POSGRADOS
Facultad de Ingeniería y Diseño

UNIVERSIDAD ICESI | A OTRO NIVEL

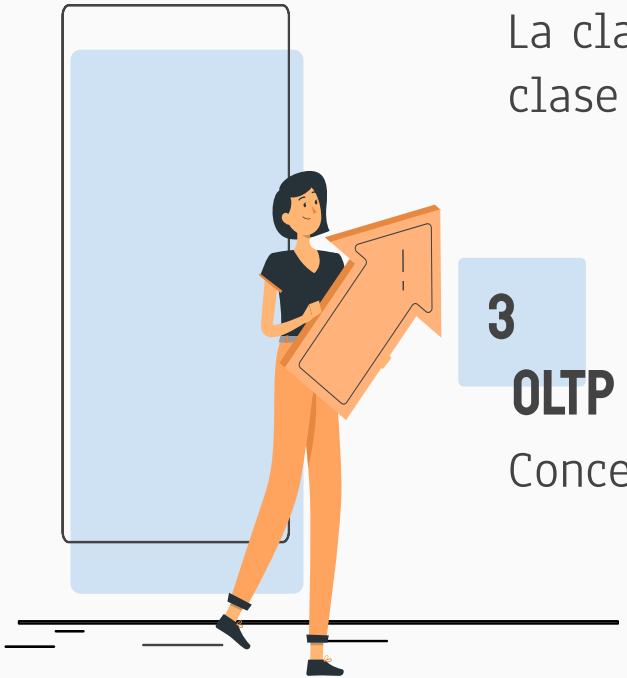
Maestría en Ciencia de Datos

Facultad de Ingeniería, Diseño y Ciencias

INFRAESTRUCTURA Y ARQUITECTURA DE TI

Ángela Villota Gómez
apvillota@icesi.edu.co

UNIVERSIDAD ICESI



1
INTRODUCCIÓN
La clase anterior/esta clase

2
PATRONES DE ARQUITECTURA
Conceptos, tecnologías y herramientas

3
OLTP - OLAP
Concepts

4
OLTP -PRÁCTICA
Una base de datos relacional

INTRODUCCIÓN

LA SEMANA PASADA

Temas

1. Introducción
2. Los tipos de datos y el proceso de ciencia de datos

Tareas:

- **Asignación:** Lectura [link] Tablero [Miró]
- **Ejercicio de limpieza de datos**

EJERCICIO

En la carpeta compartida del curso encontrará la carpeta datos que contiene 3 archivos: data.txt, data1.txt y data3.txt

1. Construya un solo archivo consolidando los datos que provienen de las 3 fuentes

Tenga en cuenta:

⟶ Los datos pueden tener alguno de los siguientes errores:

- Errores de digitación
- Espacios en blanco redundantes
- Valores imposibles
- Outliers
- Valores faltantes
- Diferentes unidades de medida
- Diferentes niveles de agregación

⟶ Es necesario combinar los datos usando un joining y appending

2. Proponga dos o más columnas dummies a partir de los datos obtenidos en el punto anterior.
3. Si tuviera que ejecutar las tareas anteriores con la ayuda de una herramienta, ¿qué herramienta usaría y por qué?

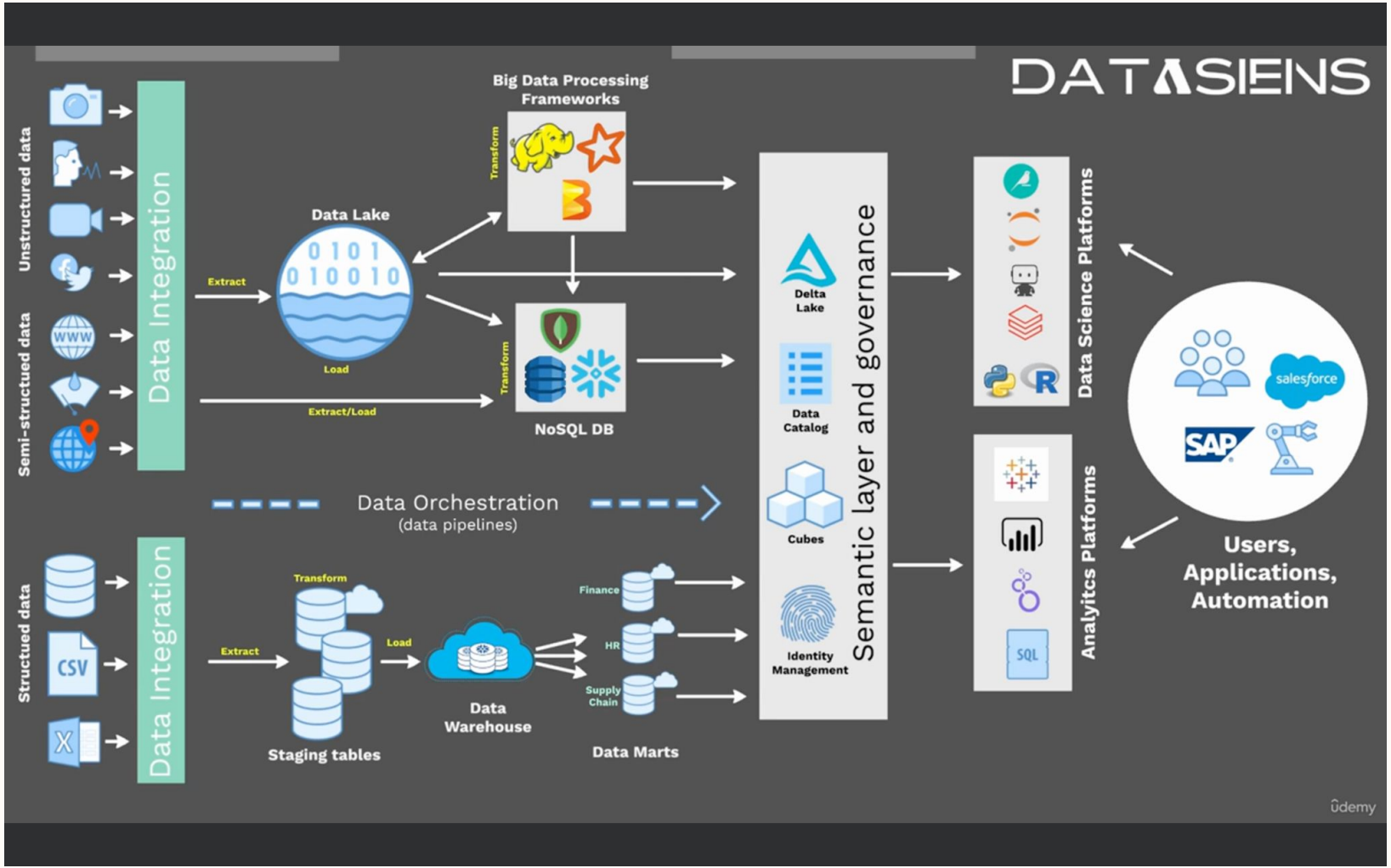
RECURSOS DE ESTA CLASE

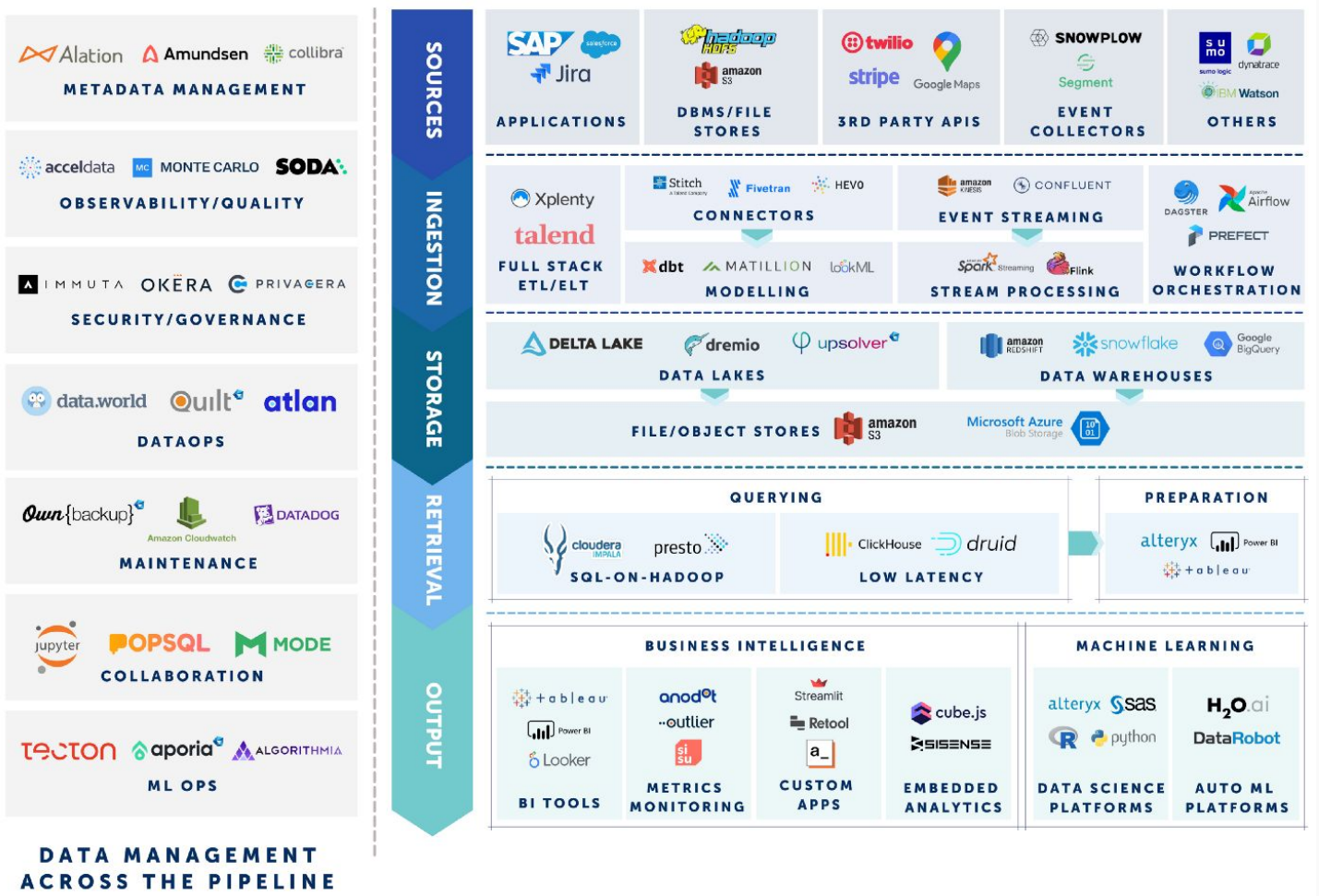
En la carpeta de la Semana 2

- Lecturas:
  - Lectura: From Data Warehouses and Lakes to Data Mesh: A Guide to Enterprise Data Architecture [link]
  - CH 32, 33, 34 Connolly-Begg (lectura complementaria)
- Slides
- Code
  - Instrucciones
  - Modelo

PATRONES DE ARQUITECTURA GESTIÓN DE DATOS


IIOT
BIG DATA
ThingsBoard


DATASIENS
Unstructured data
Semi-structued data
Structued data
Data Integration
Data Integration
Big Data Processing Frameworks
Transform
Data Lake
Extract
Load
Extract/Load
Transform
NoSQL DB
Data Orchestration
(data pipelines)
Transform
Extract
Load
Staging tables
Data Warehouse
Finance
HR
Supply Chain
Data Marts
Delta Lake
Data Catalog
Cubes
Identity Management
Semantic layer and governance
Data Science Platforms
Analyitcs Platforms
SQL
salesforce
SAP
Users, Applications, Automation
udemy



https://medium.com/vertexventures/trends-in-the-modern-data-stack-looking-ahead-c4ff8b08f0f1[1]

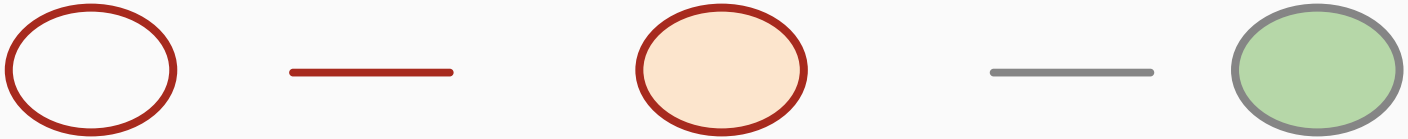
PATRONES DE ARQUITECTURA

- Arquitectura de procesamiento por Lotes (Batch Processing)
- Arquitectura de procesamiento en Tiempo Real (Real-Time Processing)
  - Arquitectura Kafka
  - Arquitectura Lambda
  - Arquitectura Kappa
  - Arquitectura Delta ¿? Nuevas?

ARQUITECTURA DE PROCESAMIENTO POR LOTES (BATCH PROCESSING)

- Esta arquitectura es utilizada para procesar grandes cantidades de datos en lotes, lo que implica que los datos son procesados en bloques y no en tiempo real.
- La arquitectura se compone de tres capas: la capa de datos, la capa de procesamiento y la capa de presentación.

ARQUITECTURA DE PROCESAMIENTO POR LOTES (BATCH PROCESSING)



Capa de datos

Responsable de almacenar los datos en un sistema de almacenamiento, como una base de datos o un sistema de archivos distribuidos. Los datos pueden ser de diferentes fuentes, como bases de datos transaccionales, sistemas de archivos, dispositivos de IoT, entre otros.

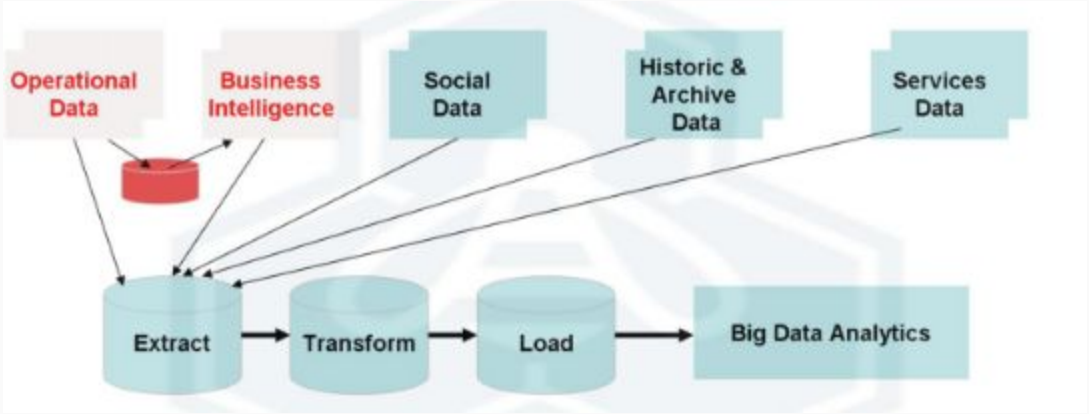
Capa de procesamiento

Responsable de procesar los datos en bloques o lotes. En esta capa, se utilizan tecnologías de procesamiento de datos como Apache Hadoop, Apache Spark, Apache Hive, entre otros, para procesar los datos y extraer información valiosa. En este proceso, los datos se dividen en lotes y se ejecutan en paralelo, lo que permite procesar grandes cantidades de datos de manera eficiente.

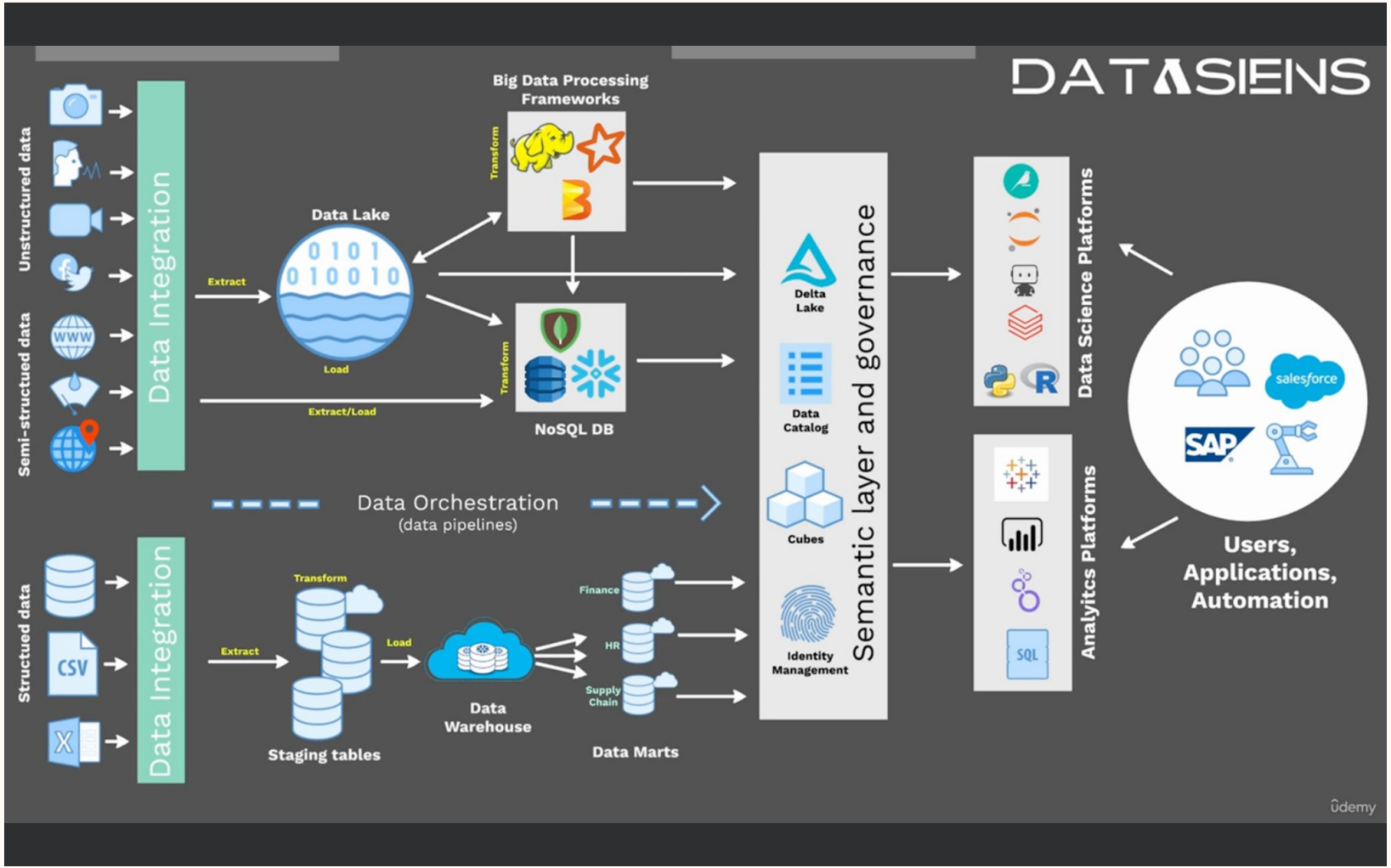
Capa de Presentación

Responsable de proporcionar los resultados del procesamiento de los datos a los usuarios finales. Los resultados pueden ser presentados a través de informes, cuadros de mando, gráficos y otras herramientas de visualización de datos.

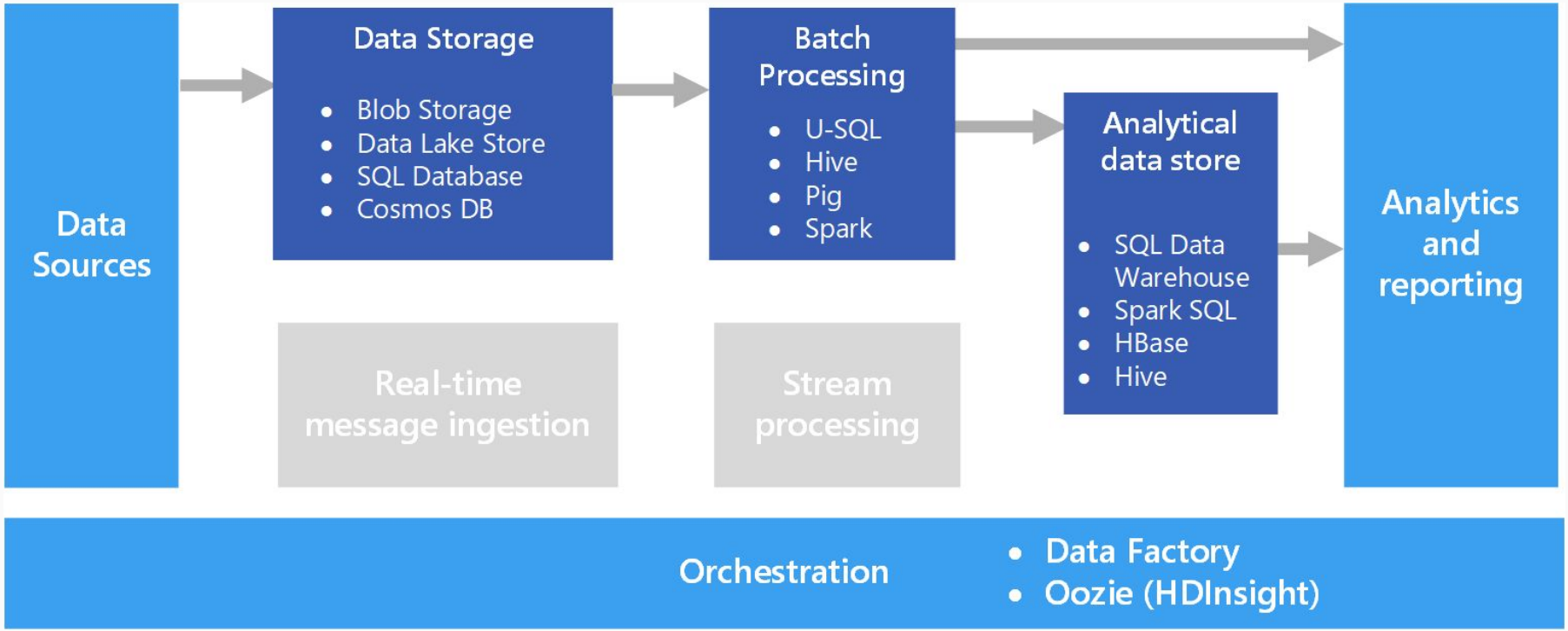
ARQUITECTURA GENERAL



https://blogs.deusto.es/bigdata/paradigma-batch-para-sistemas-big-data-i/

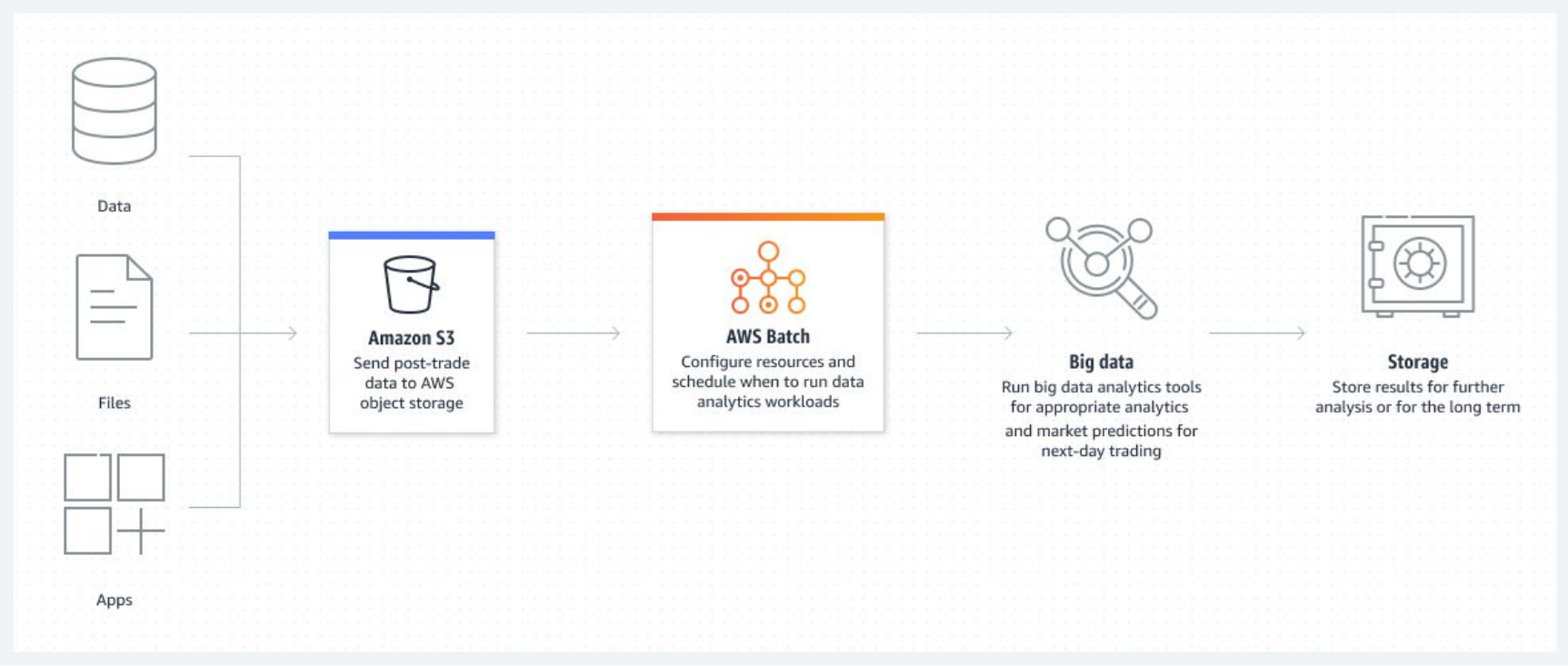

DATASIENS
Unstructured data
Semi-structued data
Structued data
Data Integration
Data Integration
Big Data Processing Frameworks
Transform
Data Lake
Extract
Load
Extract/Load
Transform
NoSQL DB
Data Orchestration
(data pipelines)
Transform
Extract
Load
Staging tables
Data Warehouse
Finance
HR
Supply Chain
Data Marts
Delta Lake
Data Catalog
Cubes
Identity Management
Semantic layer and governance
Data Science Platforms
Analyitcs Platforms
SQL
salesforce
SAP
Users, Applications, Automation
udemy

SOLUCIÓN AZURE



https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/batch-processing

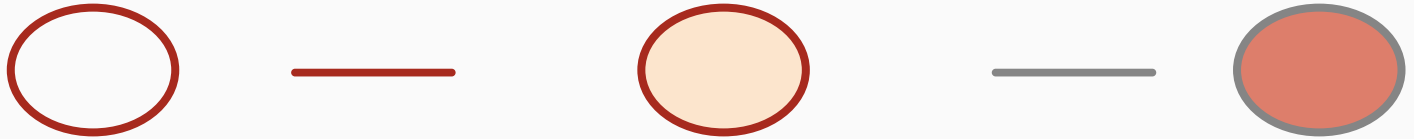
SOLUCIÓN AWS



ARQUITECTURA DE PROCESAMIENTO EN TIEMPO REAL (REAL-TIME PROCESSING)

- Esta arquitectura es utilizada para procesar datos en tiempo real.
- Esta arquitectura es ideal para proyectos que requieren respuestas rápidas a las consultas.

ARQUITECTURA DE PROCESAMIENTO EN TIEMPO REAL (REAL-TIME PROCESSING)



Capa de adquisición de datos

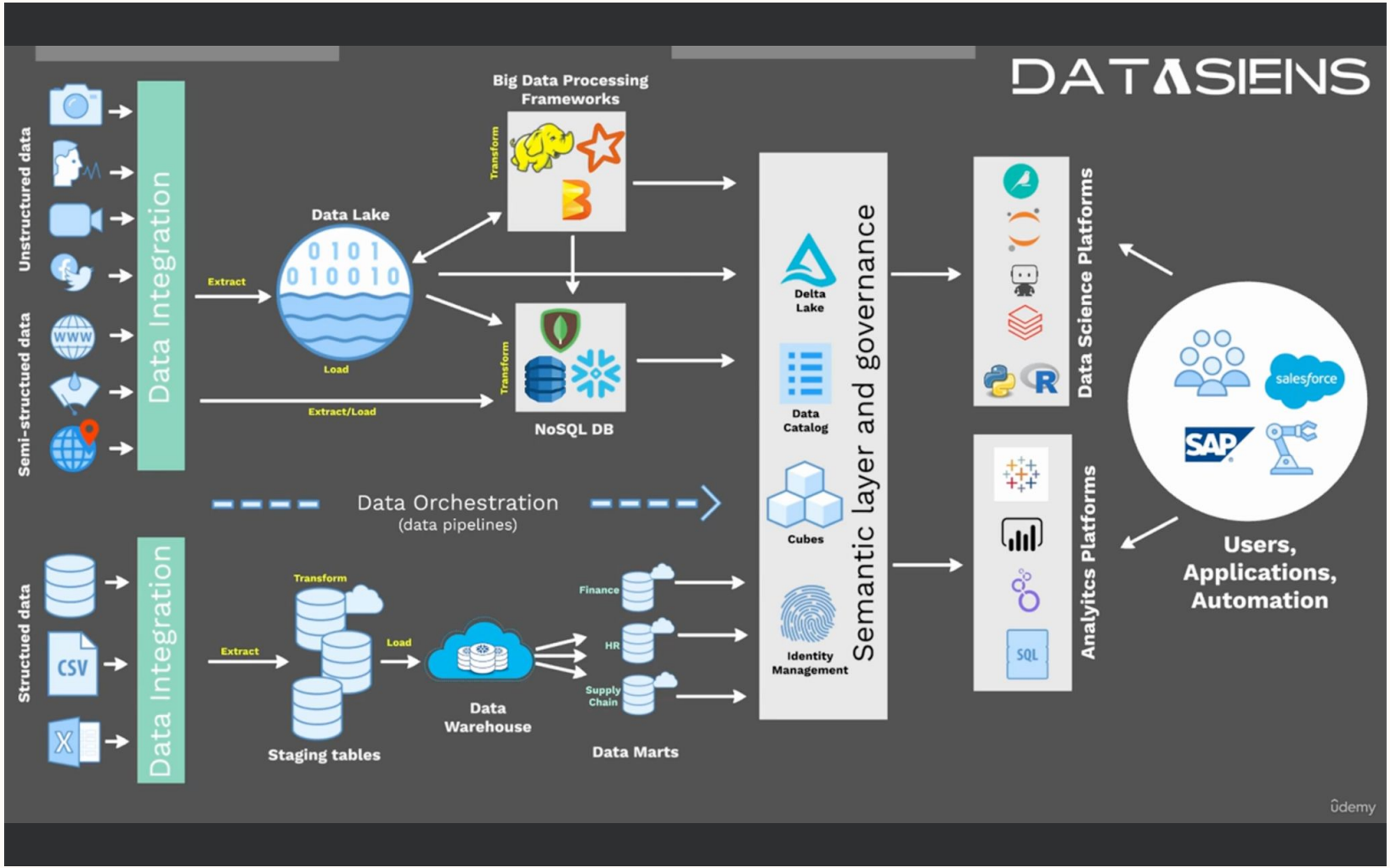
Responsable de recopilar los datos desde múltiples fuentes en tiempo real. Puede incluir: sensores, dispositivos IoT, sistemas de registro de transacciones y cualquier otra fuente de datos en tiempo real. Se encarga de garantizar la calidad y la integridad de los datos adquiridos.

Capa de procesamiento

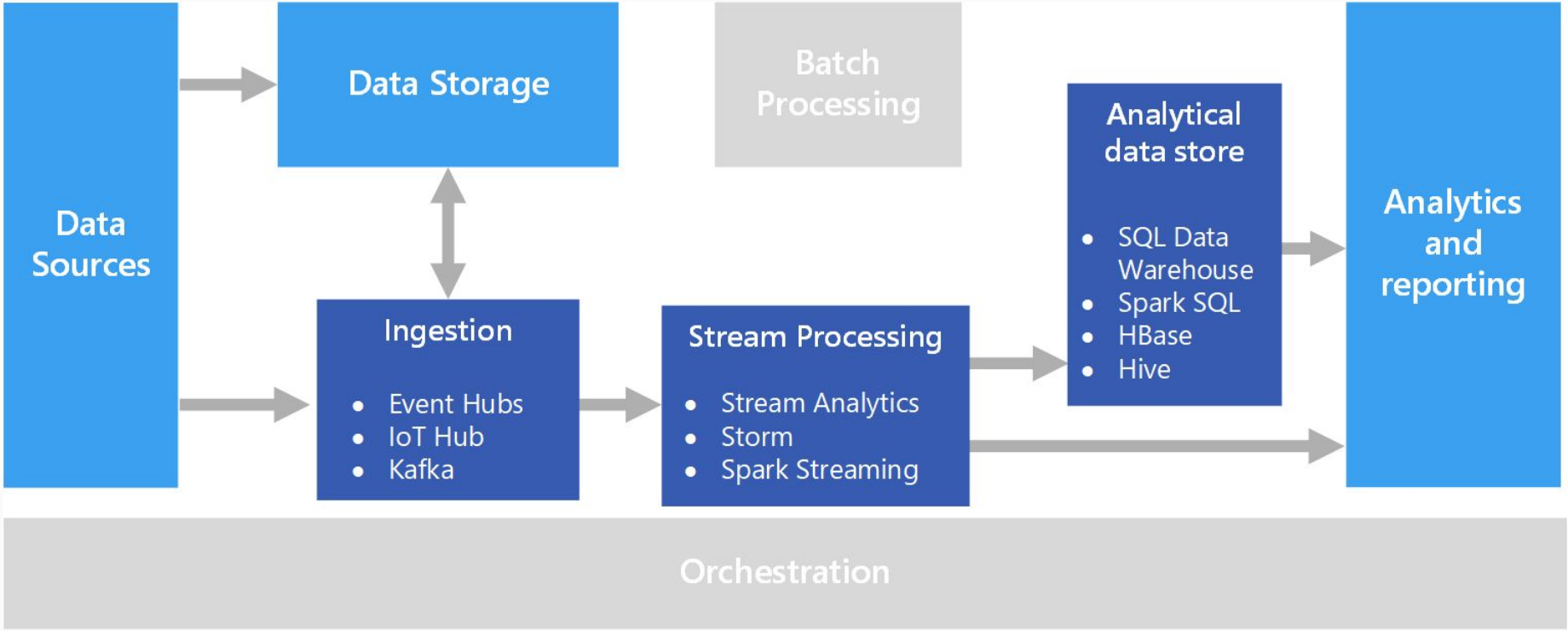
Procesa los datos capturados en tiempo real. Utiliza técnicas como el procesamiento: distribuido, en tiempo real, de flujos y de eventos complejos, para analizar y transformar los datos. Esta capa puede incluir herramientas como Apache Kafka, Apache Flink y Apache Storm.

Capa de almacenamiento y análisis de datos

Responsable de recopilar los datos desde múltiples fuentes en tiempo real. Puede incluir: sensores, dispositivos IoT, sistemas de registro de transacciones y cualquier otra fuente de datos en tiempo real. Se encarga de garantizar la calidad y la integridad de los datos adquiridos.


DATASIENS
Unstructured data
Semi-structued data
Structued data
Data Integration
Data Integration
Big Data Processing Frameworks
Transform
Data Lake
Extract
Load
Extract/Load
Transform
NoSQL DB
Data Orchestration
(data pipelines)
Transform
Extract
Load
Staging tables
Data Warehouse
Finance
HR
Supply Chain
Data Marts
Delta Lake
Data Catalog
Cubes
Identity Management
Semantic layer and governance
Data Science Platforms
Analyitcs Platforms
SQL
salesforce
SAP
Users, Applications, Automation
udemy

SOLUCIÓN AZURE



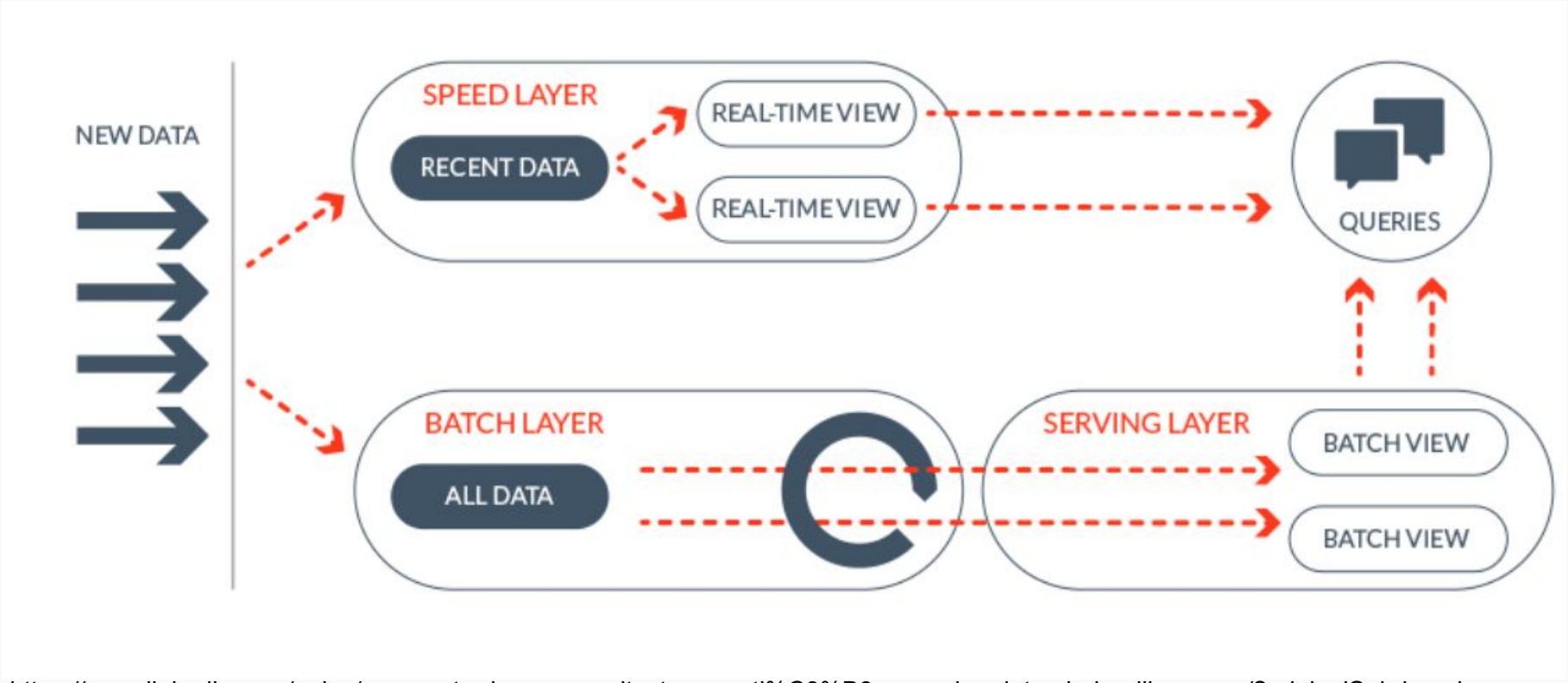
https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/real-time-processing

ARQUITECTURA LAMBDA

- Esta arquitectura combina la arquitectura **Batch Processing** y **Real-Time Processing**.
- Los datos son procesados en lotes y en tiempo real simultáneamente, lo que permite obtener resultados rápidos y precisos.
- Los datos históricos se procesan en la capa de procesamiento por lotes, mientras que los datos en tiempo real se procesan en la capa de procesamiento en tiempo real.
- La combinación de ambas capas proporciona una vista completa de los datos que puede ser utilizada para análisis, visualización y toma de decisiones en tiempo real.

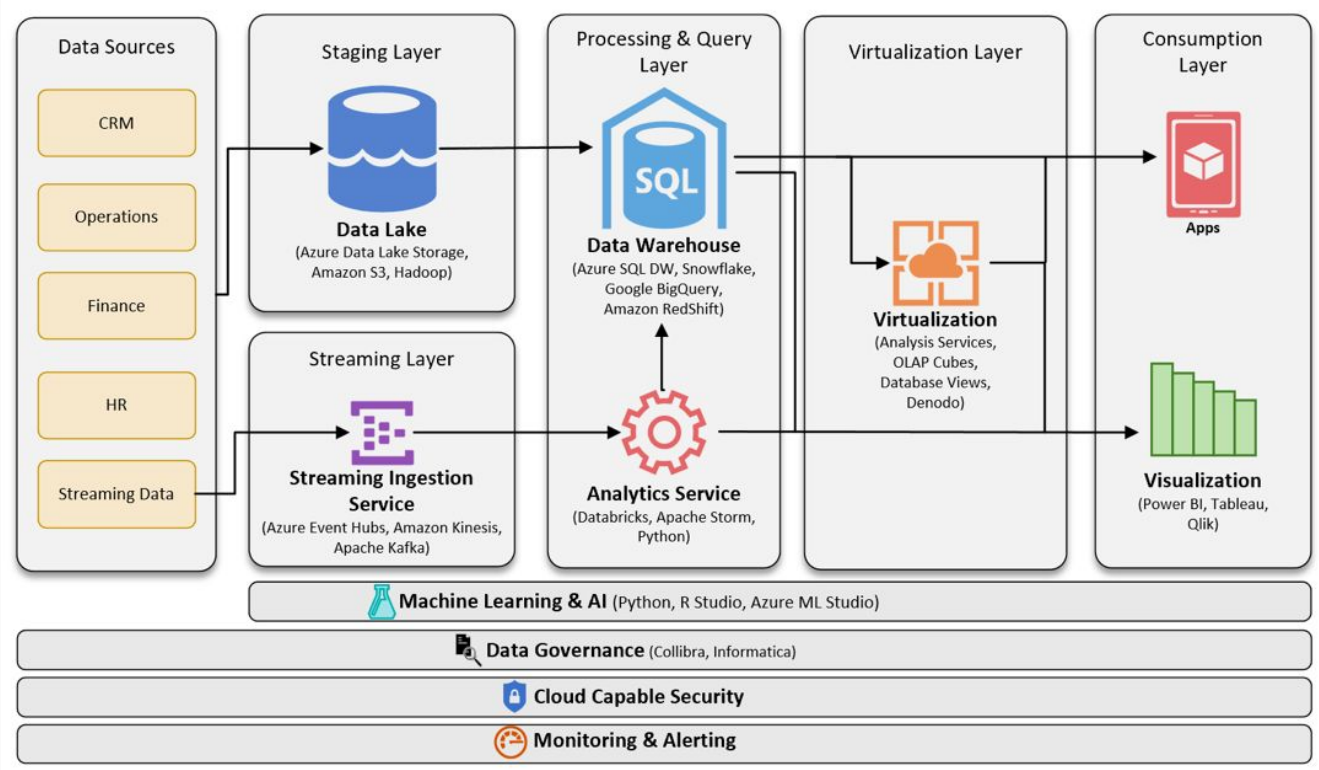
Introducido en 2012 por Nathan Marz con base en su artículo "How to bet the CAP theorem"

ARQUITECTURA LAMBDA

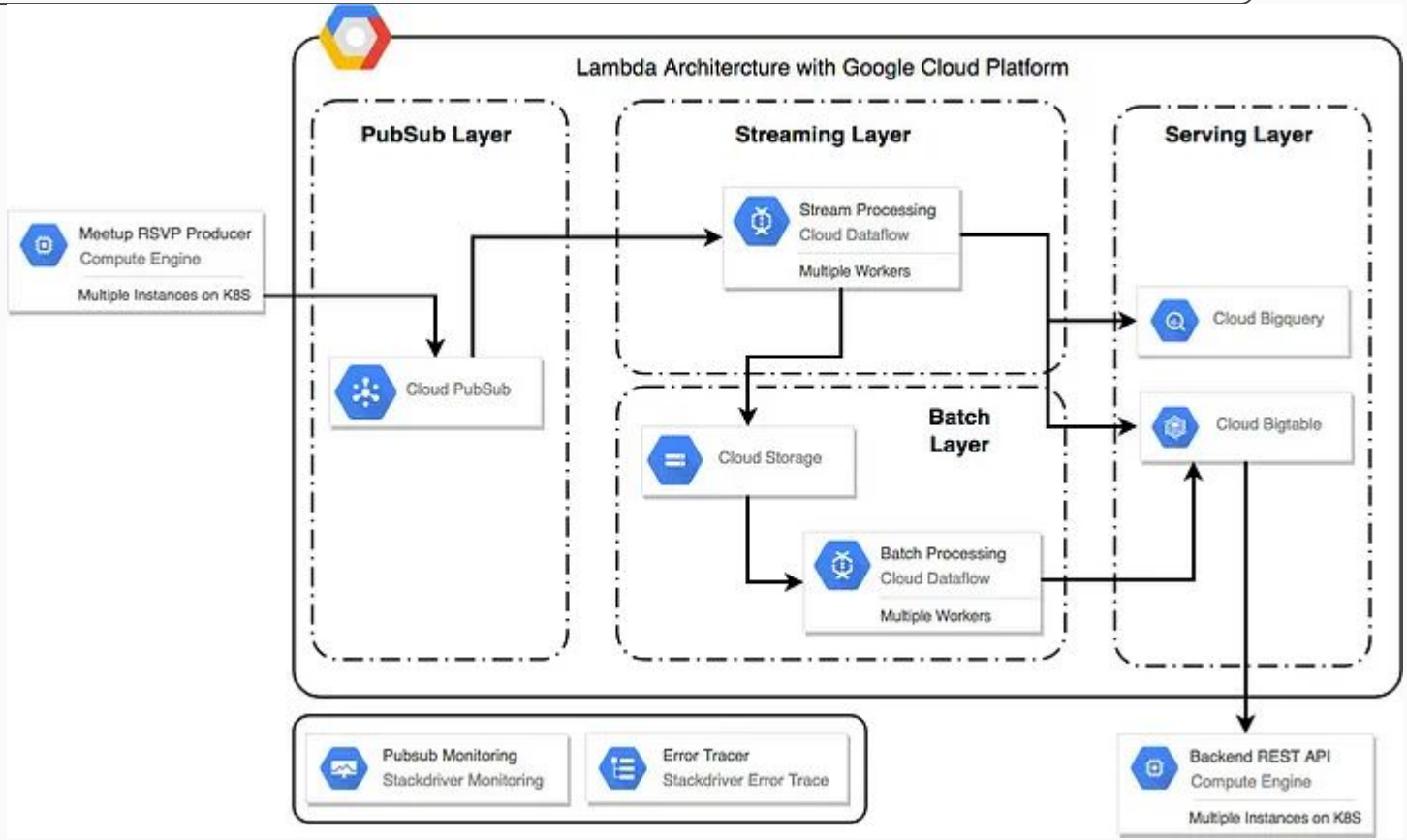


https://www.linkedin.com/pulse/propuesta-de-una-arquitectura-gesti%C3%B3n-grandes-datos-la-bonillo-ramos/?originalSubdomain=es

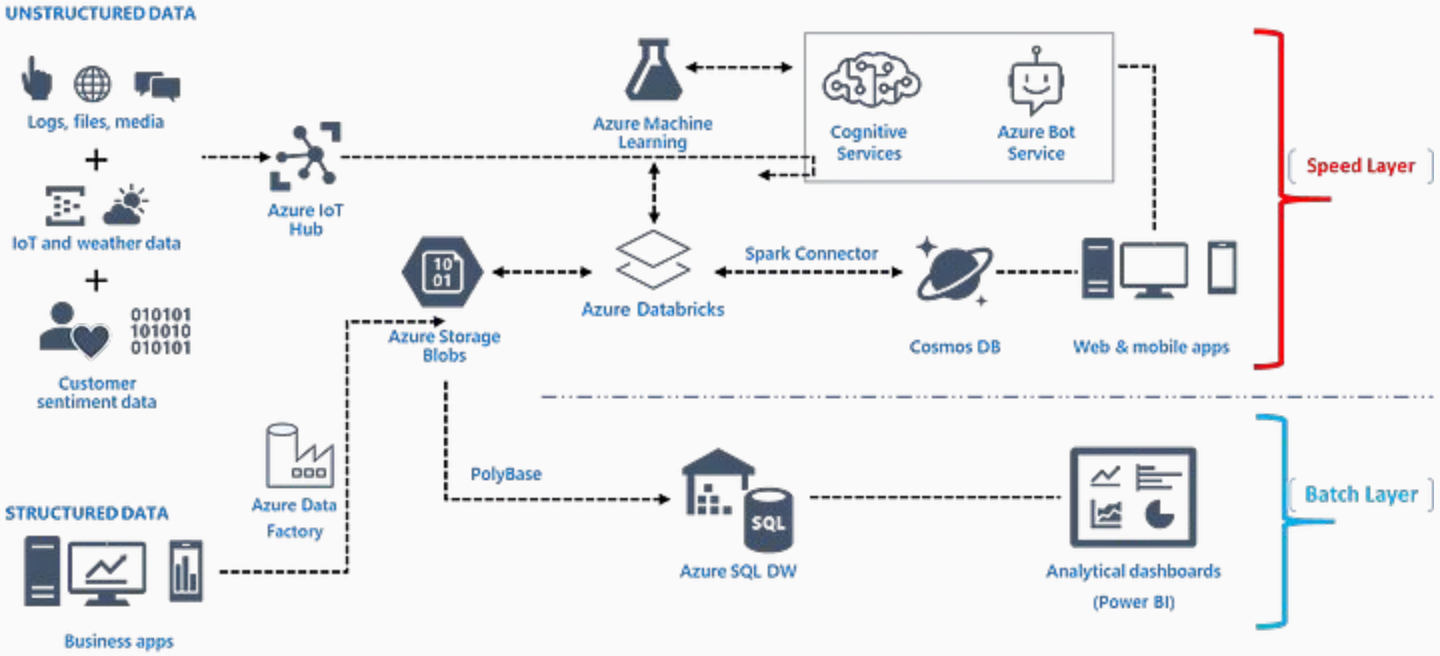
EJEMPLO ARQUITECTURA LAMBDA



SOLUCION GOOGLE

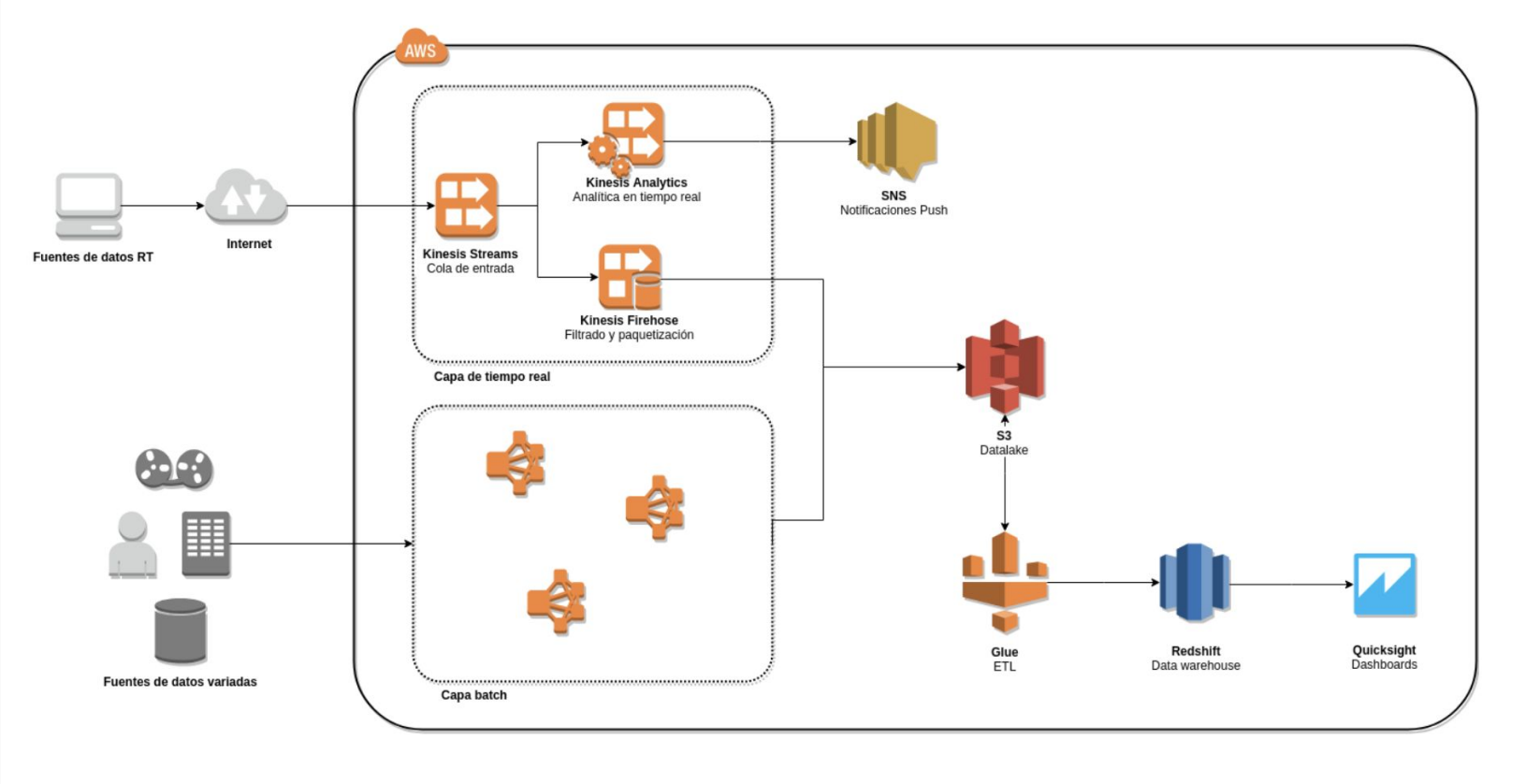


SOLUCIÓN AZURE



https://medium.com/sagar-explains-azure-and-analytics-azure-series/data-processing-architecture-lambda-and-kappa-ebb54029c893

SOLUCIÓN AWS



ARQUITECTURA KAPPA

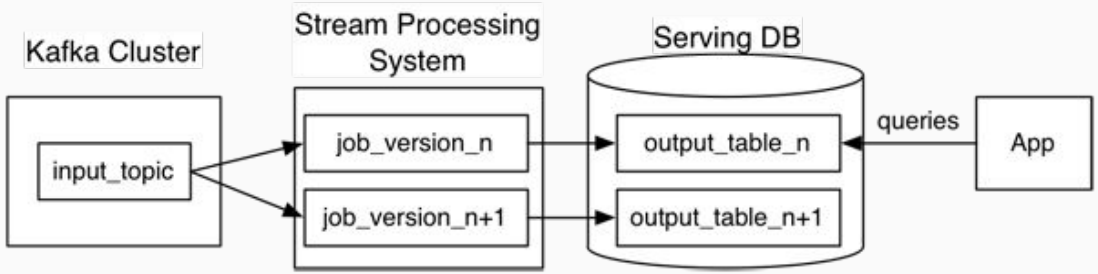
Esta arquitectura es similar a la arquitectura Lambda, pero en lugar de tener dos sistemas de procesamiento de datos, se utiliza solo uno, de procesamiento por flujo. Esto permite simplificar la arquitectura y reducir el costo.

Características:

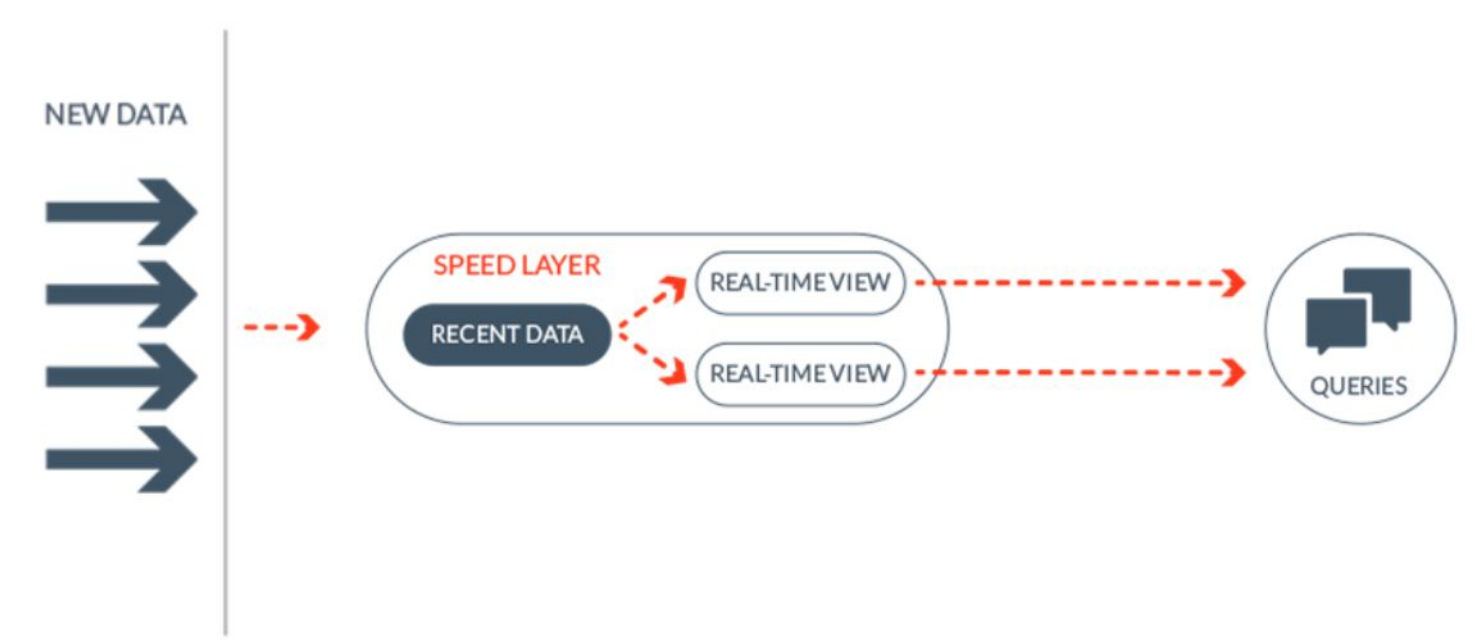
- Todo se maneja como un stream.
- Los datos origen no se modifican.
- Hay un solo flujo.
- Se puede volver a lanzar un procesamiento.

- Los eventos deben ser leídos en orden.

Introducido en 2014 por Jay Kreps en su artículo "Questioning the Lambda Architecture"

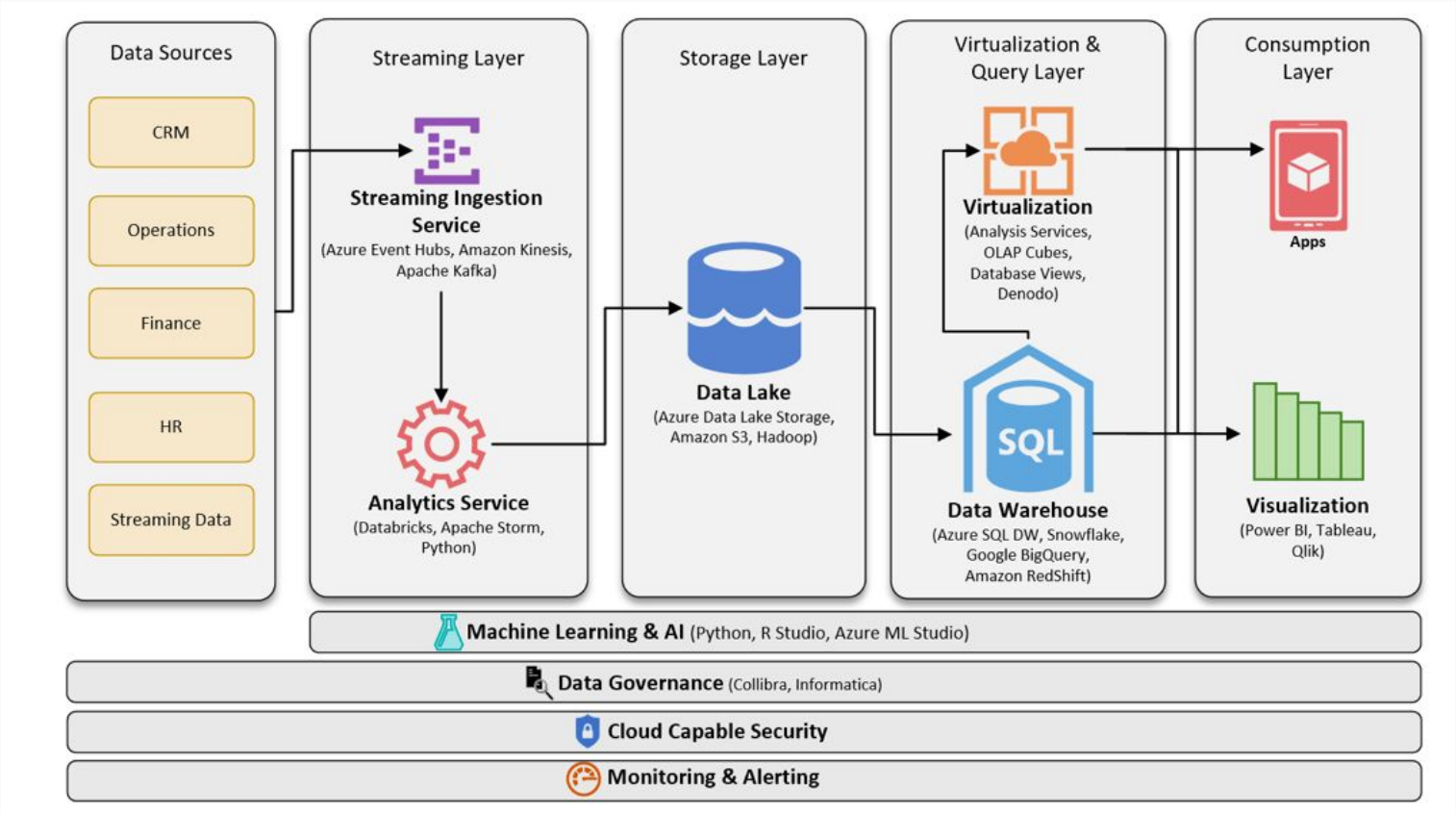


ARQUITECTURA KAPPA

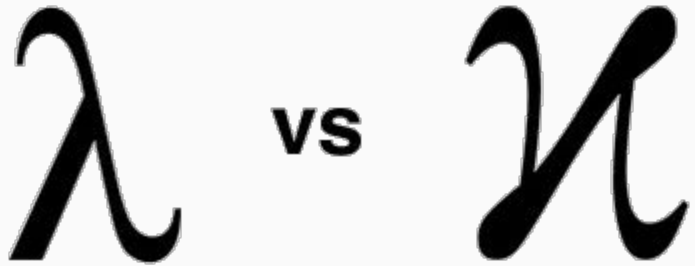


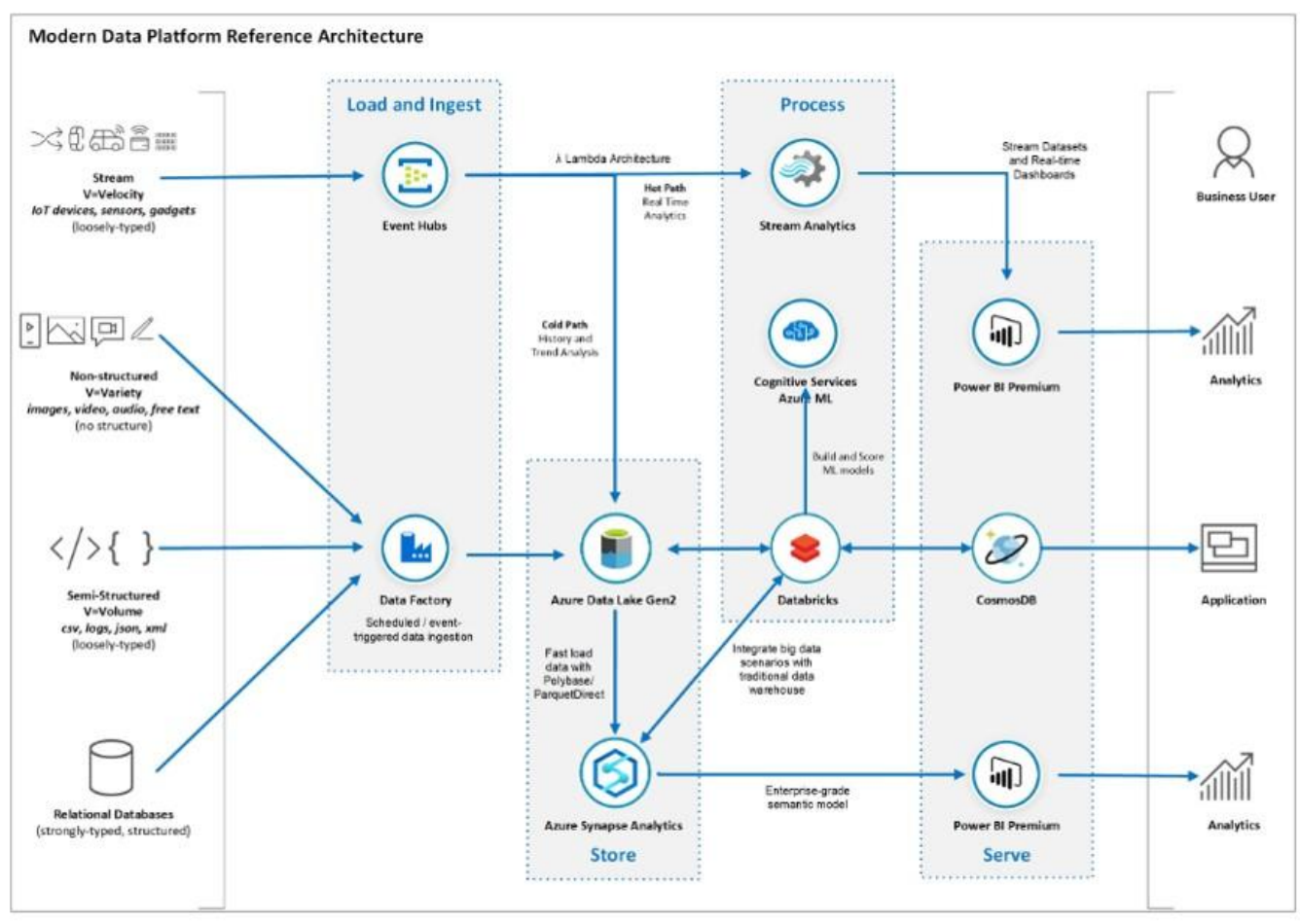
https://www.oreilly.com/radar/questioning-the-lambda-architecture/

EJEMPLO KAPPA



LAMBDA VS KAPPA

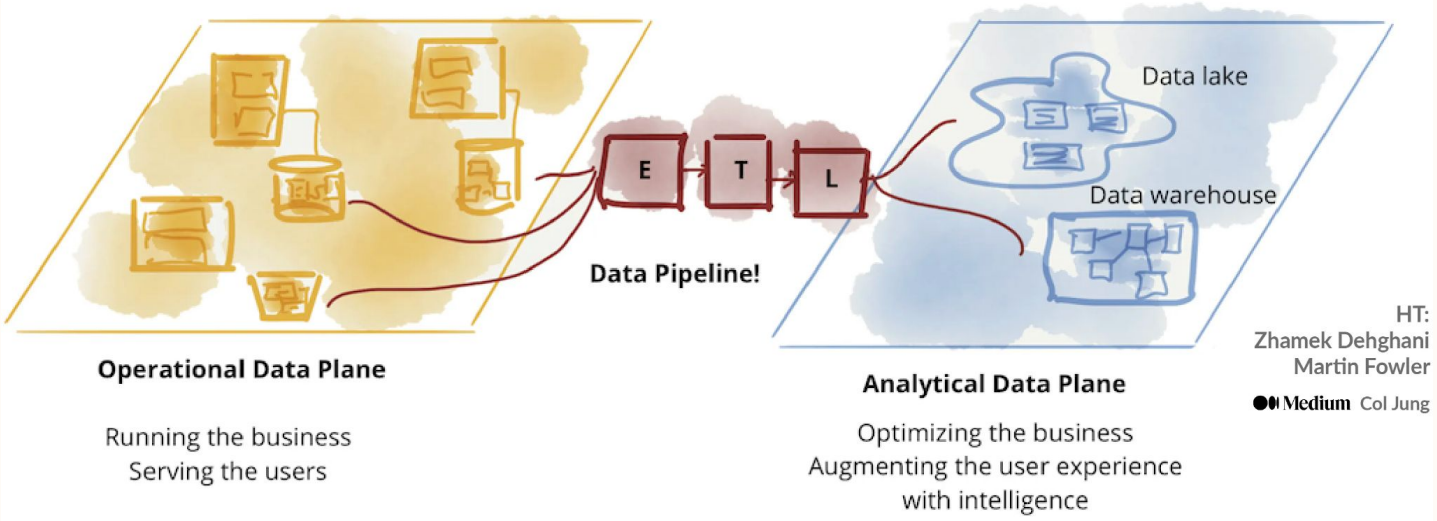



Modern Data Platform Reference Architecture
Load and Ingest
Process
Stream
V=Velocity
IoT devices, sensors, gadgets
(loosely-typed)
Event Hubs
A Lambda Architecture
Hot Path
Real Time Analytics
Stream Analytics
Stream Datasets and Real-time Dashboards
Business User
Cold Path
History and Trend Analysis
Cognitive Services
Azure ML
Power BI Premium
Analytics
Non-structured
V=Variety
images, video, audio, free text
(no structure)
Build and Score ML models
Semi-Structured
V=Volume
csv, logs, json, xml
(loosely-typed)
Data Factory
Scheduled / event-triggered data ingestion
Azure Data Lake Gen2
Databricks
CosmosDB
Application
Fast load data with Polybase/ ParquetDirect
Integrate big data scenarios with traditional data warehouse
Relational Databases
(strongly-typed, structured)
Azure Synapse Analytics
Enterprise-grade semantic model
Power BI Premium
Analytics
Store
Serve

OLTP/OLAP

Operational & Analytical Data

The Great Divide



DATA PROCESSING APPROACHES

BUSINESS LOGIC VS BUSINESS INTELLIGENCE

Business Logic

- The set of rules, processes, and algorithms that define how a business operates and makes decisions.
- Often implemented within software applications and systems to ensure that they function according to the specific requirements and processes of the business.
- It focuses on automating and enforcing the business rules and processes to maintain consistency, accuracy, and efficiency in day-to-day operations.

Business Intelligence

- Is the process of collecting, analyzing, and presenting data to gain insights and support decision-making.
- It involves the use of technology, tools, and techniques to transform raw data into meaningful and actionable information.
- aims to provide stakeholders with a holistic view of the business by
  - Consolidating data from various sources
  - applying analytical methods to uncover patterns and trends
  - Presenting the findings in reports, dashboards, and visualizations.

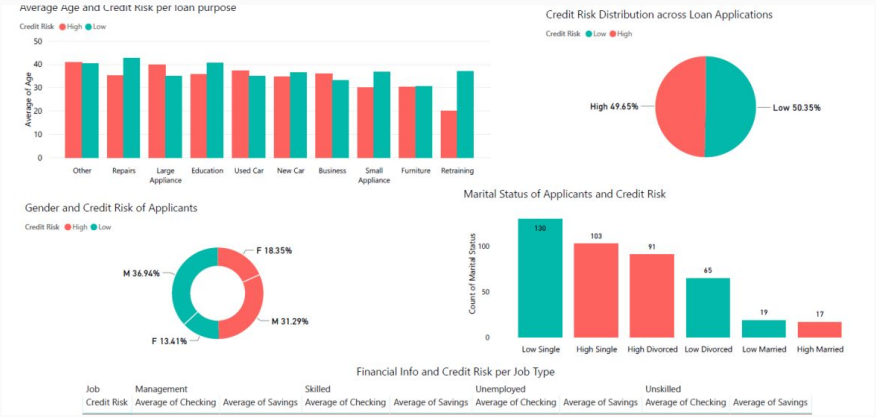
BUSINESS LOGIC VS BUSINESS INTELLIGENCE

Business Logic

- E-commerce Systems
- Banking System.
- Airline Reservation System
- Housing systems

Business Intelligence

- Sales Analysis System
- Credit Performance
- Marketing Analytics Platform.
- Supply Chain Analytics System: A system



OLTP & OLAP

OLTP (Online Transaction Processing) and **OLAP** (Online Analytical Processing) are two distinct approaches to processing and managing data in the field of database management.

- **OLTP** systems handle day-to-day operational transactions of an organization in real time. Supports **Business Logic**.
- **OLAP** systems are used for analyzing and querying data to gain insights, make informed decisions, and support business intelligence activities. Supports **Business Intelligence.**

ACID PROPERTIES AND TRANSACTIONAL SYSTEMS

Fundamental principles in database management systems that ensure the reliability and consistency of transactions.

- **Atomicity:** This property ensures that each transaction is treated as a single unit of operation. Either all the operations within the transaction are completed successfully and committed to the database, or none of them are.
- **Consistency:** Consistency ensures that the database remains in a valid state before and after the transaction.
- **Isolation:** Isolation ensures that the concurrent execution of transactions does not result in interference between them. Each transaction appears to be executed in isolation from other transactions, even if they are executed simultaneously.
- **Durability:** Durability guarantees that once a transaction is committed, its changes are permanent and remain in the system, even in the event of a system failure.

OLTP VS OLAP

OLTP

Purpose: OLTP systems focus on recording and processing individual transactions, database CRUD.

Structure: The data is typically organized in a highly normalized structure. It **minimizes redundancy** and ensures efficient transaction processing. The emphasis is on maintaining data integrity and supporting efficient transactional operations.

Workload: Handle a high volume of small, individual transactions in real time, focusing on maintaining data **consistency** and **integrity**.

OLAP

Purpose: OLAP systems provide a multidimensional view of data to support strategic decision-making.

Structure: often use a denormalized or dimensional data model. Data is structured in a way that **facilitates complex analysis** and reporting. It involves creating multidimensional structures.

Workload Deal with complex queries and aggregations performed on large volumes of historical data. The workload typically consists of read-intensive operations for **analysis,** reporting, and **decision support**.

OLTP VS OLAP

OLTP

Performance Requirements: The primary focus of OLTP systems is on transactional processing with low response times. They are optimized for quick and concurrent access to ensure efficient handling of real-time transactions.

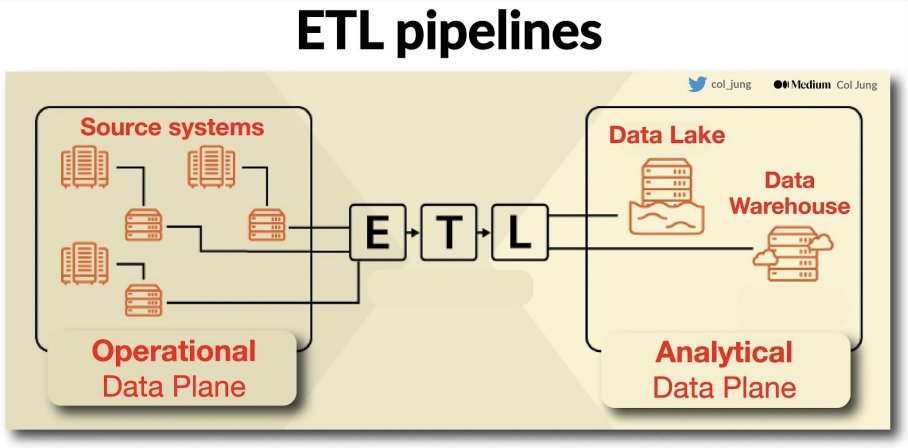
User Base: OLTP systems cater to operational staff, such as customer service representatives, who perform daily transactional tasks.

OLAP

Performance Requirements: OLAP systems prioritize query performance and the ability to process complex analytical queries efficiently. The emphasis is on providing fast response times for ad hoc queries and supporting advanced analytics.

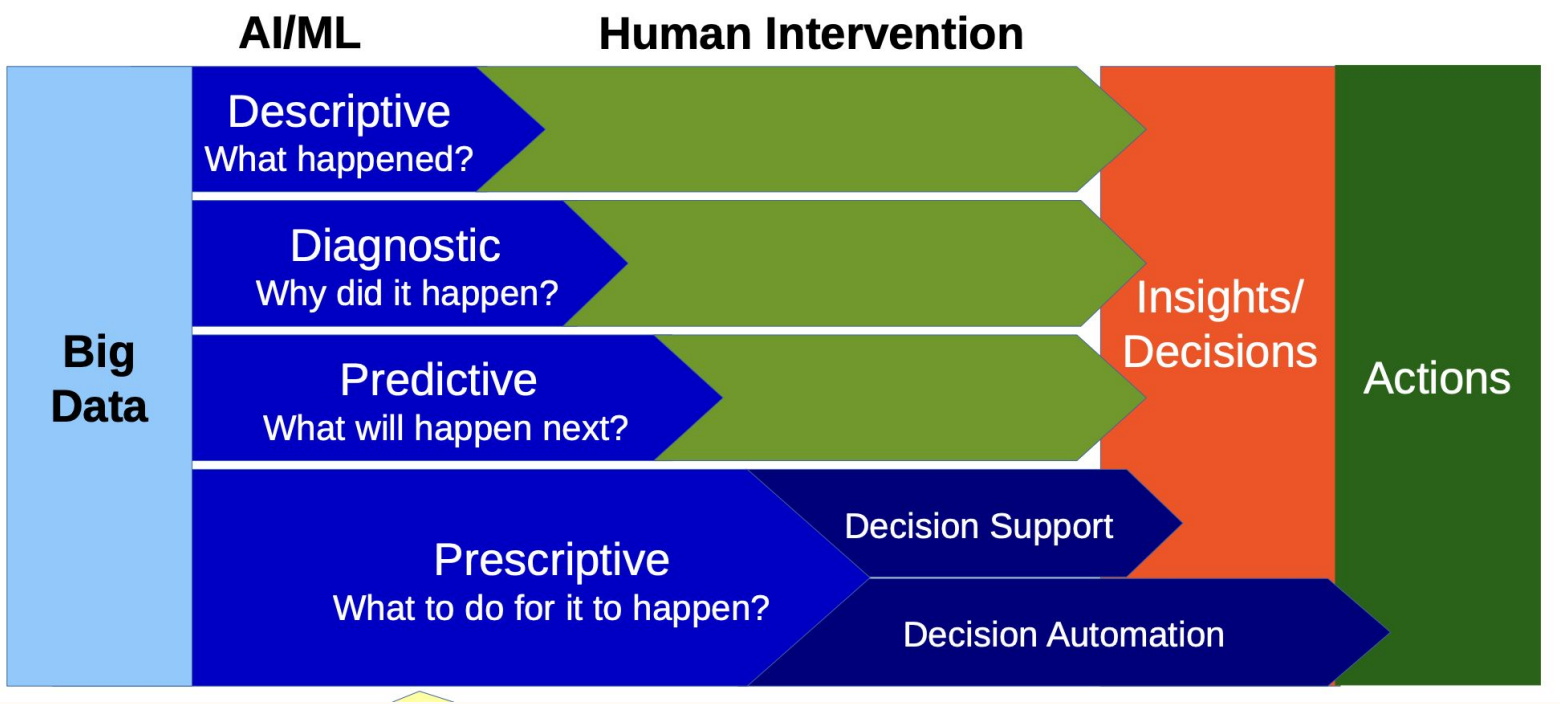
User Base: OLAP systems are typically used by business analysts, managers, and decision-makers who require in-depth analysis and reporting capabilities.

FROM OLTP TO OLAP: DATA INGESTION

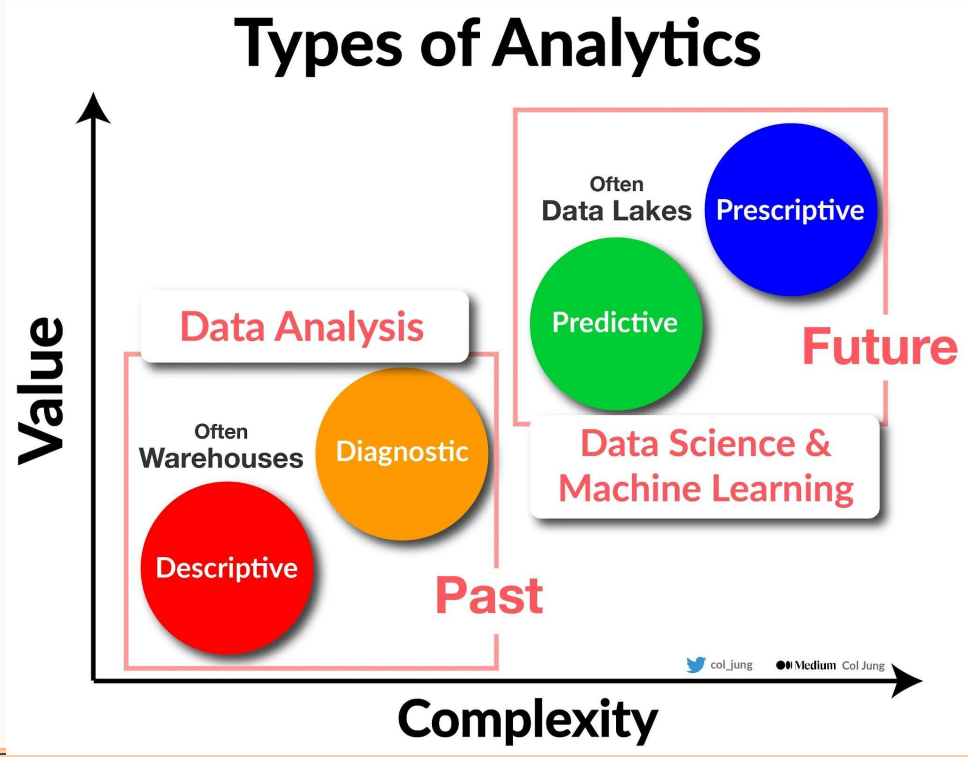


Data ingestion is the process of collecting, importing, and loading data from various sources into a storage or processing system, such as a data warehouse, data lake, or database.

It involves acquiring data from different sources, transforming it (if necessary), and making it available for analysis, processing, or storage



TYPES OF ANALYTICS

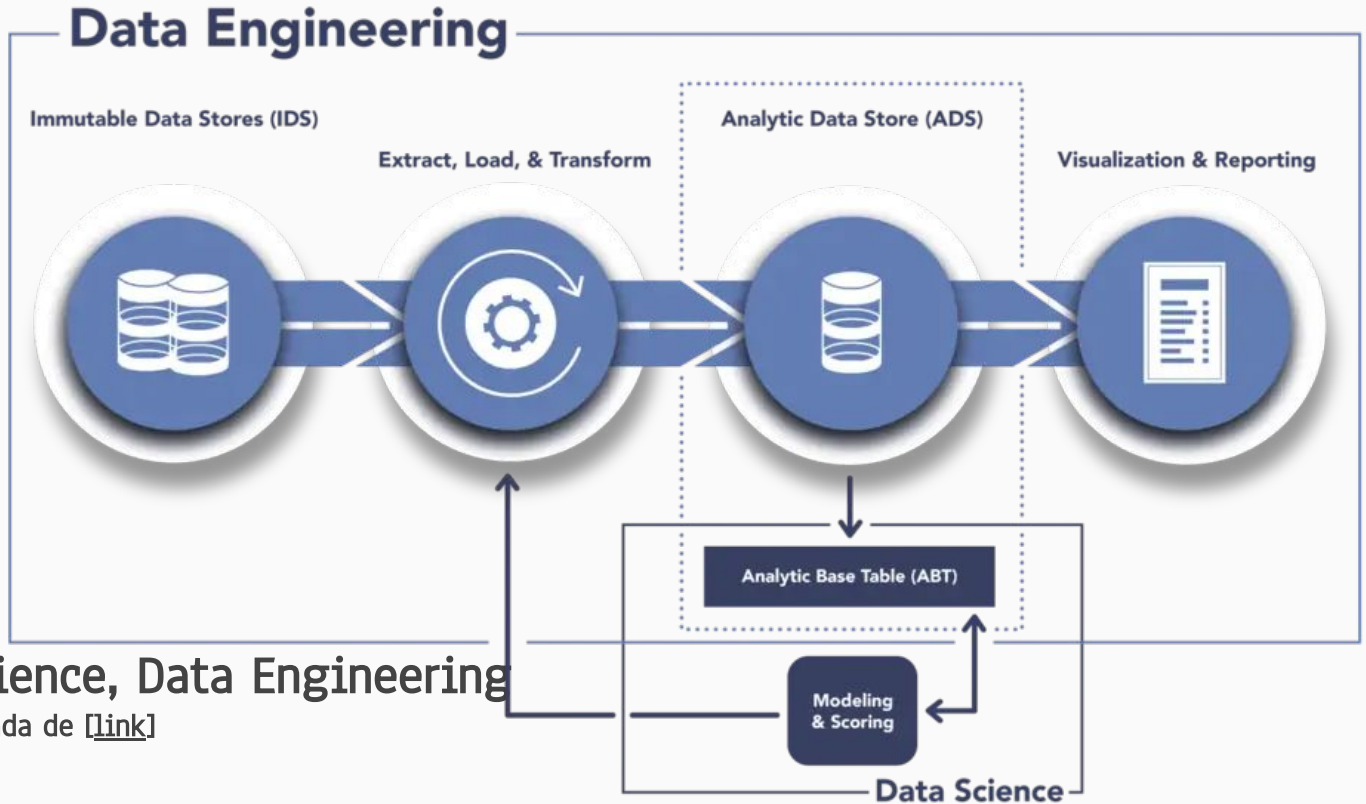


REPOSITORIES FOR BI


Data
Lake
Lakehouse
One platform to unify all of your data, analytics, and AI workloads
Modern Data
Warehouse

FLASHBACK TO THE RELATIONAL MODEL

CONTEXTO



Data Science, Data Engineering

Imagen tomada de [link]

THE RELATIONAL MODEL & THE ONE MILLION DOLLAR IDEA

The relational model is a **Data Model**. Data models are ways to structure information:

Movies	*Title*	*Director*	*Actor*
	The Trouble with Harry	Hitchcock	Gwenn
	The Trouble with Harry	Hitchcock	Forsythe
	The Trouble with Harry	Hitchcock	MacLaine

Location	*Theater*	*Address*	*Phone Number*
	Gaumont Opéra	31 bd. des Italiens	47 42 60 33
	Saint André des Arts	30 rue Saint André des Arts	43 26 48 18
	Le Champo	51 rue des Ecoles	43 54 51 60

Pariscope	*Theater*	*Title*	*Schedule*
	Gaumont Opéra	Cries and Whispers	20:30
	Saint André des Arts	The Trouble with Harry	20:15
	Georges V	Cries and Whispers	22:15

Database schema

Cinema={Movies, Location, Pariscope}
Movies={Title, Director, Actor}
Location={Theater, Address, PhoneNumber}
Pariscope={Theater, Title, Schedule}

PROPERTIES OF A RELATION (TABLE)

- Each table in a database has a **unique name.**
- Each cell contains one **atomic** value.
- Attributes in a table have unique names.
- All attributes' values belong to the same domain.
- There are **no duplicate rows** in a table.
- The order of the tuples and the order of the attributes have no meaning.

RELATIONAL KEYS

Relational keys are attributes (one or more) that uniquely identifies each tuple in a relation.

- **Primary Key.** The attribute or set of attributes selected to identify rows uniquely within a relation
- **Foreign Key.** An attribute or set of attributes, within one relation that matches the key of some other relation.

EXAMPLE

DEPTS

DEPTNO (PK)	DNAME (NN,U)	LOCATION (NN)
D1	Comercialización	10M
D2	Ventas	20M
D3	Investigación	5M

PRIMARY KEY

FOREIGN KEY

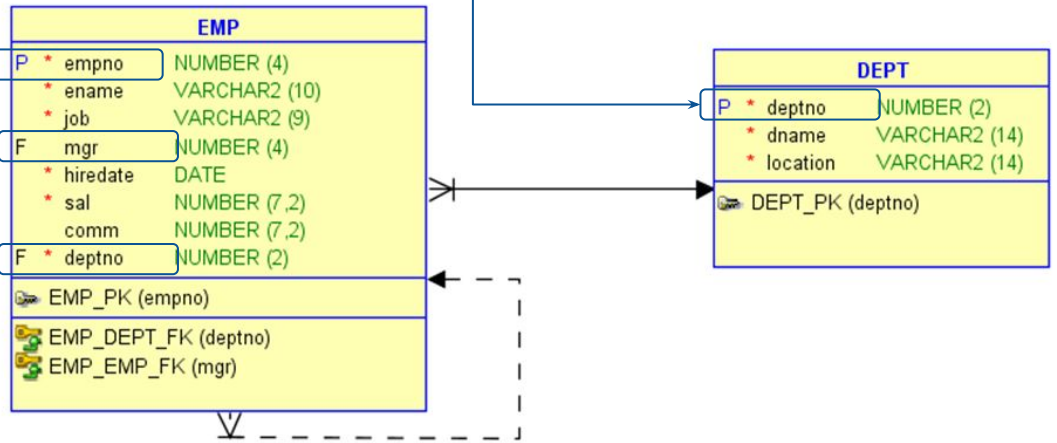
EMPS

EMPNO (PK)	ENAME	SAL	…	DEPTNO (FK)
E1	López	3400000	…	D1
E3	Gutiérrez	2000000	..	D2
E2	Fernández	800000	..	D1
E4	Jaramillo	500000	..	D3

MODEL

empno is the primary key (PK) in table EMP

deptno is the primary key (PK) identifying a department it makes a row unique in table DEPT



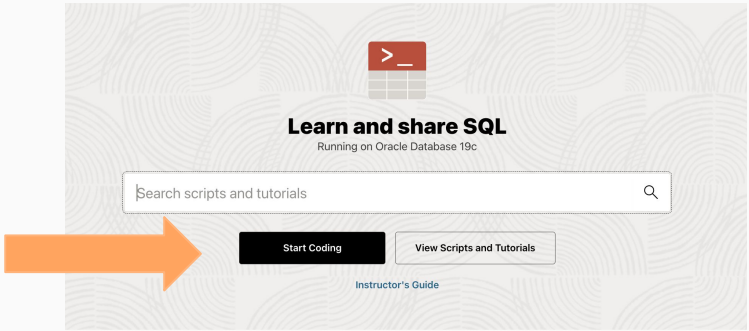
mgr is a foreign key in table EMP, it references the empno at table EMP

deptno is also a FK in table EMP. It references a row in DEPT, providing info about the department where an employee works

PRÁCTICA

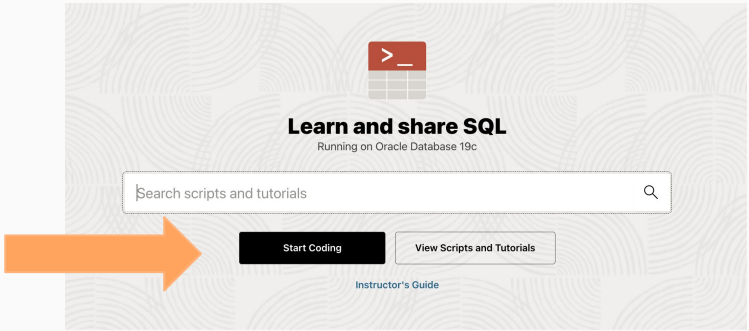
INSTRUCCIONES

- Vamos a trabajar con Oracle Live SQL [link] entonces debes ingresar (si no tienes cuenta de oracle, debes crear tu cuenta)
- Después de haber ingresado, das click en **start coding**

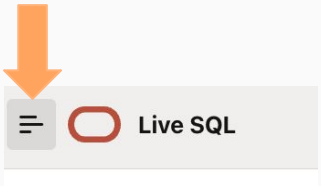


INSTRUCCIONES

- Vamos a trabajar con Oracle Live SQL [link] entonces debes ingresar (si no tienes cuenta de oracle, debes crear tu cuenta)
- Después de haber ingresado, das click en **start coding**

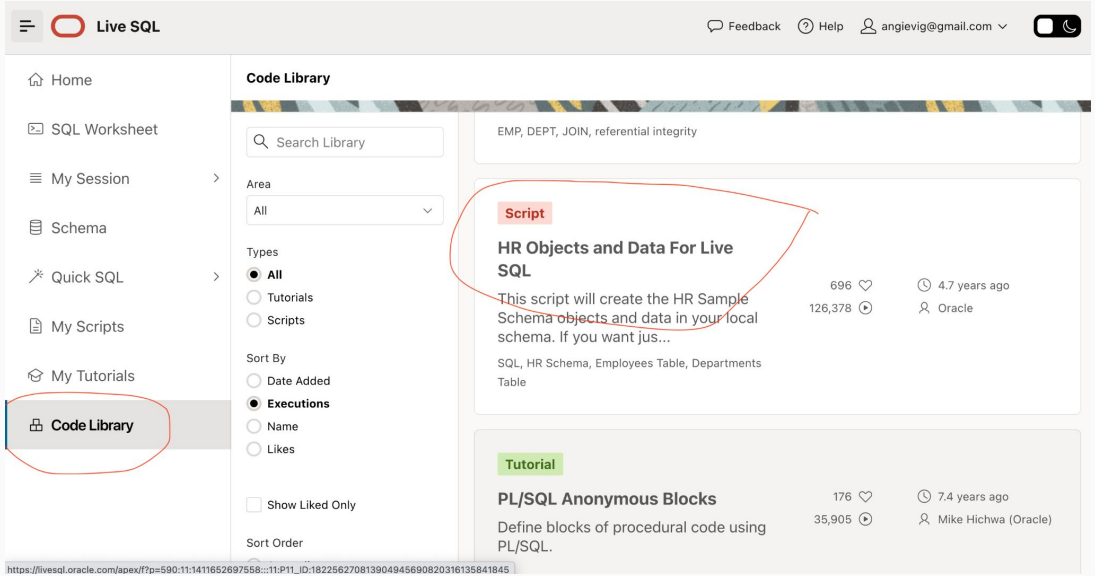


- En la esquina superior izq activas el menú



INSTRUCCIONES

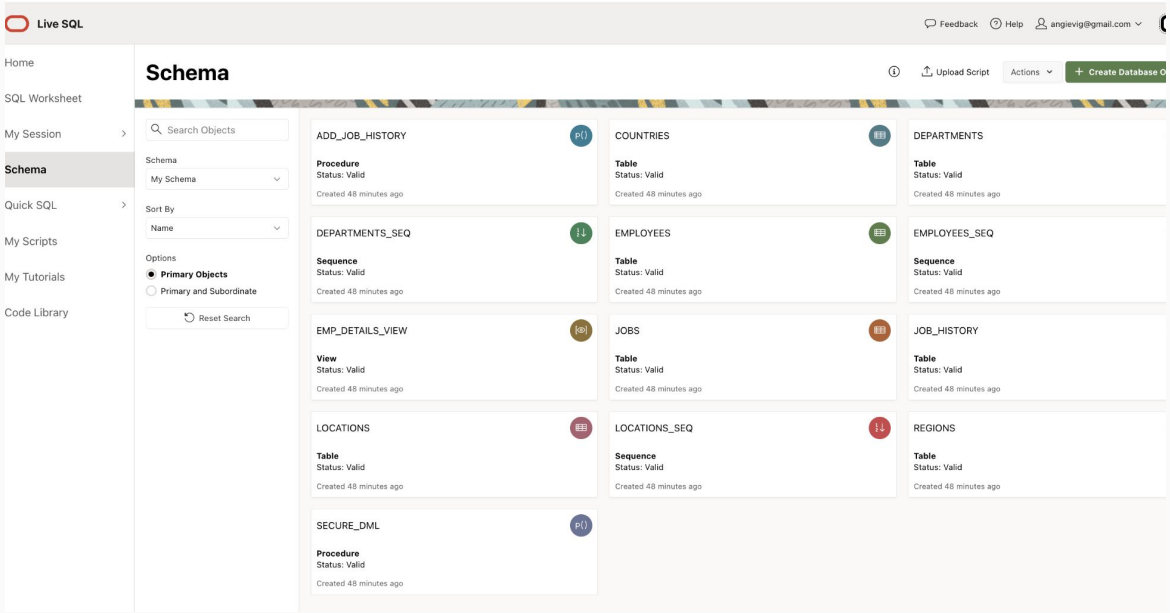
- En el menú **code library** escoge la tercera opción, para cargar el script del problema HR Objects and Data for Live



Luego, en la esquina superior derecha das click en **run script**

INSTRUCCIONES

- Luego revisa en la opción **Schema** que los objetos se hayan cargado correctamente

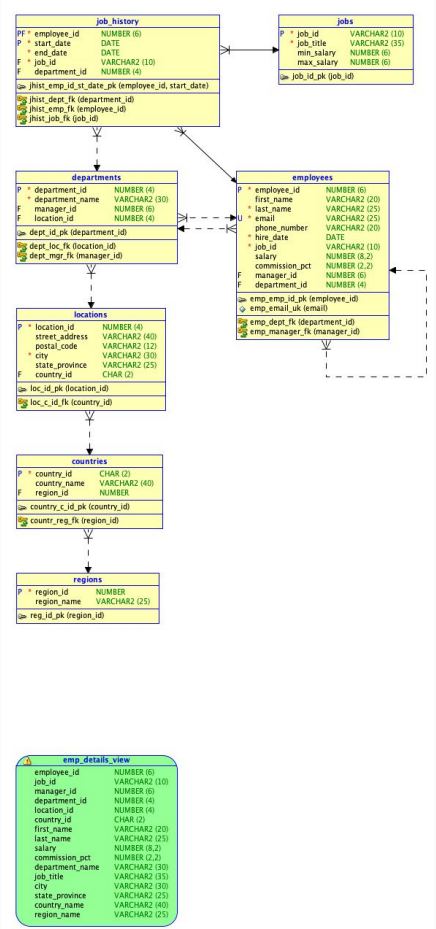


Ahora estás listo para interactuar con la bd en la opción **SQL Worksheet**

DB RELATIONAL MODEL

Open Link

Model generated with Oracle DataModeler Tutorial [link]



ACTIVIDADES

- Interprete el modelo relacional, para esto:
  - Identifique las llaves primarias y foráneas de cada tabla.
  - Describa la relación que representa cada conector o llave foránea en el modelo. **EJ:** un empleado trabaja en un departamento, una locación está ubicada en un país. etc
  - Identifique la información que se puede “cruzar” por medio de operaciones Join.
- Ejecute las consultas que encuentra en los slides que aparecen a continuación, interprete sus resultados

ACTIVIDADES

- Interprete el modelo relacional, para esto:
  - Identifique las llaves primarias y foráneas de cada tabla.
  - Describa la relación que representa cada conector o llave foránea en el modelo. **EJ:** un empleado trabaja en un departamento, una locación está ubicada en un país. etc
  - Identifique la información que se puede “cruzar” por medio de operaciones Join.
- Ejecute las consultas que encuentra en los slides que aparecen a continuación, interprete sus resultados
- Explore los datos, haciendo consultas en las tablas Employee, Department, Job, etc y responda: ¿Qué podemos decir sobre las comisiones de los empleados?

SELECT-FROM - WHERE REVISITED

The SELECT-FROM-WHERE clause can perform in a single statement, the three operations from the relational algebra: Projection, Restriction and Join.

The select is a closed operation as it is applied over tables and produces another table.

SELECT-FROM - WHERE REVISITED

General form - Syntax

SELECT [DISTINCT | ALL]{*|[attributeName [**AS** newName]] [,] }
FROM tableName [alias]
[WHERE condition**]**
[GROUP BY attributeList**] [HAVING** condition**]**
[ORDER BY attributeList**]**

Notation: [optional] {mandatory}

SELECT-FROM - WHERE REVISITED

General form - Semantics

FROM table(s) to be used

WHERE filters the rows subject of the condition

GROUP BY forms groups of rows with the same column value

HAVING filters the groups subject to some condition

SELECT specifies which columns will appear in the output

ORDER BY specifies the order of the output

EJEMPLO DE JOIN

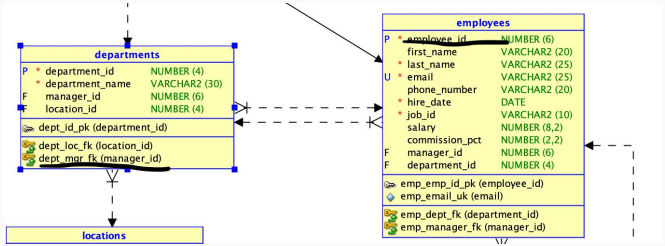
Pregunta: Cuáles son los nombres y los identificadores de los jefes de cada departamento?

Consulta: obtuvimos 11 filas

```
select d.department_Id,
       e.employee_id AS bossId,
       e.first_name AS bossName
from employees e, departments d
where e.employee_id = d.manager_id ;
```

bossId, bossName son alias para las columnas del resultado

d, e son alias para las tablas



Consulta para verificar: revisamos la tabla de departamentos, hay 27 departamentos y muchos de ellos no tienen información del jefe

Select * **from** departments;

Consulta para verificar: contamos los departamentos que tienen jefes y el resultado es !!

Select count(manager_id) from departments where manager_id IS NOT NULL

SQL AGGREGATE FUNCTIONS - GROUP - HAVING

Group-having clauses and the aggregate functions allow to include sub-totals in the reports.

- Aggregate functions cannot be part in where clauses
- Having clauses include restrictions over groups

Functions

- Count
- SUM
- AVG
- MIN - MAX

Operates Non-Null Values, except Count(*)

AGGREGATE FUNCTIONS AND NULL VALUES

Count **IGNORE Null values (Exceptions)**

```
select count(*) from employees;

select count(COMMISSION_PCT)
from employees;
```

SUM - AVG **IGNORE Null values**

```
select EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID,
      (salary + salary * COMMISSION_PCT)  AS
Total_Salary
from employees;
```

Min - Max **IGNORE Null values**

```
select
    Min(COMMISSION_PCT)  AS Min_Comission,
    MAX(COMMISSION_PCT)  AS Max_Comission
from employees;
```

Distinct **NULL counts!!**

- If Distinct is no specified, ALL is assumed
- Cannot use Distinct on Count(*)
- No value to use with MIN-MAX

```
Select Distinct  COMMISSION_PCT
from employees;
```

GROUPING DATA WITH SQL

- GROUP BY clauses may contain multiple columns (separated by comma)
- Every column in the SELECT statement must be present in a GROUP BY clause, except for aggregated calculations
- Nulls will be grouped together if the column contains null values

Examples

```
select JOB_ID, count(EMPLOYEE_ID) AS
Num_Employees
from employees
group by JOB_ID;
```

```
select  JOB_ID, COMMISSION_PCT,
count(EMPLOYEE_ID) AS Num_Employees
from employees
group by  JOB_ID, COMMISSION_PCT;
```

Preguntas:

- ¿Cómo interpretamos los resultados de las consultas de los ejemplos?
- ¿Qué pasa si invertimos las columnas `JOB_ID, COMMISSION_PCT` en el segundo ejemplo?

FILTERING GROUPS

- WHERE does not work for groups, because it filters on rows
  Use **HAVING** instead

Examples

```
select  JOB_ID, COMMISSION_PCT,
count(EMPLOYEE_ID) AS Num_Employees
from employees
group by  JOB_ID, COMMISSION_PCT
HAVING count(EMPLOYEE_ID)>1;
```

Preguntas:
¿alguna restricción para los ejemplos del slide anterior?

```
select JOB_ID, count(EMPLOYEE_ID) AS
Num_Employees
from employees
group by JOB_ID;
```

Little piece of advice: use ORDER BY to improve the results' readability

SOME ADVICES

- Use ORDER BY to improve the results' readability
- Test how many records you have **before**, and how many you have **after** aggregating data
- Filter data is a good idea to reduce the size of the dataset before applying more complex operations.
- Indent correctly your code or use Poor SQL to make the job for you [link]

CIERRE DE LA CLASE

LA PRÓXIMA SEMANA

Fecha: Viernes, 3 de mayo de 2024
Tema: Tipos de repositorios y flujos de ingesta de datos. (DW, Datalake)

Asignaciones:
- Quiz de la unidad 1 (para el inicio de la Sesión 3)
- Terminar el ejercicio del modelo relacional
- Mapa mental del texto de Connolly sobre OLAP y DW

THANKS

apvillota@icesi.edu.co
mmrojas@icesi.edu.co

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Storyset**

Does anyone have any questions?