

INFORME TÉCNICO Y DIDÁCTICO

Restaurador Super 8

Qué hace la aplicación, qué significa cada parámetro, qué modelos de inteligencia artificial utiliza, qué hemos aprendido construyéndola y hasta dónde se podría llegar con una GPU y unos modelos mejores.



Preparado para: Aldabón

Fecha: 21 de septiembre de 2026

Nivel: bachillerato (no hace falta saber programar)

Carpeta de la app: D:\Proyectos\Claude\mejorar_Super_8mm

Índice

1. Resumen en una página	3
2. Conceptos básicos	4
2.1 Fotograma y fotogramas por segundo (fps)	4
2.2 Píxel y resolución	4
2.3 Frecuencia espacial: la idea clave	4
2.4 Grano y ruido	4
2.5 Códec, contenedor y compresión	5
2.6 Profundidad de color (8 bits)	5
2.7 GPU, CUDA, Vulkan y NVENC	5
3. Cómo funciona la app	6
3.1 El recorrido de un vídeo	6
3.2 Qué hay por dentro	6
4. Los parámetros, uno a uno	7
4.1 Cadencia y encuadre	7
4.2 Estabilización	7
4.3 Parpadeo (deflicker)	8
4.4 Polvo y rayas	9
4.5 Ruido y grano	9
4.6 Color y nitidez del máster	10
4.7 Mejora (copia para ver)	11
4.8 Caras (solo en la copia)	11
4.9 Exportación	12
4.10 El plan automático	12
5. Diagnóstico y medidas objetivas	13
6. Modelos de IA y algoritmos que usa la app	14
6.1 Modelos de IA	14
6.2 Algoritmos clásicos (sin IA)	14
6.3 Motores de ejecución	15
7. ¿Mejor GPU y mejores modelos, mejor resultado?	16
7.1 Dónde se va el tiempo	16
7.2 Una GPU más potente: más velocidad, misma imagen	16
7.3 Modelos mejores: aquí sí cambia la imagen	16
7.4 Veredicto	17
8. Lo que hemos aprendido en este proyecto	18
9. Recomendaciones y próximos pasos	19
10. Glosario	19
11. Fuentes	20

1. Resumen en una página

Una película Super 8 o de 8 mm es una tira de fotogramas diminutos (unos $5,8 \times 4$ mm en Super 8). Al digitalizarla aparecen sus defectos: **temblor** (la película no pasaba siempre igual por la ventanilla), **parpadeo** de brillo, **polvo y rayas**, **grano**, colores desvaídos y poca resolución. La app corrige esos defectos en dos pasos claramente separados:

- **Máster restaurado (fiel).** Limpia y estabiliza sin inventar nada: todo lo que se ve estaba en la película. Se guarda sin pérdida (FFV1) para archivarlo.
- **Copia mejorada (para ver).** Parte del máster y usa inteligencia artificial para aumentar la resolución y la fluidez (más fotogramas por segundo). Aquí sí se *estima* detalle que no estaba, por eso es una copia aparte.

Antes de procesar, la app **diagnostica** el vídeo (mide ruido, temblor, parpadeo, motas, nitidez real...) y el botón «**Analizar y proponer parámetros**» calcula la receta más eficiente para el tiempo disponible, midiendo la velocidad real de tu equipo. Al terminar genera un **informe HTML** con lo que ha cambiado (medido, no supuesto) y un **clip comparativo**, y el comparador de la web permite ver original y restaurado a pantalla completa.

La conclusión principal

La calidad final la limita sobre todo el **escaneado**: si la digitalización ya perdió el detalle, ni la mejor GPU ni el mejor modelo lo recuperan de verdad; solo lo imitan. Una GPU más potente hace lo mismo **más rápido**; los modelos más modernos (de vídeo, con memoria temporal, o de difusión) sí pueden dar **mejor aspecto**, pero piden entre 16 y 32 GB de memoria de vídeo y más tiempo, y se inventan más cosas. El detalle está en el capítulo 7.

Tu equipo: **NVIDIA GeForce RTX 3060 (12 GB)**, controlador 528.49 (CUDA 12.0). La app usa la GPU de tres formas: PyTorch con CUDA 11.8 para los modelos .pth y las caras, Vulkan (ncnn) para RIFE y Real-ESRGAN, y NVENC para codificar el vídeo final.

2. Conceptos básicos

Estos conceptos aparecen una y otra vez en los parámetros. Merece la pena entenderlos antes de seguir.

2.1 Fotograma y fotogramas por segundo (fps)

Un vídeo es una sucesión de imágenes fijas (**fotogramas**) que, pasadas deprisa, parecen movimiento. La **cadencia** es cuántas se muestran por segundo. El cine doméstico se rodaba a **16 fps** (8 mm) o **18 fps** (Super 8), a veces a 24. La televisión europea usa 25 fps y los ordenadores 30 o 60.

Al digitalizar una película de 18 fps en un fichero de 25 fps, el equipo de telecine **repite** algunos fotogramas para rellenar (patrón de duplicados). Si la app detecta ese patrón puede **deducir** la cadencia real y quitar los repetidos; si no lo detecta, **conserva** la del fichero para no cambiar la duración ni la velocidad de la acción. Esta prudencia se añadió tras un caso real: un vídeo de 28,268 fps que, suponiendo 18 fps, habría durado 5 minutos en vez de 3.

2.2 Píxel y resolución

Cada fotograma es una rejilla de **píxeles**. La **resolución** es su tamaño: 720×576 (definición estándar), 1280×720 (720p), 1920×1080 (1080p, «Full HD»), 3840×2160 (4K). Tener más píxeles **no** equivale a tener más detalle: si amplías una foto borrosa, obtienes una foto borrosa más grande. Por eso la app distingue entre la resolución del fichero y la **resolución útil** (el detalle que de verdad contiene).

2.3 Frecuencia espacial: la idea clave

Igual que un sonido se compone de graves y agudos, una imagen se compone de **frecuencias espaciales**: las bajas son zonas suaves (cielo, una pared) y las altas son bordes finos y texturas (pelo, hojas, letras). La **transformada de Fourier** (FFT) descompone la imagen en esas frecuencias, como un ecualizador gráfico.

En una película escaneada, a partir de cierta frecuencia ya no hay imagen, solo **ruido**. La app busca ese punto, el **corte espectral**: la frecuencia donde la señal cae por debajo del «suelo» de ruido. Se expresa entre 0 y 0,5 (0,5 = el detalle más fino que cabe en un píxel). Un corte de 0,25 significa que el vídeo tiene la mitad del detalle que su resolución permitiría; en ese caso escalarlo ×4 no aporta nada real.

2.4 Grano y ruido

El **grano** son los cristales de plata o tinte de la emulsión: forma parte de la textura del cine y conviene conservar algo. El **ruido** es cualquier variación aleatoria no deseada (del escáner, de la compresión). Ambos cambian de un fotograma a otro, y eso permite reducirlos **promediando fotogramas vecinos** (lo que es fijo se refuerza, lo aleatorio se cancela). La app mide el ruido con el método de **Immerkær** y lo expresa como σ (sigma, desviación típica) en niveles de gris.

2.5 Códec, contenedor y compresión

El **contenedor** (.mp4, .mkv, .avi) es la caja; el **códec** es el idioma en que está escrita la imagen dentro. Los códecs **con pérdida** (H.264, H.265) tiran información que el ojo apenas nota para ocupar poco. Los **sin pérdida** (FFV1) guardan exactamente cada píxel: ocupan mucho, pero sirven de original de archivo. El **CRF** regula la calidad de H.264/H.265: más bajo = mejor calidad y más peso (16-18 es prácticamente transparente).

No todos los códecs se ven en el navegador: MPEG-4 parte 2, DV, FFV1 y a veces H.265 no se reproducen. Por eso el comparador crea automáticamente un **proxy** H.264 de 720p (un duplicado ligero solo para verlo), guardado en datos/proxies.

2.6 Profundidad de color (8 bits)

Cada canal de color (rojo, verde, azul) se guarda con 8 bits = 256 niveles. Es suficiente para ver, pero al hacer muchas correcciones seguidas pueden aparecer escalones en degradados (*banding*). La app trabaja en 8 bits; pasar a 16 bits es una de las mejoras pendientes.

2.7 GPU, CUDA, Vulkan y NVENC

La **CPU** es el procesador general: pocos núcleos muy listos. La **GPU** (tarjeta gráfica) tiene miles de núcleos sencillos que hacen la misma operación sobre muchos píxeles a la vez: ideal para imágenes y redes neuronales. Para hablar con la GPU hacen falta «idiomas»:

Tecnología	Qué es	Qué hace en la app
CUDA	Plataforma de cálculo de NVIDIA. La usa PyTorch.	Modelos .pth (spandrel), caras (GFPGAN) y filtros acelerados.
Vulkan / ncnn	Interfaz gráfica abierta; ncnn es un motor de redes neuronales ligero.	RIFE (interpolación) y Real-ESRGAN ncnn (escalado).
NVENC	Chip codificador de vídeo dedicado dentro de la GPU.	Codifica el MP4 final H.264/H.265 mucho más rápido que la CPU.
VRAM	Memoria propia de la GPU (12 GB en tu RTX 3060).	Limita el tamaño de los modelos y de los bloques (tiles) que caben.

Una lección aprendida

La versión de CUDA que admite el controlador debe ser **igual o mayor** que la de PyTorch. Tu controlador admite CUDA 12.0; PyTorch con CUDA 12.6 fallaba con `cudaErrorDevicesUnavailable`. Por eso el instalador elige ahora la variante según el controlador (CUDA 11.8 en tu caso) y comprueba la GPU reservando memoria de verdad.

3. Cómo funciona la app

3.1 El recorrido de un vídeo

MÁSTER FIEL (sin inventar) → FFV1 sin pérdida



Al final: informe HTML con medidas, clip comparativo y comparador a pantalla completa.

1. **Subir o indicar la ruta** del vídeo digitalizado. La app muestra una miniatura.
2. **Diagnosticar**: analiza una muestra y mide los defectos (capítulo 5).
3. **Analizar y proponer parámetros**: con el diagnóstico, una prueba de velocidad real y el tiempo que le des, rellena todos los controles con la receta propuesta y explica por qué.
4. **Muestra** (opcional): procesa unos segundos para comprobar el resultado antes de la bobina completa.
5. **Procesar**: ejecuta la cadena completa. La barra muestra la etapa, el porcentaje y el tiempo restante estimado (ETA).
6. **Revisar**: informe HTML, clip comparativo y comparador (cortinilla, lado a lado, solo A o B; teclas F, Esc, espacio y flechas).

3.2 Qué hay por dentro

Pieza	Función
EJECUTAR.bat	Arranca todo. A partir de la segunda ejecución cierra la consola anterior, el servidor, los trabajos en curso y libera el puerto 8765 antes de volver a empezar.
Instalador (PowerShell)	Crea el entorno de Python, elige PyTorch según el controlador, descarga FFmpeg con NVENC si hace falta, y guarda cachés en el disco de la app (no en C:).
Servidor (FastAPI)	Programa local en 127.0.0.1:8765 que atiende a la página web. Tiene dos colas: una rápida (diagnóstico, plan) y otra para procesos largos.
Página web	La interfaz: controles, plan automático, lista de trabajos, comparador y Ayuda.
restaura8 (motor)	Módulos de Python: análisis, filtros, modelos, plan, medidas, informe.
herramientas\	FFmpeg, RIFE ncnn, Real-ESRGAN ncnn y los modelos descargados.

Cada trabajo tiene su carpeta de salida con: *01_normalizado* (referencia para medir), el máster *.mkv*, la copia mejorada *.mp4*, la comparación, la receta usada (*_receta.json*) y el informe (*_informe.html* y *.json*).

4. Los parámetros, uno a uno

Los controles están agrupados igual que en la página web. No hace falta tocarlos: el plan automático los rellena. Esta guía sirve para entender qué ha decidido la app y para retocar si algo no te convence. Los tres **preajustes** son puntos de partida:

Preajuste	Para qué	Rasgos principales
Fiel	Archivo histórico: nada inventado.	Sin copia mejorada, poca reducción de ruido (40 %) y mucho grano (80 %), color suave, no suaviza el movimiento de cámara.
Equilibrado	Uso general (valores por defecto).	Máster limpio + copia x2 a 25 fps con un 70 % de peso de IA.
Máximo	Máximo realce para ver en pantalla grande.	Real-ESRGAN x4plus al 100 %, 60 fps con RIFE 4.6, H.265 CRF 16, polvo umbral 20. Lento (horas por bobina) y puede inventar detalle.

4.1 Cadencia y encuadre

Preparan el vídeo antes de restaurarlo (etapa de **normalización**).

Parámetro	Qué es y cómo actúa	Valores	Consejo
Cadencia de rodaje	Velocidad a la que se filmó (fps reales). Si el diagnóstico la <i>deduce</i> por el patrón de duplicados, se usa; si no, se conserva la del fichero.	16 / 18 / 24	Deja la propuesta. Cámbiala solo si sabes la cadencia real.
Quitar duplicados	Elimina los fotogramas repetidos que añadió el telecine. Solo se aplica cuando la cadencia es deducida, para no acelerar la acción por error.	Sí / No	Sí, cuando el diagnóstico diga «deducida».
Desentrelazar	Algunas capturas por vídeo analógico mezclan dos medias imágenes (líneas pares e impares) y se ven «peines» en el movimiento. El filtro bwdif las reconstruye.	Sí / No	Solo si ves peines; el diagnóstico lo detecta.
Recorte	Quita los bordes negros o la ventanilla del proyector (x, y, ancho, alto).	píxeles	Vacío = sin recorte. La app propone uno si detecta bordes.

4.2 Estabilización

Quita el temblor. La app sigue cientos de puntos entre fotogramas (**Lucas-Kanade**) y descarta los que se mueven por su cuenta, como una persona que cruza (**RANSAC**). Así obtiene el movimiento de toda la imagen: desplazamiento, giro y escala. Después separa dos componentes suavizando la trayectoria:

Temblor y trayectoria

Imagina la posición del encuadre como una línea que zigzaguea en el tiempo. Suavizar esa línea (como dibujarla a pulso con mano firme) y mover cada fotograma hasta la línea suave elimina el temblor. Se añadieron límites para que una estimación errónea no desplace la imagen de forma catastrófica.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Activar	Enciende o apaga la estabilización.	Sí / No	Casi siempre sí.
Corrección del vaivén	Qué parte del temblor rápido (la película bailando en la ventanilla) se elimina.	0-100 %	100 %: ese temblor nunca fue intencionado.
Suavizado de cámara	Qué parte del movimiento lento del operador (paneos, pulso) se suaviza.	0-100 %	0-35 %. Más alto puede parecer una grúa artificial.
Ampliación	Al compensar, aparecen bordes vacíos; un pequeño zoom los oculta.	×1,00-×1,10	×1,03-×1,05. Cada 1 % recorta imagen.
Corregir el giro	Compensa también la rotación.	Sí / No	Sí.

4.3 Parpadeo (deflicker)

La exposición variaba de fotograma a fotograma (obturador, lámpara del proyector). La app compara el brillo de cada fotograma con la media de sus vecinos y lo iguala.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Activar	Enciende la corrección.	Sí / No	Sí si el diagnóstico mide parpadeo.
Por zonas	Corrige por cuadrícula en vez de globalmente: sirve cuando el centro brilla más que las esquinas y además varía (<i>hotspot</i> del proyector).	Sí / No	Solo si el diagnóstico lo recomienda.
Fuerza	Cuánto se acerca cada fotograma a la media.	0-100 %	100 % normalmente.
Ventana	Cuántos fotogramas vecinos forman la media. Pequeña = corrige cambios rápidos; grande = más estable pero puede aplanar cambios de luz reales.	3-40	12 (≈ medio segundo).

4.4 Polvo y rayas

Una mota de polvo aparece en **un solo fotograma**; el contenido real persiste en los vecinos. La app calcula el **flujo óptico** (hacia dónde se mueve cada píxel) con el algoritmo **DIS**, «dobla» el fotograma anterior y el siguiente para que encajen con el actual, y marca como mota lo que difiere de ambos. Luego rellena la mota con los vecinos.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Quitar motas	Activa la detección y el relleno. El plan la activa si el diagnóstico encuentra motas (aunque sean pocas).	Sí / No	Sí en casi todas las películas.
Umbral	Diferencia de brillo mínima para considerar mota. Se iguala antes la ganancia local para no confundir parpadeo o grano con polvo.	12-60	Propuesto: 32 / 28 / 24 según suciedad. Más bajo = más agresivo.
Tamaño máximo	Superficie máxima (px ²). Algo más grande probablemente es un objeto real que se mueve deprisa.	50-2000	400 (800 en «Máximo»).
Quitar rayas	Las rayas verticales duran muchos fotogramas, así que no sirve comparar vecinos: se buscan columnas anómalas y se rellenan con inpainting de Telea (rellenar desde los bordes hacia dentro).	Sí / No	Sí si se ven rayas.
Umbral de rayas	Cuánto debe destacar una columna.	2-15	4,5-5.

4.5 Ruido y grano

El promedio temporal con compensación de movimiento es la herramienta más potente: alinea los vecinos con el flujo óptico y los mezcla. Después se **devuelve** una parte del grano original para que no parezca plástico.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Reducción temporal	Intensidad del promedio entre fotogramas alineados.	0-100 %	40-60 % fiel; 90 % si se va a escalar con IA.
Grano conservado	Proporción del grano original que se reinyecta.	0-100 %	50 % (80 % fiel, 20 % máximo).
Reducción espacial	NL-means : busca trozos parecidos dentro de la misma imagen y los promedia. Muy lenta y puede borrar textura.	0-100 %	0 salvo ruido extremo.

4.6 Color y nitidez del máster

El color se corrige **por escena**: la app detecta los cortes comparando histogramas de color (espacio HSV) y calcula una corrección para cada plano.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Corregir color	Activa la corrección por escena.	Sí / No	Sí.
Balance de blancos	Neutraliza dominantes (el típico tono rojizo o magenta de la película envejecida) acercando los grises a gris.	0-100 %	60-100 %.
Niveles	Estira el contraste para que el negro sea negro y el blanco blanco (tabla LUT).	0-100 %	50-90 %.
Saturación	Intensidad de los colores.	$\times 0,6$ - $\times 1,6$	$\times 1,00$ - $\times 1,10$.
Nitidez	Máscara de enfoque adaptativa : realza bordes pero no zonas planas (donde solo realzaría ruido).	0-100 %	15-45 %.

4.7 Mejora (copia para ver)

Aquí actúa la inteligencia artificial. Nada de esto afecta al máster.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Generar copia	Crea el MP4 mejorado.	Sí / No	Sí, salvo archivo puro.
Escalado	Multiplica la resolución. El plan lo limita con el corte espectral: $\geq 0,45 \rightarrow \times 2$; $\geq 0,30 \rightarrow \times 1,75$; $\geq 0,20 \rightarrow \times 1,5$; menos $\rightarrow \times 1$.	$\times 1 / \times 2 / \times 4$	El propuesto. $\times 4$ sobre un escaneado pobre solo engorda el fichero.
Motor de escalado	Real-ESRGAN ncnn (Vulkan), modelo PyTorch (spandrel, CUDA) o Lanczos (fórmula matemática clásica, rápida, no inventa).	auto / ...	Automático.
Modelo ncnn / PyTorch	Qué red neuronal escala (capítulo 6).	lista	general-x4v3 para película real.
Peso de la IA	Mezcla entre la imagen de la IA y la de Lanczos.	0-100 %	70 %: realce sin aspecto de dibujo.
Cadencia final	fps de la copia. Si es mayor que la de rodaje, hay que interpolar (crear fotogramas intermedios).	rodaje / 24-60	25 para TV; 50-60 da fluidez «de vídeo».
Interpolación	RIFE (IA, calcula el movimiento), minterpolate (FFmpeg, clásico) o repetir fotogramas. RIFE se aplica por escenas para no mezclar planos distintos.	auto / ...	Automático (RIFE).
Nitidez final / grano añadido	Acabado tras la IA: un poco de nitidez y un grano fino para disimular el aspecto demasiado liso.	0-100 % / 0-60 %	35 % / 15 %.
Lienzo	16:9 coloca la imagen 4:3 en el centro con bandas laterales a una altura estándar (720, 1080, 1440, 2160). «Original» conserva 4:3.	16:9 / original	16:9 para televisores.
Orden	Interpolar primero o escalar primero. Con modelos pesados conviene escalar primero: se escalan menos fotogramas.	auto / ...	Automático.

4.8 Caras (solo en la copia)

Detecta rostros con **YuNet** y los reconstruye con **GFPGAN**. Atención: el modelo dibuja una cara *plausible*, no necesariamente la de esa persona.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Reconstruir caras	Activa la función.	Sí / No	Probar en una muestra antes.
Intensidad	Mezcla entre la cara reconstruida y la original.	0-100 %	40-60 %: más alto puede cambiar el parecido.
Cara mínima	Las caras más pequeñas se dejan como están (hay poca información y el riesgo de inventar es mayor).	40-300 px	80 px.

4.9 Exportación

Formatos de salida.

Parámetro	Qué es y cómo actúa	Valores	Consejo
Máster	FFV1 (.mkv) sin pérdida o ProRes 422 HQ (.mov, casi sin pérdida, cómodo en editores).	ffv1 / prores	FFV1 para archivar.
Copia	H.264 (compatible con todo) o H.265 (la mitad de tamaño a igual calidad; algunos equipos antiguos no lo reproducen).	h264 / h265	H.264 si dudas.
Calidad (CRF)	Menor = mejor y más pesado. Con NVENC se traduce a su escala equivalente (CQ).	12-28	16-18.
Codificador	NVENC (GPU) o CPU (libx264/libx265).	auto / cpu	Automático; la CPU comprime algo mejor pero es mucho más lenta.
Limpiar audio	Reduce siseo y clics si el vídeo trae sonido.	Sí / No	Sí.

4.10 El plan automático

El botón «**Analizar y proponer parámetros**» hace tres cosas: (1) diagnostica, (2) cronometra cada etapa pesada en tu equipo con unos fotogramas reales —descartando el primero, que incluye el arranque del modelo— y (3) combina esas velocidades con las de trabajos anteriores (historial de informes) para estimar el tiempo total de cada alternativa. Elige la mejor que cabe en tu presupuesto y aplica la receta sin que tengas que tocar nada.

Opción del plan	Qué hace
Tiempo disponible	Presupuesto total: 20 min, 1 hora, 3 horas o «toda la noche» (10 h).
Prioridad	Rapidez, Equilibrio o Calidad: cuánto pesa el tiempo frente a la calidad al elegir.
Máximo	Tope de resolución de salida: 1080p, 1440p o 4K.
Limpieza rápida (automática)	Si ninguna opción completa cabe en el tiempo, el plan considera una limpieza simplificada de las etapas más lentas y lo indica en la explicación.

La propuesta explica cada decisión (por ejemplo: «corte espectral 0,27: escalar más de $\times 1,5$ no añade detalle real») y muestra una tabla de alternativas con su duración.

5. Diagnóstico y medidas objetivas

Una restauración se puede juzgar a ojo, pero el ojo se engaña (sobre todo comparando vídeos de tamaños distintos). Por eso la app **mide** antes y después con las mismas fórmulas, tomando como referencia *01_normalizado* (el original ya sin duplicados ni bordes). El informe final lo presenta en el apartado «Medido, no supuesto».

Medida	Cómo se calcula	Cómo leerla
Ruido σ	Método de Immerkær: un filtro que anula la imagen suave y deja solo el ruido; se promedia su valor absoluto.	Niveles de gris. Debe bajar; bajar mucho = aspecto plástico.
Temblor	Se calcula a partir de la trayectoria de estabilización: cuánto se aparta el encuadre de su versión suavizada. (Medirlo en la imagen engañaba: los bordes fijos de la ventanilla lo falseaban.)	Píxeles. Debe acercarse a 0.
Parpadeo	Variación del brillo medio entre fotogramas consecutivos, sin contar cortes.	Porcentaje. Debe bajar.
Nitidez	Desviación típica del laplaciano (detector de bordes) dividida por el contraste, para que no «suba» solo por subir el contraste.	Sube si hay más bordes finos.
Corte espectral	Frecuencia (FFT) donde la señal cae al nivel del ruido. No depende del tamaño de la imagen.	0-0,5. Cuanto más alto, más detalle real.
Resolución útil	Corte espectral traducido a píxeles: el tamaño que de verdad aprovecha el detalle presente.	Si es mucho menor que la resolución, sobra escalar.
Motas	Porcentaje de píxeles marcados como polvo y su persistencia temporal.	Decide si se activa «Quitar motas» y con qué umbral.
Cadencia	Patrón de fotogramas repetidos (diferencia casi nula con el anterior).	deducida / contenedor / conservada.

Un matiz

En una película que ya es nítida, la nitidez medida puede bajar un poco tras la restauración: al quitar ruido desaparecen «bordes» falsos que el laplaciano contaba. Por eso conviene mirar varias medidas juntas y confirmar con el comparador.

6. Modelos de IA y algoritmos que usa la app

Una **red neuronal** es una función con millones de números ajustables (**parámetros** o «pesos»). Se **entrena** enseñándole miles de pares «imagen estropeada → imagen buena» hasta que aprende a transformar la primera en la segunda. Cuando luego ve una imagen nueva, **predice** el detalle que probablemente falta: por eso puede ser muy convincente, pero también equivocarse o inventar. El tamaño del fichero da una idea de su capacidad (y de lo que tarda).

6.1 Modelos de IA

Modelo	Tarea	Tamaño aprox.	Cómo funciona / comentarios
Real-ESRGAN animevideov3 (ncnn)	Escalado ×2-×4	≈ 2-3 MB	Red muy compacta pensada para animación: rápida, pero en película real tiende a «acuarela» (planos lisos).
Real-ESRGAN x4plus (ncnn / .pth)	Escalado ×4	≈ 64 MB (.pth), 16,7 M parámetros	Red RRDB profunda, entrenada con degradaciones realistas (desenfoque, ruido, JPEG). Más detalle, bastante más lenta.
realesr-general-x4v3 (.pth, spandrel)	Escalado ×4	≈ 5 MB, 1,2 M parámetros	Compacta y natural para imagen real. Buen equilibrio: la recomendada para Super 8.
RIFE 4.6 (ncnn)	Interpolación	≈ 10 MB	Estima el flujo intermedio entre dos fotogramas y los combina para crear el del medio. Muy rápida; falla con movimientos muy grandes u objetos que aparecen.
GFPGAN v1.4 (.pth)	Restaurar caras	≈ 350 MB	Usa un «generador de caras» (StyleGAN2) como conocimiento previo: reconstruye rasgos plausibles. Potente y peligroso para el parecido.
YuNet (ONNX)	Detectar caras	≈ 0,2 MB	Detector ligerísimo de OpenCV.

6.2 Algoritmos clásicos (sin IA)

La mayor parte de la restauración del máster **no** es IA, sino algoritmos matemáticos deterministas: dan siempre el mismo resultado, no inventan y se pueden explicar.

Algoritmo	Para qué	Idea
Lucas-Kanade + RANSAC	Estabilizar	Seguir puntos y quedarse con el movimiento de la mayoría.
Flujo óptico DIS	Polvo, ruido	Vector de movimiento por píxel para alinear vecinos.
Histogramas HSV	Cortes de escena	Un salto brusco en la distribución de colores = plano nuevo.
Inpainting de Telea	Rayas	Rellenar un hueco propagando los bordes hacia dentro.
Promedio temporal compensado	Ruido	Mezclar vecinos alineados: lo aleatorio se cancela.

Algoritmo	Para qué	Idea
NL-means	Ruido espacial	Promediar trozos parecidos de la misma imagen.
Máscara de enfoque adaptativa	Nitidez	Sumar la diferencia con una versión borrosa, solo en bordes.
Tabla LUT de niveles y ganancias	Color	Reasignar cada valor de color según una curva.
Lanczos	Escalado clásico	Interpolación con una función de ventana: nítida y honesta.
bwdif	Desentrelazar	Reconstruir líneas que faltan con información vecina.
FFT	Medir detalle	Descomponer la imagen en frecuencias.
Immerkær	Medir ruido	Estimar σ con un filtro que anula la señal suave.

6.3 Motores de ejecución

- **PyTorch 2.7 con CUDA 11.8:** ejecuta los modelos .pth en la GPU. La app lo prueba en un proceso aparte (así un fallo no tumba el servidor) y, si la GPU deja de responder, sigue por CPU sin detener el trabajo.
- **spandrel:** biblioteca que reconoce automáticamente la arquitectura de un .pth (ESRGAN, SwinIR, HAT, DAT...). Permite dejar cualquier modelo en herramientas\modelos. Procesa por **teselas** (trozos) para no agotar la VRAM.
- **ncnn + Vulkan:** RIFE y Real-ESRGAN como programas externos. Trabajan por lotes de imágenes PNG en disco (troceados en bloques de unos 2 GB).
- **FFmpeg:** lectura y escritura de vídeo, FFV1, NVENC, filtros de audio y proxies.

7. ¿Mejor GPU y mejores modelos, mejor resultado?

Hay que separar dos preguntas, porque tienen respuestas distintas: **¿más rápido?** y **¿mejor?**

7.1 Dónde se va el tiempo

En tu ejecución de más de 3 horas, casi todo el tiempo se lo llevaron la IA (escalado $\times 4$ con un modelo pesado y RIFE a 60 fps) y el trasiego de miles de PNG por el disco D: (que no es SSD). Hay que tener en cuenta que interpolar a 60 fps multiplica por más de tres los fotogramas que luego hay que escalar, y que escalar $\times 4$ multiplica por 16 los píxeles. Las etapas clásicas del máster pesan mucho menos.

7.2 Una GPU más potente: más velocidad, misma imagen

Con el mismo modelo, una GPU mejor produce **exactamente la misma imagen**, solo que antes. La velocidad en IA depende sobre todo de los núcleos (y sus unidades tensoriales), del ancho de banda de memoria y de la VRAM (que decide qué modelos caben).

GPU	Núcleos CUDA	VRAM	Ancho de banda	Velocidad IA orientativa*
RTX 3060 (la tuya)	3 584	12 GB GDDR6	360 GB/s	1× (referencia)
RTX 5060 Ti 16 GB	4 608	16 GB GDDR7	448 GB/s	$\approx 1,5\times$
RTX 5070	6 144	12 GB GDDR7	672 GB/s	$\approx 2\times$
RTX 5070 Ti	8 960	16 GB GDDR7	896 GB/s	$\approx 2,5\times$
RTX 5080	10 752	16 GB GDDR7	960 GB/s	$\approx 3\times$
RTX 5090	21 760	32 GB GDDR7	1 792 GB/s	$\approx 4-5\times$

* Estimaciones propias a partir de las especificaciones, no mediciones: la mejora real varía según el modelo. Las RTX 50 necesitan un controlador reciente y PyTorch con CUDA 12.8 (el instalador de la app ya lo elige solo). En la tarea que tardó 3 horas, solo la parte de IA se acelera así; el trasiego de disco no depende de la GPU.

Lo más rentable primero

Antes de cambiar de tarjeta hay mejoras gratuitas: (1) actualizar el controlador NVIDIA (permitiría PyTorch con CUDA 12.x, algo más rápido); (2) poner la carpeta de trabajo en el SSD C: si hay espacio; (3) el «punto 3» pendiente: procesar por lotes en la GPU sin escribir PNG intermedios; (4) no pedir $\times 4$ ni 60 fps cuando el corte espectral no lo justifica, que es justo lo que hace el plan automático.

7.3 Modelos mejores: aquí sí cambia la imagen

Los modelos de la app trabajan **fotograma a fotograma** (excepto RIFE). Eso tiene un límite: cada fotograma se «adivina» por separado, así que el detalle inventado puede cambiar de uno a otro y *hervir*. Los modelos más modernos atacan ese problema:

Familia	Ejemplos	Qué aportan	Qué piden
Superresolución de vídeo (temporal)	BasicVSR++, RealBasicVSR, VRT	Usan varios fotogramas a la vez: recuperan detalle real que está repartido en el tiempo y dan un resultado estable.	8–16 GB de VRAM; más lentos que Real-ESRGAN.
Transformers de imagen	SwinIR, HAT, DAT (vía spandrel)	Algo más de detalle y menos artefactos que ESRGAN.	Mucho más lentos; caben en 12 GB con teselas. Ya compatibles con la app.
Difusión para vídeo	SeedVR2 (3B y 7B parámetros), STAR	El salto de calidad más visible hoy: reconstruyen texturas creíbles y coherentes en el tiempo.	3B: 12 GB justos con técnicas de ahorro y muy lento; 7B: 24–32 GB. Inventan más.
Interpolación reciente	RIFE 4.2x, GIMM-VFI, FILM	Mejor en movimientos grandes y oclusiones.	Las más precisas son bastante más lentas.
Comerciales	Topaz Video AI	Modelos específicos para película antigua, interfaz sencilla.	Licencia de pago.

7.4 Veredicto

1. **El techo lo pone el escaneado.** Un reescaneado a 2K–4K fotograma a fotograma (con escáner de película, no filmando la pared) mejora más que cualquier GPU o modelo, porque aporta detalle *real*.
2. **Después, los modelos temporales o de difusión.** Con tu RTX 3060 se pueden probar modelos de vídeo medianos; SeedVR2 7B necesita una tarjeta de 24–32 GB (p. ej. RTX 5090).
3. **Por último, la velocidad.** Una GPU nueva hace que el modo «Máximo» pase de horas a minutos, pero con los mismos modelos la imagen será idéntica.

Y una advertencia: cuanto más potente es el modelo generativo, más «inventa». Para un recuerdo familiar puede ser aceptable; para un documento histórico, el máster fiel debe seguir siendo la referencia.

8. Lo que hemos aprendido en este proyecto

Construir la app ha sido también una lista de errores corregidos. Estas son las lecciones más útiles, técnicas y de método:

Lección	Qué pasó y qué cambió
Medir antes de suponer	Suponer 18 fps en un vídeo de 28,268 fps habría ralentizado la acción. Ahora la cadencia solo cambia si se <i>deduce</i> de los datos.
La GPU que «está» no siempre «funciona»	<code>torch.cuda.is_available()</code> decía que sí y luego fallaba. La prueba fiable es reservar memoria de verdad, en un proceso aparte.
Versiones que deben encajar	PyTorch con CUDA 12.6 no funciona con un controlador de CUDA 12.0. El instalador elige ahora según el controlador.
Los mensajes engañan	nvidia-smi lista programas gráficos (tipo C+G) que no bloquean nada; el aviso «GPU ocupada» asustaba sin motivo y se sustituyó por una prueba real.
El disco importa	C: se llenó a mitad de instalar PyTorch. Cachés y temporales van ahora al disco de la app, y se avisa si falta espacio antes de empezar.
Falsos positivos	El primer detector de polvo confundía parpadeo y grano con motas. Igualar la ganancia local y exigir una anomalía local lo resolvió.
Poner límites a los algoritmos	Una estimación de movimiento errónea desplazó la imagen de forma catastrófica; ahora las correcciones están acotadas.
Más píxeles no es más detalle	Un 4K del que no se nota nada llevó al corte espectral y a limitar la escala a lo que la película contiene.
Cronometrar bien	El primer plan proponía 336 minutos: medía el arranque del modelo como si fuera trabajo. Se descarta el calentamiento y se resta el tiempo de arranque.
Medir con la vara adecuada	El temblor medido en la imagen salía falseado por los bordes fijos; se calcula ahora desde la trayectoria.
Compatibilidad del navegador	El comparador mostraba negro en el original porque el navegador no lee MPEG-4 parte 2: de ahí los proxies H.264.
Separar lo fiel de lo inventado	Máster y copia mejorada en ficheros distintos: se puede disfrutar de la IA sin perder el documento.
Automatizar, pero explicando	El usuario no debe tocar 40 controles: el plan decide, aplica y explica por qué.
Verificar cada cambio	Algunas sustituciones de código fallaban en silencio; se comprueban ahora con búsquedas y pruebas antes de dar nada por hecho.

9. Recomendaciones y próximos pasos

1. Usar siempre «**Analizar y proponer parámetros**» y una **muestra** antes de la bobina entera.
2. Actualizar el controlador NVIDIA desde nvidia.com (lo tienes que hacer tú: la app no toca el sistema). Después, volver a ejecutar EJECUTAR.bat para que el instalador aproveche CUDA 12.x.
3. Si hay espacio, llevar la carpeta de trabajo al SSD (C:).
4. Guardar siempre el **máster FFV1**: es el original restaurado, del que se pueden sacar nuevas copias cuando haya mejores modelos.
5. **Punto 3 pendiente**: escalado e interpolación por lotes en la GPU sin PNG intermedios (ahorro estimado del 20-40 % en etapas de IA).
6. Otras mejoras posibles: procesar en 16 bits, evitar el doble redimensionado del lienzo 16:9, caché por bloques para reanudar trabajos, y probar un modelo temporal (BasicVSR++) o SeedVR2 3B.
7. Si alguna bobina es especialmente valiosa, valorar un **reescaneado profesional** a 2K o 4K.

10. Glosario

Término	Significado
Artefacto	Defecto creado por el propio procesado (halos, bloques, aspecto de acuarela).
Bobina	Rollo de película; una bobina Super 8 de 15 m dura unos 3-4 min.
Cadencia	Fotogramas por segundo.
CRF / CQ	Parámetro de calidad constante del codificador; menor = mejor.
Difusión	Tipo de IA generativa que parte de ruido y lo va convirtiendo en imagen.
ETA	Tiempo estimado hasta terminar.
Flujo óptico	Mapa de hacia dónde se mueve cada píxel entre dos fotogramas.
Inpainting	Rellenar una zona dañada a partir de lo que la rodea.
Interpolación	Crear fotogramas intermedios para aumentar la fluidez.
Máster	Versión restaurada de referencia, sin pérdida.
Proxy	Copia ligera de un vídeo, solo para visualizarlo.
Superresolución	Aumentar la resolución estimando detalle.
Telecine	Proceso de pasar película a vídeo.
Tesela (tile)	Trozo de imagen que se procesa por separado para que quepa en la VRAM.
VRAM	Memoria de la tarjeta gráfica.

11. Fuentes

Especificaciones de GPU y datos de los modelos generativos y de interpolación consultados en septiembre de 2026. El resto procede del código y las pruebas de la propia app.

- Wikipedia — GeForce RTX 50 series (especificaciones):
en.wikipedia.org/wiki/GeForce_RTX_50_series
- NVIDIA — GeForce RTX 50 Series: nvidia.com/es-es/geforce/graphics-cards/50-series/
- ComfyUI — documentación de SeedVR2: docs.comfy.org
- AInVFX — SeedVR2 v2.5 (requisitos y rendimiento): ainvfx.com
- seedvr2.net — guías de uso y memoria de vídeo
- AIVFI — clasificación de modelos de interpolación (GitHub)
- RunComfy — nodo RIFE VFI: runcomfy.com
- Real-ESRGAN (xinntao, GitHub) · GFPGAN (TencentARC, GitHub) · RIFE (hzwer, GitHub) · spandrel (chaiNNer-org, GitHub) · OpenCV (YuNet, DIS, Telea)