

Tu sistema de cortes de *GTA 6* con Claude Code

Le pasas un live, Claude encuentra los **gritos y reacciones**, te propone la lista y tú apruebas. Salen **cortes verticales** con subtítulos, título y zoom, más un **video largo** de mejores momentos.

19-11
de 2026: el día en que GTA 6 salió de verdad

7,91 M
de espectadores al mismo tiempo en su primer día

4 piezas
Claude Code más tres gratis: yt-dlp, ffmpeg y Whisper

Reel original de **@_aighostlab** · Datos revisados el 23-11-2026

GTA 6 ya salió. Y todos lo están *mirando*

El reel original decía "faltan 9 días". Eso quedó viejo: **Grand Theft Auto VI** salió el **19 de noviembre de 2026** en PS5 y Xbox Series X|S. Sin versión de PC por ahora.

US$ 3.790 M

En ventas del primer día, según Take-Two. Récord para el entretenimiento en cualquier medio.

125 M de horas

Vistas en streaming en las primeras 24 h, sumando Twitch, YouTube, Kick, SOOP y CHZZK.

4,5 veces

El récord anterior de un lanzamiento en streaming (VALORANT, con 1,74 M).

93 M en 24 h

Vistas del primer tráiler en YouTube. Récord de estreno para un video no musical.

Qué significa. Hay horas y horas de lives

Pico de 3,3 M en Twitch, 3,2 M en YouTube y 1,5 M en Kick. Nadie mira 8 horas de stream: la gente mira los mejores 40 segundos. Este sistema es para encontrarlos rápido. No te promete plata: te ahorra trabajo.

Fuentes: take2games.com · invenglobal.com (Streams Charts) · blog.youtube

Cuatro piezas y *tres son gratis*

`Claude Code`	**El cerebro** (necesita tu cuenta de Claude). Lee la transcripción y el volumen, elige los momentos y escribe y corre los comandos por ti.
`yt-dlp`	**Baja el VOD** (la grabación del live). Licencia Unlicense, dominio público. `github.com/yt-dlp/yt-dlp`
`ffmpeg`	**La navaja suiza del video.** Mide los picos de volumen, corta, pasa a 9:16 y quema los subtítulos. Licencia LGPL 2.1+. `ffmpeg.org`
`Whisper`	**Transcribe con tiempos.** Usa **whisper.cpp** (comando `whisper-cli`): corre local y rápido, en Mac y en Windows. Licencia MIT. `github.com/ggml-org/whisper.cpp`
`Opcionales`	**WhisperX** (tiempos por palabra, BSD-2) y la skill **/watch** (Claude "ve" el video, MIT). `m-bain/whisperX · bradautomates/claude-video`

Para instalar. En Mac o en Windows

Lo más fácil, pídeselo a Claude Code: `Instala yt-dlp, ffmpeg y whisper.cpp en mi computador y baja el modelo ggml-large-v3-turbo.bin.`

Mac (Terminal): `brew install yt-dlp ffmpeg whisper.cpp`

Windows (PowerShell): `winget install yt-dlp.yt-dlp` · `winget install Gyan.FFmpeg` · whisper.cpp: el .zip de Windows en sus Releases de GitHub.

Consigue el live y *arma la carpeta*

1. Baja el VOD (tuyo o con permiso)

```
$ yt-dlp -f "bv*[height<=1080]+ba/b" -o "live.%(ext)s" "URL_DEL_VOD"
```

Ojo: los términos de YouTube y Twitch no permiten bajar el live de otro sin permiso. Usa **tu propio VOD**, uno con **permiso escrito** del streamer o uno de una **campaña de clipping** donde el streamer lo pide.

2. Una carpeta por live

```
cortes-gta6/
└── stream-01/
    ├── live.mp4          # el VOD que bajaste
    ├── permiso.txt       # el permiso del streamer
    └── cortes/           # aquí salen los videos
```

Abre la Terminal (Mac) o PowerShell (Windows) en `stream-01` y escribe `claude`. Así Claude Code trabaja solo dentro de esa carpeta. En la página siguiente está el prompt.

El prompt *(parte 1)*

COPIA Y PEGA EN CLAUDE CODE

En esta carpeta está live.mp4, el VOD de un stream de GTA 6 que tengo permiso para cortar. Quiero un sistema de cortes. Trabaja por etapas y no avances a cortar sin mi aprobación.

ETAPA 1 · Transcribir con tiempos
Extrae el audio a audio.wav (16 kHz, mono) con ffmpeg. Transcríbelo con whisper-cli (whisper-cli.exe en Windows) y el modelo ggml-large-v3-turbo.bin, idioma español, y guarda transcripcion.srt.

ETAPA 2 · Detectar picos de volumen
Mide la sonoridad con el filtro ebur128 de ffmpeg y guarda volumen.log. Busca los picos: momentos en que el volumen sube fuerte sobre el promedio del live (gritos, risas, golpes en la mesa). Ignora los picos que sean solo música o efectos del juego sin voz encima.

ETAPA 3 · Encontrar las reacciones
Cruza los picos con la transcripción. Prioriza:
- gritos y sorpresas ("no puede ser", "¿qué?", risas largas)
- momentos que se entienden sin contexto
- remates: algo pasa y el streamer reacciona

Descarta cinemáticas del juego sin reacción y spoilers de la historia.

ETAPA 4 · Proponer la lista (y parar)
Escribe cortes.md con 10 cortes, del más fuerte al más débil. Por cada uno: número · inicio y fin (entre 20 y 60 s, con 2 s de aire antes del pico) · qué pasa · por qué funciona · título corto (máx. 6 palabras).
Muéstrame la lista y ESPERA. Yo te digo cuáles aprobar, cuáles mover y cuáles botar.

El prompt *(parte 2)*

SIGUE EN EL MISMO MENSAJE

```
ETAPA 5 · Cortes verticales (solo los aprobados)
Por cada corte aprobado:
- córtalo con ffmpeg desde live.mp4
- pásalo a 9:16, 1080x1920. Si hay cámara del streamer, encuádrala arriba y el juego abajo; si no, recorta centrado en la acción
- quema subtítulos grandes desde la transcripción: máx. 2 líneas, letra gruesa con borde, en el tercio inferior pero lejos del borde donde van los botones de la app
- pon el título arriba los primeros 3 segundos
- haz un zoom suave (unos 15 %) justo en el pico del grito y vuelve
- guarda en cortes/corte01.mp4, corte02.mp4…

ETAPA 6 · Video largo de mejores momentos
Une los cortes aprobados en mejores_momentos.mp4, horizontal 16:9, del más fuerte al más débil. Entre momento y momento deja 3 segundos libres y anótalos en narracion.md: ahí grabo mi comentario, para que sea un video mío y no una copia.

ETAPA 7 · Control de calidad
Revisa cada archivo: que dure lo que dice cortes.md, que tenga audio, que los subtítulos calcen con la voz y que nada quede cortado. Dame una tabla con archivo, duración y título.

REGLAS
No borres live.mp4. No subas nada a ninguna parte. Si un comando falla, muéstrame cuál y por qué antes de seguir.
```

Un live de 6 a 8 horas **tarda** en bajar y transcribir. Lo que sí sale en minutos es el corte, una vez aprobada la lista.

Lo que Claude *corre por ti*

No necesitas escribir esto. Pero si algo falla, así sabes qué está pasando. En Windows sirven igual en PowerShell: escribe el último en una sola línea, sin las ****.

```
# picos de volumen (gritos y reacciones)
$ ffmpeg -i live.mp4 -af ebur128=framelog=verbose -f null - 2> volumen.log

# transcripción con tiempos
$ ffmpeg -i live.mp4 -ar 16000 -ac 1 audio.wav
$ whisper-cli -m ggml-large-v3-turbo.bin -f audio.wav -l es -osrt

# un corte a 9:16 con subtítulos quemados
$ ffmpeg -ss 01:23:10 -to 01:23:55 -i live.mp4 \
  -vf "crop=ih*9/16:ih,scale=1080:1920,subtitles=corte1.srt" \
  -c:a aac corte1.mp4
```

El zoom. Sale de ffmpeg también

Se hace con el filtro `zoompan` o con un `crop` animado justo en el segundo del pico. Si quieres subtítulos que aparecen palabra a palabra, pídele que use `WhisperX` para los tiempos.

Para no meterte *en problemas*

01 **Pide permiso al streamer**

El stream es obra del streamer. Recortarlo y subirlo sin permiso puede terminar en reclamo o strike por copyright. El "uso justo" es una defensa caso a caso en EE. UU., no un permiso, y en Chile y LatAm es más estrecho.

02 **Mejor aún: campañas de clipping**

Muchos streamers quieren clippers. En plataformas como Whop Content Rewards, el streamer o una marca abre una campaña y paga por cada 1.000 vistas verificadas. Ahí el permiso viene incluido. Lee las reglas de cada campaña.

03 **YouTube: contenido reutilizado**

No se monetizan los clips montados "con poca o ninguna narrativa". Aplica **aunque tengas permiso**, y puede quitarle la monetización a todo el canal. Agrega tu voz, tu comentario o edición de verdad.

04 **Rockstar tolera, no licencia**

Su política acepta el uso no comercial de los fans y baja cinemáticas aisladas, spoilers y filtraciones. No da permiso general para monetizar videos de GTA VI.

05 **Cuidado con la música**

La radio del juego o la música del streamer activa Content ID. Twitch, además, silencia VODs con audio protegido.

Lo que *más se repite*

1. Bajar el live de cualquiera
Rompe los términos de YouTube y Twitch. Trabaja con tu VOD, con permiso o en una campaña.

2. Subir el compilado tal cual
Sin narración ni edición propia, YouTube lo trata como contenido reutilizado. Graba tu comentario en los huecos.

3. Dejar que corte sin revisar
Por eso el prompt para en la etapa 4. Tú eliges: Claude propone.

4. Esperar todo "en minutos"
Bajar y transcribir un live largo toma su rato. Déjalo corriendo y vuelve después.

5. Subir cinemáticas o spoilers
Es justo lo que Rockstar baja. Corta reacciones, no la historia del juego.

6. Usar datos viejos
"Faltan 9 días" ya no aplica: GTA 6 salió el 19-11-2026.

Sigue a @noisy.ia para más IA aplicada a tu trabajo

Reel original de `@_aighostlab` · Hecho con `Claude Code`