Problemática en la Construcción de Data Pipelines

TFG, Trabajo de Fin de Grado
Ciencia de Datos e IA

Alumna: Laura García Perrín

Universidad Politécnica de Madrid
Escuela de Ingenieros Informáticos (ETSIINF)

Julio de 2024

Índice de Contenidos

Introducción

Evolución del Enfoque Data-centric (1960 – Actualidad)

Desde la década de 1960, la gestión de datos ha evolucionado hacia un enfoque centrado en los datos, recurso clave para la toma de decisiones.

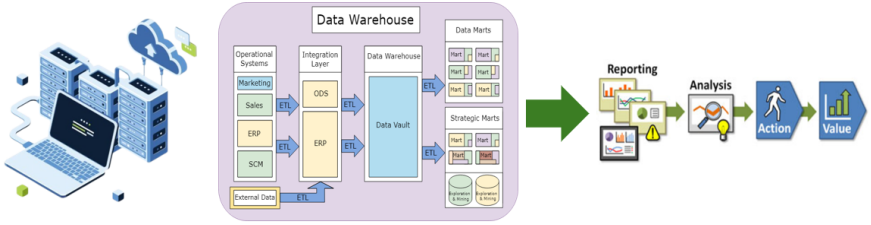


Figura 1: La cadena de valor de la analítica (Dykes, 2010). https://en.wikipedia.org/wiki/Data_mart

Planteamiento del Problema

- Creciente importancia de los datos, que se generan en cantidades masivas, a gran velocidad, en diversos formatos y desde distintas fuentes.
- Los data pipelines son la estructura clave para hacer frente a las actuales (y potenciales) adversidades que plantea este escenario.

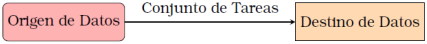


Figura 2: Ejemplo de data pipeline [Brij, 2023].

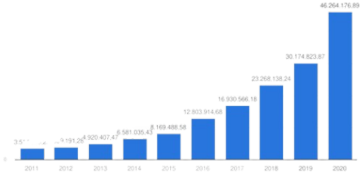


Figura 3: Tráfico de datos de los servicios de banda ancha fija en España de 2011 a 2020 (en terabytes). [CNMC Data, 2023]

Tareas y Objetivos

Objetivo principal: analizar y abordar las problemáticas asociadas con el desarrollo de data pipelines en el contexto de la Ciencia de Datos.

Objetivos específicos:

- Estudio y comprensión en el concepto de data pipelines.
- Implementación tecnológica y diseño experimental.
- Análisis crítico de los escenarios propuestos.
- Entender el alcance y las líneas futuras de los data pipelines.

Estado del Arte

Tipos de Data Pipelines

Data Pipeline

Secuencias de procesamiento de datos que incluyen la extracción, transformación y carga (ETL). Son fundamentales para la gestión de datos en tiempo real y por lotes.

Batch vs. Streaming

Dos modalidades principales de operación en data pipelines. El procesamiento por lotes maneja grandes volúmenes de datos en intervalos programados, mientras que el streaming procesa datos en tiempo real, ofreciendo respuestas más inmediatas.

Tipos de Data Pipelines

Por ejemplo, en el entrenamiento de un modelo de machine learning...

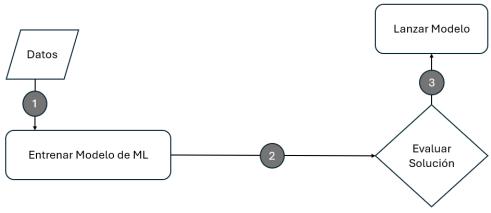


Figura 4: Entrenamiento offline (en batch)

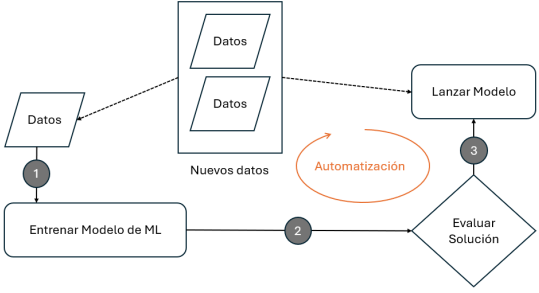


Figura 5: Entrenamiento online (streaming)

Casos Particulares de Data Pipelines

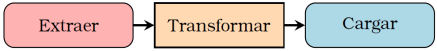


Figura 6: Data Pipeline ETL

Figura 7: Data Pipeline ELT

Caso de Estudio: Data Pipeline ETL

- ✓ Prevalencia en la Industria.
- ✓ Complejidad Técnica Representativa.
- ✓ Buen Punto de Partida: Enseñanza y Experimentación.
- ✓ Generalización a Otros Tipos de Pipelines.
- ✓ Por Requerimientos Técnicos.

Formas de Construir Data Pipelines

Modalidades

Cada modalidad atiende a una serie de necesidades y de desafíos distintos.

Figura 8: Formas de construir data pipelines.

De la misma forma, cada una plantea diferentes dificultades y problemáticas más específicas.

Problemática Común a todos los Data Pipelines

A grandes rasgos, las problemáticas comunes son:

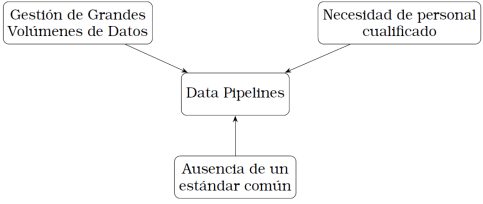


Figura 9: Problemáticas comunes a la hora de construir data pipelines.

¿Cómo es un Data Pipeline Moderno?

Data pipeline moderno

- Procesamiento de Datos Continuo y Extensible
- Acceso Democratizado a los Datos y Auto Mantenimiento
- Recursos Aislados e Independientes
- Alta Disponibilidad y Gestión ante Desastres
- Elasticidad y Agilidad en la Nube

Aspectos Clave a Tener en Cuenta

- ☐ Manejo de Grandes Volúmenes de Datos
- ☐ Dependencia del Entorno Local (si aplica)
- ☐ Escalabilidad
- ☐ Manejo de Errores y Excepciones
- ☐ Actualizaciones y Mantenimiento
- ☐ Seguridad de los Datos
- ☐ Pruebas Automatizadas

Evaluación de los Data Pipelines

- **A nivel técnico**: 24 características de Nicole Forsgren, Jez Humble y Gene Kim
  - Referencia: libro Accelerate (2018).
  - Se agrupan en 5 categorías o dimensiones.
  - Cubren varios aspectos del desarrollo y entrega de software.

Figura 10: Las 5 categorías clave para alcanzar mejoras en software

Evaluación de los Data Pipelines

- **A nivel menos técnico** una serie de preguntas tipo:

«¿Cómo de *difícil* ha sido construir el escenario?»:
«¿Cuánto tiempo me ha llevado conseguir mis propósitos?»:
«¿Qué problemas o complicaciones he encontrado?»:
«¿Me ha sido fácil aprender nuevos conceptos de software?»:
«¿Las herramientas utilizadas me han ayudado o facilitado el proyecto?»
«¿Considero útil lo aprendido?»

Desarrollo


UNIVERSIDAD POLITECNICA
MADRID
TECHNICA
IMPENDI
NATIONI

¿Cómo se procede, en líneas generales?

- Dominio: académico
- Procedimiento secuencial
- 2 casos de uso o data pipelines ETL
- Construcción, ejecución y evaluación

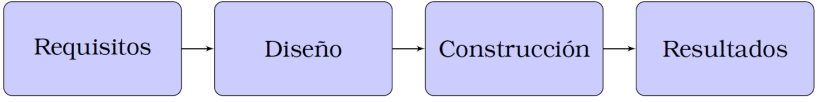


Figura 11: Fases que se seguirán en el capítulo de «Desarrollo» y, adicionalmente, «Resultados»

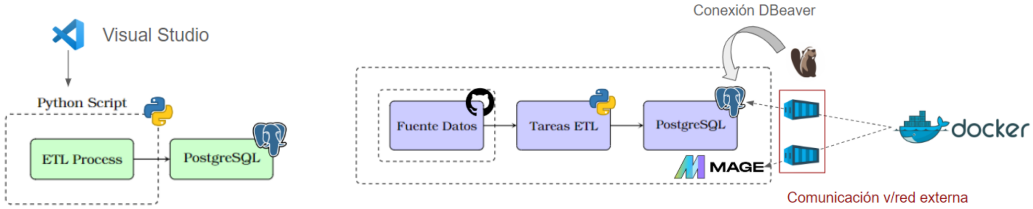
Requisitos Previos

Figura 12: Aplicaciones que serán necesarias tener instaladas.

Requisitos	**Descripción**
Conocimientos Previos	Python, YAML y Linux Shell
Selección de los datos	Datos relacionales, de carácter transaccional.
Tecnologías y Herramientas	Mage AI, PostgreSQL, Docker, DBeaver, VStudio
Escenarios	Data pipeline local y contenerizado

Cuadro 1: Resumen de los requisitos previos para el desarrollo del TFG

Diseño y Arquitectura de los Casos de Uso



(a) Implementación Local

(b) Implementación Contenerizada

Figura 13: Esquema orientativo de los escenarios que se van a implementar junto con sus herramientas y/o tecnologías correspondientes.

Etapas principales del proceso ETL

Etapa	Descripción	Tecnologías	Detalles
Extracción	Datos extraídos de una fuente externa (archivo CSV en GitHub).	Mage AI, GitHub	Extracción de datos de transacciones digitales de octubre 2023.
Transformación	Aplicación de cambios para preparar los datos para análisis.	Python, Mage AI	Eliminar columnas innecesarias, codificación *one-hot* de columnas categóricas, modificación de columnas temporales, creación de una columna derivada por agrupación, verificación de integridad y completitud de los datos transformados.
Carga	Los datos transformados se cargan en un sistema de almacenamiento para su uso posterior.	PostgreSQL, DBeaver, esquemas SQL	Escritura de datos en la base de datos PostgreSQL utilizando un esquema SQL predefinido, aseguramiento de la integridad y seguridad de los datos almacenados.

Cuadro 2: Resumen de las etapas del proceso ETL en el TFG

Conjunto de Datos

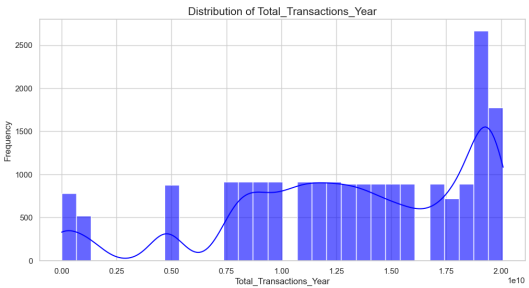
Stats NZ Tatauranga Aotearoa

19961 filas $\times 14$ columnas

Columna	Descripción
Series_reference	Identificador de la serie
Period	Periodo de tiempo de las transacciones
Data_value	Valor de las transacciones
STATUS	Estado de los datos
UNITS	Unidades de medida
Magnitude	Magnitud de los datos
Subject	Tema de la serie
Group	Grupo de datos
Series_title_1	Descripción del título de la serie

Cuadro 3: Resumen de las columnas de datos

(a) Vista general del conjunto de datos que se utiliza.



(b) Número total de transacciones por año.

Figura 14: A la izquierda, información general sobre el conjunto de datos. A la derecha, ejemplo de datos procesados para dar soporte a las acciones de *reporting* o de análisis.

Data Pipeline ETL Local

```
postgres=# \dt
                Listado de relaciones
 Esquema |        Nombre        | Tipo  |  Dueño
---------+----------------------+-------+----------
 public  | table_transactions | tabla | postgres
(1 fila)
```

Figura 15: Captura evidencia del resultado que debe aparecer en pantalla una vez se ejecuta el data pipeline ETL de forma local.

Dificultad de Construcción del Escenario:
Tiempo para Lograr Objetivos:
Problemas o Complicaciones Encontradas:
Facilidad para Aprender Nuevos Conceptos de Software:
Utilidad de las Herramientas:
Utilidad de lo Aprendido:

Dificultades Encontradas Más en Detalle

- Inicio lento y conocimiento esencial (potencial curva de aprendizaje elevada)
- Gestión y monitoreo escasamente automatizados
- Responsabilidad directa y autogestión
- Es necesario tener todo (bien) instalado

Data Pipeline ETL Contenerizado

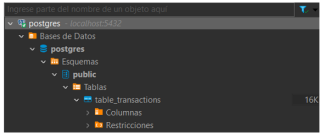


Figura 16: Árbol de Directorios que se forma en DBeaver una vez se hace la carga de datos en PostgreSQL

Dificultad de Construcción del Escenario:
Tiempo para Lograr Objetivos:
Problemas o Complicaciones Encontradas:
Facilidad para Aprender Nuevos Conceptos de Software:
Utilidad de las Herramientas
Utilidad de lo Aprendido

Dificultades Encontradas en Detalle

- Comprensión Técnica Requerida y Decisiones Críticas (docker-compose.yml vs Dockerfile)
- Problemas de Conectividad y de Orquestación de Servicios
- Interfaz, Funcionalidades y Cuellos de Botella

Resultados

Capacidad	Sí	Parcial	No
Entrega Continua			
1. Control de versiones para todos los artefactos de producción		✓	
2. Automatiza el proceso de implementación.			✓
3. Implementa la integración continua.			✓
4. Utiliza métodos de desarrollo basados en troncales.			✓
5. Implementa automatización de pruebas.			✓
6. Admite la gestión de datos de prueba.	✓		
7. Cumple con las normativas de seguridad.		✓	
8. Implementa entrega continua.		✓	
Arquitectura			
9. Usa una arquitectura débilmente acoplada.		✓	
10. Arquitecto para equipos empoderados.			✓
Producto y Procesos			
11. Recopila e implementa los comentarios de los clientes.			✓
12. Hace visible el flujo de trabajo (manual).	✓		
13. Trabaja en pequeños lotes.	✓		
14. Fomenta y permite la experimentación en equipo.	✓		
Gestión y seguimiento			
15. Procesos ligeros de aprobación de cambios adecuados al entorno local.	✓		
16. Monitorización de todas las aplicaciones e infraestructuras para tomar decistones.	✓		
17. Verifica el estado del sistema de manera proactiva.		✓	
18. Uso de límites de trabajo en proceso para gestionar el flujo de trabajo local		✓	
19. Visualiza el trabajo para monitorear la calidad y comunicarse con todo el equipo.	✓		
Cultura			
20. Apoya una cultura generativa.		✓	
21. Fomenta y apoya el aprendizaje.		✓	
22. Apoya y facilita la colaboración entre equipos.		✓	
23. Proporciona recursos y herramientas que hacen que el trabajo tenga sentido.	✓		
24. Apoya o encarna el liderazgo transformacional.		✓	

(a) Evaluación Técnica Escenario Local

Capacidad	Sí	Parcial	No
Entrega Continua			
1. Control de versiones para todos los artefactos de producción	✓		
2. Automatiza el proceso de implementación.	✓		
3. Implementa la integración continua.		✓	
4. Utiliza métodos de desarrollo basados en troncales.			✓
5. Implementa automatización de pruebas.		✓	
6. Admite la gestión de datos de prueba.	✓		
7. Cumple con las normativas de seguridad.	✓		
8. Implementa entrega continua.		✓	
Arquitectura			
9. Usa una arquitectura débilmente acoplada.	✓		
10. Arquitecto para equipos empoderados.			✓
Producto y Procesos			
11. Recopila e implementa los comentarios de los clientes.		✓	
12. Hace visible el flujo de trabajo.	✓		
13. Trabaja en pequeños lotes.	✓		
14. Fomenta y permite la experimentación en equipo.	✓		
Gestión y seguimiento			
15. Procesos ligeros de aprobación de cambios adecuados al entorno contenerizado.		✓	
16. Monitorización de todas las aplicaciones e infraestructuras para tomar decistones.	✓		
17. Verifica el estado del sistema de manera proactiva.	✓		
18. Uso de límites de trabajo en proceso para gestionar el flujo de trabajo local		✓	
19. Visualiza el trabajo para monitorear la calidad y comunicarse con todo el equipo.			✓
Cultura			
20. Apoya una cultura generativa.	✓		
21. Fomenta y apoya el aprendizaje.	✓		
22. Apoya y facilita la colaboración entre equipos.		✓	
23. Proporciona recursos y herramientas que hacen que el trabajo tenga sentido.	✓		
24. Apoya o encarna el liderazgo transformacional.		✓	

(b) Evaluación Técnica Escenario Contenedores

Análisis de Impacto

Análisis de Impacto

Impacto Personal

- ✓ Crecimiento técnico y académico
- ✓ Fomento del pensamiento crítico y de gestión de proyectos

Impacto Empresarial

- ✓ Aproximación a buenas prácticas en el manejo de datos y de infraestructuras
- ✓ Eficiencia, solidez en la toma de decisiones, seguridad de datos...

Impacto Social

- ✓ Mejora en en la gestión de datos en organizaciones no empresariales

Agenda 2030

Figura 16: ODS 9 –- Industria, Innovación e Infraestructura

Eficiencia energética

- ↓ consumo de energía en centros de datos
- ↓ emisiones de CO2

Contribución a los ODS

- ☺ Promueve la industrialización inclusiva y sostenible.

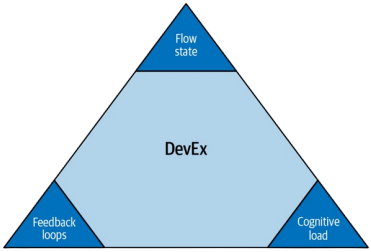
Ejemplo práctico en la industria

- Uso de contenedores Docker para monitorear y optimizar el uso de energía, escalando aplicaciones en diversas industrias.

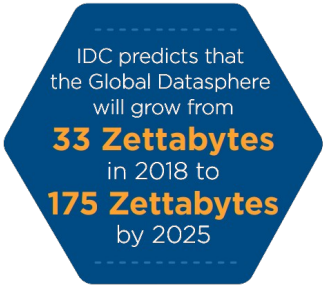
Conclusiones y Trabajo Futuro

Conclusiones

- $\exists$ varias herramientas $\rightarrow$ interesa construir data pipelines robustos.
- Hay diversas formas de construir data pipelines.
- Todavía no hay una «solución» y son necesarias buenas prácticas.



(a) La eficiencia y la escalabilidad en los sistemas ETL rata de mejorar la experiencia del desarrollador (DevEx). [Matt Palmer, 2024]



(b) Los datos generados crecerán de 16.1 zettabytes en 2016 a 163 zettabytes en 2025 [IDC, 2018].

Trabajo Futuro

- Varias tendencias emergentes.
- Influencia del *machine learning* e IA (p.e. automatizar tareas)

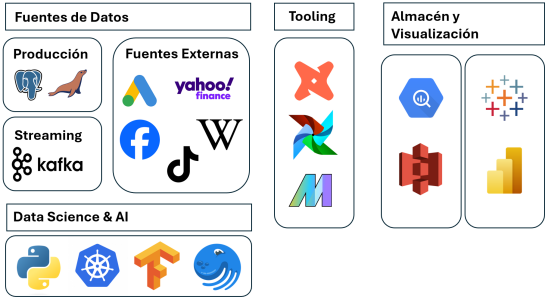


Figura 18: Arquitectura de data pipeline. [Monte Carlo, 2023]

Referencias

Referencias


Brij Kishore Pandey, Emily Ro Schoof
Building pipelines with Python (2023)
Packt Publishing Limited


Matt Palmer
Understanding ETL. Data Pipelines for Modern Data Architectures (2024)
O'Reilly


Comisión Nacional de los Mercados y la Competencia
Tráfico anual de datos de los servicios de banda ancha fija España 2011-2020 (2023)
https://es.statista.com/estadisticas/476229/trafico-anual-datos-banda-ancha-fija-espana/statisticContainer


Michael Segner, Monte Carlo Data
Data Pipeline Architecture Explained: 6 Diagrams and Best Practices (2023)
https://www.montecarlodata.com/blog-data-pipeline-architecture-explained/

Referencias

David Reinsel, John Gantz y John Rydning

The Digitization of the World From Edge to Core (2023)

https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

Laura García Perrín

Código del TFG en Github

https://github.com/lgperrin/UPM-Modules/tree/main/TFG-lgperrin

¡Gracias por su Atención!

Alumna: Laura García Perrín

Universidad Politécnica de Madrid
Escuela de Ingenieros Informáticos (ETSIINF)

Julio de 2024