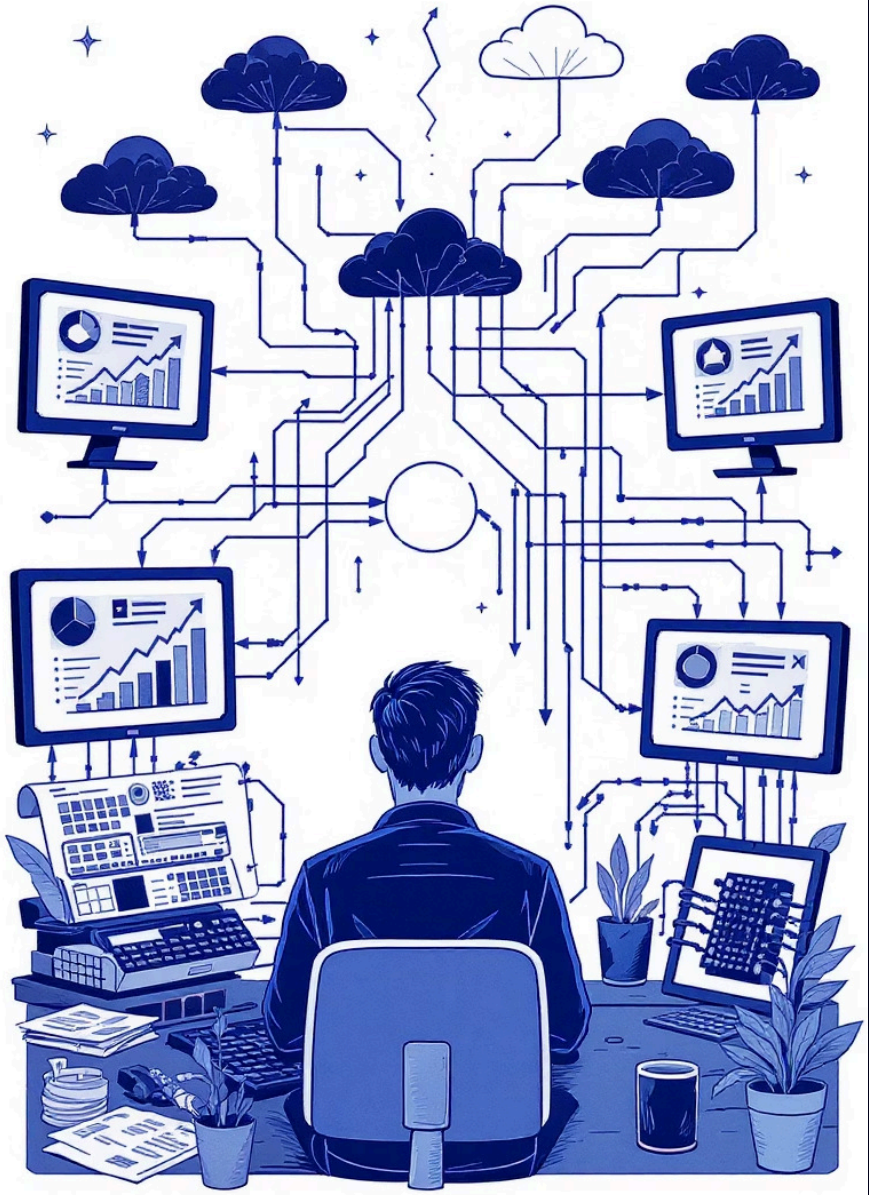


Dateneo®

Guía práctica para la Ingesta de datos

Esta guía te ayudará a considerar todas las aristas y consideraciones clave al diseñar procesos de ingesta de datos, asegurando que tus pipelines sean robustos, escalables y eficientes.

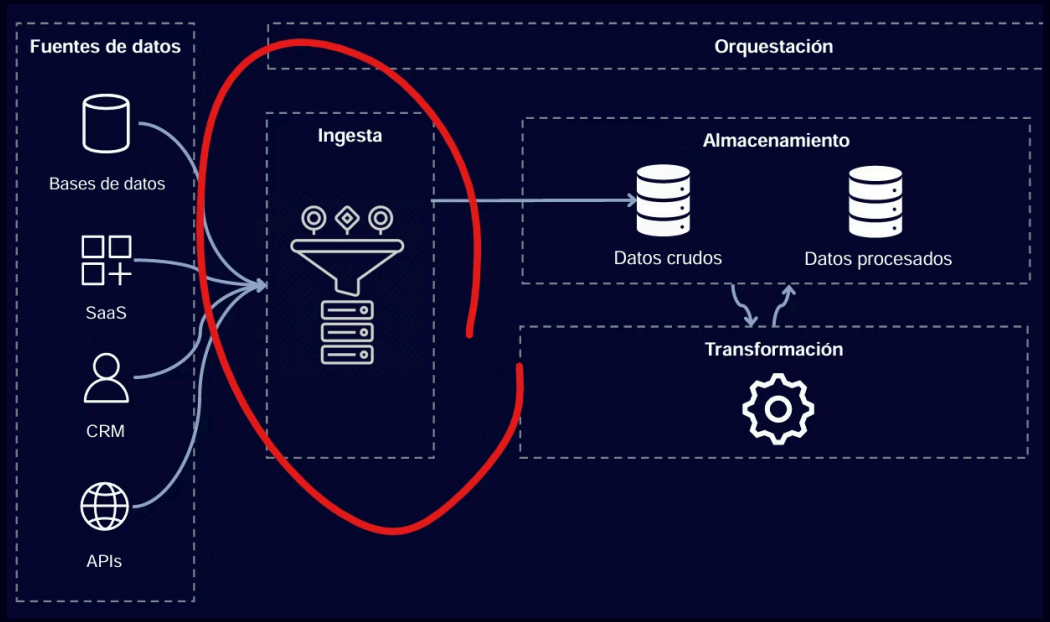
¿Qué es la ingesta de datos?

Es el proceso de mover datos de un lugar a otro dentro del ciclo de vida de la ingeniería de datos.

Implica el movimiento desde sistemas de origen hacia almacenamiento, siendo un paso intermedio pero crucial.

Es importante diferenciar la ingesta de datos de la integración de datos:

- La **ingesta** mueve datos del punto A al B
- La **integración** combina datos de fuentes dispares para crear un nuevo conjunto de datos



La ingesta marca el inicio de los pipelines de datos, donde los ingenieros comienzan a diseñar activamente las actividades del pipeline.

Consideraciones clave para la fase de ingesta

Propósito y reutilización

- ¿Cuál es el caso de uso para los datos que estoy ingestando?
- ¿Puedo reutilizar estos datos y evitar ingerir múltiples versiones del mismo conjunto?

Destino y frecuencia

- ¿A dónde van los datos? ¿Cuál es el destino?
- ¿Con qué frecuencia deben actualizarse los datos desde la fuente?

Volumen y formato

- ¿Cuál es el volumen de datos esperado?
- ¿En qué formato están los datos?
- ¿Puede el almacenamiento y la transformación downstream aceptar este formato?

Calidad y procesamiento

- ¿Los datos de origen están en buena forma para uso inmediato?
- ¿Qué post-procesamiento se requiere para servirlos?
- ¿Cuáles son los riesgos de calidad de datos?

Estas preguntas son fundamentales tanto para ingestas por lotes como para streaming, y aplican a la arquitectura subyacente que crearás, construirás y mantendrás.

Factores de diseño para la arquitectura de ingesta

Datos acotados vs. No acotados

Los datos no acotados fluyen continuamente, mientras que los acotados están agrupados por algún límite, como el tiempo.

Es útil pensar las ingestas desde este precepto:

"Todos los datos son no acotados hasta que se acotan"

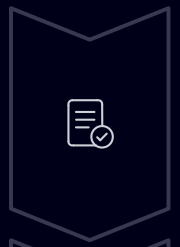
Frecuencia de ingesta

Las frecuencias de ingesta varían desde lentas (anuales) hasta casi en tiempo real (segundos):

- Por **lotes** (batch): horas, días, etc.
- **Micro-batch** o Near Real Time: segundos o minutos.
- Tiempo real (**streaming**): milisegundos.

> La ingesta de datos en tiempo real es mucho más **compleja y costosa** de implementar y mantener.
>
> Por eso, solo conviene adoptarla cuando exista un caso de negocio que realmente lo justifique, como la detección de fraudes al instante o el monitoreo de sistemas críticos, donde la inmediatez es esencial.

Sincronización



Ingesta sincrónica

La fuente, ingesta y destino tienen dependencias complejas y están estrechamente acoplados. Si el proceso A falla, los procesos B y C no pueden iniciarse.

Ingesta asincrónica

Las dependencias operan a nivel de eventos individuales. Los eventos se procesan a medida que están disponibles, en paralelo, según los recursos disponibles.

La asincronía permite procesar datos a medida que llegan, optimizando recursos y evitando saturar el sistema. Puede apoyarse en componentes como **colas de eventos, *streams* de datos, *buffers* intermedios o servicios de mensajería** para manejar picos de carga y mantener un flujo estable.

Rendimiento y fiabilidad

Rendimiento y escalabilidad

- Diseña sistemas que puedan **escalar y reducirse dinámicamente** según la demanda de datos.
- Garantiza la **capacidad de manejar ingestas intermitentes** sin comprometer la estabilidad.
- Implementa **almacenamiento en búfer** para absorber picos de tráfico y evitar saturación.
- **Aprovecha servicios gestionados** que automaticen el escalado y optimicen el rendimiento.

Fiabilidad y durabilidad

- **Fiabilidad**: diseña sistemas que se mantengan **disponibles** la mayor parte del tiempo y que tengan **mecanismos de recuperación automática** ante fallos.
- **Durabilidad**: asegura que los **datos almacenados no se pierdan ni se corrompan**, incluso ante caídas o errores.
- **Redundancia**: incorpora **copias y respaldos** que protejan la información y permitan restaurarla cuando sea necesario.
- **Costos**: evalúa el **equilibrio entre nivel de protección y costos**, considerando tanto gastos directos como impactos por posibles pérdidas de datos.

⚠ Recuerda: tu ingesta nunca debería ser un cuello de botella. Diseña para manejar tanto el volumen normal como los picos inesperados de datos.

Características de un dataset

Tipo

El tipo de datos (tabular, imagen, video, texto) influye directamente en el formato y cómo se expresan en bytes, nombres y extensiones de archivo.

Forma

Describe las dimensiones de los datos: número de filas y columnas, pares clave-valor, profundidad de anidamiento, píxeles, etc.

Tamaño

El número de bytes de una carga útil. Puede variar desde bytes individuales hasta terabytes. Considera la compresión para reducir el tamaño.

Esquema

Describe los campos y tipos de datos. Comprende la organización subyacente de los datos y los patrones de actualización relevantes.

Los cambios de esquema son frecuentes en los sistemas de origen y a menudo están fuera del control de los ingenieros de datos. Es crucial implementar estrategias para detectar y manejar estos cambios.

Patrones de ingesta

Push (Empujar)

Un sistema de origen envía datos a un destino. El origen tiene el control sobre cuándo y cómo se envían los datos.

Pull (Extraer)

Un destino lee datos directamente desde una fuente. El destino tiene el control sobre cuándo y qué datos se extraen.

Poll (Sondear)

Verificación periódica de una fuente de datos para detectar cambios. Cuando se detectan cambios, el destino extrae los datos.

> Las líneas entre estas estrategias son borrosas, y a menudo se utilizan enfoques híbridos según las necesidades específicas.

Consideraciones para ingesta por lotes (Batch)

Tipos

- **Basada en intervalo de tiempo:** procesa datos en intervalos regulares (diario, semanal)
- **Basada en tamaño:** procesa datos cuando se acumula cierta cantidad

Patrones comunes

- Extracción de instantáneas o diferencial
- Exportación e ingesta basada en archivos
- ETL versus ELT
- Inserciones, actualizaciones y tamaño de lote
- Migración de datos

Consideraciones de rendimiento

Los sistemas orientados a lotes suelen tener un rendimiento deficiente cuando se realizan muchas operaciones de lotes pequeños en lugar de un número menor de operaciones grandes.

Comprende los patrones de actualización apropiados para la base de datos o almacén de datos con el que estás trabajando.

Consideraciones para ingesta de mensajes y streaming

Evolución del esquema

Los campos pueden agregarse, eliminarse o cambiar de tipo. Usa registros de esquema para versionar los cambios y mantén comunicación con los stakeholders upstream.

Datos de llegada tardía

Los eventos pueden tener tiempos similares pero llegar en momentos diferentes. Establece un tiempo de corte para cuando los datos de llegada tardía ya no se procesarán.

Ordenamiento y entrega múltiple

Los mensajes pueden entregarse fuera de orden y más de una vez. Diseña para manejar la entrega "al menos una vez" y el posible desorden.

Reproducción y TTL

La reproducción permite solicitar un rango de mensajes del historial. El TTL (tiempo de vida) determina cuánto tiempo se conservan los eventos antes de desaparecer.

Para errores de ingesta, implementa colas de mensajes fallidos (dead-letter queues) para segregar eventos problemáticos y diagnosticar problemas.

Formas de ingestar datos

Conexión directa a base de datos

Usando ODBC, JDBC o API REST para extraer datos directamente de bases de datos. Considera la carga en el sistema de origen y posibles réplicas de lectura.

Captura de Datos de Cambios (CDC)

Ingesta cambios de un sistema de base de datos, ya sea por lotes (basado en campos de actualización) o continuo (basado en logs).

APIs y conectores gestionados

Usa APIs para acceder a datos de plataformas SaaS. Considera conectores gestionados para evitar reinventar la rueda.

Colas de mensajes y plataformas de streaming

Ingesta datos en tiempo real de aplicaciones web, móviles, sensores IoT y dispositivos inteligentes usando Kafka, Kinesis, etc.

Almacenamiento de objetos

Ideal para mover datos hacia/desde lagos de datos, entre equipos y organizaciones. Ofrece seguridad, escalabilidad y alto rendimiento.

Data sharing

Consume conjuntos de datos compartidos por proveedores en plataformas como Snowflake, BigQuery o Redshift sin ingerir físicamente los datos.

Mejores prácticas y consideraciones finales

Colaboración con Stakeholders

- Trabaja con ingenieros de software para mejorar la calidad de los datos en la fuente
- Comunica claramente el valor a los líderes empresariales
- Identifica quiénes son tus clientes finales

Seguridad y gestión de datos

- Cifra los datos en tránsito y en reposo
- Evalúa si realmente necesitas ingerir datos sensibles
- Implementa tokenización cuando sea necesario

DataOps

- Monitorea el tiempo de actividad, latencia y volúmenes de datos
- Implementa pruebas de calidad de datos
- Automatiza la detección de cambios de esquema

Ingeniería de Software

- Usa control de versiones y procesos de revisión de código
- Implementa pruebas apropiadas para código relacionado con la ingesta
- Evita sistemas monolíticos con dependencias estrechas