netmind
a bts company

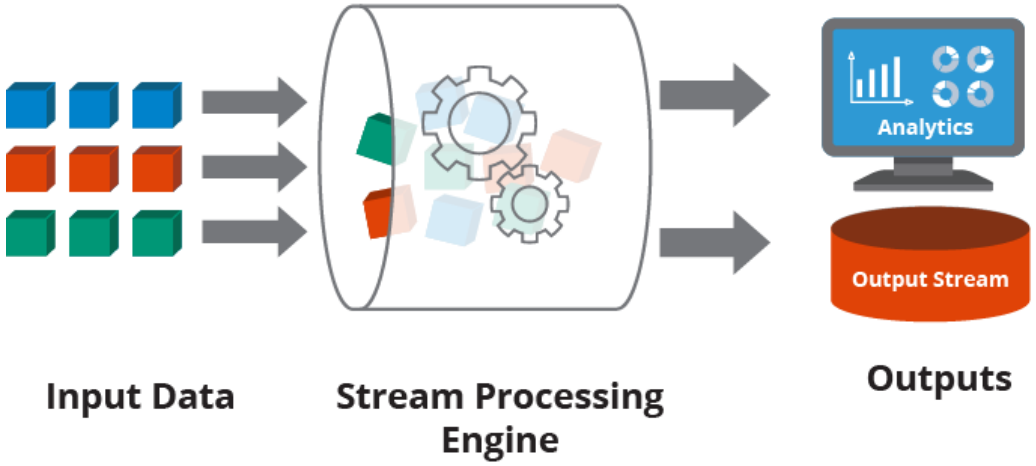
Data Science Basics

Integración de datos

01

El procesamiento de datos

Procesamiento de datos

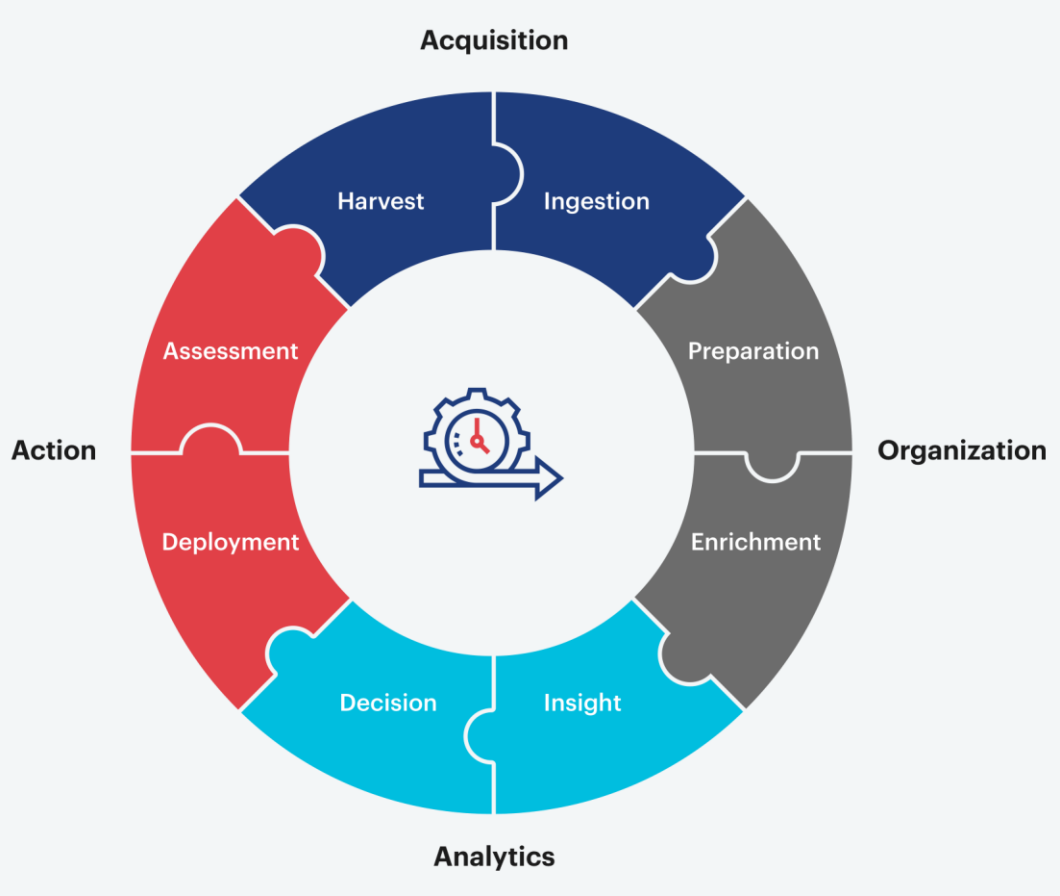


Source: hazelcast.com

El procesamiento de datos consiste en **ingerir cantidades masivas de datos en el sistema** a partir de varias fuentes diferentes, como dispositivos IoT, redes sociales, satélites, redes inalámbricas, registros de software, etc. y ejecutar lógica de negocio/algoritmos (análisis de datos) sobre ellos para extraer información significativa.

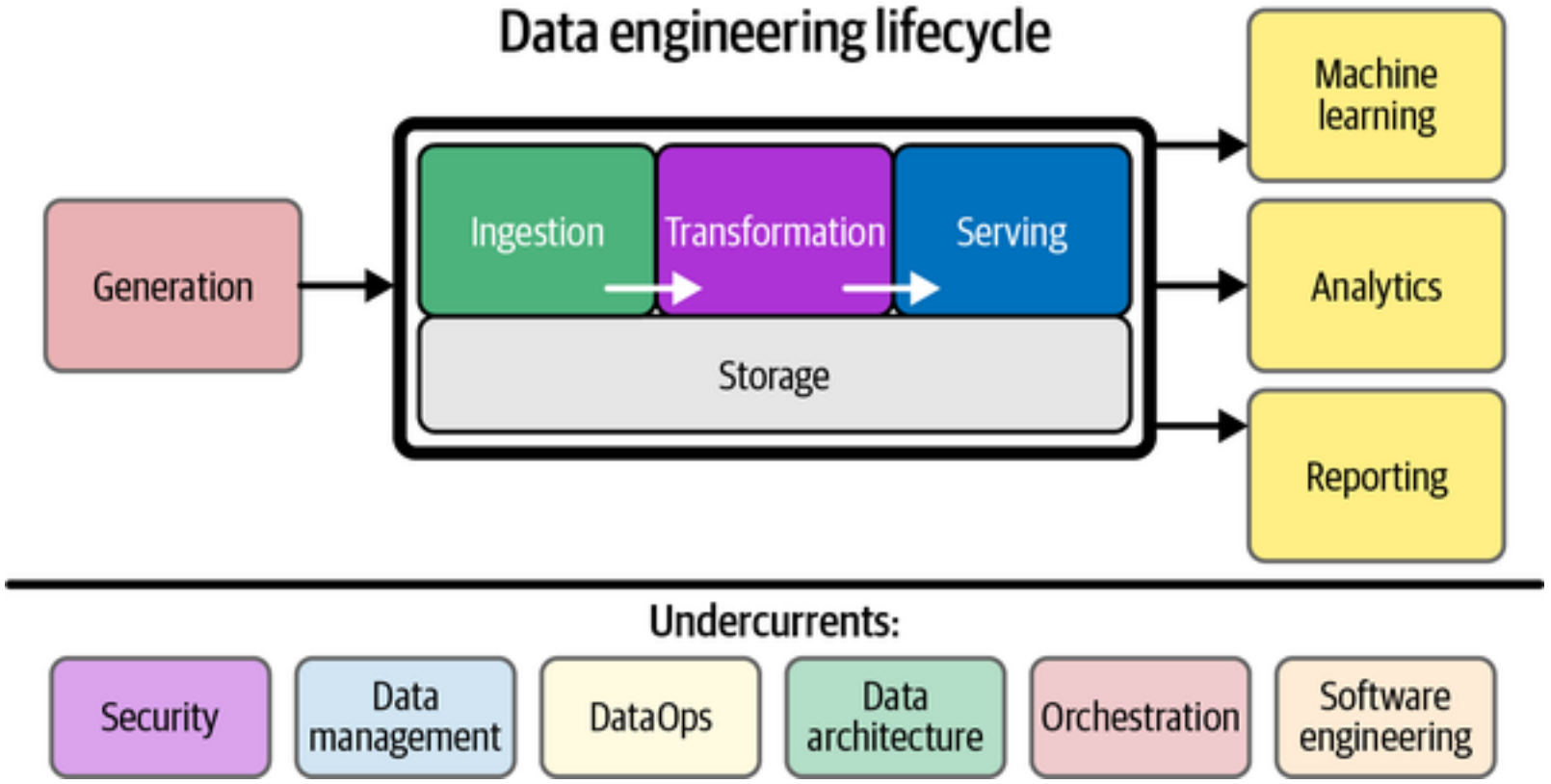
El análisis de datos ayuda a las organizaciones a utilizar la información extraída de **datos semiestructurados, no estructurados y en bruto,** a escala de **terabytes o petabytes** para crear mejores productos, comprender lo que quieren los clientes, comprender patrones de uso y, posteriormente, evolucionar el servicio o producto.

El ciclo de procesamiento de datos



Un ciclo de vida típico de ingeniería de datos se centra en adquirir los datos necesarios, organizarlos, ponerlos a disposición para su análisis y, posteriormente, derivar información o insights.

El ciclo de vida de la Ingeniería de datos



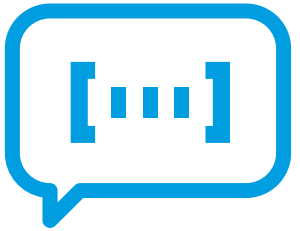
Source: Fundamentals of Data Engineering

El ciclo de vida de la ingeniería de datos comprende las etapas que convierten los ingredientes de datos sin procesar en un producto final útil, listo para el consumo por parte de analistas, científicos de datos, ingenieros de ML y otros.

Generación, Almacenamiento, Ingestión, Transformación, Servicio de datos

netmind
a bts company

02



La integración de datos

Un ejemplo de caso real…

Imaginemos que tienes un negocio de comercio electrónico y quieres crear un *modelo de propensión predictivo* para calcular el valor de vida del cliente.

Para ello, necesitarás una variedad de datos del cliente.

El problema es que la información que tiene está dispersa en diferentes sistemas aislados entre sí, incluidos

- **CRM** (Customer relationship management) con datos de clientes y ventas,
- **POS** (Point-of-sale) con historial de compras del cliente, y
- **Google Analytics** con tráfico del sitio web y datos de análisis de flujo de usuarios, por nombrar algunos.

Así, cada uno de estos sistemas contiene información relacionada con las operaciones específicas de la empresa.

Necesitas todos estos datos, algunos fragmentos de los cuales están encerrados en silos en bases de datos separadas a las que solo ciertos grupos de personas tienen acceso.

Esto se conoce como el "problema del silo de datos", lo que significa que ningún equipo o departamento tiene una vista unificada de los datos.

Y sin ellos, no se podrá construir predicciones precisas…. Desilusionante, no?

La integración de datos

La integración de datos es el proceso de **combinar datos de diferentes fuentes** en una sola vista unificada**.**

La integración comienza con el proceso de **ingesta** e incluye pasos como la **limpieza**, el **mapeo de ETL** y la **transformación**.

En última instancia, la integración de datos permite que las herramientas de análisis produzcan una **inteligencia de negocio** eficaz y procesable**.**

Patrones de diseño de integración

Fuente: striim.com

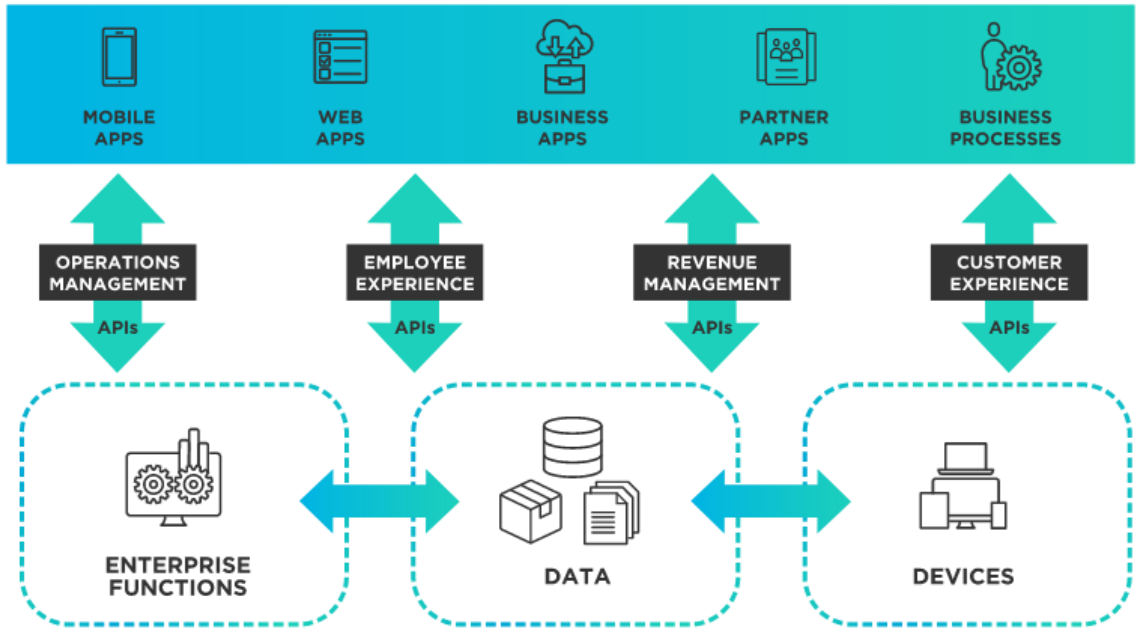
Actualmente, el **procesamiento por lotes** constituye la mayoría de los casos de uso.

Dicho esto, el **ecosistema de aplicaciones** de **micro lotes** y **streaming de datos** ha mejorado mucho y está ganando popularidad.

Casos de uso como la detección de fraudes, la gestión de inventario en tiempo real, el comercio de acciones, los juegos son algunos ejemplos.

El **auge de nuevas tecnologías** como teléfonos móviles, dispositivos IoT y plataformas Metaverse respaldadas por hardware ultrarrápido ha comenzado a producir datos más frecuentes, que a su vez deben capturarse y analizarse rápidamente.

Patrones de diseño de integración - API



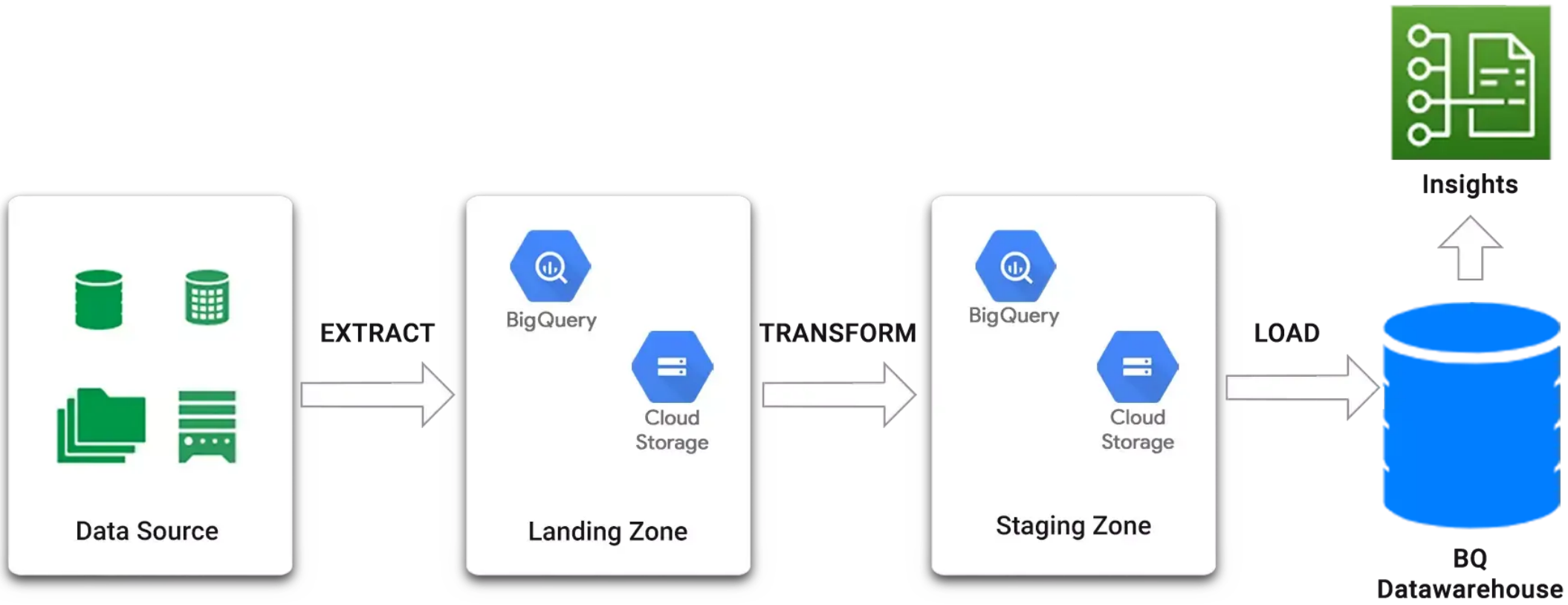
Fuente: tibco.com

La **integración basada en APIs** es el proceso de conectar datos y aplicaciones a través de Interfaces de programación de aplicaciones (API). Permite que los flujos de integración sean **definidos y reutilizados** por múltiples partes dentro y fuera de la organización.

Se está convirtiendo en una **estrategia de integración importante** debido a la creciente **complejidad de las arquitecturas de TI** con aplicaciones y fuentes de datos muy diferentes, que se alojan en las instalaciones, en la nube y mucho más.

Es base para la **interoperabilidad** en un ecosistema que abstrae las diferencias entre los activos de información. Los clientes de las API no tienen que comprender los detalles técnicos de esos activos. Este enfoque **acelera la conectividad**.

Patrones de diseño de integración - ETL

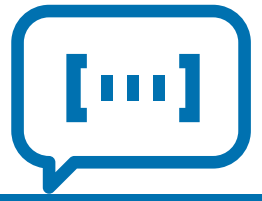


Extraer, transformar y cargar (ETL) crea tres etapas de integración de datos donde se **limpian los datos de origen** y se **transforman**, antes de ser **cargado** en el datawarehouse para su análisis avanzado.

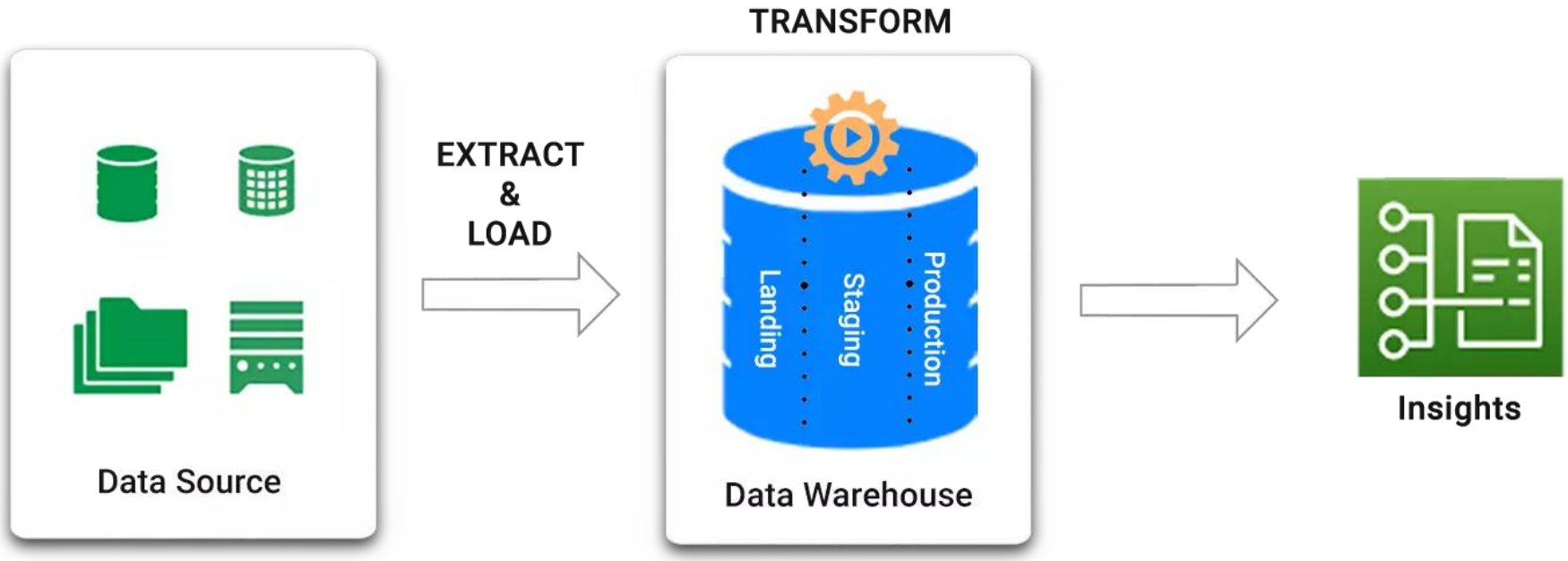
ETL Caso de uso

Problema: una empresa Fintech que ofrece hipotecas para viviendas desea conocer el valor estimado de la Vivienda.

Solución: los precios de las viviendas dependen de algunos parámetros básicos como la ubicación, el vecindario, el tamaño de la casa, la antigüedad, los impuestos, etc., y también dependen de algunos factores externos, como ventas de viviendas similares, calificación de la escuela asignada, cambios demográficos, etc.La canalización de datos utiliza datos por lotes de fuentes como el condado para el precio de la propiedad y los impuestos, MLS para ventas de viviendas similares, sistemas de clasificación de escuelas, etc. En este caso, los datos se pueden consolidar periódicamente y las estimaciones se derivan a medida que vemos nuevas actualizaciones.



Patrones de diseño de integración - ELT



Extraer, cargar, transformar (ELT) reordena la ecuación al permitir que la **plataforma de datos de destino maneje la transformación** mientras que la **plataforma de integración** simplemente **recopila y entrega** los datos.
El artículo reciente de Andreesen Horowitz sobre la infraestructura de datos moderna destacó a ELT como un **componente central de las pilas de datos de próxima generación**.

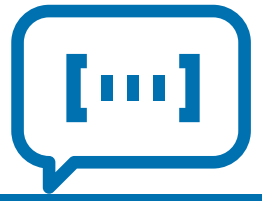
ELT Caso de uso

Problema: un minorista que intenta comprender el comportamiento del cliente para crear personalización.

Solución: la creación de una plataforma de datos de clientes implica datos de múltiples fuentes, como se indica a continuación:

- Datos transaccionales del sitio web de comercio electrónico
- Datos transaccionales de POS
- Datos transaccionales de Social
- Datos transaccionales de Mobile
- Datos analíticos
- Datos de proveedores de marketing

El proceso ELT utiliza almacenes de datos en la nube como Bigquery o Redshift que tienen una gran potencia de procesamiento y pueden manejar datos a escala de petabytes.



ELT Caso de uso

Los datos se procesarán en zonas lógicamente separadas que incluyen:

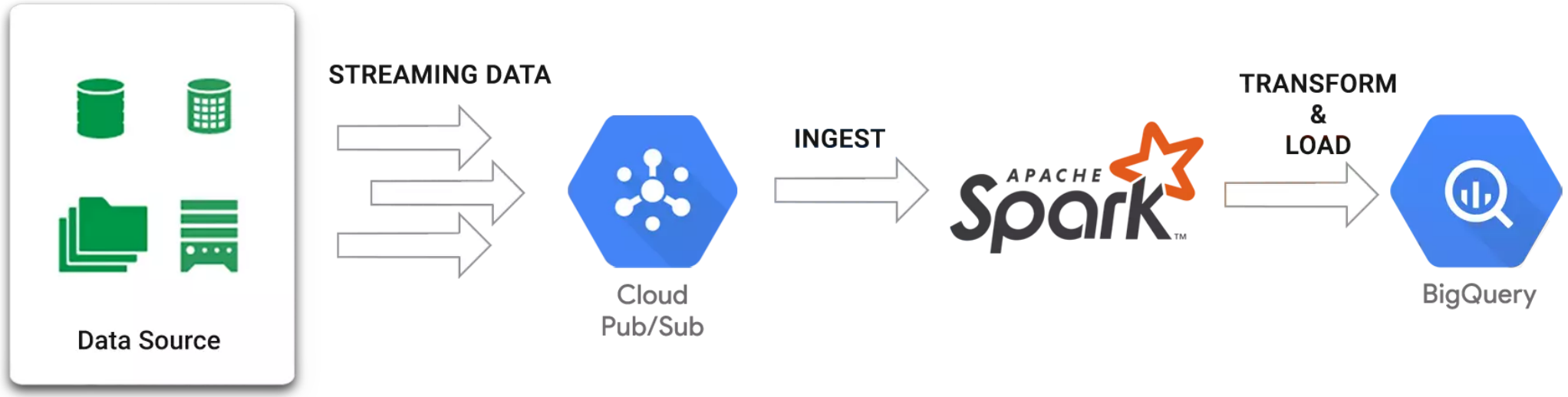
- La zona de landing que conserva los datos sin procesar de todas las fuentes.
- La zona de preparación generalmente incluye datos deduplicados y limpios a través de transformaciones.
- La zona de producción solo contendrá el subconjunto de datos de la zona de preparación que se utiliza para analizar un caso de uso particular. En este ejemplo, los datos del viaje del cliente revelan cómo un cliente se habría movido de un dispositivo móvil a una computadora de escritorio antes de realizar una compra, o cómo encontraría un producto en las redes sociales y entraría a una tienda para completar la compra.

Identificar el comportamiento del cliente con múltiples fuentes de datos proporciona una perspectiva integral para comprender y crear una cohorte para la personalización.

En este ejemplo, tenemos los puntos de datos correctos para crear una campaña para ofrecer descuentos por comprar en plataformas móviles o redes sociales que pueden reducir la cantidad de pasos por los que pasa un cliente para completar una transacción.

Esta campaña personalizada aumentaría la conversión y la tasa de éxito de la campaña.

Patrones de diseño de integración - Messaging Queue (Pub/Sub & Kafka)



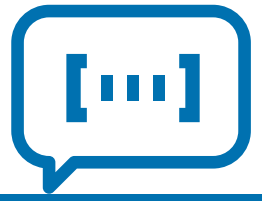
Con la popularidad de los sectores móvil, IoT y juegos, las aplicaciones comenzaron a crear **ráfagas de datos más frecuentes**. La necesidad de información en tiempo real fue respondida por **herramientas basadas en colas de mensajes** en la nube como Pub/Sub y Kafka.
Esto creó servicios de transmisión de datos como Spark, Beam, que pueden leer datos en tiempo real en las colas de mensajería, transformarlos y cargarlos en almacenes de datos en la nube.

Messaging Queue Caso de uso

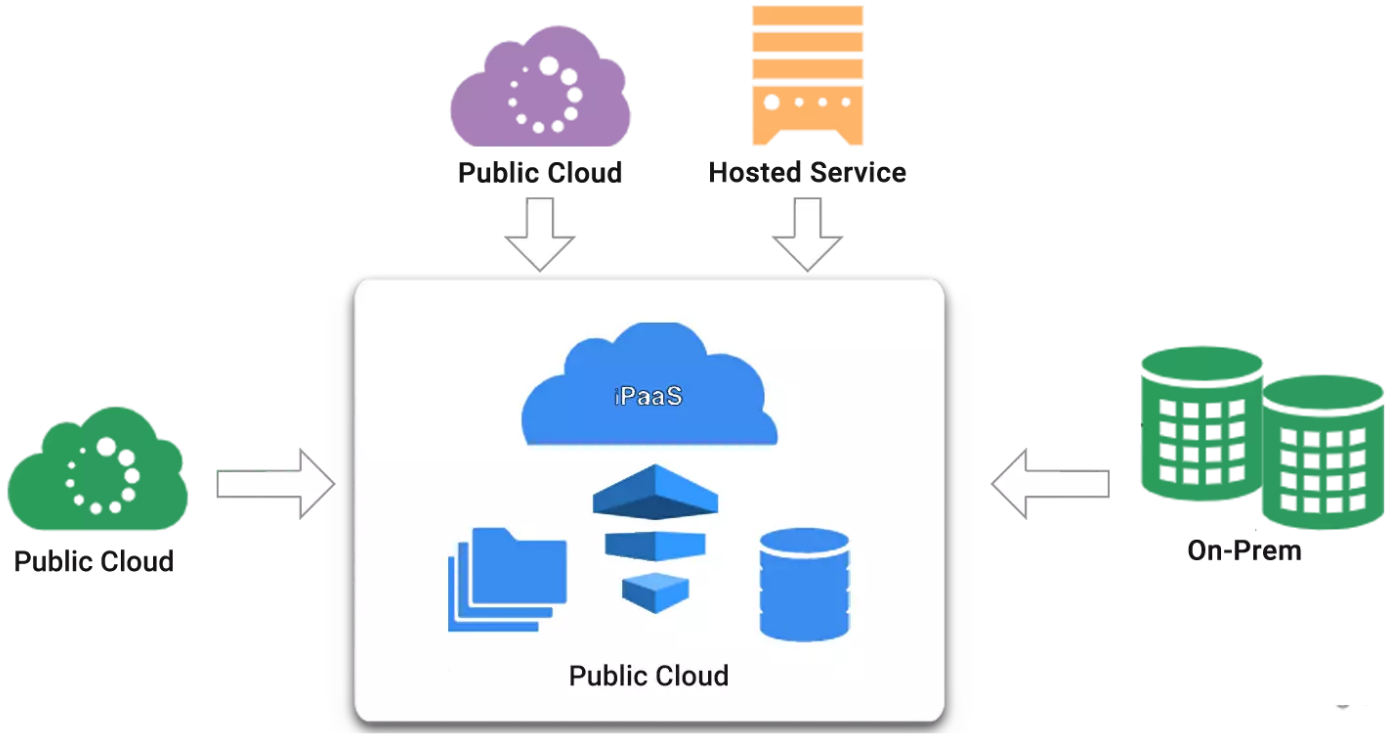
Problema: una empresa farmacéutica lanzó una nueva vacuna y le gustaría comprender su adopción e impacto.

Solución: Streaming ETL encaja bien para resolver los requisitos de análisis en tiempo real. El tiempo que se tarda en ingerir los diferentes puntos de datos es muy importante. Por lo tanto, se puede usar un pipeline de datos de transmisión basada en una cola de mensajería, que siempre está abierta para los datos entrantes.

Tan pronto como los datos de entrada se alinean, el motor de transmisión por lotes de Sparks se activa en microlotes para procesarlos, transformarlos y cargarlos en BigQuery.



Patrones de diseño de integración - iPaaS



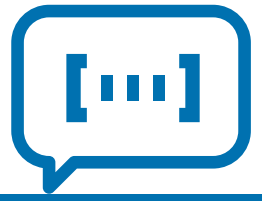
Integration Platform as a service es una plataforma de integración de próxima generación que está diseñada para la integración de aplicaciones y datos utilizando conceptos de IA/ML y Business Process Automation para proporcionar una integración perfecta entre múltiples entornos híbridos y de múltiples nubes. Proporciona una serie de servicios empaquetados para admitir aplicaciones e integraciones de datos utilizando soluciones API y sin código.

iPaaS Caso de uso

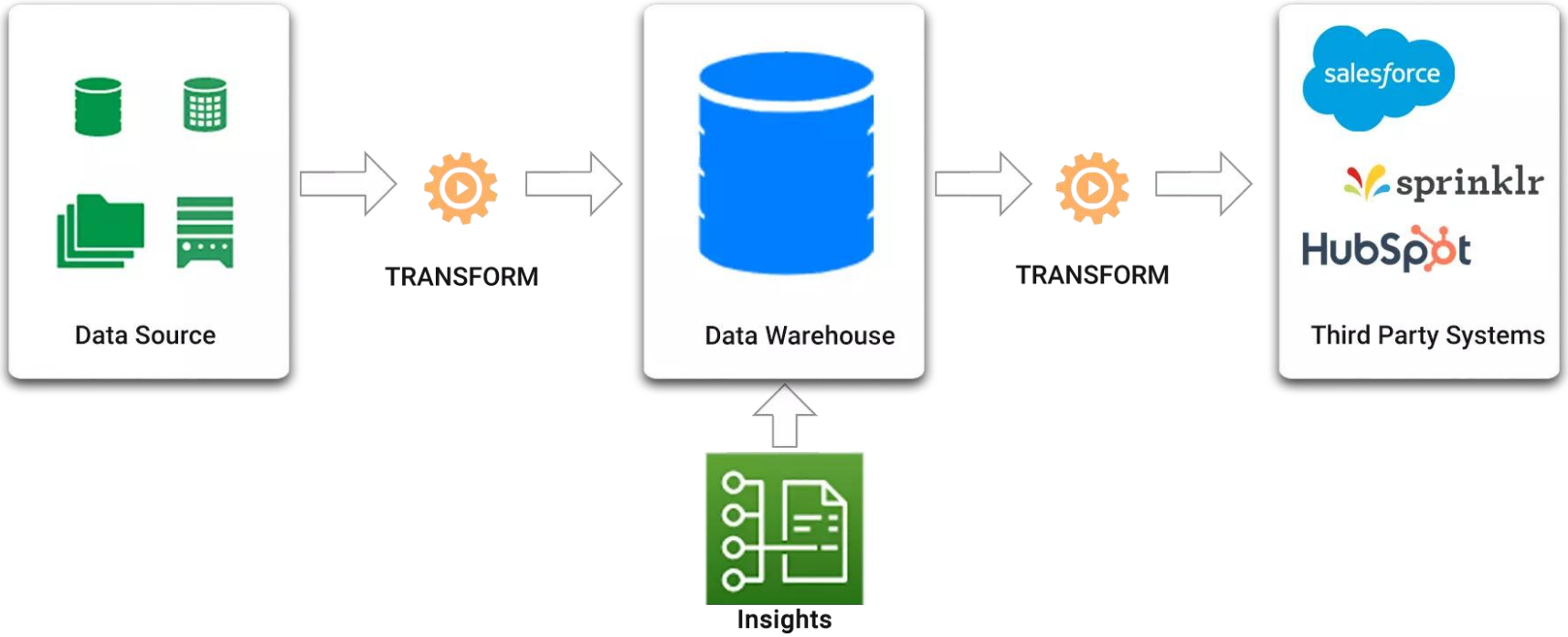
Problema: en el dominio de viajes y hospitalidad, un gigante de las aerolíneas está trabajando en una transformación digital a gran escala para cambiar su marca de una empresa de fabricación a una de tecnología.

Solución: usar la plataforma Mulesoft Anypoint para desarrollar los esfuerzos de transformación digital que pueden aportar los componentes básicos arquitectónicos necesarios para la integración de aplicaciones y datos.

Este primer enfoque de API crea una capa de aplicación débilmente acoplada, y los conectores de API basados en GUI forman una solución sin código que ayuda a conectar las API entre subsistemas en múltiples plataformas en la nube e implementaciones locales. Admite la construcción de una canalización de datos para crear una vista consolidada para la visualización.



Patrones de diseño de integración - Reverse ETL

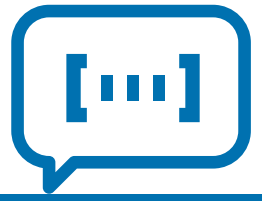


Es una tecnología emergente de transformación de datos que utiliza datos consolidados en almacenes de datos (data warehouse) para transformarlos e inyectarlos en sistemas de terceros como ERP, CDP, CRM, etc. para obtener un mejor contexto de los datos consolidados en el DW.

Reverse ETL Caso de uso

Problema: Enriquecimiento del sistema Salesforce CRM con todos los datos de la campaña de marketing para mejorar la experiencia del cliente.

Solución: más a menudo, el almacén de datos es la única fuente de verdad (SSOT) que contiene datos de múltiples sistemas de proveedores de marketing como FB Manager, Google Ads, TikTok Marketing, Tealium Audiencestream, etc. Los datos de cada fuente de datos aportan su propia dimensión. Por lo tanto, el caso de uso principal para consolidar los datos en el almacén de datos es crear la visualización de datos necesaria para los informes procesables. Estos datos también se pueden volver procesables enriqueciendo los datos en sistemas operativos como Salesforce CRM. Inducir el comportamiento del cliente rastreado por FB, TikTok, Google, etc., puede proporcionar información valiosa para construir mejores relaciones con los clientes.



03

Herramientas y servicios cloud de ingeniería y migración de datos

Herramientas de ingeniería y migración de datos

Las herramientas de ingeniería de datos son aplicaciones especializadas que hacen que la construcción de pipelines de datos y el diseño de algoritmos sean más fáciles y eficientes.

- Los sistemas de ingesta de datos como Kafka, por ejemplo, ofrecen un proceso de ingesta de datos rápido y sin inconvenientes, al mismo tiempo que permiten ubicar las fuentes de datos adecuadas, analizarlas e ingerir datos para su posterior procesamiento.
- Las herramientas soportan el proceso de transformación de datos. Esto es importante ya que el big data puede ser estructurados o no estructurados o en cualquier otro formato. Por lo tanto, se necesitan herramientas de transformación de datos para transformar y procesar big data en el formato deseado.
- Las herramientas/frameworks de bases de datos como SQL, NoSQL, etc., permiten adquirir, analizar, procesar y administrar grandes volúmenes de datos de manera simple y eficiente.
- Las herramientas de visualización como Tableau y Power BI permiten generar información valiosa y crear paneles interactivos.

Herramientas de ingeniería de datos

Herramientas de ingeniería de datos

Herramienta	Descripción	Características
Apache Spark	Apache Spark es un motor de análisis de datos de código abierto con una base de clientes de más de 52 000 organizaciones, incluidas las principales empresas como Apple, Microsoft, IBM, etc. Es una de las plataformas más rápidas para la gestión de datos y el procesamiento de transmisiones. Spark es una herramienta eficaz para la ingeniería de big data, ya que puede manejar grandes conjuntos de datos de manera eficiente y compartir tareas de procesamiento en varios dispositivos.	. Permite el procesamiento de streams en tiempo real. . Procesamiento más rápido y eficiente
Apache Hive	Apache Hive es una herramienta de gestión y almacenamiento de datos basada en Hadoop. Realiza el procesamiento de datos y la extracción de análisis utilizando un marco de trabajo similar a SQL y una interfaz de usuario.	. Gestión de la carga de trabajo . Nivel de seguridad mejorado
Apache Airflow	Con más de 8 millones de descargas al mes y 26.000 estrellas de Github, Apache Airflow es una de las herramientas de ingeniería más populares que facilita la gestión, la programación y la creación de canalizaciones de datos para los ingenieros de datos. Airflow permite una orquestación fluida de las canalizaciones de datos y, por lo tanto, es una herramienta ideal para los flujos de trabajo de ingeniería de datos. Más de 8000 organizaciones aprovechan Airflow para varios propósitos, como minimizar los silos de datos, optimizar los flujos de trabajo, etc.	. Flujos de trabajo gestionados . Extensible
Apache Kafka	Una de las herramientas de ingeniería líderes entre los profesionales de big data. Kafka es una plataforma de código abierto que ayuda a los ingenieros de datos a crear canalizaciones de datos utilizando datos de transmisión en tiempo real. Además de construir canalizaciones de datos, Kafka también permite la sincronización de datos, mensajería, transmisión de datos en tiempo real, etc.	. Actúa como intermediario . Tolerancia efectiva a fallos

Herramientas de ingeniería de datos

Herramienta	Descripción	Características
Snowflake Data Warehouse	Snowflake es un proveedor de servicios de análisis y almacenamiento de datos basado en la nube. Ayuda a los clientes a migrar rápidamente a una solución basada en la nube, y la arquitectura de datos compartidos de Snowflake la convierte en una herramienta excelente para la ciencia y la ingeniería de datos.	. Altamente escalable . Datos Semi-Estructurados
Tableau	Tableau es una de las herramientas de ingeniería de datos más antiguas y populares en la industria de big data. Tableau recopila datos de varias fuentes mediante una interfaz de arrastrar y soltar y permite a los ingenieros de datos crear tableros para la visualización. Es compatible con los ingenieros de datos en diversas actividades comerciales, como la creación de paneles en vivo y la compilación de informes de datos.	. Fácil manejo de grandes conjuntos de datos . Admite varios idiomas
Power BI	Con una cuota de mercado de BI de alrededor del 36 % desde 2021, Microsoft Power BI es una de las principales herramientas de inteligencia empresarial y visualización de datos. Los ingenieros de datos usan Power BI para generar visualizaciones dinámicas al procesar conjuntos de datos en paneles en vivo e información de análisis.	. Extremadamente asequible . Herramienta fácil de usar

Herramientas Cloud

Una de las responsabilidades más importantes de los profesionales de big data es **configurar la nube** para almacenar datos de una manera que garantice su alta disponibilidad.

Según los requisitos de almacenamiento de datos, las empresas implementan una **infraestructura de nube híbrida, pública o interna**.

AWS, Azure, GCP, etc., son algunas de las plataformas en la nube populares.

Fuente: oreilly.com

Herramientas Cloud en AWS

Herramienta	Descripción	Características
Amazon Redshift	Con más de 10 000 organizaciones de usuarios, Amazon Redshift es una solución de gestión de datos y almacenamiento de datos en la nube. Redshift es famoso por recopilar conjuntos de datos, buscar tendencias y anomalías, generar información, etc. Las funciones de compresión y procesamiento en paralelo de Amazon Redshift permiten a los usuarios administrar millones de filas a la vez, lo que reduce significativamente el tiempo de ejecución de comandos. Redshift es ideal para procesar grandes volúmenes de datos en varios almacenes de datos utilizando soluciones modernas de inteligencia empresarial.	. Procesamiento Masivamente Paralelo (MPP) . Bases de datos con columnas
Amazon Athena	AWS Athena es un servicio de consulta interactivo que le permite realizar análisis de datos en Amazon Simple Storage Service (Amazon S3) mediante el lenguaje de consulta estructurado (SQL). AWS Athena no tiene servidor, por lo que no tendrá que establecer ni mantener ninguna infraestructura. La función de escalado automático de Athena le permite realizar sus consultas en paralelo y generar resultados rápidamente, especialmente cuando trabaja con consultas complejas y grandes conjuntos de datos.	. Seguridad mejorada . Flexibilidad de alto nivel

Herramientas Cloud en Azure

Herramienta	Descripción	Características
Azure Data Factory	Azure Data Factory (ADF) is a serverless, fully managed data integration solution for gathering, processing, and modifying all of your data at scale. When shifting workloads to Microsoft Azure, Azure Data Factory is the ideal option for migrating existing ETLs (ADF). It has various use cases in almost any industry for multiple tasks, including data engineering workflows, operational data integration, works as an analytics tool, etc.	. Scalable and Efficient . Easy Migration to the Cloud
Azure Databricks	With over a $38 billion valuation, Azure Databricks is a Spark-based unified analytics engine popular among the data science and data engineering teams in almost every organization. It is a managed service that provides data analysts, data engineers, and data scientists with all of the infrastructure and support required to work with modern data analytics.	. Interactive workspace . Efficient Infrastructure

OBJETIVO

Un caso de uso con Pyspark y Databricks

INSTRUCCIONES

- Lee el artículo "Danny's Diner Case Study using Pyspark on Databricks"
  - https://www.linkedin.com/pulse/dannys-diner-case-study-using-pyspark-databricks-deepak-rajak/
- Anota tus impresiones sobre el pipeline de datos generado.
- Comenta con el resto de la clase.

20 min

A

Anexo

ETL of Data for Ice Cream Stores

An ETL process for pulling and gathering data for an Ice Cream store.

Source Data from API

We will need to hit an API to extract data about ice cream stores. We need to roll the data up to a store level and store the final result in our database.

```
[
 {
  transaction_id:"1"
  store:"10",
  date:"05/01/2020 10:05:01"
  price:100.5
 },
 {
  transaction_id:"2"
  store:"10",
  date:"05/01/2020 10:06:02"
  price:120.5
 },
]
```

ETL of Data for Ice Cream Stores

Extract Step

We are hitting the API to get the JSON response. We then flatten the JSON response using pandas and write the result to CSV.

```
import csv
import pysftp
import store_api
import pandas as pd
def get_data_from_api():
    username = os.getenv('username')
    password = os.get('password')
    response=  requests.get('https://api.github.com/user', auth=(username, password))
    json_result = request.json()
    return json_result

 def flatten_and_write_data():
    current_datetime = get_current_datetime()
    flatten_data = pd.json_normalize(data)
    flatten_data.to_csv("source_file_{}".format(current_datetime))
```

ETL of Data for Ice Cream Stores

Transform Step

In our final table, we need the total sales per store. In order to do this, we will need to perform an aggregation step. We do this by creating a python script and using the Pandas library. Our script will do the following-

1.Converting datetime to date

2. Calculating the total price by date and store

3. Finally, we want to save this result as flattened format, in this case a CSV file.

ETL of Data for Ice Cream Stores

Transform Step

```
import pandas as pd
def agg_data()
   latest_file = get_latest_file()
   data['date'] = data['datetime'].dt.date
   select_col=['store','date','price']
   new_df = data[select_col].groupby([data["store"],data["date"]]).sum()
   new_df.rename(columns={"price":"revenue"},inplace=True)
   agg_df.to_csv("stage_file_{}".format(current_date))
```

Intermediate CSV data

```
+----------------+-------+---------------------+-------+
| transaction_id | store |      datetime       | price |
+----------------+-------+---------------------+-------+
|              1 |    10 | 05/01/2020 10:05:01 | 100.5 |
|              2 |    10 | 05/01/2020 10:06:02 | 120.5 |
+----------------+-------+---------------------+-------+
```

ETL of Data for Ice Cream Stores

Load Step We want to load the data into the schema final with table name **ice_cream_store_rev**

```
from sqlalchemy import create_engine
def load_data():
   username = os.getenv('username')
   password = os.get('password')
   database = "target_database"
   engine = create_engine('postgresql://{0}:{1}@localhost:5432/{2}'.format(username,password,database))
   df = pd.read_csv("source_file_{}")
   df.to_sql('ice_cream_store_rev',schema="final", engine)
```

Final Data

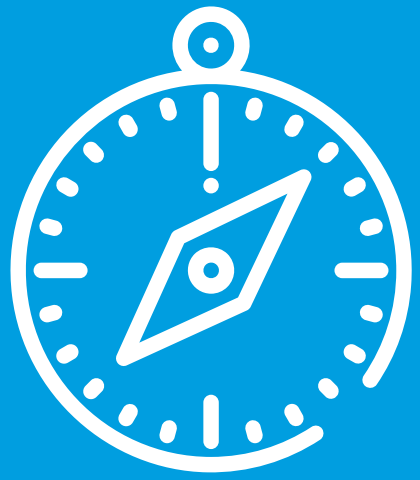
```
+-------+------------+---------+
| store |    date    | revenue |
+-------+------------+---------+
|    10 | 05/10/2020 |  220.10 |
+-------+------------+---------+
```

ETL of Data for Ice Cream Stores

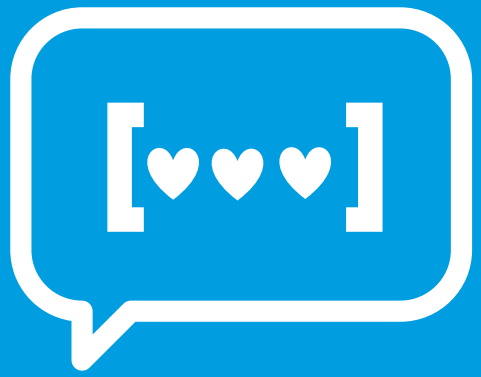
An ELT process for pulling and gathering data for an Ice Cream store.

ETL Steps

```
## Pull Data from API
data = get_data_from_api()

## Output Data to CSV
write_data(data)

## Transformation to Aggregate Data
agg_data()

## Load Data to DB
load_data()
```



Next steps



We would like to know your opinion!

Please, let us know what you think about the content.
From Netmind we want to say thank you, we appreciate time and effort you have taking in answering all of that is important in order to improve our training plans so that you will always be satisfied with having chosen us
quality@netmind.es

netmind
a bts company

Thanks!

Follow us: