

BANCO DE PRUEBAS PARA EL SOPORTE DEL SERVICIO DE VIDEO BAJO DEMANDA DE CONTENIDOS MULTIMEDIA ADAPTATIVOS, BASADOS EN EL ESTANDAR DASH (ISO/IEC 23009-1).



DUVERNEI ORTIZ TRUJILLO

Trabajo de grado en Ingeniería Electrónica y Telecomunicaciones

Director:

Mag. Gabriel Elías Chanchí Golondrino

Co-director:

PhD. José Luis Arciniegas Herrera

**Universidad del Cauca
Facultad de ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación en Servicios Avanzados de Telecomunicaciones
Popayán, diciembre de 2015**

DUVERNEI ORTIZ TRUJILLO

**BANCO DE PRUEBAS PARA EL SOPORTE DEL
SERVICIO DE VIDEO BAJO DEMANDA DE
CONTENIDOS MULTIMEDIA ADAPTATIVOS,
BASADOS EN EL ESTANDAR DASH
(ISO/IEC 23009-1).**

**Tesis presentada en la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de:**

**Ingeniero en
Electrónica y Telecomunicaciones**

**Director:
Mag. Gabriel Elías Chanchí Golondrino**

**Co-director:
PhD. José Luis Arciniegas Herrera**

**Popayán
2015**

Anexo A

Instalación de las herramientas para la generación de contenidos multimedia DASH

Las herramientas que han sido seleccionadas en el presente trabajo, para el proceso de generación de contenidos multimedia DASH, son las siguientes:

- FFmpeg
- Libwebm
- Webm-tools

Estas herramientas fueron compiladas e instaladas en el sistema operativo Linux OpenSuse 13.1 KDE. A continuación se ilustra el proceso de configuración de cada una de ellas:

A.1 Instalación de FFmpeg

Desde un navegador web primero se ingresa a la siguiente dirección:

help.opensuse.org/ksuseinstall/

Una vez en la página desplegada, se procede a dar clic sobre la opción: *Multimedia Software*, ver la figura a.1.1.

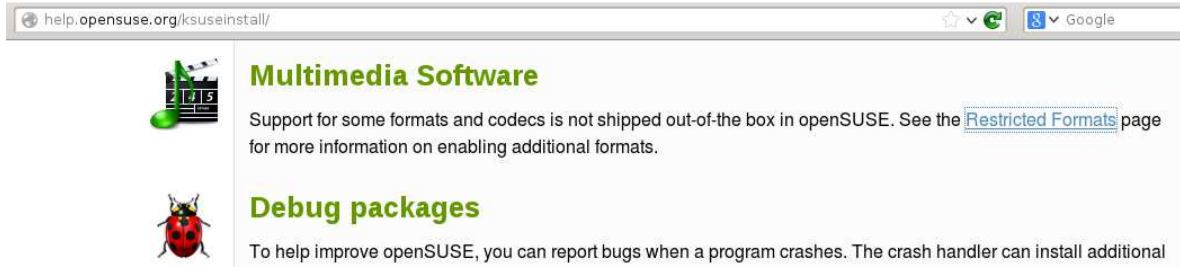


Figura A.1.1. Software disponible para OpenSuse.

En la siguiente página, se selecciona la opción *KDE Codecs*, como se indica en la figura a.1.2.

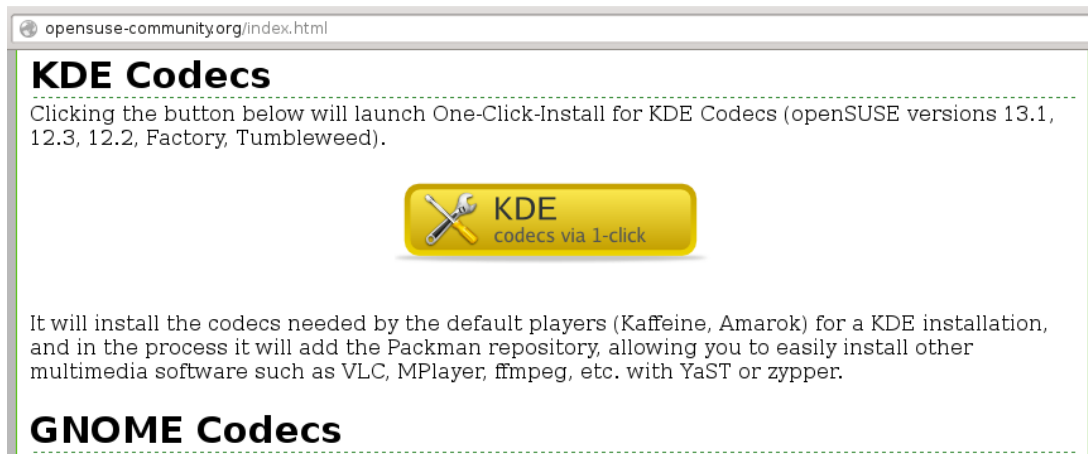


Figura A.1.2. Versión KDE para software de OpenSuse.

A continuación el sistema lanzará el gestor de instalación de paquetes software, Yast el cual solicitará confirmar la adición de los repositorios necesarios y la instalación de nuevo software, entre los cuales va incluido FFmpeg. Ver la figura a.1.3.



Figura A.1.3. Codecs multimedia para OpenSuse.

Después de confirmada la instalación de los anteriores paquetes, es posible que aparezca una nueva ventana, en donde debe elegirse las opciones convenientes para continuar con el proceso de instalación, como se muestra en la figura a.1.4. En ambos casos se elige la primera opción (instalar).

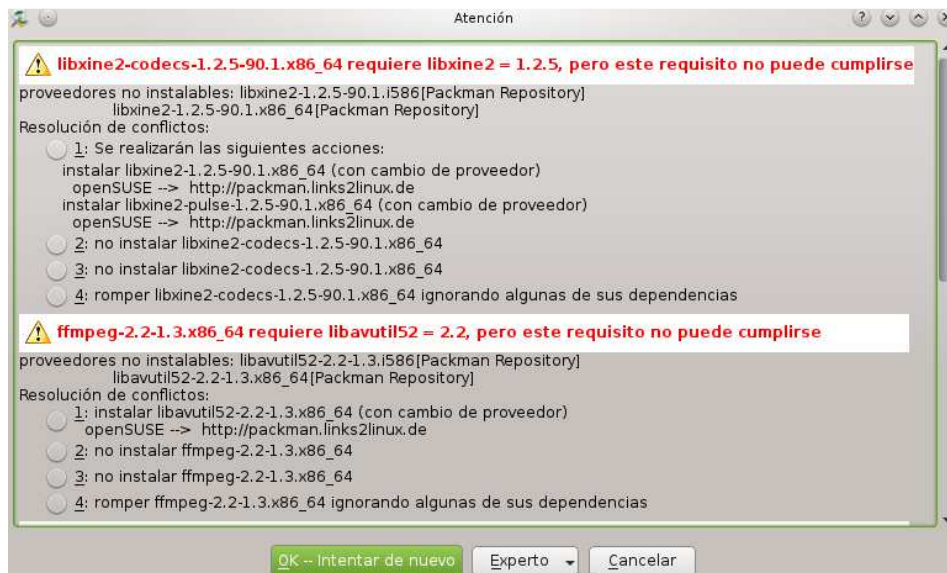


Figura A.1.4. Resolución de conflictos para la instalación de codecs en OpenSuse.

En las siguientes ventanas, se procede a confirmar la instalación de los paquetes software necesario para configurar FFmpeg, ver las figuras a.1.5 y a.1.6.

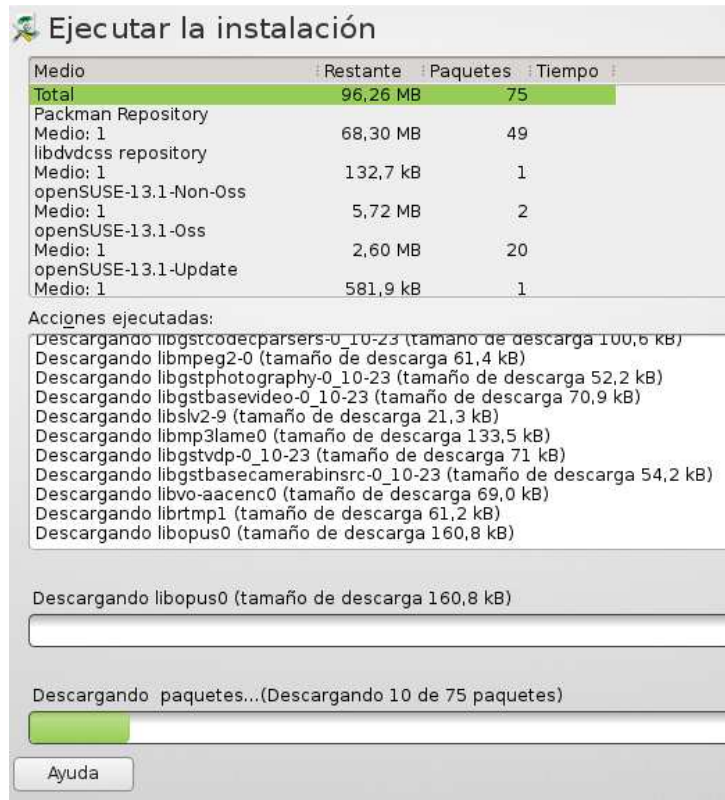


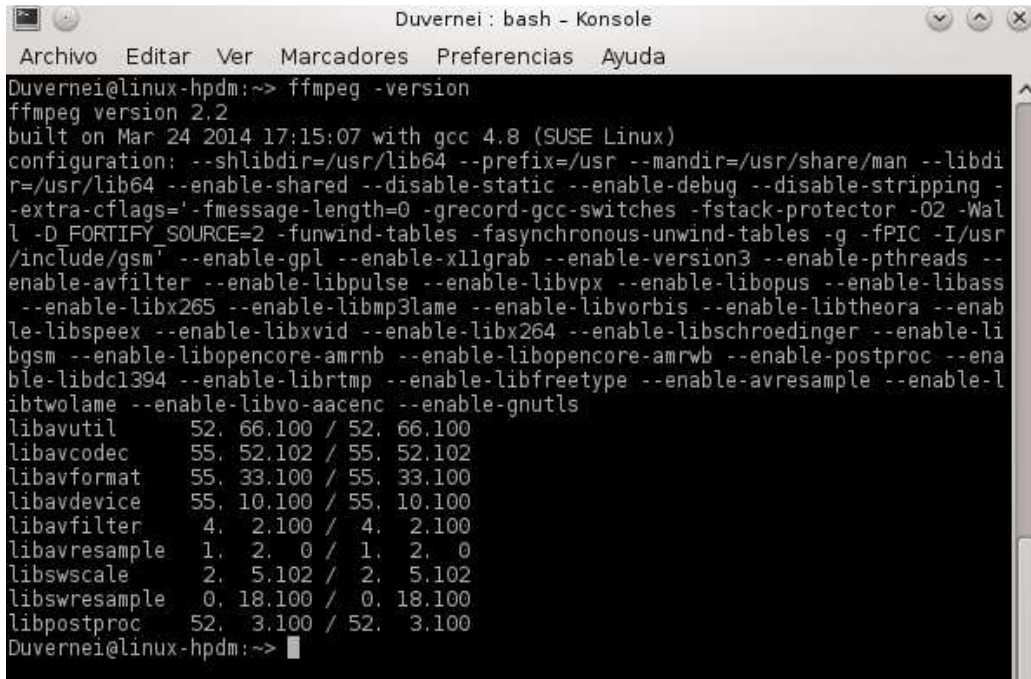
Figura A.1.5. Instalación de software en OpenSuse.



Figura A.1.6. Confirmación de instalación de software en OpenSuse.

Luego de finalizado el proceso de instalación de la herramienta, puede verificarse la instalación de la misma de la siguiente manera: primero abriendo una terminal de

Linux, al digitar el comando `ffmpeg -version` debe observarse algo semejante a lo mostrado en la figura a.1.7:



```
Duvernei@linux-hpdm:~> ffmpeg -version
ffmpeg version 2.2
built on Mar 24 2014 17:15:07 with gcc 4.8 (SUSE Linux)
configuration: --shlibdir=/usr/lib64 --prefix=/usr --mandir=/usr/share/man --libdir=/usr/lib64 --enable-shared --disable-static --enable-debug --disable-stripping --extra-cflags='-fmessage-length=0 -grecord-gcc-switches -fstack-protector -O2 -Wall -D_FORTIFY_SOURCE=2 -funwind-tables -fasynchronous-unwind-tables -g -fpic -I/usr/include/gsm' --enable-gpl --enable-x11grab --enable-version3 --enable-threads --enable-avfilter --enable-libpulse --enable-libvpx --enable-libopus --enable-libass --enable-libx265 --enable-libmp3lame --enable-libvorbis --enable-libtheora --enable-libspeex --enable-libxvid --enable-libx264 --enable-libschrödinger --enable-libgsm --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-postproc --enable-libdc1394 --enable-librtmp --enable-libfreetype --enable-avresample --enable-libtwolame --enable-libvo-aacenc --enable-gnutls
libavutil      52. 66.100 / 52. 66.100
libavcodec     55. 52.102 / 55. 52.102
libavformat    55. 33.100 / 55. 33.100
libavdevice    55. 10.100 / 55. 10.100
libavfilter     4.  2.100 /  4.  2.100
libavresample  1.  2.  0 /  1.  2.  0
libswscale     2.  5.102 /  2.  5.102
libswresample  0. 18.100 /  0. 18.100
libpostproc   52.  3.100 / 52.  3.100
Duvernei@linux-hpdm:~>
```

Figura A.1.7. Verificación de la instalación de FFmpeg mediante la terminal de Linux.

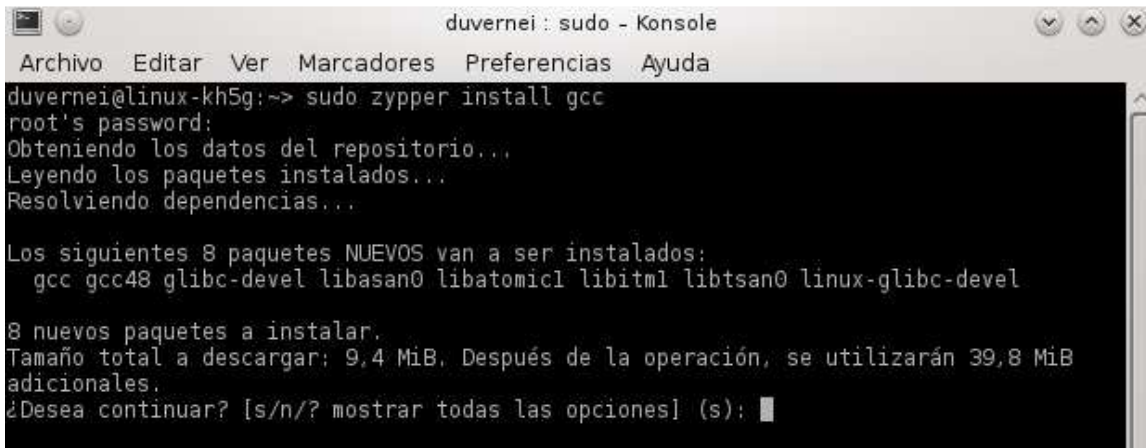
A.2 Compilación de Libwebm y Webm-tools

Para configurar las herramientas libwebm y webm-tools, es necesario instalar los compiladores de Linux para los lenguajes de programación C y C++, debido a que dichas herramientas serán construidas a partir del código fuente proporcionado por los desarrolladores del proyecto WebM.

Primero desde una terminal de comandos se digita la siguiente instrucción:

```
sudo zypper install gcc gcc-c++
```

Después de confirmar los permisos de usuario, pueden observarse los paquetes a instalar, como se muestra en las figuras a.2.1 y a.2.2:

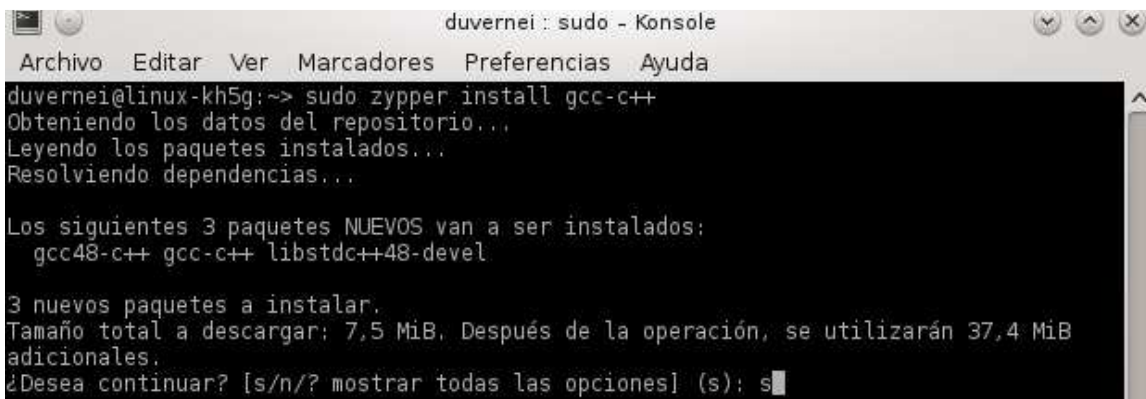


```
duvernei : sudo - Konsole
Archivo Editar Ver Marcadores Preferencias Ayuda
duvernei@linux-kh5g:~> sudo zypper install gcc
root's password:
Obteniendo los datos del repositorio...
Leyendo los paquetes instalados...
Resolviendo dependencias...

Los siguientes 8 paquetes NUEVOS van a ser instalados:
 gcc gcc48 glibc-devel libasan0 libatomic1 libitm1 libtsan0 linux-glibc-devel

8 nuevos paquetes a instalar.
Tamaño total a descargar: 9,4 MiB. Después de la operación, se utilizarán 39,8 MiB
adicionales.
¿Desea continuar? [s/n/? mostrar todas las opciones] (s):
```

Figura A.2.1. Instalación de compiladores C en OpenSuse.



```
duvernei : sudo - Konsole
Archivo Editar Ver Marcadores Preferencias Ayuda
duvernei@linux-kh5g:~> sudo zypper install gcc-c++
Obteniendo los datos del repositorio...
Leyendo los paquetes instalados...
Resolviendo dependencias...

Los siguientes 3 paquetes NUEVOS van a ser instalados:
 gcc48-c++ gcc-c++ libstdc++48-devel

3 nuevos paquetes a instalar.
Tamaño total a descargar: 7,5 MiB. Después de la operación, se utilizarán 37,4 MiB
adicionales.
¿Desea continuar? [s/n/? mostrar todas las opciones] (s): s
```

Figura A.2.2. Instalación de compiladores C++ en OpenSuse.

El código fuente de las herramientas libwebm y webm-tools, puede ser descargado desde el repositorio Git indicado en la página web oficial del proyecto WebM o mediante una descarga directa desde este mismo sitio:

<http://www.webmproject.org/>

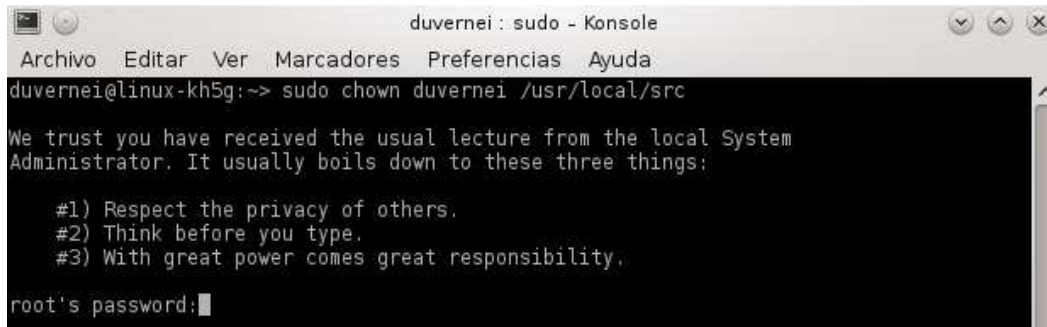
Luego de haber obtenido el código fuente de las herramientas por cualquiera de las dos formas mencionadas anteriormente, desde una terminal de Linux se procede a ejecutar las instrucciones que se describirán a continuación.

Primero se debe cambiar el propietario del directorio donde está ubicado nuestro código fuente con el comando:

sudo chown USUARIO DIRECTORIO

USUARIO: Corresponde al nuevo propietario del directorio destino.

DIRECTORIO: Corresponde al directorio donde se encuentra almacenado nuestro código fuente, en nuestro caso /usr/local/src (Ver la figura a.2.3).



```
duvernei : sudo - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:~> sudo chown duvernei /usr/local/src
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

root's password:█
```

Figura A.2.3. Configuración de permisos sobre directorios en Linux.

Posteriormente, se procede a agregar permisos de escritura y lectura a los subdirectorios y archivos contenidos en nuestro directorio de código fuente, mediante la siguiente instrucción (ver la figura a.2.4):

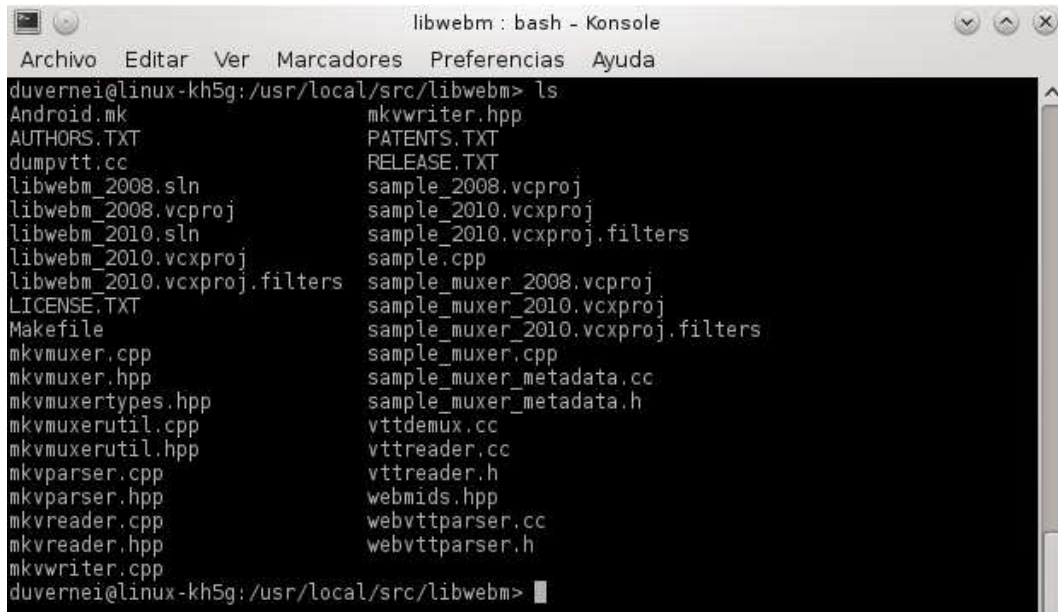
sudo chmod u+rwx DIRECTORIO



```
src : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:~> sudo chmod u+rwx /usr/local/src
duvernei@linux-kh5g:~> cd /usr/local/src
duvernei@linux-kh5g:/usr/local/src> ls
libwebm
libwebm-25025a5471d609d57f59e5d79ab7519635bfe95b.tar.gz
webm-tools
webm-tools-2dc88bf28bb79d1ed2f8cbf5858189a3f40d3dc5.tar.gz
duvernei@linux-kh5g:/usr/local/src> █
```

Figura A.2.4. Configuración de permisos avanzados sobre directorios en Linux.

Para iniciar la compilación del código fuente de las herramientas, desde la terminal debemos navegar hasta ubicarnos en la raíz de los archivos con extensión .c, .h y .cpp, los cuales contienen el código de la aplicación a compilar. Ver las figuras a.2.5 y a.2.6.

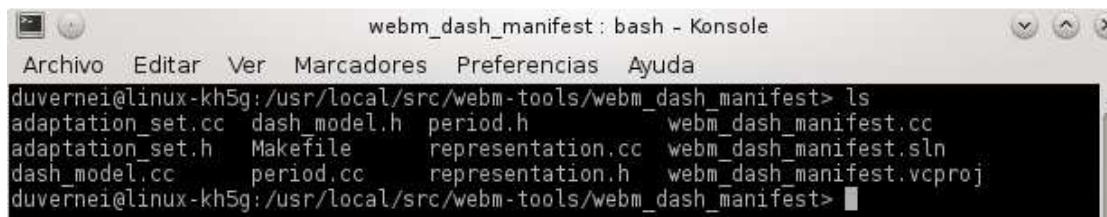


```

libwebm : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:/usr/local/src/libwebm> ls
Android.mk          mkvwriter.hpp
AUTHORS.TXT        PATENTS.TXT
dumpvtt.cc         RELEASE.TXT
libwebm_2008.sln   sample_2008.vcproj
libwebm_2008.vcproj sample_2010.vcxproj
libwebm_2010.sln   sample_2010.vcxproj.filters
libwebm_2010.vcxproj sample.cpp
libwebm_2010.vcxproj.filters sample_muxer_2008.vcproj
LICENSE.TXT        sample_muxer_2010.vcxproj
Makefile           sample_muxer_2010.vcxproj.filters
mkvmuxer.cpp       sample_muxer.cpp
mkvmuxer.hpp       sample_muxer_metadata.cc
mkvmuxerutil.hpp  sample_muxer_metadata.h
mkvmuxerutil.cpp  vttmuxer.cc
mkvmuxerutil.hpp  vttreader.cc
mkvparser.cpp      vttreader.h
mkvparser.hpp      webmids.hpp
mkvreader.cpp      webvttparser.cc
mkvreader.hpp      webvttparser.h
mkvwriter.cpp
duvernei@linux-kh5g:/usr/local/src/libwebm>

```

Figura A.2.5. Lista de contenidos de la herramienta libwebm.



```

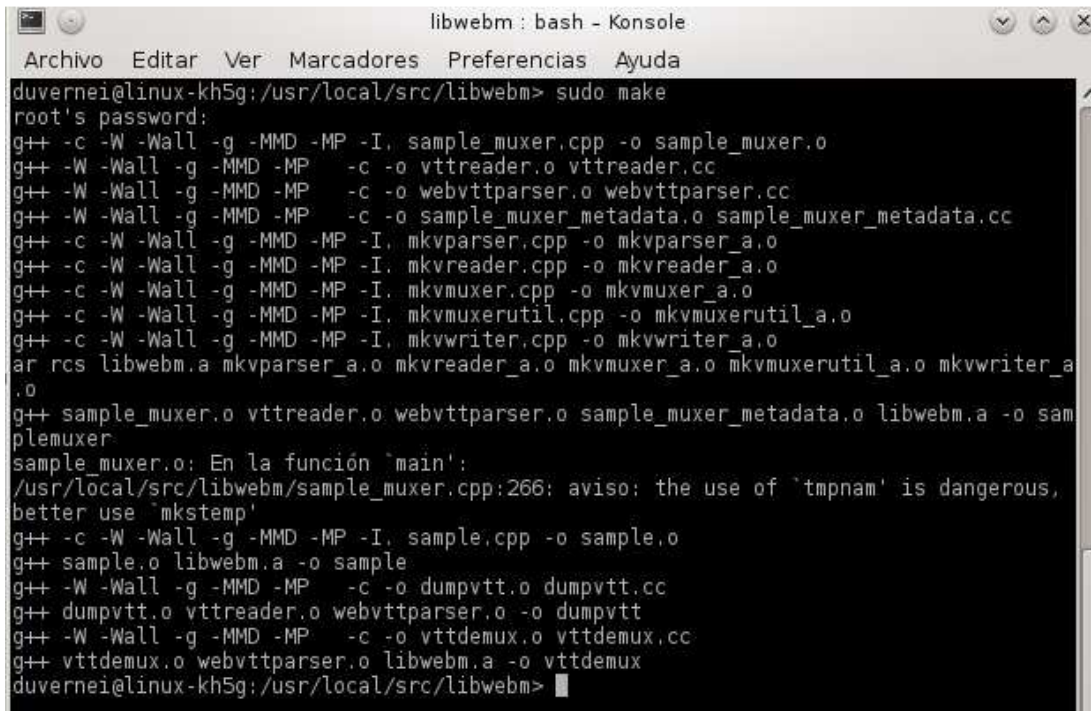
webm_dash_manifest : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest> ls
adaptation_set.cc  dash_model.h  period.h          webm_dash_manifest.cc
adaptation_set.h   Makefile      representation.cc webm_dash_manifest.sln
dash_model.cc     period.cc     representation.h  webm_dash_manifest.vcproj
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest>

```

Figura A.2.6. Lista de contenidos de la herramienta webm_dash_manifest.

Una vez ubicados en el directorio del código fuente de las herramientas, se procede a compilar el código utilizando la siguiente instrucción (ver las figuras a.2.7 y a.2.8):

sudo make

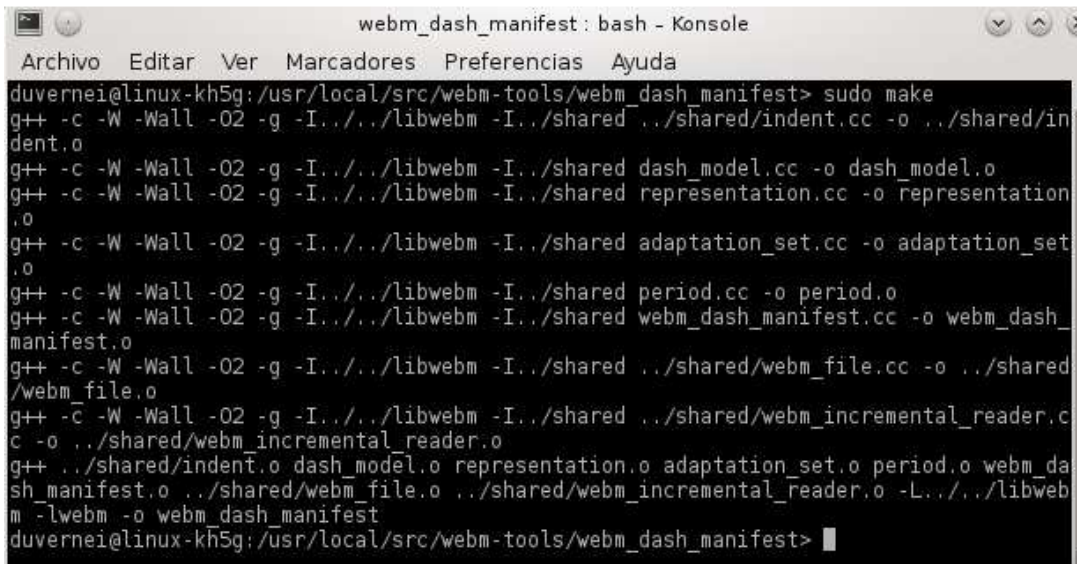


```

libwebm : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:/usr/local/src/libwebm> sudo make
root's password:
g++ -c -W -Wall -g -MMD -MP -I. sample_muxer.cpp -o sample_muxer.o
g++ -W -Wall -g -MMD -MP -c -o vttreader.o vttreader.cc
g++ -W -Wall -g -MMD -MP -c -o webvttparser.o webvttparser.cc
g++ -W -Wall -g -MMD -MP -c -o sample_muxer_metadata.o sample_muxer_metadata.cc
g++ -c -W -Wall -g -MMD -MP -I. mkvparser.cpp -o mkvparser_a.o
g++ -c -W -Wall -g -MMD -MP -I. mkvreader.cpp -o mkvreader_a.o
g++ -c -W -Wall -g -MMD -MP -I. mkvmuxer.cpp -o mkvmuxer_a.o
g++ -c -W -Wall -g -MMD -MP -I. mkvmuxerutil.cpp -o mkvmuxerutil_a.o
g++ -c -W -Wall -g -MMD -MP -I. mkvwriter.cpp -o mkvwriter_a.o
ar rcs libwebm.a mkvparser_a.o mkvreader_a.o mkvmuxer_a.o mkvmuxerutil_a.o mkvwriter_a.o
g++ sample_muxer.o vttreader.o webvttparser.o sample_muxer_metadata.o libwebm.a -o sample_muxer
sample_muxer.o: En la función `main':
/usr/local/src/libwebm/sample_muxer.cpp:266: aviso: the use of `tmpnam' is dangerous,
better use `mkstemp'
g++ -c -W -Wall -g -MMD -MP -I. sample.cpp -o sample.o
g++ sample.o libwebm.a -o sample
g++ -W -Wall -g -MMD -MP -c -o dumpvtt.o dumpvtt.cc
g++ dumpvtt.o vttreader.o webvttparser.o -o dumpvtt
g++ -W -Wall -g -MMD -MP -c -o vttmux.o vttmux.cc
g++ vttmux.o webvttparser.o libwebm.a -o vttmux
duvernei@linux-kh5g:/usr/local/src/libwebm>

```

Figura A.2.7. Compilación de código fuente en Linux de libwebm.



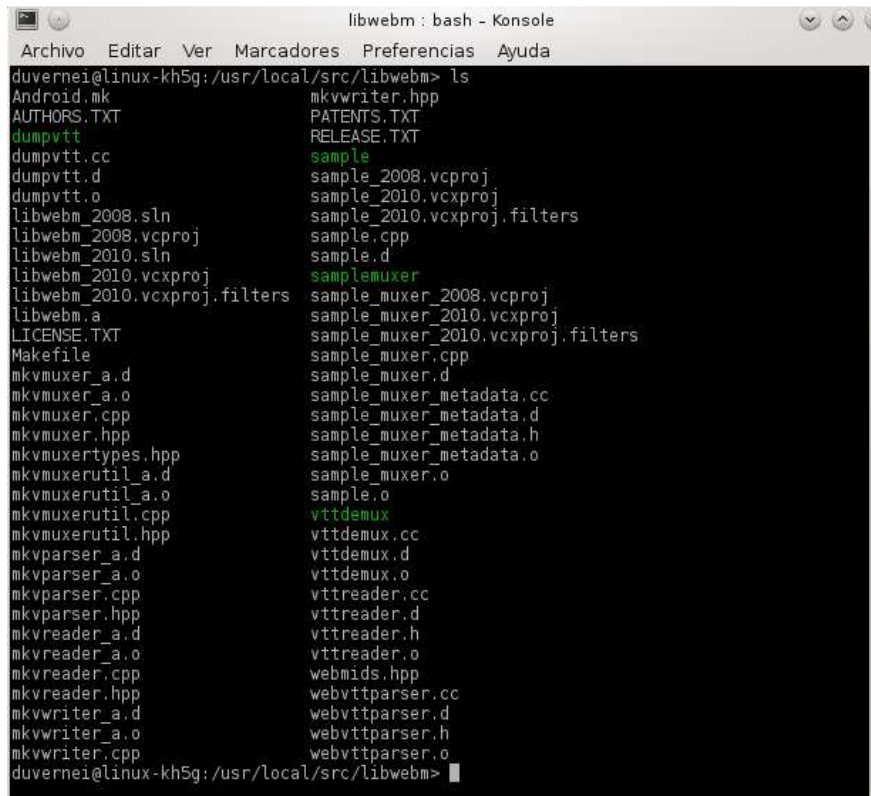
```

webm_dash_manifest : bash - Konsole
Archivo  Editar  Ver  Marcadores  Preferencias  Ayuda
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest> sudo make
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared indent.o -o ../shared/indent.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared dash_model.o -o dash_model.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared representation.o -o representation.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared adaptation_set.o -o adaptation_set.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared period.o -o period.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared webm_dash_manifest.o -o webm_dash_manifest.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared ../shared/webm_file.o -o ../shared/webm_file.o
g++ -c -W -Wall -O2 -g -I../libwebm -I../shared ../shared/webm_incremental_reader.o -o ../shared/webm_incremental_reader.o
g++ ../shared/indent.o dash_model.o representation.o adaptation_set.o period.o webm_dash_manifest.o ../shared/webm_file.o ../shared/webm_incremental_reader.o -L../libwebm -lwebm -o webm_dash_manifest
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest>

```

Figura A.2.8. Compilación de código fuente en Linux de webm_dash_manifest.

Al finalizar el proceso de compilación, observaremos los archivos generados, como se indica en las figuras a.2.9 y a.2.10:

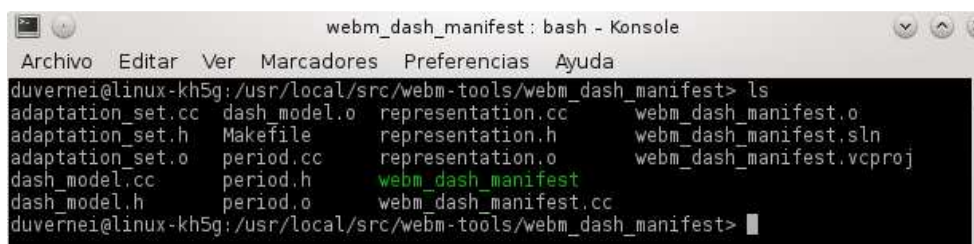


```

libwebm : bash - Konsole
Archivo Editar Ver Marcadores Preferencias Ayuda
duvernei@linux-kh5g:/usr/local/src/libwebm> ls
Android.mk          mkvwriter.hpp
AUTHORS.TXT        PATENTS.TXT
dumpvtt            RELEASE.TXT
dumpvtt.cc         sample
dumpvtt.d          sample_2008.vcproj
dumpvtt.o          sample_2010.vcproj
libwebm_2008.sln   sample_2010.vcxproj.filters
libwebm_2008.vcproj sample.cpp
libwebm_2010.sln  sample.d
libwebm_2010.vcxproj samplemuxer
libwebm_2010.vcxproj.filters sample_muxer_2008.vcproj
libwebm.a          sample_muxer_2010.vcproj
LICENSE.TXT        sample_muxer_2010.vcxproj.filters
Makefile          sample_muxer.cpp
mkvmuxer_a.d      sample_muxer.d
mkvmuxer_a.o      sample_muxer_metadata.cc
mkvmuxer.cpp      sample_muxer_metadata.d
mkvmuxer.hpp      sample_muxer_metadata.h
mkvmuxertypes.hpp sample_muxer_metadata.o
mkvmuxerutil_a.d sample_muxer.o
mkvmuxerutil_a.o sample.o
mkvmuxerutil.cpp vttmux
mkvmuxerutil.hpp vttmux.cc
mkvparser_a.d    vttmux.d
mkvparser_a.o    vttmux.o
mkvparser.cpp    vttreader.cc
mkvparser.hpp    vttreader.d
mkvreader_a.d    vttreader.h
mkvreader_a.o    vttreader.o
mkvreader.cpp    webmids.hpp
mkvreader.hpp    webvttparser.cc
mkvwriter_a.d    webvttparser.d
mkvwriter_a.o    webvttparser.h
mkvwriter.cpp    webvttparser.o
duvernei@linux-kh5g:/usr/local/src/libwebm>

```

Figura A.2.9. Archivos generados en libwebm.



```

webm_dash_manifest : bash - Konsole
Archivo Editar Ver Marcadores Preferencias Ayuda
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest> ls
adaptation_set.cc  dash_model.o  representation.cc  webm_dash_manifest.o
adaptation_set.h  Makefile      representation.h  webm_dash_manifest.sln
adaptation_set.o  period.cc     representation.o  webm_dash_manifest.vcproj
dash_model.cc     period.h      webm_dash_manifest
dash_model.h      period.o      webm_dash_manifest.cc
duvernei@linux-kh5g:/usr/local/src/webm-tools/webm_dash_manifest>

```

Figura A.2.10. Archivos generados en webm_dash_manifest.

En este punto ya se encuentran compiladas las herramientas, para ser utilizadas posteriormente en procesos de codificación de contenidos multimedia DASH.

Anexo B

Proceso de generación de contenidos multimedia DASH en formato WebM

Las siguientes herramientas han sido seleccionadas en este trabajo, para el proceso de generación de contenidos multimedia DASH y están indicadas según su orden de participación durante el proceso:

- FFmpeg
- Libwebm
- Webm-tools

Estas herramientas deben haber sido compiladas e instaladas con anterioridad en un sistema operativo Linux, como se indica en el anexo A de esta monografía. También los archivos multimedia a codificar deberán tener el formato WebM y estar dispuestos en un directorio de fácil acceso.

B.1 Generación de los flujos de video y audio

Primero desde una terminal de comandos Linux, se debe navegar hasta el directorio donde están almacenados los archivos multimedia en formato WebM, por ejemplo:

```
cd /Documentos/videos/
```

Una vez ubicados en la carpeta anterior, se procede a la codificación de los contenidos multimedia a diferentes tasas de bit.

Para ejecutar correctamente la herramienta de codificación FFmpeg se debe pasar como parámetros:

- ✓ El nombre del archivo multimedia a codificar, incluyendo la extensión del mismo (.webm).
- ✓ El códec de video adecuado.
- ✓ La tasa de bit a la cual se desea codificar el flujo de video.
- ✓ Un tamaño de resolución de pantalla.
- ✓ El parámetro para indicar que solamente se desea un flujo de video.
- ✓ El nombre del flujo de video de salida con la extensión del formato (.webm).

Las siguientes líneas describen los comandos correctos para realizar la codificación de flujos de video a diferentes tasas de bit:

```
ffmpeg -i archivo.webm -an -vcodec libvpx -vf scale:320:240 -vb 100k  
-keyint_min 150 -g 150 stream_video1.webm
```

```
ffmpeg -i archivo.webm -an -vcodec libvpx -vf scale:320:240 -vb 200k  
-keyint_min 150 -g 150 stream_video2.webm
```

...

De igual modo para generar un flujo de audio debe digitarse la siguiente instrucción:

```
ffmpeg -i archivo.webm -vn -acodec libvorbis -ab 128k stream_audio.webm
```

En este punto ya están listos los flujos de video y el flujo de audio para continuar con la siguiente parte de la generación de contenidos multimedia DASH.

B.2 Indexación de contenidos

A partir de los contenidos codificados en el paso anterior, en este paso se añaden índices de búsqueda a cada flujo de video, es decir, los fotogramas clave para realizar el canjeo entre los stream durante la reproducción del contenido multimedia DASH.

Desde una terminal de comandos de Linux y ubicados en el directorio donde están alojados los flujos de video y audio generados en la anterior parte, debe ejecutarse el script *sample_muxer* de la herramienta Libwebm. La instrucción debe contener

como parámetros el nombre del archivo origen y el nombre del archivo destino con la extensión del formato (.webm), como se indica a continuación:

```
./libwebm/samplemuxer -i stream_video1.webm -o  
stream_video1_sm.webm
```

```
./libwebm/samplemuxer -i stream_video2.webm -o  
stream_video2_sm.webm
```

...

Para el flujo de audio de esta otra manera:

```
./libwebm/samplemuxer -i stream_audio.webm -o stream_audio_sm.webm  
-output_cues 1 -cues_on_audio_track 1 -max_cluster_duration 5  
-audio_track_number 2
```

Puesto que ya no es necesario conservar los anteriores flujos de video y audio, se procede a su eliminación manualmente o desde la misma terminal, digitando:

```
rm -r stream_video1.webm  
rm -r stream_video2.webm
```

...

```
rm -r stream_audio.webm
```

Los nuevos flujos de video y audio ahora están listos para generar el archivo de manifiesto MPD, por lo tanto, en la siguiente parte se describirá la forma de hacerlo.

B.3 Generación del archivo de manifiesto MPD

Finalmente para cumplir con la especificación del estándar DASH, se debe generar un archivo descriptor MPD, por lo tanto es necesario ejecutar el script *webm_dash_manifest* de la herramienta Webm-tools, para ello desde una terminal de Linux se incluyen como parámetros cada uno de los flujos de video y audio generados en la anterior sección, la instrucción sería algo semejante a la siguiente línea:

```
./webm-tools/ webm_dash_manifest -o archivo_manifiesto.mpd
-as id=0,leng=eng -r id=0,file=stream_video1_sm.webm
-r id=1,file=stream_video2_sm.webm
...
-r id=N-1,file=stream_videoN_sm.webm
-as id=1,leng=eng -r id=N,file=stream_audio_sm.webm
```

La expresión **N-1** y **N** indican la secuencia del parámetro *id* según la cantidad de flujos de video que hayan sido codificados.

En este punto el proceso de generación de contenidos multimedia DASH, ha terminado de manera satisfactoria. Sin embargo, al momento de tratar de adaptar nuevos contenidos WebM al estándar, es necesario volver a iniciar el mismo proceso desde la sección B1. Por esta razón la siguiente sección explicará el mismo proceso con ayuda de la herramienta de codificación automática, implementada en este trabajo.

B.4 Proceso de codificación automática, utilizando la herramienta Dash WebM Converter

A partir de las anteriores secciones de este anexo, puede observarse cada una de las tareas específicas que se deben realizar para generar contenidos multimedia DASH. Por lo tanto en esta sección se presenta como alternativa el uso de la herramienta de codificación automática, desarrollada en este trabajo.

La herramienta Dash WebM Converter, se encargará de realizar el anterior proceso de forma automática, donde el usuario solo tendrá que seleccionar un par de parámetros para iniciar la codificación. A continuación en la figura b.4.1 se muestra la interfaz gráfica de dicha herramienta.

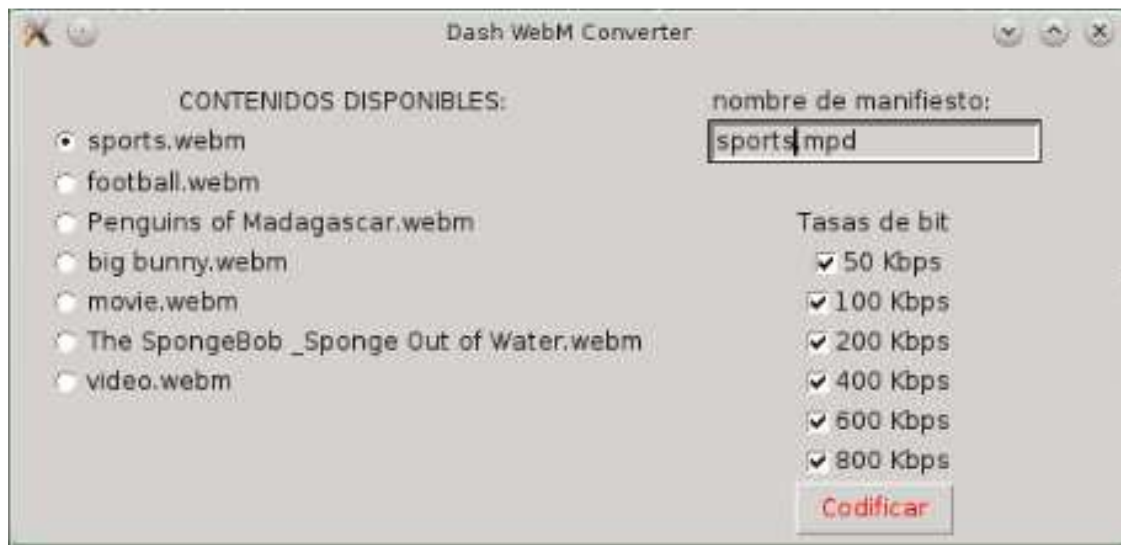


Figura B.4.1. Interfaz gráfica de Dash WebM Converter.

La función de la herramienta Dash WebM Converter es ejecutar de manera secuencial las tareas de codificación (generación de flujos de video y audio), segmentación (indexación de contenidos) y generación del archivo de manifiesto MPD para contenidos multimedia DASH en formato WebM. A continuación se enumeran cada una de las funciones específicas ejecutadas con ayuda de la herramienta:

1. Se debe seleccionar el archivo multimedia WebM a codificar, mediante el sistema de selección de la herramienta de codificación automática.
2. Es necesario seleccionar las tasas de bit para la generación de los flujos de video.
3. También se debe indicar un nombre para el archivo de manifiesto MPD.
4. Finalmente para dar inicio al proceso de codificación automática, se presiona el botón con la etiqueta codificar.

Automáticamente la herramienta sigue la siguiente secuencia de tareas:

- Se crea el directorio que va a contener los flujos de video y audio junto a su respectivo archivo descriptor MPD.
- Se generan los flujos de video según las tasas de bit seleccionadas y el flujo de audio a una tasa fija de 128 Kbps.
- Se agregan los índices de búsqueda a cada uno de los flujos de video y audio generados.
- Se eliminan los flujos de video y audio que conformaron el proceso de pre codificación.

Finalmente en la figura b.4.2 se muestra el directorio generado por la herramienta, en la cual se pueden observar los flujos de video y audio adaptados al estándar junto a su respectivo archivo descriptor MPD.

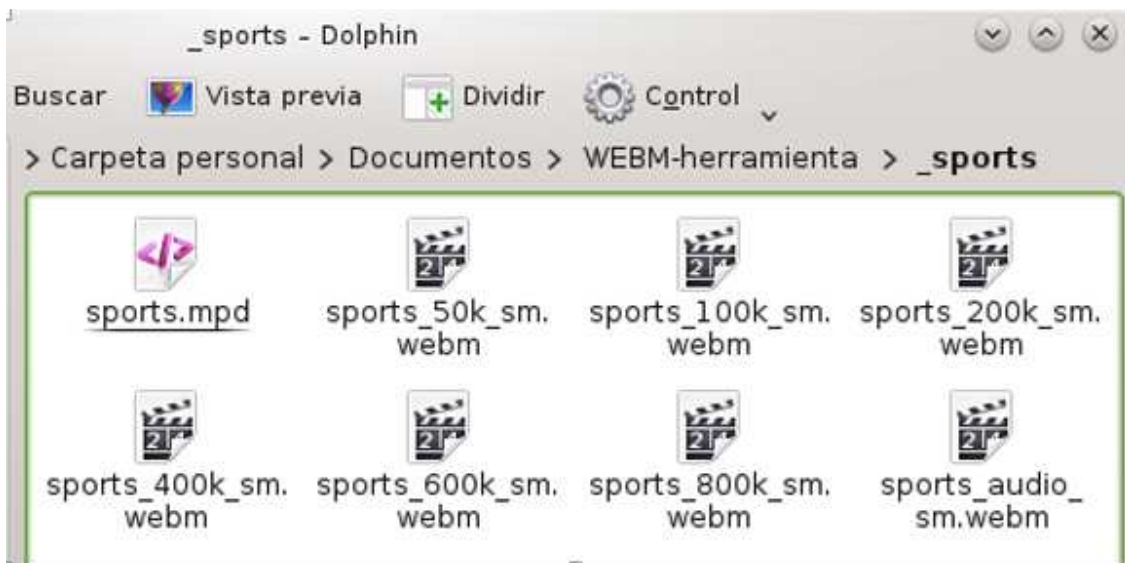


Figura B.4.2. Contenidos DASH, generados por la herramienta de codificación.

Anexo C

Manejo de la herramienta de medición de estrés apache benchmark

A continuación se describe el uso de la herramienta apache benchmark, la cual fue utilizada en este trabajo para la realización de pruebas de estrés sobre el servicio interactivo de chat, asociado al servicio de VoD.

Primero se instala la herramienta desde una terminal de Linux, mediante el siguiente comando:

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

La figura C.1 muestra el mismo procedimiento para la instalación utilizando el gestor de paquetes Synaptic de Ubuntu.



Figura C.1. Instalación de Apache Benchmark.

Una vez instalada, esta herramienta nos permite realizar múltiples peticiones http sobre el servidor webpy, el cual contiene el servicio de chat. A continuación se procede a arrancar el servidor, bien sea mediante una terminal de Linux o desde el entorno de desarrollo Geany. En la figura C.2 se muestra el entorno de desarrollo Geany a través del cual se lanzará el servicio, para ello en la parte superior de la ventana se encuentra un conjunto de controles, entre los cuales aparece un icono para ejecutar el script del servidor.

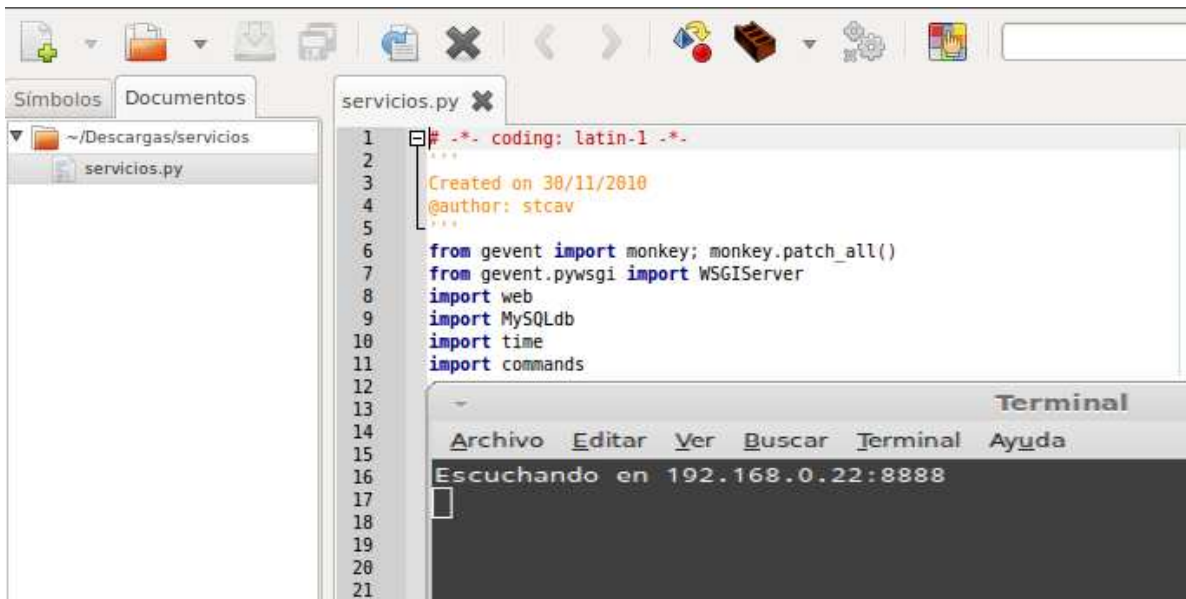


Figura C.2. Ejecución del servidor webpy desde el entorno de desarrollo Geany.

Después de ejecutar el servidor webpy, se procede a realizar pruebas de conexión sobre el servicio de chat. El modo de acceso a la herramienta apache benchmark es a través de una terminal de Linux. La figura C.3 presenta el comando utilizado para realizar 10 peticiones de forma secuencial.

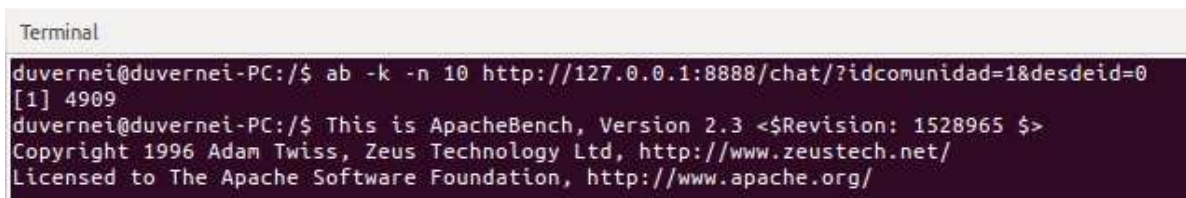


Figura C.3. Ejecución de Apache Benchmark.

La línea “**ab -k**” hace referencia a la herramienta apache benchmark para realizar peticiones a determinado servicio web, a continuación aparece el parámetro “**-n 10**” el cual alude a 10 conexiones de manera secuencial. Al final de esto debe ir la dirección en la cual se accede al servicio. La figura C.4 muestra parte del resultado después de teclear el comando de la figura C.3.

```
Concurrency Level:      1
Time taken for tests:   0.074 seconds
Complete requests:     10
Failed requests:       0
Keep-Alive requests:   0
Total transferred:     16000 bytes
HTML transferred:      15250 bytes
Requests per second:   135.51 [#/sec] (mean)
Time per request:      7.380 [ms] (mean)
Time per request:      7.380 [ms] (mean, across all concurrent requests)
Transfer rate:         211.73 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0    0  0.0      0    0
Processing:  5    7  1.6      8   10
Waiting:    5    7  1.6      7   10
Total:      5    7  1.6      8   10
```

Figura C.4. Resultado obtenido desde Apache Benchmark.

Del mismo modo para realizar peticiones http de manera simultánea, basta con agregar el parámetro “**-c 10**”, para 10 conexiones de manera concurrente, de esta manera el comando quedaría:

```
ab -k -n 10 -c 10 http://127.0.0.1:8888/chat/idcomunidad=1&desdeid=0
```

De acuerdo a la anterior sentencia, es posible combinar valores para los parámetros “**-n**” y “**-c**”. Por ejemplo se pueden realizar 100 peticiones al servicio desde el parámetro “**-n**”, con bloques de 20 peticiones de manera concurrente a través del parámetro “**-c**”, la sentencia final será la siguiente:

```
ab -k -n 100 -c 20 http://127.0.0.1:8888/chat/idcomunidad=1&desdeid=0
```


Anexo D

Manual de usuario para interacción con el servicio de Video bajo Demanda

A continuación se describen cada una de las funcionalidades del servicio de VoD, dispuestas para los usuarios que accedan al mismo. Primero se debe acceder al servicio ingresando la URL de su ubicación desde un navegador Chrome en su versión 17 o superior a esta. La figura D.1 muestra el servicio de VoD en funcionamiento.



Figura D.1. Servicio de VoD en funcionamiento.

Los contenidos presentados al usuario corresponden a los Tráilers de las películas más recientes en el cine. Las numeraciones presentes en la figura D.1 corresponden a las siguientes funcionalidades o características:

1. **Etiqueta de título:** Corresponde a la etiqueta que muestra el nombre de la película que actualmente se está reproduciendo.
2. **Reproductor de contenidos:** Es el marco en donde se podrá visualizar la película previamente elegida por el usuario. Este reproductor está basado en el uso de HTML5.
3. **Media:** Corresponde a la pestaña que despliega el catálogo de contenidos multimedia del servicio de VoD ofrecido al usuario.
4. **Selector de género:** Es una lista desplegable desde la cual el usuario elige el tipo de género de película de su preferencia. Una vez seleccionado el género se actualiza el listado de películas mostrado en la parte inferior del selector de género.
5. **Poster:** Es la captura de una escena de la película. Al dar clic sobre esta se inicia el despliegue de la película en el reproductor de contenidos.
6. **Información del contenido:** Consiste en la información más relevante del contenido disponible para el usuario, entre ella está el género de película al que pertenece dicho contenido, el año de lanzamiento de la película y el número de veces que ha sido reproducido o visitado.
7. **URL de descriptor:** Corresponde a una etiqueta que contiene la ubicación del archivo descriptor MPD, este enlace puede ser utilizado por otro cliente web basado en el estándar DASH, para hacer uso de los contenidos dispuestos en el servicio de VoD.
8. **Sistema de valoración:** Es un servicio interactivo para la valoración de contenidos multimedia del servicio de VoD, el cual permite ordenar las películas teniendo en cuenta las valoraciones hechas por los usuarios.

9. **Chat:** Es una pestaña para acceder al servicio interactivo de chat asociado al servicio de VoD. Permite al usuario participar dentro del chat general de la comunidad en la cual se encuentra registrado, ver Figura D.2.



Figura D.2. Servicio interactivo de chat.

10. **Herramienta de codificación automática:** Es una pestaña para acceder a un componente administrativo del servicio de VoD. Desde esta es posible generar contenidos multimedia DASH y publicarlos en el catálogo de contenidos del servicio, ver Figura D.3.



Figura D.3. Herramienta de codificación automática online.

11. **Medidor de ancho de banda:** Es un sistema para visualizar la calidad de imagen del video que actualmente se está desplegando.

12. **Botón de medidor de ancho de banda:** Corresponde a un botón para visualizar u ocultar la gráfica el medidor de ancho de banda.

Anexo E

Artículo:

Herramienta para la codificación automática de contenidos multimedia WebM, soportados en el estándar de streaming adaptativo DASH.

Herramienta para la codificación automática de contenidos multimedia WebM, soportados en el estándar de streaming adaptativo DASH

Duvernei Ortiz T., Gabriel Elías Chanchí G., José Luis Arciniegas H., Wilmar Y. Campo M.

RESUMEN / ABSTRACT

Tradicionalmente el videostreaming ha sido soportado por los protocolos RTP y RTSP, de modo que el servidor gestiona una sesión diferente para cada cliente y coordina la entrega de paquetes. Actualmente el estándar de streaming adaptativo DASH ofrece otro enfoque a través de HTTP, de tal forma que el cliente extrae los datos del servidor, sin mantener el estado de la sesión. Así, se tiene como ventajas el pleno uso de la infraestructura de Internet, y la adaptación del contenido multimedia al ancho de banda de la red. A pesar de lo anterior, el proceso de generación de contenidos multimedia DASH, requiere la ejecución secuencial de tareas de codificación, segmentación y creación del descriptor MPD. Para el caso de los contenidos multimedia WebM, las anteriores tareas son realizadas separadamente por un conjunto de herramientas libres, por lo que el proceso de generación del contenido DASH no es automático. En este artículo se propone la herramienta DASH WebMConverter, la cual automatiza el proceso de generación de contenidos DASH WebM. Adicionalmente, se presenta un escenario para la transmisión y consumo de streaming adaptativo, el cual fue configurado con ayuda de la herramienta propuesta y permite validar el correcto funcionamiento de esta.

Palabras claves: videostreaming, contenido multimedia, DASH, WebM.

Traditionally, video streaming technique has been supported by RTP and RTSP protocols, so that the server must manage a different session for each client and coordinate the delivery of packages. Currently the DASH standard offers a different approach using HTTP, in such a way that the client extracts data from server, without needing to maintain the session state. Thus, the advantages of DASH are: full use of existing Internet infrastructure and adaptation of multimedia content to the network bandwidth. Despite the above, the generation process for DASH multimedia content requires the sequential execution of multiple tasks such as: coding, segmentation and creation of MPD descriptor. In the case of WebM files, previous tasks are performed by a set of open source tools separately, which is why the generation process for DASH multimedia content is not automatic. In this paper we present a tool called "Dash WebM Converter", which automates the generation process for DASH WebM content. Also, in this paper we present a scenario for the transmission of DASH adaptive streaming, which was configured with the help of the proposed tool and allows validating the operation of this.

Keywords: Video streaming, multimedia content, DASH, WebM

An automatic coding tool for WebM multimedia content, supported on DASH standard