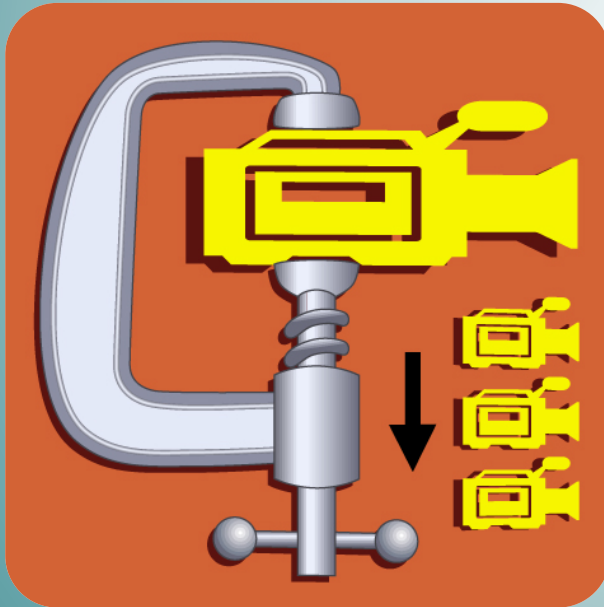


# Compresión de Vídeo

## Tema 2.10. FFMPEG



**Juan A. Michell Martín**  
**Gustavo A. Ruiz Robredo**

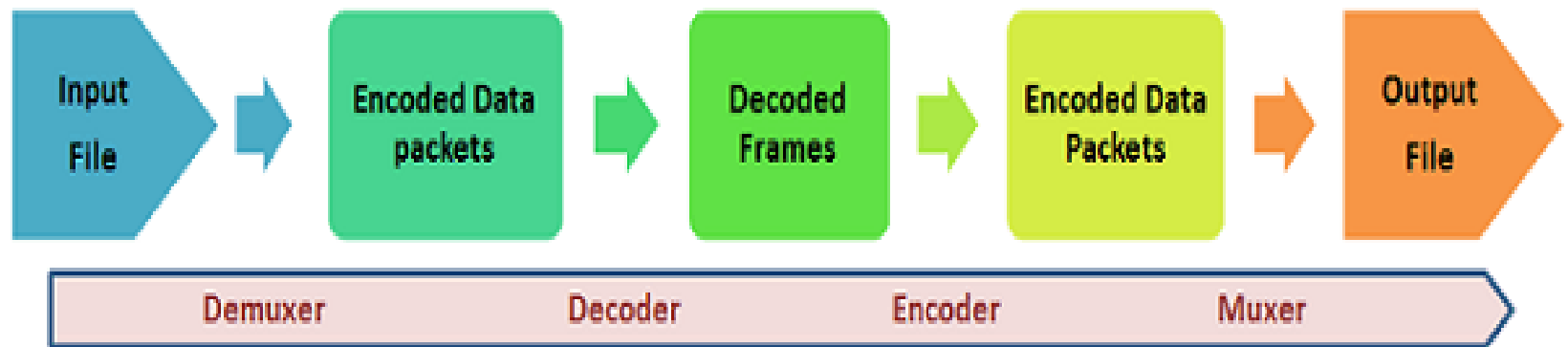
Departamento de Electrónica y Computadores

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

# FFmpeg

- **FFmpeg** es un proyecto de software libre que genera librerías y programas para el manejo de codificación multimedia.
- FFmpeg fue publicado bajo GNU Lesser General Public License 2.1+ o GNU General Public License 2+.



## Aplicaciones

- **ffmpeg.exe**: herramienta de comandos en líneas para convertir diferentes formatos multimedia.
- **ffplay.exe**: visualiza un video con diferentes formatos.
- **ffprobe.exe**: Muestra información de una multimedia.

Su página principal es: <https://ffmpeg.org>

Para descargar la versión ejecutable para Windows:  
<https://ffmpeg.zeranoe.com/builds>

➤ **Comandos de ayuda:**

**ffmpeg -h** , ayuda simple

**ffmpeg -h long** , ayuda más detallada

**ffmpeg -h full** , ayuda completa

**ffmpeg -h full | more** , ayuda completa paginada

➤ **Conocer los codecs y formatos soportados:**

**ffmpeg -formats**

**ffmpeg -formats | more**

**ffmpeg -codecs**

➤ **Obtener detalles de video:**

**ffmpeg -i Son.mpg**

**ffprobe -i Son.mpg**

Es muy importante la etiqueta de los ficheros. Por ejemplo, en este caso la etiqueta **mpg** indica que es un video en formato MPEG2, **mp4** en h.264 y **mkv** en formato hevc/h.265.

➤ **Visualizar videos:**

**ffplay son.mpg**

**ffplay -f rawvideo -pixel\_format yuv420p -video\_size 720x480 -framerate 25 formule1@218@720x480@30.yuv**

Más información en <https://ffmpeg.org/ffplay.html>

# CONVERSIÓN DE FORMATOS DE AUDIO

- Convierte el formato de audio sin comprimir (WAV) en formato comprimido MP3 (mpeg 1 layer 3):

```
ffmpeg -i stop2.wav stop2.mp3
```

- Convierte formato de audio WAV a MP3 con una frecuencia de muestreo de salida (sample rate) de 48000Hz a un bit rate de 256kbps:

```
ffmpeg -i stop2.wav -ar 48000 -ab 256k stop2.mp3
```

- Extrae al audio contenido en el video **videoplayback.mp4** y lo almacena en el fichero de **audio.mp4** para después convertirlo en formato mp3:

```
ffmpeg -i videoplayback.mp4 -acodec copy -vn audio.mp4
```

```
ffmpeg -i audio.mp4 audio.mp3
```

# MANIPULACIÓN DE IMÁGENES

- **Convierte una imagen en formato png a formato jpg:**  
**ffmpeg -i frameSon.png frameSon.jpg**
- **Cambia la imagen frameSon.png a un tamaño 320x240 pixels:**  
**ffmpeg -i frameSon.png -vf scale=320:240 frameSon2.png**
- **Cambia la imagen frameSon.png a formato jpg con un tamaño 320x240 y un factor de calidad 2. Este factor de calidad varía entre 1 (mejor) y 31 (peor). Valores recomendados de 2 a 5.**  
**ffmpeg -i frameSon.png -vf scale=320:240 -qscale:v 2 frameSon.jpg**

- **Convierte la imagen `lenna_color.tif` en formato `tif` a formato `4:2:0` con el nombre `lenna420p.yuv`:**

```
ffmpeg -i lenna_color.tif -f rawvideo -pix_fmt yuv420p  
lenna420p.yuv
```

- **Convierte la imagen `lenna_color.tif` en formato `tif` a formato `rgb24` con el nombre `lenna512x512.rgb`:**

```
ffmpeg -i lenna_color.tif -f rawvideo -pix_fmt rgb24  
lenna512x512.rgb
```

- **Para visualizar la imagen anterior:**

```
ffplay -s 512x512 -pix_fmt rgb24 lenna512x512.rgb
```



➤ **Convierte un video formato 4:2:0 a formato 4:4:4:**

```
ffmpeg -pix_fmt yuv420p -s 720x480 -r 25  
-i formule1@218@720x480@30.yuv -f rawvideo  
-pix_fmt yuv444p -s 720x480 -r 25 formule1_yuv444.yuv
```

**Parámetros:**

**-pix\_fmt**, formato yuv de los pixels, yuv420p formato 4:2:0, yuv444p, formato 4:4:4.

**-s**, tamaño del frame AnchoxAlto.

**-r**, frame rate en frames/s.

**-f**, formato de conversion, rawvideo significa sin conversión.

## MANIPULACIÓN DE SECUENCIA DE IMÁGENES Y VIDEO

### ➤ La siguiente orden:

```
ffmpeg -i son.mpg image-%03d.png
```

extrae el video son.mpg frame a frame y cada frame lo guarda con un número diferente. El formato image-%03d.png genera nombres como image-001.png, image-002.png, image-003.png, ... El formato image%d.png genera image1.png, image2.png, image2.png, ... . Si se añade además la opción **-start\_number 100**, el número que etiqueta esas imágenes empieza en 100.

### ➤ Es la instrucción inversa a la anterior. Genera un video.mpg a partir de una secuencia de imágenes:

```
ffmpeg -i image-%03d.png video.mpg
```

- Extrae el video `son.mpg` a partir del tiempo 5 segundos una frame y lo guarda como `frame.png`:

```
ffmpeg -i son.mpg -ss 00:00:05 -vframes 1 frame.png
```

- Similar a la instrucción anterior pero en formato jpeg con factor de calidad 2. Este factor de calidad varía entre 1 y 31. Valores recomendados de 2 a 5:

```
ffmpeg -i son.mpg -ss 00:00:05 -vframes 1  
-qscale:v 2 frame.jpg
```

- Extrae 5 segundos del video `son.mpg` y se convierte a formato gif para reproducirlo a un framerate de 25 frames por segundo:

```
ffmpeg -t 5 -i son.mpg -r 25 son.gif
```

- **Extrae 5 segundos de video `son.mpg` a partir del tiempo 3 segundos y se convierte a formato gif para reproducirlo a un framerate de 25 y cambia su tamaño a 128x64 pixels:**

```
ffmpeg -t 5 -ss 00:00:03 -i son.mpg -r 25 -s 128x64 son.gif
```

- **Añade la imagen `ffmpegLogo.png` al video `son.mpg` en la posición central y se construye el video `son2.mpg`:**

```
ffmpeg -i son.mpg -i ffmpegLogo.png -filter_complex  
"overlay=(main_w-overlay_w)/2:(main_h-overlay_h)/2"  
son2.mpg
```

# CODIFICACIÓN DE VIDEO

- Codificación de un video yuv, framerate 30 fps, formato 4:2:0, y tamaño 720x480 a un video **formule1.mp4** en formato h.264:

```
ffmpeg -r 30 -s 720x480 -pix_fmt yuv420p  
-i formule1@218@720x480@30.yuv  
-vcodec libx264 formule1.mp4
```

- Codificación de un video en formato h.264 a formato mpeg2:

```
ffmpeg -i formule1.mp4 formule1.mpg
```

- Codificación del video Son.mpg en formato mpeg1 a formato h.264:

```
ffmpeg -i Son.mpg Son.mp4
```

## ➤ Codificación de un video en formato h.264:

```
ffmpeg -r 30 -s 720x480 -pix_fmt yuv420p  
-i formule1@218@720x480@30.yuv -vcodec libx264 -r 30  
-b:v 1000k -g 20 -bf 3 -qmin 10 -qmax 50 -coder 1 -refs 3  
-me_method hex -crf 25 -maxrate 10000k -bufsize 1000k  
-preset veryslow -pass 1 -profile:v baseline formule1.mp4
```

**-r 30**, framerate.

**-b:v 1000k**, bitrate a 1000Kbit/s

**-g 20**, gop con un máximo de frames

**-bf 2**, número de b-frames. Valor 0 para no usarlas.

**-s 720x480**, tamaño de la entrada.

**-qmin 10**, valor mínimo de QP.

**-qmax 50**, valor máximo de QP.

**-coder 1**, codificador de entropía tipo CABAC

- refs 3, número de frames de referencia.
- me\_method hex, método de estimación de movimiento: dia (más rápido), hex, umh, full (más lento)
- crf 25, deseada QP promedio.
- maxrate 10000k, máximo bitrate.
- bufsize 1000k, tamaño del buffer.
- preset veryslow, posibles valores: ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow
- pass 1, número de pasadas. Valores 1, 2 y 3.
- profile:v baseline, posibles valores: baseline, main, high, high10, high422, high444.

➤ **Codificación de un video en formato h.265:**

```
ffmpeg -r 30 -s 720x480 -pix_fmt yuv420p  
-i formule1@218@720x480@30.yuv -vcodec libx265  
-x265-params bframes=0:crf=25:qp=25  
-preset veryslow formule1.mkv
```

**crf <0..51.0>**, bitrate variable basado en el control de calidad.  
**rect, no-rect**, permite análisis de particiones de movimiento  
rectangulares de Nx2N and 2NxN.

**preset <0...9>**, -parámetro de eficiencia de compresión: 0  
(ultrafast), 1 (superfast), 2 (veryfast), 3 (faster), 4 (fast), 5  
(medium (default)), 6 (slow), 7 (slower), 8 (veryslow), 9  
(placebo).



## **profile**, requerimientos de compresion:

- **Perfiles de 8 bit:**
  - \* **main, main-intra, mainstillpicture (or msp for short)**
  - \* **main444-8, main444-intra, main444-stillpicture**
- **Perfiles de 10 bit:**
  - \* **main10, main10-intra**
  - \* **main422-10, main422-10-intra**
  - \* **main444-10, main444-10-intra**
- **Perfiles de 12 bit:**
  - \* **main12, main12-intra**
  - \* **main422-12, main422-12-intra**
  - \* **main444-12, main444-12-intra**

**me**, método de búsqueda de estimación de movimiento: hex (default), umh, star, sea y full.

**subme <0..7>**, Subpixel refinamiento. Default 2.

**merange <0...32768>**, rango de búsqueda: If the search range were any larger than this, another CTU row of latency would be required for reference frames. Range of values: an integer from 0 to 32768

**bframes <0..16>**, número máximo de frames b consecutivos.

**bitrate <integer>**, bitrate en bits/s.

**vbv-bufsize <integer>**, tamaño del buffer en kbits

**vbv-maxrate <integer>**, máximo local bitrate (kbits/s).Será usado cuando cuando vbv-bufsize es no cero también. Ambos parámetros son requeridos en modo CRF.

**qp <0...51>**, parámetro de calidad QP

# CAPTURA VIDEO Y AUDIO DE LOS DISPOSITIVOS

- Para conocer los dispositivos de video y audio de un ordenador ejecutar:

```
ffmpeg -list_devices true -f dshow -i dummy
```

```
[dshow @ 000000000030a3e0] "USB2.0 UVC VGA WebCam"
```

```
[dshow @ 000000000030a3e0] Alternative name
```

```
"@device_pnp_\\?\usb#vid_13d3&pid_5710&mi_00#7&d317257&0&0000#{65e8773d-8f56-11d0-a3b9-00a0c9223196}\global"
```

```
[dshow @ 000000000030a3e0] DirectShow audio devices
```

```
[dshow @ 000000000030a3e0] "Micrófono (Realtek High Definit"
```

```
[dshow @ 000000000030a3e0] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\Micr | fono (Realtek High Definit"
```

```
dummy: Immediate exit requested
```

Detecta la webcam "USB2.0 UVC VGA WebCam" y el micrófono "Micrófono (Realtek High Definit".

➤ **Extracción de los formatos soportados por la cámara de video:**

**ffmpeg -f dshow -list\_options true  
-i video="USB2.0 UVC VGA WebCam"**

```
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=640x480 fps=30 max s=640x480 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=640x480 fps=30 max s=640x480 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=160x120 fps=30 max s=160x120 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=160x120 fps=30 max s=160x120 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=176x144 fps=30 max s=176x144 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=176x144 fps=30 max s=176x144 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=320x240 fps=30 max s=320x240 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=320x240 fps=30 max s=320x240 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=352x288 fps=30 max s=352x288 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=352x288 fps=30 max s=352x288 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=640x480 fps=30 max s=640x480 fps=30
[dshow @ 00000000004aa400] pixel_format=yuyv422 min s=640x480 fps=30 max s=640x480 fps=30
```

➤ **Extracción de los formatos soportados por el micrófono:**

```
ffmpeg -f dshow -list_options true  
-i audio="Micrófono (Realtek High Definit"
```

```
[dshow @ 000000000053a400] DirectShow audio only device options (from audio devices)  
[dshow @ 000000000053a400] Pin "Capture" (alternative pin name "Capture")  
[dshow @ 000000000053a400] min ch=1 bits=8 rate= 11025 max ch=2 bits=16 rate= 44100
```

➤ **Grabación de video de la webcam en formato h.264 con el nombre video.mp4:**

```
ffmpeg -f dshow -i video="USB2.0 UVC VGA WebCam" -  
vcodec libx264 video.mp4
```

➤ **Grabación de audio:**

```
ffmpeg -f dshow -i audio="Micrófono (Realtek High Definit"  
audio.mp3
```

- Grabación de video y audio de los dispositivos en formato h.264 con el nombre **video.mp4**:

```
ffmpeg -f dshow -i video="USB2.0 UVC VGA WebCam"  
-f dshow -i audio="Micrófono (Realtek High Definit"  
-r 15 -bufsize 10000k -vcodec libx264 video.mp4
```

- Grabación del escritorio en formato h.264:

```
ffmpeg -f gdigrab -i desktop -r 10 -vcodec libx264 video.mp4
```

- Grabación del escritorio en formato h.264 con audio, framerate 10fps y tamaño 640x420:

```
ffmpeg -f gdigrab -i desktop -f dshow  
-i audio="Micrófono (Realtek High Definit"  
-vcodec libx264 -r 10 -s 640x420 escritorio.mp4
```

# STREAMING VIDEO VIA INTERNET

## ➤ Streaming video en formato h.264 capturado por la webcam:

```
ffmpeg -f dshow -i video="USB2.0 UVC VGA WebCam" -  
vcodec libx264 -an -bufsize 10000k  
-f mpegts udp://127.0.0.1:1234
```

## ➤ Recepción del video:

```
ffplay udp://127.0.0.1:1234
```

IP: 127.0.0.1, Puerto de salida 1234