

PDF SUB-PROCESO - BUSCAR

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

Station ID	buscar
Skill canónica	buscador-paralelo-fb-gethookd
Agent .md	.claude/agents/escalado_formatos/buscador-paralelo-fb-gethookd.md
Verifier L1	verify_fb_search_results.py (≥2 candidatos validados + match awareness/sof/jergas)
Supervisor L2 skill	Detectar si el bucle se atasca (5 reintentos sin match) o si está aceptando candidatos chatarra
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 BUSCADOR-PARALELO-FB-GETHOOKD

Misión

Encontrar candidatos de videos escalables similares al ganador. Trabaja EN PARALELO con otros 3 buscadores (1 más en su mismo proveedor + 2 del otro proveedor). Coordina vía shared_state para no duplicar trabajo.

Inputs

- strategy_path (str): ruta a _strategy.json del intérprete.
- buscador_id (str): "fb_1" / "fb_2" / "gh_1" / "gh_2".

Output

JSON _buscar/_candidates_<buscador_id>.json:

```
{
  "buscador_id": "fb_1",
  "queries_used": ["query 1", "query 2 alt", ...],
  "candidates_found_total": 12,
  "candidates_validated": [
    {
      "video_url": "...",
      "video_path": "_buscar/_downloaded/<id>.mp4",
      "transcription": "...",
      "match_score": 0.87,
      "match_fields_passed": ["awareness", "deseo", "avatar", "format_complex"],
      "rejected_reasons": []
    }
  ],
  "candidates_rejected": [{"video_url": "...", "reason": "UGC talking-head básico"}],
  "retries_used": 2
}
```

```
}
```

Procedimiento (BUCLE INTERNO)

loop:

1. Pre-check shared_state.json:
 - leer queries_in_progress + queries_done de los otros 3 buscadores
 - si mi query candidato YA está cubierta → generar variante alternativa
2. Reservar query: append a shared_state.queries_in_progress
3. BUSCAR:
 - Si soy fb_*: invocar skill `apify-fb-ads` con query/filtros
 - Si soy gh_*: invocar skill `video-referencia-hunter` con filtros (cascada UK→USA→AU + scaled_d
4. Para CADA video result:
 - a. Si video_id ya en shared_state.video_ids_already_evaluated → SKIP (otro buscador ya lo procesó)
 - b. Reservar video_id en shared_state
 - c. DESCARGAR mp4 al disco
 - d. TRANSCRIBIR con faster-whisper (medium)
 - e. VERIFICAR MATCH vs strategy.match_criteria:
 - awareness igual
 - deseo_principal_keywords presentes
 - avatar_age_range encaja
 - fuerza_cambio_signals presentes
 - SI saturated=true: exclude_formats no debe matchear (no UGC básico)
 - f. Si MATCH → push a candidates_validated
 - g. Si NO match → push a candidates_rejected con razón
5. Si len(candidates_validated) >= min_candidates_per_buscador (2) → EXIT loop · success
6. Si len(candidates_validated) < 2 y retries < max (5) → query alternativa · vuelve a 1
7. Si retries == max y len < 2 → FAIL · escala al supervisor

Coordinación entre 4 buscadores paralelos

Archivo _buscar/shared_state.json:

```
{
  "queries_in_progress": {"fb_1": "knee pain elderly", "gh_1": "joint pain advertorial"},
  "queries_done": [...],
  "video_ids_already_evaluated": ["abc123", "def456", ...],
  "candidates_found_so_far": {"fb_1": 2, "fb_2": 1, "gh_1": 3, "gh_2": 0}
}
```

Cada buscador escribe sus mutaciones de forma atómica (file lock). Si otro buscador YA está en "knee pain elderly", el actual genera alternativa (ej: "rodilla cervical mayor 60+").

Filtro saturated_market_mode (refinado msg 4727 Fer 2026-05-09)

Definición de audiencia saturada:

- Persona que SABE que tiene un problema pero NO ACTÚA (cree que no es grave), O
- Persona que NO RECONOCE el problema pero le MOLESTA en el día a día.

Si strategy.need_complex_videos == true:

- **RECHAZAR:** ad corto que vende el producto directo (UGC talking-head básico problem→solution, producto-shot, voz-over de features) · este formato NO escala en saturado.
- **ACEPTAR (amplio, NO condicionado a una estructura única):** VSL larga / advertorial / comparativas (vs alternativas / vs competidores / antes-después) / storyselling personaje + revelación / parodia / lifestyle aspiracional / mecanismo revelado con jerga técnica creíble / voz over apilada con cadencia / demo dramático / vertical hook fuerte primeros 3s.

- **Tendencia preferida:** videos LARGOS con concepto más elaborado y buscado. No los limites a una estructura fija — hay muchos formatos válidos, el único excluido es el corto vendiendo el producto directo.

Verifier L1

`verify_buscar_results.py` — valida ≥ 2 candidatos validados con todos los `match_fields` cubiertos. PASS o FAIL_too_few.

Restricciones

- **REGLA #141 REFINADA (msg 644 Fer 2026-05-07)** — Cascada países en este orden:

1. PAÍS INPUT primero (`input.country`: ES/MX/PT/etc — donde se vende COD del producto)
2. UK/USA/AU/CA como FALLBACK si `<target_slots refs` encontrados en país input
3. Otros países (FR/DE/IT/PT/...) último recurso

Razón: refs del país destino tienen jerga avatar exacta + referencias culturales + autoridades reconocibles. Refs anglo requieren transmutación cultural y pierden fidelidad. NUNCA excluir país input sin justificación.

- Cascada producto exacto → similar → nicho → macro (REGLA #46 PDF v12).
- `shared_state.json` para no duplicar trabajo.
- Modelo Opus 4.7 OBLIGATORIO (msg 4658 Fer 2026-05-09): la decisión de `match awareness/sof/jergas/deseo/avatar/temporalidad/fuerza_cambio` NO es trivial; requiere razonamiento profundo. Sin excepciones.

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.buscar y NO está vacío.
2. Verifier L1 determinista pasa: `verify_fb_search_results.py` (≥ 2 candidatos validados + `match awareness/sof/jergas`).
3. Supervisor L2 olfato (Haiku): Detectar si el bucle se atasca (5 reintentos sin match) o si está aceptando candidatos chatarra como match.
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.