

Projetos gratuitos no GitHub para efeitos CRT em vídeos e imagens

Relatório curto e direto para escolher ferramentas, plugins, bibliotecas e coleções de shaders que realmente servem para simular CRT — seja em vídeo, imagem estática, stream, desktop ou engine.

Leitura honesta: nem todo projeto aqui exporta MP4/PNG sozinho. Alguns são **hosts** de shader, outros são **bibliotecas/presets**, e alguns são **ferramentas de export**. Misturar as categorias é a forma mais rápida de perder tempo.

Atalho por caso de uso
Quero testar CRT em imagens e vídeos sem ficar preso a engine glsViewer
Quero gravar vídeo com shader CRT por cima obs-shaderfilter
Quero aplicar CRT a qualquer janela no Windows ShaderGlass
Quero presets CRT prontos para alimentar outras ferramentas libretro/slang-shaders + crt-royale-reshade + gizmo-crt-shader
Quero usar CRT dentro de um projeto Unity RetroTVFX ou Simple-CRT-Shader
Quero exportar um shader experimental para PNG/GIF/MP4 Shadertoy Exporter

Tabela comparativa

Projeto	Tipo	Melhor uso	Licença
glsViewer	Ferramenta CLI / sandbox GLSL	Aplicar e testar shaders CRT em imagens, vídeos, GIFs e câmera; render headless; exportar imagem e sequência PNG.	BSD-3-Clause
ShaderGlass	Overlay desktop (Windows)	Aplicar CRT em qualquer janela, vídeo, emulador ou editor de pixel art no Windows.	GPL-3.0
obs-shaderfilter	Plugin do OBS	Aplicar shader CRT em fontes de vídeo ou imagem dentro do OBS e gravar/exportar o resultado.	GPL-2.0
Radiance	Software de video art / VJ	Cadeias de efeitos em vídeo e imagem com edição ao vivo.	MIT/X11
libretro/slang-shaders	Coleção de shaders	Base de presets CRT para usar em RetroArch, ShaderGlass e portas relacionadas.	Coleção multi-autor; verifique shader/arquivo específico
crt-royale-reshade	Port do CRT-Royale para ReShade	CRT de alta fidelidade em apps/games e players compatíveis com ReShade.	GPL-2.0
Simple-CRT-Shader	Shader Unity	Projetos Unity com vibe de TV antiga e pós-processamento leve.	MIT

Projeto	Tipo	Melhor uso	Licença
<u>RetroTVFX</u>	Pacote Unity	Simular RF, Composite, S-Video, Component e VGA/SCART em tempo real.	Há arquivo LICENSE; confirme os termos no repositório
<u>gizmo-crt-shader</u>	Preset/lib de shader CRT	Quem quer um CRT com scanlines menos exageradas e bom suporte a scaling fracionário.	Verificar no repositório
<u>Shadertoy Exporter</u>	Exportador de shaders	Gerar PNG/GIF/MP4 a partir de shaders do Shadertoy, inclusive CRTs encontrados lá.	MIT

Projetos que valem mais a pena

Abaixo está a leitura prática: o que cada projeto faz, quando usar e onde ele te atrapalha.

1. [glsViewer](#)

Tipo: Ferramenta CLI / sandbox GLSL

Melhor uso: Aplicar e testar shaders CRT em imagens, vídeos, GIFs e câmera; render headless; exportar imagem e sequência PNG.

Licença: BSD-3-Clause

Por que entrar na lista: É a opção mais versátil para pipeline técnico. Aceita texturas PNG/JPG/GIF/MP4/MOV, câmera e RTSP; suporta headless rendering, image export e PNG sequence export.

Limitação real: Exige mais mão técnica e linha de comando; não é o caminho mais amigável para quem quer só clicar.

Link direto: <https://github.com/patriciogonzalezvivo/glsViewer>

2. [ShaderGlass](#)

Tipo: Overlay desktop (Windows)

Melhor uso: Aplicar CRT em qualquer janela, vídeo, emulador ou editor de pixel art no Windows.

Licença: GPL-3.0

Por que entrar na lista: Já vem com biblioteca RetroArch embutida com 1200+ shaders e inclui simulação de CRT, VHS, upscale e filtros de TV.

Limitação real: É overlay em tempo real; não é a melhor escolha se sua meta principal for render offline de arquivo final.

Link direto: <https://github.com/mausimus/ShaderGlass>

3. [obs-shaderfilter](#)

Tipo: Plugin do OBS

Melhor uso: Aplicar shader CRT em fontes de vídeo ou imagem dentro do OBS e gravar/exportar o resultado.

Licença: GPL-2.0

Por que entrar na lista: Boa rota prática para vídeo: OBS aceita imagens, vídeos, captura de tela e câmera, e o plugin deixa você aplicar shader arbitrário por fonte.

Limitação real: Depende do fluxo do OBS; não é uma biblioteca reutilizável no seu site ou app.

Link direto: <https://github.com/exeldro/obs-shaderfilter>

4. [Radiance](#)

Tipo: Software de video art / VJ

Melhor uso: Cadeias de efeitos em vídeo e imagem com edição ao vivo.

Licença: MIT/X11

Por que entrar na lista: Trabalha com clipes, imagens, webcam, live streams e YouTube; traz 150+ efeitos em WGSL e integração com MPV.

Limitação real: Mais voltado a performance ao vivo do que a pós-produção convencional.

Link direto: <https://github.com/zbanks/radiance>

5. [libretro/slang-shaders](#)

Tipo: Coleção de shaders

Melhor uso: Base de presets CRT para usar em RetroArch, ShaderGlass e portas relacionadas.

Licença: Coleção multi-autor; verifique shader/arquivo específico

Por que entrar na lista: Tem pasta dedicada a CRT e um ecossistema enorme de presets/passes. Ótimo como 'biblioteca de matéria-prima'.

Limitação real: Não é app final por si só. Você precisa de um host que carregue shaders slang.

Link direto: <https://github.com/libretro/slang-shaders>

6. [crt-royale-reshade](#)

Tipo: Port do CRT-Royale para ReShade

Melhor uso: CRT de alta fidelidade em apps/games e players compatíveis com ReShade.

Licença: GPL-2.0

Por que entrar na lista: Porta um dos CRTs mais respeitados do ecossistema libretro para ReShade 4.9+.

Limitação real: É pesado e mais chato de acertar; não é a opção 'rápida' para prototipagem.

Link direto: <https://github.com/akgunter/crt-royale-reshade>

7. [Simple-CRT-Shader](#)

Tipo: Shader Unity

Melhor uso: Projetos Unity com vibe de TV antiga e pós-processamento leve.

Licença: MIT

Por que entrar na lista: Inclui white noise, screen jump, scanline, monochrome, flicker, slippage e chromatic aberration.

Limitação real: Só pipeline Built-in do Unity; URP e HDRP não são suportados.

Link direto: <https://github.com/yunoda-3DCG/Simple-CRT-Shader>

8. [RetroTVFX](#)

Tipo: Pacote Unity

Melhor uso: Simular RF, Composite, S-Video, Component e VGA/SCART em tempo real.

Licença: Há arquivo LICENSE; confirme os termos no repositório

Por que entrar na lista: Foca na estética de sinal analógico, não só em scanlines. Bom para vídeo/sprites/cenas em Unity.

Limitação real: Útil em engine, não como ferramenta universal de export.

Link direto: <https://github.com/GlaireDaggers/RetroTVFX>

9. [gizmo-crt-shader](#)

Tipo: Preset/lib de shader CRT

Melhor uso: Quem quer um CRT com scanlines menos exageradas e bom suporte a scaling fracionário.

Licença: Verificar no repositório

Por que entrar na lista: Foi feito para mimetizar comportamento CRT sem depender de scanline pesada; tem variantes para SNES, Mega Drive e N64.

Limitação real: Mais um preset/conjunto de shader do que uma ferramenta final.

Link direto: <https://github.com/gizmo98/gizmo-crt-shader>

10. [Shadertoy Exporter](#)

Tipo: Exportador de shaders

Melhor uso: Gerar PNG/GIF/MP4 a partir de shaders do Shadertoy, inclusive CRTs encontrados lá.

Licença: MIT

Por que entrar na lista: Salva frames em PNG e gera GIF/MP4 via FFmpeg; excelente para transformar protótipo de shader em mídia exportável.

Limitação real: Depende de você ter ou adaptar um shader CRT no Shadertoy.

Link direto: <https://github.com/KoltesDigital/shadertoy-exporter>

Como montar um fluxo que presta

- Para **imagens estáticas**, o caminho mais racional é: **glsViewer** + um shader CRT do **libretro/slang-shaders** ou do **gizmo-crt-shader**. Isso te dá controle real e export limpo.
- Para **vídeo gravado**, o atalho mais prático é **OBS + obs-shaderfilter**. Você carrega o vídeo ou a captura como fonte, aplica o shader e grava/exporta.
- Para **uso geral em Windows**, **ShaderGlass** é o melhor jeito de experimentar CRT em players, emuladores, navegadores e até editores sem montar pipeline.
- Para **Unity**, escolha **RetroTVFX** se você quer simular cadeia de sinal analógico (RF/composite/S-video), e **Simple-CRT-Shader** se quer um pacote menor e direto.
- Para **prototipagem de shader** que depois vira mídia, use **Shadertoy Exporter**. Ele não cria o CRT do zero; ele exporta o que você já tem em shader.
- Para **fidelidade máxima**, teste **crt-royale-reshade**, mas não entre nisso achando que será rápido. É bonito, mas mais pesado e mais chato de calibrar.

Dicas que evitam perda de tempo

- **Não confunda host com shader**. ShaderGlass, OBS e glsViewer são hosts. libretro/slang-shaders, crt-royale-reshade e gizmo-crt-shader são a matéria-prima do visual.
- **Nem todo CRT fica bom em ANSI art**. Para arte ANSI ou texto, exagerar scanline, curvatura e separação RGB destrói legibilidade. Prefira presets mais leves.
- **Se o objetivo é export final, confirme isso antes**. Overlay ao vivo é divertido, mas não resolve render offline de produção.
- **Licença importa**. Repositórios grandes de presets/shaders às vezes juntam trabalho multi-autor. Se for embutir em produto ou distribuir junto, confira o arquivo de licença de cada shader relevante.
- **Comece simples**. Primeiro acerte scanlines, máscara e leve ruído. Curvatura forte e aberração cromática pesada quase sempre deixam o material mais barato, não mais convincente.

Fontes consultadas

GitHub READMEs e páginas dos repositórios pesquisados em 18 mar 2026.

Este documento foi montado como curadoria técnica, não como dump bruto de busca.