

PDF SUB-PROCESO - ANALIZAR

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

| | |
|---------------------|--|
| Station ID | analizar |
| Skill canónica | analizador-combo-completo |
| Agent .md | .claude/agents/escalado_formatos/analizador-combo-completo.md |
| Verifier L1 | verify_analysis_report.py (tiene secciones: visual + speech + síntesis - y campo cache_hit bool |
| Supervisor L2 skill | Detectar si el análisis es superficial (≥ 1500 palabras + ≥ 10 timestamps clave) - y validar que el |
| 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 ANALIZADOR-COMBO-COMPLETO

Misión

Extraer el COMBO COMPLETO del video ganador (todo lo que necesita el siguiente agente para diseñar la estrategia de búsqueda de videos similares escalables).

Inputs

- video_path (str): ruta absoluta al .mp4 descargado.
- product (str): nombre del producto.
- country (str): país target (ES/PT/MX/...).
- saturated (bool): flag input del usuario.

Output

JSON_analysis.json con campos OBLIGATORIOS:

```
{
  "timestamp": "2026-05-09T16:50:00Z",
  "product": "...",
  "video_url": "...",
  "country": "...",
  "duration_seconds": 138,
  "transcription": "<full text>",
  "word_count": 1842,
  "language_detected": "es",
  "combo_completo": {
    "awareness": "Problem Aware (Schwartz 2)",
    "sofisticacion": "2 (mecanismo definido pero no marca)",
    "jergas": ["dolor cervical", "migraña", "almohada", "C1-C7", "apnea posicional"],
    "deseo_principal": "dormir bien y despertarse sin dolor de cabeza",
```

```

"avatar": {
  "edad": "45-65",
  "genero": "ambos · ligeramente femenino",
  "contexto": "trabajadores con dolor crónico cervical · padres",
  "profesion": "indistinta – el dolor es transversal"
},
"temporalidad": "urgencia media · dolor diario crónico · sin estacionalidad",
"fuerza_cambio": "frustración acumulada (años con migrañas + médicos sin solución) + amenaza médica",
"antagonistas_nombrados": ["fisio", "nolotil", "almohadas tradicionales", "CPAP", "cirugía maxilofacial"],
"formato_angulo": "advertorial 3D Pixar antropomorfo · partes del cuerpo hablando",
"mecanismo_revelado": "calibración clínica C1-C7 · vértebras en posición neutra toda la noche",
"estructura_narrativa": "5 actos · cuello → hombros → brazos → vértebras → médico/solución",
"duracion_clase": "medio (2-3min) · advertorial antropomorfo"
},
"cache_hit": false
}

```

Procedimiento

PASO 1 — Cache lookup OBLIGATORIO

Buscar análisis previo en (en este orden):

1. `_productos/<product>/_analysis.json`
2. `_productos/<product_lowercased>/_analysis.json`
3. `_apify_out/<product>.json`
4. `memory/feedback_<product>.md`

Si existe y `(now - timestamp).days < 30` y `country` coincide → devolver el análisis cacheado con `cache_hit: true`. **No invocar Whisper**. Ahorro masivo de tokens y tiempo.

PASO 2 — Whisper transcripción

```

from faster_whisper import WhisperModel
model = WhisperModel('medium', device='cpu', compute_type='int8')
segments, info = model.transcribe(video_path, language=country.lower()[:2], beam_size=5)

```

PASO 3 — Frame extraction

ffmpeg cada 2-3s para obtener ~30-60 frames clave.

```
ffmpeg -i video.mp4 -vf "fps=1/3" frame_%03d.jpg
```

PASO 4 — LLM analysis (Claude Opus)

Pasar al modelo: `transcription + frames + product context`. Pedir extracción de TODOS los campos del `combo_completo` en JSON estructurado.

Prompt anchoring:

> "Analiza este video ganador para el producto X en el mercado Y. Extrae el combo completo necesario para encontrar videos escalables similares. Estructura: awareness (Schwartz 1-5), sofisticación (1-5), jergas usadas, deseo principal, avatar detallado, temporalidad (urgencia + estacionalidad + etapa de vida), fuerza de cambio (qué empuja al avatar a actuar AHORA), antagonistas explícitamente nombrados en el speech, formato y ángulo, mecanismo revelado, estructura narrativa, clase de duración. JSON estricto."

PASO 5 — Guardar para futuros lookups

Guardar `_productos/<product>/_analysis.json`. La próxima ejecución del flow reutilizará este análisis (cache hit).

Verifier L1

`verify_combo_completo.py` — valida que TODOS los campos de `combo_completo` están presentes y no vacíos. PASS o FAIL_`<missing_fields>`.

Restricciones

- PROHIBIDO inventar campos sin evidencia en transcripción/frames.
- Si Whisper devuelve `<100` palabras, FAIL (audio corrupto o sin speech).
- Cache lookup OBLIGATORIO — el agente FAIL si no lo intentó.
- Modelo Opus 4.7 obligatorio para LLM analysis (REGLA: en duda, modelo más alto).

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.analizar` y NO está vacío.
2. Verifier L1 determinista pasa: `verify_analysis_report.py` (tiene secciones: `visual` + `speech` + `síntesis` · y campo `cache_hit` boolean).
3. Supervisor L2 olfato (Haiku): Detectar si el análisis es superficial (≥ 1500 palabras + ≥ 10 timestamps clave) · y validar que el cache lookup se ejecutó antes que el análisis pesado.
4. Cache lookup ejecutado ANTES del trabajo pesado (REGLA cache, ahorro tokens).
5. REGLAs duras del agente `.md` NO violadas (ver sección 'Restricciones' del agente).
6. Si paralelo: el `shared_state` del cluster fue consultado antes de actuar y actualizado después.
7. Default Opus 4.7 obligatorio (msg 4658) — sin fallback Haiku para razonamiento.
8. 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.