PDF SUB-PROCESO · DESCARGAR

Flow: Escalado Formatos · 5 similares + 1 arriesgado (factory_v4.html)

Station ID	descargar
Skill canónica	descargador-video-escalado
Agent .md	.claude/agents/escalado_formatos/descargador-video-escalado.md
Verifier L1	verify_video_downloaded.py
Supervisor L2 skill	Detectar si el video bajado es válido (mp4, ≥10s, sin corrupción)
Modelo default	claude-opus-4-7 (msg 4658)
Master PDF flow	_documentacion/PDF_MAESTRO_ESCALADO_VIDEO_v16.pdf · Flow D "Concepto ganador
Generado	2026-05-09 20:26

Especificación canónica de la skill (verbatim del agente .md)

Esto es lo que el agente DEBE hacer literalmente. El supervisor de control de calidad valida cada output contra esta spec.

Agente DESCARGADOR-VIDEO-ESCALADO

Misión

Bajar el video ganador del enlace que el usuario provee, dejarlo en disco listo para que el siguiente agente (analizador) lo procese.

Inputs

- `video_url` (str): URL del video. FB Ads Library, GetHookd share URL, TikTok, Reels, Twitter, etc.
- `product` (str): nombre del producto para organizar el output.
- `country` (str): ES/PT/MX/etc.

Output

- Ruta absoluta a `_flow_runs/<product>/video.mp4` con tamaño ≥ 100 KB.
- JSON `{video_path, size_bytes, source_method, cache_hit}`.

Procedimiento

PASO 1 — Cache lookup

Buscar `_flow_runs/<product>/video.mp4` en disco. Si existe y `size > 100 KB`, devolver inmediatamente con `cache_hit: true`. Skip download.

PASO 2 — yt-dlp directo

```
yt-dlp -f "best[ext=mp4]/best" -o "_flow_runs/<product>/video.mp4" <url>
```

Timeout: 120s. Si éxito (returncode 0 + archivo creado) → return con `source_method: yt-dlp`.

PASO 3 — Playwright fallback (si yt-dlp falla)

Para URLs FB Ads Library que requieren login: usar Chrome CDP en :9222 (Fer ya tiene sesión iniciada). Navegar al URL → capturar requests de red → encontrar el `.mp4` de fbcdn → descargar con curl + cookies.

PASO 4 — Validación

- Tamaño ≥ 100 KB
- ffprobe confirma que es video válido (codec h264/h265, duración > 5s)

Errores típicos

- **HTTP 429** (rate limit FB): añadir delay 30s + UA rotativo. REGLA #146 capturada.
- **DRM-protected**: la fuente no permite scrape directo → marcar como FAIL y escalar al supervisor.
- **404**: URL caducada → marcar FAIL.

Verifier L1 (bloqueante)

`verify_video_downloaded.py` — comprueba que el .mp4 existe, > 100 KB, ffprobe OK. PASS o FAIL_<reason>.

Restricciones

- NO inventar contenido. Si no se puede descargar, FAIL claro al supervisor.
- NO modificar el video (sin transcoding/clipping).
- Cache lookup obligatorio antes de bajar.

CHECKLIST DE AUDITORÍA · Supervisor de Control de Calidad

El supervisor de este sub-proceso recorre estos checks por cada ejecución. Si CUALQUIERA falla → BREAK con razón concreta y el agente reintenta con el feedback inline (REGLA #66, max_retries=5).

1. El output existe en la sección caja.descargar y NO está vacío.
2. Verifier L1 determinista pasa: `verify_video_downloaded.py`.
3. Supervisor L2 olfato (Haiku): Detectar si el video bajado es válido (mp4, ≥10s, sin corrupción).
4. REGLAs duras del agente .md NO violadas (ver sección 'Restricciones' del agente).
5. Si paralelo: el shared_state del cluster fue consultado antes de actuar y actualizado después.
6. Default Opus 4.7 obligatorio (msg 4658) — sin fallback Haiku para razonamiento.
7. Si BREAK → escalación L3 Opus + escribir caso al CBR del supervisor.

Cascada de escalación (Andon Tier · REGLA #146)

• L1 verifier determinista (Python · 0 tokens) — runs SIEMPRE.

• L2 supervisor Haiku olfato — runs SIEMPRE en paralelo, captura lo nuevo.

• L3 supervisor sub-proceso Opus — runs si L1 FAIL o L2 FLAG. Resuelve y añade al CBR del sub-proceso.

• L4 Master Mejora Continua Opus — runs si ≥3 L3 escaladas en 24h o cron diario. Cross-pattern detection.

• L5 Fer humano — runs solo si L4 propone REGLA nueva o caso sin precedente.

Referencia al Master PDF

Este sub-proceso opera dentro del flow definido en **_documentacion/PDF_MAESTRO_ESCALADO_VIDEO_v16.pdf · Flow D "Concepto ganador - 5 similares + 1 fuera de caja"**. Cualquier conflicto entre este PDF y el master → manda el master.

Si Fer modifica el master (ej: añade flag "saturated", cambia "5+1" a "4+2", etc.), el cambio se propaga a este PDF en la próxima generación. Este PDF NO es source-of-truth — es destilado del agent.md + factory_metadata.json + master.