



Integración omnicanal para la eficiencia logística en una empresa manufacturera

Jaime Andrés Libreros Marulanda

Proyecto de grado entregado para obtener el título de
Magister en Ingeniería de Software

Dirigido por
Mario Julián Mora Cardona

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Maestría en Ingeniería de Software
Santiago de Cali
Julio de 2026

Ficha Resumen

Trabajo de Grado Maestría en Ingeniería de Software

TÍTULO: Integración omnicanal para la eficiencia logística en una empresa manufacturera

1. Énfasis: Ingeniería de Software
2. Área de trabajo: Arquitectura de Software, Gobernanza de Datos y Logística 4.0
3. Tipo de proyecto (Aplicado, Innovación, Investigación): Aplicado
4. Estudiante: Jaime Andrés Libreros Marulanda
5. Correo electrónico: jlp61@javerianacali.edu.co
6. Dirección y teléfono: Clle 32 an N° 2b - 55, (+57) 3175716383
7. Director: Mario Julián Mora Cardona
8. Vinculación del director: Planta
9. Correo electrónico del director: mariomora@javerianacali.edu.co
10. Co-Director (Si aplica): No aplica
11. Grupo o empresa que lo avala (Si aplica): Empresa manufacturera del sector consumo masivo
12. Otros grupos o empresas: No aplica
13. Palabras clave (al menos 5): Omnicanalidad, Arquitectura Event-Driven, Gobernanza de Datos, MDM, Logística 4.0, Integración de Sistemas
14. ODS que aplica al proyecto (Agenda 2030): ODS 9 (Industria, innovación e infraestructura) y ODS 12 (Producción y consumo responsables)
15. Fecha de inicio: 5 de enero de 2026
16. Resumen: Este proyecto propone el diseño y validación de una arquitectura de integración omnicanal en tiempo cercano al real para mejorar la eficiencia logística y la consistencia de los datos en una empresa manufacturera del sector de consumo masivo. La problemática identificada corresponde a la fragmentación de información entre sistemas empresariales como ERP, POS y plataformas de comercio electrónico, cuya integración basada en procesos batch genera latencias, inconsistencias en inventario y errores en la promesa de entrega, afectando la experiencia del cliente y los indicadores operativos. El objetivo general fue diseñar una arquitectura soportada en principios de Gobernanza de Datos y un Modelo de Datos Maestros

que actuara como fuente única de la verdad para los dominios de inventario, pedidos y clientes. La metodología adoptada fue de carácter aplicado con enfoque descriptivo-propositivo, estructurada en tres fases: diagnóstico de la arquitectura actual, diseño de la solución basada en un patrón Event-Driven y validación mediante un prototipo funcional. Como resultados se obtuvo un modelo arquitectónico documentado, reglas formales de gobernanza de datos y un prototipo que evidenció reducción en latencia transaccional y mejora en consistencia de inventario bajo escenarios simulados de alta demanda. Entre las principales lecciones aprendidas se destaca la necesidad de alinear decisiones arquitectónicas con métricas de negocio, formalizar roles de datos y priorizar la observabilidad como atributo transversal para soportar la evolución hacia una estrategia de Logística 4.0.

Agradecimientos

Agradezco a Dios y a mi familia por su apoyo incondicional durante todo el proceso de la maestría; su paciencia y aliento fueron el soporte que hizo posible culminar este trabajo. En especial, a mi esposa Leslie Juliet Aramburo Lozano, por su comprensión y acompañamiento constante durante las noches y fines de semana que este proyecto exigió.

A mi director de trabajo de grado, Mario Julián Mora Cardona, por su orientación rigurosa, su disposición permanente y sus observaciones precisas, que elevaron la calidad técnica y metodológica de este proyecto.

A la empresa que sirvió de caso de estudio y a los profesionales que participaron en las entrevistas de diagnóstico, por su apertura para compartir el conocimiento operativo y técnico sin el cual este trabajo no habría tenido sustento empírico.

Finalmente, a los profesores y compañeros de la Maestría en Ingeniería de Software de la Pontificia Universidad Javeriana Cali, por los espacios de discusión que enriquecieron mi formación como arquitecto de software.

Resumen

Este trabajo aborda el desafío estratégico de habilitar una operación omnicanal confiable en una empresa manufacturera del sector de consumo masivo, donde la fragmentación de datos y la integración basada en procesos batch afectaban la promesa de entrega, la experiencia del cliente y la eficiencia logística. El proyecto tuvo como objetivo diseñar y validar una arquitectura de integración que permitiera unificar la visión de inventario, pedidos y clientes bajo principios formales de Gobernanza de Datos. Más allá de la solución tecnológica, la propuesta buscó establecer una base estructural para la toma de decisiones basada en información consistente y oportuna. La arquitectura diseñada, soportada en un enfoque de sincronización en tiempo cercano al real y un Modelo de Datos Maestros como fuente única de la verdad, fue evaluada mediante un prototipo funcional y escenarios simulados de alta demanda. Los resultados evidencian mejoras en consistencia transaccional y reducción de latencia, demostrando la viabilidad técnica y el impacto estratégico de la solución como habilitador de Logística 4.0 y crecimiento sostenible.

Palabras Clave: Omnicanalidad, Transformación Digital, Gobernanza de Datos, Arquitectura Empresarial, Logística 4.0

Abstract

This work addresses the strategic challenge of enabling reliable omnichannel operations in a manufacturing company within the mass consumption sector, where data fragmentation and batch-based integrations were negatively impacting delivery promise accuracy, customer experience, and logistical efficiency. The objective was to design and validate an integration architecture capable of unifying inventory, order, and customer domains under formal Data Governance principles. Beyond a technological solution, the proposal establishes a structural foundation for decision-making based on consistent and timely information. The architecture, supported by near real-time synchronization and a Master Data Model as a single source of truth, was evaluated through a functional prototype and high-demand simulated scenarios. The results demonstrate improved transactional consistency and latency reduction, confirming the technical feasibility and strategic impact of the solution as an enabler of Logistics 4.0 and sustainable growth.

Keywords: Omnichannel Strategy, Digital Transformation, Data Governance, Enterprise Architecture, Logistics 4.0

Índice general

Agradecimientos	5
1. Introducción	1
1.1. Definición del problema	1
1.1.1. Planteamiento del problema	2
1.1.2. Sistematización	2
1.2. Objetivos del proyecto	2
1.2.1. Objetivo General	2
1.2.2. Objetivos específicos	2
1.3. Delimitaciones y alcances	3
1.4. Justificación del trabajo de grado	3
1.5. Metodología de la investigación	3
2. Marco de referencia	5
2.1. Marco Teórico	5
2.1.1. Omnicanalidad y transformación digital	5
2.1.2. Gobernanza de Datos y Modelo de Datos Maestros	5
2.1.3. Arquitectura Event-Driven y sincronización en tiempo cercano al real	6
2.1.4. Logística 4.0 y alineación arquitectura-operación	7
2.1.5. Sistemas empresariales en arquitecturas omnicanal	7
2.1.6. Modelo C4 para representación arquitectónica	7
2.2. Estado del Arte	8
3. Desarrollo del Proyecto	11
3.1. Fase I: Diagnóstico de la arquitectura actual	11
3.1.1. Revisión documental de sistemas de la empresa	11
3.1.2. Caracterización técnica de los mecanismos de integración	12
3.1.3. Análisis estructural de la arquitectura actual	14
3.1.4. Modelado estructural mediante el modelo C4	15
3.1.5. Análisis consolidado de entrevistas a stakeholders	27
3.1.6. Análisis de indicadores de desempeño: línea base diagnóstica	34
3.2. Fase II: Diseño de la arquitectura de integración omnicanal	37
3.2.1. Principios de diseño	38
3.2.2. Framework de gobernanza de datos	39
3.2.3. Modelo de Datos Maestros (MDM)	42
3.2.4. Arquitectura TO-BE orientada a eventos	44
3.2.5. Trazabilidad diagnóstico-solución	56

4. Evaluación	59
4.1. Diseño de la evaluación	59
4.1.1. Entorno de evaluación	59
4.1.2. Alcance y supuestos del entorno controlado	59
4.1.3. Experimentos	60
4.1.4. Amenazas a la validez	61
4.2. Resultados de la evaluación	61
4.2.1. Resultados E1 — Prevención de sobreventa (TPFI)	61
4.2.2. Resultados E2 — Exactitud de Inventario y latencia (EI)	62
4.2.3. Resultados E3 — Eliminación de la causa raíz de las RMAs (COMI)	62
4.2.4. Verificación de los criterios de aceptación	63
4.2.5. Evidencia de ejecución del prototipo	63
4.2.6. Reflexión sobre los resultados obtenidos	67
5. Conclusiones	71
5.1. Conclusiones	71
5.2. Trabajos futuros	72
5.3. Lecciones aprendidas	72
Bibliografía	75

Índice de figuras

3.1. Vista de Contexto C4 – Arquitectura Actual	16
3.2. Vista de Contenedores C4 – ERP (Epicor)	18
3.3. Vista de Contenedores C4 – Sistema Retail (POS)	20
3.4. Vista de Contenedores C4 – Comercio Digital	22
3.5. Vista de Contenedores C4 – WMS	24
3.6. Vista de Contenedores C4 – Services Connect	26
3.7. Vista panorámica (<i>System Landscape</i>) – Arquitectura TO-BE	48
3.8. Vista de Contenedores C4 – Arquitectura TO-BE orientada a eventos	50
3.9. Saga orquestada del ciclo pedido–entrega (flujo OrderPlaced)	53
3.10. Visualización ilustrativa del ecosistema: camino feliz de la Saga (OrderPlaced → OrderClosed).	54
3.11. Visualización ilustrativa del ecosistema: rama de compensación ante StockUnavailable (OrderPlaced → OrderCancelled).	55
4.1. Los seis servicios del prototipo desplegados vía Docker Compose, con el broker Kafka en estado <i>healthy</i>	63
4.2. Contrato API del Order Orchestrator: GET /sagas , GET /sagas/{order_id} y GET /health	64
4.3. Respuesta real de GET /sagas : estado agregado del Event Store tras la ejecución de los experimentos (sagas cerradas y canceladas por compensación).	64
4.4. Contrato API del ERP (mock-erp): GET /stock/{sku} , autoridad del stock disponible y de las reservas.	65
4.5. Contrato API del MDM Inventory Service: GET /inventory/{sku} , consulta del <i>Golden Record</i>	65
4.6. Respuesta real de GET /inventory/SKU-SMOKE : el Golden Record refleja el movimiento de inventario propagado desde el WMS a través de la cadena ERP → MDM.	66

Índice de tablas

2.1. Tipos de sistema empresarial en una arquitectura omnicanal	8
2.2. Brechas identificadas en la literatura respecto al contexto del estudio	9
3.1. Participantes del instrumento de entrevista de la Fase I.	27
3.2. Indicadores cuantitativos del diagnóstico (línea base).	28
3.3. Mapa técnico de integraciones y frecuencias de sincronización.	29
3.4. Análisis de brechas: estado actual (AS-IS) vs. requerimientos omnicanal (TO-BE). . .	32
3.5. Indicadores clave de desempeño (KPI). Línea base diagnóstica.	34
3.6. Criterios de evaluación del prototipo basados en la línea base de KPIs.	38
3.7. Matriz de gobernanza de datos por dominio.	40
3.8. SLAs de calidad del dato por dominio.	41
3.9. Atributos canónicos del MDM por entidad.	43
3.10. Evaluación comparativa de brokers de mensajes.	46
3.11. Trazabilidad entre hallazgos del diagnóstico y componentes de diseño.	57
4.1. Resultados E1: sobreventa AS-IS (simulado) vs. TO-BE (prototipo).	61
4.2. Resultados E2: exactitud de inventario y latencia de propagación por ronda.	62
4.3. Verificación de los criterios de aceptación del prototipo.	63
4.4. Atributos de calidad, tácticas aplicadas y costo asumido.	67

Introducción

La transformación digital en el sector manufacturero ha impulsado la adopción de estrategias omnicanal como mecanismo para mejorar la experiencia del cliente y aumentar la competitividad (Mahadevan and Joshi, 2022; Yang and Hu, 2024). Sin embargo, la convergencia entre canales físicos y digitales exige que los sistemas empresariales operen con información consistente y disponible en tiempo cercano al real. En este contexto, la fragmentación de datos y la integración basada en procesos batch representan una barrera estructural para cumplir promesas de entrega confiables y mantener eficiencia logística.

El presente trabajo aborda el diseño y validación de una arquitectura de integración omnicanal orientada a unificar los dominios de inventario, pedidos y clientes en una empresa manufacturera del sector de consumo masivo. El objetivo general consistió en proponer una arquitectura soportada en principios de Gobernanza de Datos y un Modelo de Datos Maestros como fuente única de la verdad, habilitando sincronización transaccional en tiempo cercano al real.

La metodología adoptada fue de carácter aplicado con enfoque descriptivo-propositivo, estructurada en tres fases: diagnóstico de la arquitectura actual, diseño de la solución arquitectónica y validación mediante un prototipo funcional bajo escenarios simulados de alta demanda.

La relevancia del estudio radica en su contribución práctica a la alineación entre arquitectura empresarial y métricas operativas, fortaleciendo la toma de decisiones basada en datos consistentes y oportunos. Este trabajo aporta un marco replicable para organizaciones que buscan evolucionar hacia un modelo de Logística 4.0.

El documento se estructura en cinco capítulos: introducción y definición del problema; marco de referencia; desarrollo del proyecto y diseño arquitectónico; evaluación y validación de la solución; y conclusiones con implicaciones estratégicas y trabajos futuros.

1.1. Definición del problema

La empresa objeto de estudio operaba múltiples canales de venta, puntos de venta físicos y plataformas de comercio electrónico sin una arquitectura de integración unificada. Las sincronizaciones se realizaban principalmente mediante procesos batch, generando latencia en la actualización de inventario y divergencias entre sistemas. Esta situación provocaba inconsistencias en la promesa de entrega, cancelaciones por falta de stock percibido y aumento en reclamaciones de clientes.

El problema central se definió como la ausencia de una arquitectura de integración omnicanal en tiempo cercano al real que garantizara consistencia de datos y trazabilidad transaccional entre los dominios críticos del negocio.

1.1.1. Planteamiento del problema

La situación actual evidenciaba síntomas claros: discrepancias de inventario entre canales, retrasos en la actualización de pedidos y baja visibilidad transaccional. Las causas principales se asociaron a integraciones punto a punto, ausencia de un Modelo de Datos Maestros y falta de roles formales de gobernanza. De mantenerse esta situación, la expansión a nuevos canales incrementaría la complejidad operativa y el riesgo de pérdidas comerciales.

1.1.2. Sistematización

La investigación se orientó a responder las siguientes preguntas:

1. ¿Cómo diseñar una arquitectura de integración omnicanal que permita sincronización en tiempo cercano al real entre inventario, pedidos y clientes?
2. ¿Qué reglas de Gobernanza de Datos y qué estructura de Modelo de Datos Maestros son necesarias para garantizar consistencia transaccional?
3. ¿Cuál es el impacto potencial de la arquitectura propuesta sobre la latencia y la consistencia operativa?

1.2. Objetivos del proyecto

1.2.1. Objetivo General

Proponer y validar una arquitectura de integración omnicanal en tiempo cercano al real que unifique los dominios de inventario, pedidos y clientes, garantizando consistencia de información y habilitando eficiencia logística.

1.2.2. Objetivos específicos

1. Analizar la arquitectura actual de datos e integración para identificar fuentes de latencia e inconsistencias.
2. Definir reglas y roles de Gobernanza de Datos para las entidades críticas del negocio.
3. Diseñar un Modelo de Datos Maestros como fuente única de la verdad.
4. Diseñar la arquitectura de integración basada en un enfoque Event-Driven.
5. Evaluar la viabilidad técnica mediante el desarrollo y prueba de un prototipo funcional.

1.3. Delimitaciones y alcances

El trabajo se centró exclusivamente en los dominios de inventario, pedidos y clientes dentro del ciclo pedido y entrega. No se incluyó la implementación completa en producción ni la integración con sistemas de precios dinámicos o pronóstico de demanda. La validación se realizó mediante un prototipo en entorno controlado.

1.4. Justificación del trabajo de grado

Resolver el problema identificado permitió reducir riesgos operativos asociados a inconsistencias de inventario y promesas de entrega erróneas. Desde una perspectiva económica, la solución contribuye a disminuir cancelaciones y ventas perdidas. Desde el enfoque estratégico, establece una base estructural para la expansión omnicanal sostenible. En el ámbito tecnológico, fortalece la alineación entre arquitectura empresarial y métricas de negocio, consolidando prácticas de gobernanza como habilitador de transformación digital.

1.5. Metodología de la investigación

La investigación adoptó un enfoque aplicado con diseño descriptivo y propositivo. En la primera fase se realizó un diagnóstico de la arquitectura existente mediante revisión documental, entrevistas y análisis de indicadores históricos. En la segunda fase se diseñó la solución arquitectónica, incluyendo el Modelo de Datos Maestros y los patrones de integración. En la tercera fase se desarrolló un prototipo funcional para simular escenarios de alta demanda y medir latencia y consistencia transaccional. Finalmente, se compararon indicadores antes y después para evaluar el impacto de la propuesta.

Marco de referencia

En este capítulo se presentan los fundamentos conceptuales y los antecedentes investigativos que sustentan la propuesta de arquitectura de integración omnicanal. Se desarrolla el marco teórico que articula omnicanalidad, gobernanza de datos y arquitectura orientada a eventos, y posteriormente se analiza el estado actual de la literatura y de implementaciones documentadas, con el fin de identificar brechas que justifican la investigación.

2.1. Marco Teórico

2.1.1. Omnicanalidad y transformación digital

La omnicanalidad se define como la integración coordinada de múltiples canales físicos y digitales para ofrecer una experiencia coherente al cliente a lo largo de todo su ciclo de interacción. A diferencia del enfoque multicanal, donde los canales operan de manera independiente, la estrategia omnicanal exige sincronización de información, procesos y políticas operativas (Mahadevan and Joshi, 2022). La literatura reciente señala que la integración tecnológica es un habilitador crítico para garantizar calidad del servicio, visibilidad de inventario y consistencia en la promesa de entrega (Yang and Hu, 2024).

Desde la perspectiva de transformación digital, la omnicanalidad no es únicamente una estrategia comercial, sino un rediseño estructural de procesos soportado por arquitectura empresarial y gestión de datos. Esto implica que la confiabilidad de la información se convierte en un activo estratégico.

Desde una perspectiva arquitectural, Suárez Murcia (2019) formaliza la omnicanalidad como un sistema integrado de tres planos interdependientes: el plano de negocio, que define las políticas de coordinación entre canales; el plano de información, que garantiza la consistencia transaccional entre dominios; y el plano tecnológico, que provee los mecanismos de integración. La articulación simultánea de estos tres planos es condición necesaria para que la experiencia omnicanal sea operativamente sostenible y constituye el marco de referencia que estructura el diseño propuesto en el Capítulo 3.

2.1.2. Gobernanza de Datos y Modelo de Datos Maestros

La Gobernanza de Datos establece el conjunto de políticas, roles y controles necesarios para garantizar calidad, consistencia, trazabilidad y seguridad de la información organizacional. Su implementación permite definir responsabilidades claras sobre entidades críticas y establecer métricas de calidad alineadas con objetivos de negocio.

El Modelo de Datos Maestros (MDM) actúa como mecanismo para consolidar una “fuente única de la verdad” para entidades clave como cliente, inventario y pedido. Sin un MDM formal, los sistemas distribuidos tienden a generar duplicidad y divergencia de información, afectando procesos transaccionales y analíticos.

La literatura resalta que la gobernanza operativa, más que documental, es un factor determinante en la sostenibilidad de arquitecturas distribuidas (Moreira et al., 2023).

La práctica de gobernanza de datos en arquitecturas empresariales se formaliza a través del marco DAMA-DMBOK (DAMA International, 2017), que estructura once áreas de conocimiento entre las que se destacan, para el alcance de este trabajo, la gestión de datos maestros, la calidad del dato y la arquitectura de datos. Este marco define los roles fundamentales de *Data Owner*, *Data Steward* y *Data Custodian* como mecanismo de asignación de responsabilidades sobre las entidades de negocio críticas.

Loshin (2009) caracteriza tres estrategias arquitectónicas de implementación de un MDM: **Registry**, que mantiene únicamente referencias a los registros maestros sin copia local del dato; **Consolidation Hub**, que centraliza la producción del *Golden Record* en un nodo dedicado a partir de los sistemas fuente; y **Coexistence**, que mantiene sincronización bidireccional entre el MDM y los sistemas fuente. La elección entre estas estrategias depende del nivel de heterogeneidad de los datos, de la madurez técnica de los sistemas fuente y del requerimiento de latencia de propagación.

2.1.3. Arquitectura Event-Driven y sincronización en tiempo cercano al real

La Arquitectura Orientada a Eventos (Event-Driven Architecture, EDA) se basa en la producción, publicación y consumo de eventos como mecanismo de integración desacoplada entre sistemas. Este enfoque permite reducir dependencias punto a punto, mejorar escalabilidad y habilitar sincronización en tiempo cercano al real.

En contextos omnicanal, la latencia transaccional impacta directamente la precisión de inventario y la promesa de entrega. La adopción de patrones orientados a eventos facilita propagación consistente de cambios entre dominios, incrementando resiliencia y observabilidad.

La arquitectura orientada a eventos se consolida como alternativa a integraciones batch tradicionales, especialmente en entornos con alta variabilidad de demanda y múltiples puntos de interacción.

La transición desde integraciones *punto a punto* hacia arquitecturas orientadas a eventos se enmarca en los patrones de integración empresarial sistematizados por Hohpe and Woolf (2003), que sitúan la mensajería asíncrona como mecanismo para desacoplar temporal y espacialmente a productores y consumidores. Como infraestructura de mensajería, los *logs* de eventos distribuidos —cuyo exponente de referencia es Apache Kafka— aportan retención durable, capacidad de *replay* del historial y compactación por clave para conservar el último estado conocido de cada entidad (Kleppmann, 2017), propiedades que habilitan la reconstrucción del estado ante fallos y el procesamiento independiente por múltiples consumidores.

Para coordinar transacciones distribuidas que abarcan múltiples servicios, Richardson (2018) formaliza el patrón **Saga**, que descompone una transacción de negocio en transacciones locales con

acciones de compensación ante fallos. Admite dos variantes: la **coreografiada**, donde cada servicio reacciona de forma autónoma a los eventos del anterior, y la **orquestrada**, donde un coordinador central dirige la secuencia y gestiona la compensación, favoreciendo la trazabilidad y el diagnóstico de fallos.

2.1.4. Logística 4.0 y alineación arquitectura-operación

El concepto de Logística 4.0 integra digitalización, automatización y análisis de datos para optimizar la cadena de suministro. [Ellithy et al. \(2024\)](#) documenta cómo la incorporación de tecnologías de la Industria 4.0 —automatización de bodegas, vehículos de guiado automático y analítica de datos en tiempo real— transforma las operaciones logísticas hacia esquemas de mayor flexibilidad y trazabilidad.

En el dominio omnicanal, esta evolución exige reorganizar las operaciones logísticas en torno a la integración de canales: [Hübner et al. \(2022\)](#) muestra cómo la tienda física se reconfigura como nodo logístico activo —punto de *fulfillment* y de gestión de devoluciones— en lugar de un canal aislado de venta. En ambos casos, la arquitectura tecnológica deja de ser soporte operativo para convertirse en habilitador estratégico: la coherencia entre arquitectura empresarial, gobierno de datos y métricas logísticas es condición necesaria para garantizar eficiencia sostenible.

2.1.5. Sistemas empresariales en arquitecturas omnicanal

Una arquitectura de integración omnicanal se construye sobre un conjunto de sistemas empresariales especializados, cada uno responsable de un subconjunto del ciclo de negocio. Comprender el rol genérico de cada tipo de sistema es condición previa para analizar la topología de integración entre ellos. La Tabla 2.1 resume los cinco tipos de sistema relevantes para el dominio de estudio y su función en la arquitectura.

La instancia concreta de cada uno de estos tipos de sistema en la organización analizada se identifica en el Capítulo 3.

2.1.6. Modelo C4 para representación arquitectónica

El modelo C4, formulado por [Brown \(2018\)](#), propone una notación jerárquica para representar la arquitectura de sistemas de software mediante cuatro niveles progresivos de abstracción: **Contexto** (Nivel 1), que ubica el sistema en relación con sus usuarios y los sistemas externos con los que interactúa; **Contenedores** (Nivel 2), que descompone el sistema en aplicaciones y almacenes de datos desplegables; **Componentes** (Nivel 3), que detalla los bloques funcionales internos de cada contenedor; y **Código** (Nivel 4), que se aproxima a la representación de clases o estructuras de implementación.

El propósito del modelo es ofrecer una semántica común y mínima para que distintas audiencias técnicas y no técnicas puedan razonar sobre la arquitectura con un mismo lenguaje visual. Cada nivel responde a una pregunta arquitectónica distinta: el Nivel 1 responde *¿con qué se conecta el*

Tabla 2.1: Tipos de sistema empresarial en una arquitectura omnicanal

Tipo de sistema	Rol en la arquitectura omnicanal	Ejemplos comerciales
ERP	Núcleo transaccional y sistema de registro de inventario, pedidos y facturación. Concentra las reglas de negocio centrales.	SAP, Oracle, Epicor
POS	Punto de venta físico; captura transacciones en tienda y sincroniza ventas y existencias del canal presencial.	Cegid, Square, Oracle Retail
WMS	Gestión de almacén; controla ubicaciones, alistamiento (<i>picking</i>), despacho y movimientos físicos de inventario.	Manhattan, Blue Yonder, Cubic
Plataforma de comercio digital	Canal de venta en línea; gestiona catálogo, carrito y órdenes del canal digital y de <i>marketplaces</i> .	VTEX, Shopify, Magento
Integrador / <i>middleware</i>	Orquesta la comunicación entre sistemas; traduce formatos y media los flujos de integración entre canales y núcleo.	MuleSoft, Boomi, Services Connect

sistema?, el Nivel 2 responde *¿de qué partes desplegables se compone?*, el Nivel 3 responde *¿qué hace internamente cada parte?* y el Nivel 4 responde *¿cómo está implementado?*.

En el presente trabajo se emplean únicamente los niveles de Contexto y Contenedores. Esta delimitación se justifica por el objeto de estudio: el diagnóstico busca caracterizar la topología de integración entre sistemas empresariales y la naturaleza de los flujos de datos entre ellos, no la lógica interna de cada sistema. Los niveles de Componentes y Código quedan fuera de alcance porque corresponden a decisiones de diseño interno propietario de cada producto (Epicor, Cegid, VTEX, entre otros), sobre las que el proyecto no tiene visibilidad ni control, y porque su detalle no aportaría evidencia adicional para la identificación de brechas arquitectónicas de integración.

2.2. Estado del Arte

La revisión bibliográfica se construyó a partir de cinco trabajos referenciales seleccionados por su pertinencia metodológica y temporal (2022–2024), y se organizó en tres frentes: la dimensión comercial de la omnicanalidad, su aplicación operativa en logística y los enfoques de implementación tecnológica.

En el frente comercial, el estudio bibliométrico de Yang and Hu (2024) identifica cinco líneas dominantes en la literatura sobre omnicanalidad: calidad de servicio, integración *online*, *offline*

(O2O), retail omnicanal y transformación digital. La predominancia de estos enfoques refleja un sesgo hacia perspectivas de marketing y comportamiento del consumidor, con escasa profundización en los mecanismos de integración técnica que las soportan.

En el frente operativo, [Hübner et al. \(2022\)](#) formaliza la tienda como nodo logístico y modela la planificación de inventario para entornos omnicanal, mientras que [Ellithy et al. \(2024\)](#) documenta ganancias de eficiencia por automatización de bodegas y sincronización operativa. Ambos trabajos abordan el problema desde la optimización del proceso físico y dejan fuera de su alcance la arquitectura de datos que sustenta la sincronización.

En el frente tecnológico, [Batra and Kewalramani \(2024\)](#) describe estrategias progresivas de adopción omnicanal con énfasis en gestión del cambio y madurez digital. La revisión realizada en este trabajo no encontró estudios que documenten de manera integrada la implementación de un Modelo de Datos Maestros bajo una arquitectura orientada a eventos en contextos manufactureros.

Identificación de brechas

Con base en los trabajos revisados, formulamos un análisis comparativo sobre tres criterios: (i) nivel de integración tecnológica, (ii) formalización de Gobernanza de Datos, y (iii) articulación entre arquitectura y métricas operativas. La Tabla 2.2 sintetiza los hallazgos.

Tabla 2.2: Brechas identificadas en la literatura respecto al contexto del estudio

Dimensión analizada	Enfoque predominante en literatura	Brecha identificada
Integración omnicanal	Coordinación de canales y experiencia del cliente	Escasa formalización técnica sobre integración transaccional en tiempo cercano al real
Gobernanza de Datos	Recomendaciones conceptuales de calidad y roles	Limitada conexión operativa entre MDM y arquitectura de integración
Arquitectura tecnológica	Casos de adopción progresiva y gestión del cambio	Falta de documentación detallada sobre implementación conjunta de MDM y Event-Driven Architecture
Contexto industrial	Predominio de retail y servicios	Baja evidencia empírica en manufactura latino-americana
Alineación con métricas	Énfasis en satisfacción y lealtad	Ausencia de modelos que conecten arquitectura con KPIs logísticos

La revisión confirma que existen avances aislados en integración omnicanal, gobernanza conceptual y eficiencia logística, pero ninguno de los trabajos identificados articula simultáneamente Gobernanza de Datos, MDM y arquitectura orientada a eventos en un caso aplicado de manufactura con validación cuantitativa. Esa articulación es la contribución que este trabajo busca aportar.

Desarrollo del Proyecto

3.1. Fase I: Diagnóstico de la arquitectura actual

Con el fin de comprender las causas estructurales de la fragmentación de información y latencia transaccional identificadas en el problema de investigación, se desarrolló una fase diagnóstica compuesta por revisión documental, entrevistas semiestructuradas y análisis de indicadores históricos. Esta fase permitió establecer una línea base objetiva sobre el estado de la arquitectura de integración y la calidad de los datos.

3.1.1. Revisión documental de sistemas de la empresa

Se realizó un análisis estructurado de la documentación técnica disponible, incluyendo diagramas de integración, contratos de APIs, configuraciones de procesos batch y descripciones funcionales de los sistemas empresariales.

Los sistemas analizados incluyeron:

- ERP corporativo (gestión transaccional de inventario y pedidos).
- Plataforma de comercio electrónico.
- Sistema POS.
- Sistema de gestión logística.
- Herramientas de reportería y analítica.

El análisis evidenció:

- Integraciones punto a punto con alto nivel de acoplamiento.
- Predominio de procesos batch para sincronización de inventario.
- Ausencia de un repositorio consolidado de datos maestros.
- Falta de trazabilidad formal sobre modificaciones transaccionales.

Se identificó que la arquitectura existente corresponde a un patrón de integración punto a punto con sincronización batch, no diseñado para sincronización en tiempo cercano al real.

3.1.2. Caracterización técnica de los mecanismos de integración

A partir del análisis documental y las sesiones con los equipos de TI y Logística, se realizó una caracterización detallada de los mecanismos actuales de integración entre sistemas, con énfasis en sincronización de inventarios, ventas y ciclo logístico.

3.1.2.1. Gobierno y sincronización de inventarios

El ERP corporativo actúa como fuente oficial del inventario maestro. Desde este sistema se distribuye periódicamente una “foto” (snapshot) de la bodega principal PTER hacia los demás sistemas.

La sincronización de inventarios presenta las siguientes características:

- Actualización hacia Comercio Digital: cada 1 hora.
- Actualización hacia Sistema Retail (POS): cada 1 hora.
- Modelo de propagación basado en snapshot y no en eventos incrementales.

El uso de snapshots periódicos implica que los sistemas consumidores operan con una representación diferida del estado real del inventario, generando potenciales desfases temporales entre el stock físico y el stock disponible en canales.

La integración entre el WMS Cubic y el ERP es la excepción relevante: opera mediante *hooks* en tiempo real y cada movimiento físico registrado en el WMS genera una notificación inmediata al ERP sin pasar por un proceso batch. Este mecanismo es el único precedente de arquitectura orientada a eventos en el ecosistema tecnológico actual, y demuestra que la infraestructura ya soporta integración reactiva. Por lo tanto, el riesgo técnico de generalizar este modelo a los demás dominios es bajo.

3.1.2.2. Proceso de reaprovisionamiento de tiendas

El sistema POS utiliza la información recibida del ERP (bodega PTER) para calcular el disponible en el CEDI y, con base en parámetros de máximos y mínimos configurados por producto, genera solicitudes de reaprovisionamiento.

Este proceso se ejecuta mediante procesos batch programados en dos ventanas horarias:

- 3:00 AM
- 1:00 PM

Las solicitudes generadas se formalizan como documentos OTF, los cuales son gestionados por el área logística. El ciclo continúa de la siguiente manera:

1. Logística recibe documento OTF.

2. El WMS ejecuta el despacho físico.
3. El WMS confirma el despacho al ERP.
4. El ERP confirma al sistema POS.
5. El administrador de tienda confirma la recepción.
6. Se registra mensaje de recepción en el ERP.

Todo el ciclo descrito opera bajo un modelo batch secuencial, sin mecanismos de notificación en tiempo real.

3.1.2.3. Sincronización de ventas por canal

Se identificó una asimetría significativa en la frecuencia de sincronización de ventas:

- Ventas en tiendas físicas (Retail POS): sincronización con ERP cada 2 horas, entre 9:30 AM y 11:30 PM.
- Ventas en Comercio Digital: sincronización con ERP cada 15 minutos (vía API REST a través de Services Connect).

Esta diferencia genera niveles de latencia distintos entre canales, afectando la consistencia del inventario disponible consolidado en el ERP y propagado posteriormente a otros sistemas.

Un tercer grupo de canales de venta presenta un patrón de integración crítico: los **canales externos** (Bylmo, Falabella, Sodimac y similares) sincronizan sus ventas con el ERP mediante **archivos Excel de carga manual con frecuencia diaria**. Este mecanismo implica que las ventas realizadas en estas plataformas pueden acumular hasta 24 horas de latencia antes de reflejarse en el inventario consolidado del ERP, convirtiéndose en la fuente de mayor riesgo de sobreventa del ecosistema. A diferencia de los canales VTEX y POS, que tienen al menos un mecanismo técnico de sincronización programada, la integración con canales externos depende de intervención humana sin respaldo sistémico.

3.1.2.4. Despacho desde tienda (modelo Bogotá y Medellín)

Services Connect es el integrador central del canal digital: orquesta la recepción de órdenes desde VTEX y MercadoLibre mediante API REST cada 15 minutos, gestiona la sincronización de ventas del POS Cegid mediante archivos TXT cada 2 horas y actúa como puente entre los sistemas externos y el ERP Epicor. También implementa la lógica de habilitación de despacho en tienda, que se activa para pedidos con origen en Bogotá o Medellín bajo un flujo especial:

1. El pedido es validado mediante el integrador Services Connect.
2. La lógica de habilitación de despacho en tienda está parametrizada en el ERP.

3. Estos pedidos no se facturan desde el CEDI Yumbo (Cali).
4. Se facturan como venta mostrador desde la bodega de la tienda.
5. El inventario virtual se descuenta desde la tienda (ERP).
6. Se notifica al POS como salida extraordinaria.
7. Se envía correo electrónico al administrador de la tienda para ejecutar el despacho físico.

Este modelo introduce dependencias manuales y ajustes extraordinarios fuera del flujo estándar de distribución.

3.1.2.5. Falla operacional identificada en despacho tienda

Se identificó una situación crítica recurrente cuando el inventario físico en tienda no coincide con el inventario virtual registrado en el sistema:

1. El sistema asigna el pedido a una tienda con inventario virtual disponible.
2. La tienda valida físicamente y determina que no dispone del producto.
3. La tienda informa al CEDI la imposibilidad de despacho.
4. El CEDI debe generar una nota crédito.
5. El pedido se vuelve a facturar manualmente.
6. Se reasigna el despacho a otra tienda.

Este proceso es completamente manual y genera:

- Retrasos en la promesa de entrega.
- Reprocesos administrativos.
- Riesgo contable por doble facturación y notas crédito.
- Impacto negativo en la experiencia del cliente.

3.1.3. Análisis estructural de la arquitectura actual

El análisis consolidado permitió identificar los siguientes patrones arquitectónicos dominantes:

- Integraciones punto a punto entre ERP, POS, WMS y Comercio Digital.
- Predominio de sincronización batch sobre mecanismos reactivos.
- Uso de snapshots completos en lugar de propagación incremental de cambios.

- Ausencia de un modelo formal de gestión de datos maestros (MDM).
- Limitada observabilidad y trazabilidad transaccional extremo a extremo.

La arquitectura existente corresponde a un patrón de integración síncrona punto a punto con sincronización batch centralizada en el ERP, característico de soluciones orientadas a la consolidación transaccional periódica pero no a la propagación de cambios en tiempo cercano al real. Este patrón resuelve los requerimientos del modelo de operación tradicional (cierre diario, conciliación de cuentas, reportes batch) pero no soporta los requerimientos de consistencia omnicanal ni de baja latencia que demanda la estrategia de negocio actual.

Estos hallazgos definen la línea base arquitectónica sobre la cual se plantea la evolución hacia una Arquitectura Orientada a Eventos (EDA), complementada con un modelo formal de Gobierno y Gestión de Datos Maestros (MDM).

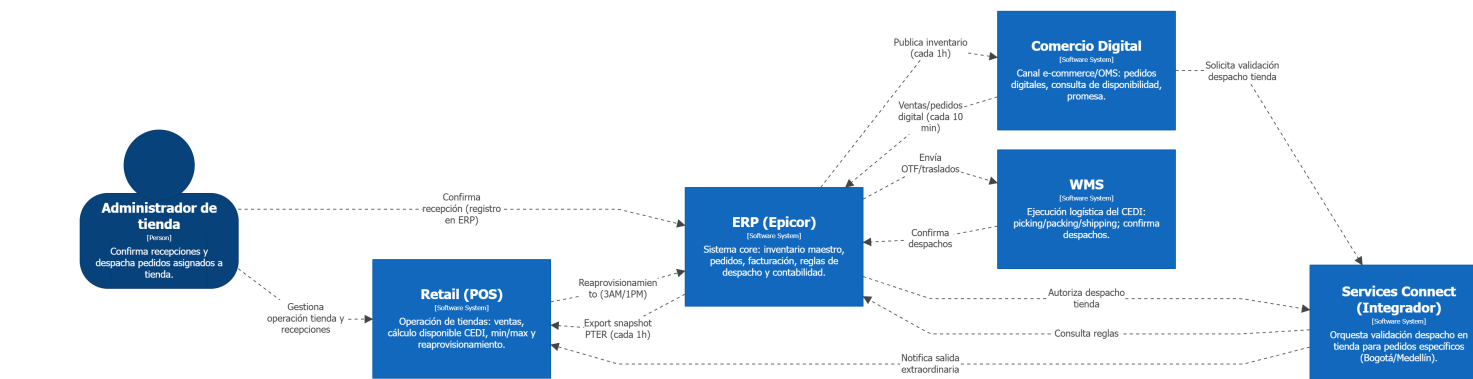
3.1.4. Modelado estructural mediante el modelo C4

Para representar formalmente la arquitectura actual y facilitar su análisis estructural, se aplicó el modelo C4 descrito en la Sección 2.1.6, en sus niveles de Contexto (Nivel 1) y Contenedores (Nivel 2). El Nivel 1 permite identificar el rol del ERP como núcleo transaccional y las dependencias de integración con los sistemas satélite; el Nivel 2 expone los mecanismos internos de sincronización (procesos batch, bases de datos independientes, integraciones punto a punto) que sostienen la topología actual y son fuente directa de las brechas identificadas en el diagnóstico.

3.1.4.1. Vista de Contexto – Nivel 1

La Figura 3.1 presenta la vista de contexto de la arquitectura actual. En ella se evidencia el rol central del ERP como núcleo transaccional, actuando como sistema de registro para inventarios, pedidos y facturación.

También se identifican integraciones punto a punto con los sistemas satélite (Retail POS, Comercio Digital, WMS y el integrador Services Connect), lo que confirma una topología centralizada y dependiente del ERP.



System Context View: ERP (Epicor)
 C4 Context - Arquitectura Actual
 Monday, March 2, 2026 at 10:54 PM Colombia Standard Time

Figura 3.1: Vista de Contexto C4 – Arquitectura Actual

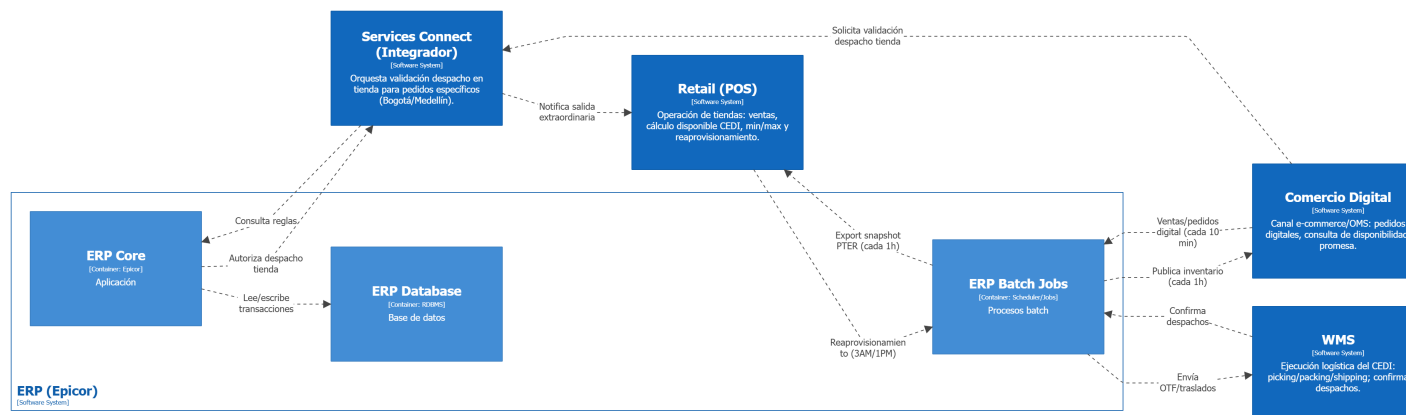
La vista de contexto permite observar que cualquier variación transaccional relevante debe atravesar el ERP, configurándolo como punto crítico de coordinación y posible cuello de botella arquitectónico.

3.1.4.2. Vista de Contenedores – Nivel 2

Con el fin de profundizar en la estructura interna de los sistemas identificados en la vista de contexto, se presentan a continuación las vistas de contenedores correspondientes.

Estas vistas permiten evidenciar la existencia de procesos batch, bases de datos independientes y mecanismos de integración específicos por sistema.

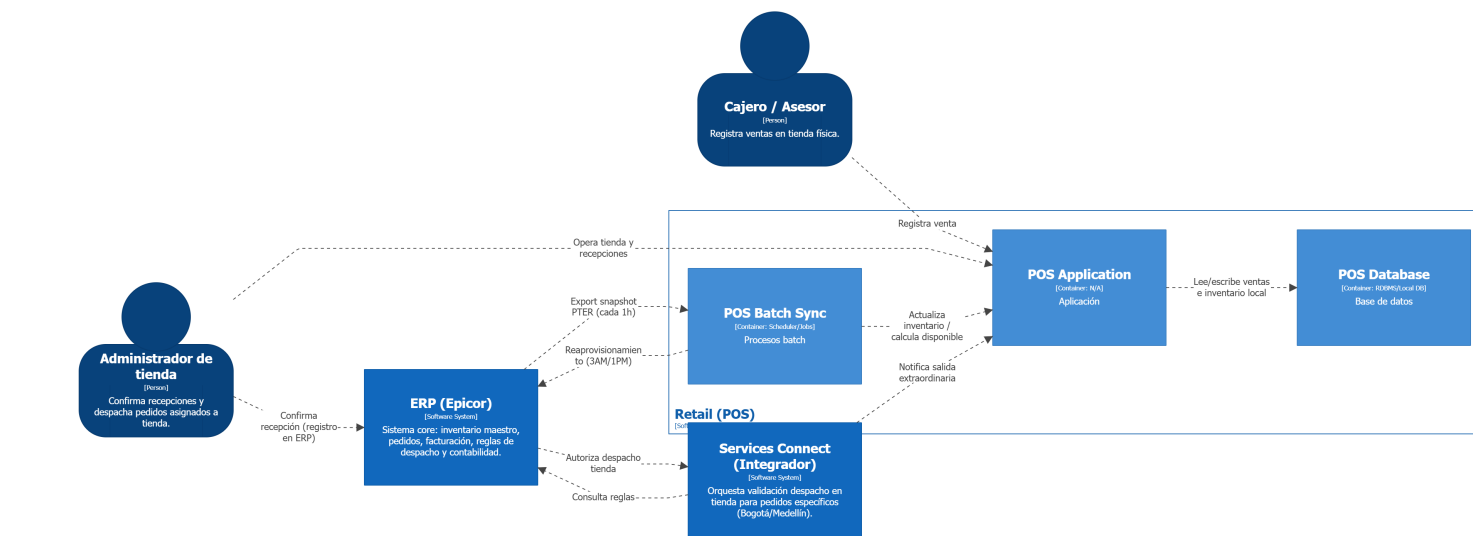
Contenedores del ERP (Epicor) La Figura 3.2 detalla la composición interna del ERP, donde se identifican claramente los procesos batch responsables de la sincronización periódica de inventarios y ventas, así como la base de datos transaccional central.



Container View: ERP (Epicor)
 C4 Containers - ERP
 Monday, March 2, 2026 at 10:54 PM Colombia Standard Time

Figura 3.2: Vista de Contenedores C4 – ERP (Epicor)

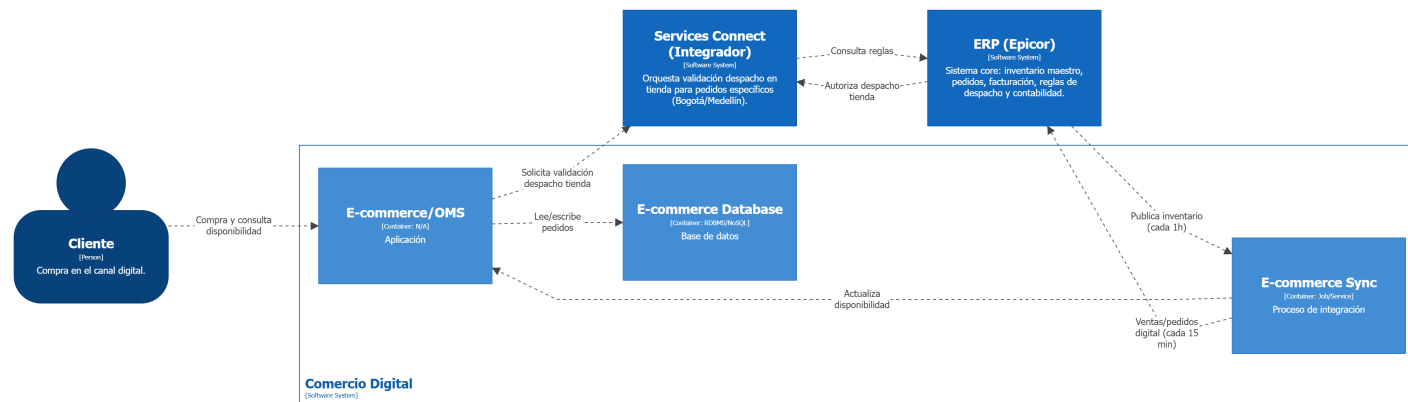
Contenedores del Sistema Retail (POS) La Figura 3.3 muestra que el sistema POS opera con base de datos propia y procesos de sincronización periódica hacia el ERP, evidenciando dependencia de mecanismos batch para la consolidación de ventas e inventario.



Container View: Retail (POS)
 C4 Containers - POS
 Monday, March 2, 2026 at 10:54 PM Colombia Standard Time

Figura 3.3: Vista de Contenedores C4 – Sistema Retail (POS)

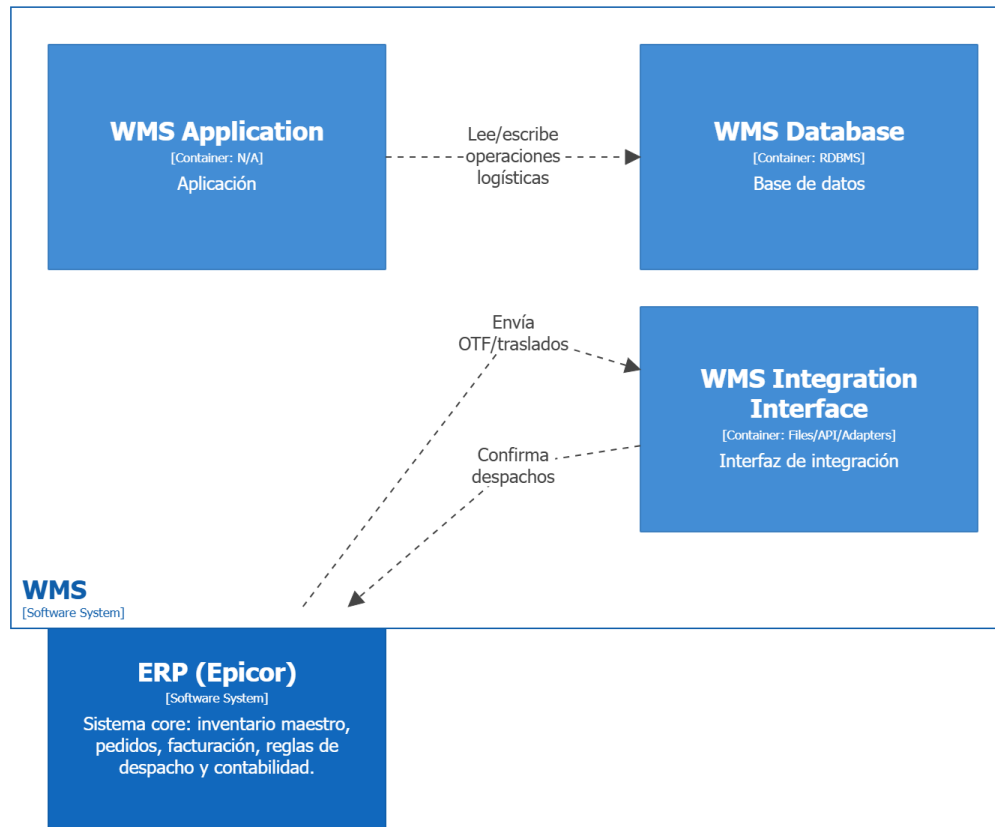
Contenedores del Comercio Digital La Figura 3.4 evidencia que la plataforma digital mantiene su propia base de datos y un proceso de integración que sincroniza ventas y pedidos cada 15 minutos con el ERP, mientras recibe el inventario publicado por el ERP cada hora, generando una latencia menor frente al canal físico.



Container View: Comercio Digital
 C4 Containers - E-commerce
 Saturday, July 11, 2026 at 12:21 PM Colombia Standard Time

Figura 3.4: Vista de Contenedores C4 – Comercio Digital

Contenedores del WMS La Figura 3.5 presenta el sistema de gestión logística, donde se observa la ejecución operativa del despacho y la posterior confirmación hacia el ERP, también bajo un modelo secuencial no reactivo.



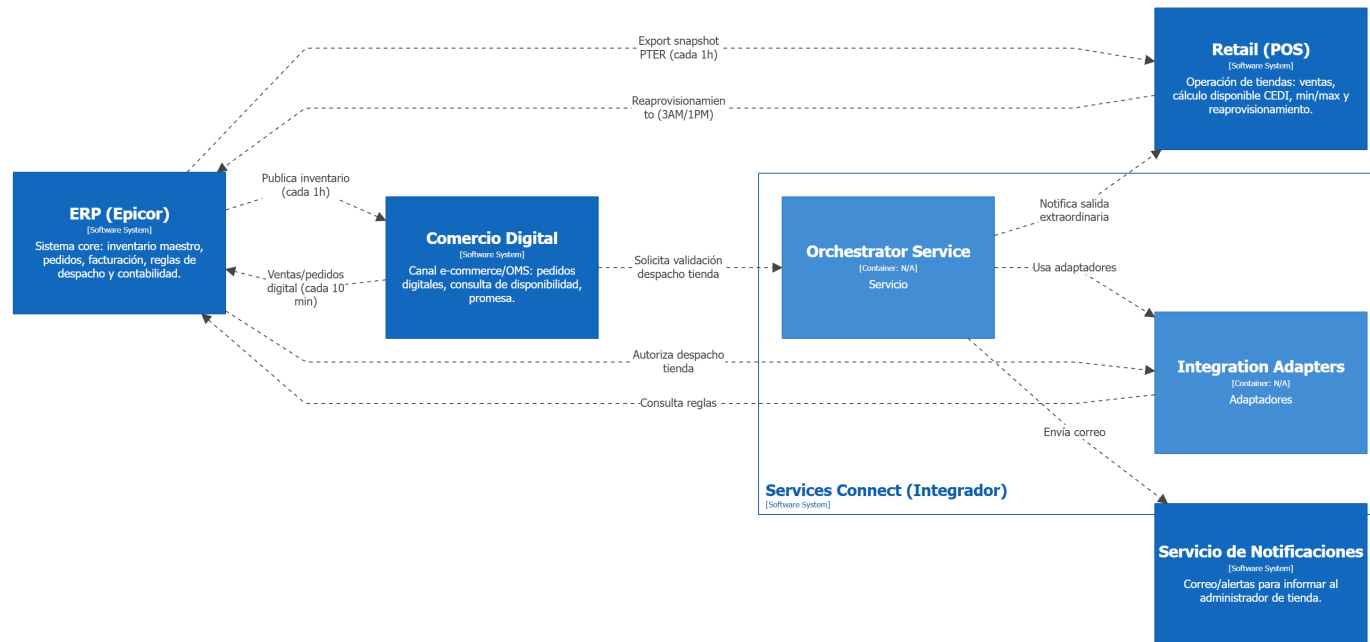
Container View: WMS

C4 Containers - WMS

Monday, March 2, 2026 at 10:54 PM Colombia Standard Time

Figura 3.5: Vista de Contenedores C4 – WMS

Contenedores del Integrador Services Connect La Figura 3.6 muestra el integrador encargado de orquestar la validación de despacho en tienda para casos específicos (Bogotá y Medellín), evidenciando lógica distribuida fuera del ERP pero dependiente de sus reglas de negocio.



Container View: Services Connect (Integrator)
C4 Containers - Services Connect
Monday, March 2, 2026 at 10:54 PM Colombia Standard Time

Figura 3.6: Vista de Contenedores C4 – Services Connect

El modelado estructural confirmó que la arquitectura actual presenta un patrón de integración fuertemente centralizado en el ERP, con predominio de sincronización batch y ausencia de mecanismos reactivos basados en eventos de negocio.

La existencia de bases de datos independientes por sistema, procesos de integración periódicos y lógica distribuida en componentes auxiliares incrementa el acoplamiento estructural y dificulta la consistencia omnicanal en tiempo cercano al real.

Estas evidencias consolidan la necesidad de evolucionar hacia un modelo arquitectónico desacoplado, orientado a eventos y soportado por un esquema formal de gestión de datos maestros (MDM).

3.1.5. Análisis consolidado de entrevistas a stakeholders

Con el propósito de complementar el diagnóstico documental y estructural presentado en las subsecciones anteriores, se diseñó y aplicó un instrumento de entrevista semiestructurada dirigido a cinco stakeholders clave de la organización. La selección de los participantes respondió al criterio de representatividad funcional: se buscó cubrir las áreas de Finanzas, Comercio Digital, Logística y Tecnología, garantizando una perspectiva multidimensional sobre el impacto de la arquitectura de integración actual. Las entrevistas fueron diligenciadas de forma autónoma por cada participante durante la última semana de abril de 2026.

3.1.5.1. Participantes del estudio

La Tabla 3.1 presenta los cinco participantes, sus roles y el área funcional que representan en la organización.

Tabla 3.1: Participantes del instrumento de entrevista de la Fase I.

Nombre	Cargo	Área	Fecha
Andrés Duque	Jefe Financiero de Canales Propios	Finanzas	22/04/2026
Julián A. Llanos Santofimio	Gerente de Negocios Digitales	Digital	23/04/2026
Paola A. Collazos Ocampo	Jefe de Almacén	Logística	24/04/2026
Pedro L. Dorado Ramírez	Jefe de Infraestructura T.I.	Tecnología	22/04/2026
Marcela Chavarro Ortiz	Analista de Sistemas	Tecnología	21/04/2026

3.1.5.2. Hallazgos cuantitativos: línea base del problema

El análisis de las respuestas permitió recolectar los indicadores cuantitativos que conforman la línea base objetiva del problema. Estos valores se utilizan luego como referencia para medir el impacto de la solución propuesta. La Tabla 3.2 consolida las métricas identificadas.

Tabla 3.2: Indicadores cuantitativos del diagnóstico (línea base).

Indicador	Valor	Fuente
Pedidos con fallo por inventario no actualizado (e-commerce)	20 % (2 de cada 10)	Llanos
Eventos críticos de inventario en e-commerce (mensual)	25 eventos	Duque
Eventos críticos de inventario en tiendas (mensual)	10 eventos	Duque
Ajustes contables mensuales por inconsistencia ERP vs. físico	12 ajustes (≈ 20 líneas ítem-tienda)	Duque
RMAs procesadas en marzo de 2026	28 (10 por tiempo; 18 por cambio de bodega)	Duque
Ticket promedio por devolución (RMA)	COP \$330.000 sin IVA	Duque
Frecuencia de discrepancias inventario virtual vs. físico	Diariamente	Collazos
Capacidad de crecimiento de la infraestructura actual	30 % (objetivo requerido: 50 %)	Dorado
Tiempo para integrar un nuevo canal de venta	4 a 8 semanas	Dorado
Margen de error de inventario tolerable en el canal digital	± 5 %	Llanos

Complementariamente, la Tabla 3.3 presenta el mapa técnico de frecuencias de sincronización entre sistemas, construido a partir de la entrevista con la Analista de Sistemas, quien proporcionó el nivel de detalle más granular sobre los mecanismos de integración vigentes.

El WMS Cubic es el único sistema que opera con integración orientada a eventos mediante *hooks*. Este precedente demuestra que el ecosistema tecnológico ya soporta mecanismos event-driven y, por lo tanto, reduce el riesgo técnico de la adopción generalizada del modelo propuesto en la Fase II.

Tabla 3.3: Mapa técnico de integraciones y frecuencias de sincronización.

Sistema origen	Sistema destino	Mecanismo	Frecuencia	Riesgo
VTEX / Mercado-Libre	Epicor ERP	API REST vía Service Connect	15 min	Medio
POS Cegid (tiendas)	Epicor ERP	Archivos TXT	2 horas	Alto
Bylmo / Falabella / Sodimac	Epicor ERP	Archivos Excel manuales	Diaria	Crítico
Epicor ERP	Todos los canales	Batch (snapshot completo)	1 a 2 horas	Alto
WMS Cubic	Epicor ERP	Hooks (eventos)	Tiempo real	Bajo
Reaprovisionamiento POS	CEDI / ERP	Batch programado	3 AM y 1 PM	Alto

3.1.5.3. Hallazgos convergentes por triangulación entre fuentes

Los siguientes hallazgos fueron identificados de manera independiente por múltiples stakeholders de distintas áreas funcionales, lo que fortalece su validez y reduce el riesgo de sesgo por perspectiva individual.

H1. El ERP actúa como fuente de verdad de facto, sin MDM formal diseñado. El ERP Epicor cumple el rol de repositorio de datos maestros por ausencia de alternativa, no como resultado de una decisión arquitectónica. Duque (Finanzas) lo expresa: *“Como se crea la orden en el ERP es como se gobierna la información del cliente”*. Collazos (Logística) confirma: *“El ERP es el mandante en el inventario”*. Dorado (TI) señala que el criterio de prevalencia del ERP existe pero no está documentado. La convergencia de tres áreas funcionales independientes sobre el mismo vacío evidencia la falta de un Modelo de Datos Maestros (MDM) formal que defina reglas canónicas, atributos de calidad y criterios de resolución de conflictos entre sistemas.

H2. Ausencia de Data Owner formal en los dominios críticos del negocio. Ninguno de los tres dominios del proyecto (Inventario, Pedido, Cliente) tiene un responsable formal documentado. En Finanzas, la gobernanza del dato depende del proceso de creación de la orden; en Logística, la responsabilidad está distribuida entre el CEDI y los administradores de tienda; en Tecnología no existe documentación del criterio de prevalencia. La toma de decisiones se basa en convención tácita, sin respaldo formal, lo que aumenta el riesgo de divergencia en situaciones no previstas.

H3. Las discrepancias de inventario son la norma operativa, no la excepción. Collazos confirma discrepancias diarias entre el inventario virtual del ERP y el inventario físico en bodegas. Duque las cuantifica en 35 eventos críticos mensuales. Llanos reporta que el 20 % de los pedidos del canal digital presenta fallos atribuibles a inventario desactualizado. Chavarro explica el

mecanismo estructural: el mismo inventario está expuesto simultáneamente a todos los canales B2C sin descuento en tiempo real, lo que produce **sobreventa sistémica**. Este hallazgo es el argumento técnico central para la adopción de una arquitectura orientada a eventos.

H4. La arquitectura de integración es batch-first, sin mecanismos de eventos ni recuperación automática. Dorado confirma que la compañía no cuenta con ninguna herramienta de mensajería asíncrona ni arquitectura event-driven. La integración es mayormente síncrona y mediante procesos batch. Chavarro detalla los fallos recurrentes: tokens OAuth que expiran sin renovación automática, *timeouts* sin reintento y archivos no recibidos sin alerta. En todos los casos el mecanismo de recuperación es manual. La falta de *circuit breakers* o reintentos automáticos convierte cualquier fallo de integración en un incidente silencioso que se detecta únicamente cuando el daño operativo ya se ha materializado.

H5. Las reglas de validación existen en código pero no están documentadas ni centralizadas. Chavarro revela que existen validaciones implementadas en Epicor y Service Connect para la consistencia de clientes, órdenes de venta y datos de facturación; sin embargo, este conocimiento está distribuido entre responsables técnicos sin documentación centralizada. Collazos confirma lo análogo en logística: existen prácticas de conciliación (descuento de ubicaciones virtuales, separación de inventario disponible versus programado) pero sin reglas formalizadas. Esto genera dependencia de personas clave y dificulta la evolución controlada de las integraciones.

H6. Duplicación de registros de clientes sin proceso de deduplicación automatizado. Llanos señala que la duplicación de clientes es frecuente, originada en patrones de integración distintos por sistema. Chavarro confirma el mecanismo: un mismo cliente puede existir como dos registros independientes si realizó una compra en tienda física y otra en el canal digital. No existe un proceso automatizado de deduplicación ni limpieza periódica; la corrección es manual y reactiva. Este hallazgo refuerza directamente la necesidad del MDM como mecanismo de consolidación de registros maestros.

H7. Dos brechas críticas de visibilidad en el ciclo pedido-entrega. Collazos identifica con precisión los dos puntos ciegos de la operación. El primero es el alistamiento en tienda, donde no existe ningún estado del sistema que indique si un pedido está en proceso o completado. El segundo es la última milla, donde no hay trazabilidad después del despacho. Las herramientas actuales de seguimiento son Excel y Epicor. Esta brecha impacta directamente la promesa de entrega y la experiencia del cliente, y valida la necesidad de observabilidad como atributo transversal de la arquitectura propuesta.

3.1.5.4. Síntesis por perspectiva funcional

Finanzas (Andrés Duque). El impacto financiero de la latencia de datos se expresa en costos directos cuantificables: 28 RMAs en marzo de 2026, con un ticket promedio de COP \$330.000, generando notas crédito, refacturación y reprocesos que involucran logística, tesorería, contabilidad y servicio al cliente. El costo oculto más significativo identificado es el retracto completo: devolución de venta, costos de logística inversa, tiempo de atención y pérdida reputacional de la marca. La fuente de verdad financiera es el ERP, pero esta dependencia no es por diseño sino por ausencia

de alternativa consolidada. Duque sugiere extender la solución con lógica de asignación dinámica de despacho, promesa de entrega ajustada por CEDI y reportería que cruce la fecha de la orden original con la fecha de entrega efectiva según paquetera.

Comercio Digital (Julián Llanos). El canal digital es el más sensible a la latencia de datos: el 20 % de los pedidos presenta fallos por inventario desactualizado. La arquitectura de canal es heterogénea: los canales dentro del ecosistema VTEX reciben actualizaciones de inventario en tiempo real, mientras que los canales externos (Mercado Libre, Falabella, Homecenter, VirtualMuebles, Bylmo) operan con actualización manual. Esta asimetría tecnológica genera una desigualdad estructural en la calidad del dato entre canales. La base de datos de clientes es propia (VTEX Master Data) y se sincroniza únicamente cuando se genera una nueva orden de compra.

Logística (Paola Collazos). La manifestación más clara del problema desde la operación de almacén es la ocurrencia diaria de discrepancias entre el inventario virtual del ERP y el inventario físico en bodegas. No existe un procedimiento formal para resolver estas discrepancias: el mecanismo vigente es un reporte conciliatorio periódico y validación manual. La responsabilidad del dato de inventario está distribuida sin un Data Owner unificado, lo que hace que la conciliación dependa del criterio individual de cada responsable operativo.

Infraestructura TI (Pedro Dorado). La confirmación técnica más relevante es la ausencia total de herramientas de mensajería asíncrona o arquitectura orientada a eventos: la integración es mayormente síncrona y batch, sin bus de servicios ni capa de integración centralizada. La infraestructura actual soporta un crecimiento del 30 % en transacciones concurrentes, sin alcanzar el 50 % requerido para escalar la operación omnicanal. Una caída de 15 minutos en el sistema de integración durante un día de alta demanda genera pérdida de transacciones sin mecanismo de compensación.

Sistemas (Marcela Chavarro). La entrevista técnica más detallada confirma que Service Connect es el orquestador central del flujo de órdenes: VTEX y Mercado Libre por API cada 15 minutos; POS Cegid por archivos TXT cada 2 horas; canales externos como Bylmo, Falabella y Sodimac por archivos Excel de forma diaria y manual. Los fallos recurrentes incluyen tokens OAuth sin renovación automática, archivos no recibidos sin alerta y *timeouts* sin reintento. El cambio técnico que Chavarro considera prioritario coincide con la propuesta central de este proyecto:

“Modificaría la arquitectura actual para migrar de un modelo batch a un modelo orientado a eventos, mediante el uso de webhooks o mensajería asíncrona. Esto permitiría la actualización en tiempo real del inventario y los pedidos entre Epicor, WMS Cubic, POS Cegid y los canales e-commerce, reduciendo la sobreventa y mejorando la consistencia de la información.”

Marcela Chavarro Ortiz, Analista de Sistemas.

Este alineamiento entre el diagnóstico técnico interno y la solución propuesta en la investigación fortalece la justificación de la Fase II y reduce el riesgo de adopción organizacional.

3.1.5.5. Análisis de brechas AS-IS versus requerimientos omnicanal

La Tabla 3.4 contrasta el estado actual identificado en el diagnóstico con los requerimientos que una arquitectura omnicanal eficiente debe cumplir, y mapea cada brecha con el componente de solución correspondiente en la Fase II.

Tabla 3.4: Análisis de brechas: estado actual (AS-IS) vs. requerimientos omnicanal (TO-BE).

Dimensión	Estado actual (AS-IS)	Requerimiento (TO-BE)	Componente de solución
Sincronización de inventario	Batch periódico (15 min a 2 h). Canales externos con actualización diaria manual.	Propagación incremental en tiempo cercano al real, orientada a eventos.	Arquitectura EDA y broker de mensajes.
Modelo de datos maestros	Cada sistema mantiene su propia copia. ERP como MDM de facto sin diseño formal.	Repositorio MDM unificado como fuente única de verdad (<i>Golden Record</i>).	MDM con dominios Cliente, Inventario y Pedido.
Gobernanza de datos	Sin Data Owner documentado. Reglas de validación implícitas, sin centralizar.	Roles formales (Data Owner, Data Steward). Políticas y SLAs definidos.	Framework de gobernanza de datos.
Integración de canales	Punto a punto. De 4 a 8 semanas para un nuevo canal. Canales externos manuales.	Capa de integración desacoplada. Nuevo canal en días, sin cambios en sistemas fuente.	API Gateway y mensajería asíncrona.
Trazabilidad transaccional	Parcial en ERP. Sin visibilidad de alistamiento ni última milla.	Trazabilidad extremo a extremo: creación, alistamiento, despacho y entrega.	Event Log centralizado y observabilidad.
Recuperación ante fallos	Manual. Tokens expirados sin alerta. Sin reintentos automáticos.	Circuit breaker, reintentos automáticos y Dead Letter Queue.	Infraestructura EDA con resiliencia.

Tabla 3.4 (Continuación)

Dimensión	Estado actual (AS-IS)	Requerimiento (TO-BE)	Componente de solución
Escalabilidad	Soporta 30 % de crecimiento. Cuellos en procesos batch y base de datos del ERP.	Soporte a 50 % o más de crecimiento transaccional sin degradación.	Arquitectura desacoplada e infraestructura elástica.
Seguridad del dato	HTTPS en tránsito. Controles de acceso heterogéneos por sistema. Sin cifrado en reposo estandarizado.	Control de acceso por dominio y protección de datos personales (PII). Cifrado en reposo y secretos como evolución posterior.	Política de acceso por dominio en el framework de gobernanza y autorización vía API Gateway; cifrado en reposo/secretos como trabajo futuro.

3.1.5.6. Conclusiones del diagnóstico

El análisis consolidado de las cinco entrevistas, triangulado con la revisión documental y el modelado C4 presentados en las subsecciones anteriores, permite establecer las siguientes conclusiones que sustentarán el diseño de la solución en la Fase II:

1. **El problema está confirmado y cuantificado.** Las discrepancias de inventario generan 35 eventos críticos mensuales, 28 RMAs y un 20 % de pedidos fallidos en el canal digital. Estos datos definen la línea base contra la cual se medirá el impacto del prototipo en la Fase III (Capítulo 4).
2. **La arquitectura actual es estructuralmente incompatible con la estrategia omnicanal.** La combinación de sincronización batch, integraciones punto a punto, ausencia de EDA y límite de escalabilidad del 30 % hace que la arquitectura vigente no pueda soportar los requerimientos de una operación omnicanal confiable sin una intervención arquitectónica de fondo.
3. **El ERP como MDM de facto es el síntoma central de la deuda de gobernanza.** Sin un MDM formal y sin Data Owners definidos, cualquier solución técnica de integración continuará produciendo inconsistencias, porque el problema no es únicamente de latencia sino de ausencia de una fuente canónica de datos con reglas de calidad documentadas.
4. **El WMS Cubic valida la viabilidad técnica del EDA en esta organización.** La integración por hooks entre WMS Cubic y Epicor demuestra que el ecosistema tecnológico

ya soporta mecanismos orientados a eventos, reduciendo el riesgo técnico de la propuesta y acelerando la curva de adopción del modelo EDA en otros dominios.

5. **Los stakeholders técnicos demandan la misma solución que propone esta investigación.** La Analista de Sistemas identifica explícitamente la migración a EDA como el cambio prioritario; el Jefe de Infraestructura confirma la necesidad de MDM y gobernanza. Este alineamiento interno es un factor crítico de éxito para la viabilidad organizacional de la solución.

Estos hallazgos cierran formalmente la Fase I del proyecto y establecen la trazabilidad entre el problema diagnosticado y los componentes de la arquitectura propuesta en la siguiente sección.

3.1.6. Análisis de indicadores de desempeño: línea base diagnóstica

Con el propósito de cuantificar el impacto operativo y financiero de las brechas identificadas, se construyó una línea base de indicadores clave de desempeño (KPI) a partir de los datos recogidos mediante las entrevistas semiestructuradas. Siguiendo el principio de triangulación metodológica, los valores de cada indicador se verificaron con al menos dos fuentes independientes antes de ser incorporados como referencia. Esta línea base cumple dos funciones: (1) dimensionar el problema con evidencia cuantitativa para justificar la inversión arquitectónica ante el jurado, y (2) establecer los valores de referencia contra los cuales se medirá el impacto del prototipo en la Fase III (Capítulo 4).

Se definieron tres KPIs que cubren las dimensiones operativa, técnica y financiera del problema documentado. La Tabla 3.5 consolida las definiciones formales, fórmulas, valores actuales y objetivos de cada indicador.

Tabla 3.5: Indicadores clave de desempeño (KPI). Línea base diagnóstica.

KPI	Fórmula	Valor actual	ac-	Objetivo TO-BE	Fuente
Tasa de Pedidos Fallidos por Inconsistencia (TPFI)	$TPFI = \frac{P_{fallidos}}{P_{total}} \times 100$	20 %		<5 %	Llanos
Exactitud de Inventario (EI)	$EI = \frac{1 - \frac{\sum V_i - F_i }{\sum F_i}}{100}$	<90 % (est.)		≥ 95 %	Collazos / Duque

Tabla 3.5 (Continuación)

KPI	Fórmula	Valor actual	ac-	Objetivo TO-BE	Fuente
Costo Operacional Mensual por Inconsistencia (COMI)	$\text{COMI} = N_{RMA} \times (\bar{U}_{RMA} \times 50\%) + C_{ajuste} \times N_{RMA}$	COP \$3,02M		<COP \$1,51M	Duque

3.1.6.1. KPI 1. Tasa de Pedidos Fallidos por Inconsistencia (TPFI)

Definición. Porcentaje de pedidos procesados en el canal digital que no pueden cumplirse por causa de inventario desactualizado o inexacto en el momento de la asignación de bodega o tienda.

Cálculo de la línea base. Llanos (Gerente de Negocios Digitales) reporta que 2 de cada 10 pedidos del canal digital presentan fallo atribuible a inventario desactualizado:

$$\text{TPFI}_{AS-IS} = \frac{2}{10} \times 100 = \mathbf{20\%} \quad (3.1)$$

Este valor se confirma estructuralmente mediante los hallazgos H3 y H4: el canal digital consulta el inventario con latencia de hasta 15 minutos (VTEX vía Service Connect), pero durante ese intervalo el mismo stock puede ser comprometido simultáneamente en tiendas físicas que sincronizan cada 2 horas, generando sobreventa sistémica.

Valor objetivo. El umbral <5 % se deriva del margen de error tolerable de $\pm 5\%$ definido por Llanos para el canal digital. La adopción de EDA con propagación en tiempo cercano al real y reserva de inventario en el momento del evento `OrderPlaced` elimina la causa raíz de la sobreventa, haciendo técnicamente alcanzable este umbral.

Método de medición en TO-BE. Ratio entre eventos `StockUnavailable` y `OrderPlaced` registrados en el Event Store de Kafka durante un período de 30 días. La medición es automatizable sin intervención manual.

3.1.6.2. KPI 2. Exactitud de Inventario (EI)

Definición. Porcentaje de concordancia entre el stock virtual registrado en el ERP y el stock físico verificado en bodega o tienda, calculado sobre el total del inventario activo. La fórmula pondera la discrepancia absoluta de cada SKU (i) respecto al stock físico real (F_i), donde V_i es el stock virtual:

$$\text{EI} = \left(1 - \frac{\sum_{i=1}^n |V_i - F_i|}{\sum_{i=1}^n F_i} \right) \times 100 \quad (3.2)$$

Estimación de la línea base. No se dispone de un conteo físico sistematizado como referencia directa; sin embargo, los indicadores recopilados en las entrevistas permiten construir una estimación razonada:

- Collazos confirma discrepancias entre inventario virtual y físico con frecuencia **diaria**.
- Duque documenta 12 ajustes contables mensuales que involucran aproximadamente 20 líneas ítem-tienda cada uno, para un total de **240 correcciones de inventario al mes**.
- Duque cuantifica 35 eventos críticos mensuales (25 en e-commerce + 10 en tiendas) atribuibles directamente a inconsistencias de inventario.
- Dorado confirma que el mecanismo de sincronización es batch y no existen alertas automáticas de desviación.

Considerando que las discrepancias ocurren diariamente y generan 240 correcciones mensuales, se estima que la Exactitud de Inventario se encuentra **por debajo del 90 %**, valor coherente con la literatura que sitúa la exactitud promedio de inventario en empresas sin MDM formal entre el 65 % y el 85 % (Loshin, 2009). Esta estimación se utilizará como límite inferior conservador para la evaluación del prototipo.

Valor objetivo. $\geq 95\%$, alineado con el SLA de calidad del dato definido en la Tabla 3.8. Este umbral corresponde al margen de error tolerable del $\pm 5\%$ declarado por Llanos para el canal digital.

Método de medición en TO-BE. Comparación periódica (semanal) entre **stockDisponible** del *Golden Record* MDM y el conteo físico del WMS Cubic. La integración EDA garantiza que el MDM recibe todos los movimientos físicos en tiempo real, por lo que la discrepancia esperada se reduce al tiempo de procesamiento del evento (< 1 minuto).

3.1.6.3. KPI 3. Costo Operacional Mensual por Inconsistencia (COMI)

Definición. Costo financiero mensual mínimo cuantificable atribuible a la inconsistencia de datos de inventario. Se distinguen dos componentes: la pérdida de margen no recuperable en las devoluciones (RMAs) y el costo de reproceso administrativo que cada devolución genera. La métrica cuantifica la pérdida económica real (margen), no la facturación bruta, para constituir un límite inferior financieramente defendible.

Cálculo de la línea base. A partir de los datos reportados por Duque (Jefe Financiero de Canales Propios) para el mes de marzo de 2026:

$$\text{COMI}_{AS-IS} = N_{RMA} \times (\bar{U}_{RMA} \times 50\%) + C_{ajuste} \times N_{RMA} \quad (3.3)$$

donde:

- $N_{RMA} = 28$ devoluciones en marzo de 2026.
- $\bar{U}_{RMA} = \text{COP } \$115,500$: utilidad bruta promedio por devolución, estimada como el 35 % del ticket promedio de COP \$330.000. El margen bruto del 35 % corresponde a un valor conservador dentro del rango típico (30 %–40 %) del sector de consumo masivo y constituye un supuesto a validar con la contabilidad de la organización.

- El factor **50 %** representa la fracción del margen *no recuperable* por devolución: la mercancía retorna y se reprocesa, pero con costo adicional de manipulación y riesgo de merma, de modo que solo se recupera aproximadamente la mitad del margen.
- C_{ajuste} = COP \$50,000 por devolución: costo de oportunidad del reproceso administrativo que cada RMA dispara (emisión de nota crédito, refacturación, coordinación entre logística, tesorería, contabilidad y servicio al cliente).

$$COMI_{AS-IS} = 28 \times (\$115,500 \times 50\%) + \$50,000 \times 28 = \$1,617,000 + \$1,400,000 = \text{COP } \$3,017,000/\text{mes} \quad (3.4)$$

Este valor representa el **costo directo mínimo confirmado**. Duque señala explícitamente que el costo real es significativamente mayor al incluir el retracto completo: costo logístico de la logística inversa, tiempo de atención al cliente y pérdida reputacional de la marca, componentes no incluidos en este cálculo por no ser directamente cuantificables con la información disponible. La cifra calculada (COP \$3,02M/mes) debe interpretarse, por tanto, como límite inferior del costo real mensual atribuible a la inconsistencia de datos.

Valor objetivo. Reducción $\geq 50\%$ respecto a la línea base: <COP \$1,51M/mes. Esta meta es conservadora: la eliminación de la sobreventa sistémica mediante EDA y MDM debería reducir los fallos de inventario (causa del 64 % de las RMAs según Duque: 18 de 28 RMAs en marzo fueron por cambio de bodega atribuible a inconsistencia) en más del 70 %.

Método de medición en TO-BE. Reporte financiero mensual de RMAs cruzado con la causa raíz registrada en el Event Store. La trazabilidad completa del evento `StockUnavailable` → `OrderCancelled` permite distinguir automáticamente las devoluciones atribuibles a inconsistencia de datos de las originadas en otras causas.

3.1.6.4. Síntesis de la línea base y criterios de evaluación

La Tabla 3.6 resume los tres KPIs con sus valores de línea base, valores objetivo y el criterio de aceptación que se aplicará al prototipo en el Capítulo 4. Un incremento ≥ 10 puntos porcentuales en EI o una reducción $\geq 50\%$ en COMI en el entorno del prototipo se considerará evidencia suficiente de viabilidad de la arquitectura propuesta.

3.2. Fase II: Diseño de la arquitectura de integración omnicanal

Los hallazgos del diagnóstico establecen ocho brechas estructurales que la arquitectura propuesta debe resolver (Tabla 3.4). En respuesta, el diseño se articula en tres capas complementarias y mutuamente dependientes: (1) un **framework de gobernanza de datos**, que formaliza los roles, políticas y responsabilidades sobre las entidades de negocio críticas; (2) un **modelo de datos maestros (MDM)**, que consolida el *Golden Record* para los dominios de Cliente, Inventario y Pedido; y (3) una **arquitectura orientada a eventos (EDA)**, que reemplaza los mecanismos

Tabla 3.6: Criterios de evaluación del prototipo basados en la línea base de KPIs.

KPI	Línea base (AS-IS)	Objetivo (TO-BE)	Criterio de aceptación (Capítulo 4)
TPFI	20 %	<5 %	Demostrar que la reserva de inventario en <code>OrderPlaced</code> previene la sobreventa en al menos 3 escenarios de prueba concurrente.
EI	<90 % (est.)	≥95 %	El MDM <i>Golden Record</i> refleja el estado del WMS en <1 min tras <code>InventoryMoved</code> en 5 pruebas consecutivas.
COMI	COP \$3,02M/mes	<COP \$1,51M/mes	Evidencia de que los flujos EDA eliminan la causa raíz de ≥50 % de los eventos que generan RMA.

de sincronización batch por propagación incremental en tiempo cercano al real. Esta estructura responde directamente a los objetivos específicos OE2 (gobernanza de datos), OE3 (modelo MDM) y OE4 (arquitectura de integración EDA) del proyecto, y materializa los tres planos del marco de omnicanalidad descrito en la Sección 2.1.1: el plano de negocio se expresa en las políticas de gobernanza, el plano de información en el modelo MDM, y el plano tecnológico en la arquitectura EDA.

3.2.1. Principios de diseño

El diseño de la arquitectura TO-BE se fundamenta en cinco principios, derivados de la evidencia recogida en la Fase I:

- P1. Desacoplamiento mediante mensajería asíncrona.** Los sistemas productores y consumidores no deben conocerse directamente. Toda comunicación entre dominios transita por un *broker* de mensajes, eliminando las integraciones punto a punto que generan acoplamiento estructural alto y tiempos de incorporación de nuevos canales de cuatro a ocho semanas (H4).
- P2. Fuente única de verdad por dominio (MDM).** Cada entidad crítica de negocio (Cliente, Inventario, Pedido) cuenta con un *Golden Record* canónico gestionado por el MDM Service. El ERP deja de ser el MDM *de facto* para convertirse en uno más de los sistemas que contribuye y consume datos maestros (H1, H5).

- P3. Propagación incremental de cambios.** Los sistemas consumidores reciben únicamente el delta de cambio, no un *snapshot* completo del estado. Esto elimina la latencia acumulada del modelo batch (entre 15 minutos y 2 horas según la brecha de sincronización identificada) y la causa raíz del 20 % de pedidos fallidos en el canal digital (H3).
- P4. Observabilidad transversal.** Todo evento relevante del ciclo pedido-entrega queda registrado en un *Event Log* inmutable. La trazabilidad es un atributo de arquitectura, no una funcionalidad opcional añadida *a posteriori* (H7).
- P5. Resiliencia ante fallos parciales.** La indisponibilidad de un consumidor no bloquea el procesamiento en los demás. El *broker* retiene los mensajes y garantiza la entrega cuando el consumidor se recupera, eliminando el mecanismo de recuperación manual identificado como único mecanismo vigente (H4).

3.2.2. Framework de gobernanza de datos

La gobernanza de datos en la arquitectura propuesta aplica el marco DAMA-DMBOK descrito en la Sección 2.1.2, adaptado a las capacidades organizacionales identificadas en el diagnóstico. El objetivo del framework es formalizar las responsabilidades sobre los datos de negocio, reemplazar la gestión tácita sin respaldo institucional evidenciada en los hallazgos H2 y H5, y establecer los Acuerdos de Nivel de Servicio (SLAs) de calidad del dato que servirán como criterios de evaluación del prototipo en la Fase III.

3.2.2.1. Estructura de roles

El framework define tres roles fundamentales de gobernanza, con asignación explícita por dominio de datos:

Data Owner. Responsable de negocio del dominio. Define las políticas de calidad, autoriza el acceso y toma decisiones ante conflictos entre fuentes. Es un rol de jefatura o gerencia, con autoridad formal sobre el dominio.

Data Steward. Responsable operativo del dominio. Supervisa la implementación de las políticas, gestiona las excepciones de calidad y actúa como enlace entre el negocio y el equipo técnico. En el contexto de la organización, este rol recae naturalmente en el área de Sistemas, por su conocimiento transversal de los mecanismos de integración.

Data Consumer. Cualquier sistema o proceso que accede a los datos del dominio. No tiene responsabilidades de gobernanza pero está sujeto a las políticas de acceso y calidad definidas por el Data Owner.

La Tabla 3.7 presenta la asignación de roles para los tres dominios críticos, mapeando los perfiles identificados en las entrevistas a los roles formales del framework.

Tabla 3.7: Matriz de gobernanza de datos por dominio.

Dominio	Data Owner	Data Steward	Sistema de Registro
Cliente	Gerencia de Negocios Digitales	Analista de Sistemas	MDM (<i>Golden Record</i>)
Inventario	Jefatura de Almacén (Logística)	Analista de Sistemas	ERP Epicor → MDM
Pedido	Jefatura Financiera de Canales	Analista de Sistemas	MDM (identidad/estado) y ERP (detalle)

La asignación del rol de *Data Steward* al área de Sistemas en los tres dominios responde a la capacidad organizacional vigente: el diagnóstico evidenció que la organización no cuenta hoy con perfiles de negocio dedicados a la administración de datos, y que el conocimiento transversal de los mecanismos de integración reside en el área de Sistemas. Esta decisión es un punto de partida de transición, no el estado objetivo. Se reconoce que concentrar la *stewardship* en el área técnica conserva parte de la centralización que el diagnóstico señala como problemática (hallazgos H1 y H2); por ello, la evolución del modelo contempla migrar progresivamente la *stewardship* operativa hacia *Data Stewards* de negocio por dominio, dejando al área de Sistemas en el rol de *Data Custodian* técnico. Esta separación de responsabilidades de negocio y técnicas es coherente con el marco DAMA-DMBOK descrito en la Sección 2.1.2.

3.2.2.2. SLAs de calidad del dato por dominio

Para cada dominio se definen Acuerdos de Nivel de Servicio basados en cuatro dimensiones de calidad: exactitud (precisión del dato respecto a la realidad), unicidad (ausencia de duplicados), oportunidad (latencia máxima de actualización) y completitud (cobertura de atributos obligatorios). La Tabla 3.8 consolida los SLAs propuestos, derivados de los indicadores cuantitativos de la línea base diagnóstica.

Los umbrales de los SLAs se derivan de la línea base diagnóstica: se anclan en la evidencia de las entrevistas cuando ésta aporta un valor explícito y, en las dimensiones restantes, en objetivos de diseño razonados a partir de los hallazgos. La exactitud de inventario ($\geq 95\%$) se deriva directamente del margen de error tolerable del $\pm 5\%$ que define Llanos para el canal digital. Los umbrales de *oportunidad* expresan la transición del modelo batch al tiempo cercano al real: frente a las frecuencias actuales documentadas (15 minutos en VTEX, 2 horas en POS y 1 a 2 horas en el *snapshot* del ERP, Tabla 3.3), se fija la propagación de inventario en menos de un minuto —por ser la sobreventa el hallazgo crítico H3— y la actualización de cliente y pedido en menos de cinco y dos minutos respectivamente. La unicidad de cliente ($< 2\%$ de duplicados) responde al hallazgo H6 de duplicación frecuente sin deduplicación automatizada. Los umbrales de exactitud de cliente

Tabla 3.8: SLAs de calidad del dato por dominio.

Dominio	Exactitud	Unicidad	Oportunidad	Compleitud
Cliente	$\geq 98\%$ atributos verificables	$< 2\%$ registros duplicados	Actualización < 5 min post-evento	$\geq 95\%$ campos obligatorios
Inventario	$\geq 95\%$ precisión vs. inventario físico	N/A (SKU único por ERP)	Propagación < 1 min post-movimiento	100 % SKUs activos sincronizados
Pedido	$\geq 99\%$ sin errores de referencia	1 pedido = 1 registro canónico	Estado actualizado < 2 min post-evento	100 % atributos de facturación

($\geq 98\%$), completitud ($\geq 95\%$) y consistencia referencial de pedido ($\geq 99\%$) constituyen objetivos de diseño sobre las dimensiones de calidad del dato (DAMA International, 2017), coherentes con la criticidad financiera evidenciada por Duque (28 RMAs mensuales por inconsistencia), y se calibrarán empíricamente en la validación del prototipo. Estos SLAs constituyen los niveles de servicio técnicos que la arquitectura se compromete a cumplir; la evaluación del prototipo en el Capítulo 4 se estructura sobre los criterios de aceptación de los KPIs de negocio (Tabla 3.6), con los que los SLAs son consistentes: en particular, el SLA de exactitud de inventario ($\geq 95\%$) coincide con el objetivo del KPI de Exactitud de Inventario, y los SLAs de oportunidad habilitan su método de medición.

3.2.2.3. Política de deduplicación y resolución de conflictos

Dado que el hallazgo H6 documenta la ausencia de un proceso automatizado de deduplicación de clientes, se formaliza la siguiente política aplicable al dominio Cliente:

- Nivel 1. Coincidencia exacta:** Si dos registros comparten el mismo número de identificación (NIT o CC), se fusionan automáticamente (*auto-merge*). El registro más reciente prevalece en atributos de contacto; el registro más antiguo acumula el historial transaccional completo.
- Nivel 2. Coincidencia probable:** Si dos registros comparten correo electrónico y número de teléfono, pero difieren en el identificador, se marcan con estado **PROBABLE DUPLICATE** para revisión por el Data Steward. El MDM mantiene ambos registros vinculados hasta la resolución manual.
- Nivel 3. Sin coincidencia:** El registro se incorpora como nuevo cliente al *Golden Record* sin intervención adicional.

Esta política elimina la dependencia del criterio individual de cada área operativa para la resolución de duplicados, que era el mecanismo vigente identificado como inexistente en H6.

3.2.2.4. Control de acceso y protección de datos personales

El framework de gobernanza incorpora el control de acceso como responsabilidad explícita del *Data Owner* de cada dominio, cerrando en su dimensión de gobernanza la brecha de seguridad del dato identificada en la Tabla 3.4. El acceso a los datos maestros se rige por el principio de mínimo privilegio: cada *Data Consumer* —sistema interno o canal externo— accede únicamente a los atributos del dominio que su función requiere, según las políticas autorizadas por el *Data Owner*. La autorización se aplica de forma centralizada en el API Gateway mediante OAuth 2.0 y reglas de acceso por canal, evitando que cada sistema fuente implemente controles heterogéneos.

El dominio Cliente recibe tratamiento diferenciado por contener datos personales (identificación, correo electrónico, historial de compras). Su *Data Owner* (Gerencia de Negocios Digitales) autoriza y audita el acceso a estos atributos, y el MDM Service restringe su propagación a los consumidores con base legítima de uso, en línea con la Ley 1581 de 2012 de protección de datos personales en Colombia ([Congreso de la República de Colombia, 2012](#)). La consolidación del *Golden Record* de Cliente reduce además el riesgo de exposición al eliminar las copias duplicadas y desgobernadas documentadas en el hallazgo H6.

Los controles de seguridad de nivel de infraestructura —cifrado en reposo estandarizado y gestión centralizada de secretos— exceden el alcance de gobernanza de datos de este trabajo y se identifican como trabajo futuro.

3.2.3. Modelo de Datos Maestros (MDM)

3.2.3.1. Estrategia MDM adoptada

De las tres estrategias MDM descritas en la Sección 2.1.2 (Registry, Consolidation Hub y Coexistence) se adopta el modelo **Consolidation Hub**, justificado en tres criterios: (1) los sistemas fuente (ERP Epicor, VTEX, POS Cegid) mantienen representaciones heterogéneas del mismo cliente y del mismo SKU, haciendo necesaria una instancia de normalización centralizada que el modelo Registry no provee; (2) el modelo Coexistence implicaría sincronización bidireccional completa, incrementando la complejidad de implementación más allá del alcance del proyecto; y (3) el patrón de Consolidation Hub es compatible con una arquitectura EDA: el hub consume eventos de los sistemas fuente y publica eventos de actualización hacia los consumidores, aprovechando la infraestructura de mensajería asíncrona ya propuesta.

3.2.3.2. Atributos canónicos del Golden Record por entidad

La Tabla 3.9 define los atributos canónicos del *Golden Record* para cada dominio. Los atributos fueron seleccionados con base en los requerimientos identificados en las entrevistas y la estructura de integración actual documentada en la Fase I.

Tabla 3.9: Atributos canónicos del MDM por entidad.

Dominio	Atributo	Tipo	Fuente primaria	Regla MDM
Cliente	clienteId (UUID)	Clave maestra	MDM (generado)	Inmutable
	identificacion (NIT/CC)	Identificador	ERP / VTEX	Exacto
	nombre completo	Texto	VTEX Master Data	Más reciente
	correo electrónico	Texto	VTEX Master Data	Más reciente
	historial de compras	Lista	Todos los canales	Acumulativo
Inventario	skuId (código ERP)	Clave maestra	ERP Epicor	Inmutable
	descripción	Texto	ERP Epicor	ERP prevalece
	stockDisponible	Entero	ERP Epicor	ERP prevalece
	stockReservado	Entero	ERP Epicor	ERP prevalece
	ubicación bodega	Lista	WMS Cubic	WMS prevalece
Pedido	pedidoId (UUID)	Clave maestra	MDM (generado)	Inmutable
	ordenFuente	Texto	Sistema origen	Canal de origen
	estado	Enumerado	Evento más reciente	Transición válida
	clienteId (FK)	Referencia	MDM Cliente	Referencia MDM
	líneas de detalle	Lista	ERP Epicor	ERP prevalece

El dominio Pedido se gestiona bajo un esquema de co-masterización: el MDM genera el identificador canónico (`pedidoId`) y es la autoridad sobre el estado del pedido a lo largo de su ciclo

de vida, mientras que el ERP Epicor permanece como sistema de registro del detalle comercial —líneas, precios y facturación—. Por ello, en la matriz de gobernanza (Tabla 3.7) el sistema de registro de Pedido se expresa de forma dual: identidad y estado en el MDM, detalle y facturación en el ERP.

El dominio Inventario sigue un reparto análogo: el ERP Epicor es el sistema de registro y la autoridad del stock disponible (*available-to-promise*) y de las reservas, mientras que el WMS Cubic ejecuta los movimientos físicos —recepción, picking y despacho— y los confirma al ERP, manteniendo ambos sistemas alineados. El MDM consolida el valor canónico de inventario a partir del ERP —sin sumar las cantidades de ambos sistemas, lo que evita el doble conteo— y lo propaga a los canales como vista de lectura.

3.2.3.3. Ciclo de vida del Golden Record

El *Golden Record* opera en un ciclo de producción y difusión integrado con la arquitectura EDA:

1. Un sistema fuente registra un cambio en una entidad de negocio y emite un evento de dominio (e.g., `CustomerUpdated`, `InventoryMoved`, `OrderStatusChanged`).
2. El evento se publica en el *broker* de mensajes Apache Kafka, en el *topic* correspondiente al dominio.
3. El **MDM Service** consume el evento, aplica las reglas de normalización, validación y *matching*, y actualiza el *Golden Record*.
4. Si se detecta un duplicado de Nivel 1, el MDM fusiona los registros y emite `RecordMerged`; si es Nivel 2, emite `DuplicateDetected` con alerta al Data Steward.
5. El MDM emite un evento `MasterDataUpdated` en el *topic* de difusión del dominio.
6. Todos los sistemas consumidores suscritos reciben el evento y actualizan su representación local con la información canónica.

Este ciclo garantiza que el *Golden Record* refleje siempre el estado más reciente y normalizado, y que todos los sistemas downstream operen con el mismo estado canónico sin requerir consultas síncronas al MDM.

3.2.4. Arquitectura TO-BE orientada a eventos

3.2.4.1. Selección del broker de mensajes

La selección del *broker* de mensajes es la decisión tecnológica más crítica de la arquitectura EDA, ya que condiciona las capacidades de retención, entrega y escalabilidad de todos los flujos de evento. Se evaluaron tres alternativas ampliamente adoptadas en arquitecturas de integración empresarial: Apache Kafka, RabbitMQ y AWS SQS/SNS. La Tabla 3.10 presenta la evaluación comparativa basada en los criterios relevantes para el caso de uso.

Los criterios de evaluación no se definieron de forma *ad hoc*, sino que se derivaron de dos fuentes trazables. Por un lado, de los atributos de calidad propios del paradigma orientado a eventos descritos en la Sección 2.1.3 —retención durable del log, capacidad de *replay*, multiconsumo independiente y compactación por clave—, que condicionan la capacidad del *broker* para sostener el desacoplamiento temporal y la reconstrucción de estado. Por otro, de las restricciones operativas levantadas en el diagnóstico AS-IS: la necesidad de despliegue *on-premise* sobre la infraestructura existente y la compatibilidad con los *hooks* en tiempo real del WMS Cubic, único precedente EDA en la organización. Cada criterio de la Tabla 3.10 responde, por tanto, a un requisito arquitectónico o a una limitación del entorno, y no a una preferencia tecnológica previa.

Las valoraciones constituyen una comparación cualitativa de capacidades —presencia o ausencia de una característica (Sí/No) y órdenes de magnitud para los atributos cuantitativos— y no una medición empírica realizada en el entorno de la organización. Los valores de *throughput* corresponden a configuraciones con garantía de orden de entrega (Kafka particionado, RabbitMQ con colas durables y SQS en modo FIFO), por ser el orden de los eventos un requisito de los dominios de inventario y pedido; se expresan como órdenes de magnitud reportados en la documentación oficial de cada producto (Apache Software Foundation, 2025; RabbitMQ, 2025; Amazon Web Services, 2025) y corroborados, en el plano conceptual, por la literatura técnica de sistemas distribuidos (Kleppmann, 2017). Al tratarse de una evaluación documental de selección tecnológica —y no de un *benchmark* controlado—, su validez se sustenta en la trazabilidad de las fuentes y en la pertinencia de los criterios respecto al caso, no en la reproducción de mediciones de desempeño.

Se selecciona **Apache Kafka** como *broker* de mensajes; sobre las propiedades generales descritas en la Sección 2.1.3, pesan en esta decisión cuatro razones aplicadas al caso: (1) la retención durable y la capacidad de *replay* son críticas para el dominio de inventario, donde el estado actual debe poder reconstruirse ante fallos de procesamiento; (2) la compactación de log por clave permite mantener el último estado conocido de cada SKU, reduciendo el volumen de almacenamiento requerido para el historial de inventario; (3) el soporte nativo de múltiples *consumer groups* garantiza que POS Cegid, VTEX y los canales externos procesen los eventos de inventario de forma completamente independiente; y (4) el despliegue *on-premise* es un requerimiento operativo dado que la infraestructura actual de la organización opera en centros de datos propios (Kleppmann, 2017).

3.2.4.2. Componentes de la arquitectura TO-BE

La arquitectura TO-BE propuesta está compuesta por seis componentes principales, cuyas responsabilidades están delimitadas por dominio de negocio y no por sistema fuente:

Apache Kafka (Event Broker). Bus de mensajería asíncrona central. Gestiona los *topics* particionados por dominio de evento. Garantiza entrega durable con semántica *at-least-once*. Los *topics* principales son: `inventory.movements`, `mdm.inventory.updated`, `orders.new`, `orders.lifecycle`, `customers.raw` y `mdm.customer.updated`.

API Gateway. Punto de entrada unificado para canales externos (MercadoLibre, Falabella, Sodimac, Bylmo). Abstrae los sistemas internos de los canales de consumo externo, implementa

Tabla 3.10: Evaluación comparativa de brokers de mensajes.

Criterio	Apache Kafka	Kafka	RabbitMQ	AWS SQS/SNS
Retención durable del log de eventos	Configurable (días / TB)		No (consumo elimina)	Máx. 14 días
Capacidad de <i>replay</i> por <i>offset</i>	Sí		No	No
<i>Throughput</i> sostenido	>1M msgs/seg		≈50K msgs/seg	≈3K msgs/seg (FIFO)
Multi-consumidor nativo	Sí (<i>consumer groups</i>)		Manual (<i>fanout exchange</i>)	Manual (fan-out SNS)
Compactación de log por clave	Sí		No	No
Despliegue <i>on-premise</i>	Sí		Sí	No (cloud AWS)
Compatibilidad hooks WMS Cubic	Alta Sink Connector)	(HTTP Connector)	Alta	Media

autenticación OAuth 2.0, autorización por canal y *rate limiting* centralizado. Reduce el tiempo de incorporación de un nuevo canal de integración de cuatro–ocho semanas a horas o días, al eliminar la necesidad de modificar los sistemas internos para cada nuevo canal.

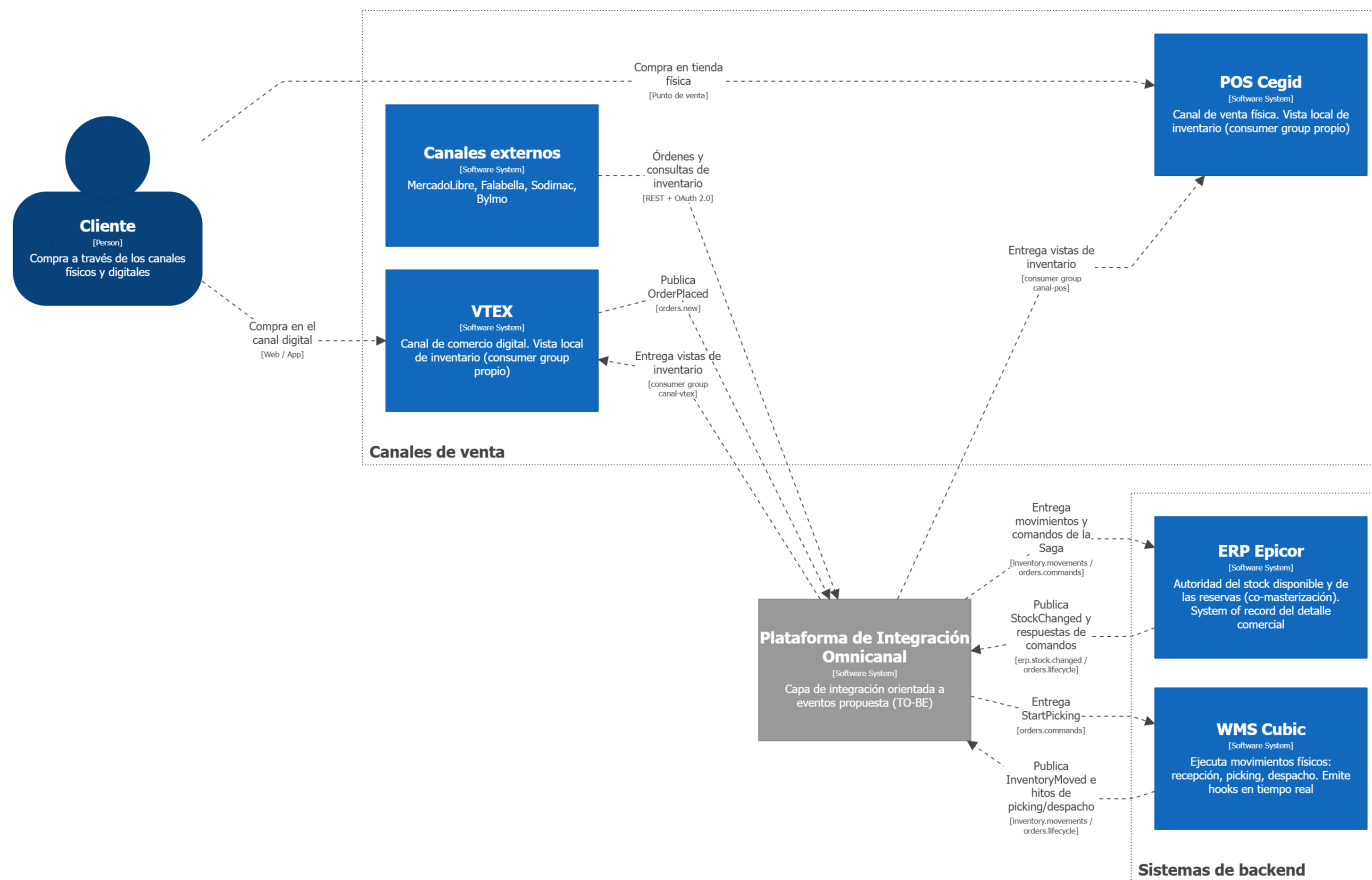
MDM Service. Servicio dedicado a la producción y gestión del *Golden Record*. Expone una API REST para consulta de datos maestros por dominio y publica eventos `MasterDataUpdated` en Kafka cuando el *Golden Record* es modificado.

Order Orchestrator. Servicio de orquestación del ciclo de vida del pedido. Implementa el patrón **Saga orquestada**: actúa como coordinador central del flujo pedido–inventario–despacho–entrega, emite comandos hacia cada servicio participante en secuencia y gestiona las Sagas de compensación ante condiciones de fallo. Al centralizar la lógica de coordinación en un único componente, el Order Orchestrator facilita el monitoreo del estado del pedido y la trazabilidad transaccional extremo a extremo.

Event Store / Audit Log. Registro inmutable de todos los eventos de negocio. Implementado como *topic* de Kafka con retención máxima configurada. Provee la trazabilidad extremo a extremo que cierra la brecha de visibilidad documentada en H7: creación, alistamiento, despacho y entrega son estados explícitos y consultables.

Monitoring & Alerting. Capa de observabilidad transversal. Supervisa la salud de los *topics*, latencia de procesamiento, tasa de errores y disponibilidad de consumidores. Reemplaza la detección manual de fallos (tokens OAuth expirados, archivos no recibidos sin alerta) identificada como mecanismo vigente en H4.

La Figura 3.7 presenta la vista panorámica de la arquitectura propuesta a nivel de sistemas: en contraste con la topología radial centrada en el ERP del estado actual (Figura 3.1), todos los sistemas empresariales se relacionan a través de la Plataforma de Integración Omnicanal, que absorbe la complejidad de integración y elimina las dependencias directas entre sistemas.



System Landscape View
 Vista de Panorama C4 - Sistemas involucrados
 Sunday, July 12, 2026 at 12:44 PM Colombia Standard Time

Figura 3.7: Vista panorámica (*System Landscape*) – Arquitectura TO-BE

La Figura 3.8 presenta la vista de contenedores de la arquitectura propuesta. En contraste con la topología punto a punto del AS-IS (Figura 3.1), toda comunicación entre dominios transita por el broker de mensajes: los sistemas existentes se conectan como productores y consumidores de eventos, y ningún componente conoce directamente a otro.

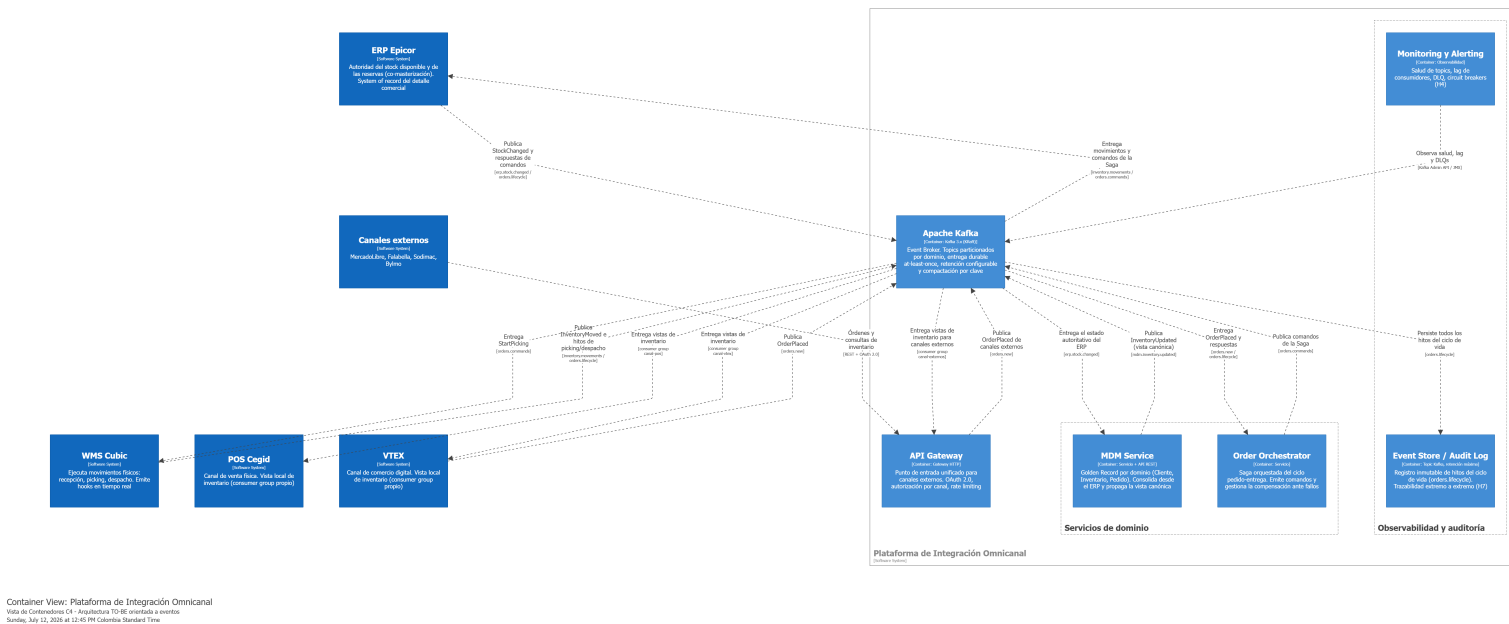


Figura 3.8: Vista de Contenedores C4 – Arquitectura TO-BE orientada a eventos

3.2.4.3. Flujos de eventos por dominio

Dominio Inventario: flujo `InventoryMoved`. El flujo de actualización de inventario en tiempo cercano al real reemplaza la actualización batch de una a dos horas que genera el 20 % de pedidos fallidos en el canal digital:

1. El WMS Cubic registra un movimiento físico (entrada, salida o ajuste) y emite el evento `InventoryMoved` mediante los *hooks* existentes. Estos *hooks* ya operan en tiempo real y son el único precedente EDA en la organización.
2. El evento se publica en el *topic* `inventory.movements` de Kafka.
3. El MDM Service consume el evento y actualiza la representación canónica de inventario para su propagación a los canales; el ERP Epicor, autoridad del stock disponible, registra el mismo movimiento y su valor prevalece ante discrepancias, manteniéndose alineado con el WMS. El MDM emite `InventoryUpdated` en el *topic* `mdm.inventory.updated`.
4. Los consumidores suscritos mediante sus respectivos *consumer groups* (POS Cegid, VTEX, API Gateway para canales externos) reciben el evento de forma independiente y actualizan su representación local de stock.
5. El *consumer group* de auditoría registra el evento en el Event Store para trazabilidad histórica.

Este flujo garantiza que todos los canales dispongan del mismo estado de inventario en menos de un minuto tras el movimiento físico, cumpliendo el SLA de oportunidad definido en la Tabla 3.8.

Conviene distinguir dos regímenes de consistencia que conviven en el dominio de inventario. La propagación de `InventoryUpdated` hacia los consumidores (POS Cegid, VTEX, canales externos) alimenta *vistas de lectura* eventualmente consistentes, cuyo fin es mostrar la disponibilidad aproximada por canal con latencia inferior a un minuto. La reserva de stock, en cambio, no se resuelve contra esas vistas locales: la ejecuta el ERP Epicor como autoridad consistente del stock disponible, de modo que los pedidos concurrentes sobre el mismo SKU se serializan en un único punto y solo uno obtiene la reserva cuando el stock es escaso. El MDM consolida el valor canónico a partir del ERP y lo propaga a los canales, mientras que el Order Orchestrator coordina la Saga sin asumir la autoridad del stock. Para preservar el orden de los movimientos de un mismo SKU, el *topic* `inventory.movements` se particiona por `skuId`, aprovechando la garantía de orden que Kafka ofrece a nivel de partición. Esta separación entre lectura eventual y reserva consistente elimina el riesgo de sobreventa sin sacrificar el desacoplamiento de los canales de consulta.

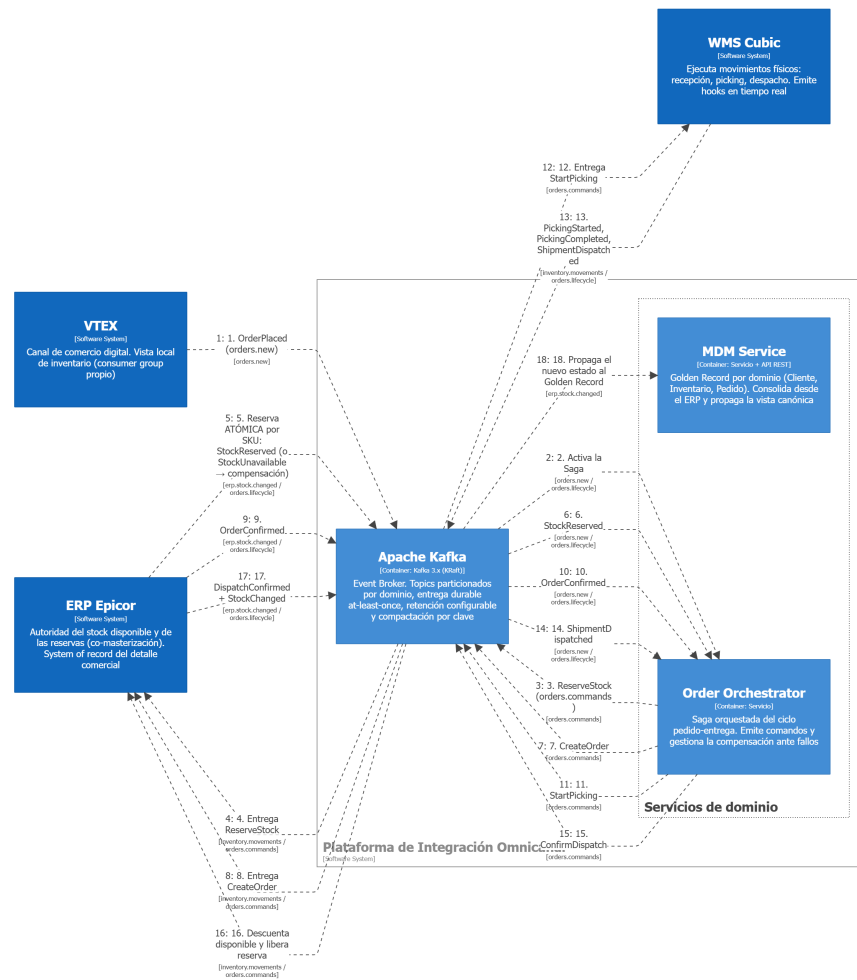
Dominio Pedido: flujo `OrderPlaced` (Saga de cumplimiento). El ciclo de vida del pedido implementa la variante **orquestada** del patrón Saga descrito en la Sección 2.1.3: el Order Orchestrator concentra la lógica de coordinación y compensación, emite los comandos hacia los servicios participantes y gestiona la Saga de compensación cuando una condición de fallo interrumpe el proceso, habilitando la auditoría del estado de cada pedido en tiempo real:

1. El canal de ventas publica **OrderPlaced** en el *topic orders.new*. Este evento activa el Order Orchestrator, que toma control de la Saga.
2. El Order Orchestrator emite el comando **ReserveStock** hacia el ERP Epicor, sistema de registro y autoridad del stock disponible, que ejecuta la reserva de forma atómica sobre la cantidad autoritativa y responde **StockReserved** si hay disponibilidad o **StockUnavailable** en caso contrario. Al concentrarse la reserva en una única autoridad, los pedidos concurrentes sobre el mismo SKU se serializan y se elimina la condición de sobreventa. Ante **StockUnavailable**, el Order Orchestrator emite **CancelOrder** e inicia la Saga de compensación (**OrderCancelled** → notificación al cliente).
3. El Order Orchestrator emite el comando **CreateOrder** al ERP. El ERP genera la orden interna y responde con **OrderConfirmed**.
4. El Order Orchestrator emite el comando **StartPicking** al WMS. El WMS reporta progreso mediante **PickingStarted** y **PickingCompleted** con estampas de tiempo.
5. El WMS notifica **ShipmentDispatched** con el código de seguimiento de la paquetera. El Order Orchestrator registra el evento en el Event Store y delega la notificación al cliente al servicio de notificaciones.
6. Al recibir **DeliveryCompleted** desde la paquetera vía API Gateway, el Order Orchestrator cierra la Saga y actualiza el estado final del pedido en el MDM.

La Saga de compensación se activa desde el paso 2 si se emite **StockUnavailable**: el Order Orchestrator ejecuta comandos de reversión en orden inverso (**ReleaseStock** si ya se reservó stock, **CancelOrder** al ERP si ya se creó la orden), garantizando que ningún servicio quede en estado inconsistente. Esto elimina el reproceso manual identificado en la falla operacional documentada en la sección 3.1.2.

Los eventos **PickingStarted**, **PickingCompleted**, **ShipmentDispatched** y **DeliveryCompleted** cierran explícitamente las dos brechas de visibilidad identificadas por Collazos en el hallazgo H7: el alistamiento en tienda y la última milla.

La Figura 3.9 representa la secuencia completa de la Saga, incluyendo la serialización de la reserva en el ERP y los hitos operativos emitidos por el WMS.



Dynamic View: Plataforma de Integración Omnicanal
 Saga orquestada: camino feliz del ciclo pedido-entrega
 Sunday, July 12, 2026 at 12:45 PM Colombia Standard Time

Figura 3.9: Saga orquestada del ciclo pedido-entrega (flujo OrderPlaced)

Con fines de comunicación, se construyó adicionalmente una visualización interactiva del ecosistema a partir de la especificación de eventos y del contrato de topics del prototipo (Sección 4.1). No sustituye la evidencia de ejecución presentada en el Capítulo 4 — es una representación ilustrativa, no una captura de telemetría en vivo — pero facilita seguir visualmente el camino feliz y la rama de compensación de la Saga. Las Figuras 3.10 y 3.11 muestran ambos casos.

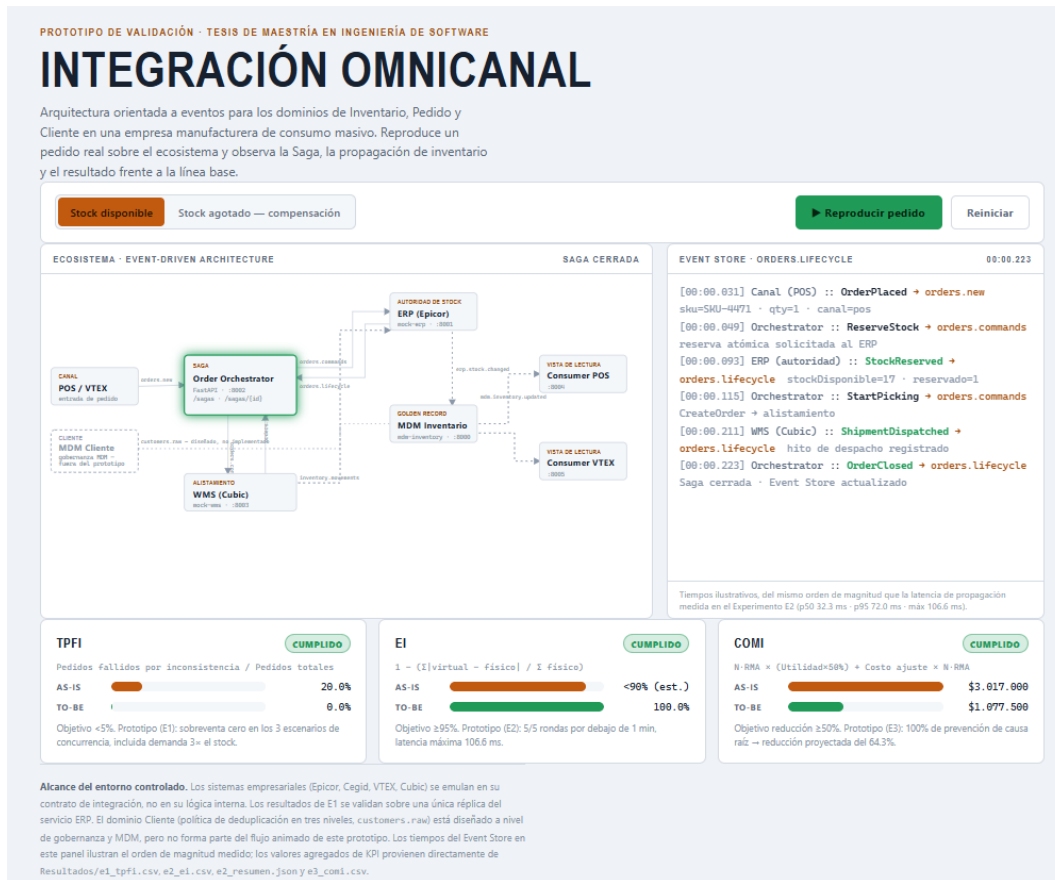


Figura 3.10: Visualización ilustrativa del ecosistema: camino feliz de la Saga (OrderPlaced → OrderClosed).

Dominio Cliente: flujo CustomerRegistered.

1. Cualquier canal (tienda física, VTEX, canal externo) genera **CustomerRegistered** con los atributos de identificación del cliente y los publica en el *topic* **customers.raw**.
2. El MDM Service consume el evento y aplica las reglas de *matching* de la política de deduplicación.
3. Si coincidencia exacta (Nivel 1): fusión automática, emisión de **RecordMerged**.

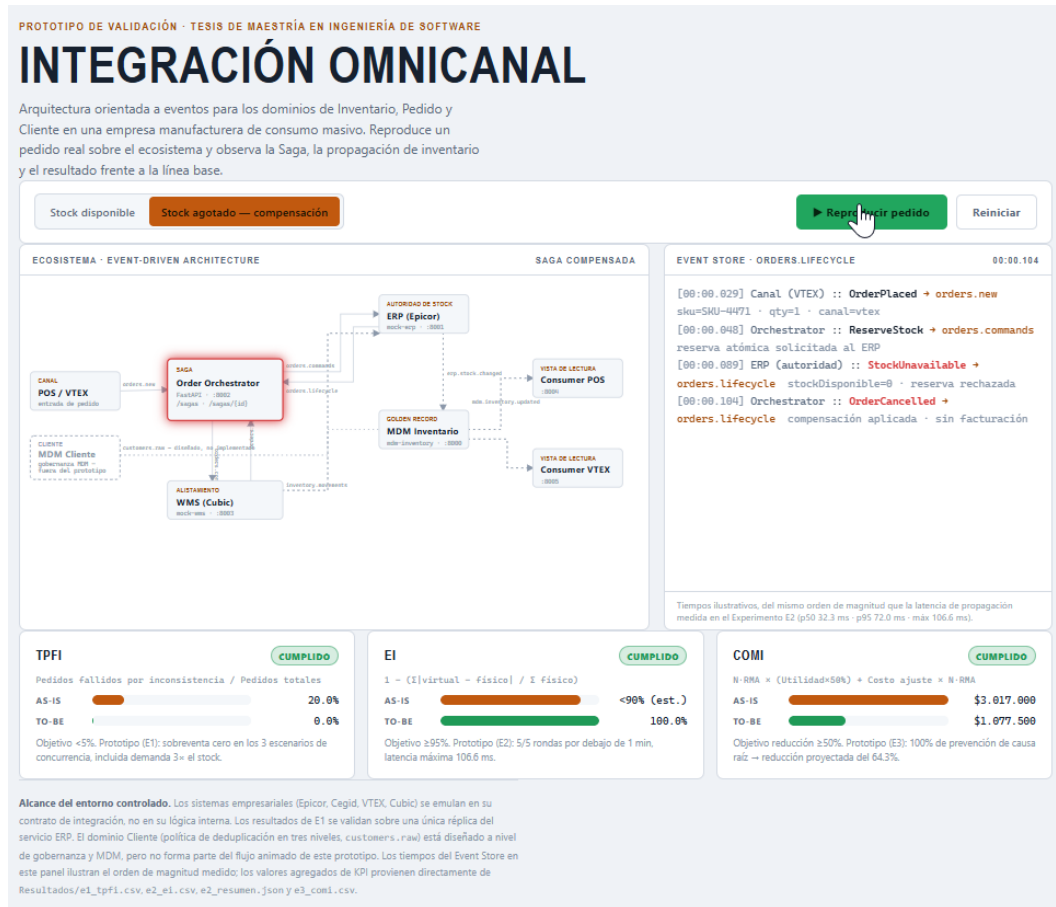


Figura 3.11: Visualización ilustrativa del ecosistema: rama de compensación ante StockUnavailable (OrderPlaced → OrderCancelled).

4. Si coincidencia probable (Nivel 2): emisión de `DuplicateDetected` con alerta al Data Steward; los sistemas continúan operando con el registro más reciente mientras se resuelve.
5. El *Golden Record* actualizado se propaga mediante `CustomerUpdated` en el `topic mdm.customer.updated` a todos los sistemas suscritos, garantizando consistencia de la entidad cliente en todos los canales.

3.2.4.4. Estrategia de resiliencia

La arquitectura incorpora cuatro mecanismos de resiliencia en respuesta al hallazgo H4, que documenta la ausencia total de recuperación automática ante fallos de integración:

Dead Letter Queue (DLQ). Los eventos que fallan en tres intentos consecutivos de procesamiento son enrutados automáticamente a una cola DLQ específica por *topic*. El contenido de cada DLQ es monitoreado en tiempo real, y los eventos fallidos quedan disponibles para revisión y reintento manual por el equipo de operaciones.

Política de reintentos con *backoff* exponencial. El primer reintento ocurre a 1 segundo; el segundo a 5 segundos; el tercero a 30 segundos. Este esquema reduce la presión sobre sistemas temporalmente degradados sin bloquear el procesamiento de otros eventos en el mismo *topic*.

Idempotencia de consumidores. Cada evento incluye un `eventId` UUID inmutable. Los consumidores mantienen un registro de los últimos eventos procesados, garantizando que un reintento no produzca duplicación de efectos en el estado del negocio: la entrega *at-least-once* a nivel de transporte, combinada con la idempotencia del consumidor, produce un efecto *effectively-once* sobre el estado de negocio (Kleppmann, 2017).

Circuit Breaker por consumidor. Si un consumidor registra una tasa de error superior al 50 % en una ventana de 60 segundos, el circuit breaker se activa: los eventos se acumulan en el *offset* de Kafka y el consumidor entra en modo de recuperación. Esto garantiza que un fallo en POS Cegid, por ejemplo, no bloquee la actualización de VTEX ni de los canales externos.

Los parámetros concretos de estos mecanismos —número de reintentos, intervalos de *backoff*, y umbral y ventana del *circuit breaker*— corresponden a valores iniciales basados en prácticas estándar de resiliencia en arquitecturas de mensajería, sujetos a calibración empírica a partir del comportamiento observado durante la operación y la validación del prototipo.

3.2.5. Trazabilidad diagnóstico-solución

La Tabla 3.11 presenta el mapa de trazabilidad entre los siete hallazgos convergentes del diagnóstico (H1–H7) y los componentes de diseño que los resuelven. Esta trazabilidad garantiza que cada componente propuesto responde a evidencia empírica documentada y no a preferencias tecnológicas; y permite al jurado verificar la coherencia entre problema, solución y criterios de evaluación del prototipo.

Tabla 3.11: Trazabilidad entre hallazgos del diagnóstico y componentes de diseño.

Hallazgo	Problema documentado	Componente de diseño que lo resuelve
H1	ERP como MDM de facto sin diseño formal	MDM Service, Consolidation Hub y <i>Golden Record</i> por dominio.
H2	Ausencia de <i>Data Owner</i> en dominios críticos	Framework de gobernanza: matriz de roles por dominio (Tabla 3.7).
H3	Discrepancias de inventario como norma operativa	Flujo <i>InventoryMoved</i> EDA y <i>Golden Record</i> de Inventario.
H4	Arquitectura batch sin recuperación automática	Apache Kafka, DLQ, reintentos con <i>backoff</i> exponencial y <i>circuit breakers</i> .
H5	Reglas de validación no documentadas ni centralizadas	MDM Service: políticas de calidad y deduplicación formalizadas.
H6	Duplicación de clientes sin deduplicación automática	Política de deduplicación en tres niveles del MDM Service.
H7	Brechas de visibilidad en alistamiento y última milla	Eventos <i>PickingStarted</i> , <i>PickingCompleted</i> , <i>ShipmentDispatched</i> y <i>DeliveryCompleted</i> registrados en el Event Store.

Evaluación

4.1. Diseño de la evaluación

La evaluación de la arquitectura propuesta se realizó mediante un prototipo funcional ejecutado en un entorno controlado, en correspondencia con la Fase III de la metodología (Sección 1.5) y con la delimitación del proyecto, que excluye la implementación en producción. El propósito de la evaluación no es medir el desempeño absoluto de la infraestructura, sino verificar que los mecanismos arquitectónicos diseñados en el Capítulo 3 eliminan las causas raíz documentadas en el diagnóstico. Para ello, cada experimento se deriva directamente de los criterios de aceptación definidos en la Tabla 3.6 y se contrasta contra la línea base de KPIs de la Tabla 3.5.

4.1.1. Entorno de evaluación

El prototipo materializa los componentes centrales de la arquitectura TO-BE descritos en la Sección 3.2: un broker Apache Kafka en modo KRaft; un servicio que emula el ERP Epicor como autoridad del stock disponible y de las reservas (co-masterización del dominio Inventario); el MDM Service, que consolida el *Golden Record* a partir del ERP y lo propaga a los canales; el Order Orchestrator, que implementa la Saga orquestada del ciclo pedido–entrega con su lógica de compensación; un servicio que emula el WMS Cubic emitiendo los hitos de alistamiento y despacho; y dos consumidores de canal (POS y comercio digital) suscritos con *consumer groups* independientes. El topic `orders.lifecycle` cumple la función de *Event Store*: todo hito del ciclo de vida del pedido queda registrado con estampas de tiempo, lo que habilita tanto la trazabilidad extremo a extremo (hallazgo H7) como la instrumentación de los experimentos. Los servicios se implementaron en Python 3.11 y se despliegan mediante contenedores Docker sobre una única máquina.

El flujo de propagación de inventario reproduce la cadena definida en el diseño: el WMS emite `InventoryMoved` hacia `inventory.movements`; el ERP aplica el delta como autoridad y publica su estado en `erp.stock.changed`; el MDM actualiza el *Golden Record* y difunde `InventoryUpdated` en `mdm.inventory.updated`; los canales actualizan sus vistas locales de lectura. La reserva de stock, en cambio, se resuelve de forma atómica en el ERP a través del comando `ReserveStock` de la Saga, serializada por SKU mediante el particionado por clave del topic de comandos.

4.1.2. Alcance y supuestos del entorno controlado

Cuatro supuestos delimitan la validez de los resultados. Primero, los sistemas empresariales reales (Epicor, Cegid, VTEX, Cubic) se sustituyen por servicios que emulan su contrato de integración, no su lógica interna; esto es suficiente porque las brechas diagnosticadas residen en los

mecanismos de integración entre sistemas, no en la funcionalidad interna de cada uno. Segundo, la comparación con el modelo AS-IS se realiza mediante simulación del mecanismo batch documentado en la Sección 3.1: cada canal opera contra una copia local del inventario (*snapshot*) que solo se refresca periódicamente, reproduciendo a escala el comportamiento que en la operación real presenta latencias de 15 minutos a 24 horas. Tercero, el impacto financiero (COMI) se evalúa como proyección anclada en evidencia: el prototipo mide la tasa de prevención de la causa raíz que genera las devoluciones por inconsistencia, y esa tasa se aplica a la composición causal de las RMAs documentada en la línea base (18 de 28 RMAs de marzo de 2026 atribuibles a cambio de bodega por inconsistencia). Cuarto, el prototipo acota el ciclo de la Saga descrito en la Sección 3.2 al tramo verificable dentro del entorno controlado: cierra en `ShipmentDispatched` → `OrderClosed` y no incorpora el hito `DeliveryCompleted` de última milla, que depende de la paquetera externa y queda fuera del alcance experimental; de igual forma, los tres experimentos ejercitan la compensación temprana (`StockUnavailable` → `OrderCancelled`, antes de reservar) pero no la rama de `ReleaseStock` posterior a una reserva ya efectuada, que permanece como parte del diseño TO-BE sin validación empírica en esta fase.

4.1.3. Experimentos

Se diseñaron tres experimentos, uno por cada KPI de la línea base diagnóstica.

Experimento E1 — Prevención de sobreventa (TPFI). Reproduce el mecanismo estructural de la sobreventa identificado en el hallazgo H3: el mismo inventario expuesto simultáneamente a múltiples canales sin descuento en tiempo real. Se ejecutan tres escenarios de concurrencia con demanda superior al stock disponible (demanda $2\times$ y $3\times$ el stock) y un escenario de control (demanda igual al stock). En la condición AS-IS simulada, tres canales aceptan pedidos contra snapshots locales independientes; en la condición TO-BE, los pedidos ingresan como `OrderPlaced` y la Saga solicita la reserva atómica al ERP. La métrica es la sobreventa (pedidos comprometidos por encima del stock real) y el criterio de aceptación exige sobreventa igual a cero en los tres escenarios TO-BE.

Experimento E2 — Exactitud de Inventario y latencia de propagación (EI). Verifica el SLA de oportunidad del dominio Inventario (Tabla 3.8): propagación inferior a un minuto tras cada movimiento. Se ejecutan cinco rondas consecutivas de diez movimientos aleatorios (entradas y salidas sobre cinco SKUs); al cierre de cada ronda se compara la “verdad física” registrada por el WMS contra el *Golden Record* del MDM (cálculo de EI según la Ecuación 3.2) y se mide la latencia extremo a extremo de cada evento, desde la emisión del `InventoryMoved` hasta la recepción del `InventoryUpdated` en el punto de consumo del canal. El criterio de aceptación exige latencia máxima inferior a un minuto en las cinco rondas y $EI \geq 95\%$.

Experimento E3 — Eliminación de la causa raíz de las RMAs (COMI). Reproduce dieciocho veces el mecanismo causal de la falla operacional documentada en la Sección 3.1: un pedido asignado sobre la última unidad disponible mientras un segundo canal compromete simultáneamente la misma unidad contra su vista desactualizada. En el modelo AS-IS este escenario desemboca en facturación, imposibilidad de despacho, nota crédito y refacturación —el ciclo que genera la RMA por inconsistencia—. En el TO-BE, la reserva atómica en el ERP garantiza que solo un pedido

comprometa la unidad y el segundo se cancele mediante compensación antes de facturar. La tasa de prevención medida se aplica a las 18 RMAs mensuales atribuibles a inconsistencia para proyectar el COMI TO-BE mediante la Ecuación 3.3. El criterio de aceptación exige la eliminación de la causa raíz en al menos el 50 % de los eventos generadores de RMA.

4.1.4. Amenazas a la validez

Se reconocen tres amenazas principales. En cuanto a la *validez de constructo*, la simulación AS-IS reproduce el mecanismo causal del modelo batch (vistas desactualizadas e independientes por canal) pero no su escala temporal real; esta reducción es conservadora, pues intervalos de desactualización mayores solo incrementarían la sobreventa del AS-IS. En cuanto a la *validez interna*, los servicios emulados ejecutan lógica simplificada; para mitigarlo, los mecanismos evaluados (reserva atómica, propagación por eventos, compensación de la Saga) son precisamente los componentes reales de la arquitectura y no simulaciones. En cuanto a la *validez externa*, los resultados de latencia corresponden a un despliegue local y no son extrapolables como valores absolutos a la infraestructura productiva; por ello los criterios de aceptación se formulan sobre umbrales holgados (un minuto) y sobre propiedades estructurales (sobreventa cero) en lugar de valores finos de desempeño.

4.2. Resultados de la evaluación

Los tres experimentos se ejecutaron sobre el entorno descrito en la Sección 4.1, con los seis servicios desplegados sobre Kafka en modo KRaft. Los resultados crudos y la evidencia de ejecución (estado de los servicios, log del Order Orchestrator y resumen de sagas) se conservan como anexos digitales del proyecto.

4.2.1. Resultados E1 — Prevención de sobreventa (TPFI)

La Tabla 4.1 contrasta la sobreventa observada en los tres escenarios: la simulación AS-IS (snapshot independiente por canal) frente a la ejecución real TO-BE sobre el prototipo.

Tabla 4.1: Resultados E1: sobreventa AS-IS (simulado) vs. TO-BE (prototipo).

Escenario	Stock	Pedidos	AS-IS (simulado)		TO-BE (prototipo)	
			Sobrev.	TPFI	Sobrev.	TPFI
E1-A (demanda 2×)	10	20	10	50,0 %	0	0,0 %
E1-B (demanda 3×)	5	15	10	66,7 %	0	0,0 %
E1-C (control 1×)	8	8	0	0,0 %	0	0,0 %

En los escenarios de sobredemanda, el modelo AS-IS comprometió el doble y el triple del stock real (TPFI del 50 % y 66,7 %), reproduciendo el mecanismo de sobreventa sistémica del hallazgo H3 e incluso superando el 20 % observado en la operación real, como corresponde a escenarios de demanda concentrada. En la condición TO-BE, la reserva atómica en el ERP comprometió

exactamente el stock disponible en los tres escenarios (10, 5 y 8 pedidos respectivamente), canceló mediante compensación los pedidos excedentes antes de facturar y cerró con inventario final coherente (cero unidades comprometidas de más). Con sobreventa igual a cero en los tres escenarios, incluido el de control, el criterio de aceptación del Experimento E1 queda satisfecho.

4.2.2. Resultados E2 — Exactitud de Inventario y latencia (EI)

La Tabla 4.2 presenta los resultados de las cinco rondas consecutivas de movimientos de inventario.

Tabla 4.2: Resultados E2: exactitud de inventario y latencia de propagación por ronda.

Ronda	Movimientos	Discrepancia (unid.)	EI	Latencia máx. (ms)
1	10	0	100 %	44,4
2	10	0	100 %	72,0
3	10	0	100 %	106,6
4	10	0	100 %	62,0
5	10	0	100 %	51,4

El *Golden Record* del MDM coincidió con la verdad física del WMS en la totalidad de los cincuenta movimientos (EI del 100 % en las cinco rondas, sin discrepancias de unidades). La latencia extremo a extremo —desde la emisión del `InventoryMoved` hasta la actualización de la vista canónica— registró una mediana de 32,3 ms, un percentil 95 de 72,0 ms y un máximo de 106,6 ms: tres órdenes de magnitud por debajo del umbral de un minuto del SLA de oportunidad (Tabla 3.8). Frente a la línea base AS-IS —donde la propagación batch acumulaba entre 15 minutos y 24 horas— el prototipo redujo la latencia de sincronización en un factor superior a 10^4 , cumpliendo tanto el umbral de oportunidad como el piso de exactitud (≥ 95 %) exigidos por el Experimento E2.

4.2.3. Resultados E3 — Eliminación de la causa raíz de las RMAs (COMI)

En las dieciocho réplicas del mecanismo causal (dos canales comprometiendo simultáneamente la última unidad disponible), la reserva atómica del ERP y la compensación de la Saga previnieron la doble facturación en la totalidad de los casos: dieciocho prevenciones en dieciocho réplicas, es decir, una tasa de prevención del 100 %. En cada réplica, exactamente un pedido obtuvo la reserva y completó su ciclo hasta `OrderClosed`, mientras el pedido concurrente fue cancelado mediante `OrderCancelled` antes de generar facturación, eliminando el ciclo nota crédito–refacturación que origina la RMA por inconsistencia.

Aplicando la tasa de prevención medida a la composición causal de la línea base (18 de 28 RMAs mensuales atribuibles a inconsistencia), la proyección financiera mediante la Ecuación 3.3 es la siguiente: las RMAs mensuales se reducirían de 28 a 10, y el COMI pasaría de COP \$3.017.000 a COP \$1.077.500 mensuales. Esa reducción del 64,3 % supera tanto el objetivo del 50 % definido

en la Tabla 3.6 como el valor objetivo TO-BE de la línea base (<COP \$1,51M, Tabla 3.5), con lo que el Experimento E3 también satisface su criterio de aceptación.

4.2.4. Verificación de los criterios de aceptación

La Tabla 4.3 consolida el veredicto de los tres experimentos contra los criterios de aceptación definidos en la Tabla 3.6.

Tabla 4.3: Verificación de los criterios de aceptación del prototipo.

KPI	Criterio (Tabla 3.6)	Resultado	Veredicto
TPFI	Sobreventa cero en al menos 3 escenarios concurrentes	Sobreventa = 0 en E1-A, E1-B y E1-C	Cumplido
EI	<i>Golden Record</i> refleja el WMS en <1 min en 5 pruebas consecutivas	Latencia máx. 106,6 ms; EI 100 % en 5 rondas	Cumplido
COMI	Eliminación de la causa raíz de ≥ 50 % de los eventos que generan RMA	Prevención del 100 % en 18 réplicas; reducción proyectada del COMI del 64,3 %	Cumplido

4.2.5. Evidencia de ejecución del prototipo

Las Figuras 4.1 a 4.6 documentan el prototipo en ejecución al momento de capturar los resultados reportados en esta sección: los seis servicios desplegados sobre Docker, sus contratos API expuestos vía OpenAPI/Swagger, y respuestas reales de las consultas de estado del sistema.

```

PS C:\Users\jpe1\OneDrive\Documents\Maestria Ing> Soft\Proyecto de Grado\Proyecto de Grado\04_Protoptipo_Evaluacion\prototipo> docker compose ps
NAME                IMAGE                COMMAND                SERVICE                CREATED                STATUS                PORTS
consumer-ps        prototipo-consumer-ps  ["nginx", "service.châ...  consumer-ps            10 days ago          Up about an hour          0.0.0.0:8004-8005/tcp, [::]:8004-8005/tcp
consumer-vtx       prototipo-consumer-vtx  ["nginx", "service.châ...  consumer-vtx           10 days ago          Up about an hour          0.0.0.0:8005-8006/tcp, [::]:8005-8006/tcp
kafka              kafka:3.8.0           ["kafka", "start.sh", ...  kafka                  10 days ago          Up about an hour          0.0.0.0:29092->29093/tcp, [::]:29092->29093/tcp
mdm-inventory      prototipo-mdm-inventory  ["nginx", "service.mdm...  mdm-inventory          10 days ago          Up 19 minutes             0.0.0.0:8000-8000/tcp, [::]:8000-8000/tcp
mdm-erp            prototipo-mdm-erp      ["nginx", "service.mdm...  mdm-erp                10 days ago          Up about an hour          0.0.0.0:8001-8001/tcp, [::]:8001-8001/tcp
mock-mock-mdm      prototipo-mock-mdm     ["nginx", "service.mdm...  mock-mock-mdm          10 days ago          Up about an hour          0.0.0.0:8003-8003/tcp, [::]:8003-8003/tcp
order-orchestrator prototipo-order-orchestrator  ["nginx", "service.mdm...  order-orchestrator     10 days ago          Up about an hour          0.0.0.0:8002-8002/tcp, [::]:8002-8002/tcp

```

Figura 4.1: Los seis servicios del prototipo desplegados vía Docker Compose, con el broker Kafka en estado *healthy*.

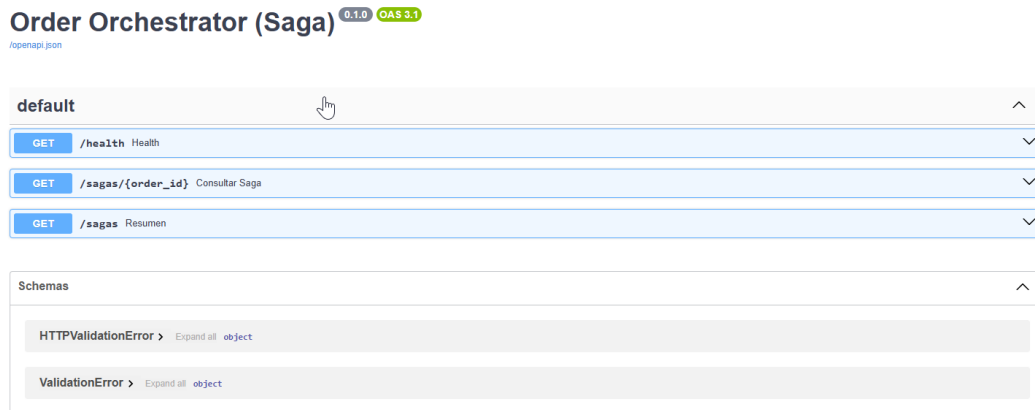


Figura 4.2: Contrato API del Order Orchestrator: GET /sagas, GET /sagas/{order_id} y GET /health.

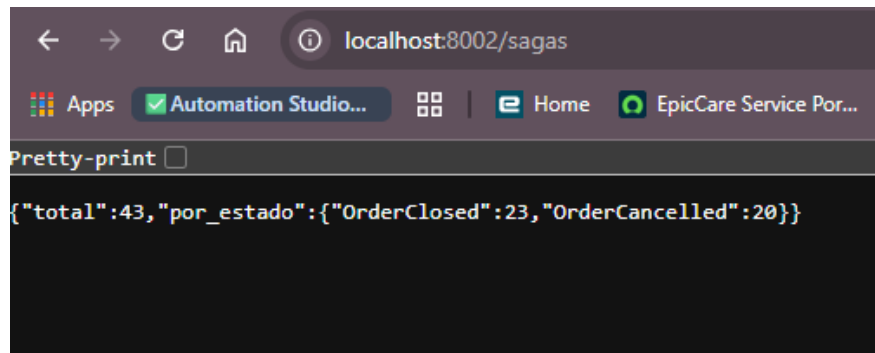


Figura 4.3: Respuesta real de GET /sagas: estado agregado del Event Store tras la ejecución de los experimentos (sagas cerradas y canceladas por compensación).



Figura 4.4: Contrato API del ERP (mock-erp): GET /stock/{sku}, autoridad del stock disponible y de las reservas.

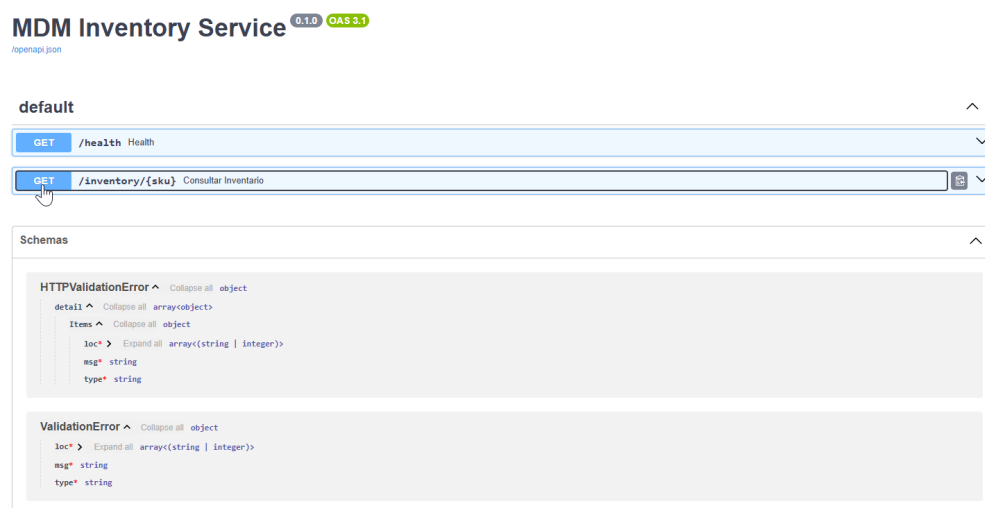


Figura 4.5: Contrato API del MDM Inventory Service: GET /inventory/{sku}, consulta del *Golden Record*.

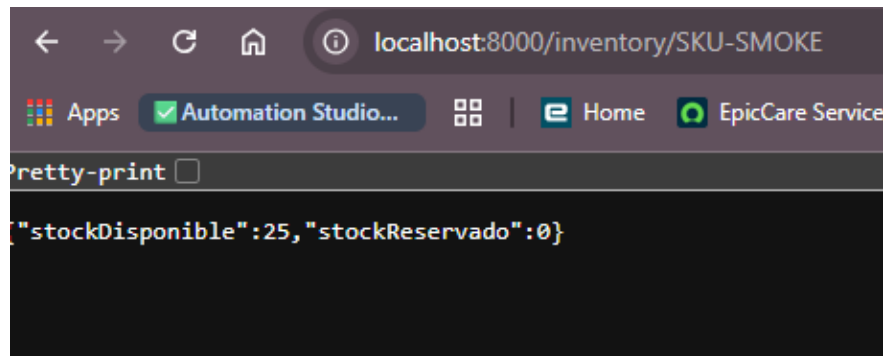


Figura 4.6: Respuesta real de GET /inventory/SKU-SMOKE: el Golden Record refleja el movimiento de inventario propagado desde el WMS a través de la cadena ERP → MDM.

4.2.6. Reflexión sobre los resultados obtenidos

Los tres experimentos verifican los criterios de aceptación, pero el veredicto por sí solo no explica *por qué* la arquitectura se comporta como lo hace ni qué se cedió para conseguirlo. Esta sección lee los resultados en términos de atributos de calidad y tácticas de arquitectura (Bass et al., 2021): qué propiedad quedó efectivamente bajo prueba en cada experimento, mediante qué mecanismo se obtuvo y a costa de qué. El supuesto que organiza la lectura es que ninguna de las decisiones del Capítulo 3 fue neutral: cada una privilegió unos atributos sobre otros, y el valor de la evaluación está tanto en lo que confirma como en los compromisos que deja a la vista.

Tabla 4.4: Atributos de calidad, tácticas aplicadas y costo asumido.

Atributo	Táctica aplicada	Evidencia	Costo asumido
Consistencia del inventario	Reserva atómica serializada por clave de partición en el ERP	E1: sobreventa cero en tres escenarios; E3: 18 de 18 réplicas	Disponibilidad de la ruta de confirmación: el ERP queda como punto único de fallo
Desempeño (oportunidad)	Propagación incremental de deltas en lugar de <i>snapshot</i> completo (P3)	E2: mediana 32,3 ms, máximo 106,6 ms	Estado materializado duplicado entre ERP y <i>Golden Record</i>
Modificabilidad	Intermediario <i>publish-subscribe</i> que elimina el acoplamiento punto a punto (P1)	Incorporación de un canal por suscripción a vistas canónicas	Inversión previa en modelo canónico y gobernanza
Disponibilidad (propagación)	DLQ, reintentos con <i>backoff</i> , idempotencia y <i>circuit breaker</i> (P5)	Diseñada; no ejercitada por los experimentos	Complejidad operativa y parámetros pendientes de calibración
Observabilidad	<i>Event Log</i> inmutable del ciclo pedido-entrega (P4)	<code>orders.lifecycle</code> con hitos fechados por pedido	Retención creciente y deberes de protección de datos

Consistencia frente a disponibilidad: por qué se priorizó la primera. Este es el compromiso central del trabajo y el que explica el resultado de E1. Ante una partición o una degradación del ERP, la arquitectura debe elegir entre aceptar el pedido contra una vista posiblemente desactualizada —privilegiando disponibilidad— o rechazarlo hasta poder confirmar la reserva —privilegiando consistencia—. El diseño propuesto opta por lo segundo, y la justificación no es técnica sino económica: la línea base diagnóstica permite poner precio a cada error. Una inconsistencia en la reserva deriva en una venta que no se puede cumplir y desencadena el ciclo nota crédito-refacturación, con un ticket promedio de COP \$330.000 y dieciocho ocurrencias mensuales documentadas; una indisponibilidad transitoria de la ruta de reserva, en cambio, se traduce en un pedido que se confirma algunos segundos más tarde, sin costo contable ni afectación de la promesa de entrega. La asimetría entre ambos costos es de varios órdenes de magnitud y es lo que ordena la prioridad.

Conviene precisar el alcance de esa decisión, porque no se trata de una elección global de la

arquitectura sino de una elección por operación. Únicamente la reserva de inventario se resuelve con consistencia fuerte: es la operación donde un error produce un compromiso comercial irreversible. Todo lo demás —la consolidación del *Golden Record*, la propagación a las vistas de canal, la actualización de los consumidores— opera bajo consistencia eventual y privilegia disponibilidad, de modo que la caída de un canal no detiene a los otros ni bloquea al ERP. La arquitectura es, en consecuencia, mayoritariamente disponible y puntualmente consistente, y el punto de sincronía se reduce a la única transacción que lo exige. Un desacoplamiento llevado hasta sus últimas consecuencias habría situado también la reserva sobre vistas eventualmente consistentes, que es precisamente el mecanismo que produce la sobreventa del modelo actual: la sobreventa cero de E1 no es una propiedad emergente de la arquitectura orientada a eventos, sino el efecto de no haber desacoplado la decisión que exige consistencia fuerte.

El costo de esa prioridad es explícito y conviene no minimizarlo. El ERP permanece en la ruta crítica de todo pedido, lo que lo convierte en un punto único de fallo para la confirmación: si deja de responder, los pedidos no se pierden —quedan retenidos en el *topic* y la Saga los retoma— pero sí dejan de confirmarse mientras dure la indisponibilidad. A ello se suma que los experimentos se ejecutaron sobre una única instancia del ERP, de manera que el comportamiento de la contención bajo la concurrencia de la operación real queda sin medir. Atenuar ese costo sin renunciar a la consistencia exigiría replicar la autoridad de reserva o introducir un mecanismo de reserva anticipada por canal con cupos acotados, alternativas que desplazan el problema pero no lo eliminan y que quedan señaladas como línea de trabajo posterior.

Lo que la latencia mide y lo que no. La mejora en un factor superior a 10^4 respecto de la línea base no constituye un hallazgo de desempeño sino una consecuencia del cambio de modelo de propagación: la latencia del esquema *batch* es estructural —está determinada por el intervalo de sincronización— mientras que la del esquema por eventos es de procesamiento. Las cifras medidas (mediana de 32,3 ms, máximo de 106,6 ms) sostienen una afirmación más modesta y más útil: la arquitectura no introduce sobre costo apreciable sobre el tiempo de propagación. En producción, la red, el cifrado en tránsito, los conectores reales y la contención elevarán esos valores, y por esa razón el SLA de oportunidad se fijó en un minuto y no en milisegundos. El margen de tres órdenes de magnitud es lo que da holgura a esa traslación.

Resiliencia diseñada frente a resiliencia verificada. La disponibilidad es el atributo con la mayor distancia entre lo diseñado y lo evidenciado. Los cuatro mecanismos descritos en la Sección 3.2 responden al hallazgo H4, pero ninguno de los tres experimentos indujo un fallo: no se detuvo un consumidor, no se llenó una *Dead Letter Queue* ni se activó un *circuit breaker*. Los parámetros de reintento y los umbrales del interruptor se declaran como valores iniciales de práctica estándar sujetos a calibración. En consecuencia, las propiedades de resiliencia se apoyan en las garantías documentadas del *broker* y en el diseño, no en medición propia; verificarlas exige un plan de pruebas de inyección de fallos que este trabajo no ejecutó y que queda señalado como trabajo futuro.

Saga orquestada frente a coreografiada. La elección de un coordinador central privilegió la observabilidad y la facilidad de diagnóstico: el estado de la transacción de negocio queda en un único lugar y cada hito es consultable, que es lo que exigía el hallazgo H7 sobre ausencia de trazabilidad. La variante coreografiada habría eliminado ese componente central, pero al precio de

dispersar el estado del pedido entre los servicios participantes y de hacer que reconstruir el ciclo completo requiriera correlacionar eventos de varias fuentes (Richardson, 2018). El costo asumido es que el *Order Orchestrator* concentra la lógica del flujo y se convierte en un componente cuya propia disponibilidad debe garantizarse: un fallo suyo no corrompe el estado —el *Event Store* permite reconstruirlo— pero sí detiene el avance de las sagas en curso.

La modificabilidad. Incorporar un canal nuevo mediante suscripción a las vistas canónicas resulta barato únicamente después de que el modelo canónico y la asignación de responsabilidades existen. El beneficio en modificabilidad no proviene del *broker* sino del acuerdo previo sobre qué significa cada entidad y quién responde por ella; el *broker* solo lo hace operativo. Esto explica por qué la capa de gobernanza no es accesoria respecto de la capa tecnológica: sin reglas de prevalencia y sin responsables por dominio, la infraestructura de eventos se limitaría a propagar datos inconsistentes con mayor velocidad, sustituyendo un problema de latencia por uno de calidad.

Observabilidad frente a custodia del dato. El registro inmutable que cierra H7 genera obligaciones en sentido contrario. Los eventos del ciclo pedido-entrega transportan atributos de identificación del cliente, de modo que la retención prolongada que da valor a la trazabilidad entra en tensión con los principios de minimización y finalidad que exige la normativa colombiana de protección de datos personales (Congreso de la República de Colombia, 2012). El diseño difiere el cifrado en reposo y la gestión centralizada de secretos a una evolución posterior, lo que significa que esta tensión queda reconocida pero no resuelta en la presente fase.

En síntesis, los resultados sostienen que la combinación de gobernanza, datos maestros y arquitectura orientada a eventos es técnicamente viable y que las tácticas aplicadas producen los efectos que se les atribuyen bajo condiciones controladas. No sostienen afirmaciones sobre desempeño a escala productiva, sobre disponibilidad ante fallos reales, ni sobre el ahorro financiero como valor realizado, que permanece como proyección anclada en la tasa de prevención medida. Delimitar con precisión qué atributo favoreció cada decisión y qué se cedió a cambio es lo que permite que la arquitectura propuesta se discuta y se revise en lugar de adoptarse como un conjunto cerrado de elecciones tecnológicas.

Conclusiones

5.1. Conclusiones

Este trabajo abordó la ausencia de una arquitectura de integración omnicanal en tiempo cercano al real en una empresa manufacturera del sector de consumo masivo, cuyos síntomas (sobreventa sistémica, discrepancias diarias de inventario y devoluciones por inconsistencia) fueron cuantificados en una línea base de tres indicadores (TPFI del 20 %, EI inferior al 90 % y COMI de COP \$3,02 millones mensuales). Las conclusiones se presentan como respuesta a cada objetivo específico.

El diagnóstico de la arquitectura actual, que respondía al primer objetivo, permitió identificar y triangular las fuentes de latencia e inconsistencia: integraciones punto a punto centralizadas en el ERP, sincronización batch con frecuencias de 15 minutos a 24 horas, propagación por *snapshots* completos y ausencia total de mecanismos de recuperación automática. Los siete hallazgos convergentes (H1–H7), validados por stakeholders de cuatro áreas funcionales, establecieron que el problema no es únicamente tecnológico sino también de gobernanza: el ERP opera como fuente de verdad de facto sin diseño formal ni responsables de datos documentados.

En materia de gobernanza –segundo objetivo–, se definió un framework basado en DAMA-DMBOK que formaliza los roles de *Data Owner*, *Data Steward* y *Data Consumer* para los tres dominios críticos, junto con SLAs de calidad del dato anclados en los umbrales operativos levantados en el diagnóstico y una política de deduplicación de clientes en tres niveles que reemplaza la corrección manual reactiva vigente.

El tercer objetivo se tradujo en el diseño de un Modelo de Datos Maestros bajo la estrategia *Consolidation Hub*, con atributos canónicos y reglas de prevalencia explícitas por entidad. La decisión de co-masterización, con el ERP como autoridad del stock disponible y de las reservas y el MDM como capa canónica de consolidación y propagación, resolvió la tensión entre consistencia transaccional y desacoplamiento, y constituye una de las contribuciones de diseño más relevantes del trabajo.

Sobre la arquitectura de integración, cuarto objetivo del proyecto, el enfoque orientado a eventos sobre Apache Kafka, con Saga orquestada para el ciclo pedido–entrega y cuatro mecanismos de resiliencia (DLQ, reintentos con *backoff* exponencial, idempotencia de consumidores y *circuit breakers*), da respuesta trazable a cada una de las brechas del diagnóstico, como lo documenta la matriz de trazabilidad hallazgo–componente.

Finalmente, la evaluación del prototipo funcional –quinto objetivo– verificó los tres criterios de aceptación definidos en la línea base (resultados detallados en la Sección 4.2): la reserva atómica en el ERP eliminó la sobreventa en los tres escenarios de concurrencia evaluados (sobreventa igual a cero frente a la sobreventa sistemática del modelo batch simulado); la propagación de movimientos

de inventario al *Golden Record* y a las vistas de canal se completó en menos de un minuto en las cinco rondas de prueba, con una exactitud de inventario del 100 % en el entorno controlado; y la Saga de compensación previno la causa raíz de las devoluciones por inconsistencia en el 100 % de las réplicas, lo que proyecta una reducción del COMI del 64.3 % respecto a la línea base, superando el objetivo del 50 %.

En síntesis, el trabajo demuestra que la combinación de gobernanza formal de datos, un MDM con reglas de prevalencia explícitas y una arquitectura orientada a eventos es técnicamente viable en el contexto de una manufacturera latinoamericana de consumo masivo, y que dicha combinación ataca las causas estructurales, no los síntomas, de la ineficiencia logística omnicanal. La articulación simultánea de estos tres planos en un caso aplicado con validación cuantitativa constituye la contribución que la revisión de literatura identificó como vacante.

5.2. Trabajos futuros

El alcance delimitado del proyecto deja abiertas líneas de trabajo naturales. La puesta en producción de la arquitectura exige abordar el cifrado en reposo y la gestión centralizada de secretos, declarados como evolución posterior en el framework de gobernanza, así como la integración real con los sistemas comerciales (Epicor, Cegid, VTEX, Cubic) mediante conectores y la migración progresiva de los canales externos que hoy operan con archivos planos hacia el API Gateway. A nivel funcional, extender el MDM a dominios excluidos del alcance (precios dinámicos y pronóstico de demanda) permitiría cerrar el ciclo completo de la promesa de entrega. Ya en operación, una medición longitudinal (30 a 90 días) permitiría validar los KPIs financieros con datos contables efectivos, calibrar los parámetros de resiliencia declarados como valores iniciales de práctica estándar y contrastar la exactitud de inventario contra conteos físicos sistematizados. Por último, la incorporación de analítica sobre el *Event Store*, con aplicaciones como la detección de anomalías de inventario y la asignación dinámica de despacho sugeridas por el área financiera, abre una línea de investigación aplicada hacia la Logística 4.0 predictiva.

5.3. Lecciones aprendidas

La lección más clara fue metodológica: anclar cada decisión arquitectónica en evidencia diagnóstica trazable, y no en preferencias tecnológicas, transformó la conversación con los stakeholders, pues cada componente del diseño puede defenderse mostrando el hallazgo que lo origina. La matriz de trazabilidad hallazgo-componente resultó ser tan valiosa como herramienta de comunicación organizacional como lo es de rigor académico.

A esto se suma una lección de diseño: la pregunta crítica en una arquitectura de datos distribuida no es qué tecnología adoptar sino dónde reside la autoridad de cada dato. La decisión de co-masterización del inventario (autoridad en el ERP, consolidación en el MDM) surgió de confrontar el diseño inicial con la operación real de la empresa, y evitó un error estructural: haber asignado la reserva de stock a vistas eventualmente consistentes habría reproducido la sobreventa que se buscaba eliminar.

Desde el punto de vista organizacional, la gobernanza de datos operativa (roles asignados a cargos reales, SLAs derivados de umbrales que el negocio ya reconoce) tiene mayor probabilidad de adopción que los marcos documentales genéricos. El alineamiento espontáneo entre el diagnóstico técnico interno y la solución propuesta redujo la resistencia organizacional y confirmó que la validación temprana con stakeholders es un factor crítico de éxito.

A título personal, el proyecto consolidó la competencia de traducir un problema de negocio difuso —“el inventario no cuadra”— en un problema de ingeniería especificable, medible y validable, que es en esencia el oficio del arquitecto de software.

Bibliografía

- Amazon Web Services (2025). Amazon simple queue service developer guide. <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/>. Consultado en 2026.
- Apache Software Foundation (2025). Apache kafka documentation. <https://kafka.apache.org/documentation/>. Consultado en 2026.
- Bass, L., Clements, P., and Kazman, R. (2021). *Software Architecture in Practice*. Addison-Wesley, Boston, MA, 4th edition.
- Batra, P. and Kewalramani, C. (2024). Implementing omnichannel strategies for global customer engagement: A project and change management approach. *International Journal of Computer Trends and Technology*, 72(4):63–70.
- Brown, S. (2018). The C4 model for visualising software architecture. <https://c4model.com>. Consultado en 2026.
- Congreso de la República de Colombia (2012). Ley estatutaria 1581 de 2012: por la cual se dictan disposiciones generales para la protección de datos personales. Diario Oficial No. 48.587, Bogotá.
- DAMA International (2017). *DAMA-DMBOK: Data Management Body of Knowledge*. Technics Publications, 2nd edition.
- Ellithy, K., Salah, M., Fahim, I. S., and Shalaby, R. (2024). Agv and industry 4.0 in warehouses: a comprehensive analysis of existing literature and an innovative framework for flexible automation. *International Journal of Advanced Manufacturing Technology*, 134:15 – 38.
- Hohpe, G. and Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley.
- Hübner, A., Hense, J., and Dethlefs, C. (2022). The revival of retail stores via omnichannel operations: A literature review and research framework. *European Journal of Operational Research*, 302:799 – 818.
- Kleppmann, M. (2017). *Designing Data-Intensive Applications*. O'Reilly Media.
- Loshin, D. (2009). *Master Data Management*. Morgan Kaufmann / OMG Press.
- Mahadevan, K. and Joshi, S. (2022). Omnichannel retailing: a bibliometric and network visualization analysis. *Benchmarking*, 29:1113 – 1136.
- Moreira, A., Alves, C., Machado, J., and Santos, M. F. (2023). An overview of omnichannel interaction in health care services. *Mayo Clinic Proceedings: Digital Health*, 1:77 – 93.

-
- RabbitMQ (2025). Rabbitmq documentation. <https://www.rabbitmq.com/docs>. Consultado en 2026.
- Richardson, C. (2018). *Microservices Patterns*. Manning Publications.
- Suárez Murcia, N. L. (2019). Marco arquitectural para la adopción de entornos omnicanal. Master's thesis, Universidad de los Andes.
- Yang, Z. and Hu, D. (2024). Digital technology-empowered omnichannel integration: a review and research agenda. *International Journal of Retail and Distribution Management*, 52:407 – 424.