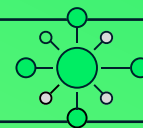




Database Digest

ISSUE 02



La capa de
inteligencia
unificada:
impulsando la era
de los agentes



© 2026 MongoDB, Inc.
Todos los derechos reservados

Publicado desde 2025
por MongoDB,
1633 Broadway, piso 38
Nueva York, NY 10019, EE. UU.

Editores colaboradores:

Boris Bialek, Sebastián Rojas Arbulu,
Daniel Jamir

Creatividad de marca:

Andrew Considine, Hans Bennewitz

Agradecimientos

Soluciones para la industria: Taylor Hedgecock, Gabriela Preiss, Genevieve Broadhead, Peyman Parsi, Humza Akhtar, Francesc Mateu, Jeff Needham, Benjamin Lorenz, Thorsten Walther, Rama Gabbita, Ken Wiebke, Diego Albano, Ainhua Música, Raphael Schor

Relaciones con desarrolladores: Apoorva Joshi, Franck Pachot, Mikiko Bazeley

Educación: Raghu Viswanathan, Rita Rodrigues, Thalita Carrico, Emily Pope

Marketing de producto: Jad Jarouche

Marketing corporativo global: Joanne Johnson, Laila McDonald, Kevin O'Rourke

Arquitecto de soluciones: Diana Annie Jenosh, Ashwin Gangadhar

Marketing internacional: May Petry, Penta Stanley, Cathy Gallego, Ilana Matro

Centro de Demanda Global: Steve Jurczak, Jack Yallop, Brandon Lee

Carta del editor



Boris Bialek
vicepresidente,
Soluciones para
la industria
en MongoDB

En mis 35 años en este negocio, nunca había visto nada parecido a la situación actual. Por lo general, algunos sectores toman decisiones importantes mientras que otros observan y piensan si tienen que actuar. Esta vez no es el caso. Los bancos, la automoción, los seguros y la manufactura, entre otros sectores, se mueven al mismo tiempo, y la mayoría de ellos (como me dijo un cliente) lo están apostando todo.

Eso ha favorecido la toma de algunas decisiones de arquitectura extrañas. A principios de este año, me reuní con un cliente cuya prueba de concepto para un flujo de trabajo con un solo agente especializado involucraba a 14 proveedores diferentes integrados. El cliente estaba bastante orgulloso. Le pregunté, con sutileza, cómo pensaba ponerlo en producción. Me miró como si la pregunta no se le hubiera ocurrido y dijo: “Todavía no hemos llegado a esa parte”. Le mostré un competidor que ya tenía la producción en marcha después de ocho semanas. Se quedó sin palabras.

Esa pequeña anécdota personal es, a mi parecer, el momento determinante de la IA de agentes en 2026: un equipo inteligente descubre que ha compilado una demo preciosa que no puede convertirse en un negocio.

Nuestro presidente y director ejecutivo, CJ Desai, se pronunció en MongoDB.local Londres en mayo de 2026 diciendo: “La parte más difícil de operar agentes en producción no es el modelo. Es la capa de datos subyacente”. Ninguna de las personas presentes precisó más explicaciones.

Tenía motivos para ser tan tajante. Gartner prevé que el 40 % de los proyectos con IA de agentes se cancelarán para 2027. Según Deloitte, solo el 11 % de las empresas utiliza sistemas de agentes en un entorno de producción, lo que significa que el 89 % aún no ha dado ese paso. El motivo, casi siempre, no es el modelo. Es todo aquello que lo subyace: seguridad incorporada al final, registros de auditoría que nadie había previsto, datos en tiempo real que residen en cuatro sistemas a la vez y llegan al agente medio retrasados.

Considere la alternativa. El mayor distribuidor farmacéutico de Estados Unidos mueve 1200 millones de contenedores al año, y detrás de cada uno de esos contenedores hay una persona: un paciente, esperando el medicamento adecuado en la dosis adecuada en el día adecuado. A la Ley de Seguridad de la Cadena de Suministro de Medicamentos no le importa cuán sofisticado es su modelo. Lo que importa es que pueda realizar el seguimiento de cualquier contenedor, de principio a fin, en tiempo real, en todo momento. Hace unos años, McKesson tomó una decisión arquitectónica discreta y poco atractiva: retirar una pila heredada monolítica y, en su lugar, operar sobre una plataforma de datos unificada. Sus operaciones se multiplicaron por 300. Siguen velando por la seguridad del paciente. Y ese logro no es de un agente. Es de lo que subyace.

Defendemos la necesidad de esa base sólida durante el resto de la edición. La sección 01 mide el costo real de la fragmentación y la alternativa que McKesson eligió. En la sección 02 se aborda la alternativa a la fragmentación: la capa de inteligencia unificada aplicada a los sectores en los que ofrece un retorno de la inversión más rápido. En la sección 03 se aborda cómo lograr que los agentes que se encuentran sobre la base sean realmente precisos, y un ejemplo de LG U+ aplicado en la producción. En la sección 04, llegamos al punto culminante: el cambio de los sistemas de registro a los sistemas de acción.

Quiero concluir con un momento que me ha marcado. El día en que su equipo puso en marcha el proyecto, el CTO de una gran cadena minorista me dijo que el proyecto que habíamos desarrollado juntos había quedado obsoleto. Por un momento, me desilusionó. Pero luego se explayó. “Boris, está obsoleto porque está en funcionamiento”, dijo. “Los datos eran MongoDB. Esa es mi base. Puedo reemplazar el LLM y el modelo en cualquier momento. Puedo sustituir el marco superior. Los datos son míos”.

Lo que hemos aprendido al dar estos pasos importantes es que las decisiones arquitectónicas que se toman ahora perdurarán más que las herramientas de IA para las que fueron elegidas. Por eso, la capa de datos es la decisión más crucial.





SECCIÓN UNO

Cuando la IA supera a la pila

El costo oculto de la arquitectura fragmentada.

p. 6

SECCIÓN DOS

La capa de inteligencia unificada

Una sola plataforma para el ciclo de vida completo de recuperación.

p. 15

SECCIÓN TRES

Precisión a escala

Fundamentar los agentes de IA en la verdad operativa en tiempo real.

p. 23

SECCIÓN CUATRO

Construyendo la era de la IA

De los sistemas de registro a los sistemas de acción.

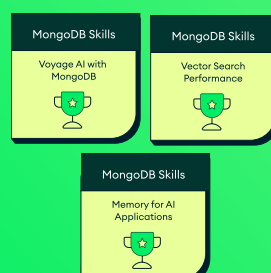
p. 31



LIBRO DESTACADO

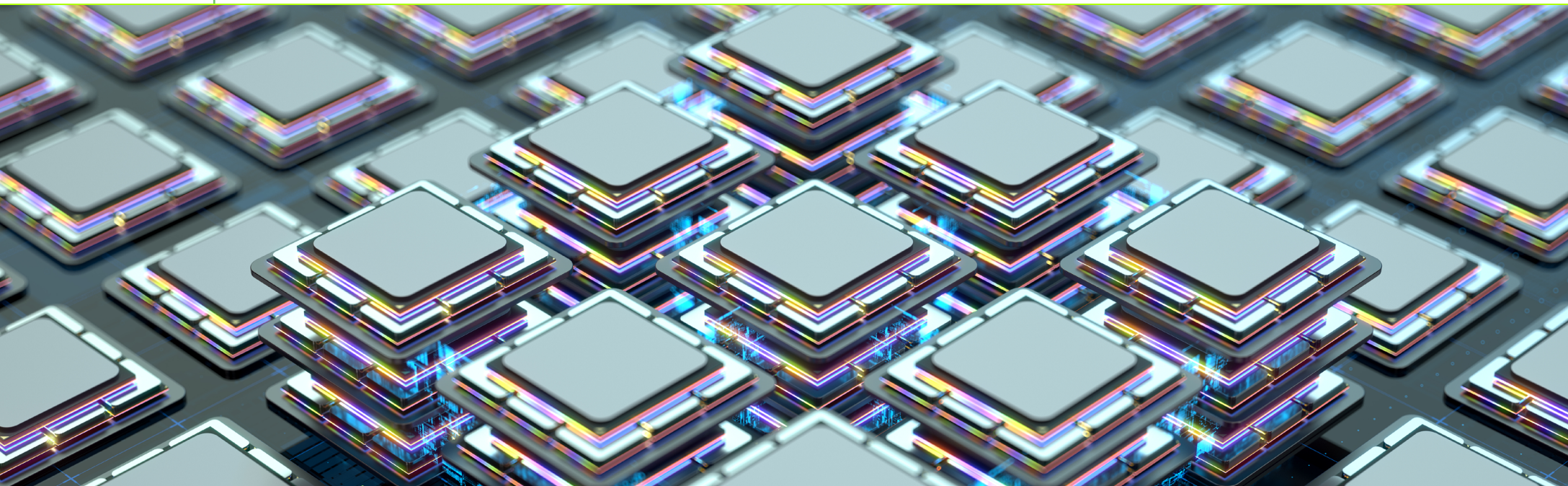
Architectures for the Intelligent AI-Ready Enterprise

Marcos de trabajo prácticos para construir arquitecturas preparadas para la IA que ofrecen un valor empresarial duradero.



APRENDIZAJE Y CERTIFICACIÓN

Tres nuevas insignias de habilidades de IA de MongoDB: Voyage AI with MongoDB, Vector Search Performance y Memory for AI Applications.



SECCIÓN UNO

Cuando la IA supera a la pila

El costo oculto de la arquitectura fragmentada

Dos empresas inician el mismo proyecto de IA el mismo día.

Al final del trimestre, una de ellas ya ha puesto en marcha un agente operativo en el entorno de producción. Se fundamenta en los datos operativos de su empresa. Tiene memoria entre sesiones. Pasó la revisión de seguridad. El costo es predecible.

Dieciocho meses después, el segundo sigue en fase piloto. El agente trabaja en el entorno de demostración. Alucina en la preproducción. El repositorio vectorial y el repositorio operativo han empezado a desviarse, y nadie sabe quién es responsable de ese problema. El departamento de compras pregunta si la factura de la nube es correcta. El departamento legal pregunta cómo se puede auditar algo que nadie puede describir bien.

El mismo talento. Los mismos modelos. El mismo presupuesto. La única variable es la arquitectura inicial.

Esa brecha ya tiene un nombre. Se denomina **sobrecarga arquitectónica** y se trata del peso acumulado que una pila fragmentada impone a todos los equipos que intentan implementar la IA sobre ella. [Un nuevo estudio](#) de International Data Corporation (IDC), encargado por MongoDB y realizado entre 1400 empresas de ocho mercados de Asia-Pacífico, cuantifica este fenómeno. El 43 % de las empresas afirma que su arquitectura actual supone un obstáculo importante para desarrollar nuevas aplicaciones sin una modernización a gran escala. Los sistemas son demasiado rígidos, demasiado costosos y demasiado lentos. Y el costo de no hacer nada se agrava: IDC prevé que las organizaciones que no aborden la deuda técnica se enfrentarán a una tasa de fracaso de la IA un 50 % mayor para 2027.

Esa tasa de fracaso no es un problema de calidad del modelo. Es lo que ocurre cuando los agentes se topan con una pila que se ha creado para recuperar datos y no para actuar. Los chatbots leen. Los agentes leen, deciden, escriben, realizan transacciones y cambian de estado, y luego escriben algo más, porque cada decisión que toma un agente deja un rastro. Registros de decisiones. Estados intermedios. Registros de auditoría. Memoria del agente que debe persistir en la siguiente sesión. La sobrecarga es lo que ocurre cuando a una pila ensamblada para responder preguntas se le pide hacer todo eso a la vez.

Eso afecta a cuatro factores al mismo tiempo. El almacén de vectores no puede guardar, pero el agente tiene que cambiar de estado. El almacenamiento operativo y el almacenamiento de vectores no pueden compartir una transacción,

pero la acción del agente tiene que ser atómica o se completa a medias: se cobra al cliente, pero el pedido no se registra. La sincronización entre ellos se retrasa, y el agente decide sobre la verdad de ayer. Y cuando un regulador pregunta qué hizo el agente, el registro de auditoría cubre cuatro de los seis pasos porque ningún sistema vio la secuencia completa.

La diferencia entre ambas empresas no es el talento ni el presupuesto. Es que una de ellas modernizó la capa de datos mientras desarrollaba el agente. La otra intentó hacerlo después, y perdió un año por trabajos de integración que deberían haberse hecho antes.

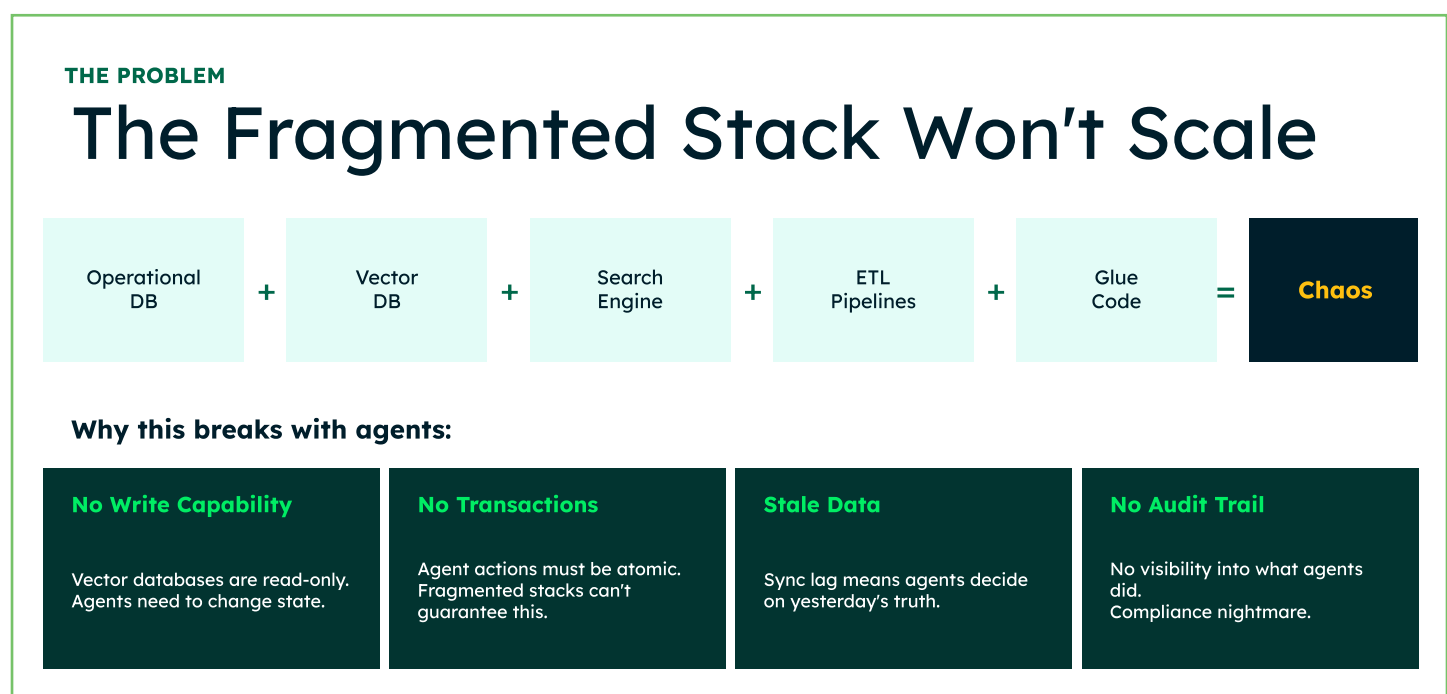
No aparece como un ítem. Se traduce en una hoja de ruta más lenta, revisiones de seguridad más laboriosas y proyectos de IA que desaparecen discretamente de la siguiente revisión trimestral.



“La empresa quiere moverse rápido, pero los sistemas subyacentes se niegan. Lo que debería tardar dos semanas, tarda seis meses.”

Thorsten Walther *director general, CXO Advisory Asia en MongoDB*

Figura 1.
Por qué falla la pila fragmentada con los agentes



**LEER LA
INVESTIGACIÓN
DE IDC**

El resto de esta sección analiza el problema en dos partes.

Primero, el diagnóstico. Por qué las arquitecturas divididas (almacenes vectoriales y operacionales separados unidos mediante pipelines de extracción, transformación y carga [ETL]) imponen un costo que solo se muestra a gran escala. Por qué el mismo desajuste aparece a nivel del modelo de datos cada vez que las cargas de trabajo de documentos se ven forzadas a motores relacionales. Y un marco de decisión honesto para

elegir entre Postgres y MongoDB, de un arquitecto que ha pasado 30 años en ambos lados de ese debate.

Luego, un ejemplo de una empresa que lo abordó. McKesson mueve 1200 millones de contenedores de medicamentos al año y decidió desde temprano que la capa de datos era la parte de la pila que valía la pena tratar correctamente. Lo que construyeron y lo que cambió porque lo construyeron es el estudio de caso que cierra esta edición.

Arquitectura estratégica de bases de datos para IA: unificada o dividida

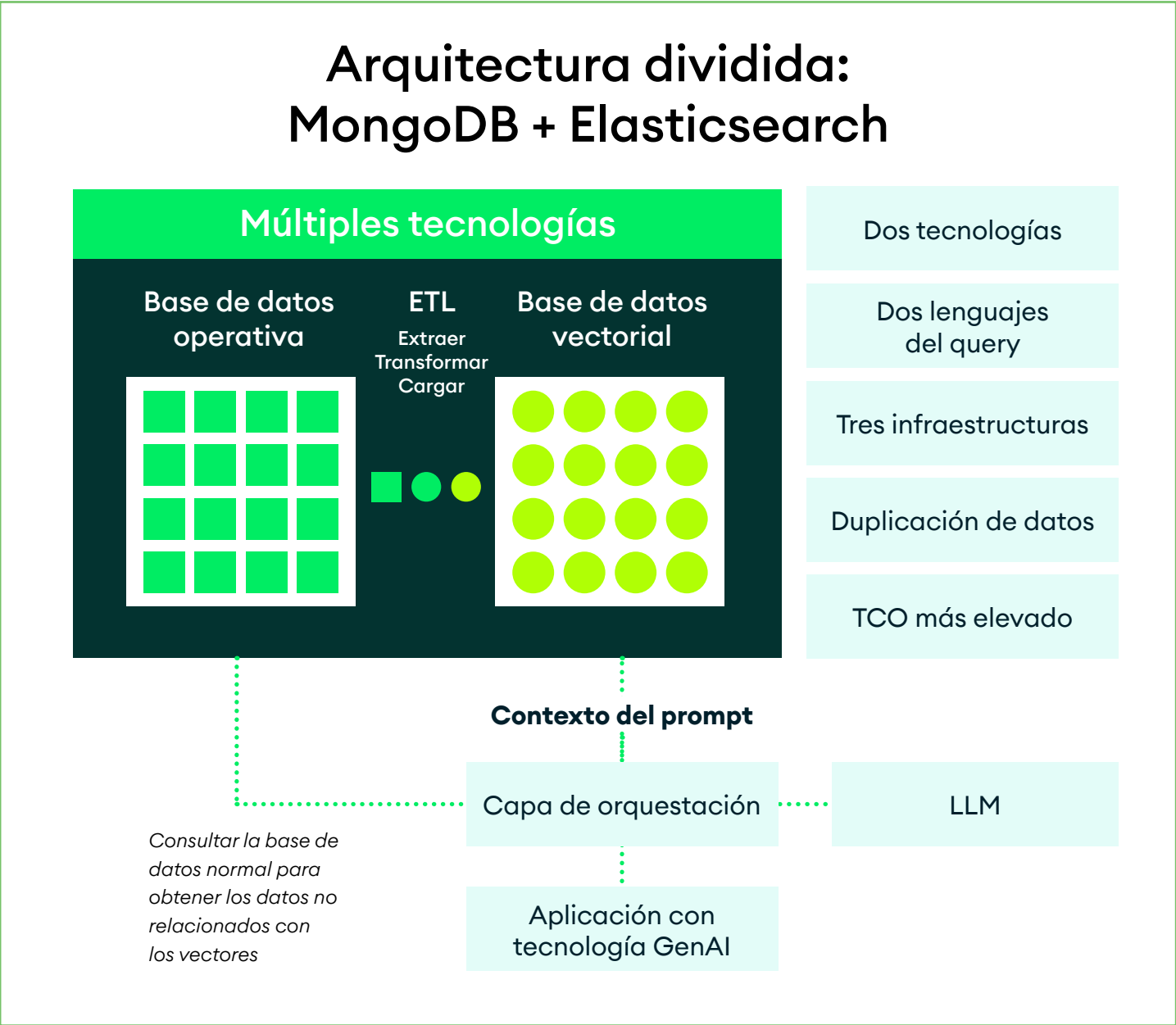
Cada revisión de arquitectura para un proyecto de IA de agentes termina con la misma pregunta: ¿la pila que ya tenemos puede soportar lo que estamos a punto de construir? Resulta tentador responder a esta pregunta comparando el rendimiento de las bases de datos vectoriales. La cuestión más profunda es si su arquitectura puede responder a preguntas sobre datos y significado en una misma query, con las mismas garantías.

Eso es lo que un agente realmente necesita. Es la parte de la pila que la mayoría de los equipos colocan en último lugar, después de que el resto del sistema ya haya tomado forma. La base de datos operativa se da por sentada. Según [Harvard Business Review Analytic Services](#), solo el 15 % de las empresas consideran que su fundamento de datos está lista para adoptar la IA de agentes. El resultado es una arquitectura dividida que

parece razonable en una pizarra y empieza a desmoronarse en producción.

Si somos honestos, vale la pena comparar los dos patrones. En una arquitectura unificada, una sola plataforma gestiona juntos los datos operativos y la búsqueda vectorial. En una arquitectura dividida, un almacén de vectores dedicado (Pinecone, Weaviate o un motor de búsqueda como Elasticsearch) se encuentra junto a la base de datos operativa, con un pipeline ETL que mantiene los dos sincronizados. Una combinación común, la cual se muestra en la Figura 2, es MongoDB como sistema de registro para documentos de productos y Elasticsearch como motor de búsqueda vectorial para la incrustación vectorial (embedding). Ambos funcionan en una demostración. Sin embargo, las diferencias solo aparecen al escalar.

Figura 2.
Arquitectura dividida:
base de datos operativa
MongoDB + almacén
vectorial Elasticsearch



Los verdaderos modos de fallo del enfoque dividido son predecibles. La complejidad de la sincronización crece con cada campo nuevo. Cuando un documento se elimina en el almacén operativo, la incrustación vectorial correspondiente puede permanecer como un documento fantasma, devolviendo resultados que ya no existen. Las transacciones atómicas no son una opción porque dos sistemas no pueden compartir un límite de transacción.

Los costos operativos se acumulan de forma igual de silenciosa. La velocidad de desarrollo se ve afectada porque la mayor parte del esfuerzo de ingeniería se destina al código de reconciliación más que a las funcionalidades. Estrategias de copia de seguridad separadas, procedimientos de conmutación por error separados, pilas de supervisión separadas y rotaciones de guardia separadas. Y cada mes llega la factura: dos

tecnologías de bases de datos, dos lenguajes de consulta, tres componentes de infraestructura que hay que proteger y pagar, datos duplicados y un mayor costo total de propiedad.

El caso unificado no es abstracto. Cuando el documento y su incrustación coexisten, una actualización es una sola escritura atómica, un borrado remueve ambos, y los mismos controles de acceso protegen a ambos. Las plataformas como MongoDB Atlas, que adoptan el enfoque unificado, utilizan nodos de búsqueda dedicados para aislar cargas de trabajo vectoriales del nivel operativo para que puedan escalar de forma independiente. La unificación no significa renunciar al margen de rendimiento que exigen las cargas de trabajo vectoriales. El costo total de propiedad refleja esa simplicidad: se paga una plataforma, se asegura una plataforma, se opera una plataforma.

Figura 3.
Arquitectura de división:
el costo oculto

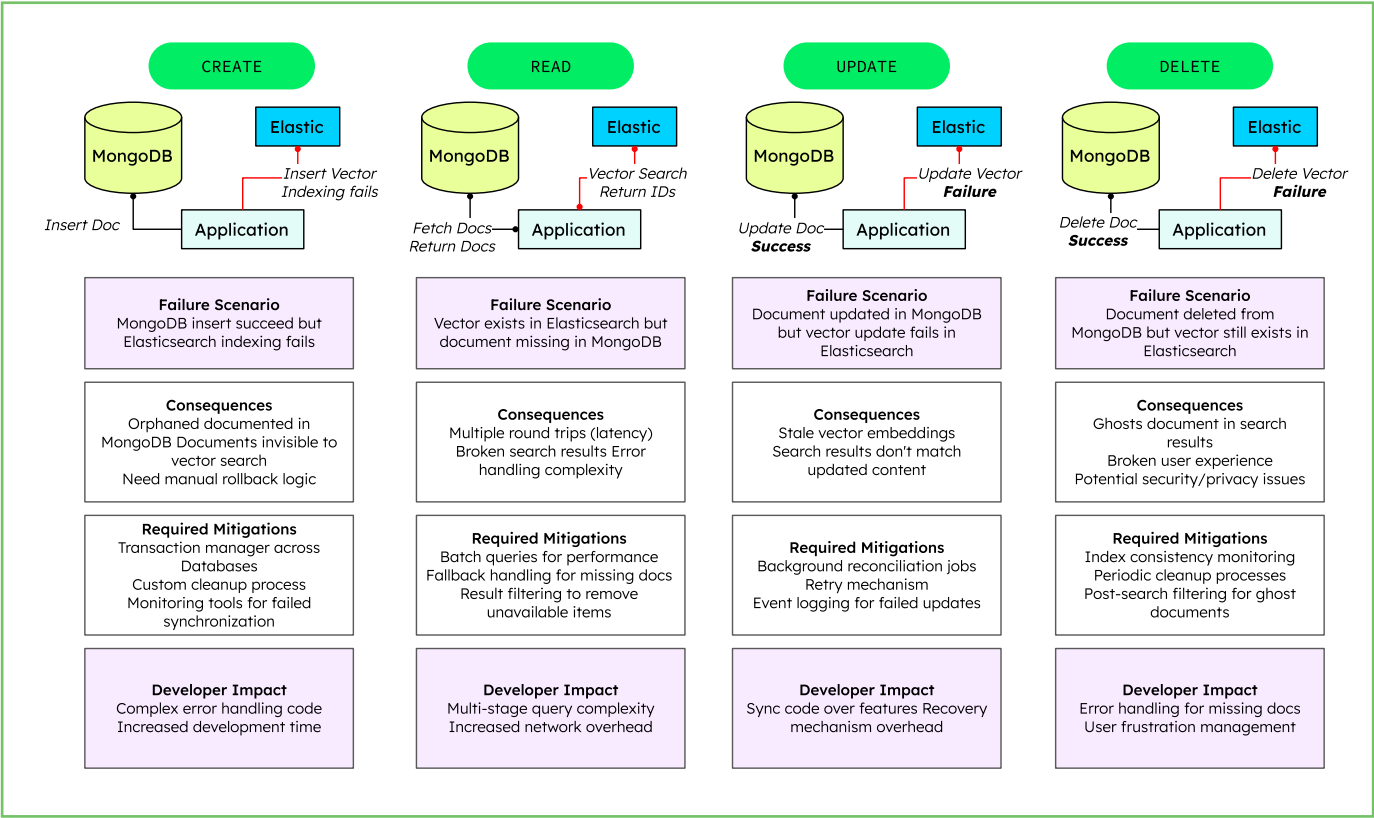
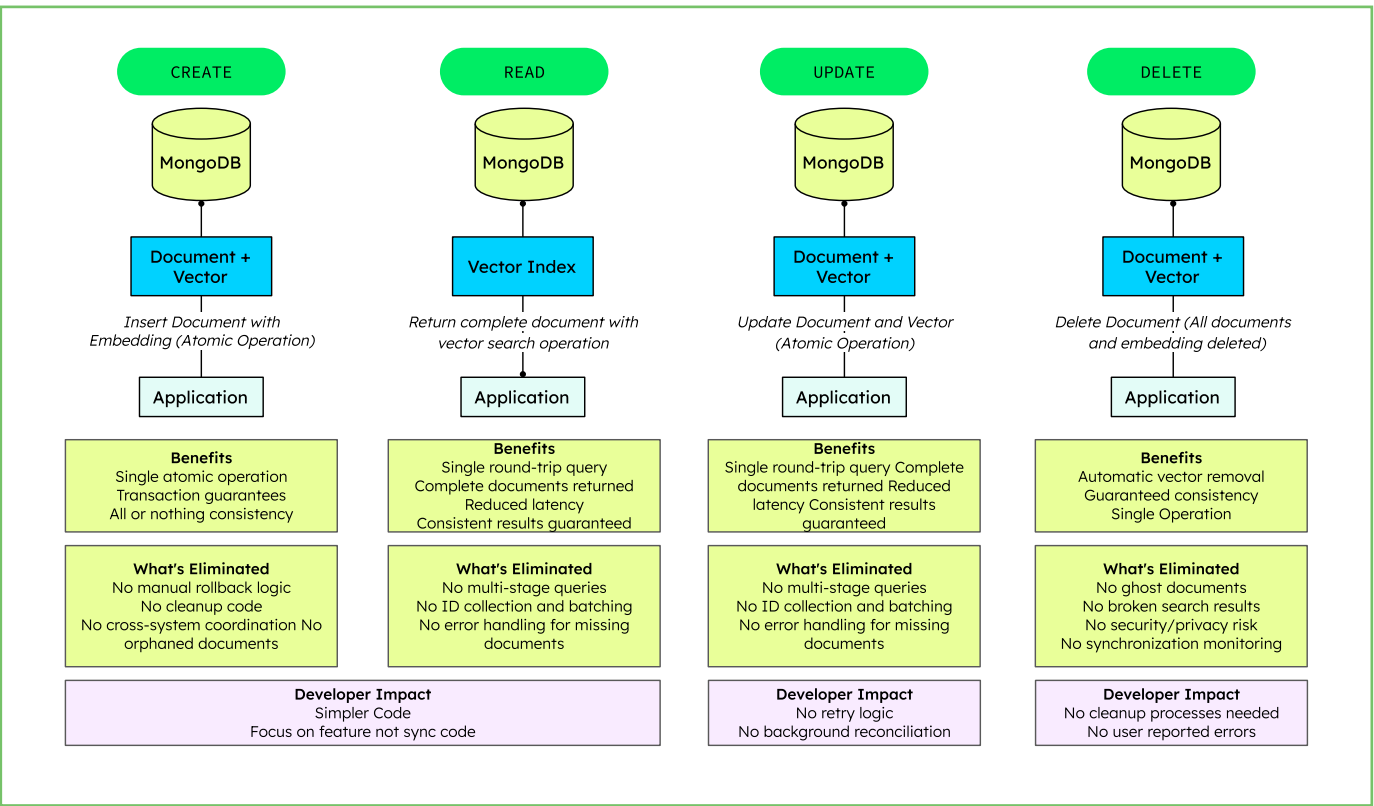


Figura 4.
Operaciones CRUD
en una arquitectura
unificada:
MongoDB Atlas con
búsqueda vectorial.



LEER EL
ARTÍCULO



MongoDB o SQL: sin mitos

La pila web moderna habla un lenguaje: JSON (JavaScript Object Notation), el formato de texto ligero que los navegadores, las API y las aplicaciones móviles utilizan para intercambiar datos. El navegador envía JSON. La API procesa objetos JavaScript. La aplicación móvil espera JSON. Colas de mensajes, WebSockets, microservicios: todo JSON. Elija una base de datos cuyo modelo de datos coincida con el resto de la pila, y desaparecerá toda una categoría de trabajo. Elija una que no lo haga y su equipo se pasará la carrera traduciendo entre formatos. Ese es el argumento que defiende Tim Carter Clausen, quien escribe [The Decipherist](#) y ha operado MongoDB en producción durante una década, en un artículo que merece leerse completo.

Cada pocos meses, una entrada de blog conocida se vuelve viral: “¿Por qué volvimos a Postgres?” Los hilos de comentarios se llenan de los mismos argumentos. *MongoDB no escala. No existen joins. Las transacciones son débiles.* La respuesta de Clausen no es repetir la habitual postura neutral sobre las fortalezas de ambas bases de datos. Se trata de abordar cada crítica por separado, con 10 años de datos de producción de respaldo.

Hay dos temas con los que se identificarán los responsables técnicos que evalúan una decisión sobre la pila tecnológica en 2026.

El primero es el pipeline JSON. Una solicitud SQL típica realiza ocho cambios de formato en un solo viaje de ida y vuelta:

1. El cliente envía JSON.
2. La API lo convierte en un objeto JavaScript.
3. Una herramienta de mapeo relacional de objetos (ORM) descompone ese objeto en filas dispersas en las tablas normalizadas.
4. La base de datos hace su trabajo.
5. Las filas se vuelven a ensamblar.
6. Se asignan de nuevo a un objeto.
7. El objeto se serializa de nuevo a JSON.
8. Se envía la respuesta.

El equivalente de MongoDB son cuatro transformaciones, y Clausen sostiene que llamarlas cuatro es exagerar porque la forma de los datos en realidad nunca cambia. Cada cambio de formato en la ruta SQL supone un impacto en la CPU, la memoria y la latencia, además de constituir un lugar donde puede ocultarse un error. Nada de esto hace que el producto sea mejor. Todo tiene que seguir funcionando de todos modos.

El segundo es la evolución del esquema. Cambiar el nombre de un campo, añadir un objeto anidado o reestructurar un documento ocurre en tiempo real en MongoDB sin interrupciones. La misma operación en una tabla relacional de gran tamaño puede bloquear las escrituras durante varios minutos, a veces incluso horas. Para equipos con envíos semanales, la diferencia no es teórica. Es la diferencia entre añadir un campo esta tarde y planificar un periodo de mantenimiento para el siguiente trimestre.

El diagnóstico de Clausen es directo. El equipo que culpa a MongoDB de los fallos suele ser el que lo creó como SQL: colecciones modeladas como tablas, Mongoose utilizado como ORM, llamadas a `find()` esparcidas por los archivos, nada desnormalizado, nada indexado. Luego, una publicación de 3000 palabras que afirma que la base de datos es el problema. “No lo es”, escribe. “Lo son”.

Nada de esto es un argumento para considerar que los motores relacionales están obsoletos. Siguen siendo la respuesta adecuada para muchas cargas de trabajo, y las grandes empresas las ejecutarán durante décadas. La cuestión es más específica. Cuando la carga de trabajo tiene estructura JSON, está orientada a documentos y se evoluciona cada semana, las suposiciones bajo el diseño relacional dejan de ser neutrales. Esto plantea una pregunta diferente: ¿los motores relacionales modernos realmente cierran esa brecha, o solo la cubren? Postgres con soporte nativo de JSON es el ejemplo principal.



**LEER EL
ARTÍCULO**

¿Postgres o MongoDB?

El árbol de decisiones de un arquitecto

Hay una objeción obvia: los motores relacionales modernos ahora soportan columnas JSON, índices vectoriales y campos de arreglo. Sin duda, una implementación de Postgres con JSONB y pgvector puede llevar a una empresa a la era de la IA sin forzar una migración de la capa de datos, ¿verdad? La pregunta es pertinente y merece una respuesta basada en criterios técnicos, más que en lealtades partidistas. Franck Pachot, un veterano de 30 años en Oracle, Postgres, YugabyteDB y MongoDB, además de AWS Data Hero y Oracle Certified Master, aborda exactamente esa pregunta en una charla que merece verse en su totalidad. Su enfoque no es partidista; de hecho, recomienda Postgres para algunas cargas de trabajo incluso desde un rol de MongoDB. Ese equilibrio es lo que hace que valga la pena tomarlo en serio.

Dos observaciones técnicas desplazan la comparación de la preferencia a la arquitectura. La primera es lo que sucede a un documento de 10 MB en cada motor. Si lo inserta en MongoDB, un seguimiento de Linux muestra llamadas de escritura de 10 MB cada una, porque el documento permanece unido como una hoja de B-tree. Si lo inserta en Postgres, el mismo seguimiento muestra escrituras de 8 KB, porque Postgres divide el documento en bloques de 8 KB usando TOAST. La segunda es lo que realmente significa “una sola tabla”. Una tabla de Postgres que contiene una columna JSON no es una sola tabla en realidad. Son dos tablas y dos índices: la tabla principal con su índice de llave primaria y una tabla de fragmentos TOAST con su propio índice. La recuperación de un documento de gran tamaño es, a nivel mecánico, un join por bucle anidado entre ambos.

Figura 5.

La realidad física de una tabla de Postgres que contiene una columna JSON. Una instrucción CREATE TABLE, cuatro objetos internamente: una tabla de tuplas, una tabla de fragmentos TOAST y un índice para cada una.

Physical data
locality

MongoDB clusters data physically

PostgreSQL splits it to multiple tables, multiple indexes

```
create table "MongoDB" (  
  _id UUID primary key  
    default uuidv7(),  
  data JSONB  
);  
  
CREATE TABLE  
  
select amname, count(*)  
from pg_class c join pg_am a on c.relam=a.oid  
where c.oid >= 'MongoDB'::regclass  
group by amname;
```

```
1 select c.oid, relkind, amname, relname from pg_class c  
2 join pg_am a on c.relam=a.oid  
3 where c.oid >= 'MongoDB'::regclass  
4 order by c.oid;
```

oid	relkind	amname	relname	
16384	r	heap	MongoDB	table for tuples
16389	t	heap	pg_toast_16384	table for toasts
16390	i	btree	pg_toast_16384_index	index for toasts
16391	i	btree	MongoDB_pkey	index for tuples

CONTINÚA EN LA PÁGINA SIGUIENTE



Esa misma divergencia se observa en la capa de indexación. Consideremos una consulta con la que está familiarizado cualquiera que opere un sistema transaccional: mostrar los últimos 10 pedidos de un producto determinado en un país determinado. En el modelo orientado a documentos, un único índice compuesto cumple esta función directamente, incluyendo los campos anidados dentro de los arreglos. En un sistema relacional que contiene los mismos datos que JSON, el equivalente suele requerir un índice GIN para el contenido del arreglo, un índice B-tree separado para los campos escalares y un ordenamiento que el planificador no puede evitar. El plan lee más filas de las necesarias y su escalabilidad empeora a medida que el conjunto de datos crece. Nada de esto es visible desde la capa de aplicación. Todo eso se refleja en la factura de la nube.

La perspectiva estructural que subyace a ambos puntos es la localidad de los datos. En el modelo orientado a documentos, el modelo lógico y el modelo físico son lo mismo. La forma que escribe la aplicación es la forma que almacena la base de datos y la forma que almacena la base de datos es la forma que la capa de recuperación devuelve a un agente de IA. Esa equivalencia no es un eslogan. Es lo que permite que los datos, el índice y la ruta de recuperación funcionen como un todo, sin necesidad de los pipelines de sincronización, la lógica de doble escritura ni las tareas de conciliación que, de otro modo, las arquitecturas

fragmentadas necesitarían para aproximarse al mismo resultado.

El marco de decisión de Pachot se orienta hacia las observaciones técnicas. Una base de datos central que sirve a muchas aplicaciones, donde aún no se conocen todos los casos de uso, es una tarea relacional, y Postgres es la respuesta relacional adecuada. Una única aplicación construida a partir de un modelo de dominio y almacenada de forma persistente a medida que la aplicación razona sobre ella es un trabajo orientado a documentos, y ahí es donde MongoDB consolida su posición. Ambos motores son respuestas adecuadas a diferentes preguntas. La siguiente pregunta que merece la pena hacerse es cuál describe el trabajo que se está desarrollando hoy. ¿Se trata de microservicios con contextos delimitados, esquemas que se actualizan cada semana u objetos de dominio almacenados tal y como los concibe la aplicación?

Vale la pena llevar la línea de cierre de Pachot a cualquier conversación sobre bases de datos en 2026: “Una base de datos es buena y rápida si se usa de forma correcta, y lo más importante es elegir una que conozca o que quiera aprender”.

La responsabilidad vuelve a donde pertenece: al arquitecto, no al motor. La siguiente sección muestra cómo es esa responsabilidad cuando la carga de trabajo es la seguridad del paciente a escala de 1200 millones de contenedores al año.



**VER
LA CHARLA**



McKesson cumple con las normativas a gran escala con MongoDB

Los argumentos de arquitectura son abstractos hasta que el mayor distribuidor de medicamentos de Estados Unidos apuesta por la seguridad del paciente.

Cuando la Ley de Seguridad de la Cadena de Suministro de Medicamentos de los Estados Unidos entró en vigor, McKesson se enfrentó a un mandato no negociable: rastrear más de 1200 millones de números de serie al año en toda su cadena de suministro, y cada escaneo autentica un contenedor hasta el fabricante, en tiempo real. Su infraestructura actual, compuesta por SAP y Postgres, tenía dificultades para ofrecer la flexibilidad y la escalabilidad que exigía la normativa.

El planteamiento del arquitecto principal de TI de McKesson, Brian Schmidt, fue particularmente operacional: los datos que acompañan a cada contenedor de medicamento son tan críticos como el medicamento mismo. Los medicamentos que llegan sin datos verificables y entregables no pueden entregarse al paciente que los necesita.

McKesson eligió MongoDB como base para dos nuevos sistemas. El **Repositorio Central de Datos** gestiona los datos de números de serie de los 350 000 clientes a los que McKesson presta servicio a diario. El **Repositorio Distribuido de Números de Serie** es una aplicación distribuida que gestiona las solicitudes de verificación en toda la red de centros de distribución de McKesson. El modelo orientado a documentos resultaba ideal para los datos jerárquicos de la cadena de suministro, en los que un único envío puede contener docenas de productos, cada

uno con su propio historial de procedencia, lote, caducidad y cadena de custodia. Modelarlo en tablas planas habría supuesto un grafo de uniones que se habría convertido en un problema de latencia en sí mismo. Modelarlo como documentos significaba que la estructura de los datos reflejaba el objeto de negocio.

Lo que dice Schmidt sobre la puesta en marcha es la línea que merece ser la cita de la sección: “No hubo ni un parpadeo en el radar”. Para una de las 10 principales empresas de Fortune 500 que opera con un plazo de cumplimiento federal y que suministra medicamentos a un tercio del país, esta declaración no es un logro técnico menor. Es la diferencia entre una puesta en marcha regulatoria exitosa y un fracaso público con consecuencias para la seguridad del paciente.

La conclusión que cabe extraer de todo esto no es que la asistencia sanitaria sea especialmente difícil (aunque lo es). La cuestión es que el cumplimiento a escala y la agilidad arquitectónica no son fuerzas opuestas. McKesson se modernizó por el cumplimiento normativo, no por la IA. La base que construyó el equipo es la que ahora utilizan para explorar la IA generativa y la siguiente clase de casos de uso. La decisión arquitectónica que hizo posible el cumplimiento normativo a gran escala es la misma que hace posible la ejecución de la siguiente carga de trabajo.

“La escala que logramos con MongoDB es abrumadora. Es un hito que debe inspirar orgullo a todos.”

Upendra Kulkarni, *director de producto principal, McKesson*



[LEER LA HISTORIA](#)



[VER LA CONFERENCIA PRINCIPAL](#)

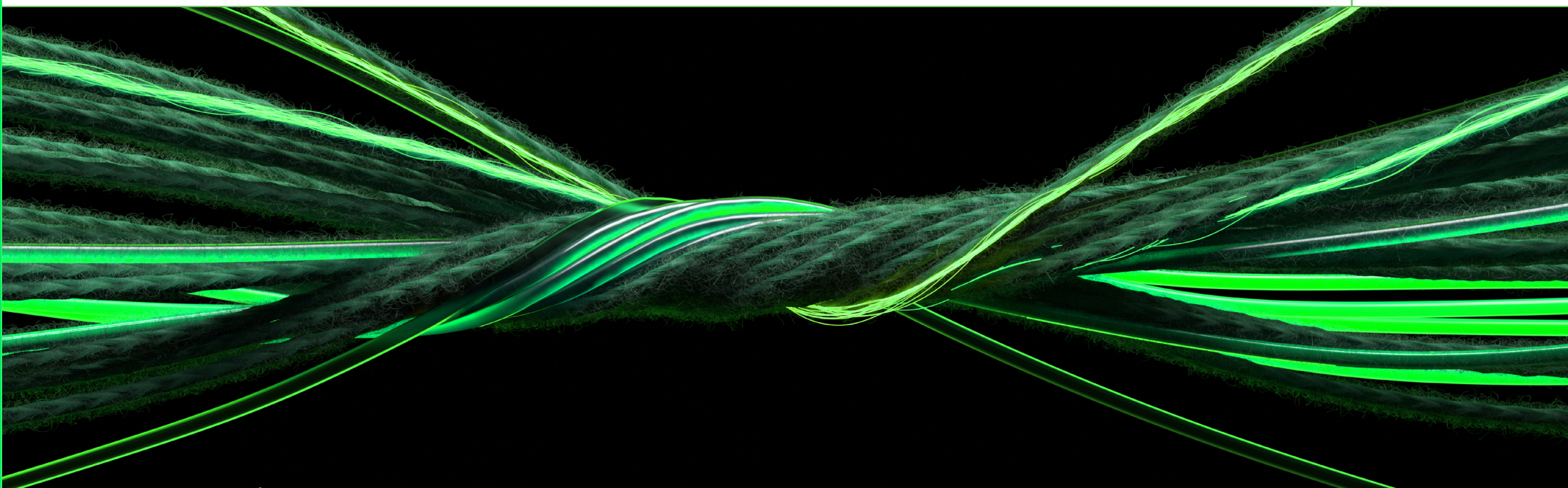
De la sobrecarga arquitectónica a la capa de inteligencia unificada

La Sección 01 fue el diagnóstico. La sobrecarga arquitectónica es real, la investigación de IDC lo valida, y McKesson muestra cómo se refleja la reducción de esta carga a escala. Sin embargo, un diagnóstico sin un plan de tratamiento es solo una auditoría.

1. ¿Cómo es, en realidad, una capa de inteligencia unificada?
2. ¿Qué cargas de trabajo tiene que absorber?
3. ¿Qué se siente al dirigir una empresa de este tipo cuando su negocio depende de agentes que actúan en producción todos los días?

En las próximas 10 páginas, responderemos a estas tres preguntas, analizando los sectores en los que este patrón unificado ofrece resultados más rápidamente: la industria manufacturera, la venta minorista, los servicios financieros y el sector de la automoción.

La sección 02 revela el plan de tratamiento. Una vez que se acepta que unir almacenes separados y adaptar las cargas de trabajo de documentos a tablas relacionales impone el mismo costo en diferentes monedas, las preguntas se vuelven prácticas:



SECCIÓN DOS

La capa de inteligencia unificada

Una sola plataforma para todo el ciclo de vida de recuperación.

Una capa de inteligencia unificada no es una idea. Es una lista de capacidades que tienen que estar en el mismo lugar. Esta sección las desglosa una a la vez, mostrando su importancia cuando un agente tiene que actuar sobre lo que lee.

El patrón arquitectónico en torno al que se basa este volumen tiene un nombre: **Converged Datastore para IA de agentes**. Las entidades sobre las que opera una

empresa, las incrustaciones sobre las que razonan sus agentes de IA y el estado que esos agentes acumulan a lo largo de las interacciones conviven en un mismo lugar: bajo una única API, un único lenguaje del query, y un único modelo de seguridad. Los datos operativos, la búsqueda vectorial, la búsqueda de texto completo, la generación de incrustaciones, la memoria del agente y el Stream Processing dejan de ser seis problemas de integración y se convierten en una única plataforma.

Los modelos seguirán mejorando. Los marcos de trabajo seguirán cambiando. La capa de datos es la parte que tiene que durar más que ambas. Si la diseña bien, los cambios de modelos y las migraciones de marcos de trabajo se convierten en refactorizaciones en vez de reconstrucciones. Si la arquitectura no es la adecuada, cada nueva ola de herramientas de IA le obligará a cambiar de plataforma.

La sección 01 expuso ese caso a nivel arquitectónico: dividida o unificada, con la sobrecarga arquitectónica que recae sobre quien optó por dividir. El mismo contraste se produce en la propia carga de trabajo de la IA, capacidad por capacidad.

MongoDB Data Platform



MCP



Database



Text



Vector



Embedding
models

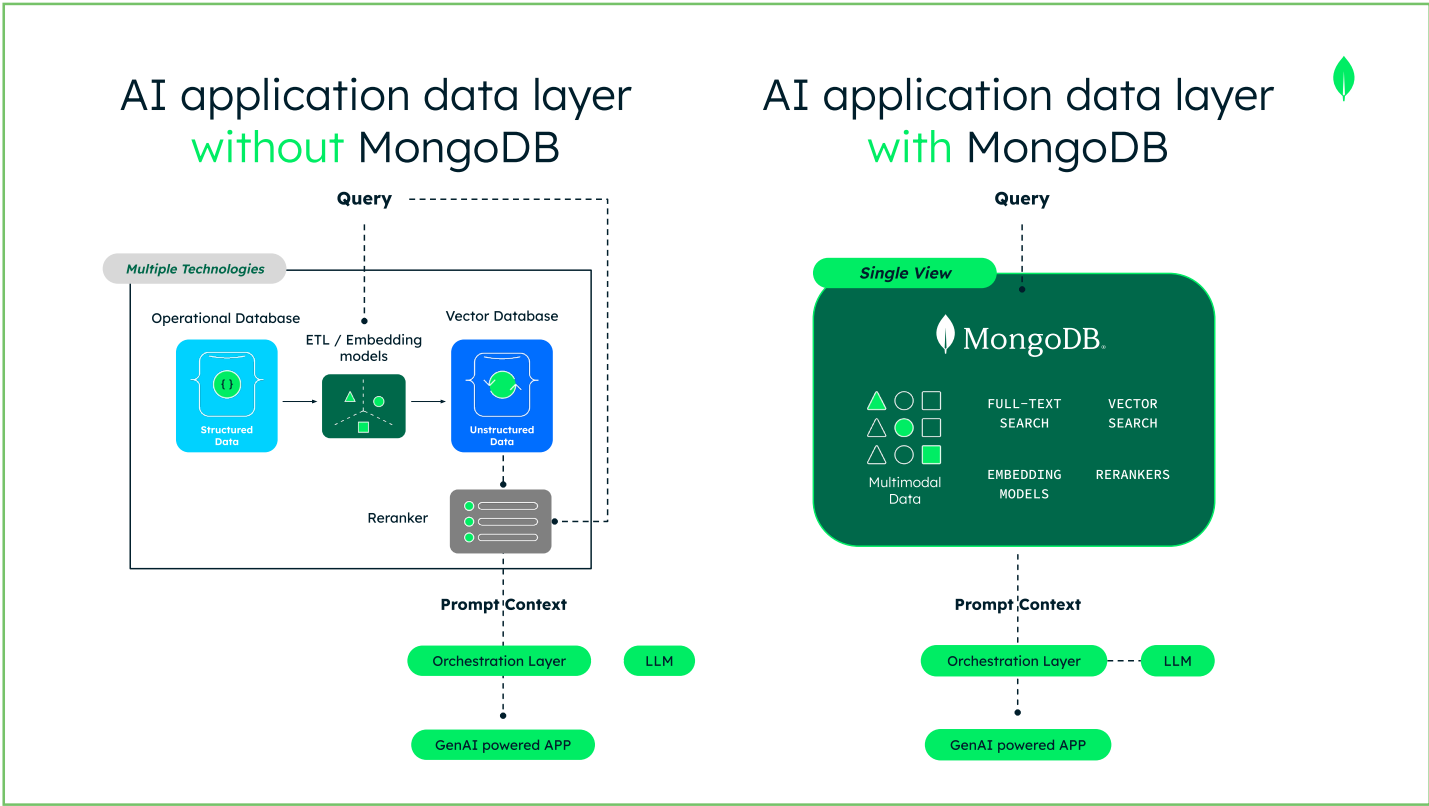


Rerankers



Local
dev

Figura 6.
La capa de datos de la aplicación de IA, antes y después de la consolidación. A la izquierda, varias tecnologías unidas con pipelines ETL: base de datos operativa, modelos de incrustación, base de datos vectorial, un reclasificador (reranker). A la derecha, las mismas capacidades nativas de un solo almacén.



A continuación, se recorren cuatro industrias. La IA en sí misma es independiente del dominio: un robot de soldadura que informa sobre la torsión, un investigador de fraudes que busca transcripciones, un técnico que compara un conector corroído con un árbol de fallos, un agente que recupera el historial de pedidos de un cliente. Distintos vocabularios, distintos usuarios, misma base subyacente. Así que en esta sección, se analizan los sectores de forma deliberada y se muestra qué cambia en la carga de trabajo y qué permanece igual.

Empezamos con la fabricación, donde el patrón de espacio de nombres unificado ya existe en la mayoría de las plantas de fábrica y solo requiere memoria para estar preparado para agentes. Nos trasladamos a la venta minorista, donde el comercio unificado resuelve un problema de silos de datos de hace una década y, de paso, hace

posible el agente de IA que gestiona las consultas de soporte. Luego, los servicios financieros, donde una plataforma que impulsa la búsqueda, la similitud semántica, el razonamiento de grafos y el aprendizaje automático lado a lado es la única respuesta honesta ante los delitos financieros a velocidad de máquina. Cerramos con la automoción, donde el diagnóstico posventa es uno de los pocos lugares en los que una capa de inteligencia unificada se amortiza en un solo ciclo de producto.

El hilo conductor de los cuatro es el mismo: la consolidación da sus frutos en ámbitos que parecen no tener relación entre sí hasta que se observa el diagrama de arquitectura.

La sección 01 concluyó con una pregunta: ¿Qué beneficio real obtiene por pagar la sobrecarga arquitectónica? La sección 02 responde a esa pregunta.



[LEER EL BLOG DE
CONVERGED DATASTORE](#)



[VER ENTREVISTA
DE REDMONK](#)

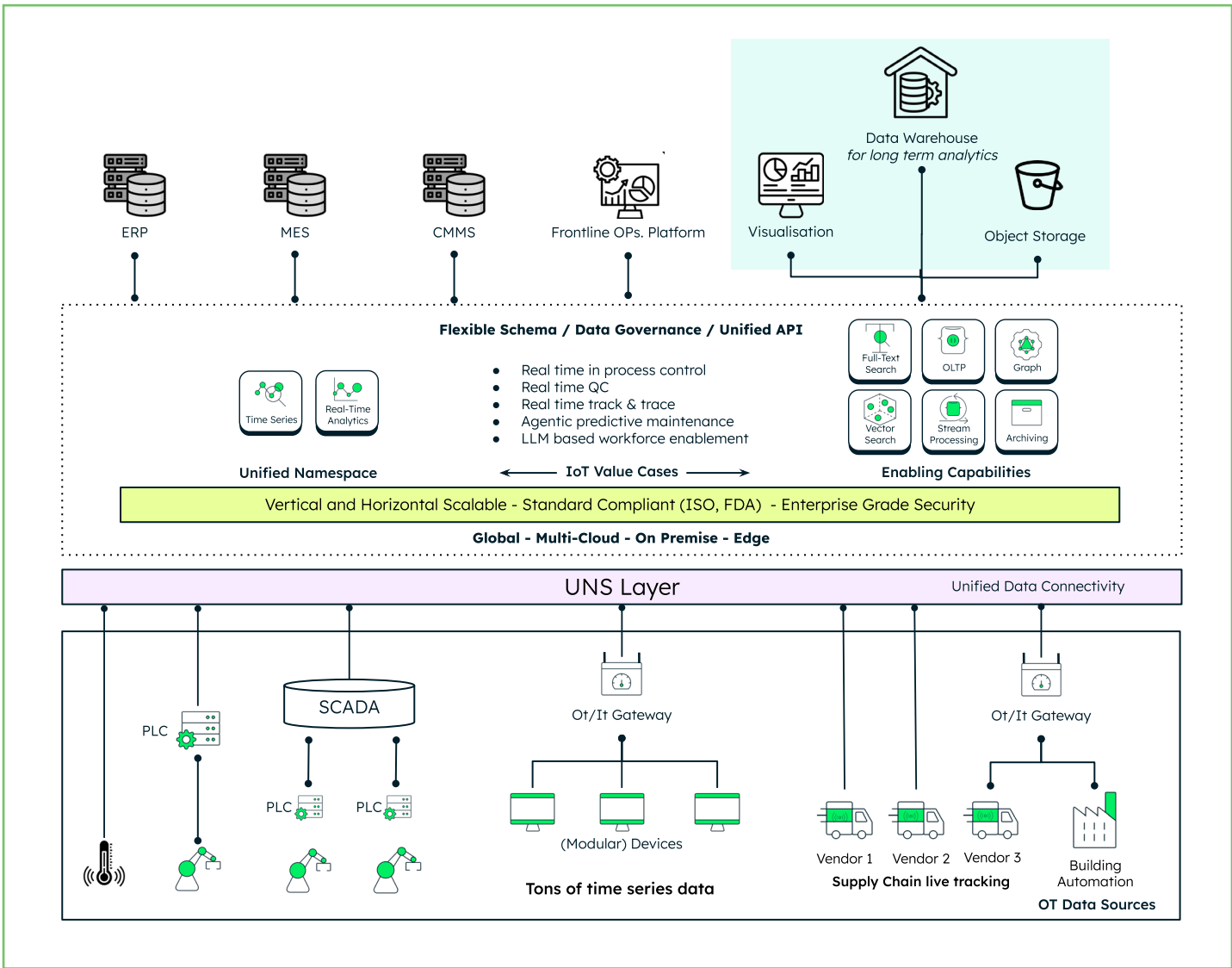
La capa de datos de fabricación necesita memoria para la IA de agentes

El patrón de Unified Namespace (UNS) unificado existe en la mayoría de las plantas de fabricación modernas, ya sea que alguien lo llame así o no: una jerarquía semántica única donde cada controlador, sensor y sistema de registro publica datos de producción en vivo, y cada suscriptor lee desde el mismo lugar. Es la respuesta arquitectónica a una generación de integraciones punto a punto que nadie quiere mantener. Pero el namespace por sí solo no llevará a los fabricantes a la era de agentes. Un namespace es transitorio. Los agentes necesitan memoria.

Los controladores lógicos programables (PLC), los pequeños ordenadores que ejecutan las máquinas individuales, junto con los sensores y

los sistemas de control de supervisión, publican en el namespace; el sistema de ejecución de fabricación (MES), la planificación de recursos empresariales (ERP) y los sistemas de calidad se suscriben a él. Eso es suficiente para operar una fábrica moderna. No es suficiente ejecutar un agente por encima. Un agente que responde a una deriva de tolerancia, programa un mantenimiento no planificado o coordina un cambio de línea no solo necesita el estado actual de la situación. Necesita incidentes previos, los datos de calidad de la semana pasada, las notas del operador del último turno y las incrustaciones de los manuales de mantenimiento que consultó la última vez. El namespace publica el presente. La memoria contiene el pasado.

Figura 7. Una arquitectura unificada que vincula los datos de tecnología operativa (OT) a través del namespace unificado con una plataforma de datos industriales escalable y en tiempo real.



Aquí es donde MongoDB extiende el namespace a una capa de datos con memoria. Los mismos documentos que contienen telemetría operativa también contienen incrustaciones, historial de conversaciones y el registro de auditoría de las decisiones de los agentes. La lectura de la torsión del robot de soldadura se encuentra junto a la orden de trabajo, el número de pieza, el operador y el resultado de calidad final. Así, cuando la línea de soldadura muestra una deriva de torsión a lo largo de tres turnos, el agente puede extraer cada soldadura previa con firmas de deriva similares, compararlas con los registros de mantenimiento y las notas del operador, y abrir una orden de trabajo de mantenimiento antes de la siguiente inspección fallida. Se indica el componente sospechoso, el técnico con la certificación adecuada y una ventana en el cronograma de producción en la que se puede detener la línea.

Las capacidades que lo hacen posible provienen del mismo motor. [Las colecciones de series de tiempo](#) gestionan los flujos de sensores de alta cardinalidad. [La búsqueda vectorial](#) gestiona la recuperación semántica en manuales e informes de incidencias. [Stream Processing](#) gestiona el enriquecimiento en tiempo real a medida que llegan los datos. En conjunto, estas capacidades convierten el namespace de una simple fuente de datos en una base sólida. La siguiente carga de trabajo de IA que la fábrica desee añadir se integrará en él, y no en uno nuevo.



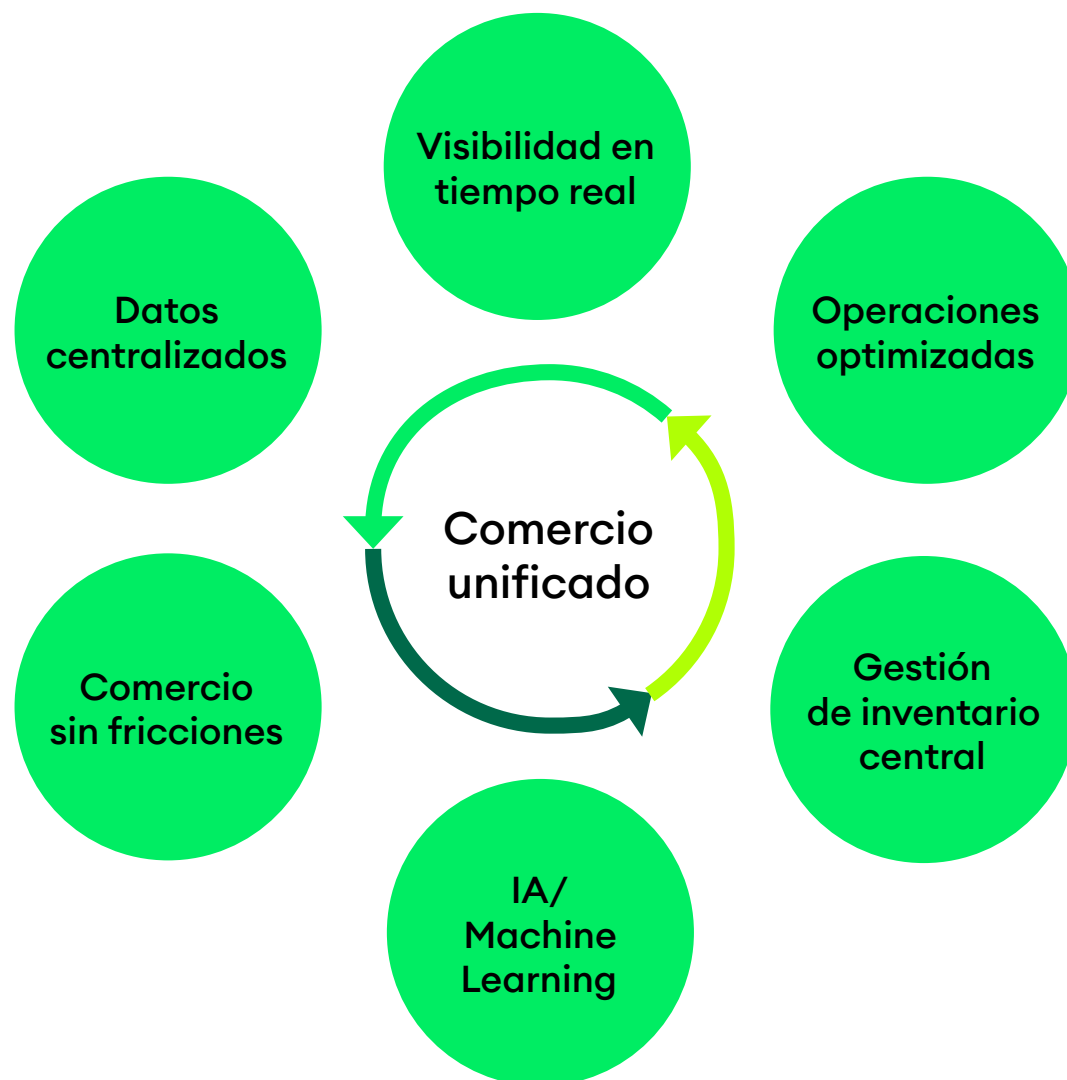
**DESCARGAR
EL INFORME
TÉCNICO**



Comercio unificado con búsqueda inteligente

Los equipos de venta minorista han dedicado la mayor parte de la última década a intentar ofrecer una experiencia unificada a través de la web, el móvil, la tienda física y el centro de atención telefónica. Los datos siempre estuvieron disponibles. Simplemente estaban dispersos en varios sistemas. El resultado era un cliente que podía ver un inventario en línea y otro diferente en la tienda, recomendaciones que se contradecían entre sí o una barra de búsqueda que devolvía resultados diferentes en la aplicación que en el escritorio.

El comercio unificado resuelve los desafíos aparentes mediante la integración de los datos subyacentes. La plataforma de datos MongoDB gestiona el catálogo de productos, inventario, precios, historial de pedidos y personalización, mientras que una capa de búsqueda inteligente combina precisión léxica, significado semántico y señales geoespaciales: la misma query que clasifica productos por relevancia también puede mostrar la tienda más cercana con stock.



**EXPLORAR LA
SOLUCIÓN**

Lo que se vuelve posible una vez que se unifican los datos es más interesante que lo que se resuelve. La búsqueda híbrida ofrece resultados que tienen en cuenta tanto la precisión de las palabras clave como el significado semántico. La personalización se basa en el historial multicanal sin necesidad de un proceso ETL nocturno. Y la misma capa de datos que alimenta

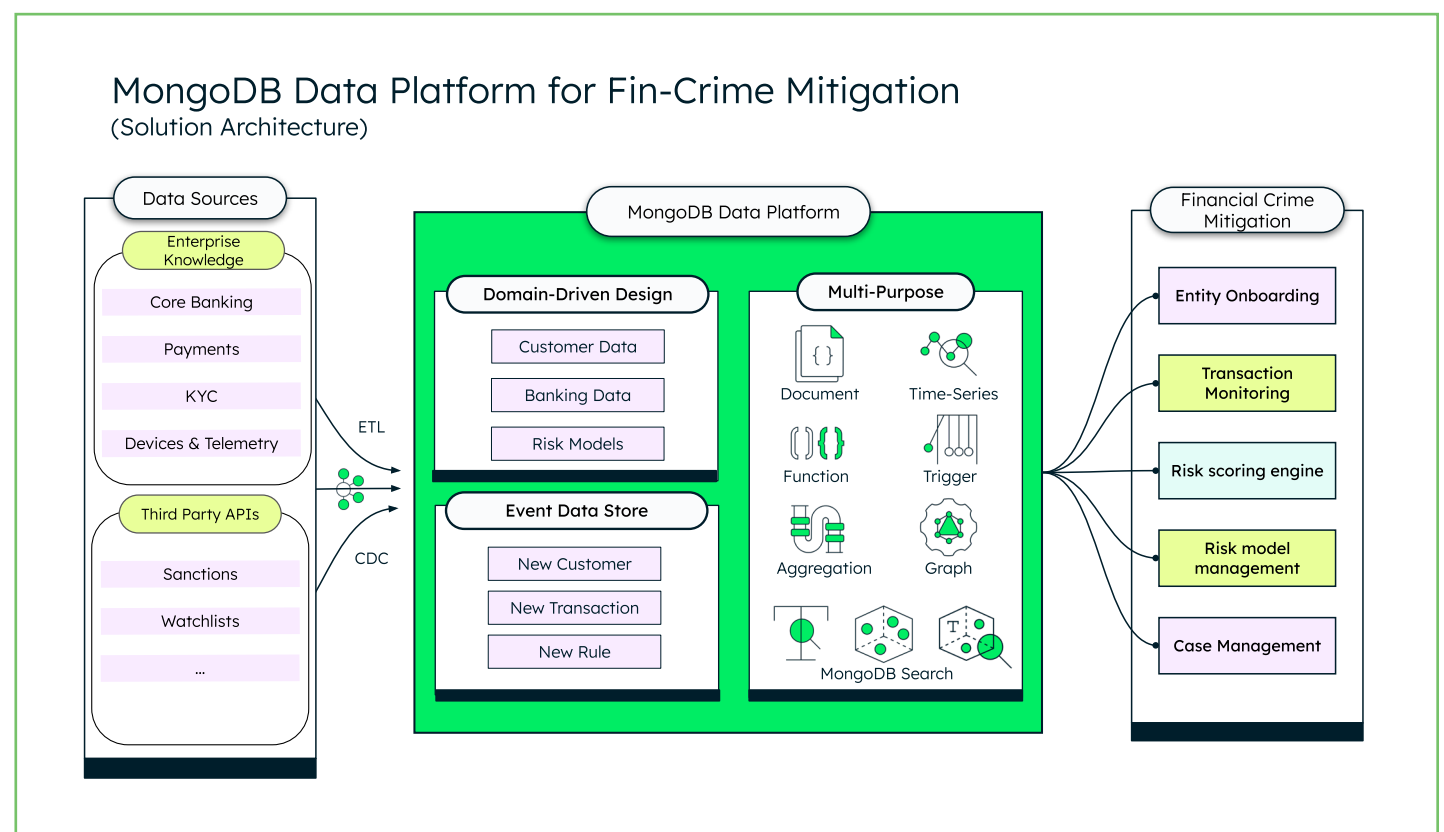
la interfaz de usuario impulsa un agente de IA que gestiona las consultas de soporte, porque solo hay una fuente de verdad que fundamenta ambos. El patrón que resulta beneficioso para la interfaz de usuario también lo es a velocidad de máquina, para cargas de trabajo como la detección de delitos financieros.

Desarrollo de una plataforma de mitigación de delitos financieros con MongoDB

La detección de delitos financieros siempre ha estado en la intersección de dos problemas de datos. El primero es estructurado: transacciones, jerarquías de cuentas, campos de Conozca a su cliente (KYC) y listas de sanciones. El segundo no es estructurado: notas de casos, transcripciones, documentos y el contexto extenso que explica por qué una transacción puede ser sospechosa. Durante décadas, los bancos han intentado resolver ambas mitades en sistemas separados y conciliarlos a nivel de alerta. Los investigadores han pagado el costo, más visiblemente en el 90 % de las alertas de los sistemas heredados que resultan ser falsos positivos que abruma a los equipos de cumplimiento mientras las amenazas reales pasan desapercibidas.

Los riesgos han superado a la arquitectura. Las instituciones financieras mundiales pierden más de 2 billones de dólares al año a causa de los delitos financieros, una cifra que se prevé que casi se triplique hasta alcanzar los 6 billones de dólares en 2030, y los adversarios que utilizan la IA ya operan a la velocidad de una máquina. La solución es una plataforma única que integre, en paralelo, la búsqueda, la similitud semántica, el razonamiento de grafos, las reglas y el aprendizaje automático. Los controles de acceso adecuados son la opción predeterminada de confianza para cualquier plataforma que gestione una cantidad tan elevada de datos sensibles. La búsqueda híbrida permite a los investigadores preguntar tanto “¿quién más ha movido dinero en este dispositivo en los últimos 90 días?” como “¿quién más ha utilizado un lenguaje similar al de este caso en sus transcripciones de asistencia?” desde la misma interfaz de query, contra la misma fuente de referencia.

Figura 8. Arquitectura de alto nivel de una plataforma unificada contra los delitos financieros con MongoDB.



La fragmentación no hace que una plataforma de delitos financieros sea más segura. Hace que la detección sea más lenta, más difícil de auditar y la operación más costosa, un costo que el siguiente componente rastrea a través de los diagnósticos automotrices y la línea de montaje.



**LEER EL
BLOG**

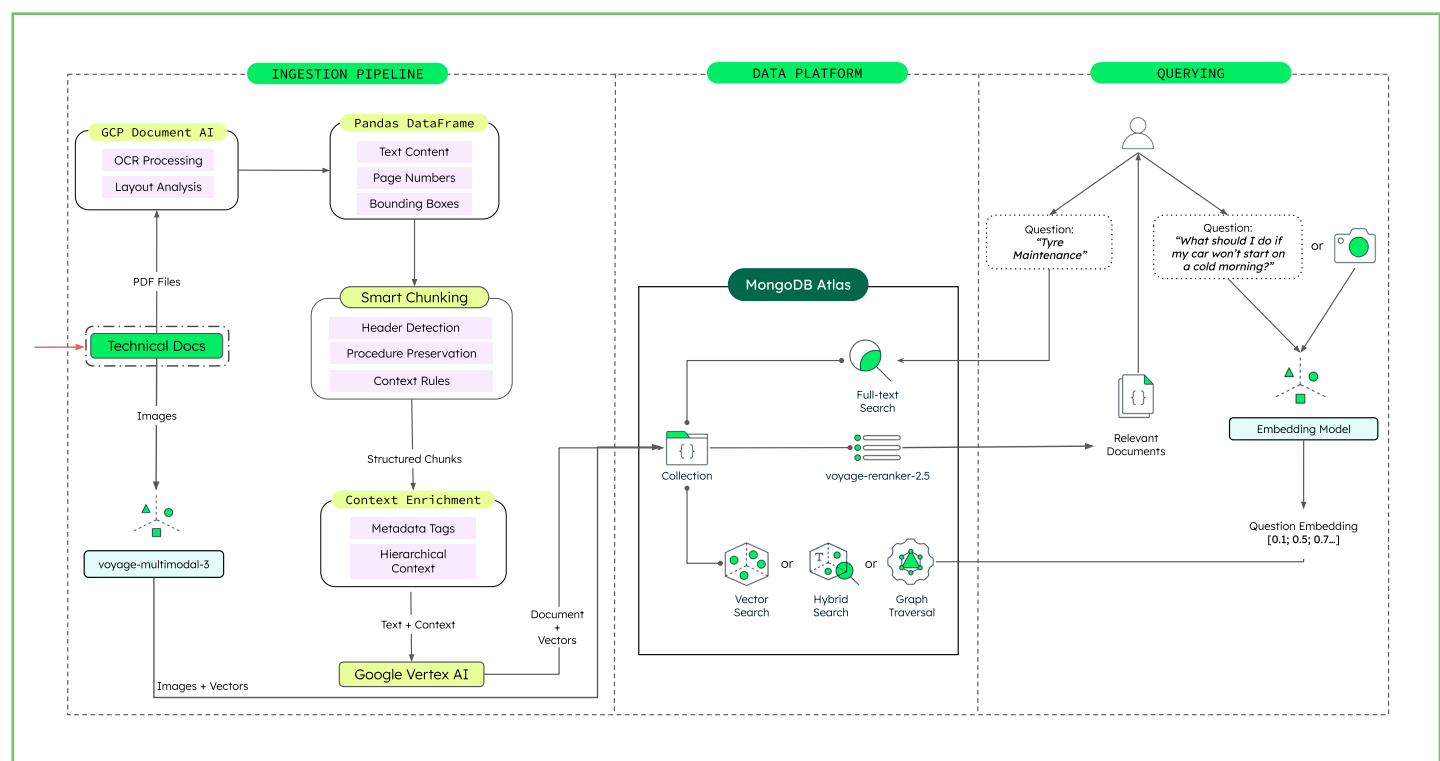
Aplicación inteligente de diagnóstico posventa automotriz

La conversación de diagnóstico que antes ocurría entre un asesor de servicio, un técnico y un compendio de manuales de reparación ahora se realiza cada vez más entre una tableta del área de servicio y un agente de IA que ha leído todos los manuales y boletines emitidos para esa plataforma. La arquitectura que hace que esto funcione es más interesante de lo que sugiere el frontend. Los riesgos son mayores. Los técnicos pueden pasar hasta el 30 % de su tiempo buscando información en lugar de reparar vehículos, y los eventos No Fault Found (NFF), en los que se reemplaza una pieza que funcionaba porque el diagnóstico no fue concluyente, representan casi el 30 % de los costos de garantía.

El patrón comienza con los manuales. La documentación de servicio y los boletines técnicos del fabricante del equipo original (OEM, el propio fabricante de automóviles) se procesan mediante reconocimiento óptico de caracteres (OCR), se fragmentan para preservar la integridad del procedimiento y se incorporan con texto

y modelos multimodales, para que un técnico pueda describir un síntoma con palabras o tomar una foto de un conector corroído. La capa de datos está unificada dentro de MongoDB Atlas: las incrustaciones de texto, las incrustaciones de imágenes multimodales y el grafo de relación entre síntomas, sistemas y causas raíz, todos se encuentran en la misma colección. El agente utiliza la búsqueda vectorial para encontrar la sección del manual relevante, recorre el árbol de fallos con \$graphLookup (el operador de MongoDB para consultar datos conectados recursivamente) para sugerir el siguiente paso lógico y aplica un reclasificador (reranker) de Voyage AI que coloca las advertencias de seguridad y las correcciones verificadas en la parte superior. Los mecánicos piensan en términos de sistemas, no de palabras clave: una batería descargada podría deberse a un cierre del maletero con consumo parasitario, y es precisamente un backend unificado basado en grafos y vectores lo que permite al agente sugerir que se compruebe el cierre cuando el técnico busca la batería.

Figura 9. Arquitectura unificada vectorial, grafo y multimodal para diagnósticos automotrices en MongoDB Atlas. El pipeline de ingestión procesa manuales mediante OCR e incrustaciones; las consultas pueden ser de texto o de imágenes.



**DESARROLLAR
LA APLICACIÓN**

Para los equipos automotrices, este es uno de los pocos lugares donde una capa de inteligencia unificada rinde frutos dentro de un solo ciclo de producto: menor tiempo de diagnóstico, menos escalamientos, reparaciones correctas a la primera más rápidas y reducciones significativas en los costos de garantía por NFF. El cambio arquitectónico es el requisito previo, no la consecuencia.

Desde una capa de inteligencia unificada hasta la precisión a escala

La sección 02 abarcó la arquitectura. El almacén de datos convergente es real, cuatro industrias demostraron que agrupa diferentes cargas de trabajo bajo un mismo techo, y el hilo conductor en fabricación, venta minorista, servicios financieros y automoción es el mismo argumento arquitectónico expresado con diferentes términos. Sin embargo, una arquitectura sin confianza no es más que un diagrama.

La sección 03 trata sobre la disciplina que se gana la confianza. Una capa unificada puede almacenar datos operativos, vectores, búsqueda vectorial y memoria de agentes en un solo lugar. Pero si las recuperaciones que le proporciona a los agentes están obsoletas, incompletas o imprecisas, el agente genera resultados erróneos y la elegancia de la arquitectura se convierte en un problema para el cliente. Una vez que se acepta que las alucinaciones son un problema de recuperación de la información y no un

problema de modelo, las preguntas pasan a ser de carácter práctico:

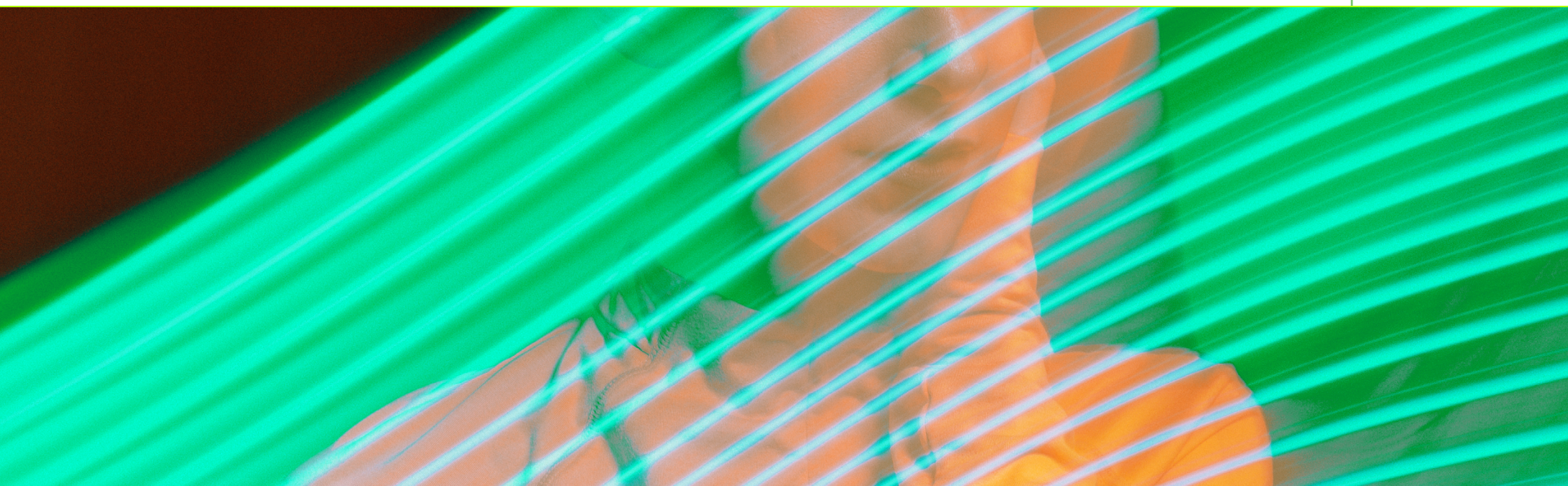
¿Qué se necesita para fundamentar la respuesta de cada agente en la verdad operativa en tiempo real?

¿Cómo mantiene las incrustaciones, la recuperación y el cambio de clasificación de forma rápida y barata sin perder la precisión?

¿Qué debe entregar la capa de datos antes de que un agente de producción pueda enviarse con seguridad?

En las próximas ocho páginas, respondemos a los tres: cómo Voyage 4 aumenta el límite de precisión, las decisiones de ingeniería que hacen que la recuperación sea un problema de elegir tres en vez de dos y el centro de llamadas de LG U+, que ya ejecuta todo esto en producción a 3 millones y medio de llamadas al mes.





SECCIÓN TRES

Precisión a escala

Fundamentar los agentes de IA en la verdad operativa en tiempo real.

En la sección 02 se expusieron los argumentos a favor de una capa de datos unificada. Esta sección trata sobre lo que esa capa debe entregar antes de que un agente desarrollado sobre ella pueda ser fiable en producción.

Las alucinaciones no son solo un problema del modelo. Son un problema de recuperación. Cuando un modelo de lenguaje inventa un número, una entidad o una política que no existe, casi siempre se debe a que la recuperación que alimentaba la entrada era inexacta, obsoleta o incompleta. Hay un nombre para la distancia entre un sistema que devuelve una respuesta y un sistema cuyas respuestas pueden ser utilizadas: la **brecha de confianza**. Esta brecha no es abstracta. Un agente que razona sobre el panorama de ayer de los datos de los clientes, un vector que está rezagado con respecto al registro que describe o una recuperación cuya fundamentación se ha desviado silenciosamente porque se cambió

el modelo de incrustación y los datos no se volvieron a procesar: ninguno de estos produce una respuesta visiblemente errónea. Produce una respuesta segura. Una alucinación en una sola demo es una curiosidad. La misma alucinación aplicada a miles de interacciones con los clientes por minuto es un incidente.

MongoDB ha dedicado los dos últimos eventos MongoDB.local a presentar los componentes básicos para cerrar esa brecha. En el MongoDB.local de San Francisco, celebrado en enero de 2026, el mensaje fue enviar más rápido: [la serie Voyage 4](#) elevó el límite máximo de precisión en la recuperación de datos, [Embedding and Reranking API](#) incorporó de forma nativa modelos de recuperación de alta fidelidad en la capa de datos, y [las incrustaciones de Automated Voyage AI](#) comenzaron a generarse en el momento en que se escriben los datos, de modo que los agentes siempre dispongan de un contexto actualizado. En el MongoDB.local de Londres, celebrado en mayo de 2026, el mensaje se centró en ejecutar agentes en producción a escala empresarial: MongoDB 8.3 se puso a disposición del público general con hasta un 45 % más de lecturas y un 35 % más de escrituras que la versión 8.0, y el [almacén de memoria a largo plazo LangGraph.js](#) también se puso a disposición del público general, lo que proporcionó a los equipos de JavaScript y TypeScript la misma memoria persistente de la que ya disponían los desarrolladores de Python.

CONTINÚA EN LA PÁGINA SIGUIENTE





Octubre de 2024

Disponible hoy

8.0 → 8.3

35%

Más rendimiento
de escritura

45%

Más rendimiento
de lectura

15%

Más rendimiento
de transacciones

Dos eventos, un mismo argumento. La precisión es una propiedad arquitectónica. Podríamos llamarlo integridad contextual: la disciplina de mantener de forma automática la consistencia de los datos operativos, las incrustaciones, los índices de búsqueda y el estado de los agentes, en vez de reconciliarlos a posteriori. Cuando esos cuatro elementos se encuentran en el mismo registro servido por la misma ruta de

query, no hay ningún paso de sincronización en el que pueda producirse una pérdida de precisión. La recuperación que recibe un agente se fundamenta en la verdad operativa del mismo instante en que fue emitida. Las compensaciones que quedan (cuantización, dimensionalidad, elección de incrustaciones) se ajustan sobre una base cuya precisión no se erosiona bajo ellas.



**RESUMEN DE .LOCAL
EN SAN FRANCISCO**



**RESUMEN DE
.LOCAL LONDRES**

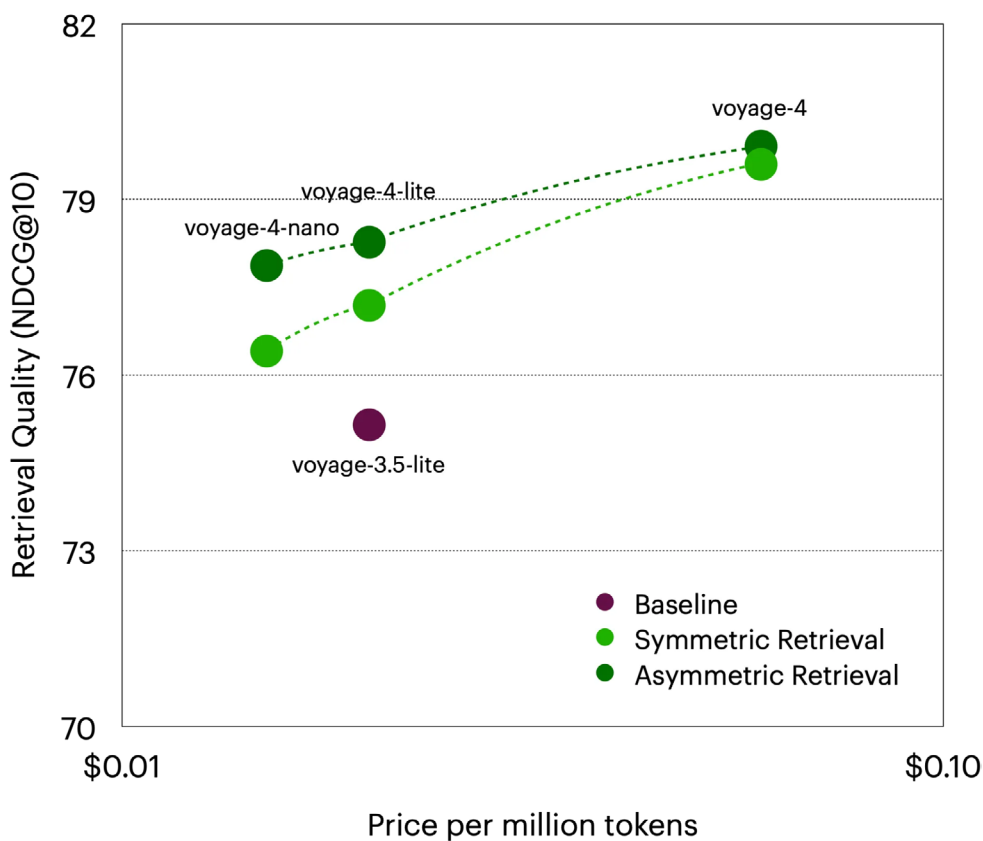
La serie Voyage 4 ya está disponible

Voyage 4 es una familia, no un modelo. La serie se comercializa en cuatro tamaños: *voyage-4-large*, *voyage-4*, *voyage-4-lite* y el *voyage-4-nano*, este último de pesos abiertos (*open-weights*). Lo más destacado es la precisión en la recuperación de información. Los modelos de incrustación de Voyage AI lideran el Retrieval Embedding Benchmark (RTEB), superando a modelos comparables de Gemini, Cohere y OpenAI en el tipo de tareas que se dan en el RAG empresarial. La clasificación del RTEB sigue siendo lo más parecido que tiene el sector a una medida objetiva de la capacidad de un agente para encontrar lo que necesita.

La innovación arquitectónica es el espacio de incrustación compartido, una primicia en la industria. Las generaciones anteriores requerían modelos idénticos en ambos lados de una query, lo que significaba que los equipos se limitaban a un único perfil de precisión y costo cuando compilaban el índice. Los modelos

Voyage 4 son interoperables: los cuatro producen incrustaciones compatibles. De ello se deriva el patrón de producción recomendado: la recuperación asimétrica. Integre su corpus de documentos una sola vez con *voyage-4-large* para obtener la máxima precisión por un costo único. Integre las query de forma continua con *voyage-4-lite* o *voyage-4-nano* para mantener la latencia y el costo por query bajos. La precisión de los modelos a gran escala reside en el índice; la rentabilidad de los modelos a pequeña escala se manifiesta en el momento de la entrega. La apertura de la sección nombraba “el modelo de incrustación se intercambiaba y los datos no se volvieron a procesar” como uno de los modos de fallo que produce la brecha de confianza. El espacio de incrustación compartido elimina la condición previa para que se produzca ese fallo: no es necesario volver a procesar nada, ya que el nuevo modelo utiliza el mismo lenguaje de coordenadas que el anterior.

Figura 10.
La recuperación asimétrica eleva la calidad al mismo costo.



El modelo *voyage-4-large* en sí utiliza una arquitectura de mezcla de expertos (MoE) que ofrece una precisión de vanguardia a un costo de servicio aproximadamente un 40 % menor que los modelos densos comparables. Es el primer modelo de incrustaciones para producción que utiliza MoE. Para cargas de trabajo de alto volumen

donde un solo agente puede emitir docenas de consultas por tarea, esa relación de costos se incrementa rápidamente. Pero el modelo es solo el límite. Alcanzarlo en producción es un problema de ingeniería.



[LEER EL ANUNCIO](#)



[GANAR LA INSIGNIA DE HABILIDAD](#)

Desarrollar un sistema de recuperación rápido, económico y preciso

El triángulo de rápido, barato y preciso se supone que es un problema de elegir dos. Es en los pipelines de recuperación donde realmente se convierte en un problema de elegir tres, porque las decisiones equivocadas desde el principio generan costos y latencia que ninguna actualización del modelo puede solucionar. Para ser honestos, la recuperación en producción debe resolverse de forma simultánea en tres ejes: precisión (el contexto correcto, fundamentado en la verdad operativa del momento en que se emitió la query), latencia (una respuesta correcta que llega después de que el usuario haya avanzado es, para fines operativos, una respuesta incorrecta) y costo (una pipeline que se ejecuta en cada query y se vuelve a ejecutar varias veces por interacción autónoma no puede escalar económicamente si cada llamada es costosa).

Voyage 4 eleva el nivel máximo de precisión. El trabajo de ingeniería para alcanzarlo en producción abarca tres optimizaciones, cada una dirigida a un cuello de botella diferente. **La reducción de dimensionalidad** utiliza el aprendizaje de representaciones Matryoshka para truncar las incrustaciones conservando al mismo tiempo el significado: 50 % menos almacenamiento y memoria, unas query 40 % más rápidas y 98 % de la precisión de referencia conservada. **La cuantización binaria** comprime vectores de alta precisión a un bit por dimensión: 96 % menos memoria, unas query 60 % más rápidas y 99 % de precisión conservada. **La recuperación asimétrica** combina un modelo más grande para los documentos con uno más pequeño para las query: un costo de incrustación por query un 50 % menor con voyage-4, y hasta un 83 % menor con voyage-4-lite. Cifras concretas, no impresiones.

Optimización	NDCG@10 medio	Latencia P50 (ms)	Latencia P95 (ms)
Ninguna (línea base)	0,65	1,772	2,11
Reducción de la dimensionalidad	0,643	1,049	1,265
Cuantización binaria	0,646	0,718	0,841
Recuperación asimétrica	0,64	1,722	2,006

Figura 11. Se evaluaron técnicas de optimización de recuperación en el conjunto de datos NFCorpus. Los tres enfoques conservan entre el 98 % y el 99 % de la precisión inicial, a la vez que ofrecen mejoras en la latencia de entre el 40 % y el 60 %.

El escalado de costos es lo que vale la pena trasladar a la revisión de arquitectura. Con 10 000 consultas al día, la recuperación asimétrica con voyage-4-lite ahorra aproximadamente 11 000 dólares al año en comparación con la línea de base. Con 1 millón de consultas al día, la misma configuración ahorra más de 1 millón de dólares al año. Y cuanto más grande sea ese número, más importante será la capa de datos, porque cada optimización debe componerse de manera limpia sin obligar al equipo a cambiar la plataforma subyacente.

La conclusión es que la precisión a escala es una pila de opciones, no una sola optimización. Si logra la incrustación del modelo correcta, limitará el margen de precisión. Si logra las dimensiones correctas, limitará la capacidad de almacenamiento y memoria. Si logra la cuantificación correcta, limitará la latencia. Si logra la estrategia de incrustación para query correcta, limitará el costo por query. Si logra la capa de datos correcta, estas decisiones se complementarán entre sí en vez de entrar en conflicto.



LEER EL
BLOG

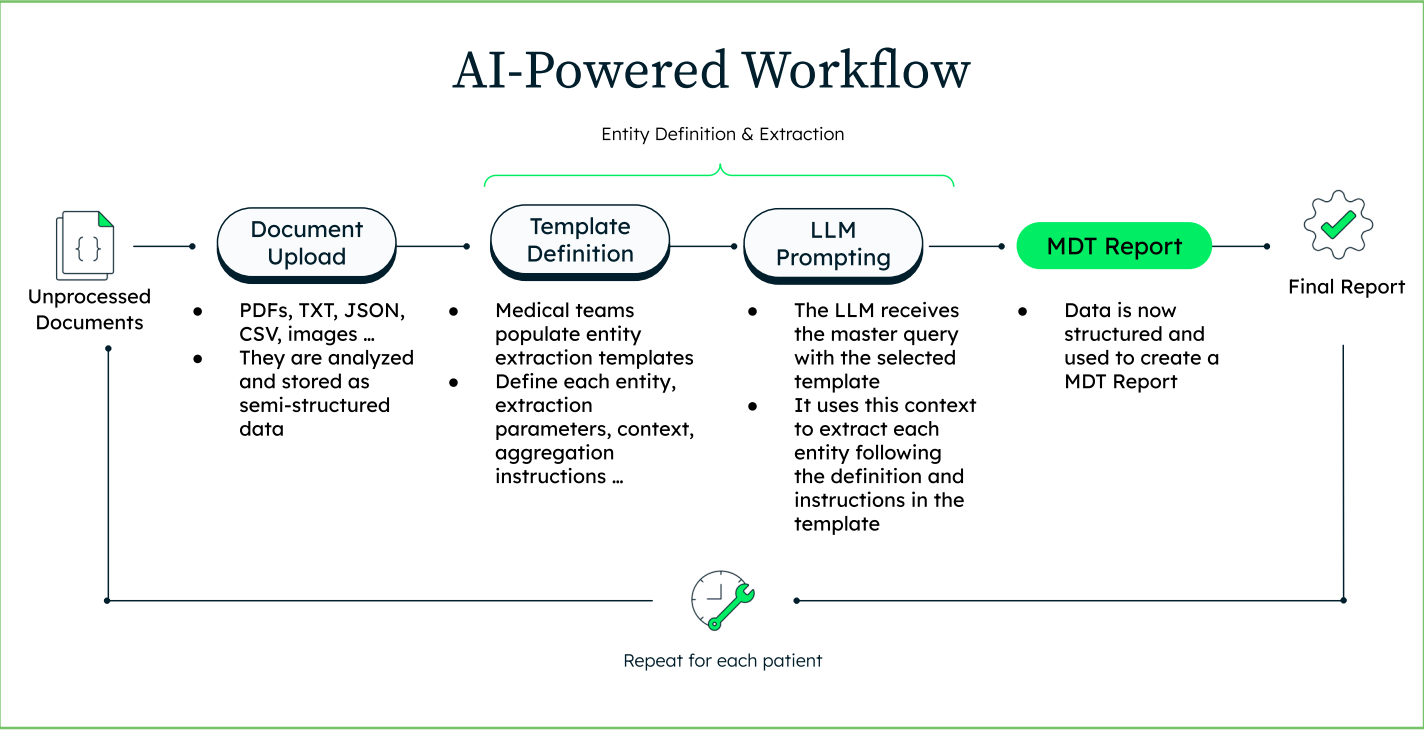
Generador de informes médicos impulsado por IA

La documentación clínica es la parte de la medicina que consume más tiempo y produce más agotamiento. En unidades de especialidad como oncología, los pacientes llegan con un largo recorrido de informes escaneados, archivos PDF, informes de patología, resúmenes de imágenes y resultados de laboratorio dispersos en sistemas desconectados. Antes de cada reunión del equipo multidisciplinario, los coordinadores y residentes dedican horas a reconstruir la historia: extrayendo hechos, resolviendo contradicciones, normalizando terminología y construyendo una línea de tiempo coherente. El problema de los datos detrás de la crisis de la documentación vuelve la promesa de la IA difícil de cumplir.

La arquitectura de referencia para un generador de informes médicos impulsado por IA fusiona datos estructurados de Historial Clínico Electrónico (EHR), notas clínicas no estructuradas, metadatos de imagen y la literatura médica relevante en un único pipeline. MongoDB actúa como la capa unificada de datos, que contiene los registros de los pacientes, las incrustaciones del contenido clínico y el registro de auditoría necesario para la regulación clínica.

La referencia expresa una opinión sobre la precisión. Cada paso de la generación se fundamenta en fuentes recuperadas. Cada fuente recuperada es identificable. Cada edición del profesional sanitario se retroalimenta al sistema como una señal. La idea no es sacar al profesional sanitario del circuito. La idea es que el profesional sanitario haga el trabajo que requiere un grado alto de criterio, y no la transcripción.

Figura 12.
Diagrama de flujo de trabajo de IA



**EXPLORAR
LA SOLUCIÓN**

Builder's Journey: crear aplicaciones de IA con MongoDB

Para los ingenieros cansados del contenido conceptual sobre IA que quieren ver cómo se construye algo de principio a fin, la charla “Builder’s Journey” de MongoDB.local en San Francisco es una oportunidad ineludible. Crea una aplicación de productividad para desarrolladores en dos evoluciones: V1 lanza un asistente de IA para la lluvia de ideas de proyectos con búsqueda semántica sobre trabajos anteriores; V2 añade ingestión de imágenes, búsqueda multimodal a través de texto e imágenes, y “memorias”: preferencias extraídas y detalles de implementación que personalizan proactivamente el siguiente proyecto. El cambio arquitectónico entre V1 y V2 es un único campo que cambia de 1 a 2. Las mismas tres colecciones. Las mismas query. El mismo clúster.

Esta charla es útil para los líderes técnicos, incluso para aquellos que no escriben código día a día, por un argumento implícito de la arquitectura y otro explícito. Desde el punto de vista implícito: cada paso reutiliza la misma capa de datos. El mismo clúster de Atlas impulsa los datos

operativos, la búsqueda vectorial, la memoria del agente y la pipeline de recuperación. No hay ningún interludio del tipo “y ahora añadimos una base de datos vectorial”. Desde el punto de vista explícito: la charla se detiene sobre la búsqueda vectorial filtrada y explica por qué MongoDB filtra primero por ID de usuario y luego ejecuta la búsqueda vectorial, mientras que Postgres ejecuta la búsqueda vectorial primero y filtra después, lo que significa que los resultados relevantes pueden salir del conjunto de candidatos antes de que sean evaluados. Una decisión arquitectónica pequeña. Una consecuencia importante a gran escala.

También es una herramienta útil para la incorporación. Recomendar la charla a un miembro nuevo del equipo lo volverá productivo más rápido que cualquier diagrama de arquitectura, porque verá cómo se toman las decisiones en contexto y los sacrificios que se evitan. En la charla se explica el patrón. La siguiente historia muestra que soporta una carga real en producción.



[VER
LA CHARLA](#)



LG U+ escala el servicio al cliente con IA mediante MongoDB Atlas

Los argumentos de la arquitectura alcanzan un techo diferente cuando la carga de trabajo es un centro de llamadas que recibe 3 millones y medio de llamadas al mes.

LG U+, la empresa de telecomunicaciones coreana que dirige uno de los centros de atención al cliente más grandes del país, se propuso desarrollar Agent Assist: una herramienta de IA que brinda a 4000 agentes humanos transcripción en tiempo real, categorización automática de consultas de clientes y resúmenes contextuales inteligentes durante llamadas en vivo. La base de datos relacional subyacente no podía gestionar, en una misma ruta, los datos vectoriales generados por las funciones de IA, las pruebas que el equipo debía ejecutar sobre dichas funciones durante las consultas y las consultas operativas que requería el servicio.

LG U+ construyó Agent Assist desde cero en MongoDB Atlas y adoptó Atlas Vector Search al principio del desarrollo. Al migrar los datos

operativos de Postgres al mismo almacén que las incrustaciones, el equipo eliminó por completo la capa de sincronización entre los sistemas vectorial y operativo. El modelo orientado a documentos se adaptó a las variadas formas de los datos de consulta, y \$vectorSearch ejecutó la similitud semántica directamente en las mismas colecciones que los agentes ya consultaban para obtener respuestas transaccionales.

El servicio se lanzó cuatro meses después de que comenzara el proyecto. El tiempo de procesamiento por llamada se redujo un 7 %. La eficiencia de los recursos mejoró un 30 %. Se mantuvo una latencia inferior a un segundo bajo carga en las consultas de \$vectorSearch a gran escala, y el escalado automático absorbió el pico matutino cuando se produjo un aumento repentino del volumen de solicitudes. El servicio, que se desplegó para 1200 agentes en un principio, ahora gestiona más de un millón de consultas de usuarios por semana y está escalando hacia 2300 agentes.

“Gestionar los datos operativos y vectoriales en MongoDB le abrió un nuevo mundo a nuestro equipo.”

Minkyu Ha, *director sénior de ingeniería de software, LG U+*

La decisión que cierra la historia tiene más peso que los porcentajes. Más allá de Agent Assist, LG U+ está extendiendo la misma arquitectura a sus callbots y chatbots, expandiendo el servicio al mercado B2B y trabajando con MongoDB en una modernización más amplia de la capa de datos que incluye la migración de aplicaciones locales sustanciales a Atlas. La decisión arquitectónica adoptada para una aplicación concreta se ha convertido en la base de la cartera de productos de IA de la empresa.

Así es como se manifiesta la integridad contextual a escala de una empresa nacional de telecomunicaciones. Las incrustaciones vectoriales y los datos operativos se encuentran en el mismo registro. La recuperación que una IA presenta a un agente humano en una llamada de un cliente se basa en la verdad operativa del mismo instante en que se está produciendo la llamada. La brecha de confianza con la que se abre esta sección no es un proceso que se pueda gestionar. Es un modo de fallo que la arquitectura ha sido diseñada para eliminar.



[LEER LA HISTORIA](#)

CONTINÚA EN LA PÁGINA SIGUIENTE





Lo que muestra LG U+ es la lista de requisitos mínimos que debe cumplir la capa de datos antes de que un agente de producción pueda lanzarse al mercado con total seguridad.

- El índice de búsqueda de la IA y los datos en tiempo real del negocio tienen que estar en el mismo lugar, por lo que el agente no responde desde una versión de la verdad mientras actúa en otra.
- Las consultas semánticas deben devolver resultados en menos de un segundo, incluso bajo carga real de producción. Si el proceso es más lento, el agente llega demasiado tarde para ser útil.
- La plataforma debe escalar de forma automática cuando el volumen de llamadas aumenta de repente, sin la necesidad de que el equipo reconstruya algo para absorber el pico.
- Y el modelo de datos debe ser lo suficientemente flexible para adaptarse a cualquier forma que tome la conversación, sin necesidad de una migración de esquema cada vez que el negocio aprende algo nuevo. Nada de esto es un problema de calidad del modelo.

Todo es arquitectura, que se decide antes de que se escriba el primer prompt.

De la precisión a escala a los sistemas de acción

La sección 03 trató sobre la fundamentación. El Voyage 4 amplió el margen de precisión, y las técnicas de ingeniería que lo respaldaron (reducción de dimensionalidad, cuantización y recuperación asimétrica) hicieron que ese margen fuera alcanzable en producción. Luego, LG U+ mostró cómo se ve la recuperación fundamentada a escala de producción, con 3 millones y medio de llamadas al mes ejecutándose en una capa de datos unificada. Un agente fundamentado que recupera de la verdad operativa, en el mismo instante en que se formula la pregunta, es algo más que un chatbot sofisticado. Es un agente que está listo para actuar.

Una vez resuelta la precisión a escala, la pregunta cambia de nuevo. ¿Qué puede desarrollar una empresa que no pudiera desarrollar antes? La respuesta se encuentra en la sección 04.



SECCIÓN CUATRO

Construyendo la era de la IA

De los sistemas de registro a los sistemas de acción.

Hay una diferencia entre un agente que resume un reclamo y otro que lo paga. Entre un agente que recomienda una ruta de pago y otro que ejecuta la transferencia. Entre un agente que redacta una decisión de préstamo y otro que financia el préstamo. En el momento en que un agente realiza una transacción, los requisitos se multiplican. La memoria tiene que ser duradera a lo largo de las sesiones. El estado tiene que sobrevivir a los fallos. Los permisos tienen que ser auditables. La precisión era la parte fácil. La acción es la más difícil.

La industria tiene un nombre para el cambio: **sistemas de registro o sistemas de acción**. Un sistema de registro almacena, recupera y muestra información. Un sistema de acción cambia el estado del mundo en el que opera. Realiza transacciones, activa flujos de trabajo, reserva citas, actualiza libros contables y realiza pagos. La sección 03 detalló la primera mitad. Esta sección trata sobre lo que cambia cuando la precisión ya no es lo único que puede fallar.

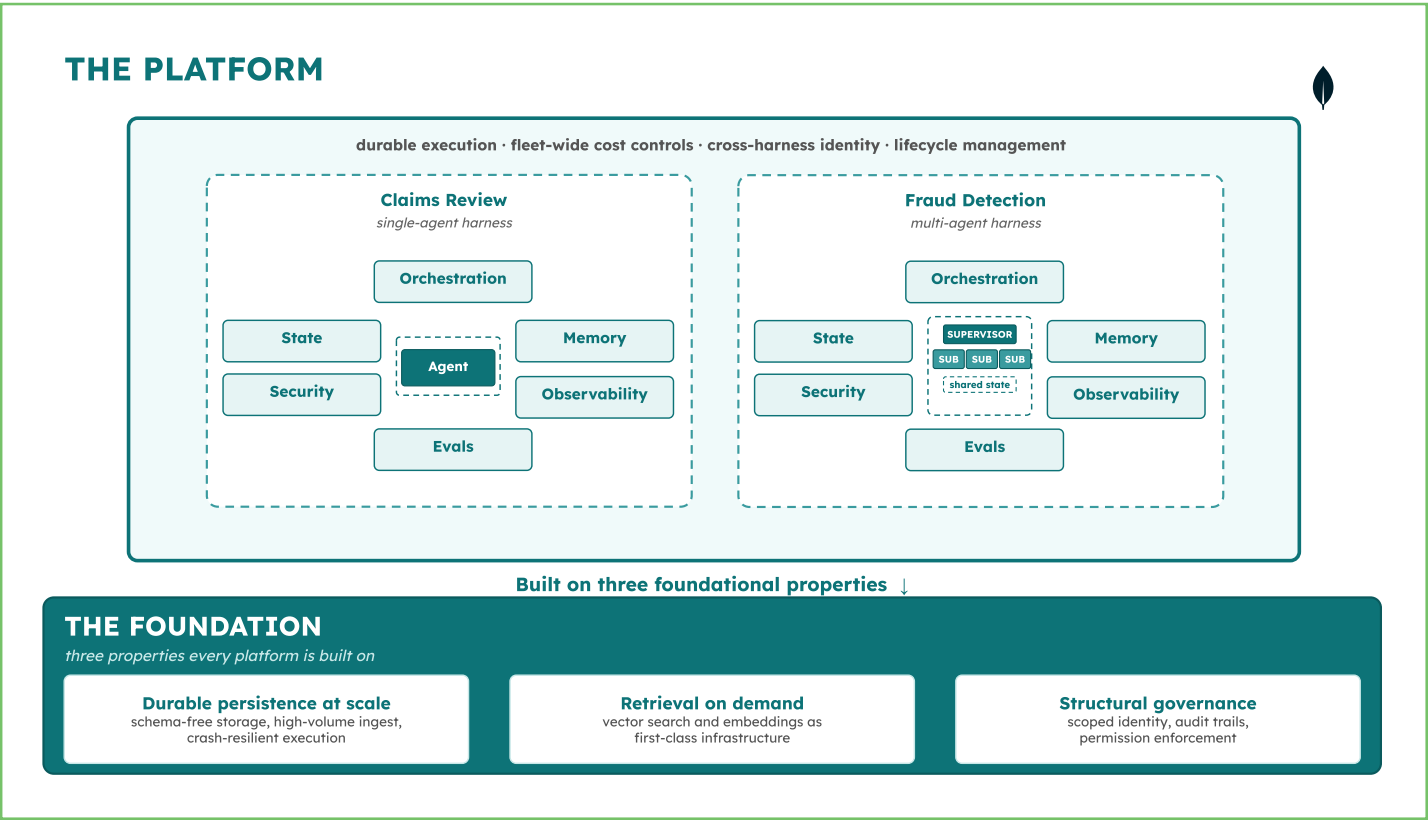
Los riesgos pueden medirse. [Gartner](#) predice que el 40 % de las aplicaciones empresariales incorporarán agentes de IA para finales de 2026. Y el 40 % de los proyectos de IA con agentes se cancelarán para finales de 2027. La simetría no es accidental. Las cancelaciones no son un problema de calidad del modelo. Los proyectos de agentes empresariales se estancan en tres puntos predecibles: en la revisión de seguridad, donde no existe un conjunto de controles estándar de la industria para sistemas autónomos; en los límites de costo, donde el gasto en tokens supera el valor comercial; y en el límite de integración, donde la capa de datos no puede mantenerse al día con lo que el agente necesita saber.

Los sistemas de agentes de producción tienen más en común con los sistemas de producción de aprendizaje automático de lo que la mayoría de los equipos cree. El modelo recibe la atención. La infraestructura que rodea al modelo decide si algo realmente funciona: la memoria, el estado, la orquestación, la observabilidad, las evaluaciones, la seguridad y la capa de datos que alimenta todo ello. La serie [Diseño de una plataforma de agentes](#) de MongoDB desarrolla esta arquitectura en profundidad. Así es su estructura:

CONTINÚA EN LA PÁGINA SIGUIENTE



Figura 13.
Una plataforma
permite la ejecución
de muchos arneses y
agentes en una amplia
variedad de casos
de uso, dominios y
marcos de trabajo.



Cada casilla de la figura es un harness (arnés de ejecución): el entorno que impulsa a un agente o a un conjunto coordinado de agentes a través de su ciclo de decisión. Las seis disciplinas de ingeniería que contiene (orquestración, estado, memoria, seguridad, observabilidad, evals) son donde se produce la mayor parte del trabajo de producción. La plataforma se sitúa por encima de ellos, ejecutando muchos arneses en muchos equipos a lo largo del tiempo.

Tres propiedades fundamentales sostienen la plataforma. Persistencia duradera a gran escala, para los datos sin esquema que generan los arneses. Recuperación bajo demanda, mediante incrustaciones y búsqueda vectorial, por lo que el contexto está disponible en el instante en que un agente lo necesita. Y gobernanza estructural, con identidad delimitada, registros de auditoría y aplicación de permisos que se proporcionan una vez en la plataforma, en vez de acoplarse a cada arnés. Una plataforma que carece de cualquiera de los tres produce arneses que parecen funcionar en condiciones controladas y fallan bajo carga de producción.



**LEER LA
SERIE DE
PLATAFORMAS**



Los agentes deben ser invisibles: los objetos de negocio son más importantes que la memoria de la IA

Existe un error arquitectónico sutil en el fondo de la mayoría de las primeras plataformas de agentes. Los equipos tratan la memoria del agente como la abstracción central, con todo lo demás organizado a su alrededor: el cliente, la póliza, el reclamo y la transacción. Funciona en la versión de demostración porque esta solo necesita un agente y un flujo de trabajo. Se rompe en producción porque las empresas reales tienen muchos agentes, muchos flujos de trabajo y una verdad subyacente: el objeto de negocio.

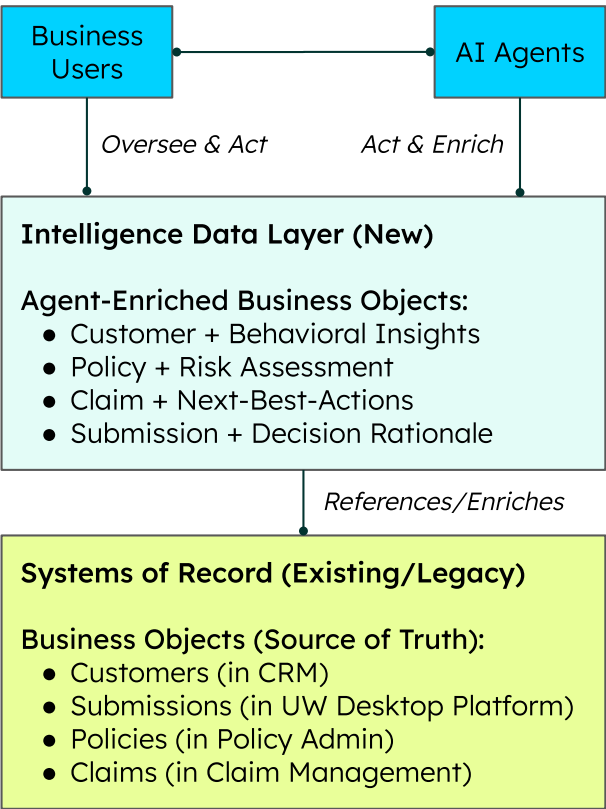
Los agentes deben ser invisibles. El cliente no interactúa con un agente. En la industria de seguros, el cliente interactúa con su póliza, su reclamo y su cuenta. En la venta minorista, con su pedido. En el ámbito sanitario, con su caso. El sistema utiliza agentes para actuar sobre esos objetos. Si organiza la arquitectura según los objetos de negocio, en vez de la memoria de los agentes, surgen tres consecuencias. Los nuevos agentes se integran en los objetos existentes sin necesidad de rediseñar la capa de datos. Varios agentes pueden coordinarse según el mismo objeto sin interferir entre sí. El registro de auditoría

se acumula naturalmente a nivel de objetos, donde los reguladores y los operadores lo desean.

La capa de datos que esto requiere tiene un nombre: la **capa de datos de inteligencia**, donde los objetos de negocio, las incrustaciones, las memorias de los agentes, el estado del flujo de trabajo y el rastro de auditoría se encuentran en un solo lugar coherente. El modelo orientado a documentos de MongoDB es una opción natural. Los objetos de negocio están inherentemente anidados. Los controles de acceso se pueden aplicar a nivel de campo. Y la misma interfaz de consulta responde a preguntas operativas, semánticas e históricas. Este es el patrón de Converged Datastore aplicado al caso donde se amortiza más rápido.

La conclusión práctica es diseñar primero el modelo de objetos. Los agentes van y vienen. Los objetos permanecen. Si el modelo de objeto Realm es correcto, todos los agentes subsiguientes se forman sobre la misma base. Si es incorrecto, cada agente debe reconstruir su propia arquitectura.

Figura 14.
El sistema de acción: permitir una supervisión y actuación más inteligente por parte de los usuarios empresariales



[LEER LA PARTE 1](#)



[LEER LA PARTE 2](#)

Desarrolle agentes de IA que valgan la pena conservar: el marco de trabajo de Canvas

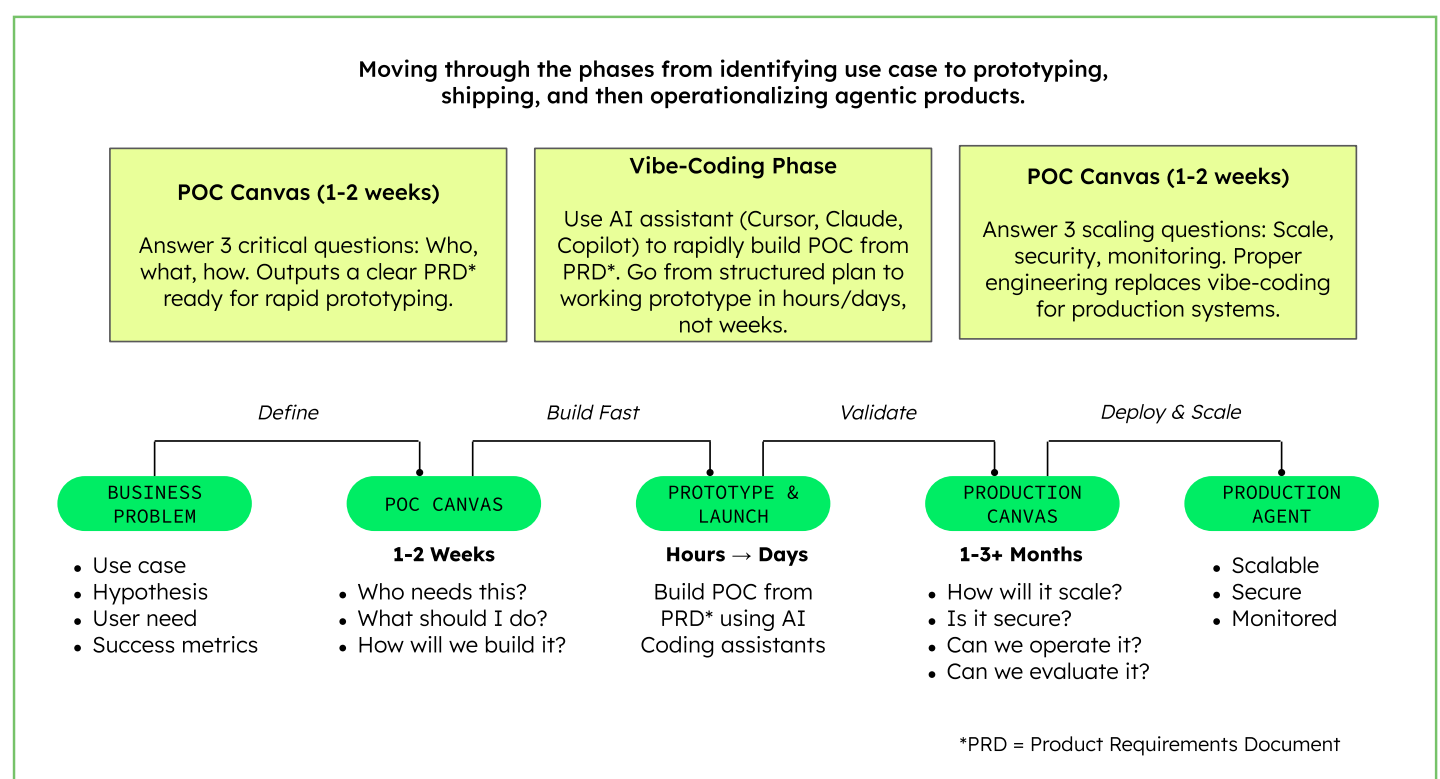
La mayoría de los proyectos de agentes fallan de la misma manera. El equipo comienza con una elección de marco, elige un modelo, desarrolla algo interesante y luego se da cuenta demasiado tarde de que nadie necesitaba lo que desarrollaron. [La investigación de McKinsey](#) es coherente en esto. Casi el 80 % de las empresas informan que utilizan IA generativa, pero menos del 10 % de los casos de uso superan la fase piloto. El patrón es tan común que los investigadores del MIT tienen un nombre para ello: la división de IA generativa.

El marco de trabajo de lienzo es una respuesta directa a este patrón. Mikiko Bazeley, Staff Developer Advocate en MongoDB, lo diseñó según el lienzo del modelo de negocio. La estructura obliga a los equipos a responder a las preguntas correctas en el orden correcto: primero el producto, luego el agente, luego los datos y luego el modelo. La premisa es que el diseño de los agentes se sitúa entre el producto y los datos, y

determina qué conocimientos necesita el agente y qué modelos pueden ofrecerlos. Si invierte el orden, como hacen la mayoría de los equipos, acabará con la tecnología en busca de un problema.

El lienzo de prueba de concepto (POC) consta de ocho cuadrados enfocados destinados a validar rápidamente, organizados en cuatro fases: producto, agente, datos e integración de modelos. El lienzo de producción agrega 11 cuadrados en cinco fases: caso de negocio y escalado, arquitectura robusta de agentes, infraestructura de datos unificada, operaciones de modelo y el refuerzo que omiten los POC y exigen las empresas. Revisión de seguridad, límites máximos de costos, presupuestos de latencia, observabilidad, gobernanza, pipelines de evaluación y estrategia de ajuste fino. El marco obliga a los equipos a planificarlo todo antes de escalar, no después.

Figura 15.
El marco de trabajo de lienzo de IA de agentes. Un enfoque estructurado de cinco fases que va desde la definición del problema de negocio hasta la prueba de concepto, prototipo, marco de producción e implementación de agentes en producción.



**LEER EL
BLOG**

Lo que el marco de trabajo acierta, y lo que la mayoría de las guías de los agentes pasa por alto, es que la elección de la capa de datos es fundamental. Los equipos que intentan posponerlo acaban teniendo que reconstruir la arquitectura a mitad de camino. Los equipos que eligen una plataforma unificada desde el primer día se dedican a la ingeniería de

prompts y la orquestación en vez de gestionar la infraestructura. El lienzo trata esto como una decisión estructural, no como una preferencia de herramientas.

Las siguientes piezas muestran sistemas de acción en producción, en servicios financieros, pagos, cadena de suministro y préstamos.

Desbloqueando la inteligencia documental en los servicios financieros con IA de agentes

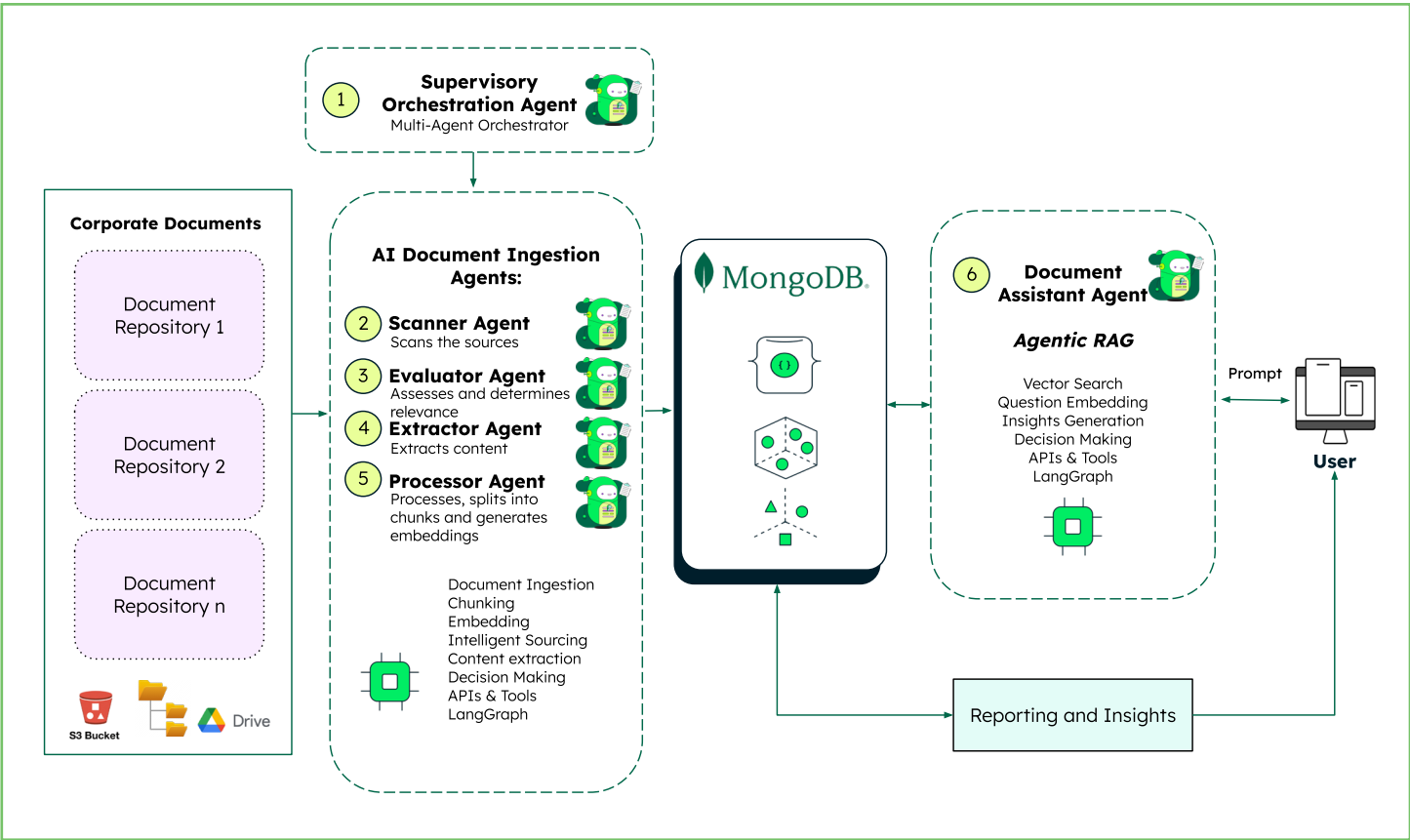
Los servicios financieros funcionan con documentos. Archivos de préstamos, paquetes de KYC, cartas de mandato, confirmaciones de derivados y atestaciones de cumplimiento. El trabajo de extraer la estructura de esos documentos ha sido históricamente un esfuerzo manual poco eficiente. La IA de agentes cambia la economía, pero solo si la capa de datos puede soportar la ingestión, extracción y reconciliación de múltiples agentes a escala.

La arquitectura de referencia para la inteligencia de documentos utiliza MongoDB como la base de datos unificada para un sistema supervisor de varios agentes. Un agente supervisor supervisa a cuatro agentes secundarios especializados que trabajan en colaboración: un escáner identifica los documentos relevantes, un evaluador los califica según su relevancia para el caso de uso, un extractor extrae el contenido y lo prepara para su segmentación, y un procesador genera incrustaciones para las tareas posteriores.

Un agente asistente de documentos de segundo nivel gestiona el RAG autónomo, decidiendo cuándo y cómo recuperar en función de la consulta del usuario. El modelo voyage-context-3 de Voyage AI produce incrustaciones de fragmentos contextualizados que capturan el contexto del documento local y global, ofreciendo una precisión de recuperación significativamente mayor a menor costo.

La memoria del agente es lo que hace la mayor parte del trabajo en silencio. Cada agente recuerda interacciones anteriores, aprende de las correcciones y compila contexto a lo largo del tiempo. Esa memoria tiene que residir en algún lugar duradero, con controles de acceso y rastros de auditoría que cumplan con la regulación de los servicios financieros. El rol de MongoDB es ser ese lugar: el documento, las incrustaciones, la memoria del agente y el registro de auditoría en una capa uniforme.

Figura 16.
Arquitectura inteligente de procesamiento de documentos con MongoDB



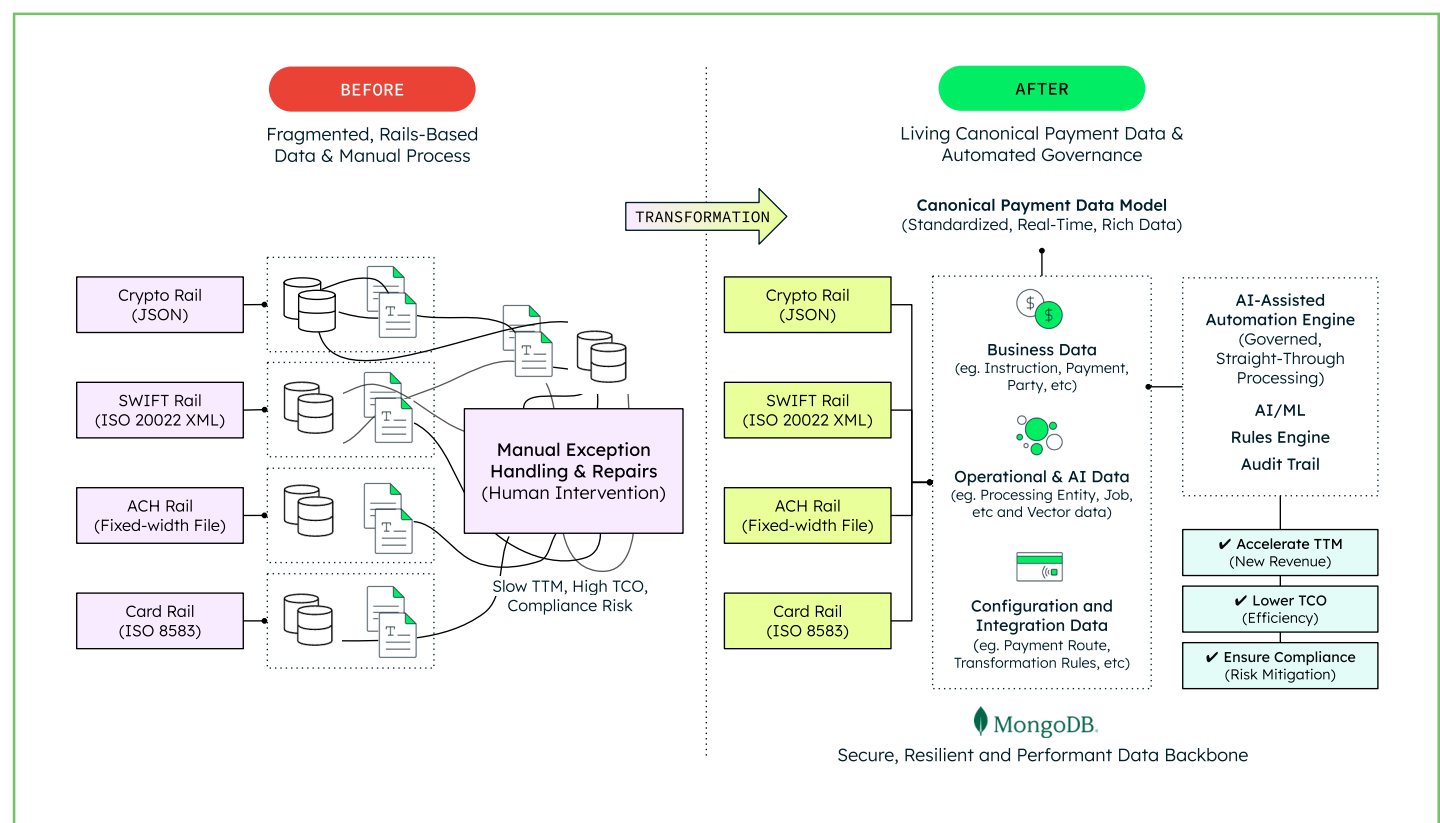
**LEA EL
ARTÍCULO**

Aprovechar el poder de los agentes para modernizar los sistemas de pago mundiales

Los sistemas de pago son la parte de la pila financiera donde la modernización es más difícil y el costo de equivocarse es más alto. Cada una de las rieles de pago que mueven dinero se construyó para una era diferente: SWIFT habla ISO 20022 XML, ACH habla archivos de ancho fijo, redes de tarjetas hablan ISO 8583 y las criptomonedas hablan JSON. La mayoría de los bancos los conectan con integraciones punto a punto y parchean las brechas con el manejo de excepciones humanas. Solo los pagos fallidos agotan la economía mundial en más de 100 000 millones de dólares cada año. El resultado es un tiempo lento de salida al mercado, un costo de propiedad alto y un riesgo persistente de cumplimiento normativo.

La solución no es cambiar las rieles. Es agrupar todo lo que emiten en un modelo de pago canónico: datos comerciales como instrucciones y partes, datos operativos y de IA como tareas de procesamiento e incrustaciones vectoriales, y datos de configuración como rutas y reglas de transformación. Además de ese modelo, un motor de automatización con IA/aprendizaje automático, reglas y un registro de auditoría de funcionalidad incorporada puede finalmente realizar trabajos útiles: enrutamiento, conciliación, clasificación de fraudes, resolución de excepciones y comunicación con el cliente.

Figura 17.
De la expansión desordenada de la vía tradicional a una plataforma de pagos canónica y autónoma.



El resultado es un tiempo de lanzamiento al mercado más rápido, un costo menor de propiedad y el trato del cumplimiento como una propiedad de los datos, en vez de una intervención de emergencia trimestral. Lo que cambia es lo que el banco puede enviar.



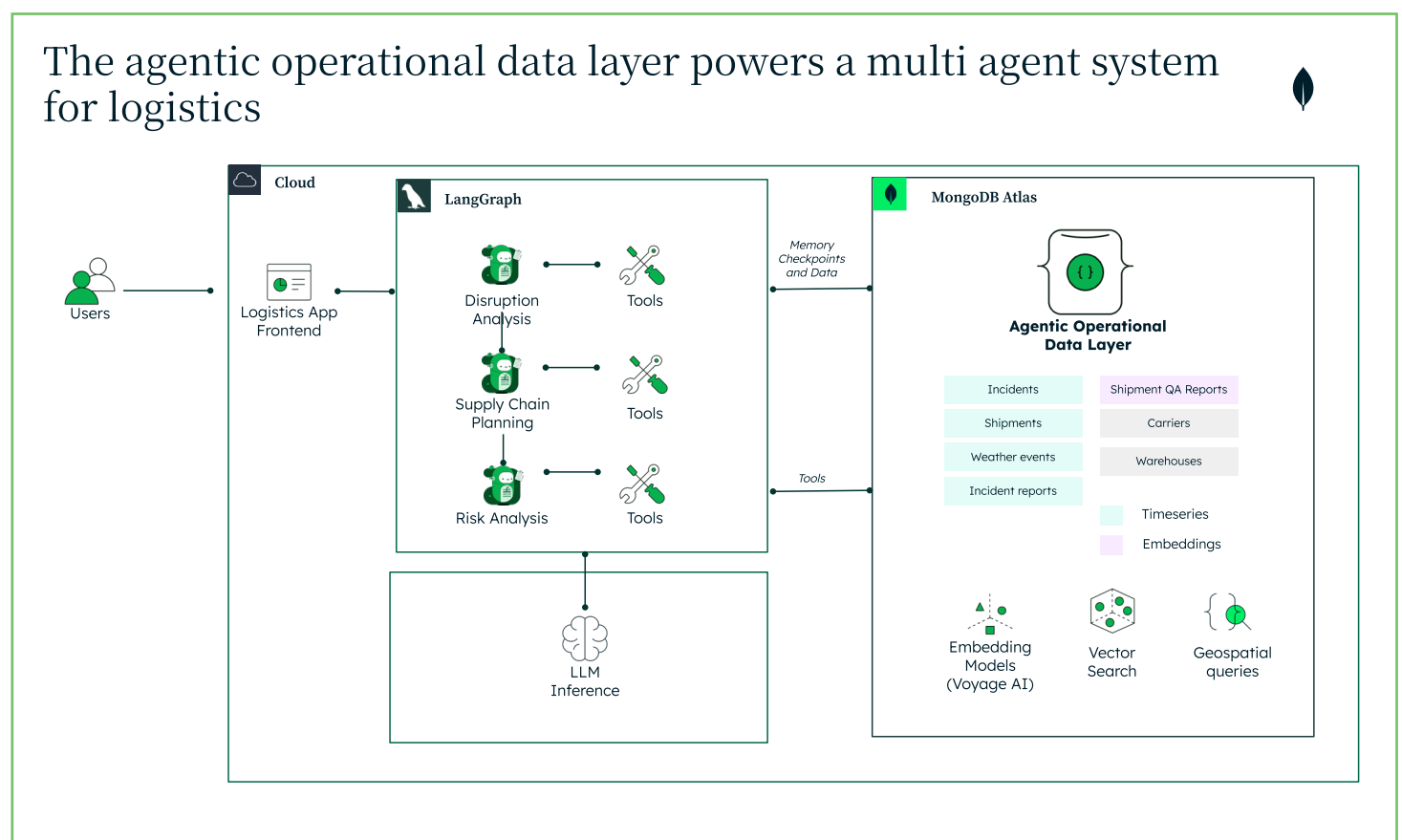
**LEER
EL BLOG**

Sistemas de múltiples agentes para la gestión de interrupciones en la cadena de suministro

La interrupción de la cadena de suministro es el problema canónico de varios agentes. Un evento meteorológico en un puerto, una acción laboral en otro, un problema de calidad en un proveedor de segundo nivel y un repunte de la demanda en un mercado clave interactúan entre sí. Ningún equipo humano puede tener el panorama completo en tiempo real. Ningún agente individual tampoco puede tenerlo. El costo de cometer errores es concreto: las interrupciones en la cadena de suministro cuestan a las empresas unos 184 000 millones de dólares anuales.

En la práctica, el trabajo se divide en roles especializados. El análisis de interrupciones vigila los incidentes a medida que se desarrollan. La planificación de la cadena de suministro analiza el impacto en las fases posteriores y propone medidas de mitigación. El análisis de riesgos evalúa el nivel de exposición y pone de manifiesto los aspectos que deben remitirse a un nivel superior. LangGraph (un marco de orquestación para flujos de trabajo de varios agentes) enruta entre ellos, pero la comprensión compartida de los agentes del mundo vive en la capa de datos subyacente: un único almacén donde se encuentran los envíos, el clima, los transportistas, los centros de flotas y los informes de incidentes integrados juntos.

Figura 18. Diagrama de arquitectura de la solución. Tres agentes especializados (análisis de interrupción, planificación de la cadena de suministro y análisis de riesgos) se orquestan a través de LangGraph, con MongoDB Atlas sirviendo como capa de datos operativos para agentes para envíos, eventos meteorológicos, transportistas, almacenes, incrustaciones vectoriales y consultas geoespaciales.



**EXPLORAR LA
SOLUCIÓN**

MongoDB Atlas aporta más que almacenamiento. Los eventos meteorológicos fluyen como series de tiempo. Los envíos y almacenes llevan coordenadas sobre las que las consultas geoespaciales pueden procesar. Los informes históricos de incidentes se incrustan con Voyage AI para que la búsqueda vectorial pueda extraer

lo que los agentes necesitan tras interrupciones previas. Y los puntos de control de memoria que LangGraph guarda entre los pasos del agente viven en el mismo almacenamiento, lo que hace que el rastro postmortem se acumule como un registro estructurado en vez de un hilo de Slack.

El proceso moderno de concesión de préstamos digitales de principio a fin, impulsado por MongoDB y la IA de agentes

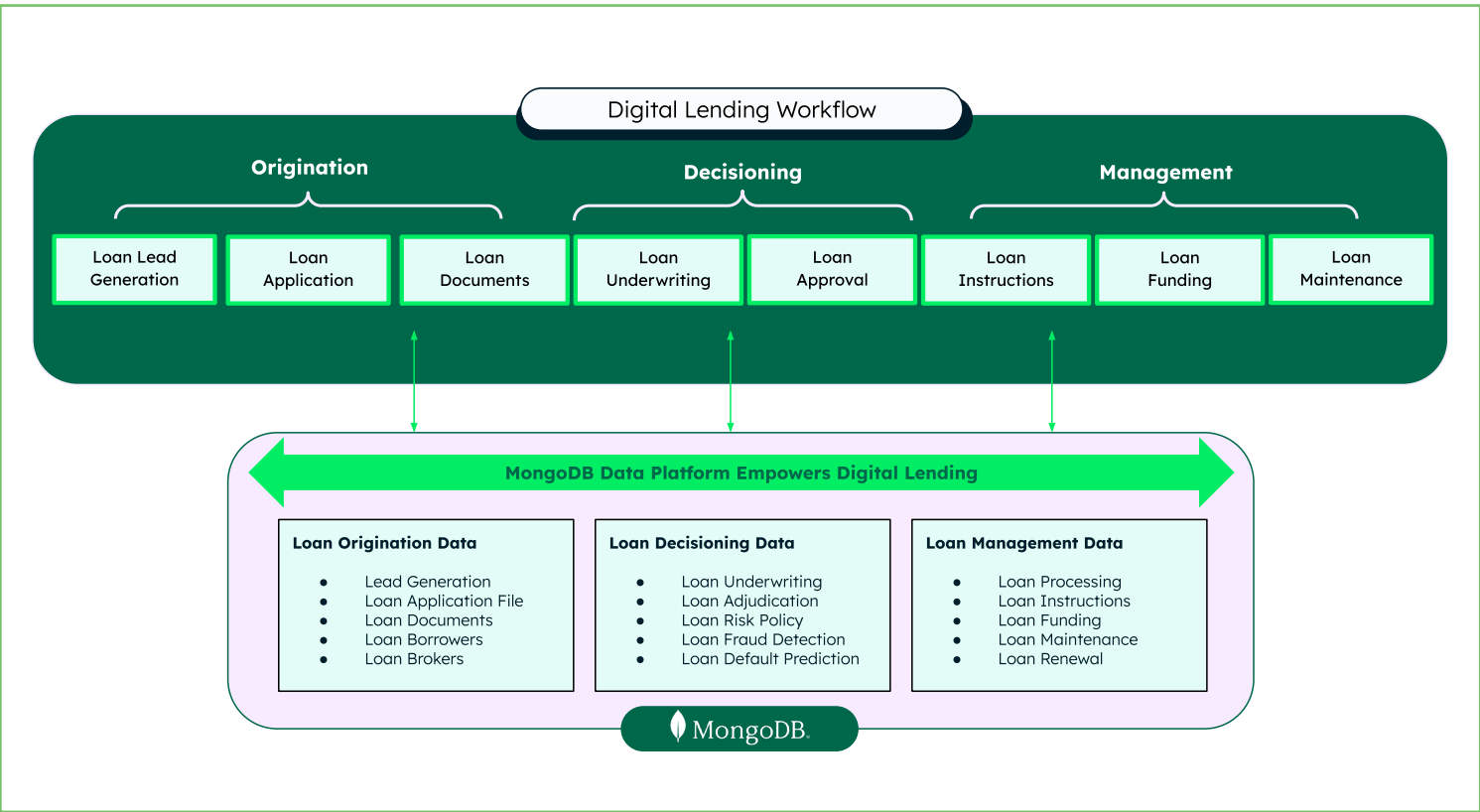
Los préstamos son el flujo de trabajo financiero por excelencia al que se le ha prometido una automatización integral durante dos décadas, pero que en su mayoría solo ha ofrecido una automatización parcial en numerosos departamentos aislados. Las razones son bien conocidas. Los datos se encuentran repartidos en una docena de sistemas, los pasos para el cumplimiento normativo requieren la intervención humana en múltiples fases y la experiencia del cliente viene determinada por aquel sistema que, en cada caso, presente el peor traspaso de datos.

La IA de agentes cambia el cuello de botella. Un recorrido de préstamo moderno puede diseñarse como un conjunto coordinado de agentes que manejan el ciclo de vida completo desde el origen hasta la renovación. **Un agente de solicitud de préstamo** actúa como la puerta de entrada inteligente, guiando a los prestatarios, validando y enriqueciendo datos en tiempo real, y extrayendo información de los documentos para producir un perfil de préstamo listo para tomar decisiones. Un **agente de suscripción** combina reglas de negocio, modelos de aprendizaje

automático y datos alternativos para evaluar la solvencia de forma continua, enrutando excepciones y casos de alto riesgo a un revisor humano para la aprobación final. Un **agente de renovación** supervisa el ciclo de vida posterior a la financiación, prediciendo refinanciamiento, dificultades y riesgo de abandono, y activando ofertas personalizadas o intervención temprana. Cada agente está delimitado, cada uno es observable y cada uno comparte una vista única de la aplicación como un objeto de negocio. El cliente ve una experiencia uniforme. El prestamista ve un rastro de auditoría.

La arquitectura importa porque el préstamo es un trabajo que requiere un alto grado de juicio. La capa de datos determina dónde la automatización es segura, dónde se requiere la revisión humana y, en última instancia, dónde está la diferencia entre un recorrido que toma minutos y uno que toma semanas. El patrón se generaliza más allá del préstamo a cualquier flujo de trabajo financiero con gran volumen de documentos que tenga que equilibrar la velocidad con el control regulatorio.

Figura 19.
Flujo de trabajo de la IA de agentes: préstamos digitales



**LEER EL
ARTÍCULO**

La prueba de portabilidad

Los casos de uso de esta sección han mostrado cómo son los sistemas de acción en el mundo real: pagos que se concilian, cadenas de suministro que responden, procesos de préstamo que avanzan desde la entrada hasta la decisión, y agentes que hacen más que recuperar información. Pero una vez que esos sistemas comienzan a tener importancia, surge otra pregunta. No solo qué hará el agente, sino dónde la empresa necesitará ejecutarlo.

Los sistemas de agentes empresariales rara vez se limitan a un solo entorno definido. Pueden abarcar nubes, regiones, plataformas modelo, herramientas de desarrollo, datos operativos y límites de seguridad. Algunas cargas de trabajo

se trasladan para reducir la latencia. Otras para la residencia de datos. Otras porque un servicio, modelo u opción de cálculo de IA en particular está disponible en otro lugar. Otras porque el propio negocio cambia: un mercado nuevo, una adquisición nueva, un requisito reglamentario nuevo y un estándar de plataforma nuevo.

Eso hace que la portabilidad sea parte de la arquitectura, no una cuestión secundaria. Si el almacenamiento operativo, la búsqueda vectorial, la memoria del agente y el modelo de seguridad tienen que moverse por separado, la pila se fragmenta de nuevo. Si pueden moverse juntos, el sistema de acción puede seguir el negocio.

La plataforma de datos se mueve con usted

La primera década de la nube consistió en elegir un proveedor. La segunda década consiste en operar varios proveedores. Las adquisiciones suponen una nube que el equipo no ha elegido. Las autoridades reguladoras regionales obligan a que las cargas de trabajo permanezcan dentro del país en el proveedor que tenga la zona local. Las mejores herramientas de IA para una carga de trabajo se encuentran en una nube distinta a la de las mejores herramientas de IA para otra carga de trabajo. Y la propia capacidad se ha convertido en una limitación: los equipos acuden a su proveedor de hiperescala y se les informa de que no hay GPU disponibles en la región que necesitan, o de que no hay capacidad alguna.

Ya aparece un patrón en producción. Una empresa de modelos de frontera se queda sin capacidad de GPU en AWS, traslada parte de la carga de trabajo a Google Cloud para TPU y acaba ejecutando su capa de datos distribuida en ambas nubes. La decisión fue forzada por la física, no por la preferencia. La arquitectura la adoptó porque la capa de datos ya era independiente de la nube.

Un solo MongoDB Atlas cluster puede abarcar AWS, Azure y Google Cloud a la vez. El mismo motor, que incluye búsqueda de texto completo y vectorial, ahora se ejecuta en MongoDB Enterprise Advanced para implementaciones on-premises y aisladas donde los requisitos de residencia de datos o soberanía lo exijan.

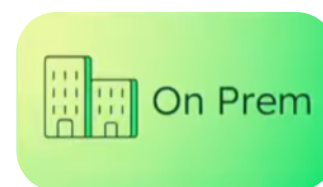
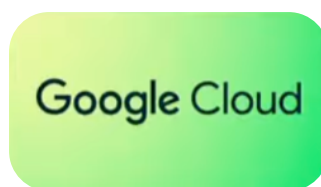
Esto protege contra la clase más costosa de error en la nube: construir una arquitectura de IA que funcione en una nube y se desmorone cuando la empresa agrega, reemplaza o se divide en otra. El año 2025 concluyó con tres interrupciones del servicio de los hiperescaladores que merecen ser mencionadas. Google Cloud en junio. AWS US-EAST-1 el 20 de octubre. Azure Front Door nueve días después. Cada una de ellas dejó sin servicio a los clientes que habían apostado por un único proveedor. Cada una fue tolerable para los clientes cuya capa de datos abarcaba más de una nube.



[LEER EL ARTÍCULO](#)



**Su plataforma de datos se mueve con usted
tanto por nube como por entornos locales**



MongoDB como libro mayor de mandatos para el comercio de agentes: A2A, AP2 y UCP

Este es el artículo más innovador de esta edición, por lo que cierra la sección. El comercio basado en agentes es el siguiente punto de inflexión para la venta minorista. El cliente cada vez delega con más frecuencia la decisión de compra en un agente. El agente compra en nombre del cliente, evalúa las opciones, finaliza la compra y hace seguimiento del pedido. Tres protocolos emergentes están determinando cómo funciona esa interacción: comunicación de agente a agente (A2A), protocolo de pagos de agentes (AP2) y protocolo de comercio universal (UCP).

Cada uno de esos protocolos requiere un libro mayor que registre la autorización del cliente, las acciones del agente y las respuestas del comerciante con el detalle suficiente para facilitar la conciliación, la resolución de disputas y la auditoría. MongoDB sirve como *libro mayor de mandatos* para ese ecosistema emergente. Un mandato es el registro duradero e inviolable de una transacción mediada por un agente, y AP2 define tres tipos: mandatos de intención (lo que el agente está autorizado a hacer),

mandatos de carrito (lo que se está comprando) y mandatos de pago (cómo y cuándo pueden transferirse los fondos). El libro mayor que los contiene debe permitir consultas tanto en dimensiones estructuradas como semánticas, ser de solo adición en la capa de aplicación y capaz de incorporar nuevos tipos de mandatos a medida que evolucionan los protocolos. Eso es exactamente lo que ofrece de forma nativa un almacén de documentos unificado con búsqueda vectorial.

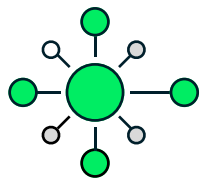
Para los equipos de venta minorista, puede concluirse que adoptar agentes no es solo un problema de integración. Es una oportunidad arquitectónica. Las marcas con mayor probabilidad de admitir transacciones de agentes en 2027 son aquellas cuyos registros contables ya son coherentes.

El ciclo termina en ese punto. Comenzamos con la sobrecarga arquitectónica que las arquitecturas fragmentadas imponen a cada proyecto. Terminamos con la arquitectura que no solo adopta la era de agentes, sino que la respalda.

Figura 20.
Conversación UCP
o compromiso AP2



[LEER EL ARTÍCULO](#)



¿Qué perdurará más que el agente?

Al principio de esta edición, dos equipos comenzaron el mismo proyecto de IA el mismo día. Al final del trimestre, uno se lanzó. El otro tenía una demostración impresionante pero no tenía futuro en la producción. Ahora es una situación conocida: el agente funciona, las diapositivas impresionan y el modelo aparenta funcionar. Hasta que surgen las preguntas reales. ¿Quién es el propietario de los datos? ¿Se puede auditar la respuesta? ¿Qué ocurre cuando el cliente espera, el regulador supervisa y el sistema tiene que actuar?

A lo largo de estas páginas, la respuesta ha sido la misma. McKesson no protegió la seguridad del paciente con un prompt inteligente. LG U+ no mejoró el servicio de atención al cliente en vivo separando la inteligencia de las operaciones. Las instituciones financieras, los fabricantes, los minoristas y los desarrolladores que aparecen en esta edición no están esperando a que surja el modelo perfecto. Están haciendo algo más discreto y duradero: están creando la capa que permite que la inteligencia sea lo suficientemente fiable como para usarla.

No se trata del espectáculo del agente, sino de la base subyacente. La memoria que persiste. Los permisos que surgen de la acción. La información operativa que llega a tiempo. El registro de auditoría que demuestra lo ocurrido posteriormente.

Los modelos se sustituirán. El nombre de los marcos de trabajo cambiará. Los protocolos madurarán.

Solo prevalecerán las arquitecturas que se adapten a todo ese cambio sin pedirle al negocio que comience de nuevo.

La era de los agentes no estará impulsada solo por los agentes. Funcionará gracias a la capa de datos que los hace seguros y confiables.

→ Las últimas páginas de esta edición incluyen un libro de arquitectura y tres insignias de habilidades para los lectores listos para comenzar.



Arquitecturas para la empresa inteligente preparada para la IA

Marcos de trabajo prácticos para construir arquitecturas preparadas para IA que ofrecen un valor empresarial duradero.

Acerca de este recurso

Entregue esto al arquitecto que haya leído la edición y se pregunte, con toda razón, dónde se encuentra el resto.

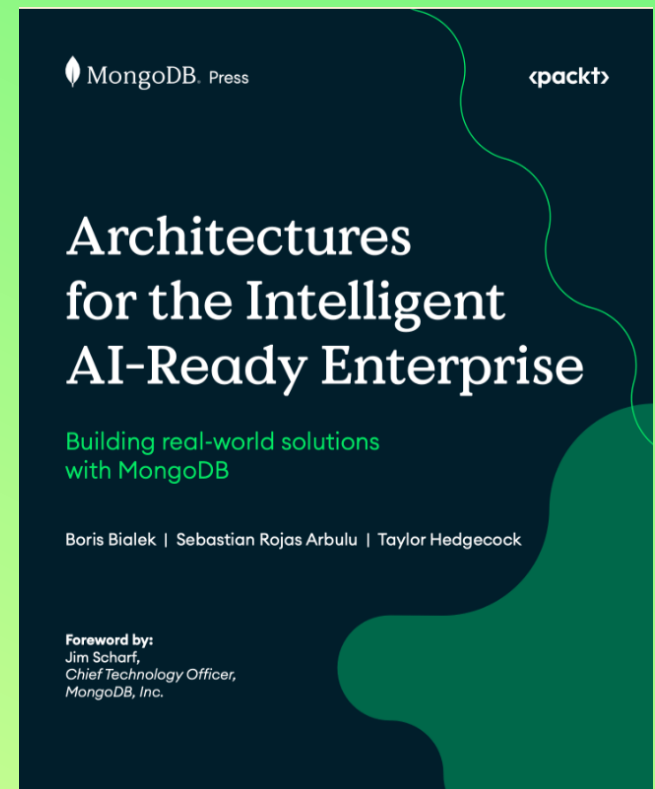
La revista defiende la necesidad de una capa de inteligencia unificada en forma de relato. El libro es la arquitectura de trabajo que hay detrás: más de 500 páginas de patrones, estudios de caso y capítulos que no entran en una edición de este tamaño.

Más de 25 casos de uso de IA en más de 6 industrias. Más de 15 estudios de caso de empresas, incluyendo Novo Nordisk y Base39. Las implementaciones de socios con Iguazio de McKinsey & Company, Cognigy, RegData, Fireworks AI, Dataworkz y Encore que solo se mencionaron en esta edición.

Capítulos fundamentales sobre lo que realmente requiere un Sistema de Acción, la ruta de modernización que permite que los sistemas heredados estén preparados para la IA sin tener que empezar de cero, la protección de datos semánticos que permite que la IA opere en industrias reguladas y los patrones de agentes y multiagentes que convierten una respuesta en una acción.

Con un prólogo de Jim Scharf, director de tecnología de MongoDB.

Cree algo que trascienda el modelo.



[OBTENER EL RECURSO](#)



[VER MUESTRA GRATUITA](#)

Insignias de habilidad

Credenciales prácticas que guían a los lectores desde el concepto hasta la implementación efectiva. Cada insignia combina un módulo enfocado con un laboratorio práctico que se completa en menos de una hora y resulta útil para el resto de un proyecto de IA.



**CONSEGUIR
LA INSIGNIA**



Voyage AI con MongoDB

Desarrolle aplicaciones de IA utilizando las incrustaciones de Voyage AI almacenadas y búsquedas de forma nativa en MongoDB Atlas. Cubre los modelos de incrustación, la ingestión y los fundamentos de la búsqueda semántica. El camino más rápido desde “He oído hablar de las incrustaciones” hasta “He implementado una búsqueda semántica funcional”.



**CONSEGUIR
LA INSIGNIA**

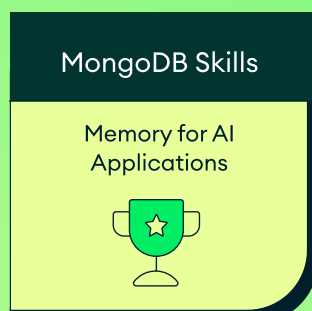


Rendimiento de la búsqueda vectorial

Utilice nodos de búsqueda, técnicas de cuantización y vistas para escalar cargas de trabajo vectoriales desde el prototipo a la producción en MongoDB Atlas. La insignia que se amortiza la primera vez que evita una regresión del rendimiento en la producción.

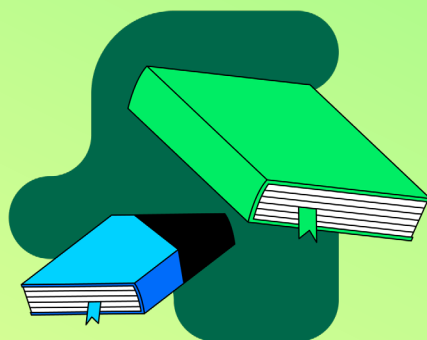


**CONSEGUIR
LA INSIGNIA**



Memoria para aplicaciones de IA

Diseñar memoria duradera y recuperable para agentes de IA. Contexto a corto término, memoria a largo plazo y patrones que mantienen la coherencia de los sistemas de varios agentes a lo largo del tiempo. Si sus agentes son olvidadizos, empiece con este módulo.



MongoDB Education ayuda a los aprendices a desarrollar habilidades prácticas de MongoDB mediante entrenamiento gratuito y flexible, rutas de aprendizaje basadas en roles, insignias de habilidades y certificaciones. MongoDB University hace que sea fácil aprender, validar habilidades y seguir creciendo gracias al contenido de ritmo flexible, los laboratorios prácticos y la instrucción de expertos.



**AVANZAR EN
SU CARRERA**

Recursos adicionales

El siguiente paso dependerá de lo que quiere lograr.



**VISITAR
EL CENTRO**

MongoDB Atlas Architecture Center

Prácticas recomendadas y orientación para diseñar y operar MongoDB Atlas a escala empresarial. Diseñado según el marco de trabajo de Atlas: seguridad, eficiencia operativa, rendimiento, confiabilidad y optimización de costos. La referencia a la que recurren los arquitectos y los ingenieros de plataformas cuando necesitan un plan coherente.



**CONSULTAR
HISTORIAS
DE CLIENTES**

Historias de éxito de clientes

Además de McKesson, LG U+ y las demás empresas que aparecen en esta edición, MongoDB publica cada mes nuevos casos de éxito de sus clientes. La biblioteca completa se puede buscar por sector, caso de uso y resultado. Un punto de partida útil si está desarrollando un caso de negocio internamente y necesita un cliente de referencia en su sector.



**COMENZAR
A APRENDER**

Centro de aprendizaje de IA

Una ruta curada de tutoriales, implementaciones de referencia e insignias de habilidades para desarrolladores que crean aplicaciones de IA en MongoDB. Organizado por nivel de habilidad, desde “Solo quiero lanzar un prototipo de RAG” hasta “Estoy diseñando una plataforma de múltiples agentes para producción”.



¿Quiere saber más?
Póngase en contacto con nosotros.

