Ingeniería de Datos

El marco completo.

Juamini 3 Pro Preview

November 23, 2025

La Ingeniería de Datos: El Marco Completo

1. ¿Qué es la Ingeniería de Datos? (La Misión)

Olvídate por un momento de Spark, Python o SQL. La definición fundamental de Joe Reis es funcional:

> **La Ingeniería de Datos es la disciplina encargada de tomar datos crudos (ingredientes) y transformarlos en un producto de alta calidad y consistente para que otros (Analistas, Científicos de Datos, ML) puedan consumirlo y generar valor.**

Si el dato no se usa o no es confiable, la ingeniería de datos falló, sin importar cuán complejo sea el código.

---

2. El "QUÉ": El Ciclo de Vida de la Ingeniería de Datos

El libro propone dejar de mirar la tecnología (que cambia cada mes) y mirar el **Ciclo de Vida** (que no cambia). Imagina esto como una línea de montaje en una fábrica.

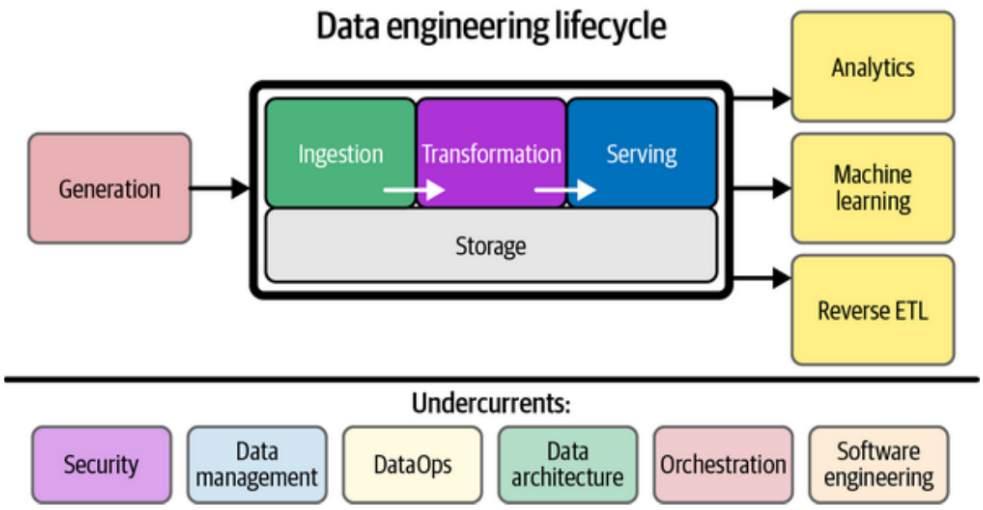

Figura 2-1: Componentes y corrientes subyacentes del ciclo de vida de la ingeniería de datos

Etapa A: Generación (Source Systems)

- **El Origen:** Los datos nacen en sistemas fuente (una App, un sensor IoT, un CRM como Salesforce).
- **El Rol del Ingeniero:** Típicamente **no eres dueño** de la fuente. Debes "llevarte bien" con ella.
- **Concepto Crítico:** El **Esquema (Schema)**.
  - *Schema-on-write (SQL tradicional):* Rígido, ordenado.
  - *Schema-on-read (NoSQL/Logs):* Flexible, pero te pasa el problema de limpieza a ti.
- **Riesgo:** Si el desarrollador de la App cambia una columna sin avisar, tu pipeline se rompe.

Etapa B: Almacenamiento (Storage)

- **El Cimiento:** No es solo un paso; los datos se almacenan antes, durante y después del proceso.

- **La Evolución:**

    1. **Data Warehouse:** Caro, rápido, estructurado (ACID).

    2. **Data Lake:** Barato, flexible, desordenado (archivos crudos).

    3. **Lakehouse:** La convergencia. Almacenamiento barato (archivos) con gestión inteligente (Delta Lake/ACID).

- **Temperatura del Dato:**

    - *Hot:* Acceso inmediato (Memoria/Cache).

    - *Lukewarm/Cold:* Acceso esporádico (S3/Glacier).

Etapa C: Ingesta (Ingestion)

- **El Movimiento:** Mover datos de A a B.

- **La Gran Decisión:** ¿Frecuencia?

    - **Batch (Lotes):** Proceso un bloque de datos cada hora/día. Es robusto, fácil de reintentar y barato. (El estándar por defecto).

    - **Streaming (Tiempo Real):** Proceso evento por evento. Es complejo y caro. Solo úsalo si el negocio realmente necesita saber algo *milisegundos* después de que ocurrió.

Etapa D: Transformación

- **El Valor:** Aquí ocurre la alquimia. El dato crudo se convierte en información.

- **Tipos:** Limpieza (quitar nulos), Normalización (dar formato), Agregación (sumar ventas), Featurization (preparar para ML).

- **Lógica de Negocio:** Aquí es donde el código refleja las reglas de la empresa (ej: "Una venta solo cuenta si ya fue pagada").

Etapa E: Servicio (Serving)

- **El Consumo:** Entregar el dato al usuario final.

- **Destinos:**
  - *Analítica (BI):* Dashboards para humanos.
  - *Machine Learning:* Datos para entrenar modelos.
  - *Reverse ETL:* Devolver datos procesados a los sistemas operativos (ej: mandarle el "Score de Riesgo" calculado de vuelta a Salesforce para que el vendedor lo vea).

---

3. El "CÓMO": Las Corrientes Subyacentes (Undercurrents)

Aquí es donde muchos fallan. No puedes simplemente conectar cables (etapas del ciclo de vida) y esperar que funcione. Necesitas **soporte transversal**. Estas corrientes afectan a **todas** las etapas del ciclo de vida simultáneamente.

1. **Seguridad:** No es un parche al final. Es el principio de *Menor Privilegio* aplicado desde que el dato se genera hasta que se sirve.
2. **Gestión de Datos (Data Management):**
   - *Gobernanza:* ¿Quién es el dueño?
   - *Calidad:* ¿El dato es basura?
   - *Linaje:* Si el reporte está mal, ¿puedo rastrear hacia atrás dónde se rompió?
3. **DataOps:** La automatización de todo. Usar CI/CD, control de versiones y monitoreo para no hacer cambios manuales peligrosos en producción.
4. **Arquitectura:** Diseñar el plano. Decidir si usar Lakehouse o Warehouse.
5. **Orquestación:** El director de orquesta (Airflow/Dagster). Coordina que la Ingesta ocurra antes que la Transformación. Gestiona dependencias.
6. **Ingeniería de Software:** Escribir código limpio, modular y testeable. Los datos son código.

---

4. El "CRITERIO": Los Ejes de Optimización

¿Cómo decides qué herramienta usar en cada etapa? ¿Cómo eliges entre Batch o Streaming? ¿Entre AWS o Azure? Aquí es donde aplicas los **Trade-offs (Compensaciones)**. Nunca puedes tener todo perfecto; debes equilibrar estos ejes:

1. **Costo:** Dinero (nube) y Tiempo (salarios).
2. **Agilidad:** ¿Qué tan rápido puedo entregar cambios?
3. **Escalabilidad:** ¿Aguanta si los datos se multiplican por 100?
4. **Simplicidad:** ¿Es fácil de mantener o creé un monstruo?
5. **Reutilización:** ¿Puedo usar esto para otro proyecto?
6. **Interoperabilidad:** ¿Esta herramienta habla bien con las otras?

La Gran Conclusión (Para tu Examen)

La Ingeniería de Datos no es saber usar Spark o Kafka. La Ingeniería de Datos es **gestionar el Ciclo de Vida de los datos** (Ingesta -> Transformación -> Servicio), asegurándose de aplicar las **Corrientes Subyacentes** (Seguridad, Calidad, DataOps) en cada paso, y tomando decisiones de arquitectura basadas en **Ejes de Optimización** (Costo vs Rendimiento) para entregar valor al negocio.

Si entiendes este flujo lógico, estás listo para rendir. ¡Mucha suerte!