🎓 PROYECTO FINAL – ARQUITECTURA MEDALLION CON AZURE + DATABRICKS

Autor: César Fernando Campos Millán

Curso: Ingeniería de Datos e IA con Databricks

Fecha: Noviembre 2025

🧭 1. Introducción

El presente proyecto implementa un flujo completo de procesamiento de datos utilizando la Arquitectura Medallion sobre Azure Databricks y Azure Data Lake Storage (ADLS).

El objetivo principal es construir un pipeline moderno y escalable que permita:

- Ingestar datos crudos desde ADLS
- Transformarlos mediante PySpark
- Organizar los datos en capas (Bronze, Silver, Gold)
- Publicar tablas curatoradas en Delta Lake
- Automatizar el ETL con Databricks Jobs
- Crear dashboards analíticos basados en la capa Gold

🏗 2. Arquitectura del Proyecto

El pipeline sigue la arquitectura Medallion:

RAW → BRONZE → SILVER → GOLD → Dashboards

Servicios utilizados:

Azure Data Lake Storage Gen2 (ADLS)

Azure Databricks (Unity Catalog)

Delta Lake

Databricks Jobs / Workflows

Databricks SQL Dashboards

```
┌──────────────────────────────────────┐
│   Azure Data Lake Storage (ADLS)   │
│      Contenedores Medallion      │
│ • bronze  • silver  • gold        │
└───────────────────┬──────────────────┘
                    │
                    ▼
┌──────────────────────────────────────────────────────────┐
│             Azure Databricks (UC)             │
│ ┌────────────────────────────────────────────────────┐ │
│ │           Unity Catalog: smartdata           │ │
│ │Schemas: bronze  |  silver  |  gold            │ │
│ └────────────────────────────────────────────────────┘ │
│ ┌────────────────────────────────────────────────────┐ │
│ │          Notebooks PySpark (ETL)          │ │
│ │  • Bronze Layer                        │ │
│ │  • Silver Layer                        │ │
│ │  • Gold Layer                          │ │
│ └────────────────────────────────────────────────────┘ │
└────────────────────────────┬─────────────────────────────┘
                    │
                    ▼
┌──────────────────────────────────────┐
│     Workflow (Job) ETL       │
│     job_medallion_etl_full       │
│  Orquesta Bronze → Silver → Gold   │
└───────────────────┬──────────────────┘
                    │
```

```
                         ▼
        ┌──────────────────────────────┐
              │        GOLD Layer        │
           │   Tabla final: sales_marketing_gold │
        └───────────────┬───────────────┘
                         │
                         ▼
        ┌──────────────────────────────┐
             │    Dashboards Analíticos    │
             │ • Ventas por Categoría       │
                │ • Spend por Segmento
```

🌩️ 3. Azure Data Lake – Capa de Almacenamiento

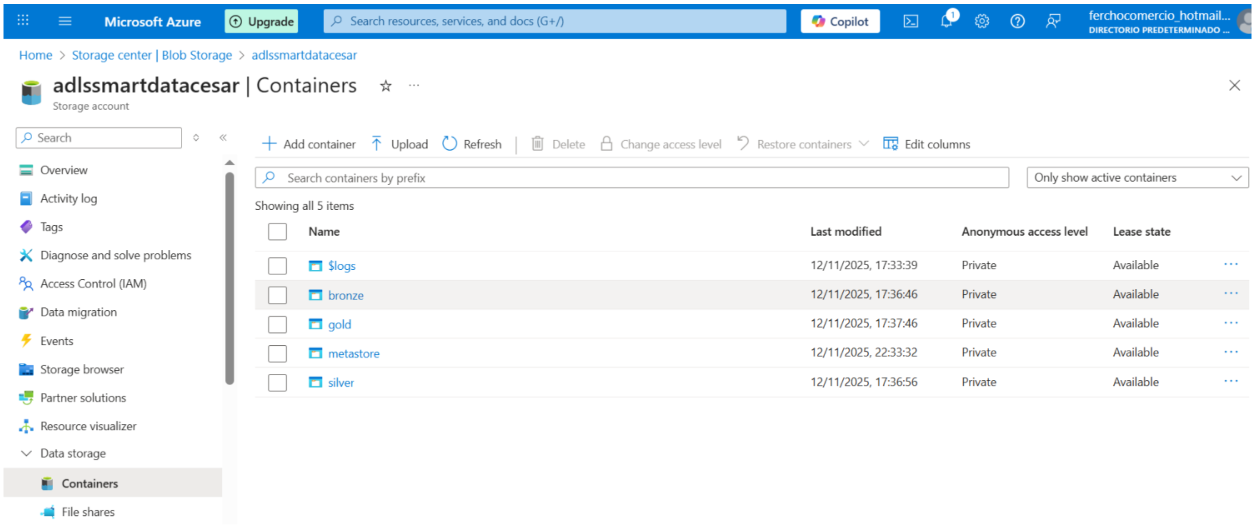
Se configuró un Storage Account con tres contenedores:

bronze

silver

gold

Cada contenedor representa una capa de la arquitectura Medallion.



Showing all 5 items

Name	Last modified	Anonymous access level	Lease state
$logs	12/11/2025, 17:33:39	Private	Available
bronze	12/11/2025, 17:36:46	Private	Available
gold	12/11/2025, 17:37:46	Private	Available
metastore	12/11/2025, 22:33:32	Private	Available
silver	12/11/2025, 17:36:56	Private	Available

🥉 4. Capa BRONZE – Ingesta de Datos

En esta capa se almacenan los datos crudos provenientes de:

DBFS (marketing_campaign.csv)

ADLS (Ecommerce_Sales_Prediction_Dataset.csv)

Transformaciones realizadas:

Lectura en formato CSV

Inferencia automática de esquemas

Normalización mínima (renombrado de columnas cuando aplica)

Código ejemplo Bronze:

```
spark.sql("USE CATALOG smartdata")
spark.sql("USE SCHEMA bronze")

df_mkt_bronze = (
  spark.read.format("csv")
```

```
    .option("header", "true")
    .option("inferSchema", "true")
    .option("delimiter", "\t")
    .load("dbfs:/FileStore/marketing_campaign.csv")
)
df_mkt_bronze.write.format("delta").mode("overwrite") \
  .saveAsTable("smartdata.bronze.marketing_raw")
```

Tablas generadas:

smartdata.bronze.marketing_raw

smartdata.bronze.ecommerce_raw

🥈 5. Capa SILVER – Limpieza y Estandarización

En esta etapa se realizan transformaciones para mejorar la calidad del dato.

Acciones principales:

Conversión de tipos: int, double, date

Corrección de formatos

Estandarización de columnas

Preparación para capa Gold

Código ejemplo Silver:

```
df_marketing_silver = (
  df_marketing
  .withColumn("Dt_Customer", F.to_date("Dt_Customer", "yyyy-MM-dd"))
  .withColumn("Income", F.col("Income").cast("int"))
)

df_marketing_silver.write.format("delta").mode("overwrite") \
   .saveAsTable("smartdata.silver.marketing_campaign_silver")
```

🥇 6. Capa GOLD – Curación y Enriquecimiento

La capa GOLD contiene la versión “curada” y enriquecida del dato.

Acciones realizadas:

Cálculo de nuevas métricas:

Age

Total_Spend

Net_Price

Revenue

Generación de tabla final unificada para análisis:

smartdata.gold.sales_marketing_gold

Código ejemplo Gold:

```
df_gold = df_ecom_silver.crossJoin(df_mkt_silver)
df_gold = df_gold.withColumn("row_id", F.monotonically_increasing_id())

df_gold.write.format("delta").mode("overwrite") \
   .saveAsTable("smartdata.gold.sales_marketing_gold")
```

🗂️ 7. Unity Catalog – Gobernanza del Proyecto

Se organizó el catálogo:

Catálogo: smartdata

Schemas:

bronze

silver

gold

Tablas creadas:

marketing_raw

ecommerce_raw

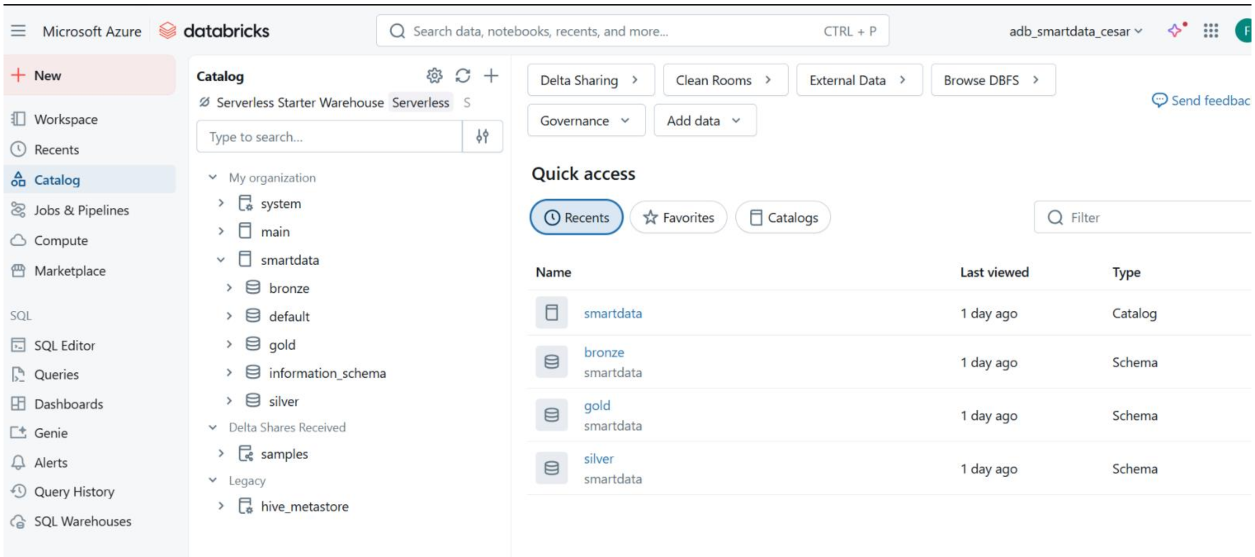
marketing_campaign_silver

ecommerce_silver

marketing_campaign_gold

ecommerce_gold

sales_marketing_gold



Name	Last viewed	Type
smartdata	1 day ago	Catalog
bronze smartdata	1 day ago	Schema
gold smartdata	1 day ago	Schema
silver smartdata	1 day ago	Schema

⚙️ 8. Databricks Workflow (Job)

Se implementó un job llamado:

🔧 job_medallion_etl_full

Este job ejecuta automáticamente las tres capas del ETL.

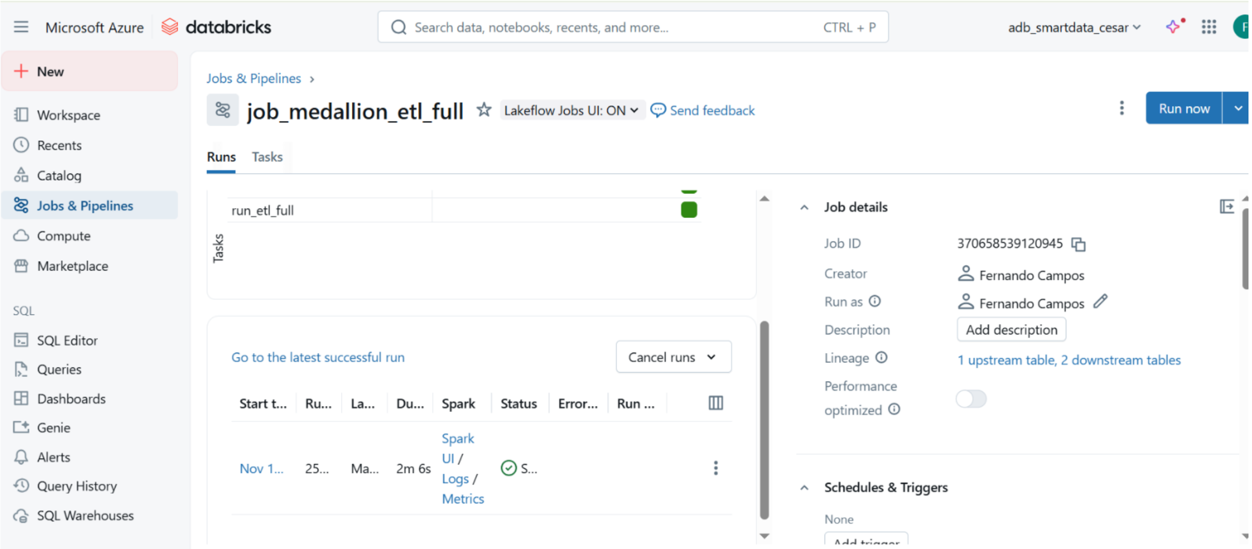
Funcionalidades:

Automatiza el pipeline

Orquesta Bronze → Silver → Gold

Permite ejecución manual o programada

Conecta al cluster de Databricks



job_medallion_etl_full

run_etl_full

Start t...	Ru...	La...	Du...	Spark	Status	Error...	Run ...
Nov 1...	25...	Ma...	2m 6s	Spark UI / Logs / Metrics	✓ S...		

Job details

Job ID	370658539120945
Creator	Fernando Campos
Run as	Fernando Campos
Description	Add description
Lineage	1 upstream table, 2 downstream tables
Performance optimized	○

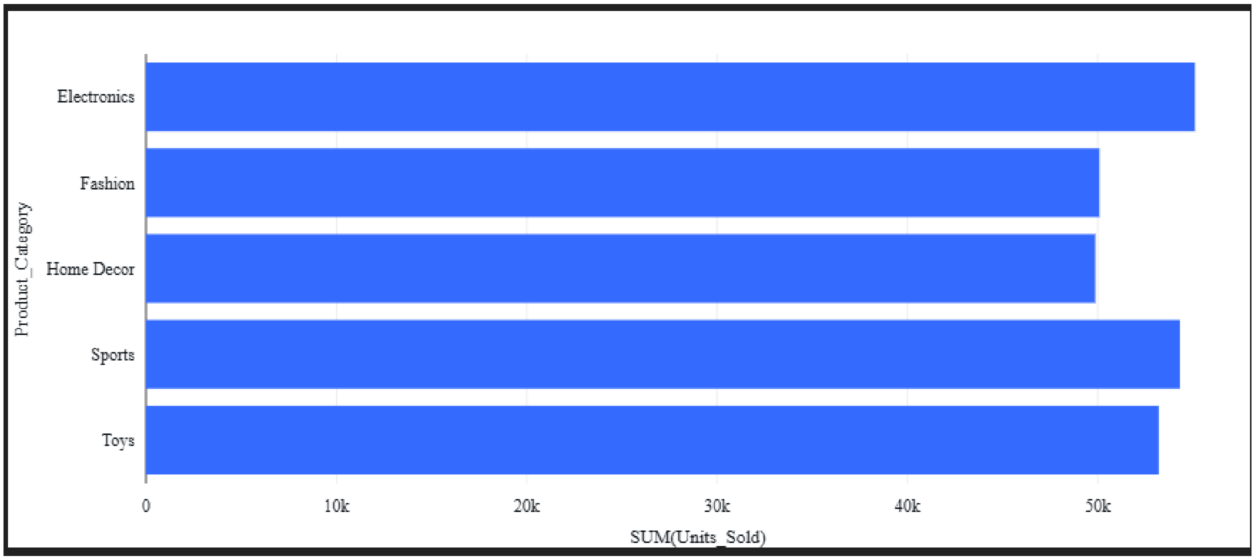
Schedules & Triggers

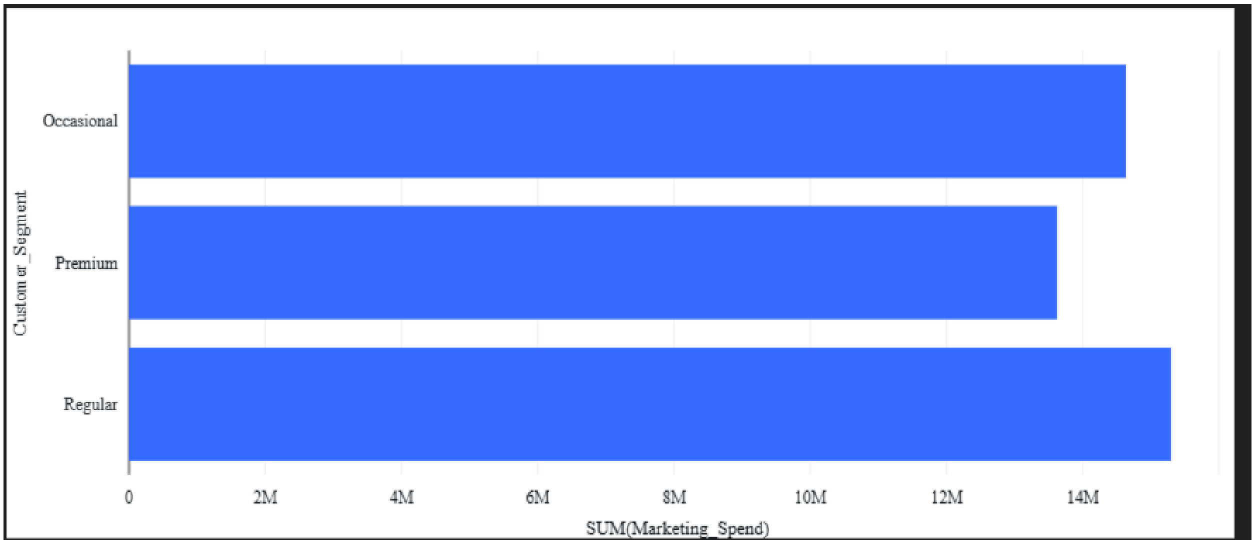
None

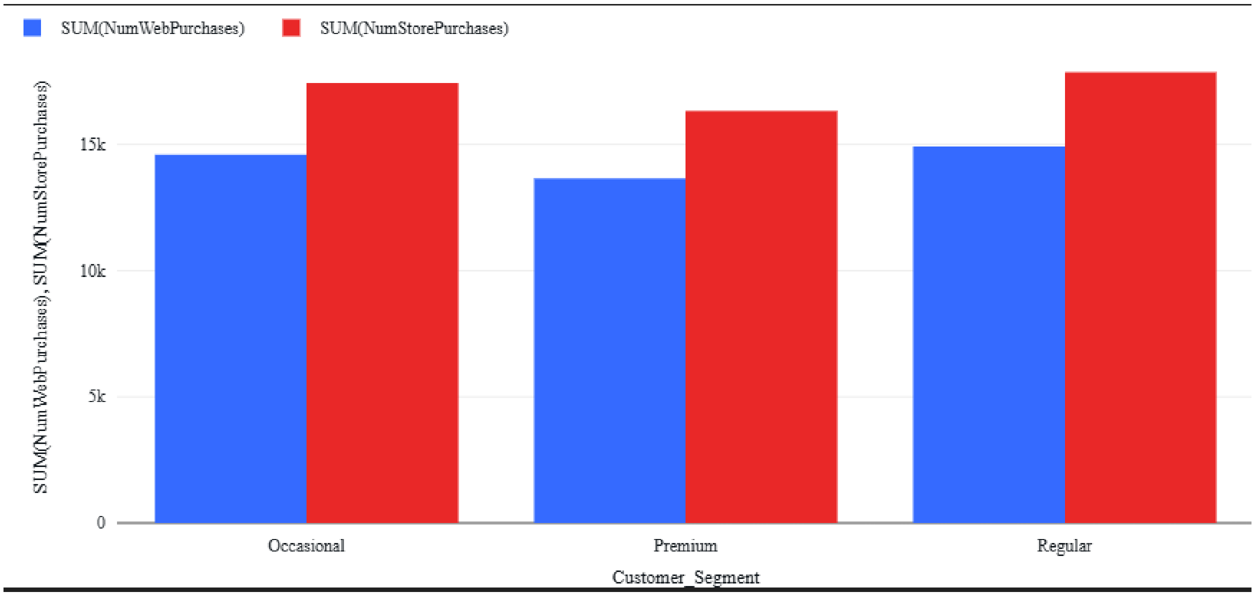
📊 9. Dashboards – Análisis Final

Se generaron 3 visualizaciones desde la tabla Gold:

- ✅ Ventas por Categoría
- ✅ Marketing Spend por Segmento
- ✅ Compras Web vs Tienda


Product_Category
Electronics
Fashion
Home Decor
Sports
Toys
0
10k
20k
30k
40k
50k
SUM(Units_Sold)


Custom er_Segment
Occasional
Premium
Regular
0
2M
4M
6M
8M
10M
12M
14M
SUM(Marketing_Spend)



🗒️ 10. Conclusiones Finales

La arquitectura Medallion permite un pipeline organizado, limpio y escalable.

La separación en capas Bronze/Silver/Gold facilitó el mantenimiento y comprensión del flujo.

El uso de Delta Lake garantizó confiabilidad, versionamiento y eficiencia en las tablas.

Unity Catalog brindó gobernanza, control de acceso y orden en los datos.

El workflow permitió automatizar por completo el ETL.

Los dashboards generados demostraron el valor final del pipeline analítico.