1 · Arquitectura actual

# Arquitectura actual de EMPRESA_JOSESAID

11 microservicios con sufijo -ms · revisión del 30 de octubre de 2026

Este documento describe la plataforma tal como opera hoy y la propuesta para incorporar inteligencia artificial mediante Nexus. La arquitectura presentada es similar a la de un proyecto previo desarrollado para un cliente anterior; por compromisos de confidencialidad, algunos componentes, nombres y detalles fueron adaptados. EMPRESA_JOSESAID es una empresa ficticia.

Hoy la plataforma está dividida en **8 servicios de negocio**, **1 de identidad** y **2 de infraestructura**. Todo el tráfico entra por **gateway-ms**, que enruta cada petición con el registro de **discovery-ms** (Eureka). Los eventos de negocio viajan por **Kafka** con patrón outbox, y cada servicio tiene su propia base de datos PostgreSQL.

**Nomenclatura**

Servicio existente · Bus de eventos Kafka · IA en el plan · IA propuesta · Sin IA en el plan




Todos los servicios reciben tráfico desde gateway-ms · base: Java 21, Spring Boot 4.1.2, PostgreSQL 17

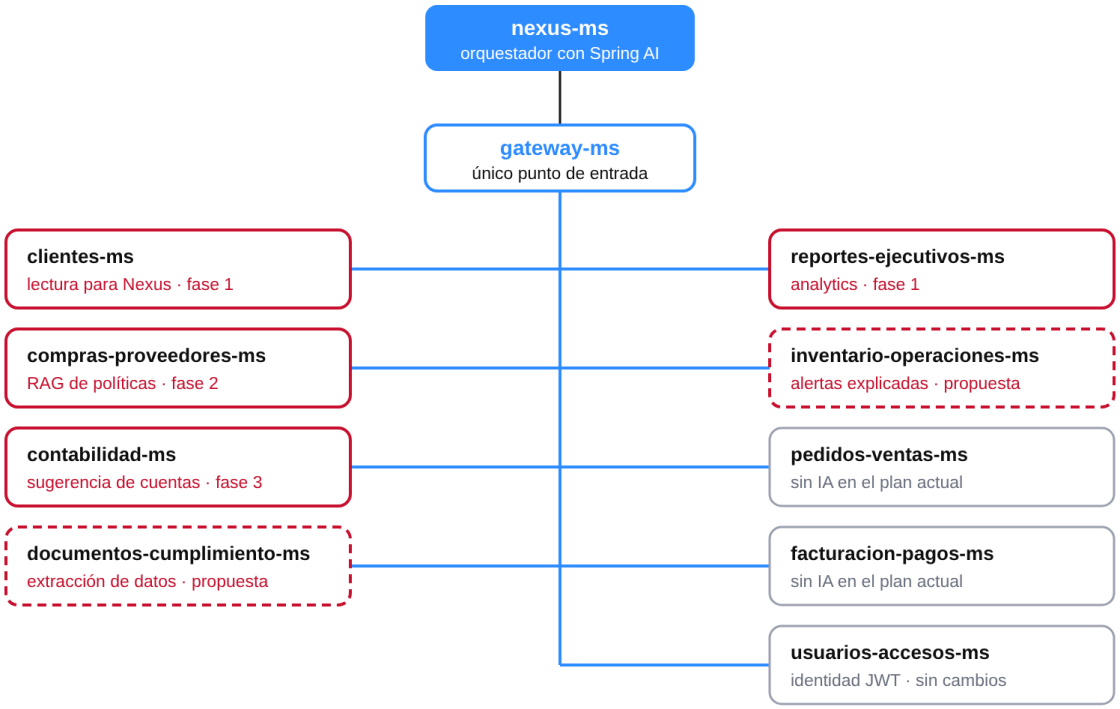
| Servicio | Área | Qué hace hoy |
|---|---|---|
| gateway-ms | Infraestructura | Punto de entrada único; enruta según Eureka |
| discovery-ms | Infraestructura | Registro y descubrimiento de servicios |
| usuarios-accesos-ms | Identidad | Autentica, emite JWT RS256, administra roles y revoca sesiones |
| clientes-ms | Comercial | Prospectos, clientes, contactos, interacciones y consentimientos |
| pedidos-ventas-ms | Ventas | Cotizaciones que pasan a pedido; confirmación, entrega, devolución y métricas |
| facturacion-pagos-ms | Facturación y pagos | Comprobantes con folio, cobros y cancelaciones (sin timbrado fiscal aún) |
| inventario-operaciones-ms | Operaciones | Existencias por almacén, reservas, lotes y órdenes de producción o servicio |
| compras-proveedores-ms | Compras | Proveedores con datos fiscales cifrados, órdenes de compra y aprobación |
| contabilidad-ms | Finanzas | Catálogo de cuentas, pólizas balanceadas, reversas y cierres de periodo |
| documentos-cumplimiento-ms | Legal | Contratos y registros con versiones, cifrado, alertas y retención |
| reportes-ejecutivos-ms | Dirección general | Indicadores consolidados por periodo y exportación |

2 · Propuesta con el uso de IA

# Dónde se incorpora la IA

Nexus es una capa nueva: no reemplaza ningún servicio existente

Nexus se incorpora como una capa nueva que consume los servicios actuales a través de **gateway-ms**, con el token de cada usuario, de modo que los permisos existentes se respetan sin cambios. El mapa muestra en qué servicio entra cada componente de IA.




- **Entrada única:** Nexus llama a los servicios solo a través de gateway-ms. Ningún servicio se conecta directo a un modelo.
- **Cuatro servicios con IA en el plan:** clientes-ms (lectura), reportes-ejecutivos-ms (analytics), compras-proveedores-ms (RAG de políticas) y contabilidad-ms (sugerencia de cuentas).
- **Dos propuestas:** documentos-cumplimiento-ms (extracción de datos) e inventario-operaciones-ms (explicación de alertas).
- **Sin cambios por ahora:** pedidos-ventas-ms, facturacion-pagos-ms y usuarios-accesos-ms, que solo aportan identidad o datos.

**Regla de seguridad:** Nexus solo lee datos. Cualquier cambio propuesto por el modelo llega como borrador y la persona lo confirma.

2 · Propuesta con el uso de IA (continuación)

## Componentes nuevos

| Componente | Implementación | Estado |
|---|---|---|
| nexus-ms | Orquestador con Spring Boot 4.1.2 y Spring AI, registrado en Eureka y enrutado como /nexus/ | Por crear |
| nexus-comun | Módulo con filtro de datos sensibles, registro de tokens y prompts versionados | Por crear |
| RAG | pgvector en la base nexus, con espacios por área y filtro por rol | Por crear |
| **Agentes** | **Uno al inicio; agentes por dominio solo si se justifican** | **Fase posterior** |
| **Servidor MCP** | **Expone herramientas con el token del usuario** | **Fase posterior** |
| **Interfaz** IMPORTANTE | **Web propia o canal existente** | **Pendiente de decidir** |

**IMPORTANTE: DECISIÓN DE LA INTERFAZ**

Se está evaluando quiénes serán los clientes de la arquitectura v2.0. Esa respuesta define si Nexus será una web propia, un canal como Teams o Slack, o ambos.

## Fases y pasos

1. **Fase 0 · Cimientos** — 8 pasos (nexus-comun)
2. **Fase 1 · Consulta de reportes y clientes** — 14 pasos (nexus-ms)
3. **Fase 2 · RAG de compras** — 6 pasos
4. **Fase 3 · Cuentas contables** — 3 pasos
5. **Fase 4 · Servidor MCP** — 3 pasos
6. **Fase 5 · Agentes (opcional)** — 2 pasos
7. **Fase 6 · Interfaz** — 4 pasos

Total: 40 pasos. Cada fase cierra con pruebas en verde y una demostración.

## Decisiones tomadas

- **Proveedor de modelo:** Claude, con cuenta propia y acceso por API ya disponible.
- **Monto límite de aprobación en compras:** hasta 1 millón de pesos mexicanos por compra individual, para rastreo por Finanzas.
- **Presupuesto de tokens:** limitado a las evaluaciones de la primera semana, con el límite diario u horario del plan de Claude como tope. El monitoreo constante se agrega después.

**Propuesta para el presupuesto de tokens (primera semana)**

1. Medir los tokens de cada caso de evaluación durante la fase 0.
2. Si una corrida supera el límite horario o diario del plan, se pausa hasta el siguiente periodo.
3. Registrar el consumo por corrida en nexus-comun, para revisarlo al cierre de la semana.

## Decisiones pendientes

- **Interfaz** IMPORTANTE: web propia o canal como Teams o Slack, según los clientes de v2.0.
- **Modelo y datos:** confirmar si se permite enviar datos aunque estén redactados.
- **Nombre:** Nexus o Nexo, por la colisión con Sonatype Nexus.