

# Webcam-Server



Materia: Laboratorio de Redes y Sistemas Operativos

Profesor: José Luis Di Biase

Integrantes: Calderón, Ingrid

Gerez, Silvana

1° cuatrimestre 2013

## Índice

<b>1. Objetivo</b>	<b>3</b>
<b>2. Introducción</b>	<b>3</b>
<b>3. Requerimientos de software</b>	<b>3</b>
<b>4. Instalación desde terminal</b>	<b>4</b>
• Preinstalación	4
• Instalación de Webcam-server	6
<b>5. Utilización de la herramienta</b>	<b>9</b>
<b>6. Referencias</b>	<b>12</b>
<b>7. Apéndice A</b>	<b>13</b>
<b>8. Apéndice B</b>	<b>14</b>

## **Objetivo**

El propósito que movilizó la confección de este trabajo fue configurar aplicaciones que nos permitan en las máquinas de una red local guardar imágenes tomadas por las cámaras web que se encuentren conectadas.

La aplicación principal que utilizaremos será webcam-server.

La instalación fue realizada en un sistema operativo Linux, específicamente en la distribución Ubuntu (versión 12.04) aunque también se ofrece la alternativa de instalación en Ubuntu 10.04.

Como resultado de esta tarea obtuvimos un informe unificado donde pudimos registrar los errores que pueden llegar a surgir durante la instalación y sus respectivas soluciones ya que hoy en día las alternativas publicadas en la web no se encuentran únicamente en un mismo sitio y en algunos casos no llegan a cubrir las consultas con información adecuada.

## **Introducción a la herramienta**

Según el sitio web oficial de la aplicación Webcam-server es:

“(...) un programa que permite a otros ver la cámara web desde un navegador web.

El programa en sí es un servidor que proporciona una transmisión en vivo de las imágenes a los clientes mediante un applet de Java embebido en una página web.

Webcam\_server utiliza la interfaz video4linux.”

Es necesario aclarar que éste programa realiza capturas de imágenes en formato .jpg y que tendremos que utilizar otro programa para poder unir las imágenes capturadas y crear un video con ellas.

## **Requerimientos de software**

Antes de comenzar a instalar la herramienta principal que utilizaremos para la transmisión de video, debemos verificar tener instalados en nuestra computadora los siguientes programas:

- Apache2: es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.  
Apache nos va a permitir, junto con los paquetes necesarios de Java-Sun, ver nuestra imagen desde la webcam de manera estática y también dinámica.
- Nmap: es un programa de código abierto que se utiliza para efectuar rastreo de puertos . Se usa típicamente para evaluar la seguridad de sistemas informáticos.
- Xawtv: necesario para comprobar el buen funcionamiento de la cámara.
- libv4l-0 y libv4l-dev: son librerías que Ubuntu 12.04 no trae instaladas por defecto y que permiten mostrar video en pantalla.

- VLC media player: es un reproductor y framework multimedia de distribución libre y código abierto. Es un programa multiplataforma con versiones disponibles para varios sistemas operativos.  
VLC es un reproductor de audio y video capaz de reproducir diversos codecs y formatos de audio y video.
- Java 6, específicamente:
  - sun-java6-bin,
  - sun-java6-jre,
  - sun-java6-fonts y
  - sun-java6-plugin.

En caso de no tenerlos, hay dos opciones para la instalación de los mismos:

- 1) utilizar un sistema de gestión de paquetes como, por ejemplo, Synaptic donde simplemente ejecutamos la aplicación con permisos de administrador (para poder realizar cambios en nuestro sistema operativo), buscamos los paquetes que necesitamos instalar, esto nos devuelve una lista con programas que podrían ajustarse a nuestra búsqueda, seleccionamos aquellos que necesitamos y elegimos la opción "Instalar".
- 2) instalar los paquetes necesarios desde la consola de Ubuntu mediante comandos.

## **Instalación desde terminal**

- **Preinstalación**

A continuación enumeraremos cuáles son los programas que debemos instalar y los respectivos comandos con los que se realiza la instalación:

- Apache2

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

- nmap

```
sudo apt-get install nmap
```

- xawtv

```
sudo apt-get install xawtv
```

Una vez instalado, ejecutamos la aplicación para comprobar que la cámara se encuentra funcionando correctamente.

Para ejecutar la aplicación simplemente escribimos la siguiente línea en la terminal:

xawtv

Y debería aparecer una ventana similar a la siguiente:



– Java 6: Oracle (Sun) Java Runtime Environment (JRE) no se encuentra disponible en los repositorios de software oficiales de Ubuntu a partir de la versión 10.04 dado que éste programa no es de código abierto. Entonces para poder instalarlo desde una terminal, tenemos que seguir los siguientes pasos:

1) Instalamos el paquete ubuntu-restricted-extras.

```
sudo apt-get install ubuntu-restricted-extras
```

2) Verificamos que los **repositorios restricted** y **multiverse** estén habilitados. En caso de que estén inhabilitados, realizamos lo siguiente:

- Editamos el archivo encargado de administrar los repositorios con cualquier editor de textos, por ejemplo, nano:

```
sudo nano /etc/apt/source.list
```

- Luego, editamos el *sources.list* suprimiendo los símbolos numerales (#) en cada línea donde aparece *universe* o *multiverse* y guardamos los cambios en el archivo.
- De esa manera ya tendremos todos los repositorios activados, pero siempre que cambiemos la *sources.list* es imprescindible que ejecutemos el siguiente comando:

```
sudo apt-get update
```

Esto nos sirve para actualizar la lista de paquetes que tenemos en nuestro repositorio.

3) Con los repositorios **restricted** y **multiverse** habilitados, ya deberíamos poder instalar los paquetes Java. Para verificar que los paquetes están disponibles, ejecutamos el siguiente comando:

## apt-cache search sun-java6-

Como se puede apreciar en la siguiente imagen, aparecieron los paquetes que necesitamos instalar en nuestro sistema.

```
sun-java6-bin - Sun Java(TM) Runtime Environment (JRE) 6 (architecture dependent files)
sun-java6-demo - Sun Java(TM) Development Kit (JDK) 6 demos and examples
sun-java6-fonts - Lucida TrueType fonts (from the Sun JRE)
sun-java6-javadb - Java(TM) DB, Sun Microsystems' distribution of Apache Derby
sun-java6-jdk - Sun Java(TM) Development Kit (JDK) 6
sun-java6-jre - Sun Java(TM) Runtime Environment (JRE) 6 (architecture independent files)
sun-java6-plugin - Java(TM) Plug-in, Java SE 6
sun-java6-source - Sun Java(TM) Development Kit (JDK) 6 source files
```

- 4) Instalamos los paquetes Java que necesitamos con los siguientes comandos:

```
sudo apt-get install sun-java6-bin
```

```
sudo apt-get install sun-java6-fonts
```

```
sudo apt-get install sun-java6-jre
```

```
sudo apt-get install sun-java6-plugin
```

Luego de la instalación de Java y previo a comenzar la instalación de webcam-server, debemos tener una cámara web instalada, configurada y probada. Para que todo esto funcione con normalidad, la webcam debe ser compatible con v4l / v4l2 que es una API de captura de video para Linux. Video4Linux (v4l) está integrado con el núcleo Linux.

### • Instalación de Webcam-server

- Webcam-server: cuando tengamos nuestra cámara funcionando, empezamos con la instalación de webcam-server.

La forma de instalación de la herramienta va a depender de la versión de Ubuntu que utilizemos como Sistema Operativo.

Si utilizamos Ubuntu 10.10 o menos, simplemente debemos ejecutar desde una terminal el siguiente comando:

```
sudo apt-get install webcam-server
```

Pero a partir de la versión 11.04 o más, la aplicación webcam-server dejó de estar disponible a través de los repositorios oficiales de Ubuntu. Ahora existe un archivo ".deb" que podemos descargar desde el sitio de Launchpad (plataforma de desarrollo colaborativo de software, en particular el de software libre como un servicio gratuito) a través del siguiente link:

<<https://launchpad.net/ubuntu/gutsy+package/webcam-server>>.

Después de descargar el paquete correspondiente, podemos instalarlo simplemente haciendo doble click sobre el archivo y utilizando el centro de software de Ubuntu.

Luego de instalar la aplicación webcam-server, debemos modificar el fichero de configuración de la misma.

Esta tarea la realizaremos de la siguiente manera (nosotros usamos Nano para editar el texto, pero podemos llevar a cabo esta acción con cualquier otro editor):

```
sudo nano /etc/init.d/webcam-server
```

Al abrir el fichero encontraremos que éste se encuentra vacío. Lo que debemos hacer es completarlo con los datos que se encuentran en el Apéndice A.

Una vez copiado el texto del apéndice en el fichero, guardamos los cambios y, a continuación, le asignamos permisos de la siguiente manera:

```
sudo chmod 777 /etc/init.d/webcam-server
```

Luego, iniciaremos los scripts de inicio y apagado automático con el siguiente comando:

```
sudo update-rc.d webcam-server defaults
```

Y con esto sólo basta iniciar la aplicación:

```
sudo /etc/init.d/webcam-server start
```

Para ver una imagen de manera estática, accedemos a la ruta web mediante el link que se encuentra a continuación:

< <http://localhost:8888> >

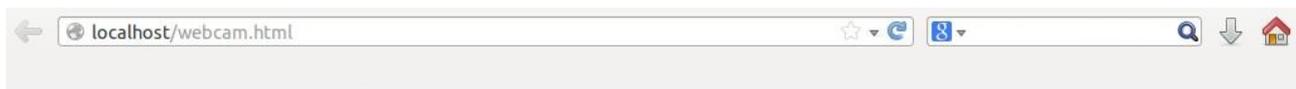
Ahora resta poder visualizar imágenes de manera dinámica en nuestro servidor, es decir, en forma de "video" copiamos los ficheros a nuestra ruta por defecto del apache:

```
sudo cp /usr/share/doc/webcam-server/applet/* /var/www/
```

Una vez ejecutado este comando, en el directorio /var/www/ aparecerán una serie de carpetas y archivos que se crearán por defecto, entre ellos encontraremos un archivo llamado webcam.html (ésta es la máquina virtual Java).

Ahora bien, para poder visualizar las imágenes en movimiento tenemos que acceder a:

< <http://localhost/webcam.html> >



Camara Vigilancia com WebCam-Server



El siguiente paso es crear, en la misma carpeta donde tenemos el applet de Java, un nuevo fichero al que llamaremos "miWebCam" (o el nombre que más prefieran) y lo llenaremos con los datos que se encuentran en el Apéndice B.

Este fichero tendrá la particularidad de que en el mismo le asignaremos nuestra dirección IP pública. Esto es necesario para que, en caso de ser requerido, se puedan ver las imágenes de nuestra webcam en cualquier computadora conectada a la misma red o desde internet. Pero además de llenar este fichero con nuestra IP, entre otras cosas, es necesario abrir un puerto.

Sin embargo, no debemos desestimar que cada vez que abrimos un puerto ésta acción trae asociado un peligro latente y por lo tanto hay que tomar los recaudos necesarios asegurizando la red para evitar ser víctima de alguna intrusión.

Dado que este tema excede los objetivos de este trabajo, a continuación dejaremos el siguiente link donde puede conseguir información al respecto:

< <http://www.ubuntu-es.org/node/129680> >

Luego de creado el fichero miWebCam.html, cada vez que queramos acceder a la webcam, copiamos en la URL del navegador la siguiente ruta:

< <http://IPpública:puertoAbierto/miWebcam.html> >

En esta instancia podremos tener acceso a nuestra herramienta desde nuestro host local < <http://localhost/webcam.html> > o desde el applet de Java <<http://IPpública:puertoAbierto/miWebcam.html> >

## Utilización de la herramienta:

Luego de que la herramienta webcam-server se encuentre instalada y funcionando correctamente, vamos a configurarla para poder hacerla funcional a nuestras necesidades.

En nuestro trabajo vimos conveniente poder guardar la imágenes que la cámara muestra con una respectiva etiqueta que nos de la fecha y la hora que fue tomada.

En esta sección abordaremos cómo realizar el almacenamiento de las imágenes capturadas por segundo de la webcam y luego también explicaremos cómo crear un "video" de dichas imágenes.

Para poder unir las imágenes .jpg que toma webcam-server al estar activada, utilizaremos un programa llamado VLC media player.

Para instalarla, abrimos la terminal y ejecutamos el siguiente comando:

```
sudo apt-get install vlc
```

Una vez instalado el programa, podremos acceder a la herramienta desde una interfaz gráfica o desde terminal escribiendo vlc.

También utilizaremos los siguientes tres comandos:

- **wget:** "GNU Wget es una herramienta libre que permite la descarga de contenidos desde servidores web de una forma simple".
- **watch:** " es un comando de la familia de los Sistemas Operativos Unix y sirve para ejecutar periódicamente un comando"
- **date:** "El comando **date** es un comando existente en sistemas Unix que muestra la hora y la fecha del sistema y el administrador también puede cambiarla". Nosotros usaremos del comando date las especificaciones %d-%B-%Y que nos muestran el día, el mes, año y también el %X que nos muestra la hora, en horas minuto y segundos.

Para tener una mayor cantidad de información respecto a las acciones que pueden llevar a cabo con los comandos descriptos con anterioridad hay que leer los manuales de los mismos con un simple comando:

```
man nombreDeComando
```

Utilizando cualquier editor de texto, transcribimos el siguiente script:

```
echo
```

```
watch -n 1 'wget localhost:8888 ; a=`date +%d-%B-%Y` b=`date +%X`;  
mv index.html $a-$b.jpg'
```

Este script nos permitirá poder ver la fecha y el horario en el que las imágenes fueron tomadas. Esto nos facilitará, más adelante, realizar la tarea de seleccionar las fotos de un determinado lapso de tiempo.

Cuando terminamos de pasar el texto del script en un archivo de texto, debemos guardarlo con la extensión .sh para, luego, asignarle permisos de ejecución. Esto último lo realizamos de la siguiente manera:

```
sudo chmod 777 nombreDelArchivo.sh
```

Posteriormente a asignarle permisos al script, iniciamos webcam-server. Ahora, nos ubicamos en la ruta donde guardamos nuestro archivo de extensión .sh y debemos correrlo desde terminal (previamente a ubicarnos en la ruta donde hemos guardado el archivo). Luego, ejecutamos el script.

```
cd /directorioDondeEstaElArchivo/
```

```
sudo ./nombreDelArchivo.sh
```

Lo que hará la ejecución de este script, es guardar en formato .jpg las imágenes capturadas por la cámara web, en el directorio donde nos paramos anteriormente, cada segundo desde que lo iniciamos con el comando comentado.

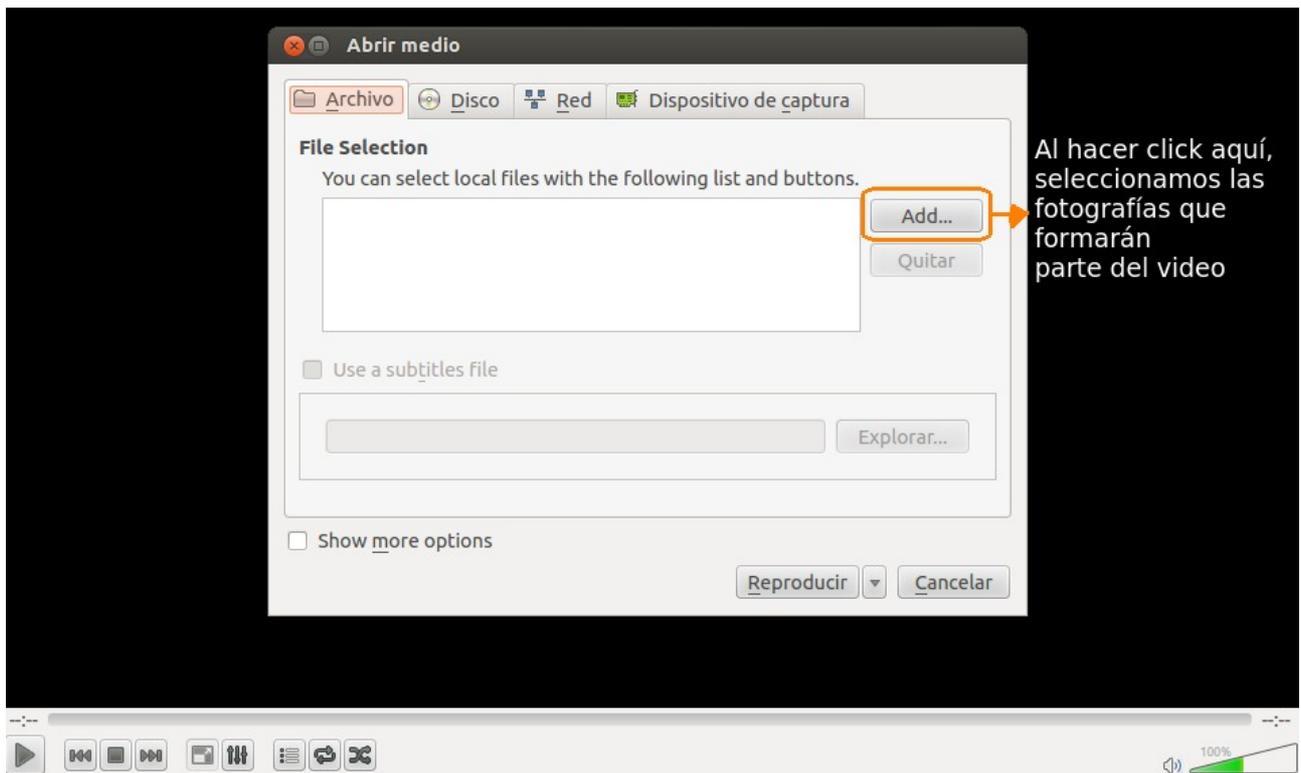
Ahora recordemos un poco la herramienta que descargamos previamente: VLC (como hemos comentado, este programa nos ayudará a unir las imágenes capturadas por webcam-server y podremos crear un video con ellas).

Tener que mirar una imagen por una para poder saber qué fue lo que capturó la aplicación en un horario determinado es demasiado tedioso y muy poco práctico. Por lo tanto, una vez que hayamos guardado nuestras imágenes en un directorio con la utilización de VLC podemos obtener un video de una manera muy sencilla.

En primer lugar, debemos ingresar a la herramienta. Esto lo podemos realizar de dos maneras:

- Ingresando por terminal.
- Utilizando interfaz gráfica.

Una vez que hayamos ingresado a la aplicación, hacemos click derecho y seleccionamos la opción "Abrir medio". Así, se abrirá un menú desplegable del cual debemos elegir la opción "Abrir archivo" y se nos abrirá la siguiente pantalla:



Ahora seleccionaremos las fotos que deseamos que se encuentren en nuestro video y elegimos la opción "Abrir". Esto hará que retornemos a la pantalla anterior donde vamos a seleccionar "Reproducir" y nos mostrará las imágenes seleccionadas simulando ser un video.

## **Referencias**

- Sitio web de webcam-server  
<http://webcamserver.sourceforge.net/>
- Comunidad Ubuntu en español  
<http://www.ubuntu-es.org/>
- Wikipedia
  - <http://es.wikipedia.org/wiki/Date>
  - [http://es.wikipedia.org/wiki/GNU\\_Wget](http://es.wikipedia.org/wiki/GNU_Wget)
  - [http://es.wikipedia.org/wiki/Watch\\_%28Unix%29](http://es.wikipedia.org/wiki/Watch_%28Unix%29)
- DASIEC  
<https://sites.google.com/site/dasiecinc/news-and-blogs/linux-tips/servidorwebcamenubunturapidoyfacil>
- Videovigilancia con Webcam-Server  
<http://www.youtube.com/watch?v=pbGgglNlswU>
- Keopx.net  
<http://www.keopx.net/blog/hacer-streaming-con-webcam-en-debianubuntu>

## Apéndice A

```
#!/bin/sh
SERVER_BIN=webcam-server
LOCK_FILE=/var/lock/$SERVER_BIN
RTRN=0
OPTIONS='-v -g 320x240 -p 8888 -c hacktivation.com'
start() {
[ -f $LOCK_FILE ] && echo "$SERVER_BIN already started"
[ -f $LOCK_FILE ] && return
echo -n "Starting $SERVER_BIN: "
export LD_PRELOAD=/usr/lib/i386-linux-gnu/libv4l/v4l1compat.so
nohup $SERVER_BIN $OPTIONS > /dev/null 2>/dev/null &
RTRN=$?
[ $RTRN -eq 0 ] && echo Started! || echo FAIL
[ $RTRN -eq 0 ] && touch $LOCK_FILE
}
stop() {
[ -f $LOCK_FILE ] || echo "$SERVER_BIN is not running"
[ -f $LOCK_FILE ] || return
echo -n "Stopping $SERVER_BIN: "
pkill -f "$SERVER_BIN $OPTIONS"
RTRN=$?
rm -f $LOCK_FILE
[ $RTRN -eq 0 ] && echo Stopped! || echo FAIL
}
case "$1" in
start)
start
;;
stop)
stop
;;
restart)
stop
start
;;
*)
echo "Usage: $0 {start|stop|restart}"
RTRN=1
esac
exit $RTRN
```

## **Apéndice B**

```
<!DOCTYPE html PUBLIC "-//W3E//DTD HTML 4.01 //EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>

<title>miwebcam</title>

<meta http-equiv="refresh" content="1">
</head>
<body>

</html>
```