

# Aplicación de FFmpeg

José María Pereyra, Alejandro Hector Gonzalez  
Creative Commons CC By

## ¿Qué es?

Software libre que funciona a través de la consola de comandos y nos permite realizar varias opciones sobre archivos de audio y video.

Alguna de estas opciones:

- ✓ Convertir de un formato a otro
- ✓ Cambiar la calidad o bitrate de un archivo
- ✓ Aumentar la ganancia de un archivo (subir el volumen)
- ✓ Ajustar las opciones de paneo (paneos izquierdo y derecho en un audio estéreo)
- ✓ Cambiar la escala de un video
- ✓ Opciones de rotación
- ✓ Opciones de compresión
- ✓ Cambio de codecs

## Instalación

- 1) Descargar la build static del sitio oficial: <https://ffmpeg.zeranoe.com/builds/>

**FFmpeg Builds**

FFmpeg is the leading multimedia framework to decode, encode, transcode, mux, demux, stream, filter and play. All builds require at least Windows 7 or Mac OS X 10.10.

20200623-ce297b4

Windows 64-bit

Static

GPL 3.0

**Version**  
Release builds are published approximately every six months, while Git builds are compiled multiple times a week. Git builds should be preferred unless a specific release version needs to be targeted.

**Architecture**  
Builds are provided for Windows and macOS, with very few configuration differences between the two. 32-bit Windows builds are provided for historical purposes, but will be removed in the future.

**Linking**  
Static builds include all components linked into the executable, while shared builds rely on libraries (.dll or .dylib) to operate. Dev builds provide header files, and will be used in development with a shared build. Most users should use the static build, while developers need to use both the shared and dev builds.

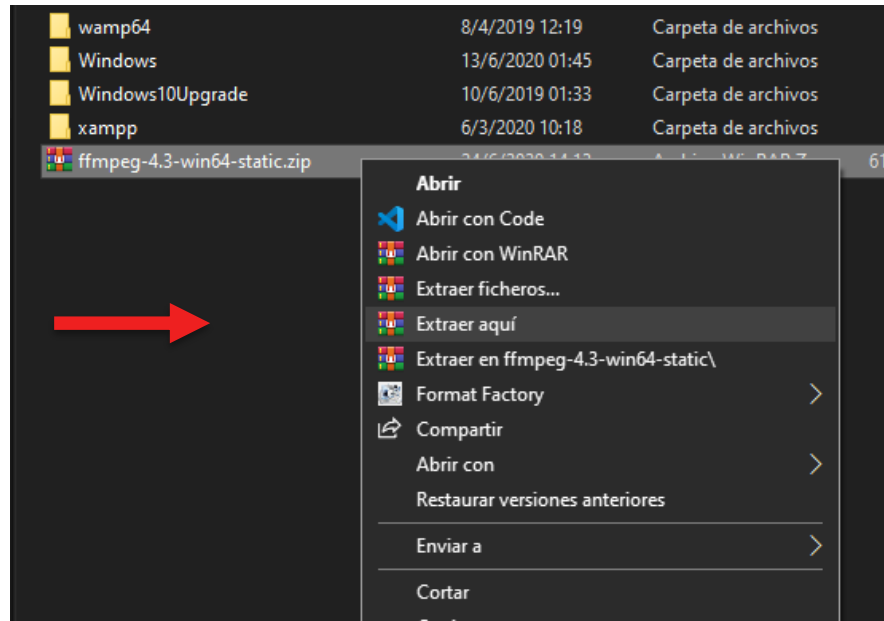
**License**  
All nightly builds are licensed as GPL 3.0, while GPL 3.0 and LGPL 3.0 builds are provided for releases. GPL 3.0 builds are more complete than LGPL 3.0 builds as they can include x264 and x265.

**Download Build**

Name	Size	Modified
ffmpeg-20200623-ce297b4-win64-static.zip	70.62 MB	Jun 24, 2020 6:21 AM



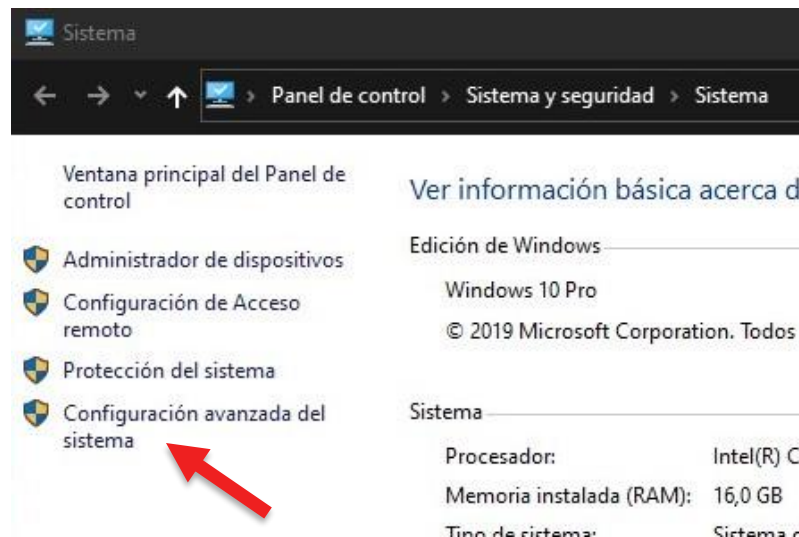
## 2) Descomprimir el archivo descargado



Utilizar cualquier software para descomprimir los datos. Ejemplo: WinRAR, 7-Zip

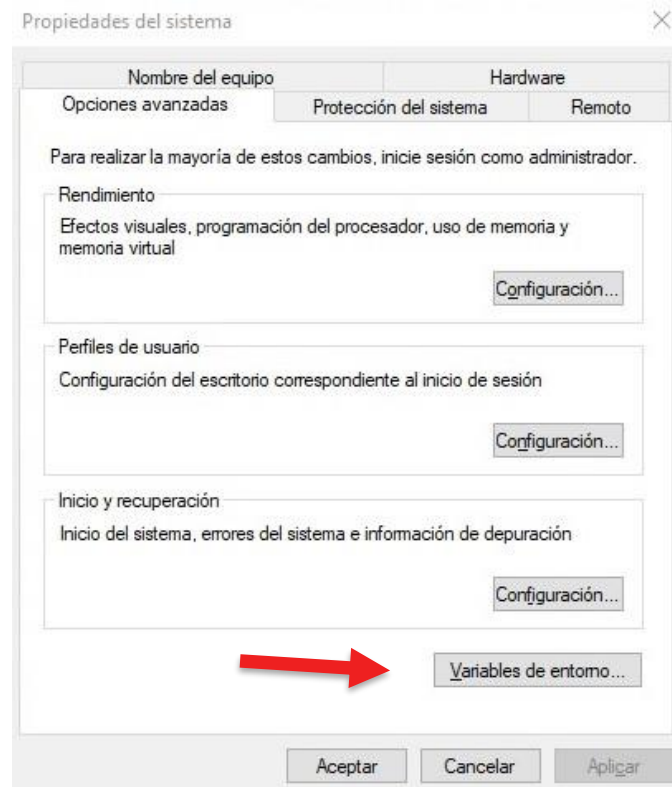
**IMPORTANTE:** es necesario recordar el directorio donde se encuentra el archivo para el próximo paso. Una vez que lo descomprimos, también podemos cambiarle el nombre a la carpeta por uno más sencillo (Ej: ffmpeg-4.3-win64-static por Ffmpeg)

## 3) Agregar una variable de entorno con la ruta a donde lo dejamos





Para eso, nos vamos a ir a Panel de control-> Sistema y seguridad-> Sistema. Luego seleccionamos la configuración avanzada



A continuación, ingresamos en "Variables de entorno..."



Agregamos una nueva con nombre "path" y escribimos la dirección absoluta donde se encuentra la carpeta bin

Video de instalación: <https://www.youtube.com/watch?v=X7wLMeiOjjM>



## Uso

Una vez creada la variable de entorno, podemos empezar a utilizar la herramienta. Desde la consola de Windows debemos dirigirnos a la carpeta donde se encuentra el video/audio en cuestión.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18363.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Joma>cd C:\Users\Joma\Videos\Desktop
```

Para verificar que la herramienta se instaló correctamente podemos ejecutar el comando *ffmpeg –versión*

Nos debería aparecer algo como esto:

```
C:\Users\Joma\Videos\Desktop>ffmpeg -version
ffmpeg version 4.3 Copyright (c) 2000-2020 the FFmpeg developers
built with gcc 9.3.1 (GCC) 20200621
configuration: --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --enable-libbluray --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus --enable-libshine --enable-lbsnappy --enable-libsoxr --enable-libsrt --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvmaf --enable-libvorbis --enable-libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libaom --enable-libgsm --disable-w32threads --enable-libbfx --enable-ffnvcodec --enable-cuda-llvm --enable-cuvid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-libopenmpt --enable-amf
libavutil      56. 51.100 / 56. 51.100
libavcodec     58. 91.100 / 58. 91.100
libavformat    58. 45.100 / 58. 45.100
libavdevice    58. 10.100 / 58. 10.100
libavfilter     7. 85.100 / 7. 85.100
libswscale     5.  7.100 / 5.  7.100
libswresample  3.  7.100 / 3.  7.100

C:\Users\Joma\Videos\Desktop>
```



## Comprimir un video mp4:

Para disminuir el tamaño de nuestro video, vamos a proceder a ejecutar el siguiente comando:

```
ffmpeg -i ENTRADA -q NUM SALIDA
```

```
C:\WINDOWS\system32\cmd.exe  
  
C:\Users\Joma\Desktop>ffmpeg -i mi_video.mp4 -q 15 nuevo_video.mp4
```

**-i (input):** indica que el siguiente parámetro es el archivo que se debe procesar

**ENTRADA:** nombre completo del archivo a comprimir. Ejemplo: mi\_video.mp4

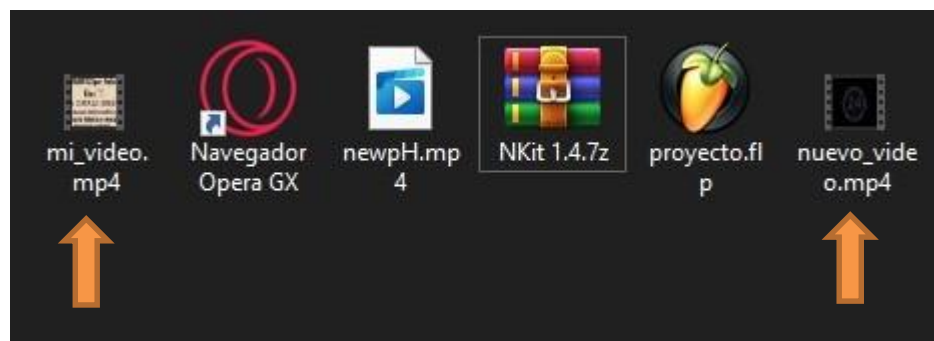
**-q:** indica la calidad o el bitrate que queremos otorgarle al archivo

**NUM:** valor numérico entero. El valor recomendado es entre 10 y 18. Con valores chicos tenemos un mayor bitrate (mayor calidad) pero a su vez mayor tamaño de video. Con valores grandes, tenemos un menor bitrate (menor calidad) pero el tamaño del video disminuye.

**SALIDA:** nombre completo del archivo resultante. Ejemplo: nuevo\_video.mp4

**IMPORTANTE:** No utilizar el espacio dentro de los nombres de los archivos para evitar errores de la herramienta. Ejemplo: si tenemos un archivo con nombre “mi video.mp4”, renombrarlo como “mi\_video.mp4”.

Como resultado, una vez que se termine de procesar el video, nos van a quedar ambos archivos





Si comparamos el tamaño que ocupan estos dos, vamos a notar una gran diferencia

