

Ingesta de datos

Big Data




Introducción


- Formalmente, la ingesta de datos es el proceso mediante el cual se introducen datos, desde diferentes fuentes, estructura y/o características dentro de otro sistema de almacenamiento o procesamiento de datos.



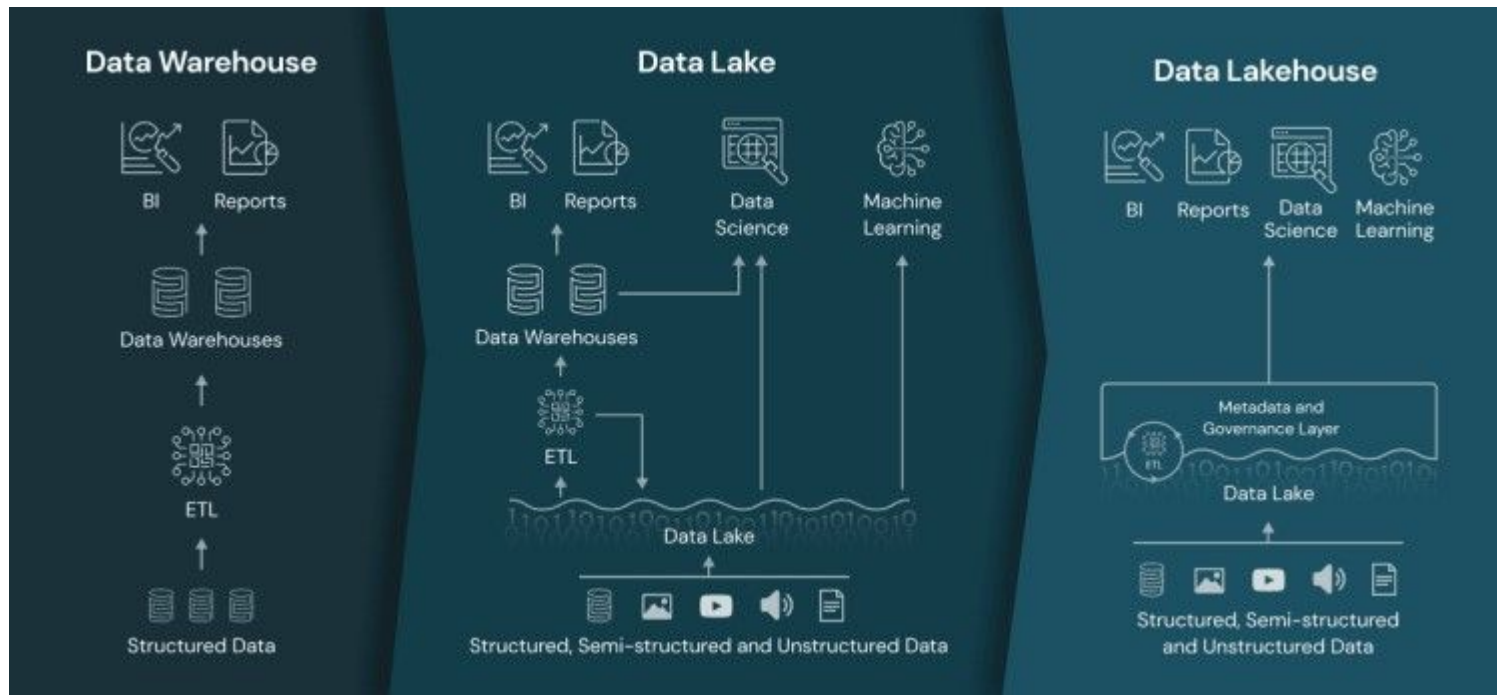
Introducción

- Estos procesos deben ser flexibles y ágiles, ya que una vez puesta en marcha, los analistas y científicos de datos puedan construir un pipeline de datos para mover los datos a la herramienta con la que trabajen.
 - Entendemos como **pipeline** de datos un proceso que consume datos desde un punto de origen, los limpia y los escribe en un nuevo destino.
- 

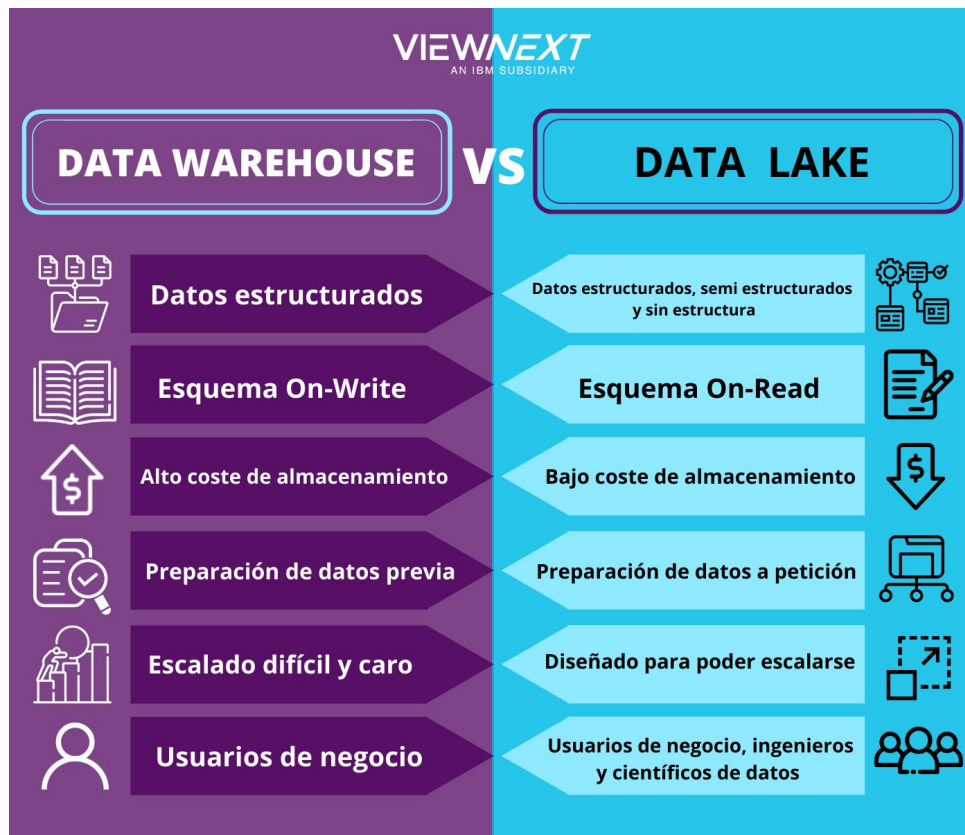
Introducción

- Dada la gran cantidad de datos que disponen las empresas, toda la información que generan desde diferentes fuentes se deben integrar en un único lugar (**data lake**), asegurándose que los datos son compatibles entre sí.
 - Gestionar tal volumen de datos puede llegar a ser un procedimiento complejo, normalmente dividido en procesos distintos y de relativamente larga duración
- 

Data Lake



Data Warehouse vs Data Lake



Pipeline de datos

- Un **pipeline** es una construcción lógica que representa un proceso dividido en fases.
- Los pipelines de datos se caracterizan por definir el conjunto de pasos y las tecnologías involucradas en un proceso de movimiento o procesamiento de datos.




Pipeline de datos

- Son necesarios, ya que no debemos analizar los datos en los mismos sistemas donde se crean (principalmente para evitar problemas de rendimiento).
- El proceso de analítica es costoso computacionalmente, por lo que se separa para evitar perjudicar el rendimiento del servicio.



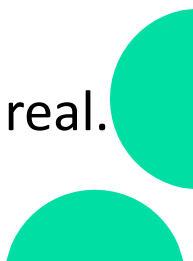
Pipeline de datos

- De esta forma, tenemos sistemas **OLTP** (como un CRM), encargados de capturar y crear datos, y de forma separada, sistemas **OLAP** (como un Data Warehouse), encargados de analizar los datos.
 - OLTP (Sistemas de Procesamiento Transaccional Online)
 - OLAP (Sistemas de Procesamiento Analítico)
- 

1. Sistemas OLTP (Online Transaction Processing)

- Se encargan de procesar transacciones en tiempo real.
- Son sistemas diseñados para capturar, almacenar y gestionar datos.
- Se enfocan en la velocidad y la fiabilidad de las operaciones, ya que suelen manejar grandes volúmenes de transacciones.

Ejemplo: Cuando un cliente realiza una compra en un sistema CRM, como Salesforce, el sistema OLTP registra:


- El nombre del cliente, la fecha de la transacción, los productos comprados, el precio total, etc.
 - En este caso, el sistema OLTP está capturando datos en tiempo real.
- 

2. Sistemas OLAP (Online Analytical Processing)

- Analizan grandes cantidades de datos para tomar decisiones estratégicas.
- Trabajan con datos históricos y procesados, y permiten consultas complejas (como informes, gráficos o predicciones).
- Están diseñados para ser eficientes en análisis y generación de informes, no en la captura de datos en tiempo real.

Ejemplo: Imagina que los datos capturados por el CRM se almacenan y se analizan en un Data Warehouse (como Google BigQuery o Amazon Redshift).

Los gerentes podrían generar reportes para responder preguntas como:

- ¿Qué productos son los más vendidos en los últimos 6 meses?
 - ¿Cuál es el comportamiento de los clientes según su ubicación?
 - ¿Qué clientes tienen el mayor valor de compra promedio?
- 

Diferencia práctica OLTP-OLAP

1. OLTP (Captura):

- Cliente “María López” compra un producto en línea a las 10:30 a.m.
- Se registra inmediatamente en el sistema OLTP del CRM.

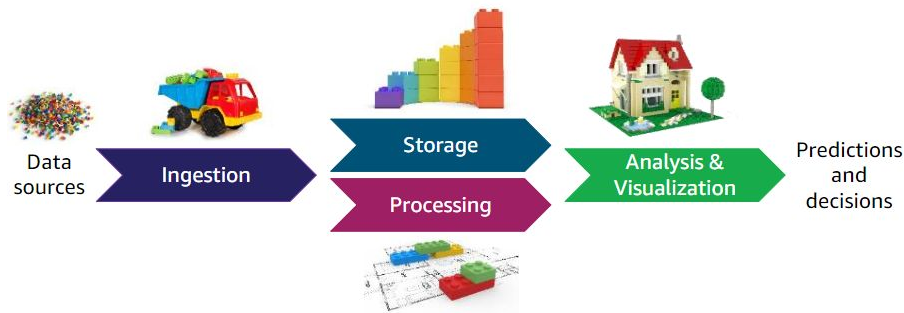
2. OLAP (Análisis):

- Al final del mes, el sistema OLAP genera un informe que muestra que “María López” gastó \$500 en los últimos 30 días, y su producto más comprado fue “Zapatos Deportivos”.


Pipeline de datos

- Los movimientos de datos entre estos sistemas involucran varias fases. Por ejemplo:

1. **Ingesta:** Recogemos los datos y los enviamos a un topic de Apache Kafka.
2. **Almacenamiento.** Kafka actúa aquí como un buffer para el siguiente paso.




Pipeline de datos

3. **Procesamiento:** Mediante una tecnología de procesamiento, que puede ser streaming o batch, leemos los datos del buffer.
 4. **Análisis y Visualización:** Por ejemplo, mediante Spark realizamos la analítica sobre estos datos (haciendo cálculos, filtrados, agrupaciones de datos, etc...).
 5. Finalmente, podemos visualizar los resultado o almacenarlos en una base de datos NoSQL como DynamoDB o un sistema de almacenamiento distribuido como AWS S3.
- 

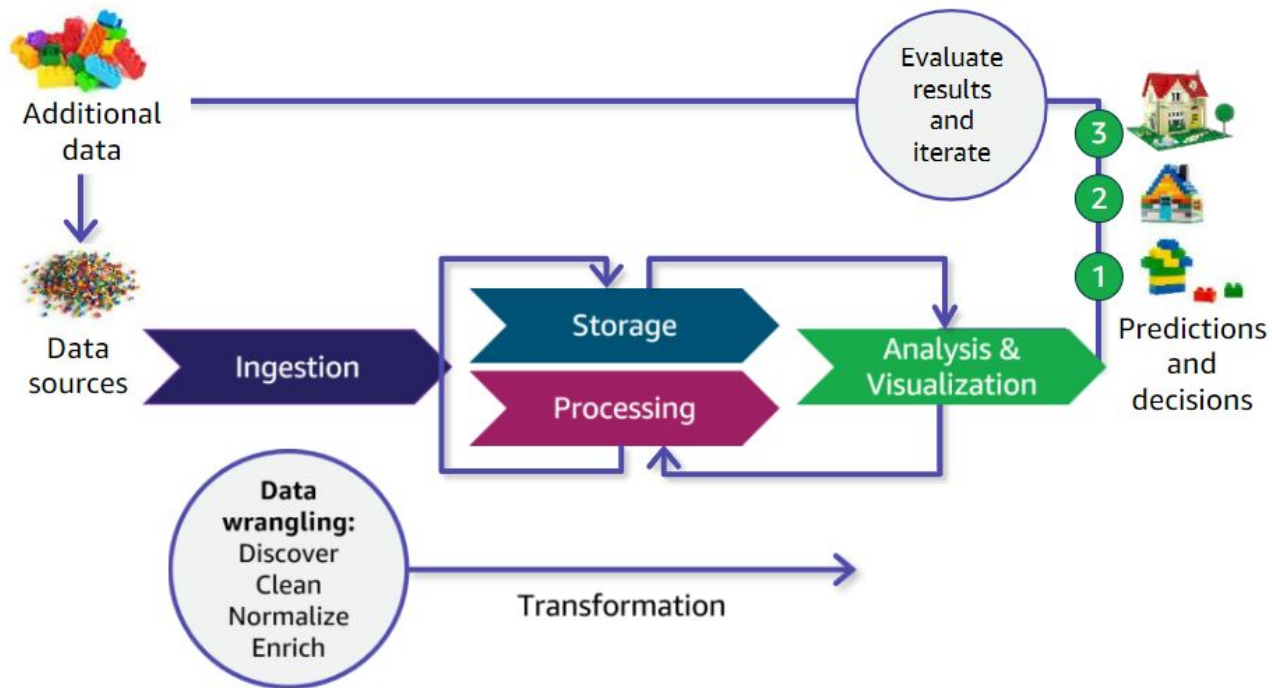
Pipeline iterativo

En la práctica, comprobaremos los datos almacenados, y si no disponemos de la información necesaria, recogeremos nuevos datos. Estos nuevos datos pasarán por todo el pipeline, integrándose con los datos ya existentes.

En la fase de analítica, si no obtenemos el resultado esperado, nos tocará volver a la fase de ingesta para obtener o modificar los datos recogidos, y así, de forma iterativa, hasta producir el resultado esperado.



Pipeline iterativo




Desarrollo iterativo de un pipeline de datos - AWS

Pipeline de datos vs ETL

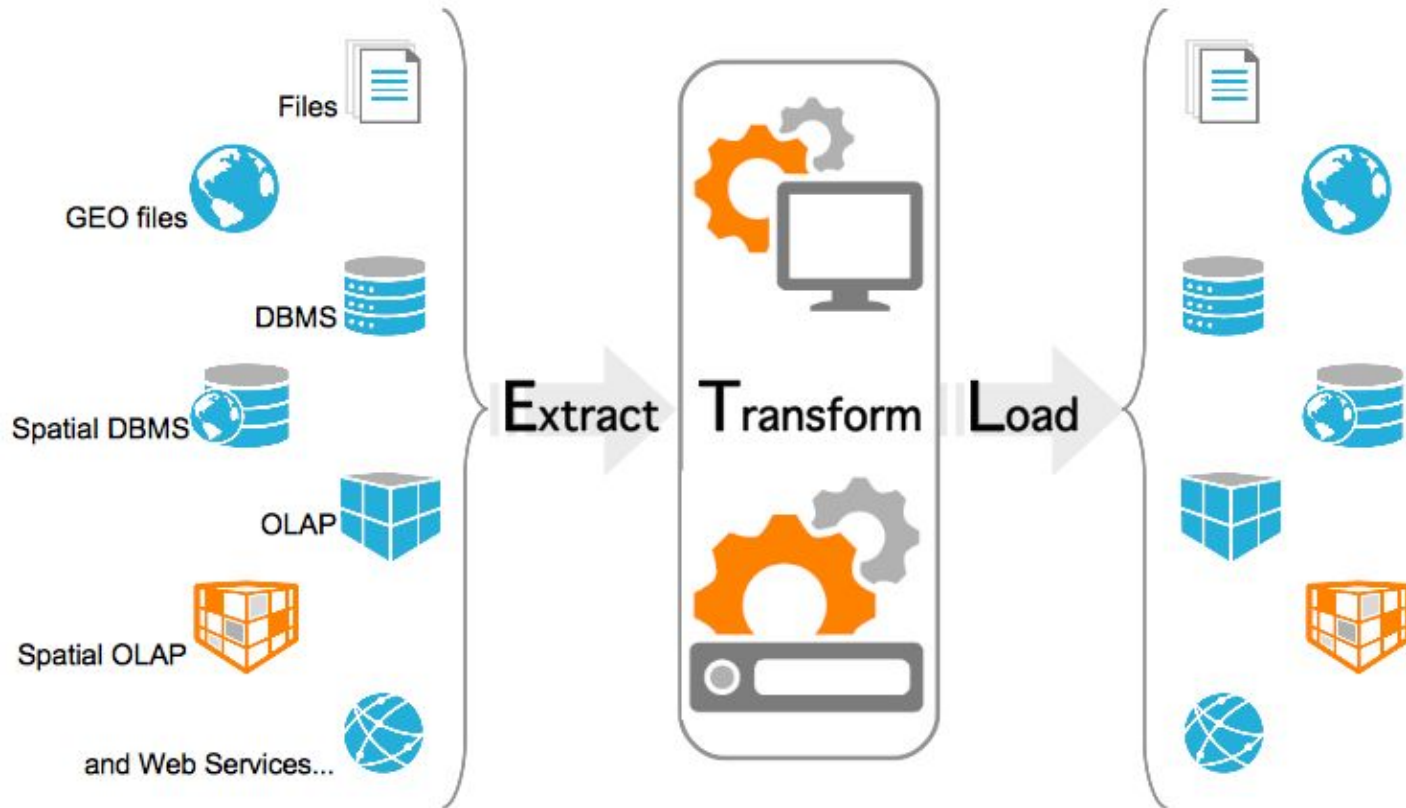
Aunque a menudo se intercambian los términos de pipeline de datos y ETL, no significan lo mismo.

Las **ETLs** son un caso particular de pipeline de datos que involucran las fases de extracción, transformación y carga de datos.

Los **pipelines de datos** son cualquier proceso que involucre el movimiento de datos entre sistemas.

Two overlapping teal circles are located in the bottom right corner of the slide, serving as a decorative element.


ETL



Extracción

Encargada de recopilar los datos de los sistemas originales y transportarlos al sistema donde se almacenarán, de manera general suele tratarse de un entorno de Data Warehouse o almacén de datos.

Las fuentes de datos pueden encontrarse en diferentes formatos, desde ficheros planos hasta bases de datos relacionales, pasando por mensajes de redes sociales como Twitter (X) o Reddit.

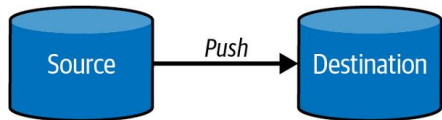


Extracción: Estrategias de ingesta

A la hora de obtener datos desde una fuente origen y llevarlos a un destino, podemos seguir tres planteamientos:

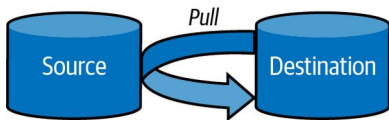
Push

El origen envía los datos al destino



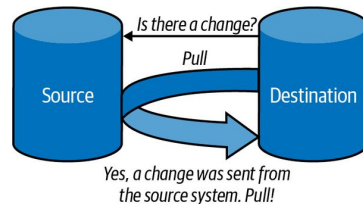
Pull

El destino recupera los datos del origen




Poll

El destino comprueba de forma periódica el origen para ver si ha habido cambios. Si hubiera cambios, hace un *pull* de los mismos.



Transformación


En esta fase se espera realizar los cambios necesarios en los datos de manera que estos tengan el formato y contenido esperado.

- Cambios de codificación.
 - Eliminar datos duplicados.
 - Cruzar diferentes fuentes de datos para obtener una fuente diferente.
 - Agregar información en función de alguna variable.
 - Tomar parte de los datos para cargarlos.
 - Transformar información para generar códigos, claves, identificadores...
 - Estructurar mejor la información.
 - Generar indicadores que faciliten el procesamiento y entendimiento.
- 
- Two large, overlapping teal circles are located in the bottom right corner of the slide, serving as a decorative element.

Carga

Fase encargada de almacenar los datos en el destino, un data warehouse o en cualquier tipo de base de datos. Por tanto, la fase de carga interactúa de manera directa con el sistema destino, y debe adaptarse al mismo con el fin de cargar los datos de manera satisfactoria.

Cada BBDD puede tener un sistema ideal de carga basado en:

- SQL (Oracle, SQL Server, Redshift, PostgreSQL, ...)
 - Ficheros (PostgreSQL, Redshift, ...)
 - Cargadores Propios (HDFS, S3, ...)
- 

ELT

- ELT cambia el orden de las siglas y se basa en extraer, cargar y transformar. Es una técnica de ingestión de datos donde los datos que se obtienen desde múltiples fuentes se colocan sin transformar directamente en un data lake o almacenamiento de objetos en la nube. Desde ahí, los datos se pueden transformar dependiendo de los diferentes objetivos de negocio.
- ELT permite que los analistas y científicos de datos realicen las transformaciones, ya sea con SQL o mediante Python.

ELT

- Los ingenieros de datos generan un data lake con los datos obtenidos de las fuentes de datos más populares, dejando que la transformación la realicen los expertos en el negocio.
- Esto también implica que los datos estén disponibles antes, ya que mediante un proceso ETL los datos no están disponibles para los usuarios hasta que se han transformado.



Herramientas ETL

Las soluciones más empleadas son:

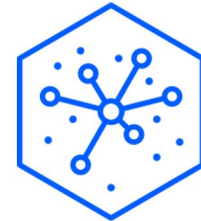
- Pentaho Data Integration (PDI)
- NiFi
- Oracle Data Integrator
- Talend Open Studio
- Mulesoft
- IBM Cloud Pak
- Informatica Data Integration



Herramientas ETL




IBM Cloud Pak for Data




Herramientas ETL

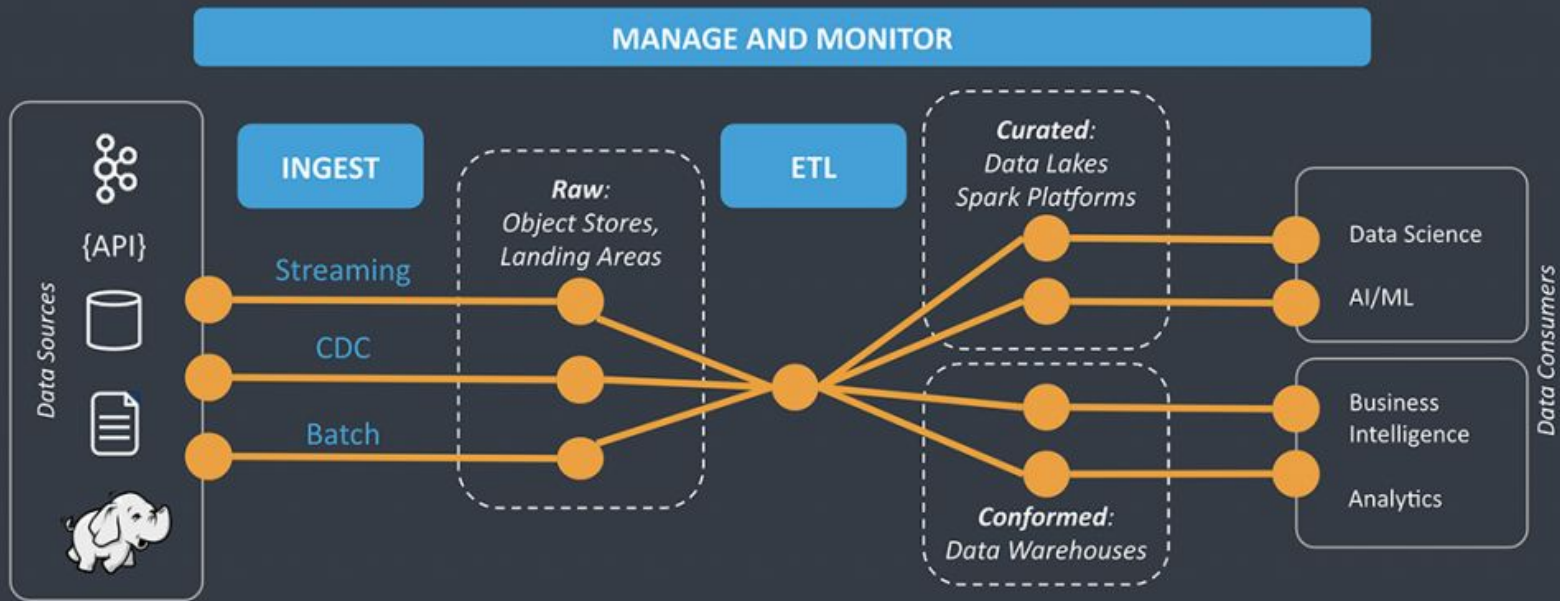
Las características de las herramientas ETL son:

- Permiten conectividad con diferentes sistemas y tipos de datos.
 - Excel, BBDD transaccionales, XML, ficheros CSV / JSON, Teradata, HDFS, Hive, S3, ...
 - Peticiones HTTP, servicios REST...
 - APIs de aplicaciones de terceros, logs...
 - Permiten la planificación mediante batch, eventos o en streaming.
- 

Herramientas ETL


- Capacidad para transformar los datos:
 - Transformaciones simples: tipos de datos, cadenas, codificaciones, cálculos simples.
 - Transformaciones intermedias: agregaciones.
 - Transformaciones complejas: algoritmos de IA, segmentación, integración de código de terceros, integración con otros lenguajes.
 - Metadatos y gestión de errores
 - Permiten tener información del funcionamiento de todo el proceso.
 - Permiten el control de errores y establecer políticas al respecto.
- 

La ingesta por dentro



La ingesta de datos

Las fuentes más comunes desde las que se obtienen los datos son:

- Servicios de mensajería como Apache Kafka, los cuales han obtenido datos desde fuentes externas, como pueden ser dispositivos IOT o contenido obtenido directamente de las redes sociales.
 - Bases de datos relacionales, las cuales se acceden, por ejemplo, mediante JDBC.
 - Servicios REST que devuelven los datos en formato JSON.
 - Servicios de almacenamiento distribuido como HDFS o S3.
- 

La ingesta de datos

Los destinos donde se almacenan los datos son:

- Servicios de mensajería como Apache Kafka.
- Bases de datos relacionales.
- Bases de datos NoSQL.
- Servicios de almacenamiento distribuido como HDFS o S3.
- Plataformas de datos como Snowflake o Databricks.



Batch vs Streaming

El movimiento de datos entre los orígenes y los destinos se puede hacer mediante un proceso:

- Batch: el proceso se ejecuta de forma periódica a partir de unos datos estáticos. Muy eficiente para grandes volúmenes de datos, y donde la latencia no es el factor más importante. Algunas de las herramientas utilizadas son Apache Sqoop, trabajos en MapReduce o de Spark jobs, etc...
- Streaming: tiempo real, donde los datos se leen, modifican y cargan tan pronto como llegan a la capa de ingesta (la latencia es crítica). Ejemplos: Apache Storm, Spark Streaming, Nifi, Kafka, etc...

Herramientas de Ingesta de datos



Apache Sqoop:

Permite la transferencia bidireccional de datos entre Hadoop/Hive/HBase y una base de datos SQL (datos estructurados). Aunque principalmente se interactúa mediante comandos, proporciona una API Java.

Herramientas de Ingesta de datos



Apache Flume

Sistema de ingesta de datos semiestructurados o no estructurados sobre HDFS o HBase mediante una arquitectura basada en flujos de datos en streaming.

Herramientas de Ingesta de datos



Apache Nifi

Herramienta que facilita una interfaz gráfica que permite cargar datos de diferentes fuentes (tanto batch como streaming), los pasa por un flujo de procesos (mediante grafos dirigidos) para su tratamiento y transformación, y los vuelca en otra fuente.

Herramientas de Ingesta de datos




AWS Glue

Servicio gestionado para realizar tareas ETL desde la consola de AWS. Facilita el descubrimiento de datos y esquemas. También se utiliza como almacenamiento de servicios como Amazon Athena o AWS Data Pipeline.

Herramientas de Ingesta de datos

Por otro lado existen sistemas de mensajería con funciones propias de ingesta, tales como:


- **Apache Kafka:** sistema de intermediación de mensajes basado en el modelo publicador/suscriptor.
 - **RabbitMQ:** sistema de colas de mensajes (MQ) que actúa de middleware entre productores y consumidores.
 - **Amazon Kinesis:** homólogo de Kafka para la infraestructura AWS.
 - **Microsoft Azure Event Hubs:** homólogo de Kafka para la infraestructura Microsoft Azure.
 - **Google Pub/Sub:** homólogo de Kafka para Google Cloud.
- 

Pentaho Data Integration (PDI)

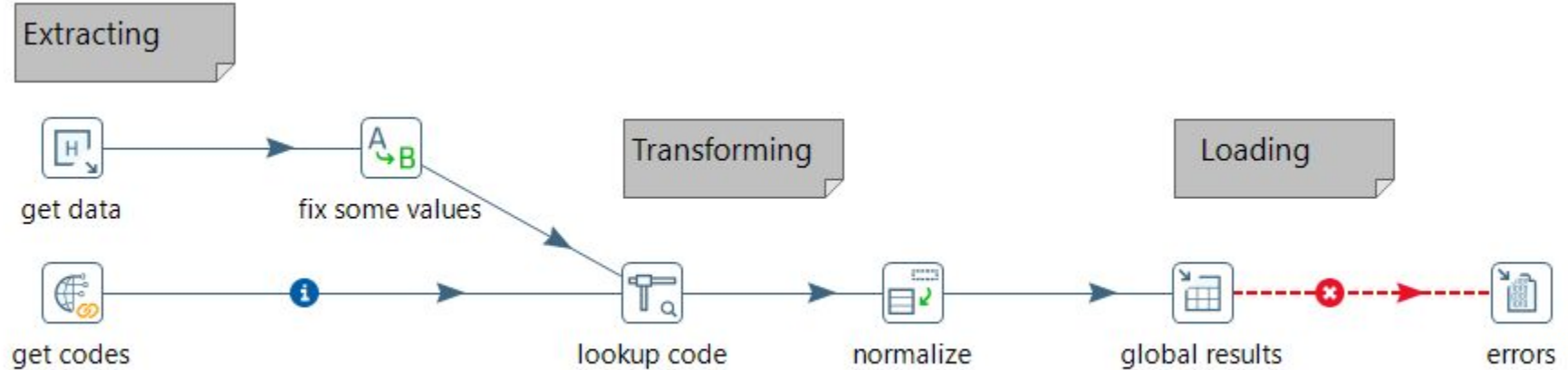
- **Pentaho Data Integration (PDI)** es una herramienta de código abierto para integración de datos y ETL. PDI se ejecuta en una máquina y le permite a los usuarios diseñar, probar y ejecutar flujos de trabajo de integración de datos.
- PDI proporciona una plataforma para integrar datos de múltiples fuentes y aplicaciones y permite a los usuarios crear flujos de trabajo visualmente mediante el uso de gráficos y diagramas. Los flujos de trabajo se pueden ejecutar de manera manual o programada y se pueden escalar para manejar grandes volúmenes de datos.

Pentaho Data Integration (PDI)

Kettle es un componente de Pentaho Data Integration (<https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho/download-pentaho.html>) que a su vez contiene a **Spoon**. Mediante Spoon se pueden realizar procesos ETL de manera visual, de forma muy fácil y rápida, como por ejemplo:

- Conexiones a los datos.
 - Transformaciones (filtrado, limpieza, formateado, ... y posterior almacenamiento en diferentes formatos y destinos).
 - Inserción de fórmulas.
 - Desarrollo de data warehouses con estructura en estrella.
- 

Pentaho Data Integration (PDI)



Pentaho - Ejemplo de flujo ETL