Ventajas de OpusMagnum v1.14.0: RPA, ETL, ESB/WSO2, API Gateways y Plugins

1. MATRIZ 22 × 22

462 pares · 2.407 entradas

22 tablas YAML (1.499 exact · 525 equivalent · 383 partial) sobre 162 operaciones canónicas; AlchemyBridge (.nap) como destino extra.

2. BLUEPRINT + AUDITORÍA

Estática · sin IA

opus document escribe PROCESS_BLUEPRINT.md/.json; fallas latentes, CWE y deprecados con la evidencia del código.

3. ORACULUM

3 modos de IA

docs documenta; annotate agrega estrategia; remediate traduce nodos sin mapear y optimiza pasos degradados con el catálogo oficial del destino.

4. PLUGINS + UPGRADE

--plugin · WSO2 MI 4.3.0

SDK de plugins (archivo .py, carpeta o módulo) y actualización in-place WSO2 → MI 4.3.0 con lo aplicado y lo pendiente.

✕ LO QUE SUELE SALIR MAL EN UNA MIGRACIÓN

1. Migración manual: lenta y cara, y puede introducir deuda nueva: saltos GOTO, pasos deshabilitados que se arrastran y conexiones a base de datos sin cerrar. Nadie cuenta qué se perdió.

2. Un solo LLM genérico: no es determinista, puede inventar acciones que no existen en el destino y pierde contexto en proyectos grandes; tampoco dice qué dejó afuera.

3. Documentación inexistente: el PDD se levanta a mano con entrevistas y queda desactualizado frente al código real.

4. Bloqueos en ejecución desatendida: cuadros MessageBox y pulsaciones SendKeys detienen los bots nocturnos. En un portafolio real de 30 procesos, sus conversion.json cuentan 483 MessageBox y 2.026 SendKeys.

✓ QUÉ HACE OPUSMAGNUM v1.14.0

1. Dos etapas (opus document → PROCESS_BLUEPRINT.md): documenta el proceso y después migra desde el blueprint, que se rehidrata desde sus trazas ocultas con la tabla de la plataforma de origen (editar las tablas visibles no cambia la migración).

2. Auditoría estática determinista: fallas latentes, vulnerabilidades CWE y componentes deprecados con la línea de evidencia, y un prompt sugerido para revisarla con IA que OpusMagnum no ejecuta. En WSO2, actualización in-place a MI 4.3.0.

3. Recomendador por reglas fijas (opus recommend): clasifica el origen en uno de 6 arquetipos y propone al menos 3 opciones (simples o compuestas, open source y comercial/cloud) con puntajes heurísticos declarados. No usa IA.

4. Plugins (opusmagnum.plugins · --plugin): dialectos propietarios con detección Boot-First (artefactos activos y muertos), tabla Rosetta y ganchos de auditoría, sin tocar el núcleo.

¿Cómo Funcionan el Motor Determinista y Oraculum?

CAPAS 1 Y 2: MOTOR DETERMINISTA + ORÁCULO (/ai/decide)

Paso 1 — Lectura al IR: el lector de cada plataforma arma el IR canónico: flujos, pasos, variables, expresiones y selectores. Los pasos deshabilitados se trasladan, salvo con --omitir-deshabilitados (se declara cuántos con RPX-C018).

Paso 2 — Mapeo y escritura con balance: cada acción se traduce con la tabla de su plataforma y queda rotulada exact, equivalent, lossy, shim o manual; lo que no tiene destino queda como nodo sin mapear con diagnóstico. El balance leídos = emitidos + conservados (Δ = 0) es un conteo de pasos, no una prueba de ejecución.

Paso 3 — Oráculo /ai/decide (opcional): pregunta a un motor local o remoto permitido y lee probabilidades del primer token. Son probabilidades sin calibrar; si el oráculo no responde, la salida registra el fallback.

- **choice (A / B / C):** A, proceso intensivo en datos/SQL/Excel (candidato a modelo híbrido con Cloud); B, intensivo en interfaz gráfica o SAP GUI (Power Automate Desktop local); C, mixto, con refactorización previa del control de flujo.
- **noul (0 a 1) y score (0 a 3):** probabilidad de que el proceso necesite refactorización previa y nivel de riesgo operativo.

CAPA 3: CONSEJO Y TRES MODOS DE INTERVENCIÓN

Ronda 1 — investigadores en paralelo: por defecto, motores locales (ollama y lmstudio_coder; LM Studio sólo si se configura OPUS_LMSTUDIO_BACKEND; si no, responde Ollama y el informe lo dice). Claude sólo con ANTHROPIC_API_KEY y OPUS_AI_DATA_CLASS distinto de transactional.

Crítica del juez: el juez (lmstudio_agentic o claude-opus-5-5) compara los diagnósticos y emite una directiva de refinamiento.

Ronda 2 y veredicto: los investigadores responden la directiva; el juez devuelve un JSON final con su puntaje por ronda (autoinformado y sin calibrar) y la estrategia propuesta.

docs y annotate: docs enriquece el PDD sin tocar el AST; annotate agrega estrategia y puede convertir MessageBox en Logging.LogMessage cuando noul ≥ 0,4.

remediate: invoca al agente codificador con el catálogo oficial del destino. Traduce nodos unmapped, optimiza SendKeys y actualiza estadísticas y diagnósticos.

Soberanía y costo: los datos transactional quedan en motores locales; cada llamada registra modelo, tokens y costo estimado dentro del presupuesto configurado.

5 Puntos Concretos del Código Migrado

Tema	Riesgo habitual	Qué hace OpusMagnum	Cómo se comprueba
1. Pasos sin equivalente	Quedan en blanco o se pierden sin aviso.	El motor conserva el hueco con diagnóstico. annotate agrega estrategia; remediate puede traducirlo con el catálogo oficial de acciones del destino.	conversion.json: unmapped, remediated y diagnósticos
2. Ejecución desatendida	Los MessageBox y SendKeys heredados vuelven frágil el bot.	annotate puede convertir MessageBox según noul; remediate también optimiza SendKeys y recalcula la conversión.	conversion.json: msgbox_sanitized_count y remediation
3. Pasos deshabilitados	Se arrastran líneas comentadas al destino.	--omitir-deshabilitados no los traslada y declara cuántos (RPX-C018).	Diagnóstico RPX-C018 en la salida
4. Secretos y datos personales	Contraseñas y cadenas de conexión pasan al destino.	El escáner de secretos (reglas SEC001-SEC019) los informa en opus audit y los tapa (<REDACTED:...>) en el paquete de conocimiento.	opus audit (sin --sin-secretos)
5. Destino según el origen	Se migra todo al mismo robot por costumbre.	opus recommend propone destinos por arquetipo con reglas fijas, incluidas combinaciones (gateway + servicio).	JSON con engine = rule-based y score_basis

¿QUÉ APORTAN ORACULUM Y EL AGENTE CODIFICADOR?

- **Intervención elegida:** docs sólo documenta; annotate agrega estrategia; remediate permite cambios completos del AST mediante el agente codificador.
- **Consumo con trazas:** cada llamada registra modelo, tokens y costo estimado; la validación contra el runtime real del destino sigue siendo parte del proyecto.

[QUÉ ESTÁ MEDIDO Y QUÉ NO · AUDITORÍA DEL 6/11/2026] Cifras que la versión anterior presentaba como medidas, contra lo que hace el código

LÁMINA 4 / 6

Qué Está Medido y Qué No en OpusMagnum v1.14.0

Tema	Lo que se afirmaba antes	Lo que hace el código	Evidencia en el repositorio
1. Matriz	23 × 23, 529 rutas, Δ = 0 en todas.	22 × 22: 462 pares dirigidos más AlchemyBridge como destino; Δ cuenta pasos.	MATRIZ_DOBLE_ENTRADA_23X23.html · tools/audit_catalog_metrics.py
2. Calidad del consejo	97,1 % y curvas por ronda medidas.	Puntaje que el juez devuelve en su JSON, sin calibrar; si falta, «desconocida».	EVALUACION_COMPARATIVA_…_30_PROCESOS.md
3. Costo de IA	Gasto real facturado; 65,9 % absorbido en local.	Tokens de usage o estimados; el costo es siempre estimado.	ORACULUM_MULTI_AGENT_COUNCIL_Y_CLAUDE.md
4. Laboratorio	22/22 PASS con CLIs reales en Docker.	Runners simuladores en opusmagnum/engines-lab:1.0; no ejecutan los motores reales.	GUIA_REPRODUCCION_Y_CONTENEDORES.md
5. AlchemyBridge	24,42 ms y 52× / 594× más rápido.	~1 ms por pedido tras 324 ms de arranque y 36 MB de RSS; ejecuta 6 de 14 pasos. Sin comparación medida con otros productos.	BENCHMARK_TIEMPOS_Y_RECURSOS.md
6. Gateways y Power Apps	1:1 exacto; «cero cajas vacías».	Gateways: 0 exact · 3 equivalent · 63 partial · 3 gap por tabla; pasos como trazas (RPX-C073) o marcadores (RPX-C074).	mapping/tables/*_canonical.yaml

QUÉ SE PUEDE MOSTRAR HOY

• **Ejemplos que corren:** los 7 ejemplos de examples/ terminan bien en la Mac mini M2 (6/11/2026): UiPath → Robocorp, PDI → Hop, n8n → Ballerina, PAD → PA Cloud, síntesis 1:N, WSO2 (blueprint, auditoría y upgrade) y plugin de dialecto propio.

• **Datos de un portafolio real:** los conversion.json de 30 procesos de un cliente están en el repositorio, con procedencia parcialmente desconocida; la evaluación corregida explica cómo leerlos. Sus nombres no se repiten.

¿Qué Recibe el Cliente por Cada Proceso?

ENTREGABLES QUE GENERA LA CLI

1. Paquete del destino (opus convert <ruta> --to <destino> --out <dir>): por ejemplo, la solución de Power Automate Desktop (.zip, con los scripts Robin si se pide --robin), un proyecto de Hop o Talend, o un compuesto AlchemyBridge con su .nap firmado.

2. PDD y SDD (opus docs o convert --docs): inventario, reglas, consultas SQL, variables y mapeo paso a paso, extraídos del código.

3. Diagramas e inventario: flujo.mmd y secuencia.mmd (Mermaid) e inventario.csv, del análisis estático, sin IA.

4. conversion.json y diagnósticos: conservación, cobertura y fidelidad; diagnósticos en texto, JSON o SARIF. Con --ai-agent, además el bloque ai_agent con las llamadas, los modelos que respondieron y el criterio de parada.

MUESTRA DE 4 PROCESOS: CÓMO LEERLA

Procesos 1 a 3 (A360 → PAD): fidelidad antes → después del pase de IA: 75,6 % → 76,1 %, 81,1 % → 81,1 % y 84,8 % → 84,8 %. La subida del proceso 1 fue un cambio de etiqueta de 4 pasos, no una mejora.

Proceso 4 (PAD migrado a mano → PAD): 40,0 % → 90,0 %: es el cambio de etiqueta de 1.086 pasos shim → equivalent que hacía el código de entonces; el código actual ya no lo hace.

MessageBox: 0 transformados en los 4 procesos (la versión anterior decía 68).

Nombres y datos: los procesos son de un cliente y se los llama Proceso 1 a 4. Sus entregables completos, con datos del cliente, no están en el repositorio.

Tres Pilares: Velocidad, Calidad Verificable y Ecosistema

1. VELOCIDAD

Documentación desde el código: PDD, SDD, blueprint e inventario salen de la CLI sin entrevistas. No hay una medición publicada de horas ahorradas.

Carteras completas: opus batch recorre un árbol de exports con varios trabajadores (--jobs) y genera informes por proceso y de cartera.

Menos retrabajo: huecos, MessageBox, SendKeys, pasos deshabilitados y secretos se informan antes de probar.

Destino por caso de uso: opus recommend sugiere opciones simples o compuestas con reglas fijas.

2. CALIDAD VERIFICABLE

Determinismo: con --ai-mode none, la misma entrada da la misma salida.

Balance Δ por proceso: leídos = emitidos + conservados; cada paso con su fidelidad y sus diagnósticos (opus rules lista el catálogo).

Pruebas: suites con pytest en tests/ (más las del motor local de Oraculum) y 7 ejemplos ejecutables en examples/; cualquiera puede correrlas.

Consejo de IA con trazas: cada llamada registra el modelo que respondió, sus tokens y un costo estimado.

3. ECOSISTEMA NETALCHEMY

Augur + SCP: Augur consume /scp/execute para dossiers y onboarding. SCP suma 88 providers desplegables, 794 operaciones y 82/82 emuladores; 29 providers cubren las fuentes requeridas por Augur.

Oraculum: gobierna modelos locales o remotos con modos docs, annotate y remediate, reglas de residencia, presupuesto y trazas.

ORQ: coordina tareas, mensajes, dependencias y agentes desde Orq Studio; el MCP nativo de 21 herramientas sigue como siguiente etapa.

Límite operativo: emulador y prueba automatizada no equivalen a homologación contra el host oficial; cada provider conserva sus requisitos de credenciales, licencia y consentimiento.