

Agent Chaining y Buffer Social: De la Nota a LinkedIn con Fricción Cero

Estantería: DevSecOps e Infraestructura

Subtema: Gobernanza

mercedev.es — 2026-05-14 | Epic 3 - Fase 1

El Desafío (Síntoma)

La creación de contenido técnico y su posterior promoción social requerían excesiva intervención manual. Obligar a un único Small Language Model (SLM) local a actuar como ingeniero técnico y como copywriter de marketing simultáneamente provocaba alucinaciones, saturación de la ventana de contexto y pérdida de formato. Además, publicar inmediatamente saturaba las redes sociales.

La Maniobra (Lógica)

Se implementó el patrón de **Agent Chaining** (Encadenamiento de Agentes). El Agente Bibliotecario (`merci-librarian.py`) estructura el documento técnico y, al finalizar, invoca programáticamente al Agente Blogger (`merci-blogger.py`), pasándole el control y calculando matemáticamente la URL canónica de destino. El Blogger redacta un *teaser* comercial y lo sella en el YAML Frontmatter con el metadato `estado_social: "en_cola"` . Finalmente, un demonio en el sistema operativo (Cronjob) ejecuta `merci-linkedin.py --auto` cada 3 días para consumir este *Buffer Social* de forma asíncrona.

El Aprendizaje / Deuda Técnica

Separation of Concerns (Separación de Responsabilidades) aplicado a la Inteligencia Artificial. Los modelos locales rinden de manera espectacular cuando se les asigna una micro-tarea con un *System Prompt* hiperespecializado. Orquestar el flujo conectando agentes a través de scripts en Python y usar el YAML como una Máquina de Estados transforma un simple repositorio de código en una plataforma completa de *Developer Relations* (DevRel) 100% privada y a coste cero.