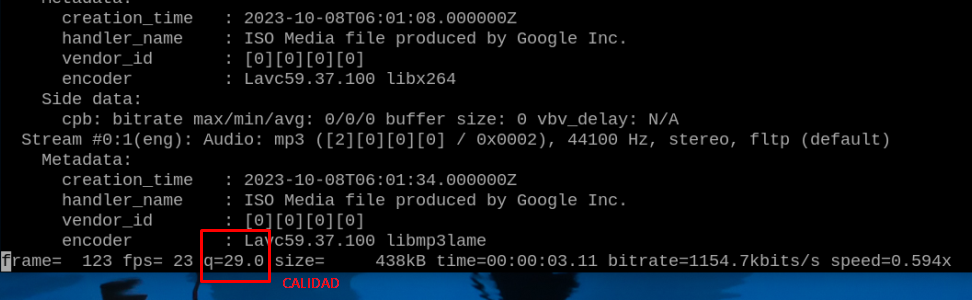
**.:: Streaming Desde la Terminal – Raspberry pi ::.**

**By Freddy Alcarazo | @surflaweb**

1. **Streaming a una calidad de 30FPS si audio:**

$ ffmpeg -re -i /home/pi5/Videos/videos-stream/naturaleza.mp4 \

-vcodec libx264 -f flv rtmp://127.0.0.1:1935/live/mystream



\*\* Esto es mucha potencia y hace que el procesamiento de la raspberry pi 5 sea más intensivo, se nota en la temperatura y velocidad del ventilador.

1. **Streaming con audio, pero todavía con la máxima calidad:**

**Agregar lo siguiente: -acodec aac -ar 44100 -b:a 128k**

$ ffmpeg -re -i /home/pi5/Videos/videos-stream/naturaleza.mp4 \

-vcodec libx264 -acodec aac -ar 44100 -b:a 128k -f flv \

rtmp://127.0.0.1:1935/live/mystream

1. **Streaming Forzando la tasa de cuadros (FPS) a 23.976 fps (NTSC 24)** - Estándar de cine NTSC (aproximadamente 24000/1001) - **Recomendado**

$ ffmpeg -re -i /home/pi5/Videos/videos-stream/naturaleza.mp4 \

-r 24000/1001 \

-vcodec libx264 -preset veryfast -tune zerolatency \

-acodec aac -ar 44100 -b:a 128k \

-f flv rtmp://127.0.0.1:1935/live/mystream



**Si se quiere cambiar a 30 fps (NTSC estándar):**

Usar -r 30000/1001

1. **Streaming video en bucle (repetir siempre el Streaming cuando termine):**

Para transmitir tu video en bucle infinito usando ffmpeg, solo necesitas agregar la opción:

**-stream\_loop -1**

$ ffmpeg -stream\_loop -1 -re -i /home/pi5/Videos/videos-stream/naturaleza.mp4 \

-r 24000/1001 \

-vcodec libx264 -preset veryfast -tune zerolatency \

-acodec aac -ar 44100 -b:a 128k \

-f flv rtmp://127.0.0.1:1935/live/mystream

1. **Streaming de una WebCam:**

Ver que se le pasa la ruta de la webcam: **-i /dev/video0**

$ ffmpeg -f v4l2 -framerate 24 -video\_size 1280x720 -i /dev/video0 \

-r 24000/1001 \

-vcodec libx264 -preset veryfast -tune zerolatency \

-acodec aac -ar 44100 -b:a 128k \

-f flv rtmp://127.0.0.1:1935/live/mystream

1. **Streaming de una WebCam y guardar video**

Usar la opción -map de ffmpeg para dividir la salida: una para el **stream RTMP**, y otra para el **archivo de video en disco**.

$ ffmpeg \

-f v4l2 -framerate 24 -video\_size 1280x720 -i /dev/video0 \

-r 24000/1001 \

-vcodec libx264 -preset veryfast -tune zerolatency \

-map 0:v \

-f flv rtmp://127.0.0.1:1935/live/mystream \

-map 0:v \

-f mp4 -movflags +faststart -y webcam\_record.mp4

**\*\* Si se desea guardar el video con un nombre con la fecha y tiempo usar lo siguiente:**

-f mp4 -movflags +faststart "webcam\_$(date +%Y-%m-%d\_%H-%M-%S).mp4"

**Quedaría así:**

$ ffmpeg \

-f v4l2 -framerate 24 -video\_size 1280x720 -i /dev/video0 \

-r 24000/1001 \

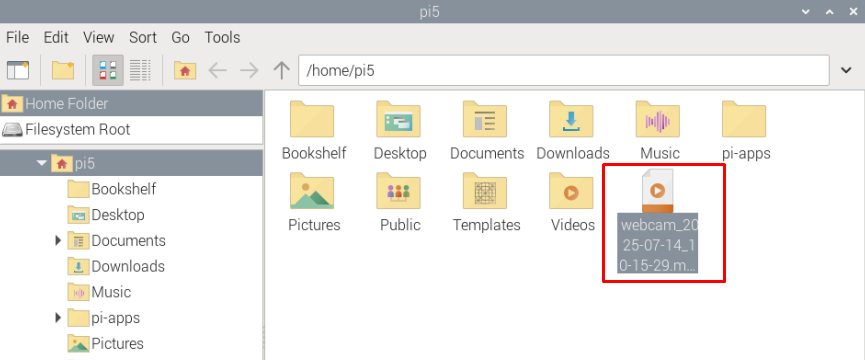
-vcodec libx264 -preset veryfast -tune zerolatency \

-map 0:v \

-f flv rtmp://127.0.0.1:1935/live/mystream \

-map 0:v \

-f mp4 -movflags +faststart "webcam\_$(date +%Y-%m-%d\_%H-%M-%S).mp4"

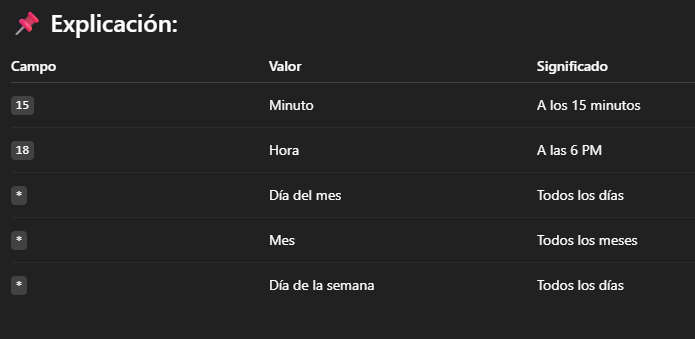


1. **Streaming programado usando CronJob**

Editar el cron del sistema para agregar la tarea/job.

$ sudo crontab -e

$ 15 18 \* \* \* /usr/bin/ffmpeg -re -i /home/pi5/Videos/videos-stream/naturaleza.mp4 -r 24000/1001 -vcodec libx264 -preset veryfast -tune zerolatency -acodec aac -ar 44100 -b:a 128k -f flv rtmp://127.0.0.1:1935/live/mystream > /home/pi5/stream\_log.txt 2>&1



\*\*\* Actualizar en su caso el tiempo y la ruta del video a transmitir.

**Confirmar que está activo el job:**

$ sudo crontab -l

**Ver proceso en caso de querer detenerlo:**

$ ps aux | grep ffmpeg

$ sudo kill xxx (Reemplazar xxx por el número del proceso del streaming).

Tutorial: <https://youtu.be/nV4cNQpvixo>

Video demo: <https://youtu.be/DS-1-Fm0ZBY>

**Perú – 14 de Julio del 2025.**

**FIN**