

Estrategias de Integración Headless: WP y WooCommerce

Compendio | Vol. 1

mercedev.es — 2026-05-07 | Fase 4

Estrategias de Integración Headless: WP y WooCommerce

El Desafío (Síntoma)

La integración de un CMS (Content Management System) pesado como WordPress (WP) y WooCommerce en un núcleo estático de alto rendimiento (Core Web Vitals 100/100) introdujo múltiples vectores de fallo. Los desafíos documentados incluyeron:

- **Ceguera de Proxy:** Varnish y Nginx ofuscaban el protocolo HTTPS y purgaban las cabeceras de autorización estándar, provocando errores de *Mixed Content* y bloqueos de la API REST (HTTP 401).
- **Deriva de Datos (Data Drift):** Diferencias de IDs entre el entorno de desarrollo local y producción generaban conflictos al sincronizar contenido.
- **Contaminación de Rendimiento:** WooCommerce inyectaba scripts en línea agresivos (Speculation Rules, fragmentos de carrito AJAX) que violaban la Política de Seguridad de Contenidos (CSP) y hundían el rendimiento móvil.

La Maniobra (Lógica)

Para domar el CMS, se desplegó una estrategia de defensa en profundidad:

1. **Bypass de Proxy y SSL:** Se inyectó una cabecera HTTP gemela (`X-Authorization`) desde el orquestador Python (`merci-wp.py`) para atravesar Varnish, restaurándola luego a nivel de PHP en el Child Theme. El *Mixed Content* se resolvió forzando URLs relativas al protocolo (`//`) y leyendo `home_url()` en lugar de las variables brutas de servidor (`$_SERVER`).

2. **SSOT Dinámico por Slug:** Se erradicó el uso de IDs rígidos. `merci-wp.py` ahora interroga a la API REST de destino usando el nombre del archivo (slug). Esto logra paridad Dev/Prod absoluta, permitiendo actualizar el mismo archivo local contra diferentes bases de datos sin colisiones.
3. **Escudo de Rendimiento y Kill-Switch:** Se implementaron bloqueos estrictos en `functions.php` (desencolado masivo y Whitelist Criptográfico CSP en Nginx). Adicionalmente, el orquestador Python aplica un "Kill-Switch" que expulsa los artículos despublicados (`estado: "borrador"`) de vuelta al directorio de incubación local, previniendo posts fantasmas.

El Aprendizaje / Deuda Técnica

Aislar un CMS por infraestructura (Proxy Inverso) es solo la mitad del trabajo; la otra mitad es aislarlo lógicamente. Confiar en un "Slug" como identificador universal (Single Source of Truth) desacopla el contenido del estado temporal de cualquier base de datos.

Cuadernillos de Referencia (Profundización táctica)

Para estudiar las batallas específicas libradas para domar el CMS, consulta:

- [Ceguera de Proxy y Bypass en Headless CMS](#)
- [Única Fuente de Verdad: Resolviendo la deriva de slugs entre SSG y CMS](#)
- [Inyección Headless en WooCommerce con Python Puro](#)
- [Domando a WooCommerce: Jerarquía de plantillas en temas custom](#)