EVIDENCIA · Obligación 4 · INSOR Conecta · Junio 2026 · Carlos Andrés Molina

Agente de Contenidos — INSOR Conecta

> Plugin `insor-agente-contenidos`: asistente guiado (wizard) para que los gestores creen, editen,
> publiquen y eliminen contenido del portal conversando, sin tocar el editor de WordPress.
> Construido en sesiones de **2026-06-07/08**. Versión actual: **0.5.3**.
>
> **Docs detalladas (viven con el plugin):**
> `wordpress/plugins/insor-agente-contenidos/ESTADO-Y-CONTINUACION.md` (historial completo por versión) y
> `AGENTE-CONTENIDOS.md` (spec). Este documento resume **qué se hizo** y, sobre todo, el **playbook de
> despliegue y aprendizajes** para futuras tareas en este servidor.

---

1. Qué hace (estado final)

Un **wizard por reglas, 100% determinístico** (sin IA, sin llaves externas, gratis e instantáneo) que
aparece como **botón flotante ✦ en todas las páginas del front** (solo para usuarios logueados con
`edit_posts`) y también en **WP-admin → "Agente de Contenidos"**.

Capacidades:

Elegir tipo de contenido (11 tipos, ver §4) → **Crear / Editar / Enviar a papelera / Cambiar tipo**.
Captura **paso a paso** con botones de respuesta rápida (selects, taxonomías, omitir/mantener).
Subir imagen / PDF / archivo desde el equipo (a la Biblioteca de Medios) o pegar URL.
Buscador en vivo al editar/eliminar: lista navegable con scroll infinito, miniatura + estado,
filtra mientras escribes.

Publicar con confirmación. Al **editar** se **preserva el estado** (lo publicado sigue publicado;
los borradores siguen borrador). Lo nuevo nace como **borrador**.

↺ Empezar de nuevo en cualquier paso (nada se guarda hasta pulsar "Guardar").
Bitácora de acciones (`option iac_log`).

> Historia: hubo una etapa con chat LLM (Gemini/Claude BYOK) y con *embeddings* (sugerir sección /
> detectar duplicados). Ambos se **retiraron** por decisión del cliente — el wizard guiado basta y evita
> costos/dependencias. El código LLM quedó **dormido** (no se usa); los embeddings se eliminaron.

---

2. Arquitectura

Todo es **dirigido por esquema** — agregar o cambiar un tipo de contenido NO requiere tocar el motor.

Archivo	Rol
`insor-agente-contenidos.php`	Menú admin, widget en front (`wp_footer`), enqueue, AJAX: `iac_chat` (turnos del wizard), `iac_upload` (subir archivo), `iac_search` (buscador paginado).

`includes/schemas.php`	**`iac_schema($type)`**: define los campos de cada tipo y su **destino** (`title`/`content`/`acf`/`meta`/`thumbnail`/`taxonomy`). `iac_schema_types()` y `iac_managed_types()`.
`includes/wizard.php`	**`iac_wizard($state,$input)`**: máquina de estados (type→intent→field/pick→confirm→saved/delconfirm). Estado viaja cliente⇔servidor (servidor sin sesión).
`includes/tools.php`	Persistencia: `iac_tool_guardar` (informe, histórico), **`iac_wizard_generic_save`** (resto, genérico por destino), `iac_tool_publicar`, `iac_tool_eliminar`, helpers `iac_link_val`/`iac_sideload`/`iac_acf_field_choices`/`iac_clean_value`.
`includes/gemini.php`	Adaptador LLM **dormido** (no se usa; BYOK si se reactivara).
`assets/chat.js` / `chat.css`	UI del wizard (mensajes, botones, subida, buscador en vivo).
`assets/widget.js` / `widget.css`	Botón flotante + panel lateral.

Tipos de campo del wizard: `text | textarea | url | select | taxonomy | upload`

(`upload` con `accept` = `image|pdf|media|file`; `select` toma sus opciones **en vivo** del campo ACF).

---

3. Cómo añadir / ajustar un tipo de contenido (futuro)

1. **Inspeccionar** los campos reales del CPT en el WP (ACF y/o meta) con un `wp eval-file` (ver snippet

en §6). Anotar: post type, taxonomías, campos ACF (nombre + tipo + choices) y/o meta propia.

2. Añadir una entrada a `iac_schema()` en `schemas.php` con sus campos y `target` de cada uno.

3. Si es un tipo nuevo, agregarlo al orden en `iac_schema_types()`.

4. Desplegar (§5) y probar con `wp eval-file` (crear borrador → verificar campos → borrar).

No hay que tocar `wizard.php` ni `tools.php`: son genéricos sobre el esquema.

---

4. Inventario de tipos de contenido del portal (descubierto 2026-06-08)

Post type	Etiqueta	Creado por	Campos clave
`informe`	Informes (ITPS)	tema/ACF	taxonomías `itps`+`ano`; `parrafo`(meta), `autor`, `imagen`, `video`, `btn-*`
`documento`	Documentos	ACF	`ano`(select), `archivos`(file) + imagen destacada
`guia`	Guías	ACF	contenido (post_content), `capitulo`(select) + portada
`infografia`	Infografías	ACF	`info1`/`info2`/`info3`(file: imagen o PDF)
`mapa`	Mapas	ACF	`categoria_de_mapa_`(select) + imagen + descripción
`insor_ca_item`	Contenidos Accesibles	plugin `insor-contenidos-accesibles`	meta `_insor_ca_enlace`(YouTube)/`_entidad`/`_anio`/`_tags`
`insor_ev_item`	Eventos en Vivo	plugin `insor-eventos-vivo`	meta `_insor_ev_fecha`/`_enlace`
`insor_short`	Shorts	plugin `insor-shorts-slider`	ACF `youtube_short_url`, `custom_cover_image`
`insor_slide`	Slides	plugin `insor-slider`	ACF `texto_banner`,`button_text`,`button_url`,`video_id` + imagen
`insor_acceso`	Accesos Directos	plugin `insor-accesos-directos`	ACF `acceso_imagen`, `acceso_url`
`insor_grid_item`	Grid Gallery	plugin `insor-grid-gallery`	título + imagen destacada

> El resto de CPTs registrados son del constructor (Salient/Nectar/WPBakery) o internos de ACF — no son

> contenido editorial.
