

Procesamiento de datos (Data Mining)

Pipeline de Datos

Clase 3

MAESTRÍA EN
SISTEMAS DE INFORMACIÓN
Mención Data Science

La excelencia no se improvisa



1. INTRODUCCIÓN DE LA CLASE (2 párrafos)

La presente clase introduce la tubería de datos en ciencia de datos, también denominada Pipeline de datos o Data Science Pipeline. Para ello, contextualiza metodologías utilizadas en la construcción de tuberías tecnológicas que permiten mover datos de un sitio a otro, como es el caso de OLAP y ETL. También es necesario conocer cómo se extraen los datos en sistemas transaccionales, ya que la complejidad de la tubería depende de ello.

El contenido hace referencia tanto a herramientas de software libre como a software propietario, ubicándolas dentro de los pasos de construcción de la tubería. La literatura utilizada está actualizada hasta 2024 e inicios de 2025, debido a la constante evolución de nuevas herramientas que combinan varios pasos del pipeline.

Clase 3: Pipeline de Datos

RdA.1. Aplicar las técnicas de preprocesamiento adecuadas para diferentes tipos de datos y problemas que se presentan en las organizaciones.

3. Pipeline de datos

3.1. Introducción

Pipeline es un término inglés que significa tubería o conducto, dando la idea que hay una conducción de algo a través de una conexión entre un punto y otro. Un pipeline de datos es un proceso. Es una esquematización lógica de un conjunto de pasos o fases. Un pipeline de datos define las fases y herramientas incorporadas en el procesamiento de datos. Un pipeline de datos optimiza los procesos y potencia la capacidad de extraer valor estratégico de los datos.

Por las distintas maneras en que se desarrolla la tecnología y sus metodologías y herramientas se intersecan; así los pipelines de ciencia de datos integran conceptos como ETL (Extract Transform Load) pipeline, OLAP (Online Transaction Processing), OLTP (Online Analytical Processing). En términos generales los sistemas OLTP generan los datos operativos que un ETL pipeline transforma y organiza para alimentar datos a sistemas OLAP para análisis o a un Data Science pipeline para crear modelos predictivos.

3.2. Componentes

Dependiendo de la metodología, del esquema de procesamiento de datos o de los objetivos de extracción, los componentes del pipeline de datos pueden variar en su alcance. Sin embargo, es posible generalizar el pipeline de datos en tres fases: ingesta, procesamiento y almacenamiento. Como se observa en la Ilustración 1, la ingesta consiste en extraer y almacenar datos. El procesamiento organiza los datos con un propósito específico.

Finalmente, los datos organizados se almacenan en bases de datos desde donde pueden extraerse conocimientos y visualizarse de diversas maneras. Para comprender mejor estas fases, se presentan cuatro componentes fundamentales del pipeline de datos: OLAP, OLTP, ETL pipeline y Data Science Pipeline. Al final, estos componentes se integran en una propuesta general de pipeline de datos: <https://youtu.be/VtzvF17ysbc?si=mnDGB1tRf73w64gs>

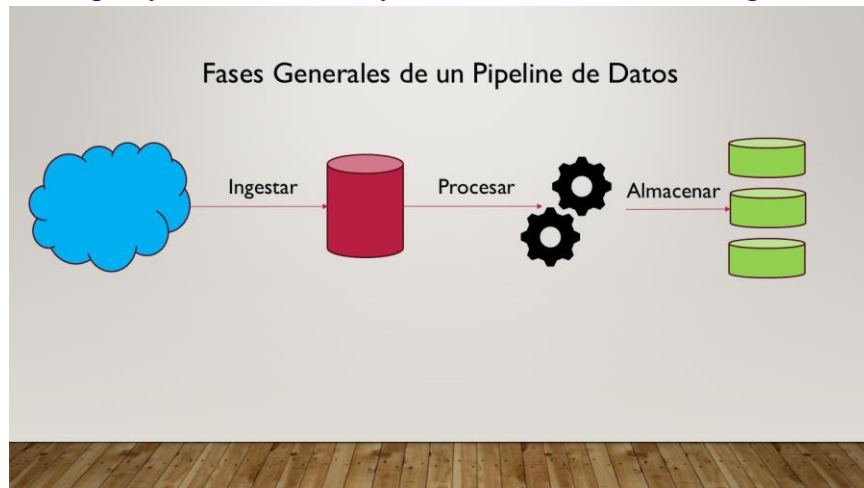


Ilustración 1. Fases Generales de un Pipeline de Datos (Melgarejo, 2025)

Para una mejor comprensión de estas fases, se analizan a continuación cuatro componentes del pipeline de datos: OLAP, OLTP, ETL pipeline y Data Science Pipeline. Al final, estos componentes son integrados en una propuesta general de pipeline de datos.

3.2.1. OLTP

3.2.1.1. Definición

OLTP (del inglés Online Transaction Processing ~ Procesamiento Transaccional en Línea) es un sistemas que capturan y crean datos.

3.2.1.2. Propósito (Fernández, 2024)

OLTP está diseñado para gestionar transacciones en tiempo real, como las operaciones bancarias. Estas operaciones requieren alta velocidad (con tiempos de respuesta muy bajos), capacidad de procesar un gran número de transacciones simultáneamente, consistencia y fiabilidad de los datos. Los usuarios de los sistemas OLTP son operativos, como el cliente que retira dinero de un cajero automático o el cajero bancario que atiende a los clientes.

Tipos de datos y Operaciones

OLTP gestiona datos de transacciones realizadas en tiempo real, actualizando dinámicamente los registros en bases de datos, como cuentas bancarias o inventarios (ventas y stock).

Para garantizar una gestión eficiente, los datos en OLTP suelen ser específicos y de tamaño reducido, y las operaciones incluyen creación, lectura, actualización y eliminación (CRUD: Create, Read, Update, Delete).

3.2.1.3. Diseño y estructura

OLTP posee una estructura optimizada para la integridad referencial y la rapidez de las transacciones. La integridad es alcanzada por normalización 3FN (tercera forma normal de diseño de base de datos relacional para reducir la duplicación de los datos) o superior, lo que reduce la duplicación de datos. La Ilustración 2 muestra un diseño general de OLTP.

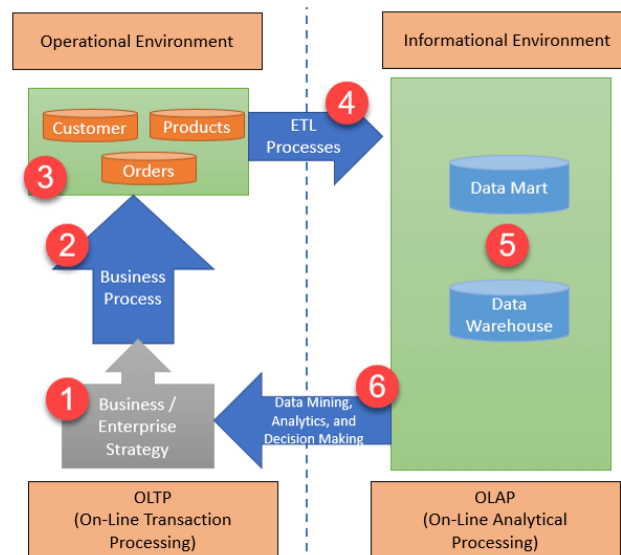


Ilustración 2. OLTP (Carter, 2024)

3.2.2. OLAP

3.2.2.1. Definición

OLAP (del inglés Online Analytical Processing ~ Procesamiento Analítico en Línea) son sistemas que analizan datos. Separar OLAP de OLTP es conveniente, ya que el análisis de datos es costoso en términos computacionales y puede afectar el rendimiento de los sistemas transaccionales.

Propósito (Fernández, 2024)

OLAP permite analizar datos históricos complejos y generar reportes para la toma de decisiones estratégicas y operativas, como informes financieros y estudios de mercado. A diferencia de OLTP, OLAP maneja menos consultas, pero estas pueden requerir tiempos de respuesta prolongados. Los usuarios principales de OLAP son analistas de datos. La Ilustración 3 muestra el proceso general de OLAP.

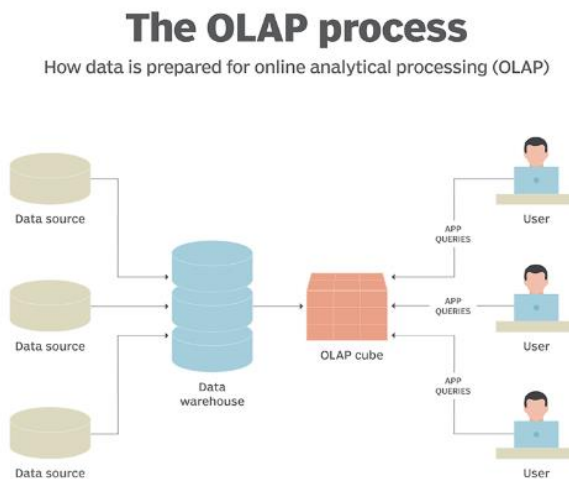


Ilustración 3. OLAP (Biscobing, 2024)

3.2.2.2. Tipos de datos y operaciones

OLAP trabaja datos históricos y consolidados, es decir agregados o resumidos. Las operaciones de agregación, análisis multidimensional y minería de datos permiten consultas complejas, para obtener patrones y tendencias que permitan la toma de decisiones.

3.2.2.3. Diseño y estructura

El diseño de OLAP está optimizado con una estructura desnormalizada, como los modelos estrella y copo de nieve, que facilitan el análisis multidimensional a través de cubos de datos. La Ilustración 4 muestra las diferencias entre OLTP y OLAP.

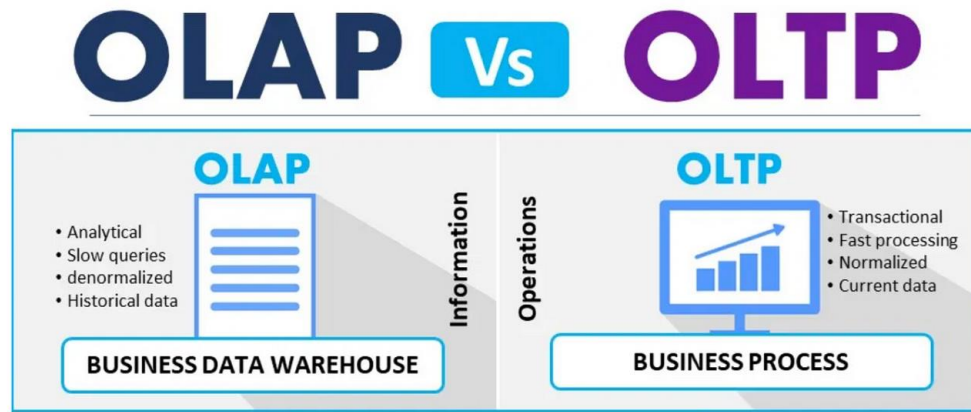


Ilustración 4. OLAP vs OLTP (Jumelle, 2022)

3.2.3. ETL Pipeline

3.2.3.1. Definición

ETL es el proceso lineal de extraer, transformar y cargar datos (Extract, Transform, Load). El proceso ETL es mover datos de un sistema de origen a un sistema de destino, es decir un pipeline, como se describe en la Ilustración 5.

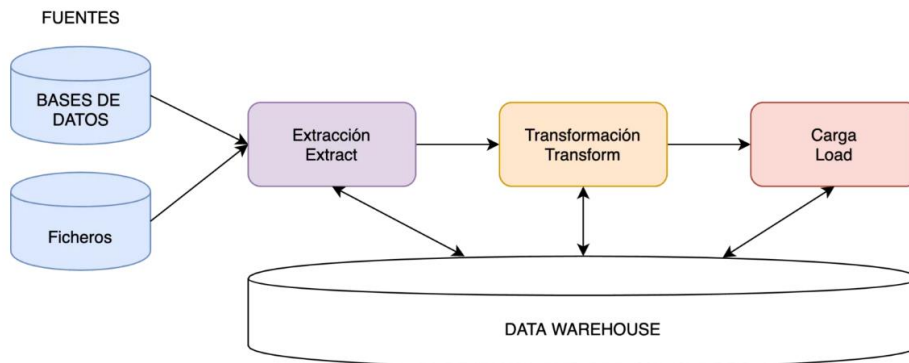


Ilustración 5. ETL Básico (Fernández, 2024)

3.2.3.2. Propósito

ETL Pipeline engloba una serie de procesos que ayudan a una empresa a recopilar datos de diferentes fuentes y moverlos a sistemas de destino (normalmente un sistema

de almacenamiento de datos, como un almacén de datos), mientras los transforman en el camino.

ETL es esencial para la preparación de datos en análisis visuales, permitiendo la combinación de fuentes de datos, la transformación de campos y la verificación de la confiabilidad de los datos (Costello & Blackshear, 2019; Prajapati & Mary, 2022). ETL es el motor que impulsa la analítica visual, permitiendo a los usuarios manipular la forma y estructura de los datos para satisfacer necesidades analíticas específicas (Costello & Blackshear, 2019).

Una empresa normalmente tiene muchas fuentes dispares de datos (que son generados y procesados en lo que se llama vulgarmente un Frankenstein de sistemas), así como varios objetivos que requieren que los datos se integren y se almacenen centralizadamente.

3.2.3.3. Etapas

ETL es un proceso complejo que requiere una planificación detallada para optimizar el rendimiento y la eficiencia. Se han desarrollado marcos genéricos y personalizables para apoyar a los diseñadores de almacenes de datos en la creación de escenarios ETL eficientes (Vassiliadis et al., 2005). La optimización lógica de los procesos ETL se puede modelar como un problema de búsqueda en el espacio de estados, utilizando algoritmos heurísticos para minimizar los costos de ejecución (Simitsis et al., 2005). Las etapas generales se describen a continuación sobre la propuesta de (Richman, 2024)

- 3.2.3.3.1. Extracción: o también llamada ingesta de datos, es la recolección de datos desde múltiples fuentes, como pueden ser: diversas bases de datos relacionales o no, archivos, APIs, websites, datalakes, aplicaciones SaaS, bases de datos transaccionales. La extracción puede ser automatizada a través de una herramienta ETL de un proveedor, o desarrollada “inhouse” por los técnicos que conocen los sistemas dispares de datos. Los pipelines pueden extraer datos de origen en tiempo real (del sistema OLTP), o

pueden utilizar un flujo de trabajo de procesamiento por lotes, es decir en un intervalo de tiempo establecido.

3.2.3.3.2. Transformación: según los objetivos, esta etapa incluye los subprocesos de filtrado de datos, agregación, limpieza, extracción de características, remodelación, normalización, enriquecimiento y estructuración de los datos. Estas transformaciones garantizan que los datos lleguen al almacenamiento o a otra aplicación limpios, consistentes y sin errores. Estas transformaciones son simples, mientras que las transformaciones de datos exploratorias más complejas para análisis profundos ocurren más adelante en el ciclo de vida de los datos.

3.2.3.3.3. Carga: es cargar los datos en el sistema de destino. El destino puede ser: un almacén de datos (data warehouse), aplicaciones SaaS, visualizaciones y paneles de control. Desde el destino pueden tener lugar todo tipo de flujos de trabajo analíticos: inteligencia empresarial, aprendizaje automático, análisis descriptivo o predictivo, entre otros. El pipeline (o canalización) de la carga debe integrarse con el sistema de destino para cargar los datos, donde se puede notar la latencia del flujo de datos (la cantidad de tiempo que tardan los datos ETL en moverse desde la fuente hasta el destino). Según el pipeline, la latencia puede ser de horas, minutos o, si se utiliza transmisión de datos en tiempo real, en milisegundos.

3.2.4. Data Science Pipeline

3.2.4.1. Definición

Es un flujo de trabajo sistemático que abarca todas las etapas de un proyecto de ciencia de datos, desde la recolección de datos hasta la implementación y mantenimiento de modelos predictivos o analíticos. Para una mejor comprensión revisar el video de IBM: <https://youtu.be/6kEGUCrBEU0?si=dlkTK7-n2cDcJdzM>

3.2.4.2. Propósito

El mundo actual es impulsado por los datos. Las empresas de cualquier tamaño generan y recogen grandes cantidades de datos de varias fuentes. Los datos son fundamentales para tomar decisiones críticas y encaminar la innovación. Casi siempre los datos crudos están desordenados, no están estructurados y almacenados en diferentes formatos a lo largo de múltiples sistemas. El proceso de Data Pipeline automatiza el proceso de recoger los datos de diferentes fuentes, transformarlos mediante varias estrategias, para finalmente entregarlos con un valor agregado para que sean útiles al destinatario.

3.2.4.3. Etapas

En la literatura se pueden encontrar diversas adaptaciones de los pasos generales, dependiendo de las herramientas utilizadas y los objetivos finales. Una versión que integra varias herramientas y metodologías es la presentada por Sahn Lam de la empresa Bytebytego (Lam, 2024), que ilustra la Imagen 6. Se considera que esta es una versión extensiva, que puede simplificarse o intercambiarse los pasos. A continuación, se presenta un

resumen y adaptación del datapipeline propuesto por Lam.

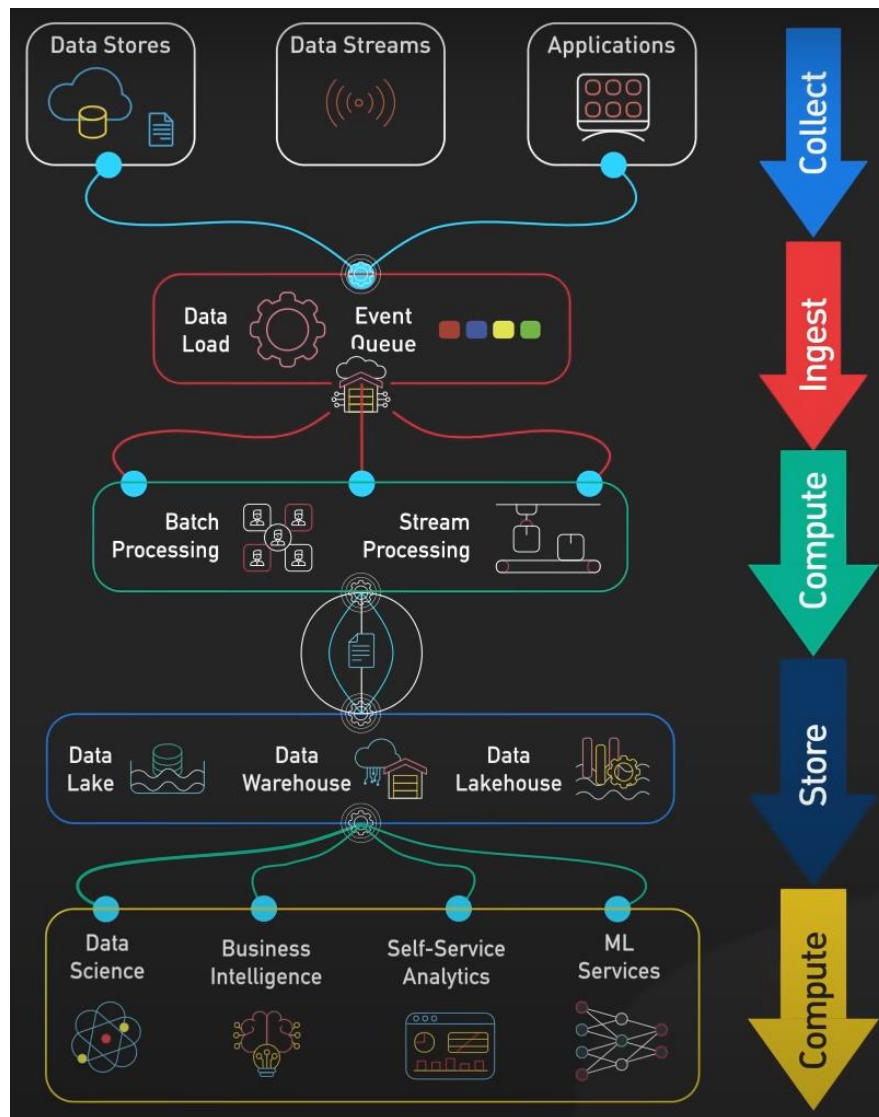


Ilustración 6. Data Pipeline (Lam, 2024)

- 3.2.4.3.1. Recolectar “Collect” datos almacenados en múltiples fuentes como Bases de datos, APIs, web scrapping, Data Stores, Data Streams, o Aplicaciones varias.
- 3.2.4.3.2. Ingestar los datos en un sistema de tubería sea a sistemas de carga de datos, o a una cola intermedia de eventos.
- 3.2.4.3.3. Computar los datos sea por medio de procesamiento por lotes o procesamiento de flujo en línea. Esta es la etapa del ETL.

3.2.4.3.3.1. El procesamiento por lotes o “Batch Processing” extrae datos de fuentes “fijas” o donde están los datos ya almacenados. Carga los datos periódicamente a un almacenamiento temporal o lagos de datos (Data Lakes). Luego los transforma y los carga en almacenes de datos (Data Warehouse). Ver Ilustración 7.

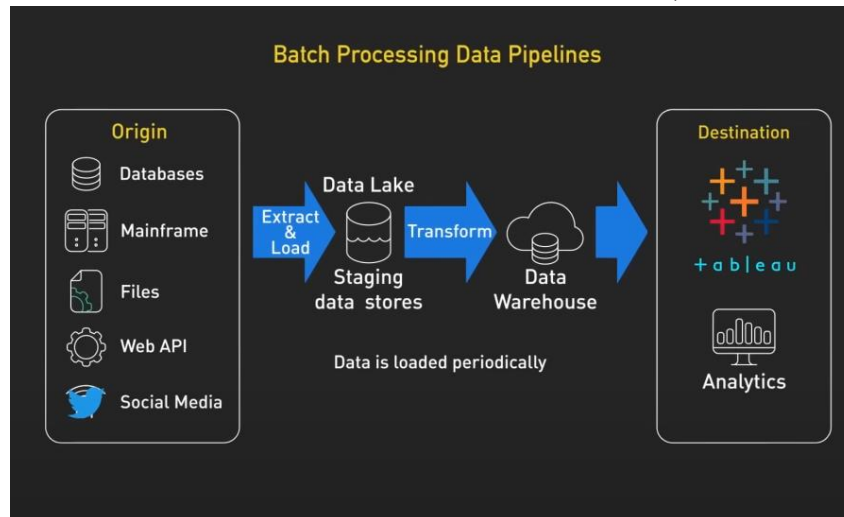


Ilustración 7. Data Pipeline en Batch (Lam, 2024)

3.2.4.3.3.2. El procesamiento en línea difiere del anterior porque extrae los datos conforme los datos son generados por las fuentes de origen, tales como redes sociales o el Internet de las Cosas. Ver Ilustración 8.

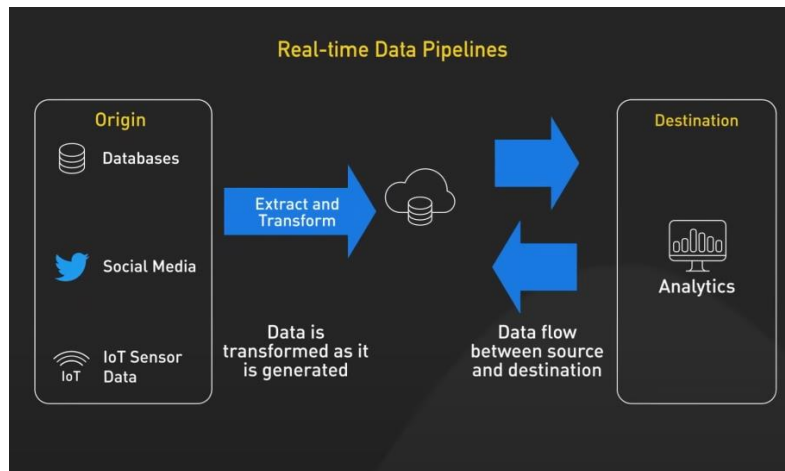


Ilustración 8. Data Pipeline en Tiempo Real (Lam, 2024)

3.2.4.3.4. Almacenar los datos luego del ETL sea en Data Lake, Data Warehouse, o Data Lakehouse, , una vez que los datos están transformados, es decir, que están organizados y tienen un valor agregado. El Data Lake se utiliza normalmente para almacenamiento masivo de datos variados y no estructurados. El Data Warehouse se utiliza para almacenar datos organizadamente que serán analizados rápidamente para reportes empresariales. El Data Lakehouse combina las características de los dos anteriores, permitiendo una analítica avanzada.

3.2.4.3.5. Computar los datos en forma avanzada consiste en utilizar la ciencia de datos, la inteligencia de negocios, la analítica de autoservicio, y el aprendizaje automático. Los pasos en que puede descomponerse esta etapa son:

- 3.2.4.3.5.1. Procesamiento y limpieza: identificación y tratamiento de datos faltantes, inconsistencias o ruido.
- 3.2.4.3.5.2. Exploración y análisis: EDA (análisis exploratorio de datos), visualización y estadística descriptiva.
- 3.2.4.3.5.3. Ingeniería de características: creación y selección de variables relevantes.
- 3.2.4.3.5.4. Modelado: selección, entrenamiento y validación de modelos de

aprendizaje automático (Machine Learning).

3.2.4.3.5.5. Evaluación y optimización mediante el uso de métricas para medir el rendimiento del modelo y ajustarlo.

3.2.4.3.5.6. Despliegue del modelo

3.2.4.3.5.7. Monitoreo y mantenimiento del modelo para detectar y corregir sesgos, o pérdida de rendimiento.

3.2.4.4. Diferencias entre OLTP, OLAP, ETL Pipeline y Data Science Pipeline

Inicialmente podemos considerar al Data Science Pipeline como un sistema OLAP integrado a un ETL Pipeline. Sin embargo, hay diferencias, que pueden ser inferidas de la información presentada anteriormente en esta clase. Para una mejor comprensión, la Tabla 1 presenta una síntesis y comparación entre estos procesos.

Tabla 1. Diferencias entre OLTP, OLAP, ETL Pipeline y Data Science Pipeline (Melgarejo, 2025)

Aspecto	OLTP	OLAP	ETL Pipeline	Data Science Pipeline
Propósito	Procesar transacciones en tiempo real.	Analizar datos históricos y generar insights.	Extraer, transformar y cargar datos para integrarlos en un almacén.	Crear modelos predictivos y analíticos.
Tipo de datos	Datos actuales y transaccionales.	Datos históricos y consolidados.	Datos operativos y estructurados.	Datos estructurados, no estructurados e históricos.
Operaciones principales	CRUD (creación, lectura, actualización, eliminación).	Consultas complejas y análisis multidimensional.	Extracción, limpieza, transformación, carga.	Exploración, modelado, evaluación, despliegue.
Frecuencia	Continua y en tiempo real.	Periódica (diaria, semanal o según necesidad).	Programada o en tiempo real.	Iterativa y dependiente del análisis.

Estructura	Bases de datos normalizadas.	Modelos desnormalizados (estrella, copo de nieve).	Procesos lineales y sistemáticos.	Procesos iterativos y exploratorios.
Escalabilidad	Escalable para múltiples transacciones simultáneas.	Escalable para análisis de datos masivos.	Altamente escalable para manejar grandes volúmenes de datos.	Escalable para modelos más complejos y grandes.
Usuarios principales	Usuarios operativos (empleados, sistemas automáticos).	Gerentes, analistas y estrategas empresariales.	Ingenieros de datos.	Científicos de datos y analistas.
Enfoque temporal	Datos en tiempo real.	Datos históricos para análisis estratégico.	Prepara datos actuales o históricos.	Modelos basados en datos históricos y actuales.
Interacción con otros sistemas	Proporciona datos para sistemas ETL o análisis.	Consume datos de ETL o transacciones históricas.	Alimenta almacenes de datos o sistemas analíticos.	Puede consumir datos de ETL o sistemas OLTP.
Finalidad	Garantizar la integridad y rapidez de transacciones.	Soporte a la toma de decisiones estratégicas.	Preparar datos para análisis o almacenamiento.	Resolver problemas específicos con machine learning o IA.

3.3. Tecnologías

En el mercado se encuentran varias tecnologías para las diferentes etapas del Data Science Pipeline. La elección de tecnologías para un pipeline de ciencia de datos depende de:

- 3.3.1. Funcionalidad y rendimiento.
- 3.3.2. Escalabilidad requerida: Apache Spark para big data, pandas para datasets pequeños.
- 3.3.3. Ecosistema y extensibilidad: disponibilidad de conectores, plugins y contribuciones de la comunidad especialmente para herramientas de código abierto. Empresas que usan Tableau, AWS, entre otras.
- 3.3.4. Casos de uso específicos: MLOps para producción o análisis exploratorio.
- 3.3.5. La Ilustración 9 presenta un ejemplo de implementación de Pipeline con tecnologías open source.

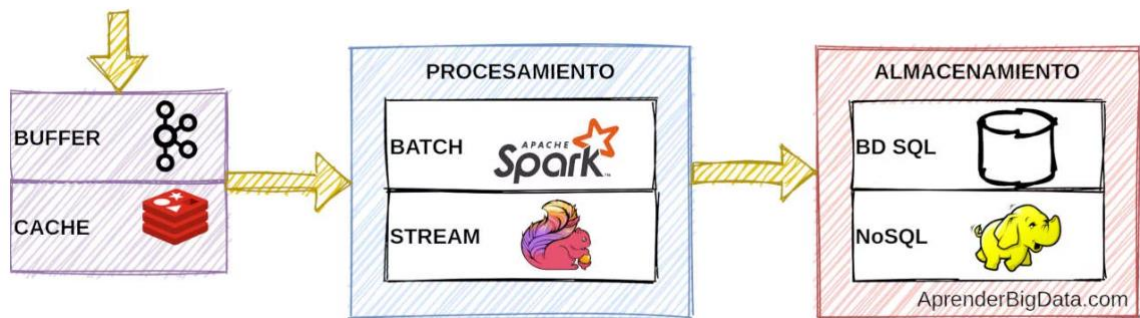


Ilustración 9. Pipeline con Spark y Hadoop (Fernández, 2024)

3.3.6. La Ilustración 10 presenta el diseño de un pipeline de datos para BigData de la empresa Amazon.

Architecture diagram of setting up the Amazon ECS cluster and CI/CD pipeline.

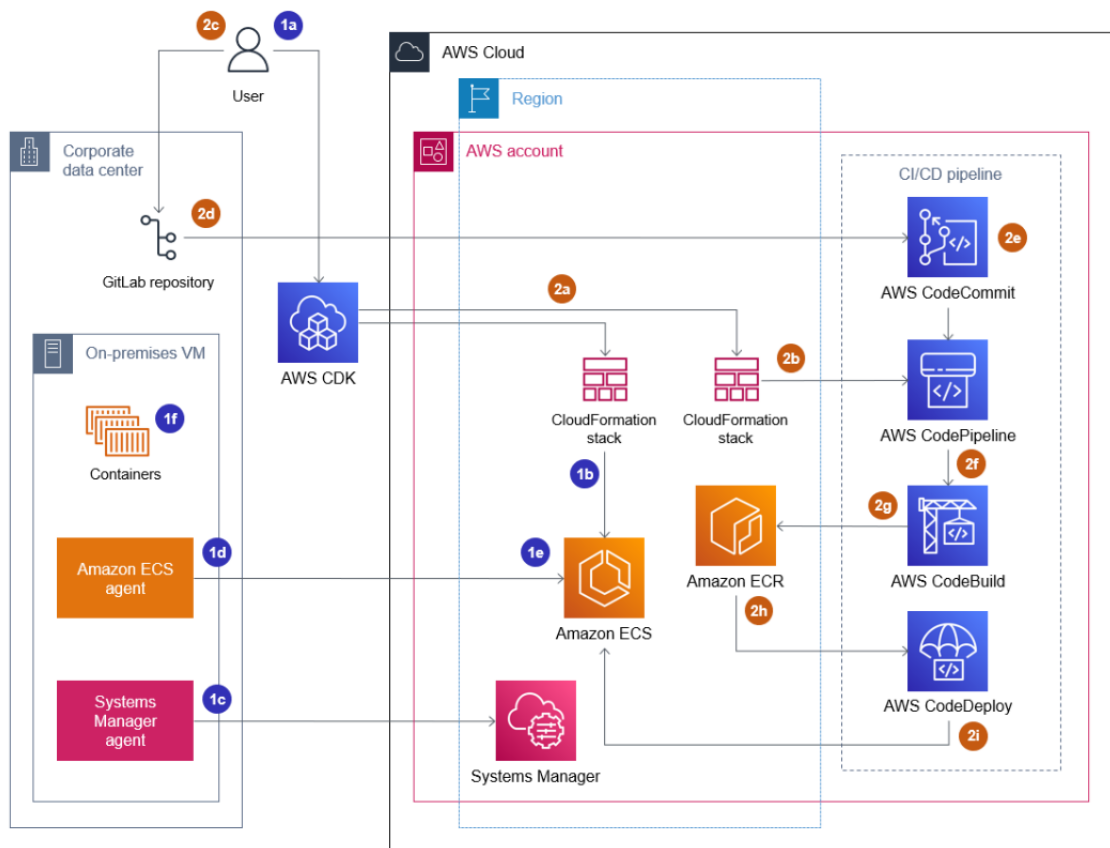


Ilustración 10. PipeLine BigData Amazon (Gaikwad, 2025)

3.3.7. La Tabla 2 presenta las tecnologías más conocidas según la etapa y los mecanismos utilizados.

Tabla 2. Tecnologías de Data Science Pipeline. Adaptado de (Lam, 2024)

Etapas	Mecanismos	Tecnologías	Uso común
--------	------------	-------------	-----------

Recolectar	Data Stores	BDD Relacionales: MySQL, PostgreSQL, SQL Server, Oracle DB, BDD NoSQL: DynamoDB, MongoDB, Cassandra, Elasticsearch, Google Cloud Spanner, Azure Cosmos DB	Almacenamiento de datos transaccionales y operativos, como registros de usuarios, pedidos de clientes, y logs de sistemas.
	Data Streams	Apache Kafka, Amazon Kinesis, RabbitMQ, Google Pub/Sub, MQTT, Azure Event Hubs, Apache Pulsar	Procesamiento de flujos de datos en tiempo real, como análisis de sensores IoT, monitoreo de logs y transmisión de eventos de aplicaciones.
	Applications	Firebase, AWS AppSync, Google App Engine, Apache Flink, Node.js APIs Web scraping: BeautifulSoup, Scrapy APIs: REST, GraphQL, Postman	Captura de datos interactivos desde aplicaciones móviles o web para análisis o almacenamiento.
Ingestar	Data Load	Apache Kafka, Amazon Kinesis, Apache NiFi, Azure Data Factory, Talend, Informatica PowerCenter, AWS Glue	Carga y transferencia de datos desde diferentes fuentes hacia sistemas de almacenamiento como data warehouses o data lakes.
	Event Queue	Apache Kafka, Amazon SQS, Azure Event Hubs, ActiveMQ, Redis Streams	Manejo de colas de eventos para coordinar y priorizar la ingestión de grandes volúmenes de datos.
ETL	Batch Processing	Apache Spark, Apache Hadoop, Apache Hive, Talend, Informatica, Airflow, Arktos, Pygrametl, Petl, Scriptella, R etl	Para extracción, transformación y carga de datos
	Stream Processing	Apache Flink, Google Cloud Dataflow, Apache Storm, Apache Samza, Azure Data Factory	
Almacenar	Datalake	Amazon S3, Hadoop, Azure Data Lake, Google Cloud Storage	Para almacenamiento de datos estructurados
	Data Warehouse	Snowflake, Amazon RedShift, Google Big Query	
	Data lakehouse	ClickHouse, Apache Druid, Snowflake, Amazon Redshift	
Analítica de datos	Data Science	NumPy, SciPy, matplotlib, pandas, IP[y]: Ipython Interactive Computing, scikit learn, PyTorch, Google Colab, R Studio, DataBricks, H2O.ai	Para modelamiento predictivo

	Business Intelligence	tableau, Power BI	Dashboards interactivos
	Self-service Analytics	looker (LookML)	Realizar consultas, sin necesidad de conocimiento técnico
	ML Services	Python, R, Jupyter, TensorFlow, scikit-learn, Keras, XGBoost, LightGBM	Aprendizaje continuo y mejoramiento, como modelos de detección de fraude

3.4. Aplicaciones

Las aplicaciones de los Pipeline de datos son cada vez más amplias y comunes, sobre todo en las empresas. A continuación, se presentan algunas aplicaciones realizadas dentro de proyectos que están debidamente documentadas:

Pipeline de datos de extremo a extremo para el desarrollo de modelos de sistemas de recomendación. Incluye almacenamiento de datos, ETL, inteligencia empresarial, entrenamiento y prueba de modelos, y aprendizaje incremental, proporcionando una estructura más organizada y eficiente para la preparación y evaluación de modelos (Zhang, 2024).

- 3.4.1. Análisis de Pipelines de ciencia de datos en diferentes escalas, desde propuestas teóricas hasta implementaciones prácticas en competencias de Kaggle y proyectos maduros en GitHub. Este análisis ayuda a entender las etapas típicas de un pipeline y cómo se conectan en diferentes contextos (Biswas et al., 2021).
- 3.4.2. Caso de estudio en un proyecto universitario de TIC muestra cómo los estudiantes diseñaron e implementaron un pipeline de datos para manufactura (Ciuciu et al., 2019).
- 3.4.3. Método adaptativo para la entrega de datos en pipelines que operan en infraestructuras edge-fog-cloud. Se probó en un estudio de caso con datos médicos y satelitales (Sánchez-Gallegos et al., 2023).
- 3.4.4. Pipelines de ciencia de datos aplicados a grafos, desde almacenes NoSQL hasta estudios completos de aprendizaje automático (Vargas-Solar et al., 2020).

Referencias citadas en la Clase 3.

- Biscobing, J. (2024). OLAP (online analytical processing). TechTarget. Search Data Management. [What is OLAP \(online analytical processing\)? | Definition from TechTarget](#)
- Biswas, S., Wardat, M., & Rajan, H. (2021). The Art and Practice of Data Science Pipelines: A Comprehensive Study of Data Science Pipelines In Theory, In-The-Small, and In-The-Large. 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), 2091-2103. <https://doi.org/10.1145/3510003.3510057>
- Carter, D. (2024). What is OLTP? Definition, Architecture, Example. GURU99. [What is OLTP? Definition, Architecture, Example](#)
- Ciuciu, I., Ene, A., & Lazar, C. (2019). An ICT Project Case Study from Education: A Technology Review for a Data Engineering Pipeline. **, 407-420. https://doi.org/10.1007/978-3-030-20485-3_32
- Costello, T., & Blackshear, L. (2019). What Is ETL?. **, 1-3. https://doi.org/10.1007/978-1-4842-5497-4_1
- Fernández, O. (2024). Aprender Big Data. <https://www.youtube.com/watch?v=OpYd7cFnHI4>
- Gaikwad, R. (2025). Set up a CI/CD pipeline for hybrid workloads on Amazon ECS Anywhere by using AWS CDK and GitLab. Amazon AWS. [Set up a CI/CD pipeline for hybrid workloads on Amazon ECS Anywhere by using AWS CDK and GitLab - AWS Prescriptive Guidance](#)
- Jumelle, M. (2022). NoSQL: définitions et exemples. Blent. [NoSQL : définitions et exemples | Blent.ai](#)
- Lam, S. (2024). What is Data Pipeline? | Why Is It So Popular?. ByteByteGo. [ByteByteGo Newsletter | Alex Xu | Substack What is Data Pipeline? | Why Is It So Popular?](#)

- Melgarejo, R. (2025). Integración, Síntesis y Adaptación de Contenido de Varias Fuentes.
- Richman, J. (2024). What is an ETL Pipeline? Examples & Tools (Guide 2024). ESTUARY. <https://estuary.dev/what-is-an-etl-pipeline/>.
- Sánchez-Gallegos, D., González-Compeán, J., Carretero, J., & Marin-Castro, H. (2023). A data science pipeline synchronisation method for edge-fog-cloud continuum. Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis. <https://doi.org/10.1145/3624062.3624284>
- Simitsis, A., Vassiliadis, P., & Sellis, T. (2005). State-space optimization of ETL workflows. IEEE Transactions on Knowledge and Data Engineering, 17, 1404-1419. <https://doi.org/10.1109/TKDE.2005.169>
- Vargas-Solar, G., Zechinelli-Martini, J., & Espinosa-Oviedo, J. (2020). Enacting Data Science Pipelines for Exploring Graphs: From Libraries to Studios. **, 271-280. https://doi.org/10.1007/978-3-030-55814-7_23
- Vassiliadis, P., Simitsis, A., Georgantas, P., Terrovitis, M., & Skiadopoulos, S. (2005). A generic and customizable framework for the design of ETL scenarios. Inf. Syst., 30, 492-525. <https://doi.org/10.1016/j.is.2004.11.002>
- Zhang, K. (2024). Incorporating Deep Learning Model Development With an End-to-End Data Pipeline. IEEE Access, 12, 127522-127531. <https://doi.org/10.1109/ACCESS.2024.3456113>

Definición de los términos citados en la Clase 1.

ETL Pipeline



ETL es el proceso lineal de extraer, transformar y cargar datos (Extract, Transform, Load). El proceso de conducción ETL (ETL Pipeline) es mover datos de un sistema de origen a un sistema de destino.

Data Science Pipeline

Es un flujo de trabajo sistemático que abarca todas las etapas de un proyecto de ciencia de datos, desde la recolección de datos hasta la implementación y mantenimiento de modelos predictivos o analíticos

Profundización Clase 3.

El Refuerzo de la clase 3 es un notebook de Google Colab con una implementación de un Data Science Pipeline para analizar datos del Titanic. El notebook en formato “.ipynb” está comprimido en el archivo “PipeLineDeDatosEjemplo.zip”, que se adjunta a la clase.



La excelencia no se improvisa

síguenos

