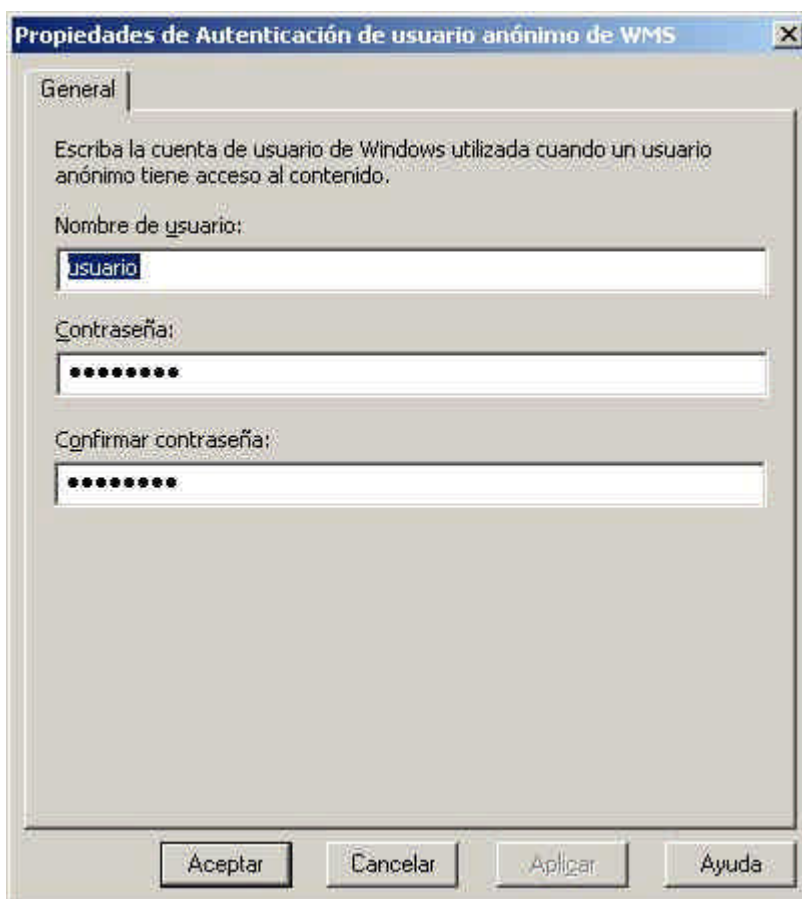


*Configuración de Permisos a nivel de Punto de Publicación*

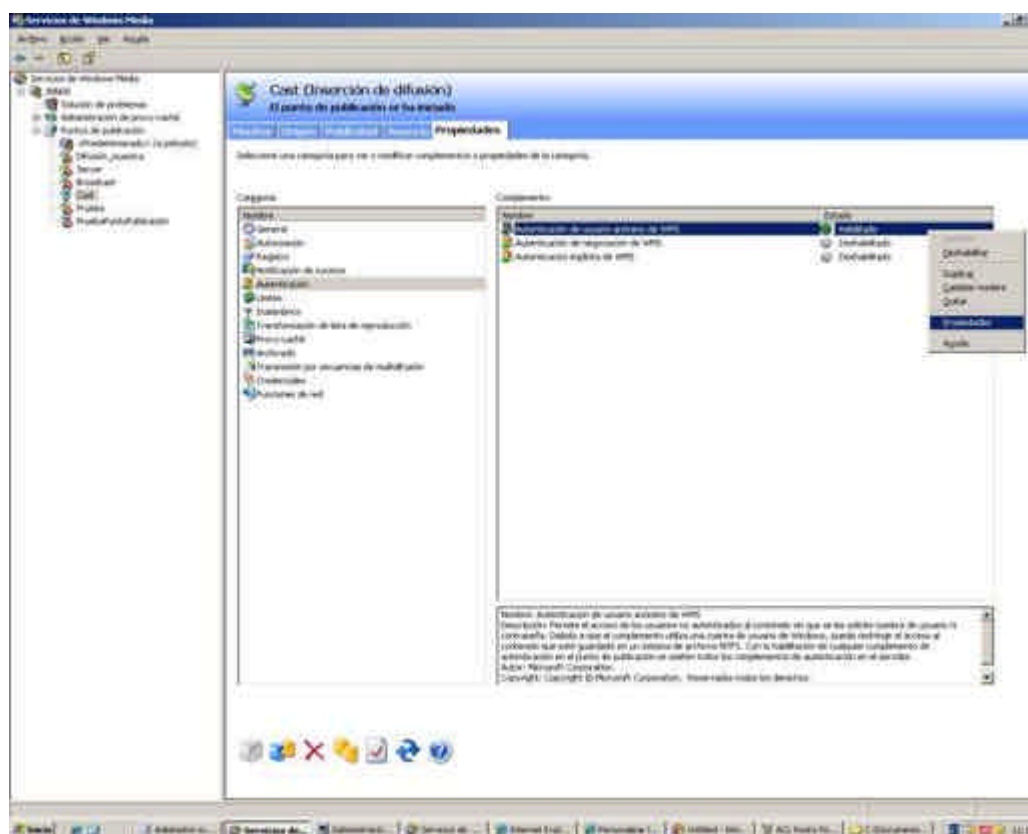
Puesto que cuando el cliente acceda al contenido no lo hará a través de la cuenta de administración, sino típicamente a través de la de usuarios anónimos, habrá que establecer permisos de autenticación para que éstos puedan acceder al punto de publicación. Si no se realiza este paso el codificador enviará el flujo de datos, pero cualquier cliente externo al servidor no podrá verlo. Al igual que en el paso anterior, hay que repetir el siguiente proceso a nivel de servidor y de punto de publicación, si éste último ya está creado.

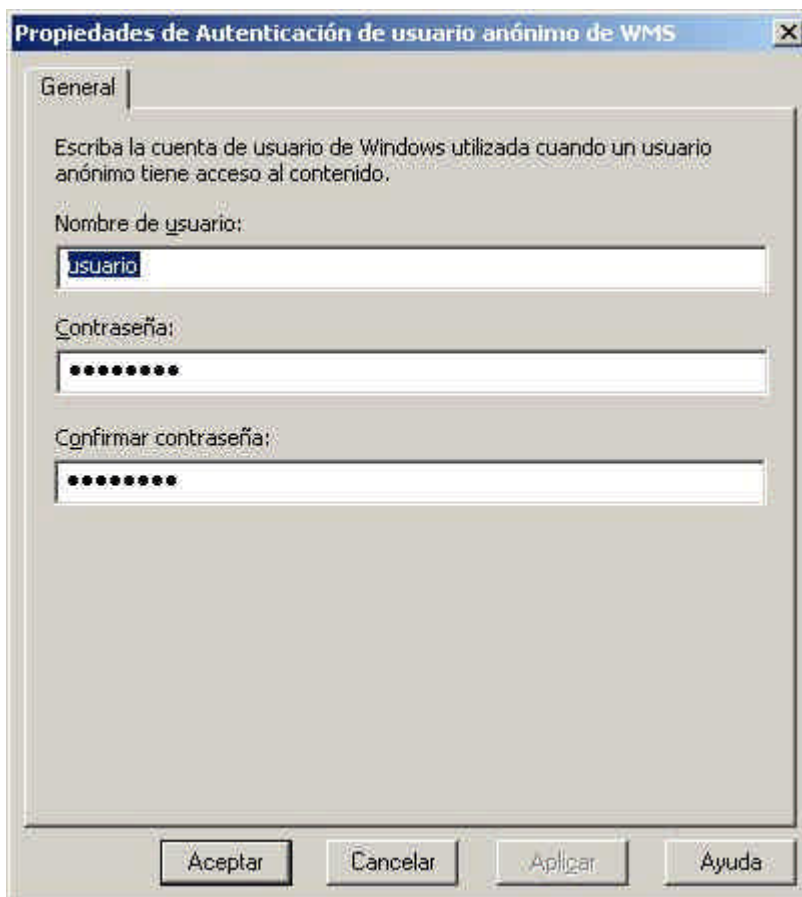
Hacemos clic sobre **Nombre\_Servidor à Autenticación à Autenticación de Usuario Anónimo de WMS à Habilitar**, y pulsamos sobre **Propiedades** desde el menú contextual abierto mediante el botón derecho del ratón. En el cuadro de diálogo siguiente, escribimos el nombre de usuario de la cuenta de usuarios anónimos que tengamos configurada en *Windows Media Server 2003* –en este caso *usuario*- y la contraseña que utilizan los usuarios de esta cuenta para acceder.





*Autenticación a nivel de Servidor*





### *Autenticación a nivel de Punto de Publicación*

Una vez se han realizado todos estos pasos previos podemos comenzar la codificación y envío del contenido multimedia.

## **Windows Media Encoder / Codificación de Contenidos**

### **Requisitos Mínimos**

- Procesador 300 MHz
- 64 MB de RAM
- Tarjeta de sonido compatible con Windows
- Tarjeta Gráfica compatible con Windows
- Windows Vista, XP, 2000 o Server 2003
- Conexión a Internet

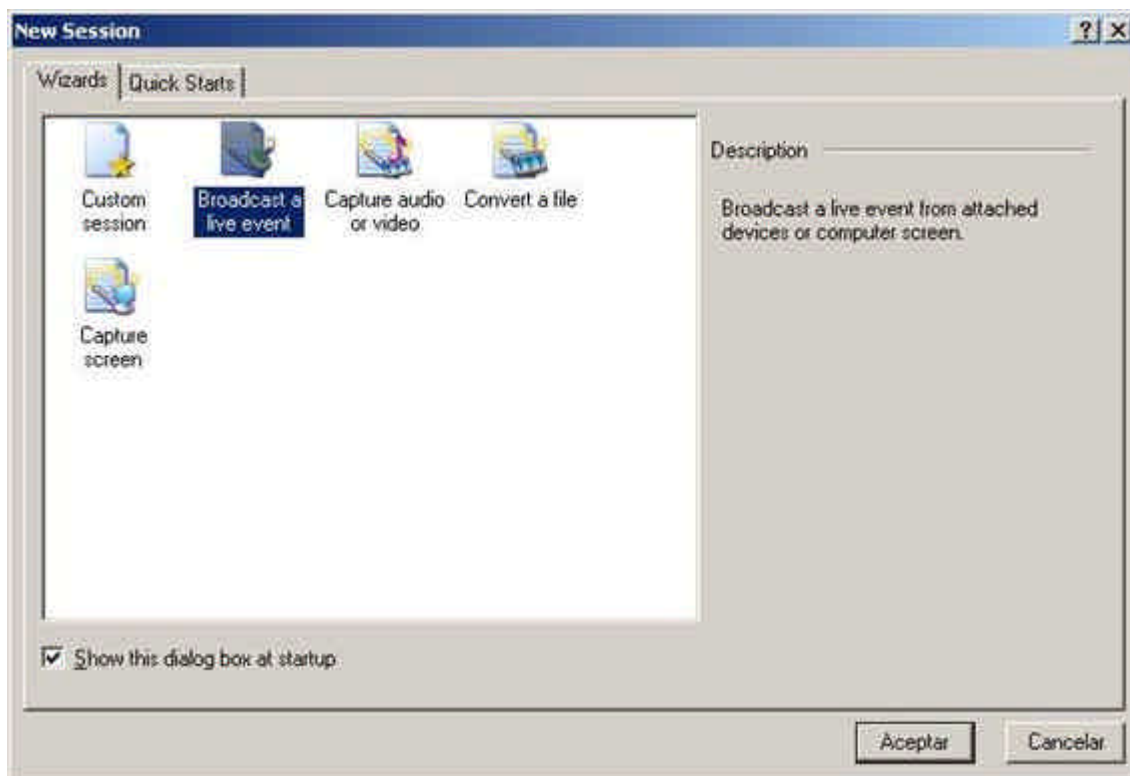
### **Requisitos Recomendados**

- Procesador 2 GHz o superior
- 256 MB de RAM
- Conexión de Internet de Alta Velocidad
- Tarjeta de video/audio Full-Dúplex

El programa productor que codificará la señal para enviarla a un servidor –remoto o no– se llama **Windows Media Encoder**, y puede descargarse en inglés a través del link

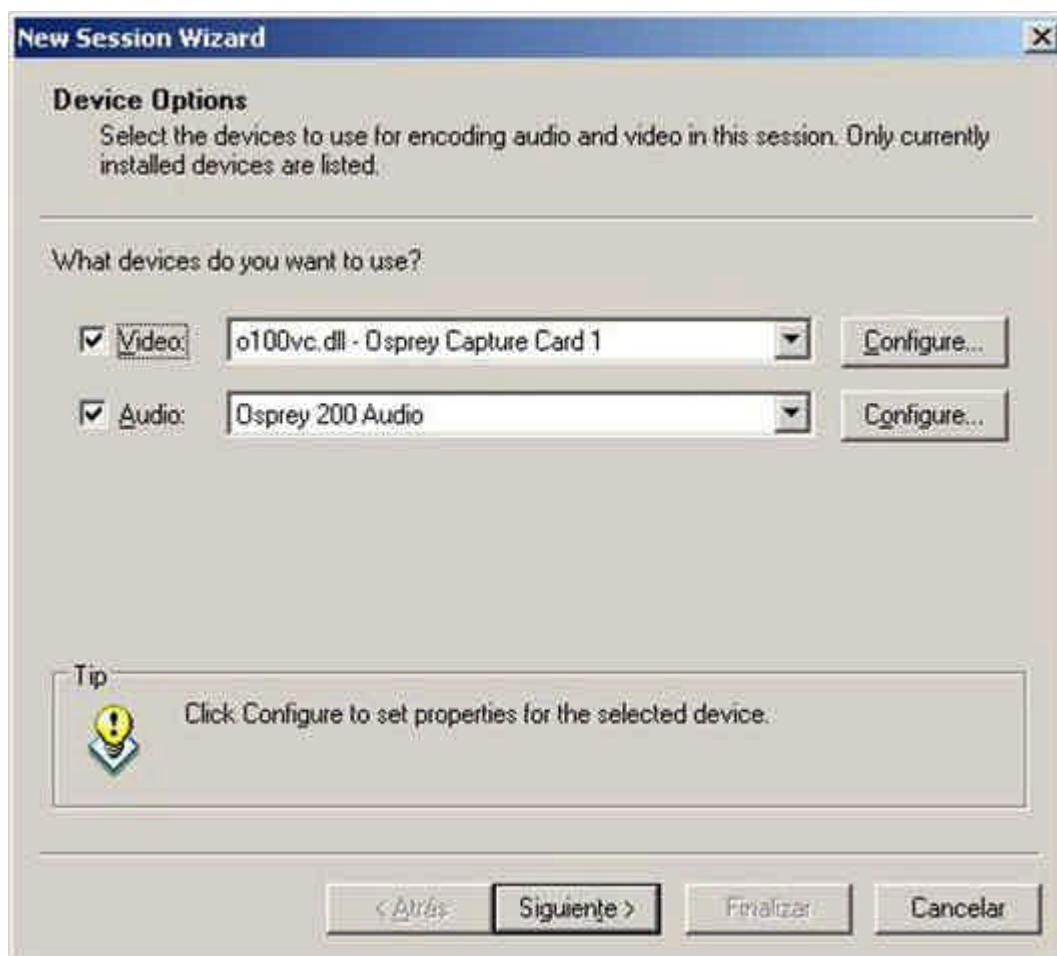
<http://www.microsoft.com/windows/windowsmedia/es/9series/encoder/default.aspx>.

Dicho programa permite la emisión en broadcast de un archivo multimedia o del flujo de datos de un dispositivo capturador utilizando un sencillo asistente de configuración. Pulsamos **New Session** –si el asistente no se ha abierto ya- y hacemos clic sobre **Broadcast a Live Event** para iniciar el asistente de configuración de una sesión de *streaming*.



En la siguiente pantalla del asistente podemos seleccionar los dispositivos de captura y configurar la resolución y el formato al que se realizará la misma. Los seleccionamos y pulsamos **Siguiente**.





Ahora podemos seleccionar el método de broadcast que queremos utilizar. Recordemos lo que indicábamos al hablar de la solución de Real Media sobre los métodos **Push** y **Pull**. Si se activa el método **Push**, el contenido se envía al servidor desde que el codificador inicia el streaming, mientras que en el método **Pull** es el cliente el que inicia la conexión cuando solicita acceso al contenido. Una opción sencilla para entornos reducidos cuando no queremos instalar el **Windows Media Server 2003** y deseamos que un determinado número de clientes accedan –por unicast- al contenido es dejar esta opción marcada como **Pull** y continuar el asistente en el siguiente punto. Este sistema funciona en otros sistemas operativos diferentes al **Windows Media Server**, pero no permite administrar de manera avanzada los contenidos ni las conexiones que se realizan contra el archivo. Si se prefiere utilizar el método **Push** se recomienda acudir a la descripción del asistente de configuración, el punto 2 de la página 43.

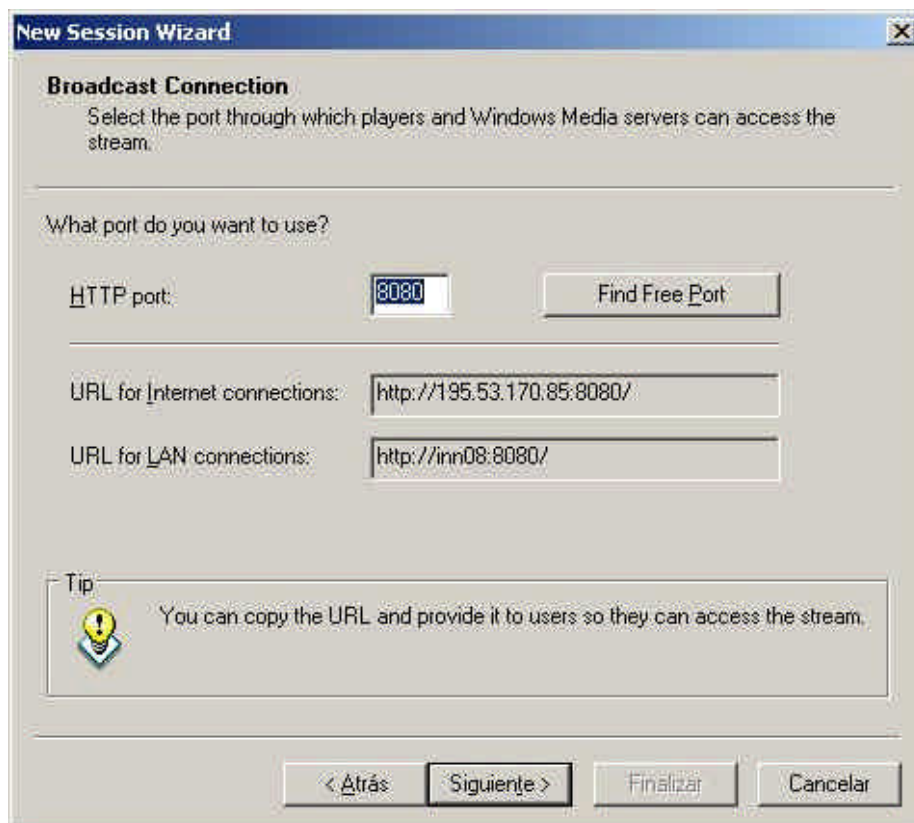
## 1. Método Pull



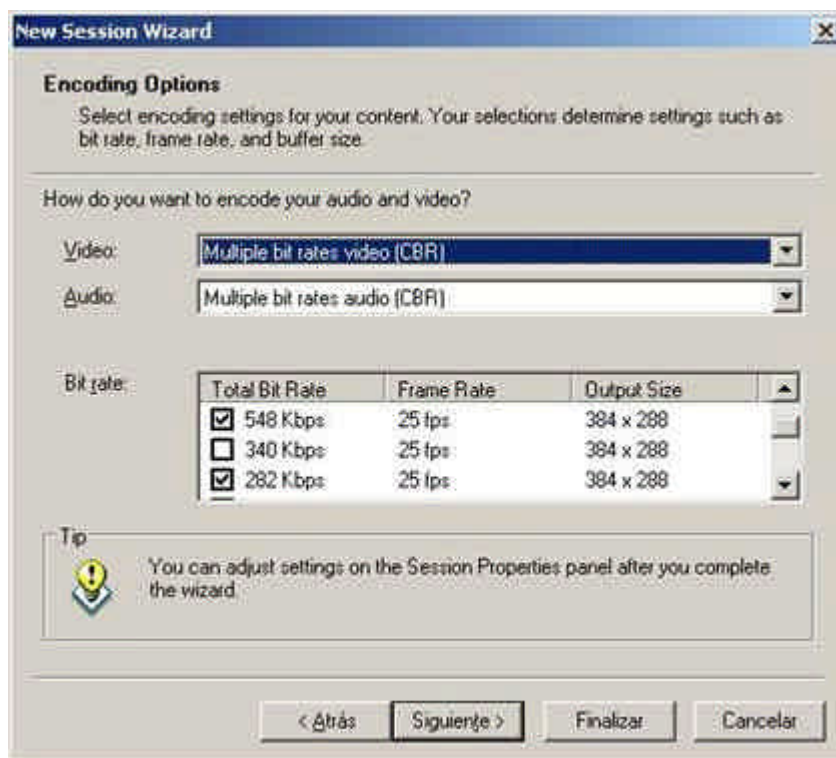
**Nota:** La configuración del broadcast como **Push** se especifica posteriormente en este mismo documento.

Tras seleccionar esta opción, el asistente pedirá configurar el puerto de entrada de las conexiones. Salvo causa justificada, es mejor dejar el puerto por defecto, 8080. Además, esta pantalla nos indicará la dirección de acceso con la que el cliente podrá acceder al contenido.

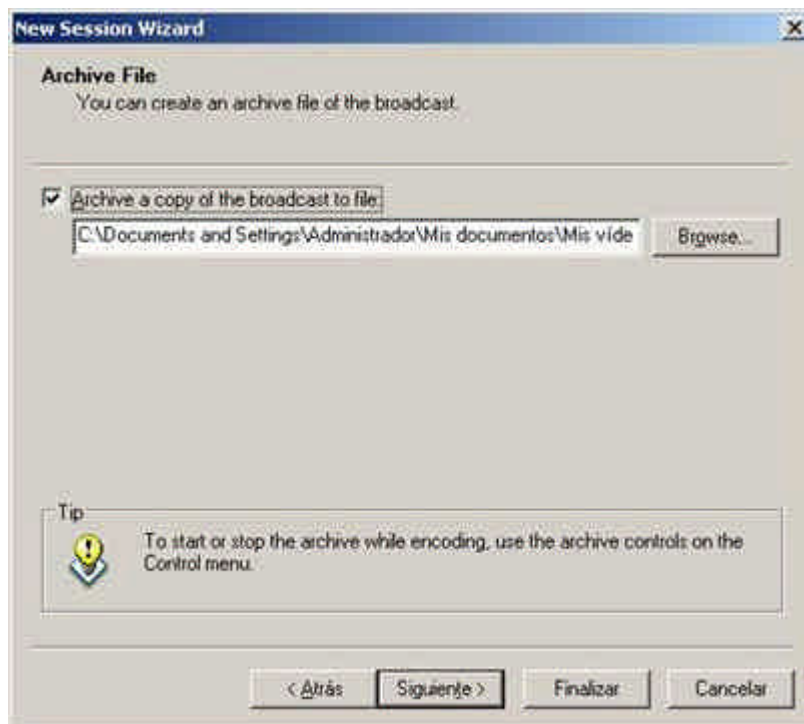
La pantalla debería aparecer de manera similar a la siguiente:



Igual que el Real Producer utilizaba la tecnología SureStream para emitir el contenido con diferentes bitrates para que sea el cliente el que defina cual se adecua mejor a su red, *Windows Media Encoder* permite configurar diferentes velocidades de transmisión a partir de la pantalla siguiente del asistente:



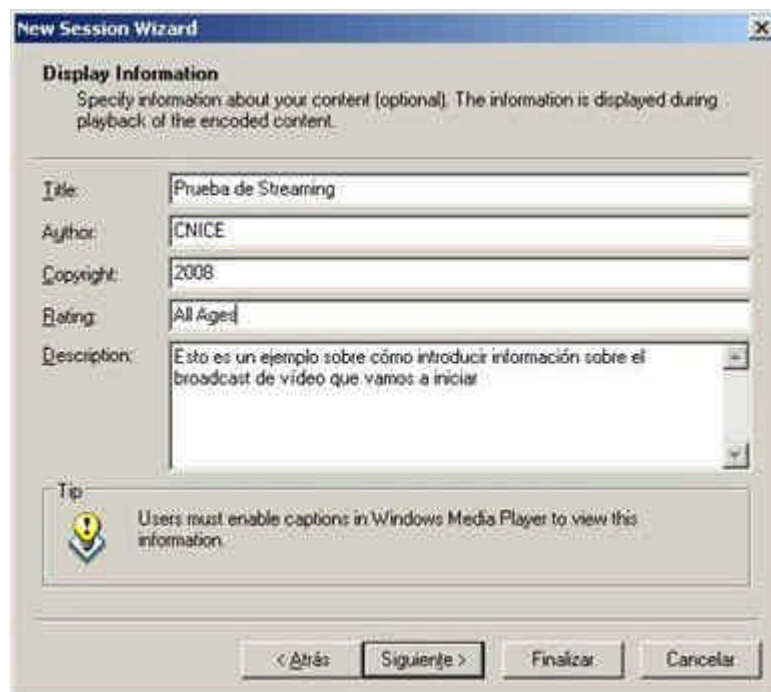
Podemos definir también si queremos guardar el contenido a un archivo en el disco duro. Puesto que la opción **Pull** la utilizaremos cuando no tengamos instalado el *Windows Media Server*, aconsejamos activar esta casilla.



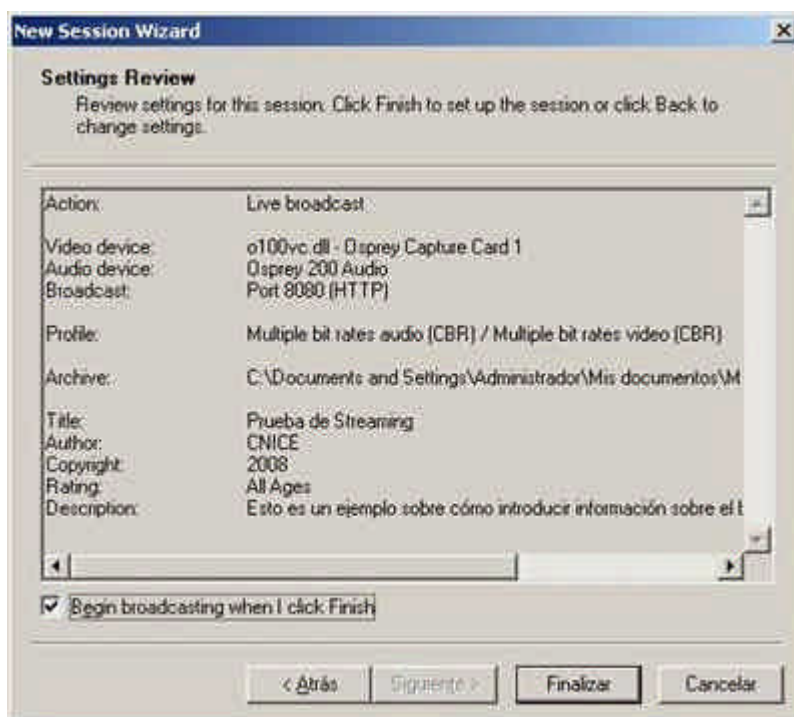
Si quisiéramos intercalar vídeos de bienvenida, anuncios y cierre al contenido multimedia podríamos hacerlo desde la siguiente ventana. Si no queremos esta opción, la desactivamos como muestra la siguiente pantalla:



Asimismo, se puede definir la información asociada al archivo que se envía, y que verán los clientes cuando se conecten al flujo de datos.



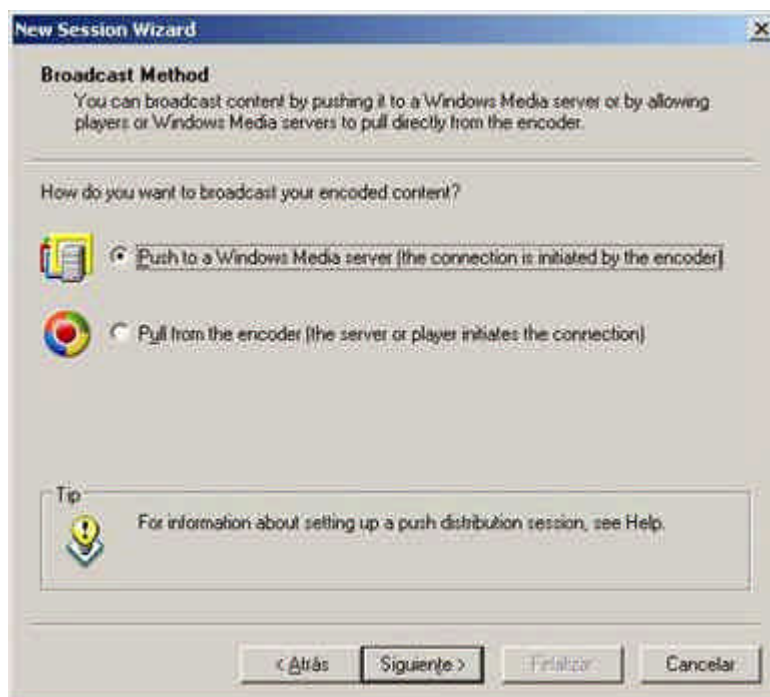
Por último, una pantalla resume la configuración del broadcast que vamos a iniciar. Si activamos la opción **Begin broadcasting when I click Finish** el codificador comenzará a transmitir nada más terminar el asistente. Si no lo hacemos, deberemos pulsar en **Start Encoding** posteriormente para lanzar el *streaming*:



Esta sería la configuración más sencilla de aplicar, cuando se trabaja sin **Windows Media Server 2003** o se desea que el cliente sea el que active el envío de streaming. Sin embargo, puesto que hemos configurado el **Windows Media Server 2003** y

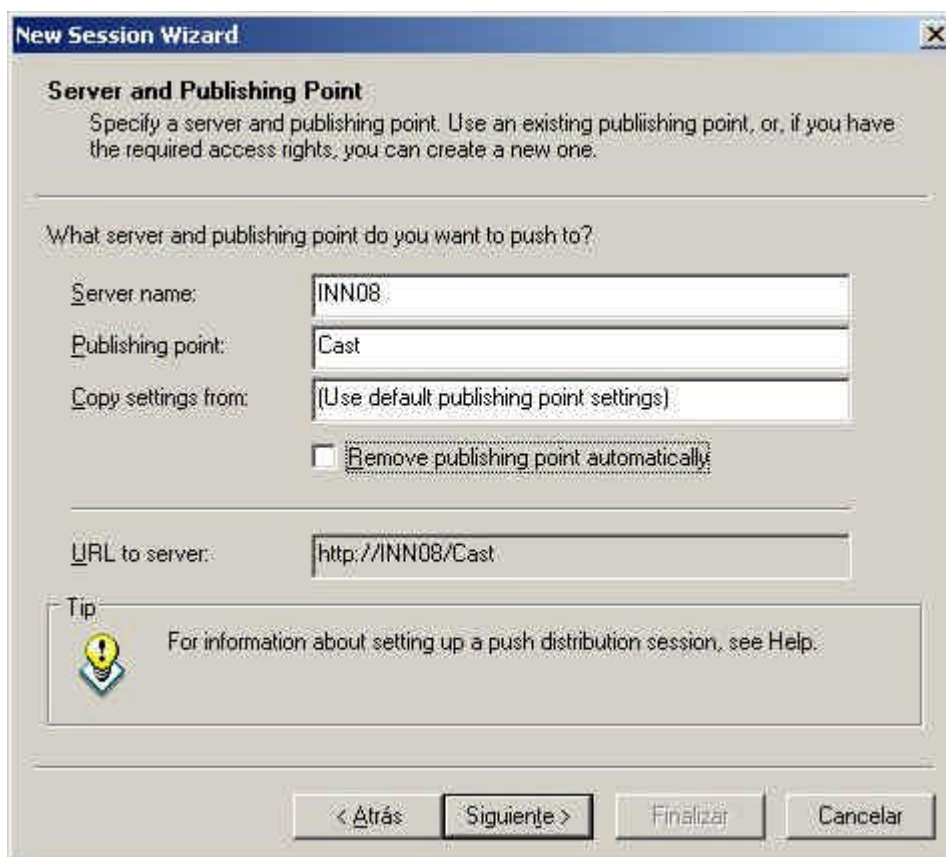
deseamos tener más control sobre el flujo de datos, volvemos sobre nuestros pasos, pulsamos en **Push to a Windows Media Server** y hacemos clic en **Siguiente**.

## 2. Método Push

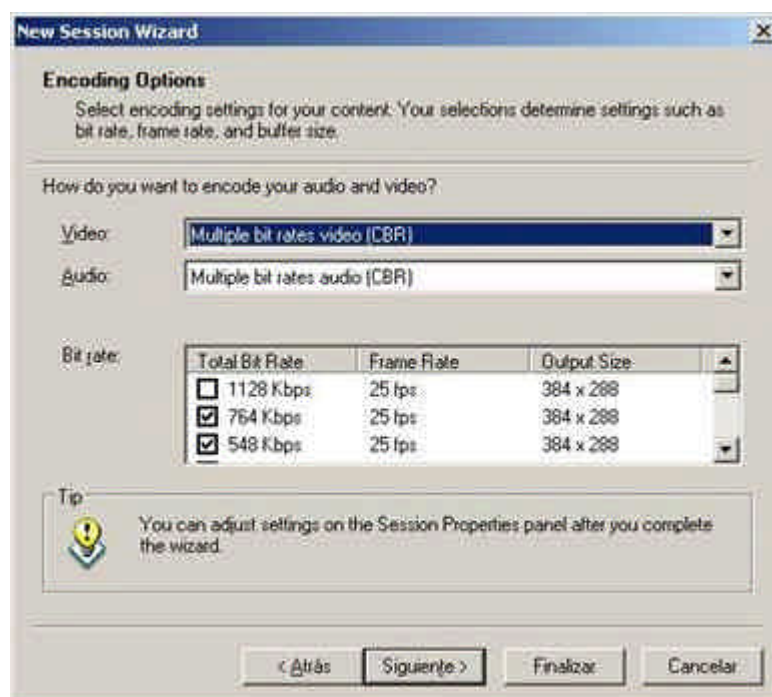


En este caso se debe especificar el servidor al cual se quiere enviar el *stream* y el punto de publicación en el que se realizará la conexión. Es posible, en este momento, definir un punto de publicación ya creado –véase el asistente de creación de puntos de publicación en el *Windows Server 2003*- o hacer que el *Windows Media Encoder* cree uno. Para conseguirlo, el servidor debe tener habilitados los permisos de creación y escritura como se especificó anteriormente. En la misma pantalla se pueden definir también las propiedades del punto de publicación –en principio dejaremos las del punto de publicación predeterminado, de forma que no haya que configurar el punto después- y se muestra la URL a través de la cual podrá acceder el reproductor. Así mismo, si se activa la casilla **Remove Publishing Point Automatically**, el punto de publicación se borrará automáticamente al terminar la sesión de broadcast. En la captura siguiente se muestra como se manda el flujo de datos al servidor *INNO8* en el punto de publicación *Cast*. Puesto que el punto de publicación que habíamos creado anteriormente se llamaba *PruebaPuntoPublicación* el encoder creará un nuevo punto de publicación utilizando la configuración del punto de publicación predeterminado, aunque no se especifica que se borre después de la transmisión.



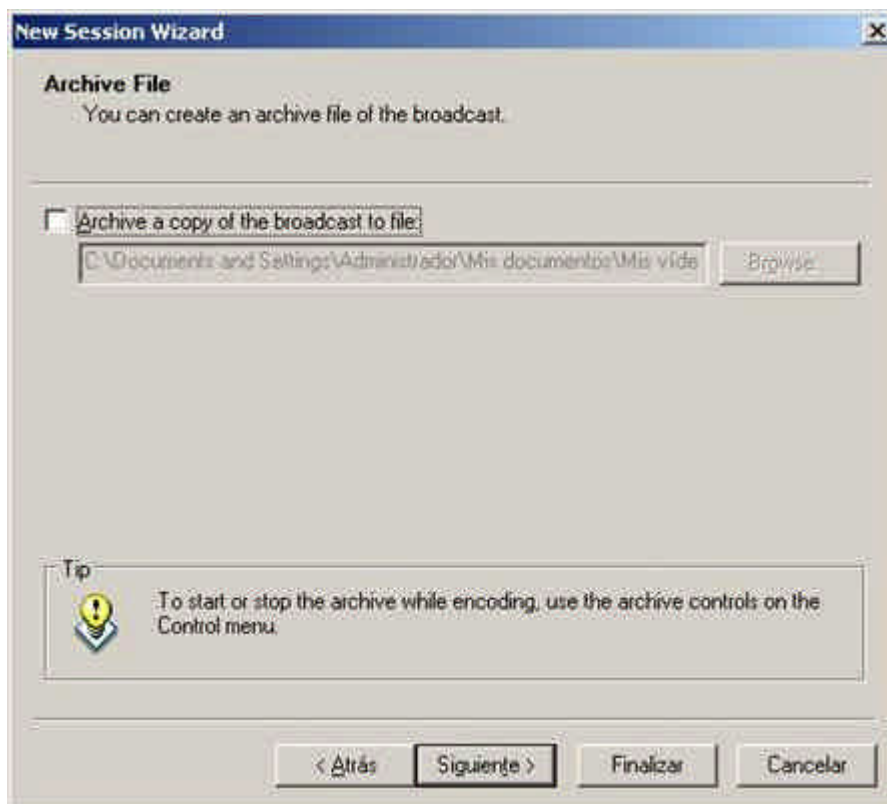


El resto de pasos son muy similares a los de la configuración de un servidor mediante el método *Pull*. A continuación se pueden definir las diferentes velocidades de transmisión – bitrates- a las que se enviará el flujo de datos.



Si pulsamos en **Siguiente** aparecerá otro cuadro de diálogo que nos permite archivar el contenido que estamos enviando. Puesto que hemos activado la opción de archivado en

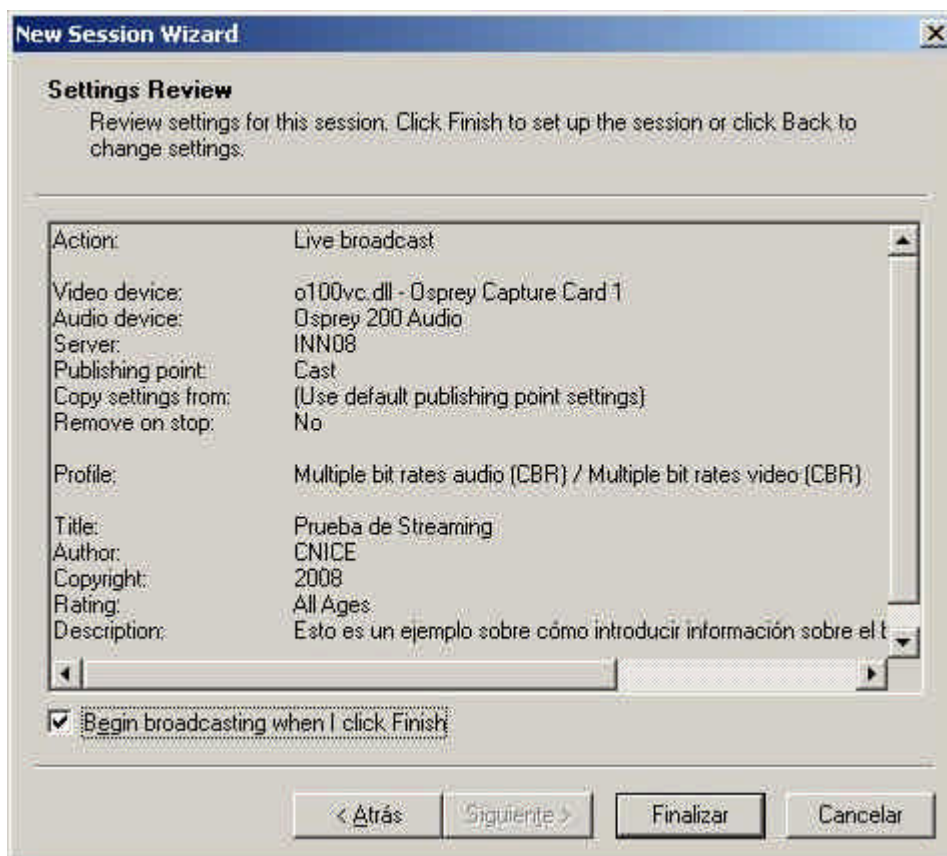
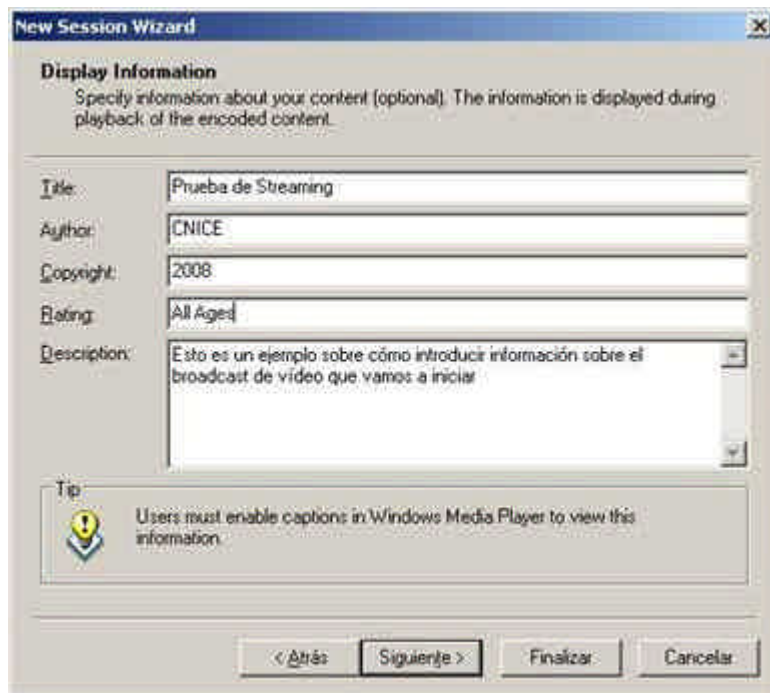
el servidor, desmarcamos la casilla a no ser que queramos realizar una segunda copia de seguridad o el servidor sea remoto:



De nuevo, como en el método *Pull*, podemos definir si queremos introducir un vídeo de bienvenida, cierre y anuncios, o sólo el vídeo que codifiquemos. Marcamos la primera casilla como indica la figura siguiente:



Por último, tendremos la posibilidad de introducir la información asociada al contenido multimedia que vamos a enviar y un resumen de la configuración efectuada, como sigue:



## **Windows Media Player / Acceso a los Contenidos**

### **Requisitos Mínimos**

- Procesador 233 MHz
- 64 MB de RAM
- 100 MB de espacio libre en disco duro
- Tarjeta de sonido de 16-bit y altavoces
- Tarjeta Gráfica Super VGA (800 x 600, 65k colores)
- Windows XP, Vista, Server 2003
- Conexión a Internet

### **Requisitos Recomendados**

- Procesador 1.5 GHz o superior
- 512 MB de RAM, 1 GB con Windows Vista
- Conexión de Internet de Alta Velocidad
- Grabadoras de CD y DVD
- Reproductor de DVD
- 60GB de espacio en disco duro
- IE 6 o posterior, Firefox

Cuando el codificador ha lanzado la secuencia de vídeo que procesa y almacena el servidor, el cliente puede acceder al contenido a través del **Reproductor de Windows Media**, que viene instalado por defecto en cualquier ordenador que tenga Windows como sistema operativo. Si hemos efectuado una sesión *Pull*, la dirección que habrá que introducir en el reproductor a través de la opción **Archivo è Abrir URL** será [http://dirección\\_IP\\_Servidor:Puerto](http://dirección_IP_Servidor:Puerto). Así, para un servidor cuya dirección IP fuera 195.53.170.85 y que transmite a través del puerto 8080, la dirección URL que le pasaremos al reproductor será

<http://195.53.170.85:8080>

Si se utiliza el método *Push* la URL es ligeramente diferente, pues hay que introducirle el punto de publicación del servidor al que el codificador está enviando el flujo de datos ([http://dirección\\_IP\\_Servidor/Punto\\_Publicación](http://dirección_IP_Servidor/Punto_Publicación)), de forma que si el servidor tiene como dirección IP 195.53.170.85 y el punto de publicación creado es *Cast* la dirección URL será

<http://195.53.170.85/Cast>

Al margen, existen otros métodos para ofrecer al cliente el flujo de datos, en lugar de que sea él quien introduzca la dirección. Se puede embeber el reproductor en una página Web con la configuración iniciada, u ofrecer un link basado en el protocolo **http** (con la misma dirección anterior) o **mms**, que lanzará el reproductor de Windows media. La dirección, en este último caso, será igual que en el anterior, pero cambiando el protocolo ([mms://dirección\\_IP\\_Servidor/Punto\\_Publicación](mms://dirección_IP_Servidor/Punto_Publicación)). La ventaja de este último sistema es que, si la opción de archivado está activada en el servidor y no se está emitiendo un evento en directo se recibirá la última emisión recibida en el punto de publicación. De nuevo en nuestro ejemplo, para una dirección IP 195.53.170.85 y un punto de publicación *Cast* la dirección que el cliente ejecuta desde el navegador Web será:

<mms://195.53.170.85/Cast>

## Adobe/Macromedia Flash

La última solución comercial analizada es una de las que más futuro tiene en la actualidad. Se basa en el uso de vídeos Flash, con formato *\*flv*, un sistema de codificación utilizado en múltiples páginas Web 2.0 como *Youtube* o *Kewego*. La calidad que podremos obtener a través de este sistema es algo inferior a las aportadas por las soluciones de Real Media o Windows Media, pero a cambio el ancho de banda se reduce razonablemente en relación a la misma calidad utilizando los otros métodos. Este problema, sin embargo, se verá solventado en un breve espacio de tiempo, pues ya se ha anunciado una nueva versión de los elementos de *Macromedia* –comprado recientemente por *Adobe*- que permitirá la edición de contenidos en alta definición.

Como siempre, se necesita un programa codificador si queremos enviar un vídeo en directo, el *Flash Media Encoder*, y un programa servidor, *Flash Media Server*. El cliente, sin embargo, no precisa de un programa reproductor. Simplemente ha de instalarse un plugin –disponible para todas las plataformas- para visionar el contenido. Al margen, se precisa el programa *Macromedia Flash* para poder publicar el archivo de streaming.

### Flash Media Server 2 / Administración de contenidos

#### Requisitos Mínimos

- Procesador Pentium 4, 3.2 GHz
- 1 GB de RAM
- Tarjeta Ethernet 1Gigabit

Windows XP, Vista, Server 2000/2003, Linux Red Hat Enterprise

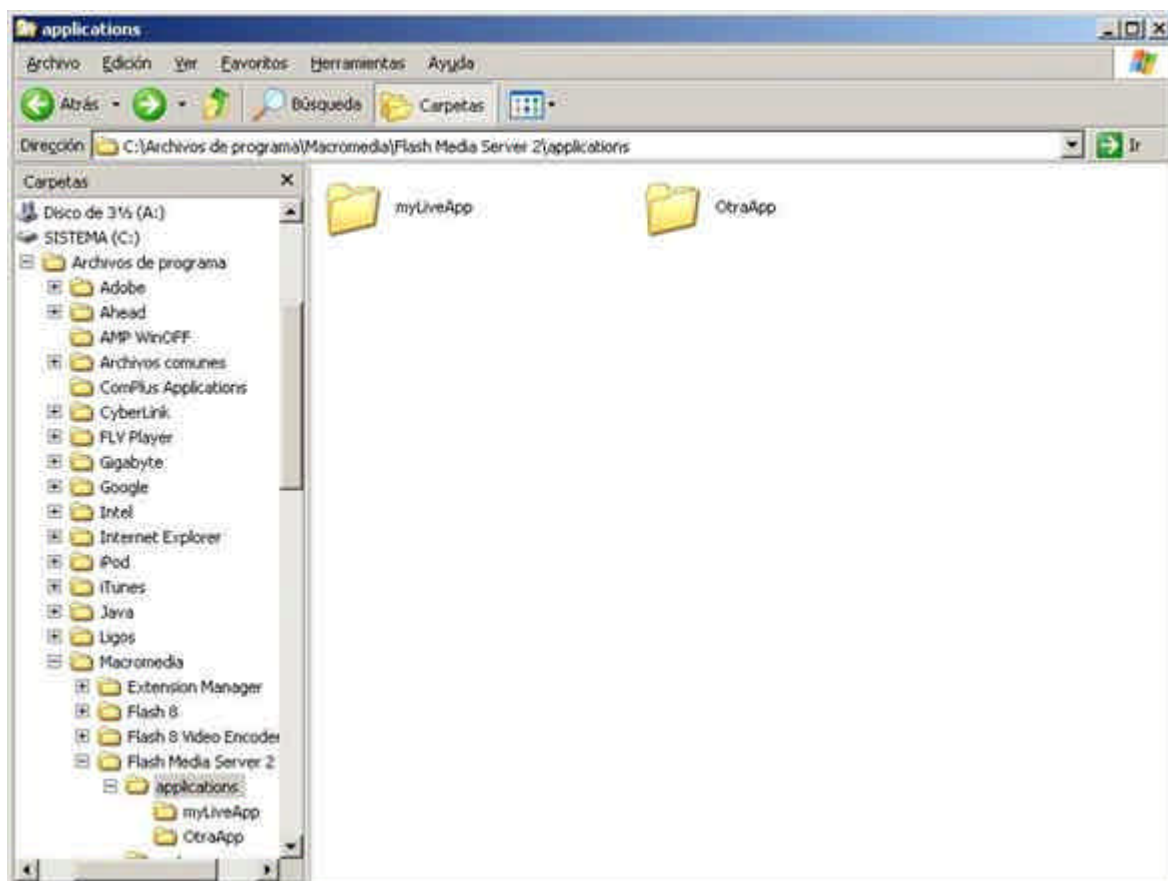
El primer paso antes de realizar la codificación consiste en poner en marcha el servidor y configurarlo adecuadamente. En el momento de realización de este tutorial la versión actual del programa es la 2, pero Macromedia ha anunciado el inminente lanzamiento de *Flash Media Server 3*

(<http://www.adobe.com/es/products/flashmediaserver/>). Por ello, aunque la base de este tutorial se mantenga intacta, algunas características pueden variar.

**Nota:** Actualización a *Flash Media Server 3* mas adelante

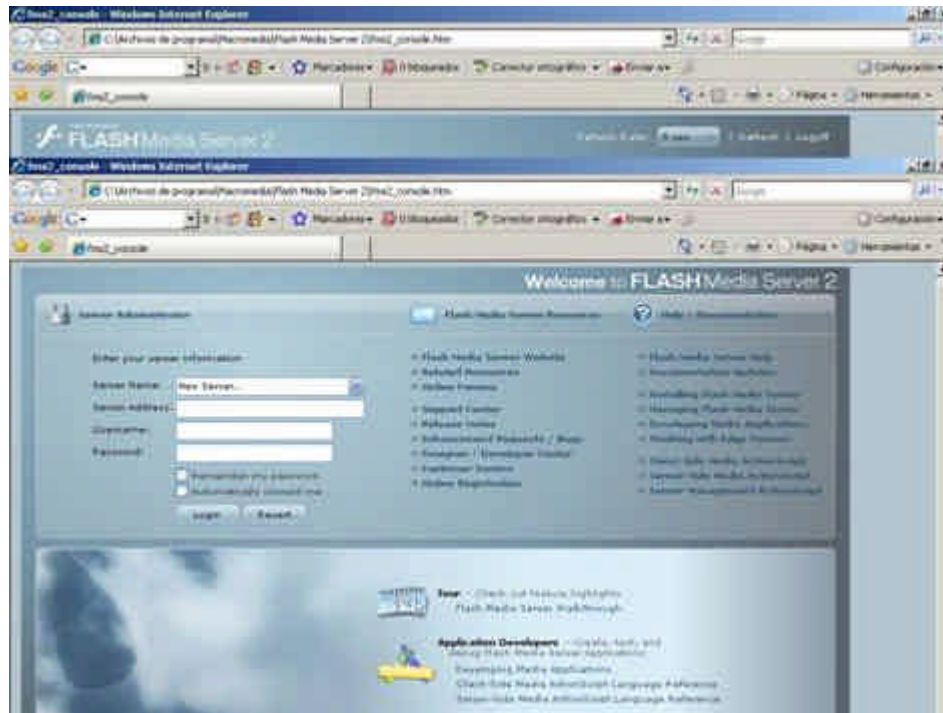
El proceso de instalación es muy sencillo y se recomienda dejar –salvo causa justificada- los valores por defecto en los puertos de streaming. Una vez instalado el servicio se ejecuta automáticamente cada vez que se inicia Windows, y sólo tenemos que pulsar en **Inicio è Programas è Macromedia è Flash Media Server 2 è Management Console**, si la instalación se ha efectuado con los valores por defecto, para lanzar la consola de mantenimiento del servidor.

Antes de abrirla, sin embargo, hay que crear una aplicación para que el servidor la lea y podamos alojar contenido en ella. Para ello, vamos al directorio *applications* de la carpeta donde hayamos instalado el servidor –por defecto *C:Archivos de ProgramaMacromediaFlash Media Server 2applications*- - crearemos una carpeta con el nombre que queramos. Es importante recordar dicho nombre, pues habrá que pasárselo después al servidor y al codificador. También se debe prestar atención a la forma de escribirlo, pues el programa es sensible a las mayúsculas. En la siguiente figura se han creado dos aplicaciones, *myLiveApp* y *OtraApp*:

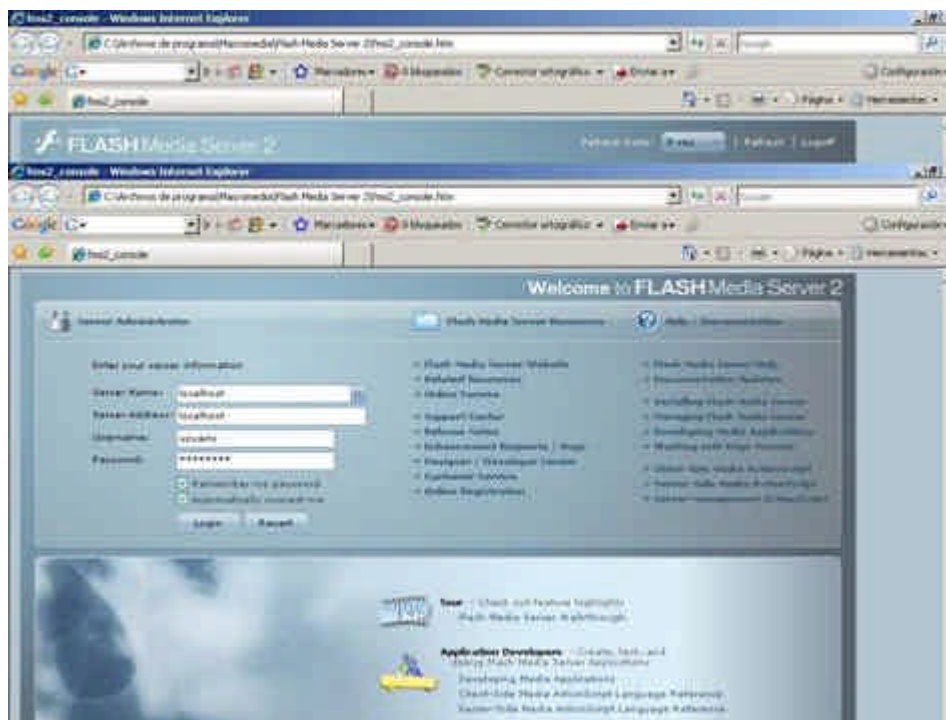


Al entrar en la consola de mantenimiento del *Flash Media Server* por primera vez tendremos que crear un servidor. Así, veremos una pantalla como la siguiente:



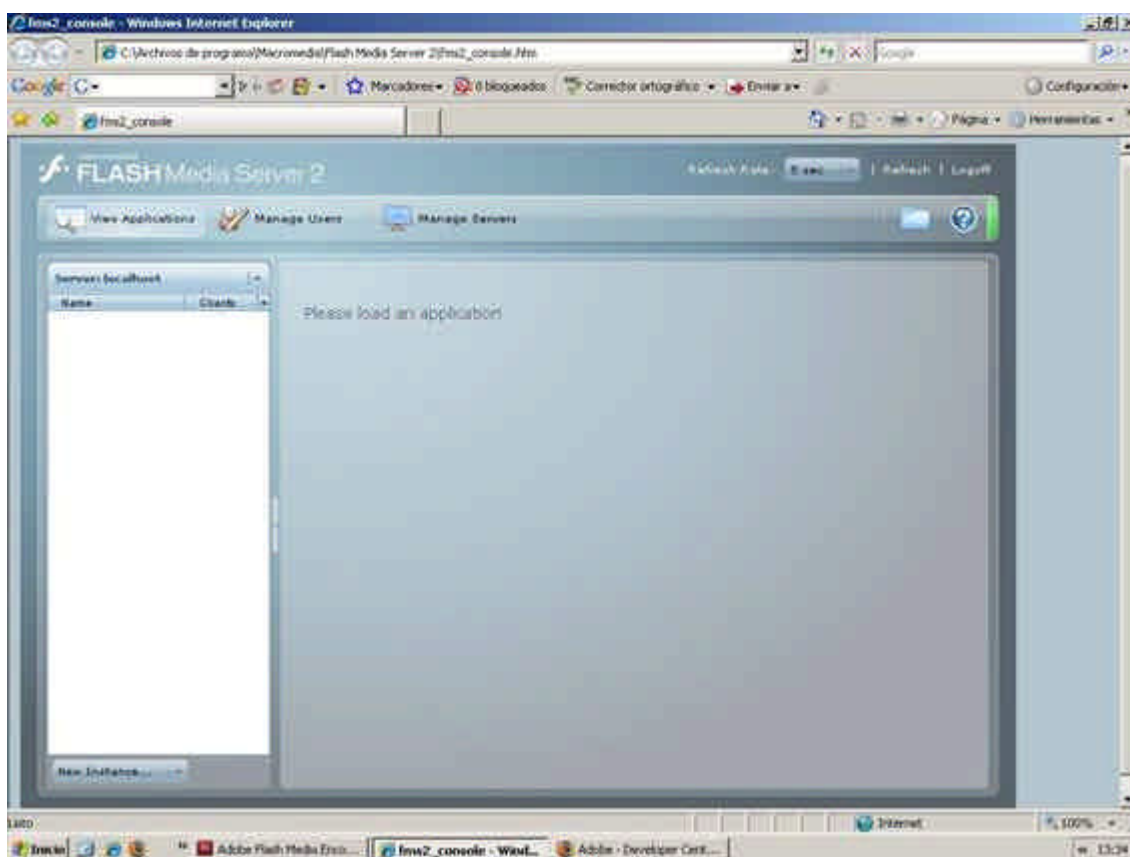


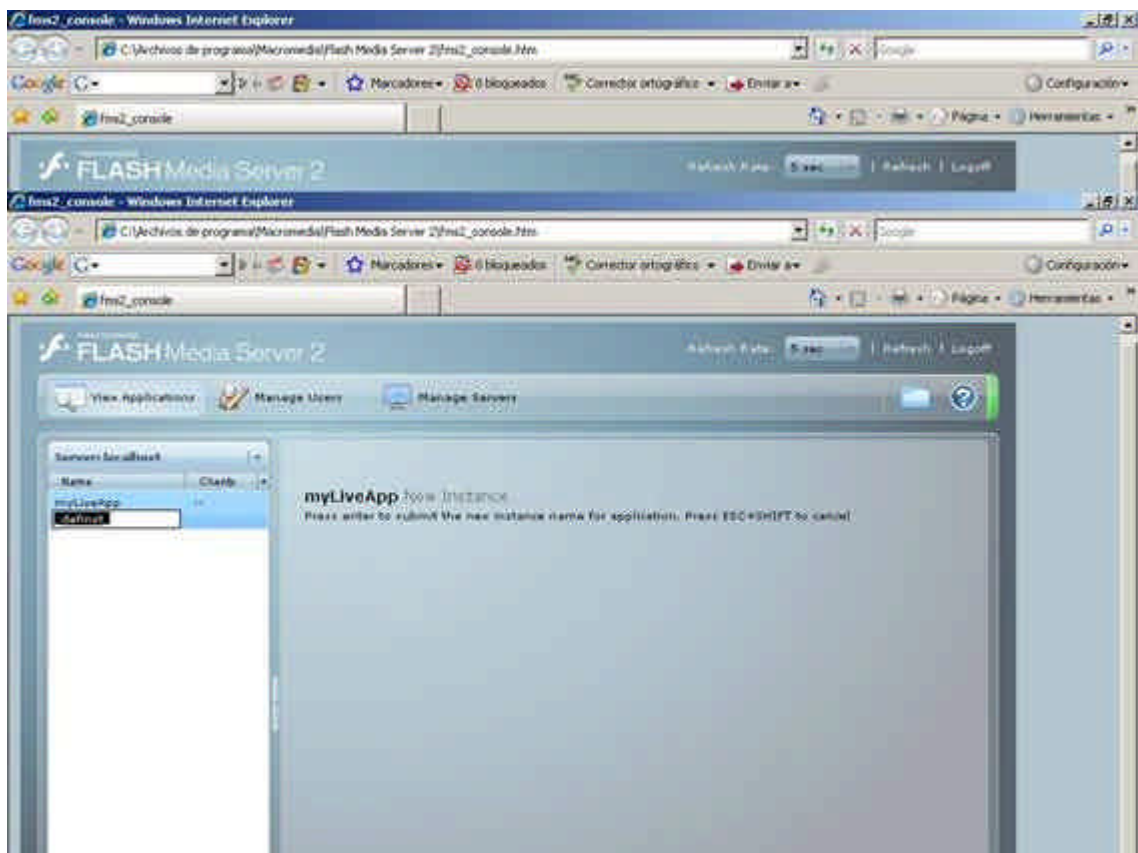
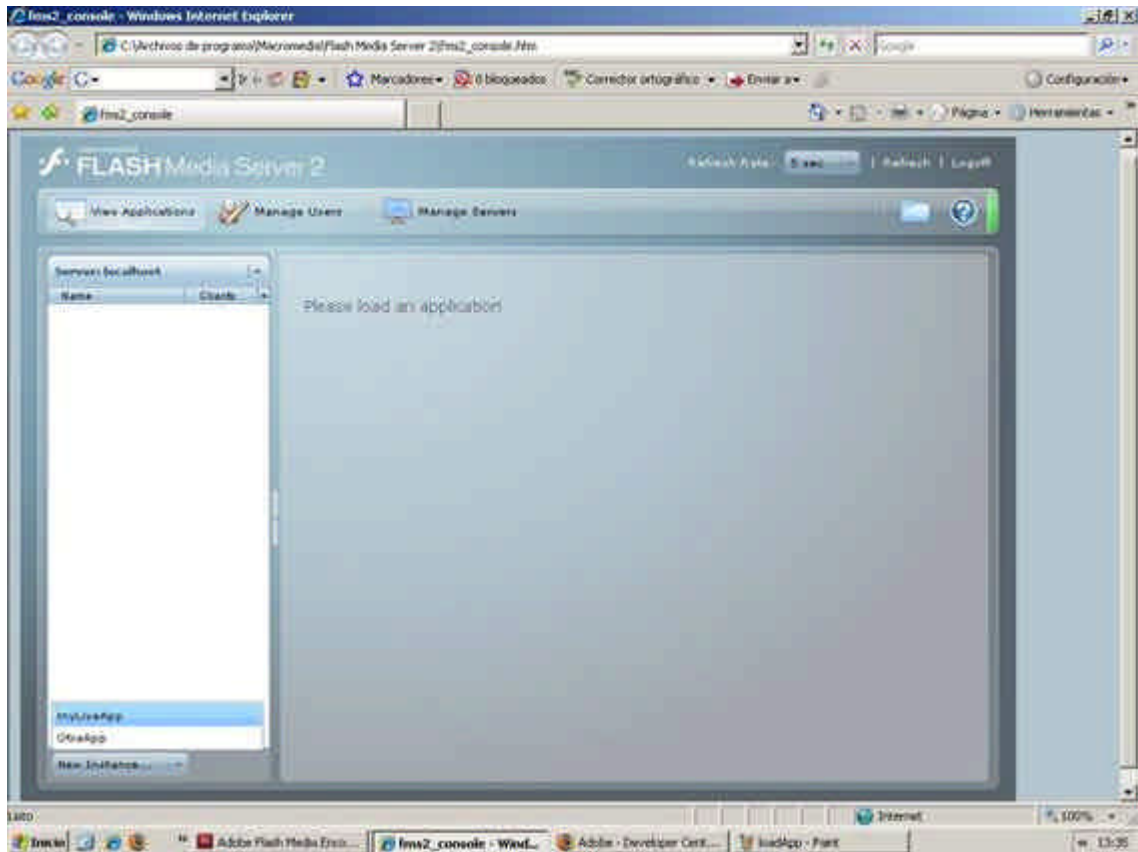
En ella podremos introducir el nombre del servidor, la dirección IP del mismo y el nombre de usuario y contraseña que utilizaremos para administrarlo. Si la autenticación no es crítica, podemos indicar al programa que recuerde el usuario y la contraseña para que inicie la consola sin preguntarnos por ella. Si el servidor se encuentra en el mismo ordenador –no es remoto– podemos utilizar el alias *localhost* y no utilizar la dirección IP. En la siguiente captura se observa el proceso de creación de un servidor de nombre localhost, con nombre de usuario *usuario* y contraseña *livecast*.

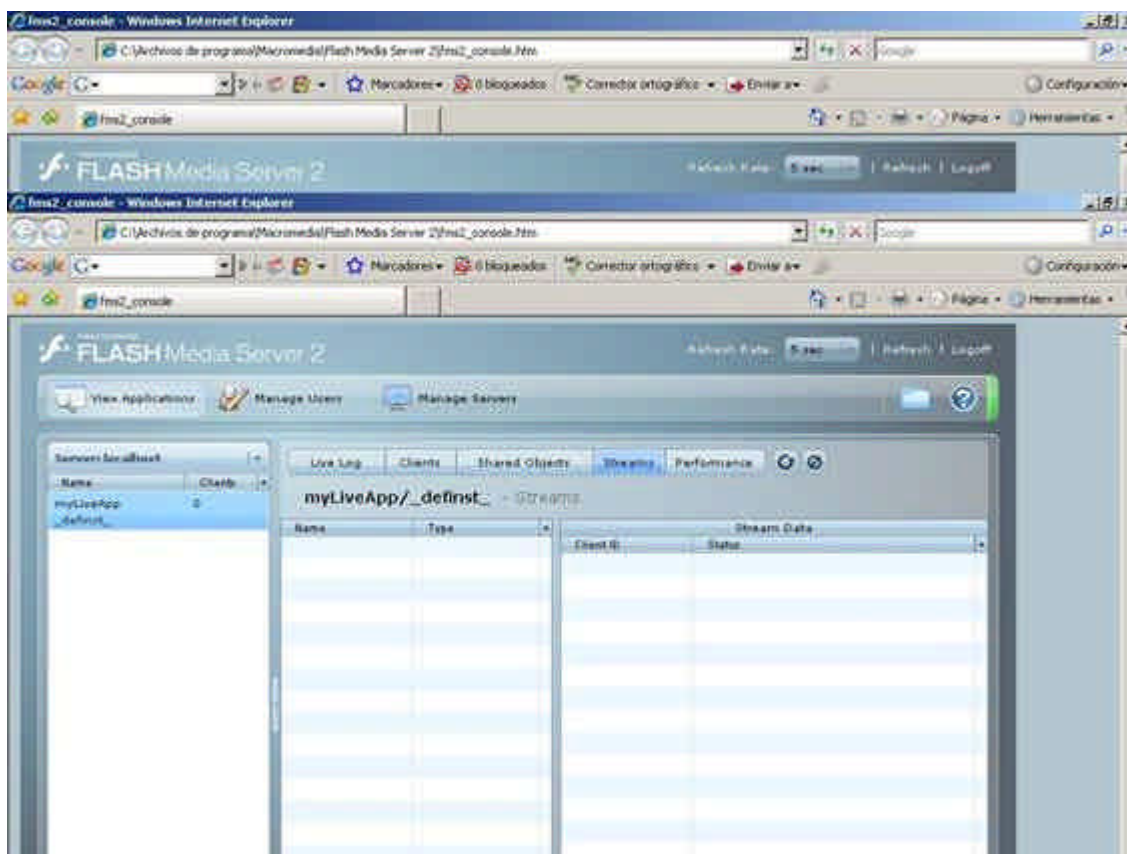


Una vez iniciado el servidor, el siguiente paso es cargar la aplicación que acabamos de crear en el directorio *applications*. Para ello nos movemos sobre la pestaña **View Applications** y pulsamos sobre **New Instance**. Hacemos clic sobre el nombre de la aplicación –en nuestro caso, *myLiveApp*–, y creamos una instancia dentro de la misma. Podemos dejar, si no queremos cambiar el nombre, la instancia por defecto, *\_definst\_*, pero hemos de tener en cuenta que el nombre que le demos es necesario para la URL que ofreceremos al cliente.

En la siguiente secuencia de imágenes se observa cómo cargar la aplicación y qué opciones tendremos una vez creada.







Si hemos utilizado las opciones por defecto no se necesita configurar nada más. Así, el siguiente paso será codificar el vídeo y enviarlo al servidor a través de la aplicación que acabamos de crear.

## Flash Media Encoder 2 / Codificación de Contenidos

### Requisitos Mínimos

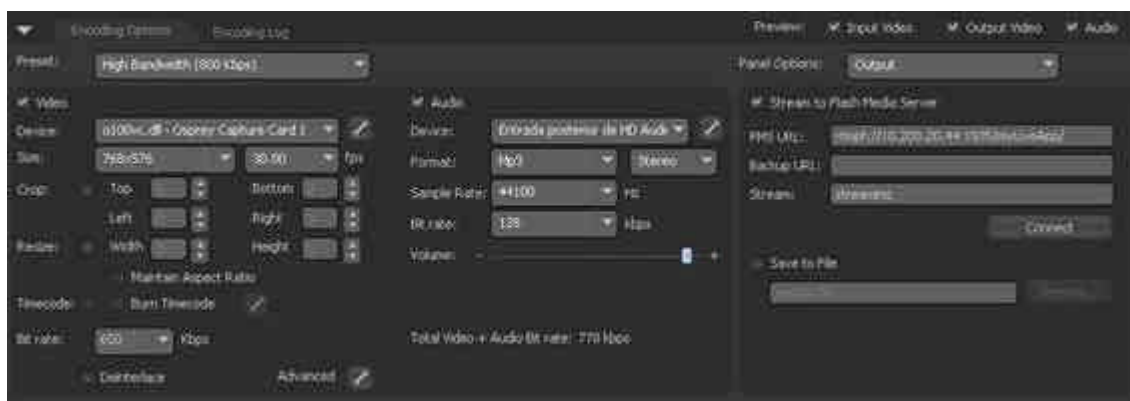
- Procesador 933 MHz
- 256 MB de RAM - 1GB Recomendado-
- 40 MB de espacio libre en disco duro
- Tarjeta de sonido de 16-bit y altavoces
- Tarjeta Gráfica Super VGA (1024 x 768, 32 bits)
- Windows XP con Service Pack 2, Vista, Server 2003
- Microsoft DirectX 9.0 o superior
- Flash Media Server 2.04 (o superior) para realizar streaming
- Flash 7 o superior para enviar el stream a un servidor Web

El programa *Flash Media Encoder*

(<http://www.adobe.com/products/flashmediaserver/flashmediaencoder/>)

posee una interfaz gráfica muy similar a los programas codificadores ya analizados: dos pequeñas pantallas muestran la señal de entrada y la de salida, mientras que la parte inferior permite la configuración de las fuentes de audio y vídeo, así como el formato de salida. En principio, el programa detecta automáticamente si hay conectada una cámara y la configura por defecto, pero si no es así podemos seleccionarla de la lista desplegable. También se puede

configurar el bitrate utilizado en la codificación y el número de cuadros por segundo que tendrá el vídeo de salida.



La única gran complicación en este proceso es definir una conexión con el servidor correcta. El protocolo que utiliza por defecto Flash para la transmisión de streaming es **rtmp (Real Time Messaging Protocol)**; sin embargo, el servidor de Flash puede no ser compatible con determinado software que tengamos instalado en el ordenador, por lo que aunque hagamos una conexión correcta el programa mostrará un mensaje de error diciendo que no puede conectarse al servidor. Conocer la razón por la que esto ocurre puede evitar grandes quebraderos de cabeza. Los programas antivirus, como el Panda o el Norton, previenen la comunicación entre el programa codificador y el servidor, por lo que nunca se realizará la conexión. La documentación del programa da pocas soluciones, y no sirve simplemente con desactivar el programa; hay que desinstalarlo. Puesto que no es recomendable quedarnos sin antivirus hay otra opción para realizar el envío: si utilizamos el protocolo **rtmpt** en lugar del **rtmp** estaremos encapsulando en peticiones **http** los datos, de forma que podremos realizar la comunicación a través de antivirus y firewalls. Si utilizamos **rtmps** lo haremos, además, de forma segura.

Aclarado esto, analizaremos cómo realizar la conexión al servidor: en la pestaña **Output** –en la parte inferior derecha de la pantalla– marcaremos la opción **Stream to Flash Media Server**, e introduciremos la URL con la siguiente sintaxis:

***rtmpt://dirección\_IP\_Servidor:Puerto/aplicación/instancia***

Por ejemplo, para un servidor cuya dirección IP es 10.200.20.44 utilizando el puerto de comunicación por defecto, 1935, y una aplicación creada como myLiveApp con la instancia por defecto, **\_definst\_**, la URL que le pasaríamos al programa sería:

***rtmpt://10.200.20.44:1935/myLiveApp/\_definst***

Si se trata de la instancia predeterminada, como es el caso, podemos omitirla y el programa se conectará a ésta:

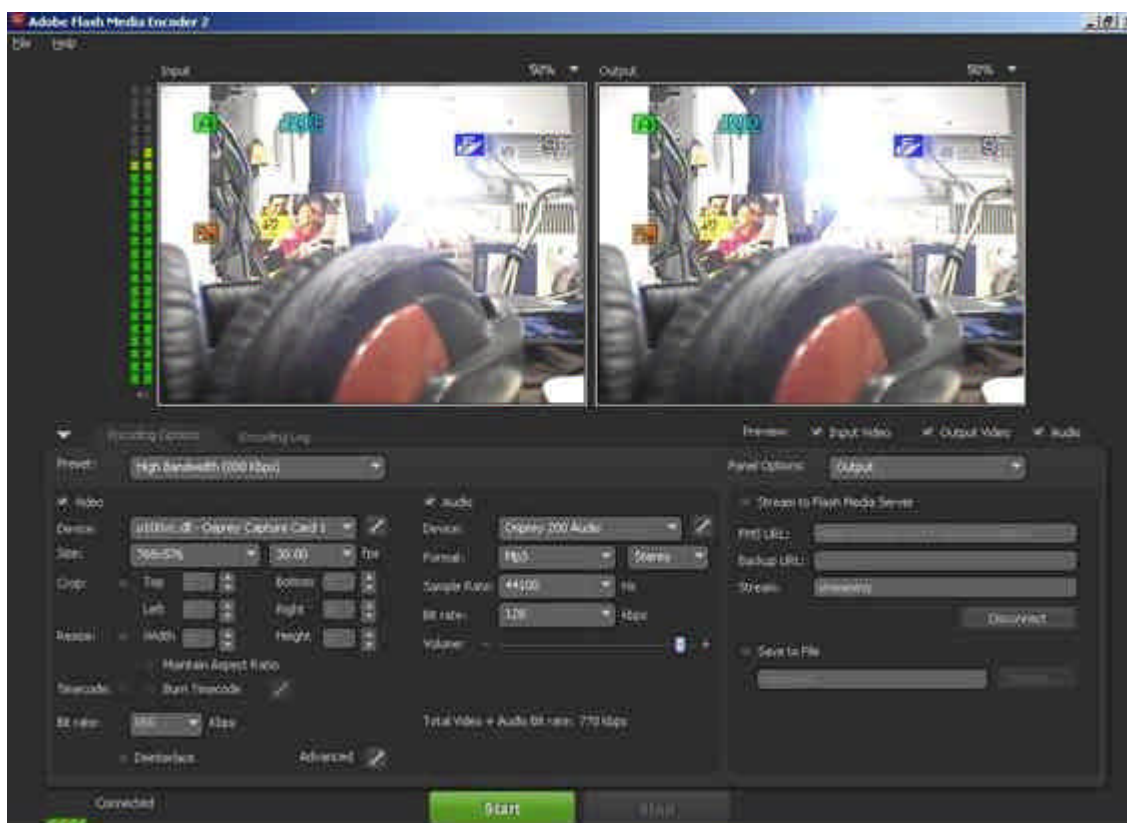
***rtmpt://10.200.20.44:1935/myLiveApp***

El otro parámetro que tenemos que configurar es el nombre del stream que enviaremos, a través de la opción **Stream**. Pondremos un nombre que identifique el contenido y le daremos a **Connect** para que establezca la conexión con el servidor en la aplicación e instancia seleccionadas. Si en este punto aparece un mensaje de error hay que revisar el



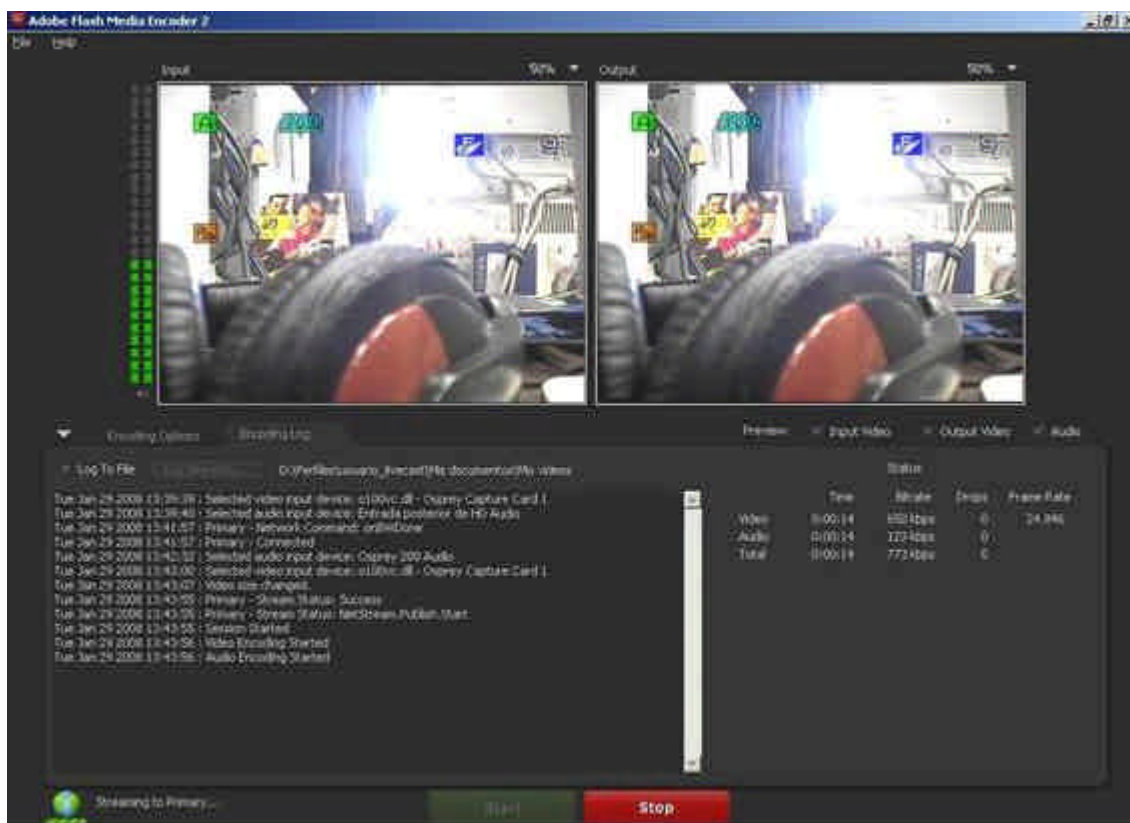
protocolo utilizado y si el servidor se encuentra funcionando en ese momento. Son de ayuda, también, los *Logs* que muestra el servidor cada vez que alguien intenta conectarse.

Si este paso se ha realizado bien, en la parte inferior izquierda de la pantalla debe aparecer un icono verde con la palabra **Connected**. También podemos guardar una copia del material codificado a través de la opción **Save to File**.

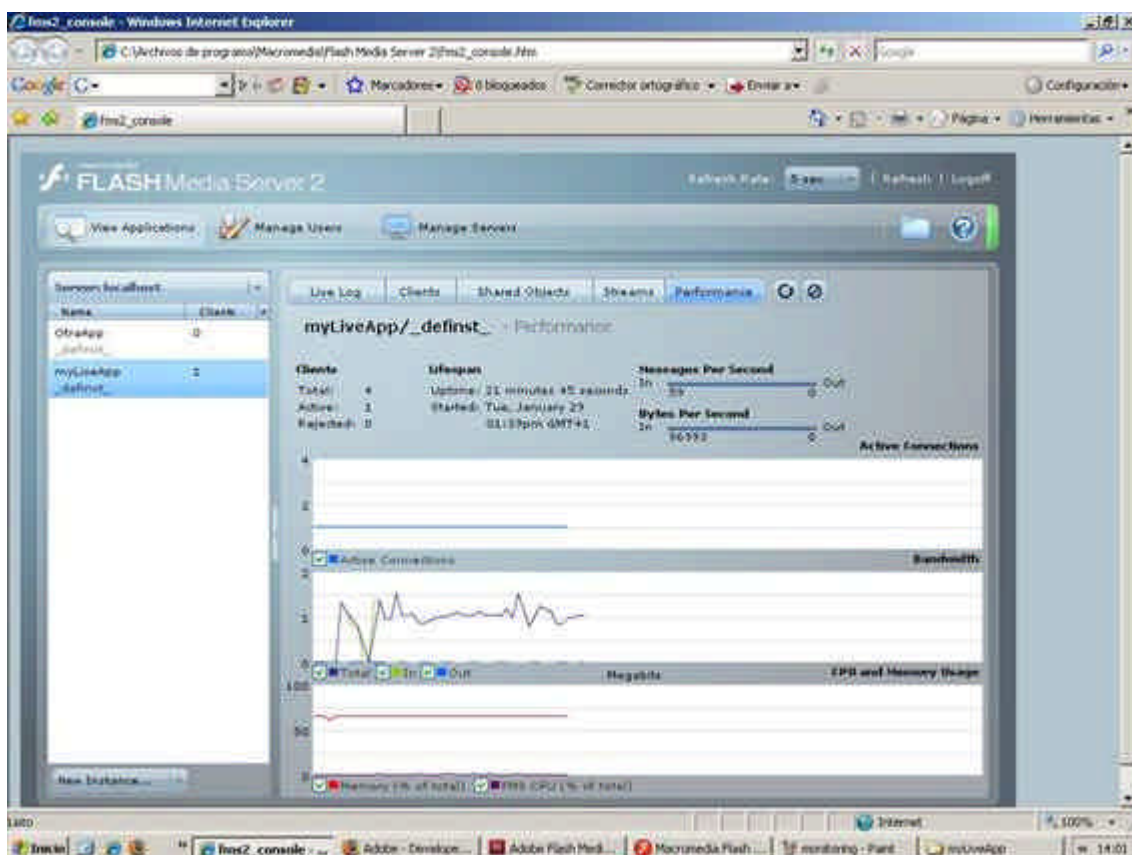
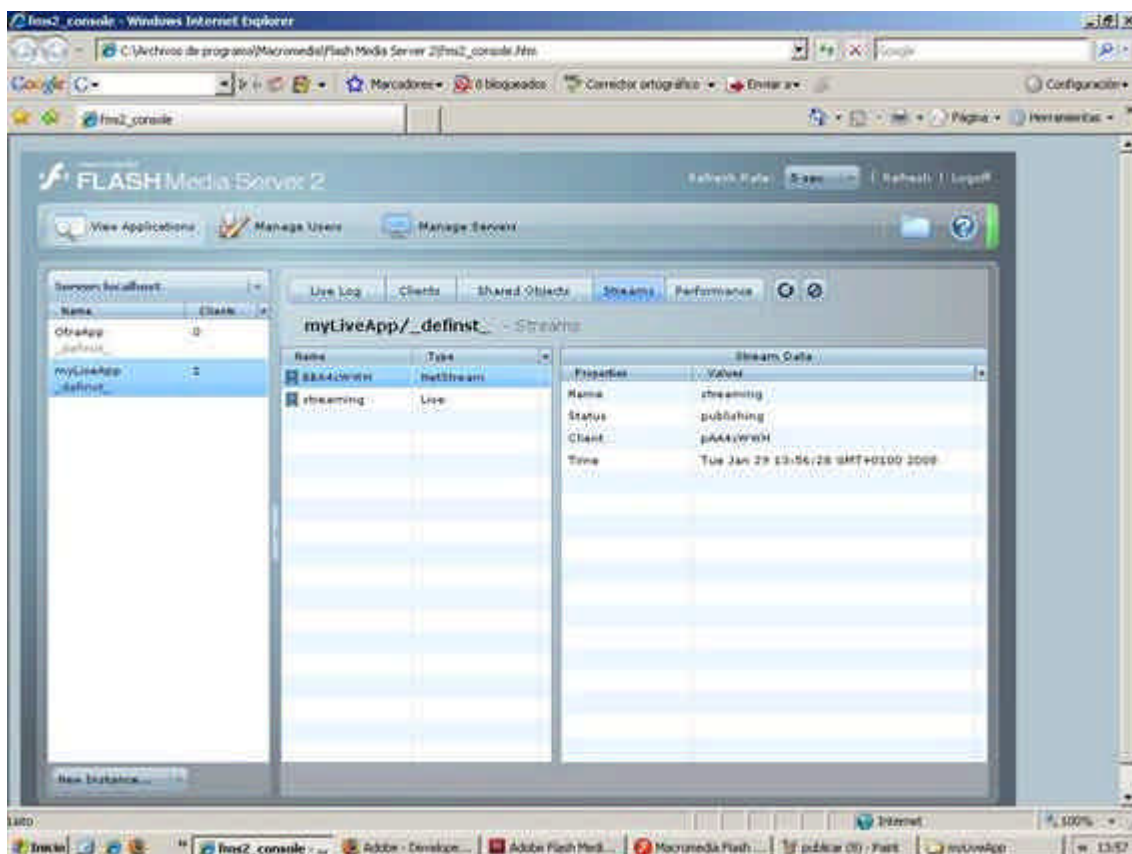


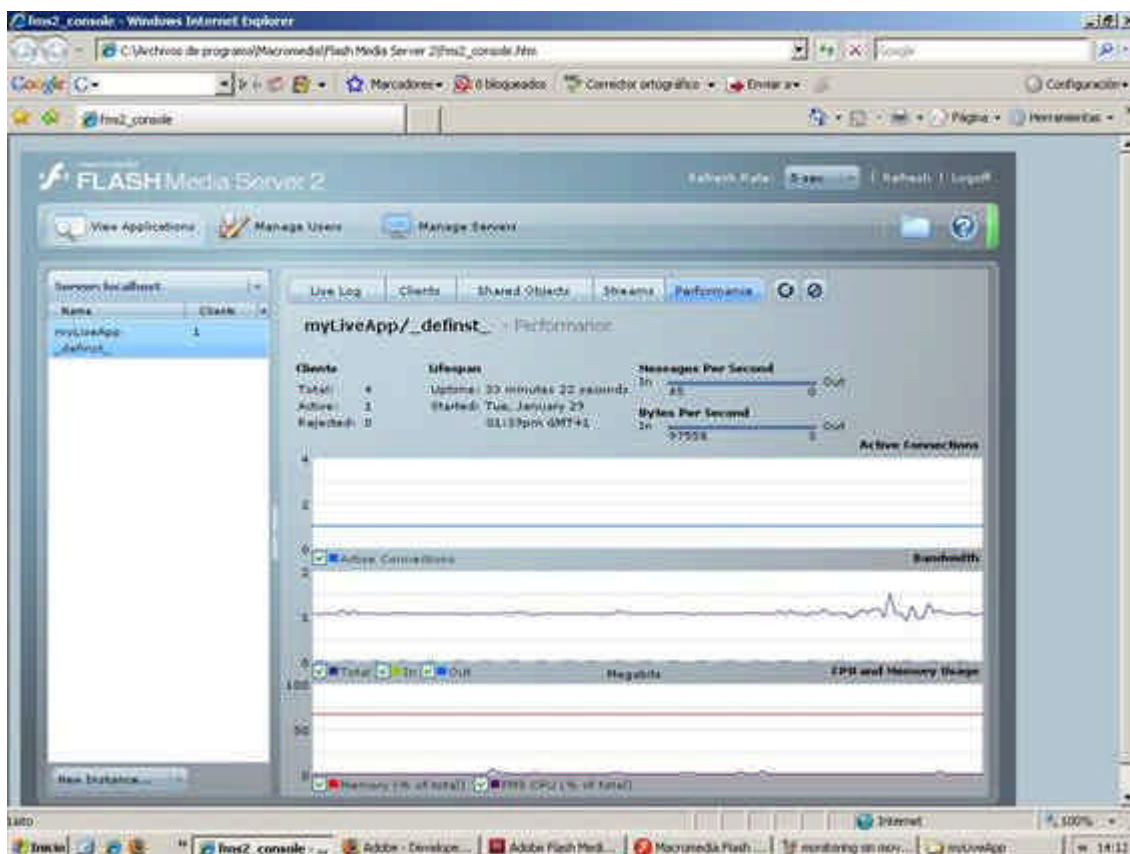
El último paso para comenzar a emitir es pulsar sobre el botón verde **Start**. Aparecerá entonces una pantalla con las señales a codificar y codificada y todos los datos de la transmisión. Para pararla simplemente habría que pulsar sobre el botón rojo **Stop**.





En este momento habremos comenzado a emitir, y podremos monitorizar la tarea desde el *Flash Media Server*. Si utilizamos la consola de administración del servidor y hacemos clic sobre la aplicación a la que estamos enviando el stream podemos ver el estado del mismo y el uso de ancho de banda y memoria que conlleva:





## Flash 8 / Preparación de contenidos

### Requisitos Mínimos

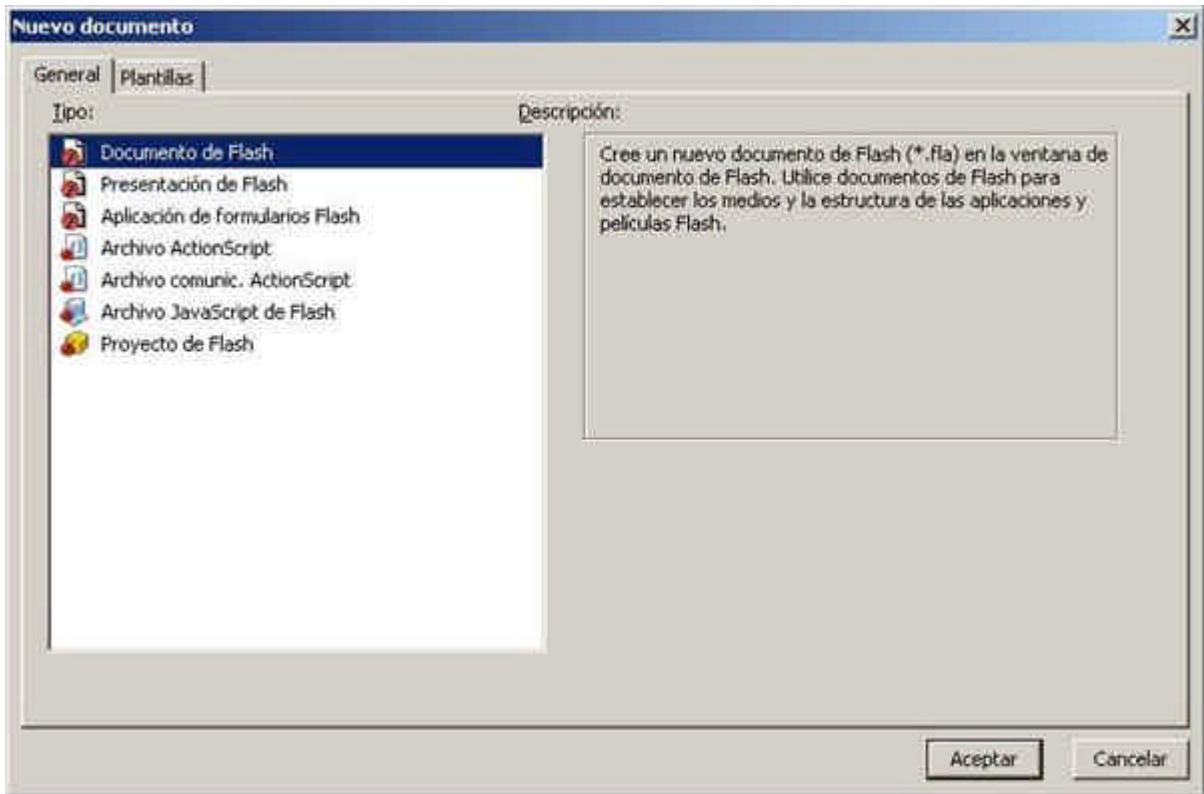
- Procesador 1 GHz
- 512 MB de RAM
- 2.5 GB de espacio libre en disco duro
- Windows XP con Service Pack 2, Vista, Mac OS X, Leopard
- Conexión a Internet

### Requisitos Recomendados

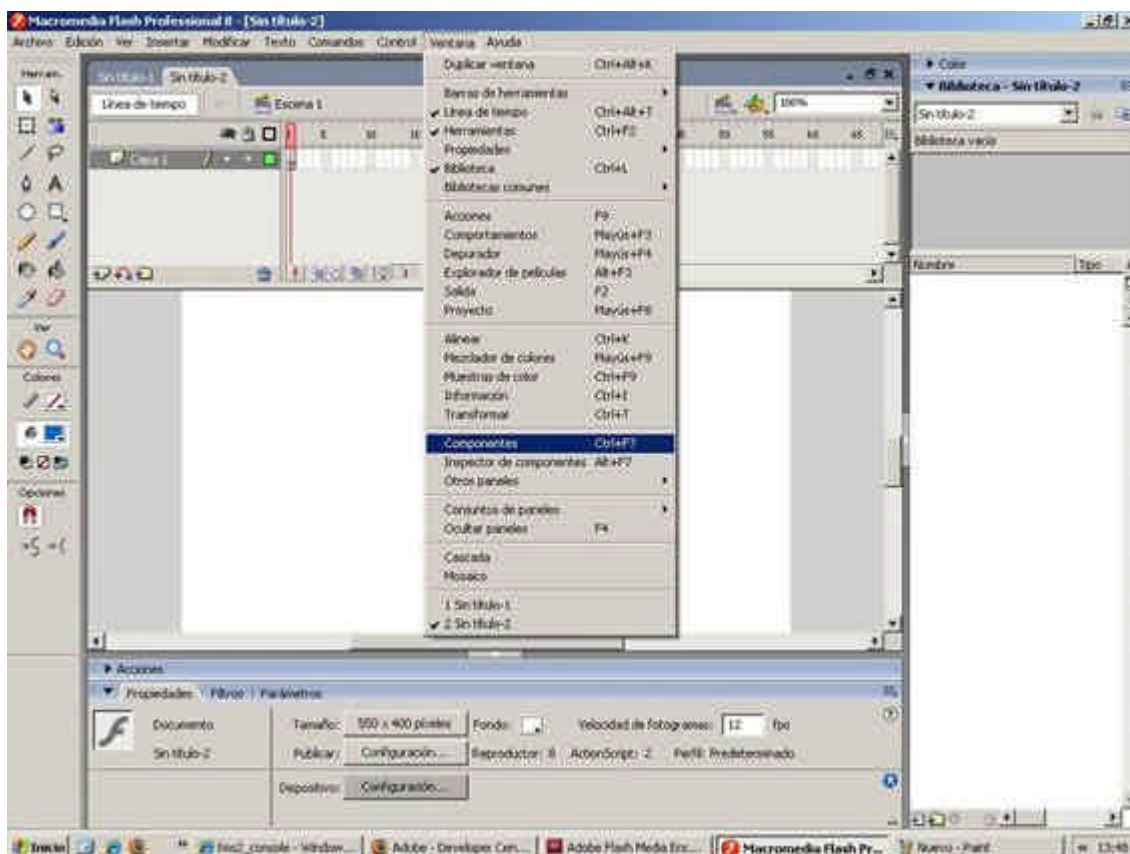
- Procesador 1 GHz o superior
- 1 GB de RAM
- Conexión de Internet de Alta Velocidad
- Grabadoras de CD y DVD
- Reproductor de DVD y software
- Quick Time 7.1.2 o posterior

El formato de salida del stream es **.flv**. Una opción para que el cliente pueda ver el mismo es embeber un reproductor de flash en una página Web, a través de un archivo **.swf**, de forma que sea más sencillo para el usuario acceder al mismo. Para realizar esta acción, Adobe Macromedia posee varios programas, pero nosotros utilizaremos el más popular, **Adobe Flash** (<http://www.adobe.com/es/products/flash/>). En nuestro ejemplo hemos utilizado la versión 8 del programa. Junto al mismo se suministra un complemento para reproducir archivos FLV, denominado **FLV Playback**.

Una vez instalado procederemos a abrir un nuevo documento de flash pulsando sobre **Archivo ò Nuevo Documento ò Documento de Flash**.



El proceso consistirá en arrastrar un componente reproductor de archivos FLV al escenario y editar sus propiedades para que reciba el stream de vídeo. Para ello, el primer paso será abrir la pestaña de complementos mediante **Ventana ò Componentes** o mediante el acceso directo **Ctrl. + F7**,

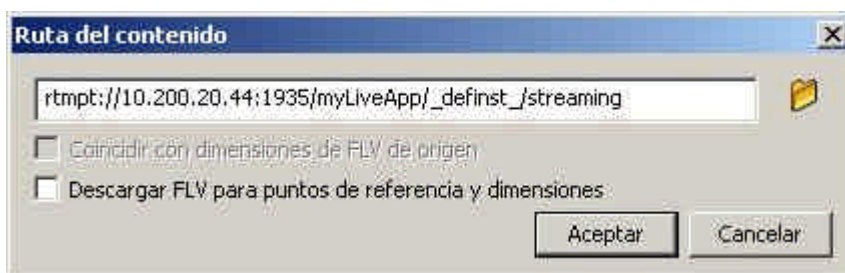


Aparecerá entonces una ventana con los distintos componentes que tenemos instalados y podremos arrastrar al escenario una instancia de FLVPlayback, como indica la siguiente figura:

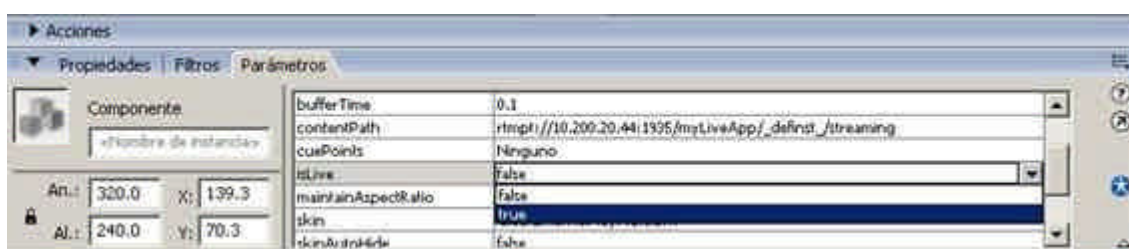


Una vez en el escenario pulsaremos sobre **Parámetros** en la ficha **Propiedades**; el primer parámetro a cambiar será la ruta donde el reproductor podrá encontrar el contenido multimedia, a través de **ContentPath**. La dirección que hemos de introducir tiene una sintaxis muy parecida a la que utilizamos al enviar, pero añadiendo el nombre del stream al final (**rtmpt://dirección\_IP\_Servidor:Puerto/aplicación/instancia/stream**). Así, para un servidor cuya dirección IP sea 10.200.20.44, en una conexión al stream *streaming* en la instancia por defecto *\_definst\_* de la aplicación *myLiveApp* a través del puerto 1935, la ruta del contenido a introducir sería:

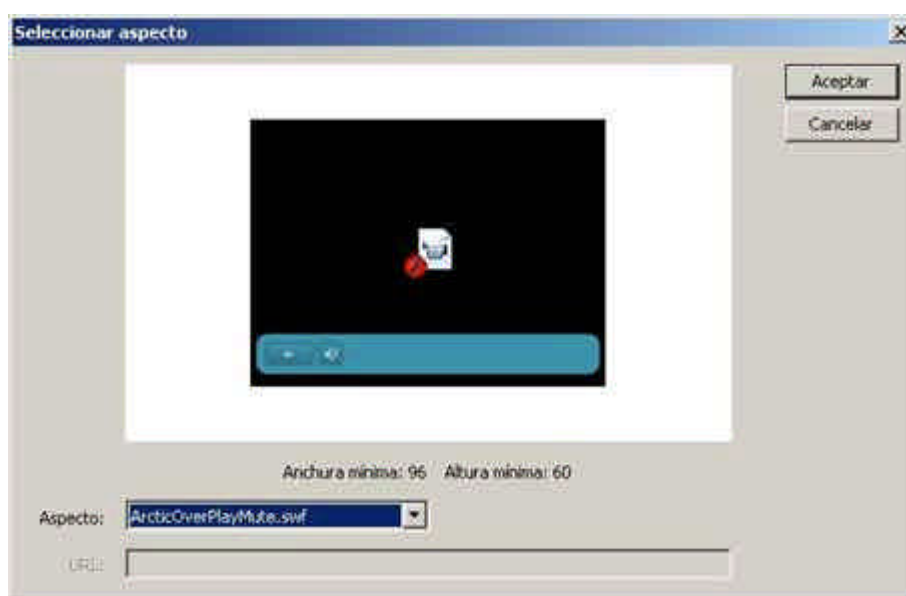




Otro de los parámetros a configurar es definir que el stream es en directo, pues por defecto viene habilitado como *video on demand (VoD)*. Para ello modificaremos el parámetro *isLive* y lo definiremos como *true*.

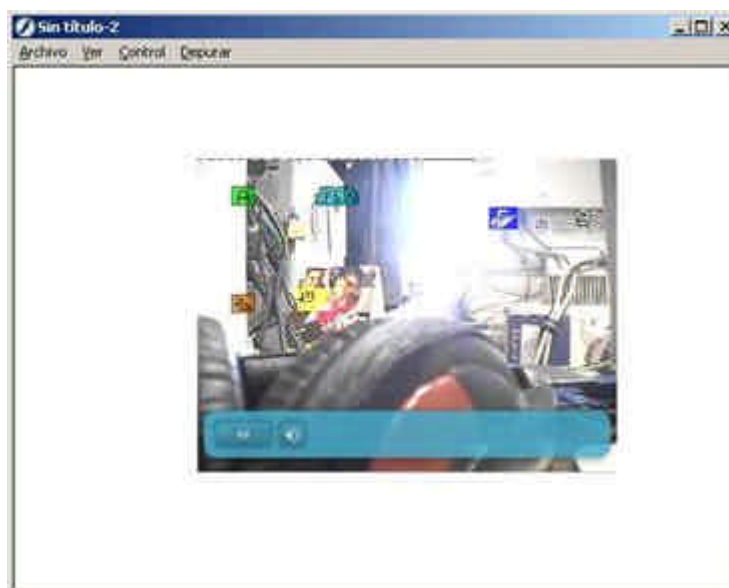
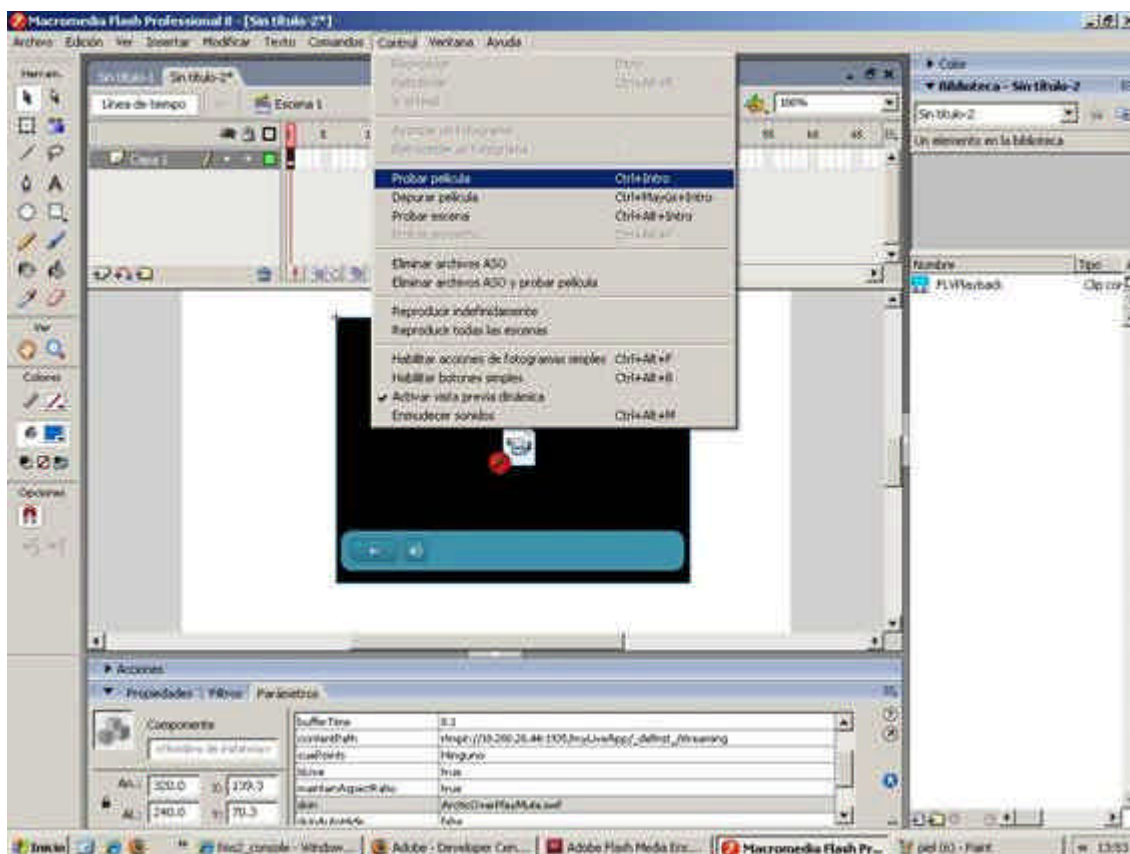


De manera opcional podemos definir el aspecto que tendrá nuestro reproductor a través de la opción *Skin*. Puesto que se trata de un vídeo en directo, la opción más recomendable es que el usuario no tenga acceso a la barra de tiempo. Una piel que realiza esta función es *ArcticOverPlayMute.swf*. Si proporcionamos un reproductor para un contenido guardado, se podría utilizar otra piel para que el usuario pueda avanzar o retroceder a su antojo el contenido multimedia. Incluso podemos personalizar a nuestro antojo una piel para que nuestro contenido sea más exclusivo.



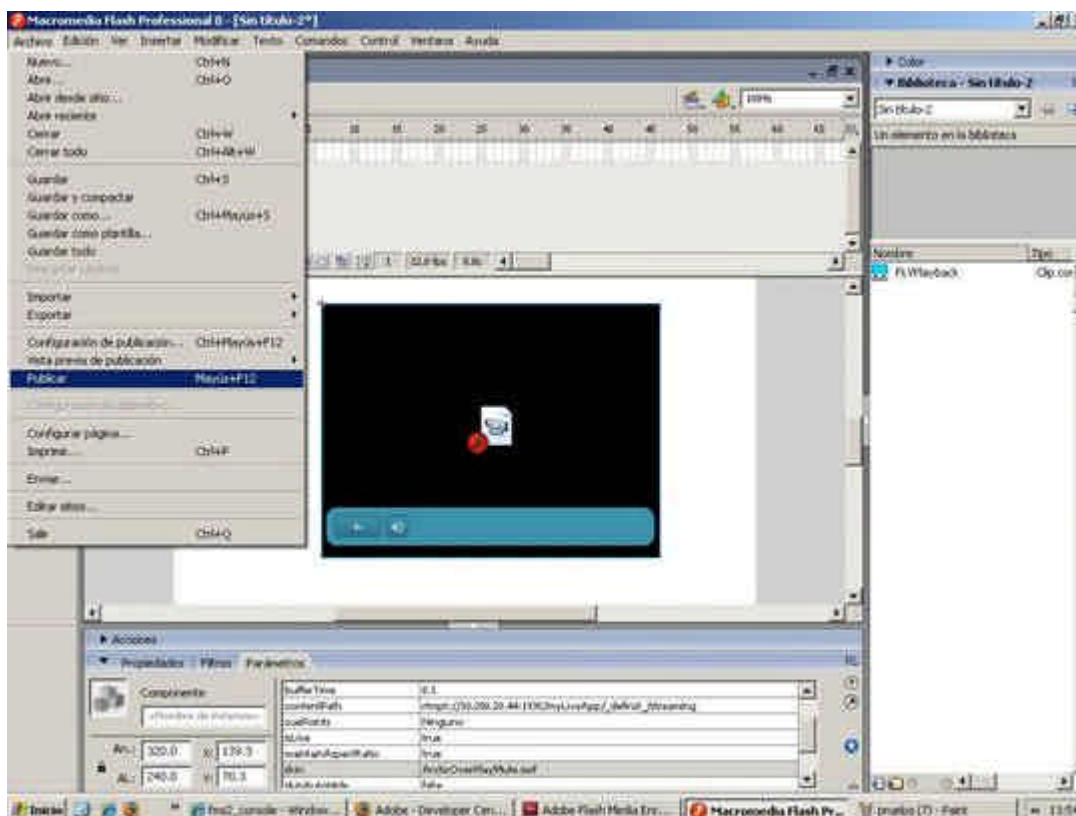
Podemos entonces probar si el vídeo se reproduce correctamente pulsando sobre **Control + Probar Película** o mediante el acceso directo **Ctrl. + Enter**.





Si aparece el reproductor con el vídeo que estamos enviando estamos en disposición de publicar el contenido a través de la opción **Archivo» Publicar** (nota: antes hay que guardar el archivo Flash a través de **Archivo» Guardar**).

Esta acción creará, por defecto, un archivo **.fla**, otro **.swf** y otro **.html** con el reproductor embebido. Ya sólo hemos de subirla al servidor y ofrecer el enlace para que el cliente pueda ver el contenido. Éste sólo necesita un plugin descargable gratuitamente – disponible en Windows, Mac, Unix...- para reproducir el archivo.

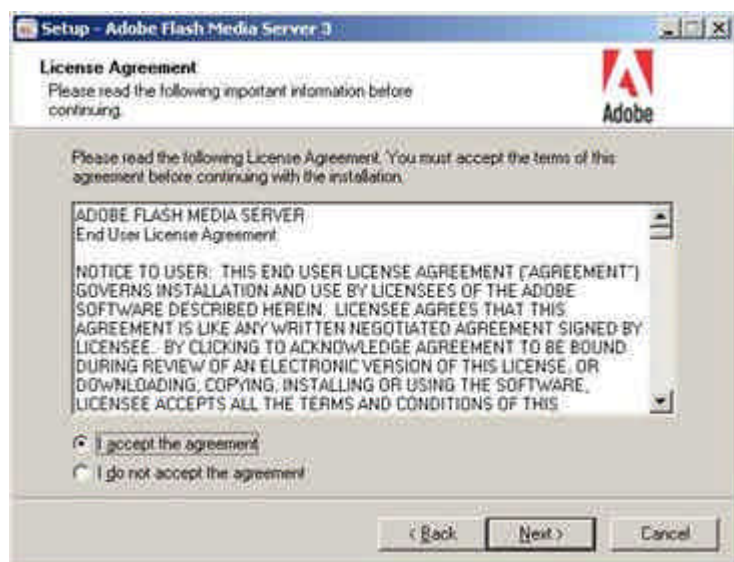


Para más información acerca de los productos de Adobe Macromedia y configuraciones más avanzadas se recomienda acudir a los tutoriales ofrecidos por la compañía

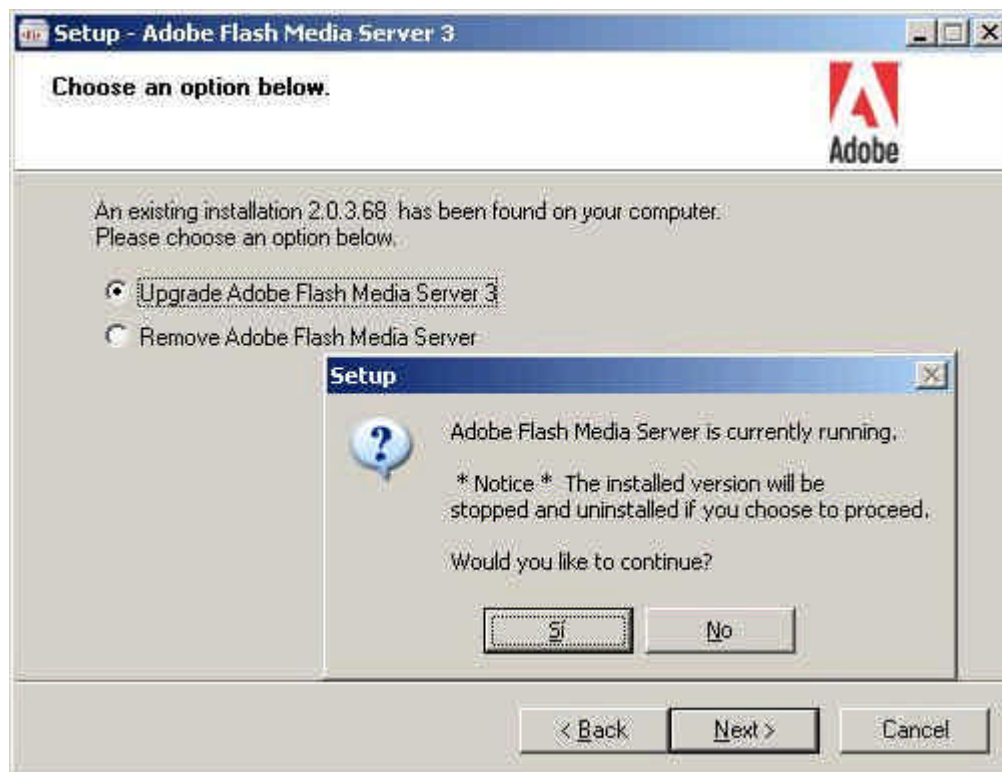
([http://www.adobe.com/devnet/flashmediaserver/articles/webcasting\\_fme.html](http://www.adobe.com/devnet/flashmediaserver/articles/webcasting_fme.html)) y a la documentación en línea.

### **ACTUALIZACIÓN: Flash Media Server 3**

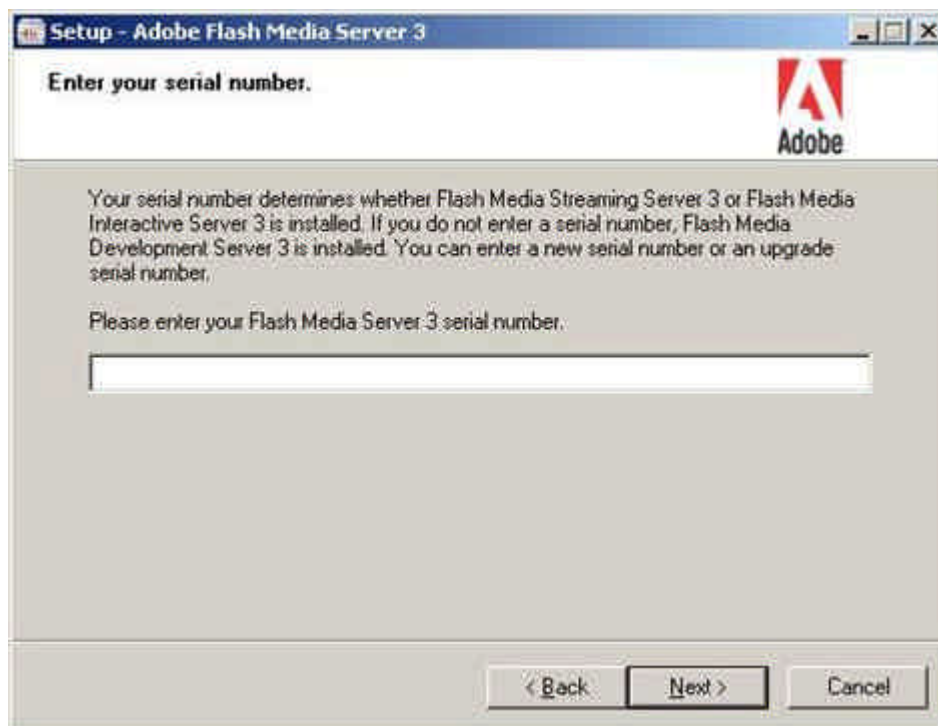
Durante la redacción de este tutorial, Adobe ha anunciado la versión 3 de su popular programa servidor. Sirva este anexo para comentar las pequeñas diferencias con la versión anterior y como descripción de los pasos a seguir para su instalación. Aún así, se recomienda la lectura de los apuntes realizados a la versión anterior, pues muchas de sus características permanecen intactas.



Tras aceptar la licencia pulsando sobre **Next**, el programa de instalación nos ofrece la posibilidad, en el caso de que tengamos FMS 2 (*Flash Media Server 2*) instalado en el mismo equipo, de actualizar el contenido o de borrar la versión antigua completamente. Si se instala la nueva versión por primera vez, este paso no es necesario.



A continuación debemos indicar un número de serie que definirá la licencia que poseemos. Si no introducimos ningún número el programa interpreta que tenemos la licencia básica y, de forma gratuita, podremos realizar streaming para 10 conexiones simultáneas.

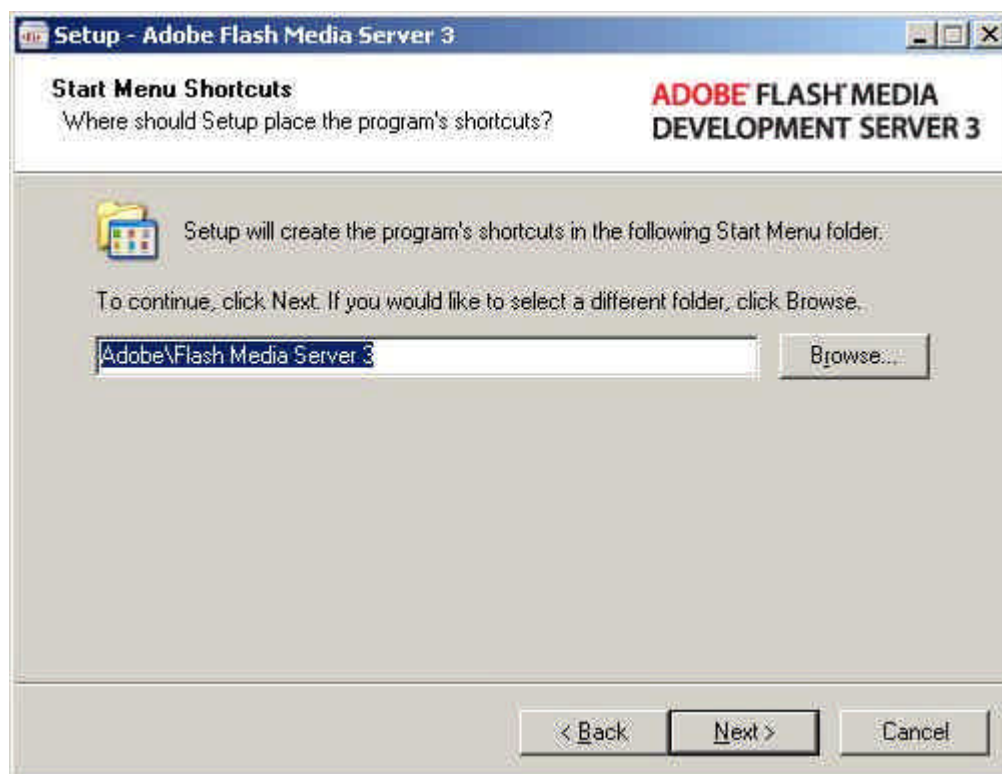




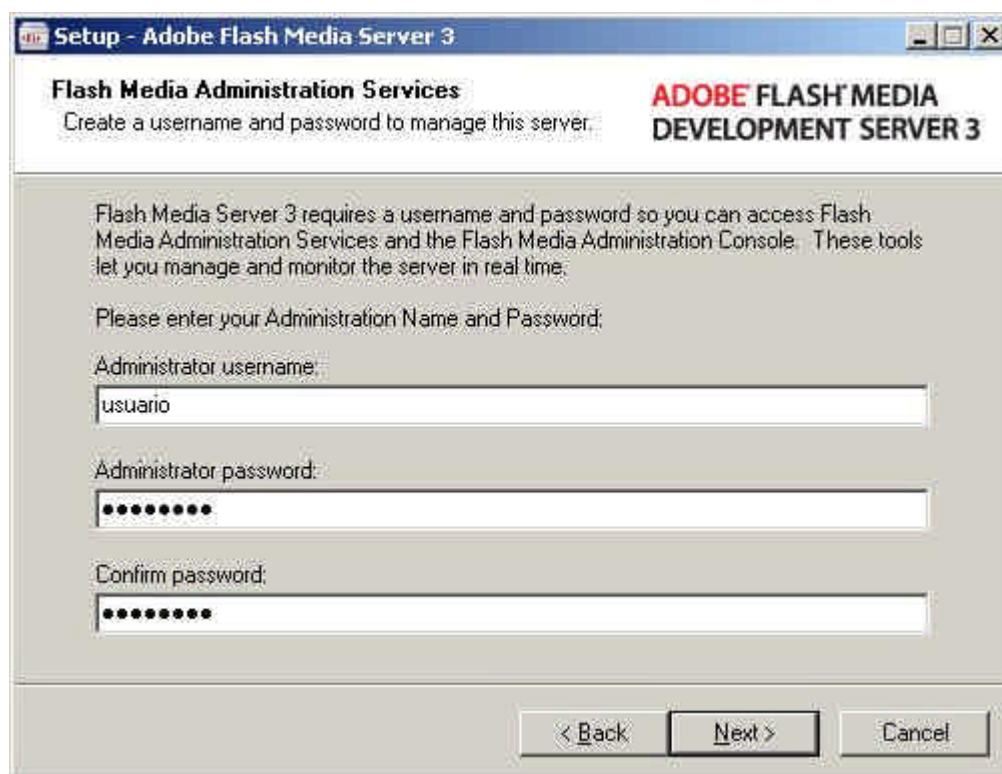
Los siguientes pasos son especificar la ruta de instalación y el grupo del menú inicio que se creará tras la instalación. Podemos dejar los valores por defecto, como indican las siguientes figuras.







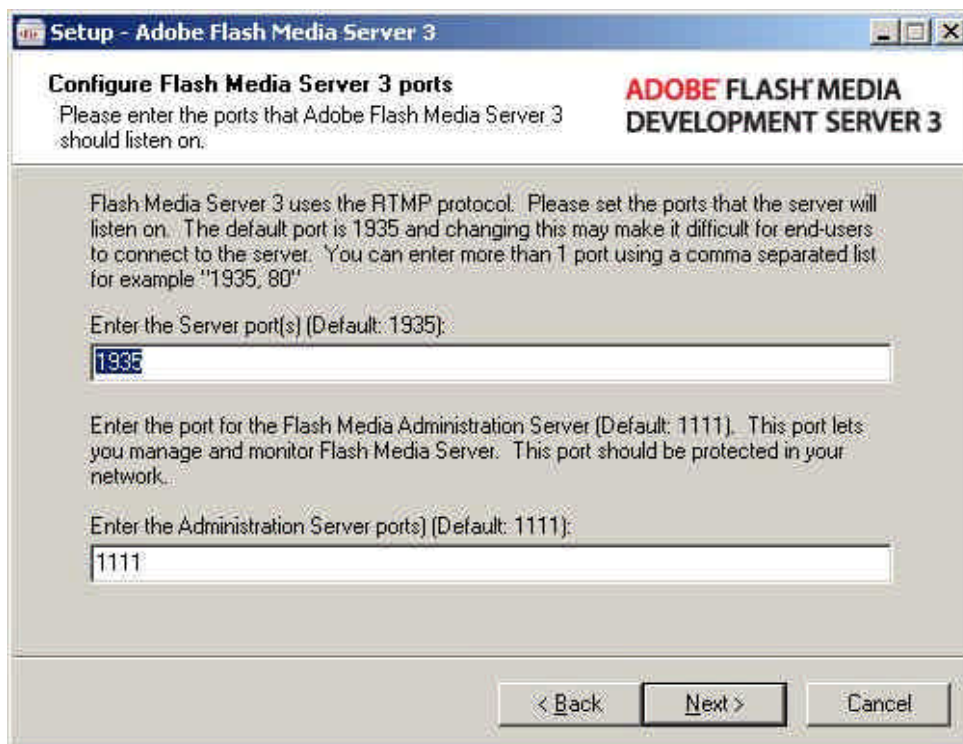
Un paso importante es definir el nombre de usuario y contraseña que utilizará el servidor en la cuenta de administrador. Es necesario recordar estos datos para poder configurar adecuadamente el mismo.



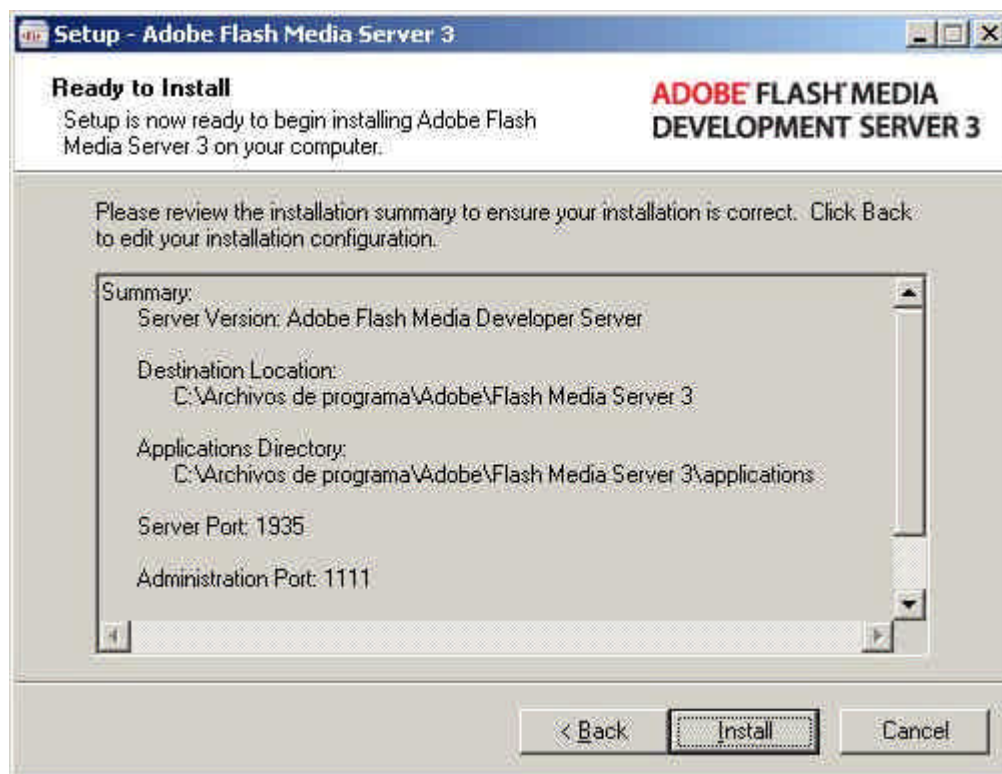
Con los puertos de administración y servidor utilizados por defecto (1111 y 1935, respectivamente) debería ser suficiente, aunque si tenemos varios servidores de



streaming en la misma máquina hay que controlar si se intentan comunicar por los mismos puertos, y cambiar el de alguno si es necesario.



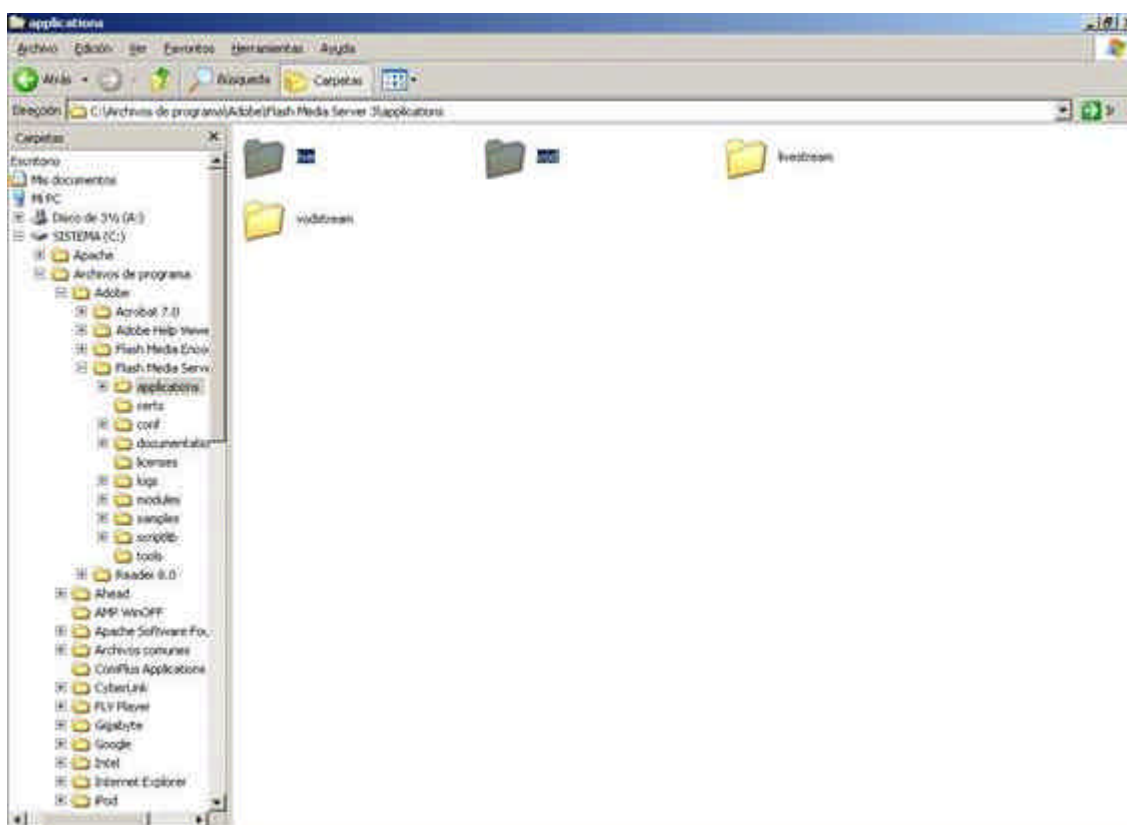
Por último, aparece un cuadro resumen con la configuración que hemos elegido.



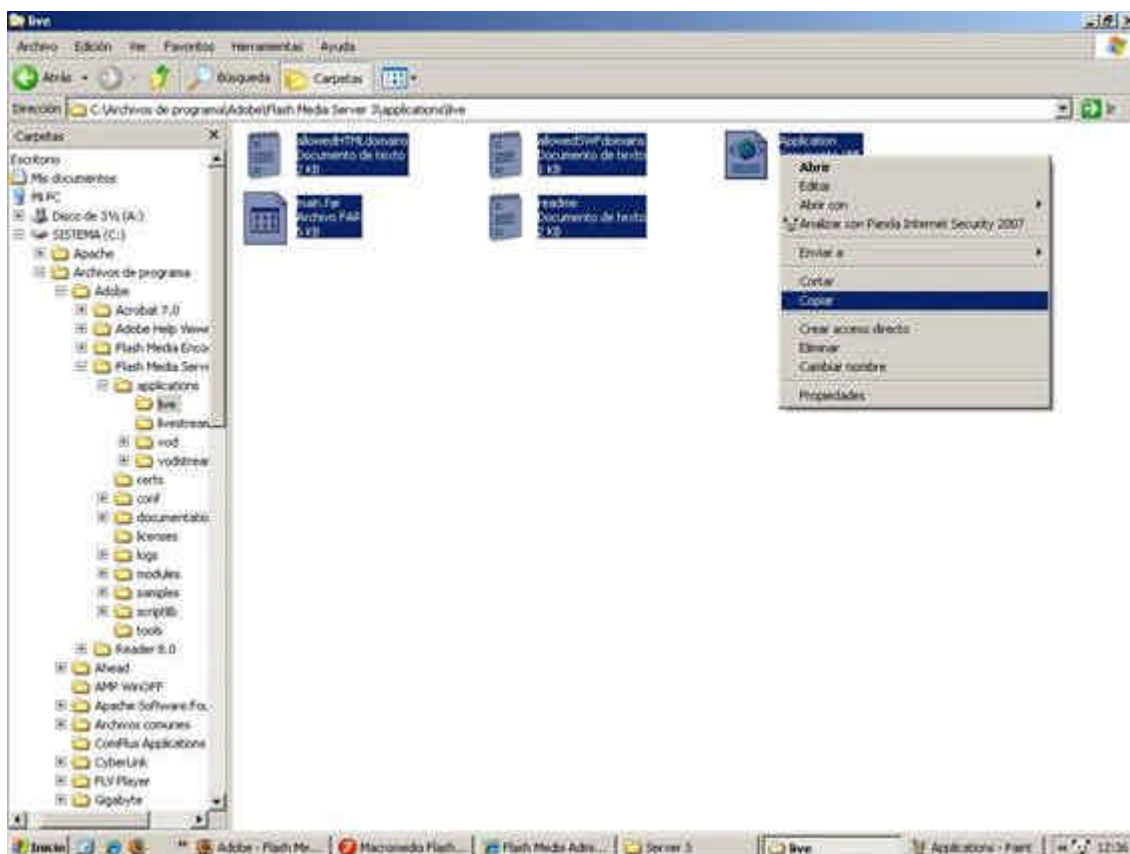
El proceso de arranque del servidor y de la pantalla de administración es idéntico al de la versión anterior, con algunas modificaciones: FMS 3 posee de forma nativa soporte para

streaming bajo demanda y en tiempo real. Para ello existen dos aplicaciones creadas por defecto bajo el nombre de **live** –para streaming en directo- y **vod** –para vídeo bajo demanda-. Dentro de esos directorios vienen los archivos necesarios para funcionar en el servidor.

Una práctica interesante consiste en crear nuestros propios directorios de aplicación para tener las aplicaciones por defecto como respaldo por si se pierden datos. Para ello, creamos dos nuevas aplicaciones en la carpeta **applications** de la ruta donde se haya instalado el Flash Media Server 3 con los nombres que elijamos. En la siguiente figura, éstos son *livestream* y *vodstream*:



Ahora nos meteremos en las aplicaciones por defecto del servidor y copiaremos los datos incluidas en cada una dentro de las carpetas que acabamos de crear. Por ejemplo, para la carpeta *livestream* copiamos los archivos de la carpeta *live* y los pegamos en la nueva aplicación:

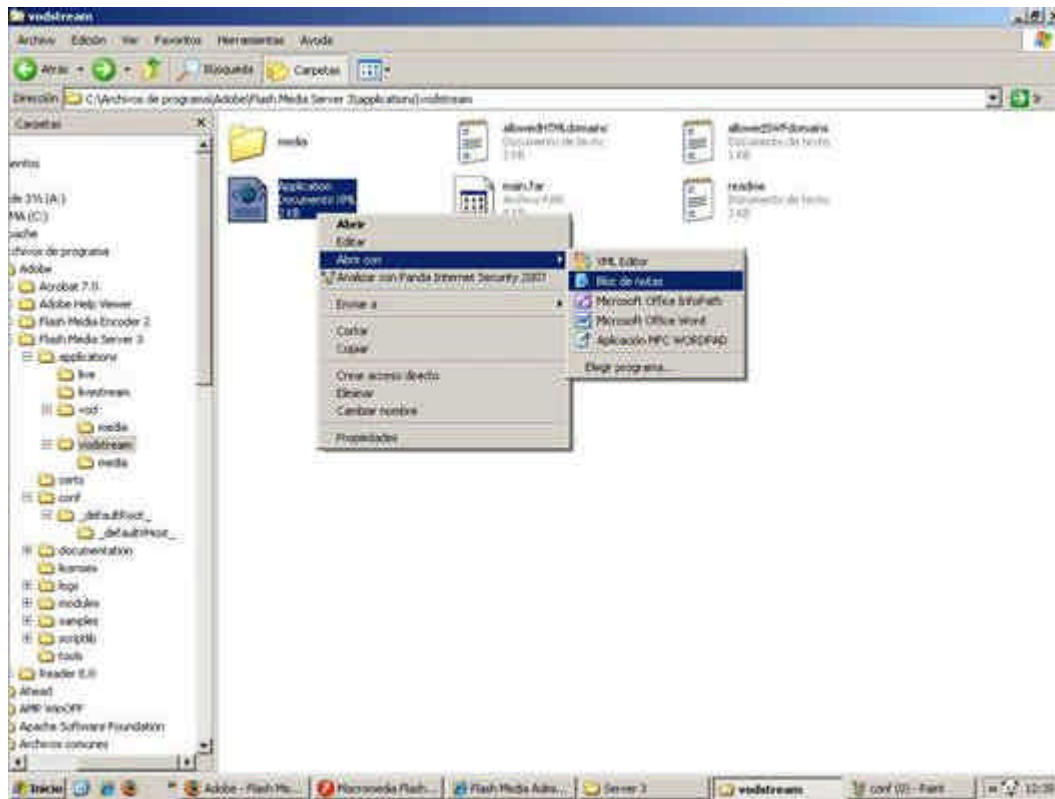


Abriremos a continuación el archivo **fms** del directorio **conf** de la ruta donde hayamos instalado el programa y editaremos las rutas de alojamiento de las nuevas aplicaciones de streaming, en las etiquetas **LIVE\_DIR** y **VOD\_DIR**, respectivamente. Se pueden añadir tantas rutas como se quieran escribiéndolas en líneas separadas una debajo de otra. Puesto que la carpeta de vídeo bajo demanda contendrá los vídeos archivados para su posterior reproducción, se recomienda crear una subcarpeta –de nombre **media**, por ejemplo- para mantener ordenados los vídeos. Si se realiza este paso, hay que especificarlo en el archivo que estamos editando. Este último paso no es necesario en el caso de emisiones en directo, pues el archivo no se almacena en la carpeta correspondiente a la aplicación.

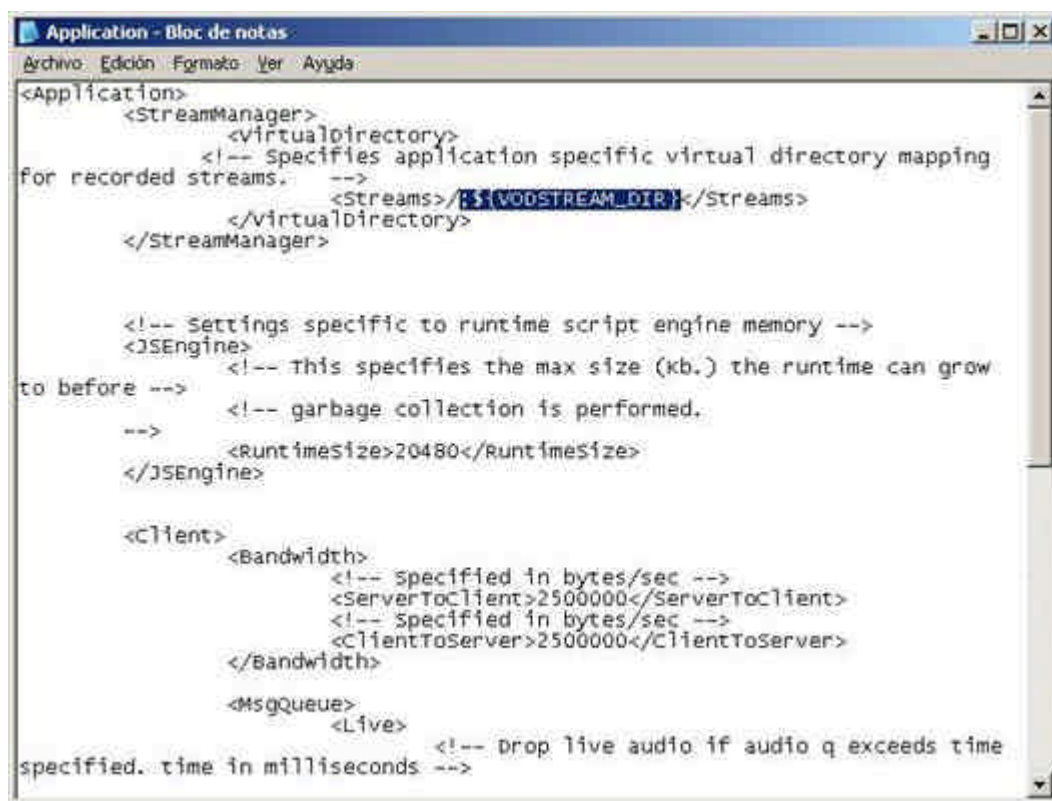
```
fms - Bloc de notas
Archivo Edición Formato Ver Ayuda
SERVER.ADMINSERVER_HOSTPORT = :1111
# User id in which to run the process (Linux only)
# For example:
#   SERVER.PROCESS_UID = 500
#
SERVER.PROCESS_UID =
# Group id in which to run the process (Linux only)
# For example:
#   SERVER.PROCESS_UID = 500
#
SERVER.PROCESS_GID =
# License key for Flash Media Server
# For example:
#   SERVER.LICENSEINFO = FMD200-XXXXX-XXXXX-XXXXX
#
SERVER.LICENSEINFO =
# LIVE_DIR denotes the full path of sample "Live" application's
# folder for storing any live stream recorded by server.
# For example:
#   LIVE_DIR = <FMS_Installation_Dir>\applications\live
#
LIVE_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\live
LIVESTREAM_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\livestream
# VOD_DIR denotes the full path of sample "VOD" application's
# folder for storing onDemand .flv/.mp3 files.
# For example:
```

```
fms - Bloc de notas
Archivo Edición Formato Ver Ayuda
SERVER.PROCESS_GID =
# License key for Flash Media Server
# For example:
#   SERVER.LICENSEINFO = FMD200-XXXXX-XXXXX-XXXXX
#
SERVER.LICENSEINFO =
# LIVE_DIR denotes the full path of sample "Live" application's
# folder for storing any live stream recorded by server.
# For example:
#   LIVE_DIR = <FMS_Installation_Dir>\applications\live
#
LIVE_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\live
LIVESTREAM_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\livestream
# VOD_DIR denotes the full path of sample "VOD" application's
# folder for storing onDemand .flv/.mp3 files.
# For example:
#   VOD_DIR = <FMS_Installation_Dir>\applications\vod
#
VOD_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\vod\media
VODSTREAM_DIR = C:\Archivos de programa\Adobe\Flash Media Server
3\applications\vodstream\media
# The maximum size of the FLV cache, in megabytes.
#The default is 500MB.
#
SERVER.FLVCACHE_MAXSIZE=500
```

Al margen, antes de empezar la transmisión ha de editarse el archivo **Application.xml** dentro de las carpetas de aplicación que acabamos de crear. La única modificación consiste en hacer una referencia a la ubicación de los vídeos, tal y como la hayamos hecho en el paso anterior. Si, por ejemplo, en el archivo **fms** hemos puesto **VODSTREAM\_DIR = Ruta\_del\_directorio**, en el archivo **Application** debemos utilizar la etiqueta **<STREAMS>/:\${VODSTREAM\_DIR}</STREAMS>**.







Una vez configurado el servidor, podemos empezar a transmitir abriendo las aplicaciones correspondientes en el servidor (ver el funcionamiento a través de FMS 2 en apartados anteriores) y acceder al stream a través de la siguiente URL:

***rtmpt://dirección\_IP\_Servidor:Puerto/aplicación/instacia/nombre\_stream***

Así, para un servidor con IP 10.200.20.44 y una comunicación a través del puerto 1935, con una aplicación de vídeo en directo *livecast* bajo la instancia por defecto, *\_definst\_* y un nombre de stream *streaming*, la dirección URL será:

***rtmpt://10.200.20.44:1935/livestream/\_definst\_/streaming***

De igual forma, si se trata de una aplicación de vídeo bajo demanda *vodstream* bajo la carpeta *media* y el nombre de stream *streaming*:

***rtmpt://10.200.20.44:1935/vodstream/media/streaming***

La nueva versión de Flash Media Server tiene soporte, así mismo, para archivos de audio **MP3** y streaming **H.264** en formato **MP4**. Si el archivo utilizado es *.flv* no es necesario especificarlo en la URL. Sin embargo, para el resto de formatos compatibles, debemos hacerlo. Por ejemplo, si el nombre del stream es *streaming.mp3* o *streaming.mp4*, las URL serían:

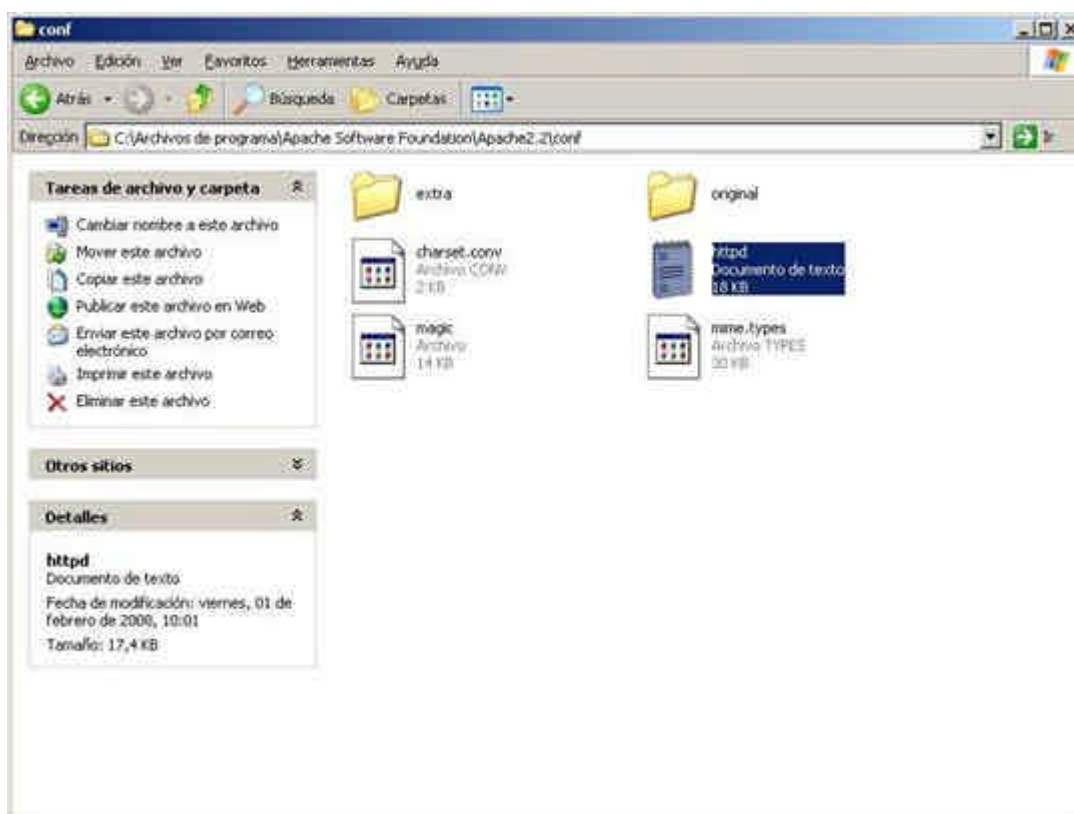
***rtmpt://10.200.20.44:1935/vodstream/media/mp3:streaming***

***rtmpt://10.200.20.44:1935/vodstream/media/mp4:streaming***

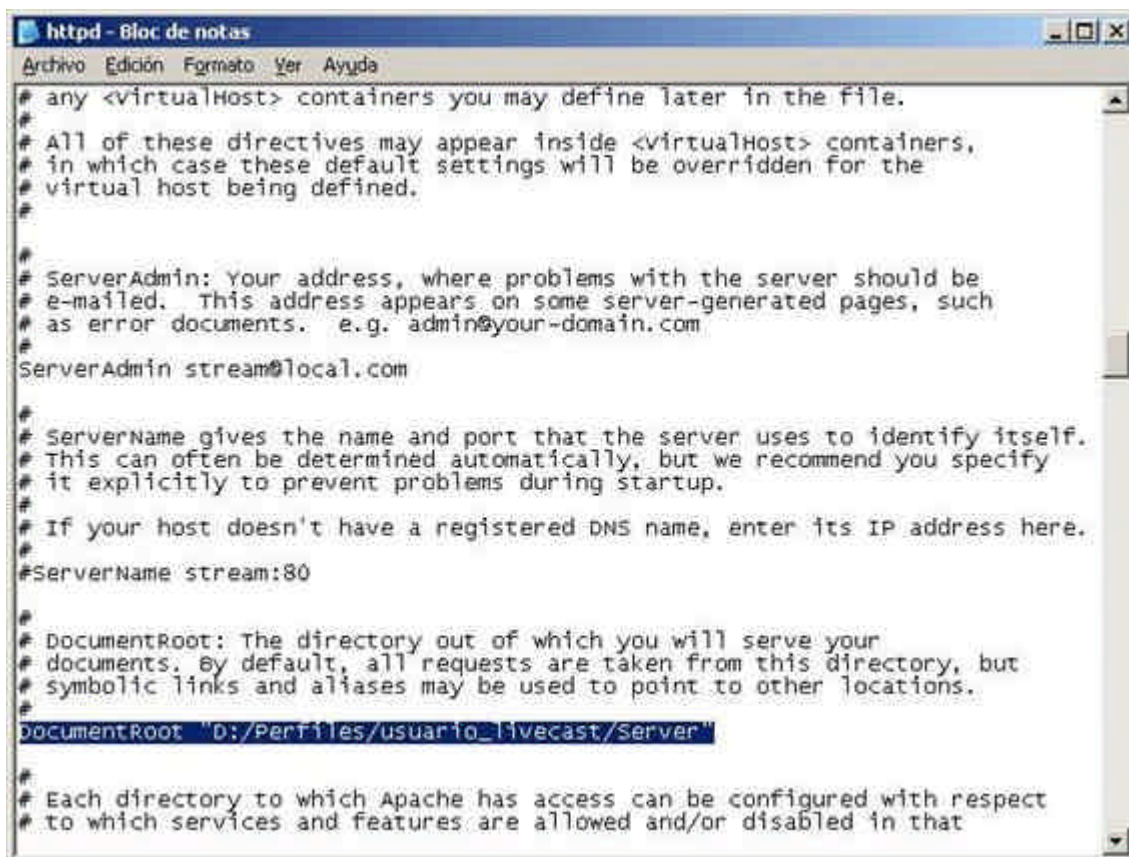


## APACHE SERVER

El último paso antes de poder servir el contenido multimedia es la configuración de un servidor que haga de intermediario entre el Flash media Server y el cliente. Podemos utilizar un servidor Web, pero se aconseja que el servidor y los archivos que ofrece el FMS estén en el mismo equipo. Uno de los servidores gratuitos más conocidos es el **Apache** (<http://www.apache.org/>). Para configurarlo correctamente únicamente habremos de instalarlo y editar el archivo **httpd.txt** en el directorio **conf** de la ruta donde hayamos instalado el servidor. Antes, habremos creado una carpeta en la que almacenaremos los archivos a servir.

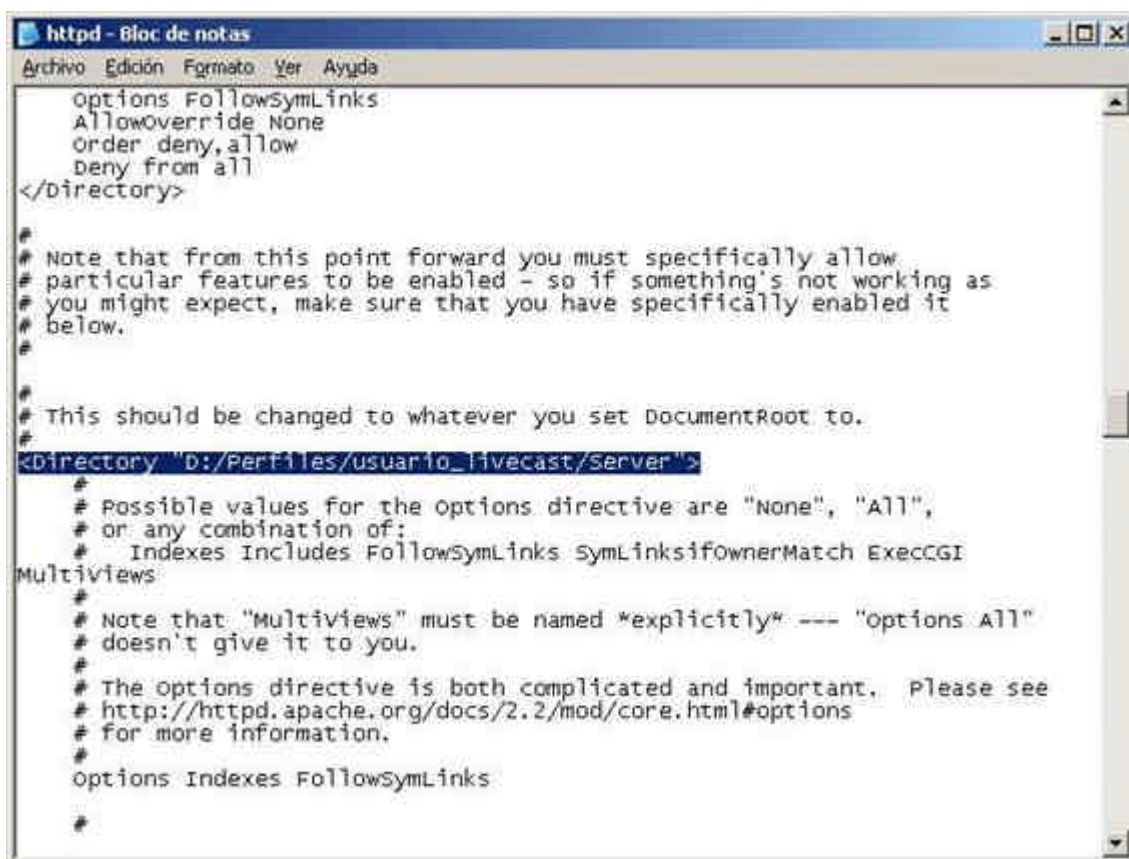


En el archivo **httpd.txt** simplemente hay que editar dos líneas correspondientes a la ruta donde se almacenan los archivos. Así, iremos a la etiqueta **DocumentRoot** y editaremos la ruta de acceso.



```
httpd - Bloc de notas
Archivo Edición Formato Ver Ayuda
# any <virtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <virtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# serverAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin stream@local.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName stream:80
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "D:/Perfiles/usuario_livecast/Server"
#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
```

La otra línea a editar es la que define la etiqueta **Directory**. Habremos de poner la misma ruta que hemos definido en *DocumentRoot*, como indica la siguiente captura de pantalla.



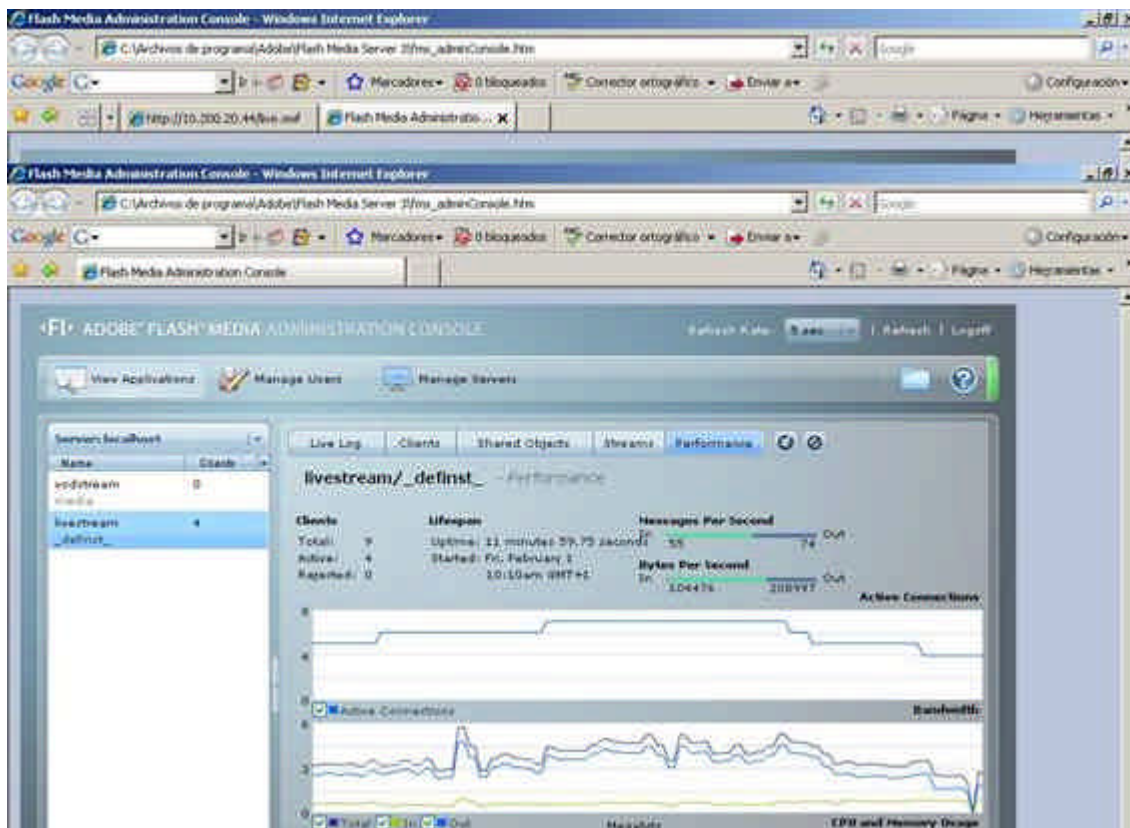
```
httpd - Bloc de notas
Archivo Edición Formato Ver Ayuda
Options FollowSymLinks
AllowOverride None
Order deny,allow
deny from all
</directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# This should be changed to whatever you set DocumentRoot to.
<Directory "D:/Perfiles/usuario_livecast/Server">
# Possible values for the options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI
Multiviews
# Note that "Multiviews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
# The options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
Options Indexes FollowSymLinks
#
```

Con estas sencillas modificaciones tendremos nuestro servidor a punto para almacenar los archivos **.html**, **.swf** y **.fla** que vayamos a publicar para que el cliente pueda acceder a los recursos.



Por último, al igual que hicimos cuando hablábamos de la solución de Real Media, se debe realizar alguna consideración acerca del ancho de banda consumido en la red, sobre todo en unicast, para efectuar el dimensionamiento de la misma. Obsérvese, en la siguiente

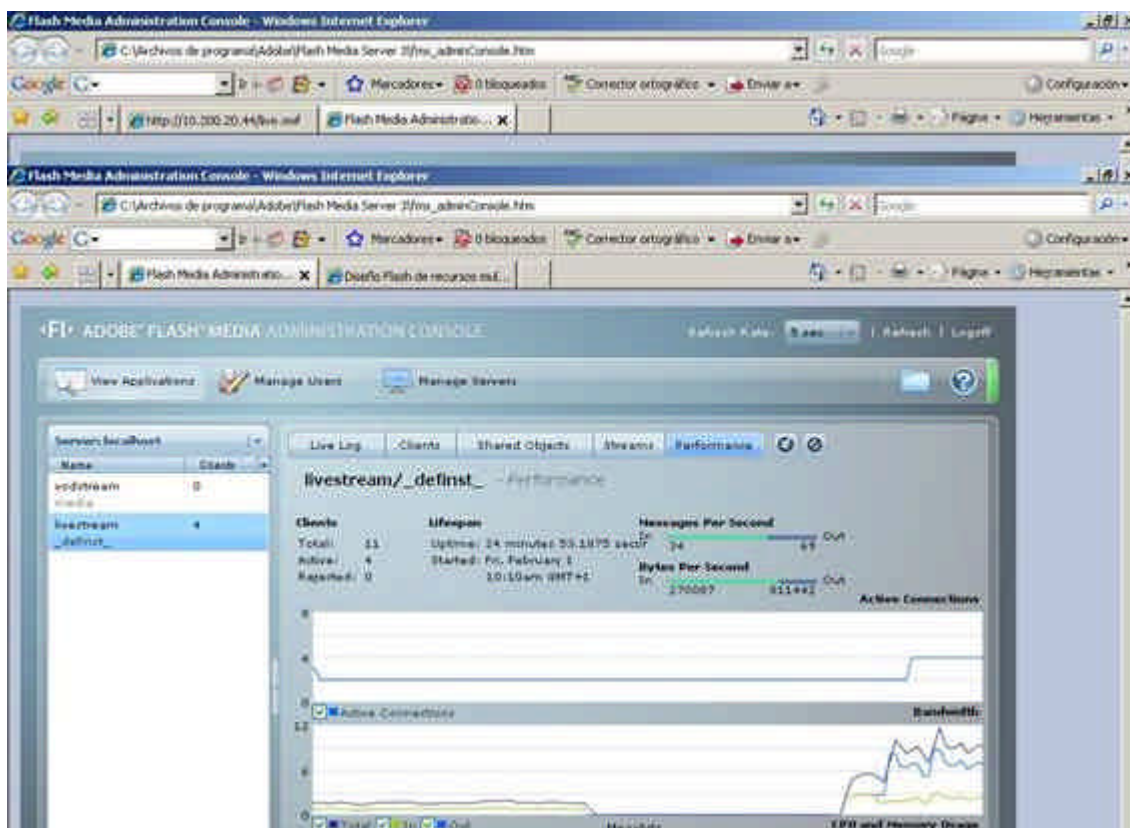
figura, la variación del ancho de banda consumido cuando se codifica a 800kbps en función del número de clientes conectados. En la gráfica de arriba se puede observar el número de conexiones activas –que, en nuestro ejemplo, varía de 4 a 7-, mientras que en la de abajo se muestra el ancho de banda consumido en red. Obviamente, el ancho de banda necesario aumenta cuando el número de conexiones activas aumenta. El ancho de banda necesario se mueve, en este caso, entre 3 y 6Mbps.



Si codificamos, sin embargo, a 2500kbps –la calidad más alta ofrecida por el Flash Media Encoder- el ancho de banda necesario para sostener únicamente 4 conexiones aumenta hasta casi los 12Mbps. Es obvio plantearse a cuántos clientes queremos servir y de cuanto ancho de banda disponemos para calcular la tasa de codificación necesaria para la transmisión.

A veces será más importante la calidad del vídeo que el número de clientes a los que podemos servir, mientras que en otros escenarios se preferirá sacrificar la calidad para que el contenido se pueda ver por un número mayor de usuarios.





## OTRAS OPCIONES

### DARWIN

**Apple** no podía quedarse fuera del ámbito de las soluciones para realizar *streaming* por Internet. Así, la compañía de Steve Jobs posee varias opciones para su sistema operativo Mac OS, así como un servidor Open Source –gratis- que puede funcionar también sobre Windows. Dicho servidor se llama **Darwin Streaming Server**, y puede descargarse gratuitamente tras un sencillo registro desde <http://developer.apple.com/opensource/server/streaming/>. El proceso requiere, sin embargo, que tengamos instalado un lenguaje de scripts llamado **Perl**. Windows XP no viene con soporte para Perl de forma nativa, de manera que habrá que descargarlo e instalarlo desde la página Web de Activestate, <http://www.activestate.com/Products/activeperl/>.

Una vez instalado y seguidas las pautas básicas de configuración el servidor permite servir ficheros en formato *QuickTime* y MPEG-4; el cliente deberá tener instalado, por lo tanto, el programa **QuickTime** para poder reproducir el archivo

(<http://www.apple.com/es/quicktime/download/>), mientras que se requerirá un programa productor si deseamos codificar en directo la señal de una cámara en lugar de servir material pregrabado. Existen distintos programas de Apple para realizar esta última tarea desde Mac –el propio *QuickTime*–, pero hemos de buscar un programa que realice esta función en Windows y se pueda comunicar con el servidor **Darwin**. Una actualización a *QuickTime Pro* –de pago- permite capturar el vídeo y el audio y mandarlo al servidor de streaming.

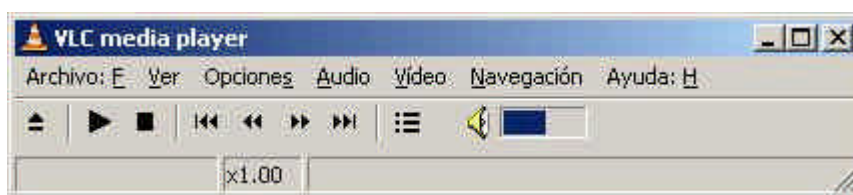
Para más información acerca de cómo poner en marcha el servidor de streaming se

aconseja acudir al documento de soporte del programa en [http://developer.apple.com/opensource/server/streaming/qtss\\_admin\\_guide.pdf](http://developer.apple.com/opensource/server/streaming/qtss_admin_guide.pdf).

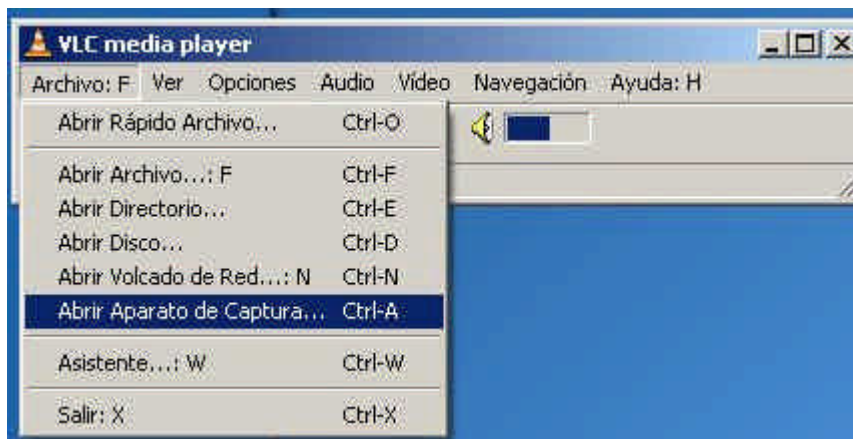
## VideoLAN

VideoLAN ([www.videolan.org](http://www.videolan.org)) es un proyecto que ofrece software libre bajo licencias GNU (General Public License). En este caso, el programa codificador, el servidor y el cliente son el mismo; el **VLC Media Player** permite enviar un flujo multimedia desde un archivo o aparato de captura, actuar de servidor sin más limitación que el ancho de banda de la red y, a su vez, en el cliente, reproducir el contenido de múltiples formatos sin la instalación de *codecs* adicionales.

El reproductor, una vez instalado, tiene una interfaz sencilla a través de la cual podemos acceder a la configuración como cliente, codificador o servidor utilizando un asistente o configuraciones más avanzadas.

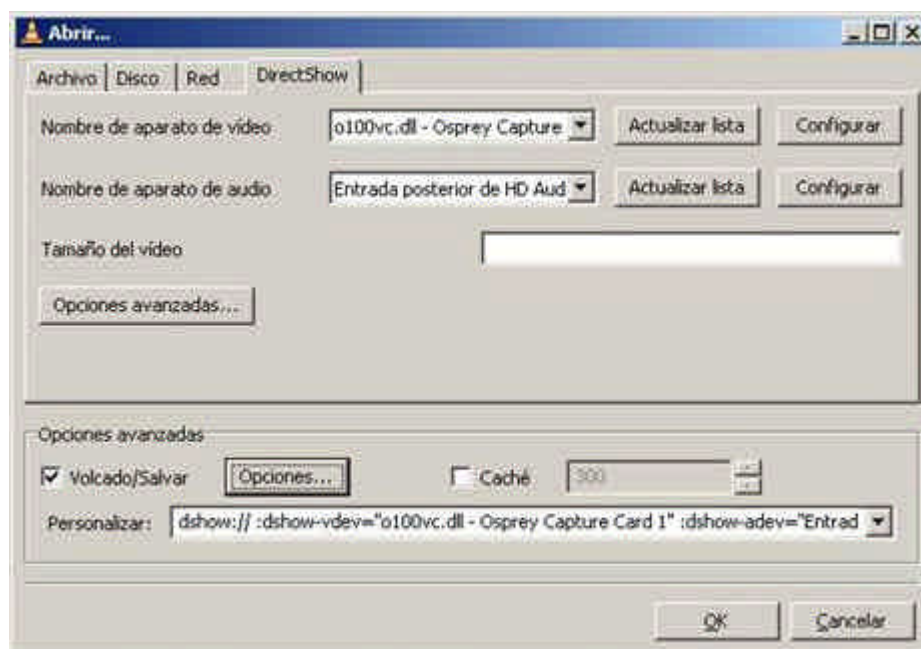


Para comenzar la emisión de un evento podemos conectar la cámara a la tarjeta capturadora y pulsar **Archivo → Abrir Aparato de Captura**; aparecerá entonces un cuadro de diálogo a partir del cual podemos seleccionar los dispositivos de audio y vídeo a partir de los cuales se capturará el vídeo, así como si deseamos realizar el volcado a la red o a otro archivo y en qué condiciones.



La opción **Volcado/Salvar** en la parte inferior de la pantalla viene, por defecto, desmarcada. Si deseamos realizar el *streaming* a otros equipos o la codificación del archivo a otro formato hemos de marcarla y configurarla a través de **Opciones**. Además, haciendo clic sobre **Configurar** cuando seleccionamos los dispositivos de captura podemos cambiar el tipo de codificación y la resolución a la que se efectúa la captura.





Antes de realizar la configuración, sin embargo, hemos de realizar un inciso para preguntarnos qué tipo de emisión queremos realizar. Si se trata de un evento a través de unicast, el rango de direcciones IPv4 van desde la 0.0.0.0 a la 223.255.255.255, mientras que para realizar multicast el rango de direcciones utilizado se mueve entre la 224.0.0.0 y la 239.255.255.255, es decir, mientras que para una dirección unicast hay separación entre la parte de red y la del host, en una dirección multicast sólo debemos usar una máscara inicial para definirla; las direcciones multicast tienen estructura plana, no jerárquica, y sólo es necesario definir la dirección de destino, no de origen:

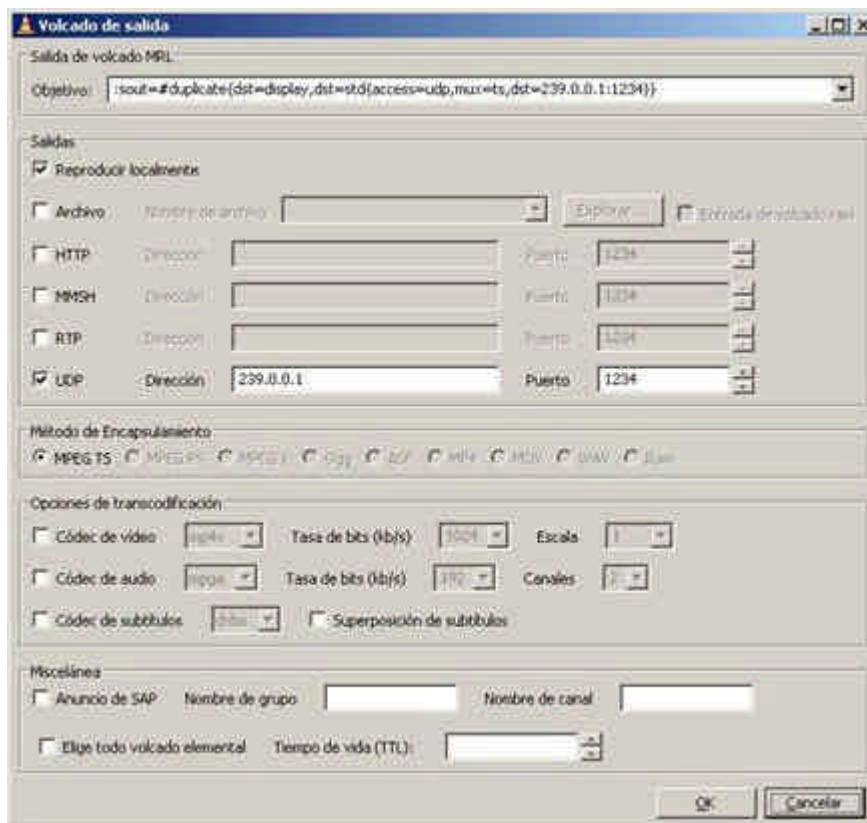
<b>Red</b>	<b>Host</b>
------------	-------------

***Dirección IPv4 Unicast: 0.0.0.0 a 223.0.0.0***

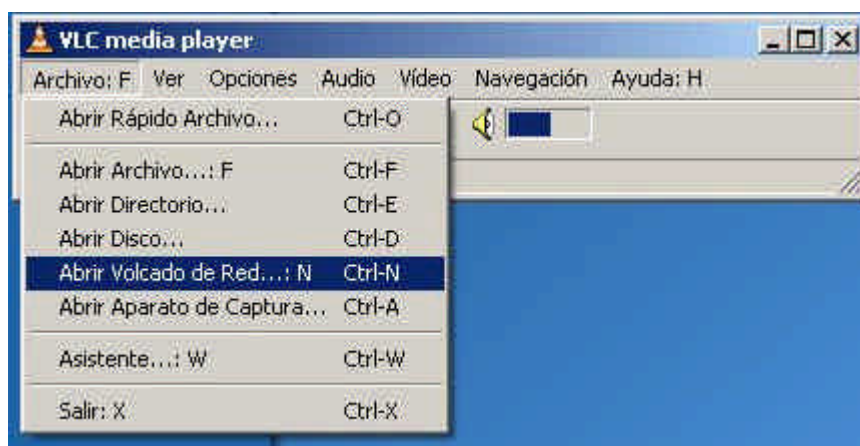
<b>1110</b>	<b>Grupo Multicast (28 bits)</b>
-------------	----------------------------------

***Dirección IPv4 Multicast: 224.0.0.0 a 239.255.255.255***

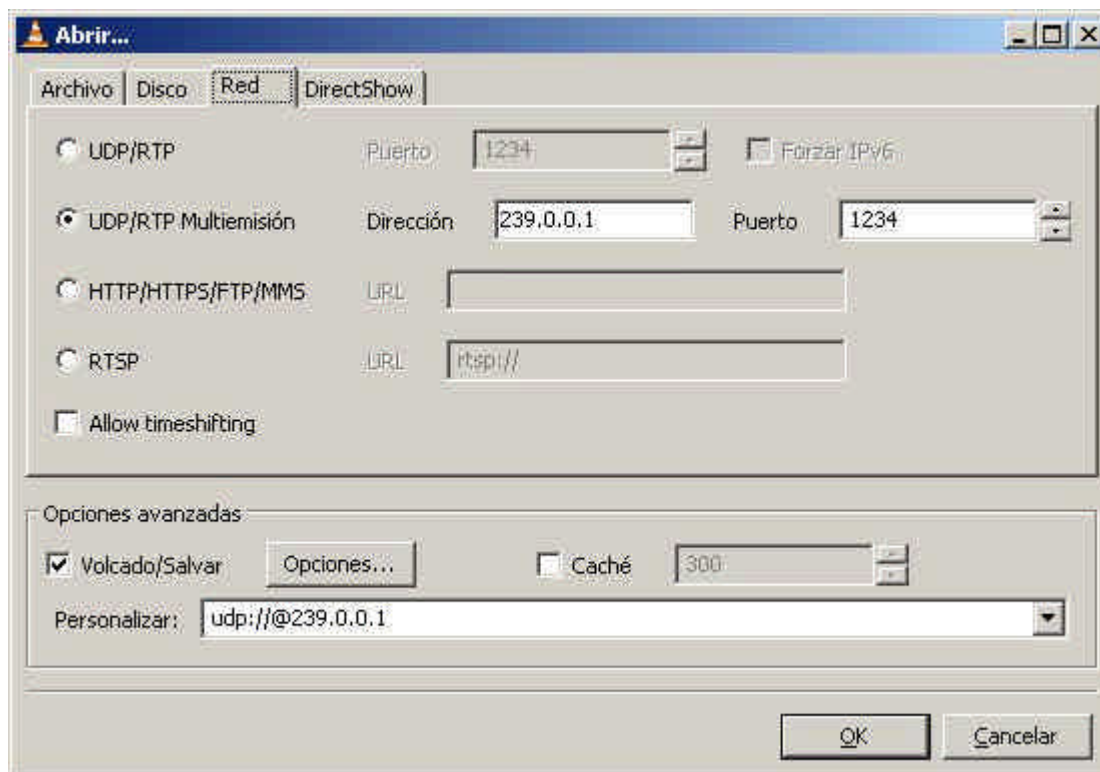
La opción más interesante a utilizar con el VLC Placer es la de multicast, pues no requiere de un archivo de licencias para procesar cuánta gente se conecta al servidor como en las opciones anteriormente analizadas. Por eso nos centraremos en configurar esta opción. Configurando **239.0.0.1** como la dirección destino del grupo multicast, a través del puerto UDP que consideremos oportuno –por defecto **1234**, el cliente debe saber en qué puerto recibirá los datos- ya estaremos preparados para emitir. Se puede marcar la opción **Reproducir localmente** para tener una referencia local de lo que se está capturando/mandando.



A través de este sencillo proceso estaremos emitiendo en multicast con los parámetros que configuramos anteriormente para cada uno de los dispositivos de captura, en los formatos y resolución deseados a través del protocolo UDP. El cliente, en una ubicación remota, sólo ha de tener instalado el reproductor **VLC Player** y pulsar sobre **Archivo** a **Abrir Volcado de Red** para configurar los parámetros de recepción.



Puesto que hemos iniciado en el servidor una conexión UDP a través del puerto 1234 en la dirección IP multicast 239.0.0.1 seleccionamos los mismos parámetros en el cliente y marcamos la opción **Volcado/Salvar**, donde se irán actualizando los datos que hemos metido.



Puesto que con esta configuración el servidor y el productor han de estar en el mismo ordenador y el servidor no es excesivamente configurable –no se trata de un Web Server o de un sistema operativo dedicado, las opciones de configuración son mínimas-, se han de tener en consideración las limitaciones de ancho de banda de las que hablábamos en la solución con *Real Media*, o *Flash*, puesto que será la red del cliente la que limite el número de clientes posibles en función del tipo de codificación, resolución y formato de vídeo utilizados.