

Roadmap Unificado (reemplaza “ROADMAP VS ROADMAP”)

Brain(F6) + “másinteligente”+F7

Fecha: 01/01/2026

Este documento reemplaza el PDF “ROADMAP VS ROADMAP” y consolida ambos roadmaps en un plan único y ejecutable.

Nota de numeración

En el PDF original hay choques de numeración: “F6.5” aparece como Telemetría en una parte y como Front/UI en otra; y “F6.6” aparece como NLG en una parte y como Observabilidad en otra.

Para evitar confusión, este documento usa la siguiente numeración sin choques:

∴ F6.5 = Telemetría / Dataset F6.6 = NLG
controlado F6.7 = Evaluación de
regresión (harness) F6.UI = Front/UI (split
de core.js) F6.OBS = Observabilidad /
Auditoría

0) Baseline + invariantes (CERRADO: no reabrir, solo proteger)

- Pending 100% server-side + confirm/cancel coherente.
 - Queries A1-A8: read-only, sin pending, sin botones, sin mutar memoria.
 - Pipeline por flows (anti-parches en handle_chat).
 - ResponseBuilder único.
 - Full nunca al modelo (solo debug humano + assert).

Hallazgos / deuda a corregir sin romper UX

- Front: assets/js/admin-agent/core.js es “god file” grande.
- Riesgo conceptual: Brain menciona “openai_client” / modelo hardcodeado.
- Deuda: ModelFlow/ChatFlow no inyecta Context Lite estructurado (no se ve explícito).

F6.0) Auditoría técnica + FILE MAP (preparación)

- FILE MAP del plugin: carpetas/responsabilidades, dónde vive cada flow, tags @FLOW y @INVARIANT.
- Mapa dependencias LLM: dónde se llama al “cliente LLM”, dónde hay modelo hardcodeado, menciones OpenAI Key/client.
- DoD: doc maestro + lista exacta de puntos a corregir (sin tocar código aún) + checklist de pruebas base.

F6.1) SaaS-first real (sin cambiar UX)

- Objetivo: Brain decide modelo; toda IA via Core → SaaS; metadata obligatoria.
 - Eliminar/aislar strings tipo gpt-... del Brain (SaaS decide modelo/estrategia).
 - Normalizar naming: Brain habla “LLM client” abstracto (no “OpenAI” como dependencia directa).
- Metadata en requests: feature=brain, action=brain_chat|brain_parse|brain_nlg, trace_id si existe.
- DoD: ninguna ruta decide modelo; todo via SaaS; si SaaS cae → fallback seguro (1 mensaje, 1 salida).

F6.2) ChatFlow robusto (ModelFlow consolidado, NO borrar)

- Regla clave: ModelFlow NO se borra; se consolida como ChatFlow (fallback final).
 - Siempre llama SaaS.
 - No hardcodea modelo.
 - Fallback seguro ante error.
- Injectar Context Lite real (estructurado) + Full prohibido (+ assert).
- DoD: “hola/qué podés hacer” y conversación general ⇒ IA vía SaaS; Full nunca al modelo.

F6.3) Robustez conversacional “sin adivinar” (followups + números) + trazas

- Parse numérico robusto: 1.000 → 1000, 10k, 10lucas, conservar -5, reglas claras para 0.
- Followups solo con señales fuertes: pending_action o contexto fuerte (ej. last_product + TTL).
- Si falta target: preguntar con hints (último/primero/ID/SKU).
- Si falta campo: preguntar “¿precio o stock?”.
- Trazas: router_decision + followup_applied|rejected_reason.

F6.4) IntentParseFlow (brain_parse) via SaaS + JSON estricto + Validator/Policy + fallbacks

- Objetivo: comprensión libre real, pero con guardrails deterministas.

4.1) Pipeline unificado (sin duplicar lógica)

- PendingFlow
- QueryFlow (A1-A8 + followups “full/¿cuáles?”)
- FollowupFlow
- IntentParseFlow (nuevo) → SaaS brain_parse
- DeterministicFlow (fallback)
- TargetedUpdateFlow (fallback)
- ChatFlow/ModelFlow (chitchat final)

4.2) Contrato brain_parse: solo JSON, schema v1.0

- schema_version: "1.0", kind: action|query|chitchat|unknown, confidence, needs_clarification, clarify{question,options}.
- Action: intent set_price|set_stock, selector last/first/id/sku/name/unknown, field, raw_value_text (+ hints opcionales).
- Query: A1..A8 + mode summary|full|top5 + followups.

4.3) Validator/Policy (Brain) – allowlist + umbrales + reglas de seguridad

- Allowlist: acciones solo set_price/set_stock; queries solo A1..A8.
- Umbrales conservadores: action ≥ 0.65 , query ≥ 0.60 , chitchat ≥ 0.50 .
- Requeridos action: selector != unknown, raw_value_text no vacío, parse_number válido, coherencia intent↔field.
- Negativos: rechazar y pedir aclaración; 0 permitido pero con cuidado.
- Selector por name ambiguo o muy corto → pedir ID/SKU o last/first.
- Queries jamás crean pending; si mode unknown → default summary.

4.4) Fallbacks + anti-duplicación

AutoProduct AI - Roadmap Unificado (Brain)

- Si kind=query pasa validator → QueryFlow (un solo lugar).
- Si kind=action pasa validator → crear pending_action + botones.
- Si needs_clarification → preguntar con opciones.
- Si falla → Deterministic/Targeted; si no aplica → ChatFlow.
- Anti-duplicación: si un flow ya capturó acción, el otro no corre (y viceversa).

F6.5) Telemetría + dataset real (auditable)

- Extender trace: parse_json (sanitizado), validation_fail_reason, clarification_asked, user_correction_detected.
- Export dataset JSONL (input + lite_state + resultado, sin PII): endpoint admin-only o archivo rotativo.

F6.6) NLG controlado (más humano sin riesgo)

- Decisiones 100% deterministas.
- Mejorar redacción con templates/microvariantes.
- (Opcional) brain_nlg via SaaS SOLO reescritura (prohibido inventar).

F6.7) Evaluación de regresión (harness)

- Script/harness que corre dataset y mide: intent accuracy, ratio de aclaraciones, loops, errores peligrosos.

F6.UI) Front/UI — desarmar core.js sin romper contrato

- Partir assets/js/admin-agent/core.js en módulos sin bundler: chat-api.js (fetch/nonce/retries/locks), chat-render.js, chat-state.js, chat-scroll.js, chat-selectors.js.
- Mantener contrato: UI muestra botones solo si store_state.pending_action != null.
- DoD: mismo comportamiento visible; sin regresiones autoscroll/pending cards.

F6.OBS) Observabilidad y auditoría (sin tocar SaaS)

- trace_id end-to-end en responses.
- Logs con feature/action.
- Modo debug humano usa Full pero nunca al modelo.

QA — Definition of Done (antes de cada ZIP)

- Queries A1-A8: sin pending/botones/mutar memoria; A8 summary+full+top5 OK.
- Pending: confirm/cancel siempre; sin loops; sinónimos de cancelación normalizados; post-action coherente.
- Followups: “mejor a 10 lucas” solo con pending/contexto fuerte; sin target/campo ⇒ pregunta con hints.
- Conversación general: IA siempre via SaaS; si SaaS falla ⇒ fallback seguro.

AutoProduct AI - Roadmap Unificado (Brain)

- Invariantes: Full nunca al modelo (assert), ResponseBuilder único, php -l OK, sin notices/warnings nuevos.

Anti-parches (siempre)

- No agregar lógica nueva en handle_chat() salvo delegación.
- Caso nuevo ⇒ flow/handler nuevo o ajuste del flow existente.
- Mantener WHY en cambios.

F7 (post-F6 estable) — Migración Brain full al SaaS

- F7.1 Brain Engine portable (aislar dependencias WP en adapters; contrato pending/store_state idéntico).
- F7.2 Plugin tienda como Bridge Brain (UI + endpoints mínimos; arma Lite; llama SaaS Brain Engine; renderiza).
- F7.3 SaaS decide modelo definitivo (plugin nunca conoce modelos).
- DoD: switch local↔SaaS sin romper UI; misma semántica pending/store_state.

Ejecución: ¿Cuál ejecutamos en el nuevo chat?

- Camino crítico (orden recomendado): F6.0 → F6.1 → F6.2 → F6.3 → F6.4.
- Después (en paralelo o en el orden que prefieras): F6.5 / F6.6 / F6.7.
- Frontend cuando toque: F6.UI.
- Soporte/diagnóstico: F6.OBS.
- F7 solo cuando F6 esté estable.

Roadmap general FINALIZADO, DE AHORA PARA ADELANTE, SE MUESTRA LO QUE SE ESTUVO ACTUALIZANDO, PARA QUE ESTES EN CONTEXTO:

BLOQUE BASE – INVARIANTES: CERRADO – No se modifica

F6.0 FILE MAP: PENDIENTE – Documentación técnica obligatoria

F6.1 SaaS-first: HECHO ✓

F6.2 ChatFlow / ModelFlow: HECHO ✓

F6.3 Robustez conversacional: HECHO ✓

F6.4 IntentParseFlow: HECHO ✓

F6.5 Telemetría / Dataset: HECHO ✓

F6.6 NLG controlado: HECHO ✓

F6.7 Evaluación de regresión (Harness extendido): PENDIENTE – Debe cerrarse

F6.UI Front/UI: HECHO ✓

F6.OBS Observabilidad: PARCIAL – Falta cierre formal CIERRE

F6: NO COMPLETADO – Requiere F6.0 + F6.7 + cierre F6.OBS

F7 Gestor completo de tienda: FUTURO – Se inicia solo cuando F6 esté cerrado

