



Pipeline de formato medallón en Databricks

Proyecto: ETL para la empresa TechMart

Daniel Morales López – Data Engineer

2025

Introducción del Proyecto y de la Empresa

El presente proyecto titulado “**Pipeline de Formato Medallón en Databricks**” tiene como objetivo diseñar e implementar un flujo de datos escalable que permita a la empresa TechMart consolidar, transformar y analizar su información de ventas de manera eficiente. La solución se basa en la arquitectura Medallion (**Bronze → Silver → Gold**), utilizando Apache Spark en Databricks como plataforma principal. El pipeline integra los archivos de ventas generados por cada sucursal, aplica procesos de limpieza y estandarización, y finalmente produce métricas de negocio consolidadas que sirven como base para la toma de decisiones estratégicas. Además, se desarrolló un notebook con visualizaciones gráficas a partir de la capa Gold, lo que facilita la interpretación de resultados y la generación de reportes gerenciales en tiempo real. El proyecto se ejecuta mediante un Databricks Job que automatiza las tres etapas del pipeline, garantizando trazabilidad, escalabilidad y eficiencia en el manejo de datos.

TechMart es una empresa de e-commerce que opera en Costa Rica, con 12 sucursales distribuidas en diferentes ciudades del país. Su portafolio de productos abarca cinco categorías principales: **Electrónica, Ropa, Hogar, Deportes y Alimentos**. Durante el año 2024, TechMart experimentó un crecimiento significativo en sus ventas, especialmente en fechas clave como Black Friday, Cyber Monday y la temporada navideña. Sin embargo, la compañía enfrenta un reto crítico: cada sucursal gestiona sus datos de ventas de manera independiente, lo que dificulta obtener una visión unificada del negocio. Actualmente, los reportes gerenciales se generan de forma manual a partir de múltiples archivos CSV, un proceso que consume varios días de trabajo y limita la capacidad de reacción de la gerencia. Esta situación provoca que las decisiones estratégicas se basen en información desactualizada y fragmentada, lo que afecta la competitividad de la empresa en un mercado dinámico. Con la implementación del pipeline de datos en Databricks, TechMart busca superar estas limitaciones, logrando visibilidad consolidada del negocio, comparativas entre sucursales, análisis de productos más vendidos, identificación de tendencias temporales y generación de KPIs clave como ventas totales y ticket promedio.

Objetivos del Proyecto

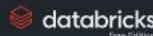
El objetivo principal del proyecto es **diseñar e implementar un pipeline de datos escalable en Databricks utilizando Apache Spark**, que permita a TechMart integrar, transformar y analizar la información de ventas proveniente de sus 12 sucursales de manera unificada. Con esta solución se busca superar las limitaciones actuales en la gestión de datos y habilitar una toma de decisiones estratégicas basada en información confiable y actualizada.

De manera específica, el proyecto persigue los siguientes objetivos:

- **Integrar los archivos de ventas diarios** generados por cada sucursal en un repositorio centralizado.
- **Estandarizar y limpiar los datos** aplicando las mejores prácticas de calidad y consistencia.
- **Transformar la información en métricas de negocio consolidadas**, incluyendo indicadores clave como ventas totales, ticket promedio y productos más vendidos.
- **Facilitar comparativas entre sucursales** para identificar patrones de rendimiento y oportunidades de mejora.
- **Habilitar el análisis de tendencias temporales**, como variaciones mensuales o estacionales en el comportamiento de compra.
- **Automatizar el proceso de integración y reportería** mediante la ejecución de un Job en Databricks que gestione las tres capas de la arquitectura Medallion (Bronze, Silver y Gold).
- **Proveer visualizaciones gráficas interactivas** basadas en la capa Gold, que permitan a la gerencia interpretar resultados y generar reportes en tiempo real.

En conjunto, estos objetivos garantizan que TechMart pueda contar con un sistema de datos moderno, confiable y escalable, capaz de transformar información dispersa en conocimiento estratégico para impulsar su crecimiento en el mercado costarricense.

Documentación Técnica del Pipeline

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace   

+ New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Nota sobre la creación del clúster

En este proyecto no se incluye el paso de creación de clúster en Databricks, en esta versión proporciona un **clúster preconfigurado por defecto**, el cual se activa automáticamente al ejecutar los notebooks. Por esta razón, no es posible crear clústeres adicionales ni personalizar su configuración, y el pipeline se desarrolló directamente sobre el clúster disponible

Features

Models

Serving

Welcome to Databricks

Send feedback

Intro to Databricks

Walk through all the features that Databricks Free Edition has to offer in 10 minutes.



Connect your data

Upload data 

Browse public datasets 

Connect to 50+ data sources 

Take a course


Data analyst
SQL & Analytics on Databricks
Beginner | Free


Data engineer
Data Engineering on Databricks
Beginner | Free


Generative AI engineer
Generative AI on Databricks
Beginner | Free

View all

Para iniciar ingresamos a Databricks, por motivos de este proyecto estemos usando la versión gratuita de Databricks llamada Databricks Community Edition, Free Edition

The screenshot shows the Databricks interface with the sidebar open. The 'Catalog' item in the sidebar is highlighted with a yellow box and circled with a yellow number 1. At the top right, the 'Add data' button is highlighted with a yellow box and circled with a yellow number 2. The main area displays a list of catalog items with columns for Name, Last viewed, and Type.

Name	Last viewed	Type
system	8 minutes ago	Catalog
workspace	8 minutes ago	Catalog
rango_experiencia workspace.default	4 days ago	Table
datos_nulos workspace.default	4 days ago	Table
salario_departamento workspace.default	4 days ago	Table
salarios workspace.default	4 days ago	Table
default workspace	4 days ago	Schema

Empezaremos creando nuestro
Catalog en Databricks

 **databricks**
Free Edition

workspace D

New

Catalog

Serverless Starter Warehouse Serverless 2XS

Type to search...

My organization

- > workspace
- > system

Delta Shares Received

- > samples

Quick access

Recents (selected)

Favorites

Catalogs

Add data

Ingest via partner

Upload to volume

Create a catalog

Create an external location

Create a volume (highlighted)

Create a credential

Create a connection

Filter

Name	Last viewed	Type
system	8 minutes ago	Catalog
workspace	8 minutes ago	Catalog
rango_experiencia workspace.default	4 days ago	Table
datos_nulos workspace.default	4 days ago	Table
salario_departamento workspace.default	4 days ago	Table
salarios workspace.default	4 days ago	Table
default workspace	4 days ago	Schema

The screenshot shows the Databricks interface for creating a new catalog. The left sidebar is visible with various navigation options like Home, Workspace, Recents, Catalog, and Jobs & Pipelines. The main area shows a 'Catalog' view for 'Serverless Starter Warehouse' with tabs for Delta Sharing, External Data, Governed Tags, and Add data. A search bar at the top says 'Search data, notebooks, recents, and more...' and has a 'CTRL + P' keyboard shortcut. On the right, there's a 'Quick access' section with Recents, Favorites, and Catalogs, and a 'Filter' button. The central part of the screen displays a modal titled 'Create a new catalog'. Inside the modal, the 'Catalog name*' field contains 'volumen_ventas' (highlighted with a yellow box and circled with a yellow number 1). Below it, the 'Type*' dropdown is set to 'Standard' (highlighted with a yellow box and circled with a yellow number 2). Under 'Storage location', there's a checkbox for 'Use default storage' which is checked (highlighted with a yellow box and circled with a yellow number 2). At the bottom right of the modal is a 'Create' button (highlighted with a yellow box and circled with a yellow number 3). To the right of the modal, a list of recently viewed items is shown, including catalogs and tables, with their last viewed times and types.

Le ponemos primero el nombre a nuestro Catalog, después por motivos de este proyecto usaremos el almacenamiento default que nos da Databricks y ya luego por últimos lo creamos el Catalog

Databricks Free Edition

workspace D

New

Catalog

Serverless Starter Warehouse Serverless 2XS

Type to search...

My organization

- workspace
- system

volumen_ventas

Delta Shares Received

- samples

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Search bar: Search data, notebooks, recents, and more... CTRL + P

Catalog Explorer >

volumen_ventas

Overview Details Permissions Policies Workspaces

Description

AI generate Add

About this catalog

Owner dmorales@ucenfotec.ac.cr

Tags Add tags

Policies New policy

volumen_ventas

Owner	Created at
dmorales@ucenfotec.ac.cr	Dec 01, 2025, 06:25 PM
information_schema	System user
dmorales@ucenfotec.ac.cr	Dec 01, 2025, 06:25 PM

Ya tendríamos creado nuestro Catalog

The screenshot shows the Databricks interface with the Catalog tab selected in the sidebar. The main area displays the Catalog Explorer for the 'volumen_ventas' catalog under the 'default' schema. A table named 'ventas_desamparados' is visible in the list. On the right side, there is a 'Create' button (marked with a yellow circle and number 1) and a 'Table' option in the dropdown menu (marked with a yellow circle and number 2).

Catalog

Serverless Starter Warehouse Serverless 2XS

Type to search...

My organization

- workspace
- system
- volumen_ventas
 - default
 - information_schema
- Delta Shares Received
- samples

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Catalog Explorer > volumen_ventas >

default ⚙️ ⚡ ⚪

Overview Details Permissions Policies

Description auto-created

Default schema (auto-created)

Filter tables Tables 4 Volumes 0 Models 0 Functions 0 Sort ▾

Name	Owner	Created at	Popularity
ventas_alajuela	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 06:49 PM	...
ventas_cartago	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 06:54 PM	...
ventas_curridabat	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 06:55 PM	...
ventas_desamparados	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 06:56 PM	...

Use with BI tools ⚙️ ⚡ ⚪

Volume

Table

About this... auto-created

Owner dmoralesl@ucenfotec.ac.cr

Model

Metric view

Add tags

New policy

Ahora insertaremos todos los CSV de la empresa al Catalog

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Add data > Create or modify table from file upload

Serverless Starter Warehouse

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

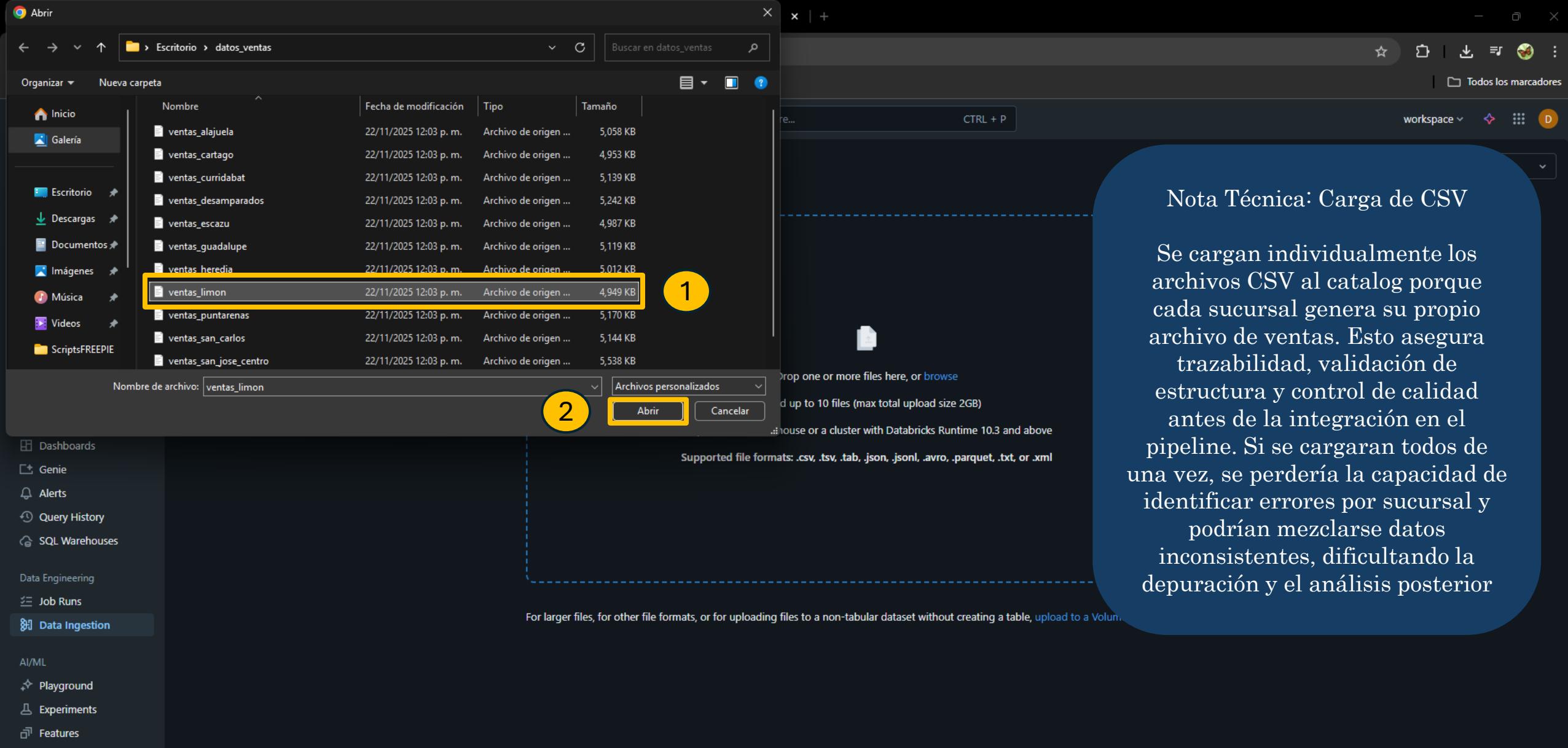
Drop one or more files here, or **browse**

Upload up to 10 files (max total upload size 2GB)

Requires a SQL warehouse or a cluster with Databricks Runtime 10.3 and above

Supported file formats: .csv, .tsv, .tab, .json, .jsonl, .avro, .parquet, .txt, or .xml

For larger files, for other file formats, or for uploading files to a non-tabular dataset without creating a table, upload to a Volume in Unity Catalog.



Ahora cargaremos cada CSV por separado en el Catalog

databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Add data >

Create or modify table from file upload

ventas_limon.csv uploaded 5.07MB X

Create new table

Preview mode Catalog Schema Table name Advanced attributes

volumen_ventas default ventas_limon

Previews 50 rows, 10 columns

id_venta	fecha	sucursal	producto	categoria	cantidad	.00	precio_unitario	.00	total	clien
V202400000008	2024-12-26T00:00:00.000Z	Limón	Nueces Mix 500g	Alimentos	3	22.34		67.02		C30871
V202400000017	2024-03-22T00:00:00.000Z	Limón	iPhone 14	Electrónica	1	929.64		929.64		C56333
V202400000024	2024-03-06T00:00:00.000Z	Limón	Monitor LG 27"	Electrónica	1	314.9		314.9		C63296
V202400000029	2024-05-22T00:00:00.000Z	Limón	Almohadas x2	Hogar	2	35.1		70.2		C64624
V202400000031	2024-02-19T00:00:00.000Z	Limón	Botella Térmica	Deportes	2	23.45		46.9		C57498
V202400000032	2024-01-28T00:00:00.000Z	Limón	Botella Térmica	Deportes	1	24.84		24.84		C86649
V202400000051	2024-03-07T00:00:00.000Z	Limón	Juego de Sábanas	Hogar	4	40.53		162.12		C35335
V202400000064	2024-10-02T00:00:00.000Z	Limón	Bicicleta MTB	Deportes	1	451.79		451.79		C95297
V202400000067	2024-08-03T00:00:00.000Z	Limón	Reloj de Pared	Hogar	1	29.61		29.61		C50695
V202400000069	2024-11-06T00:00:00.000Z	Limón	Pantalón Deportivo	Ropa	1	58.14		58.14		C43237
V202400000074	2024-03-16T00:00:00.000Z	Limón	Bolso Deportivo	Deportes	3	41.16		123.48		C91128
V202400000076	2024-08-25T00:00:00.000Z	Limón	Mouse Gaming	Electrónica	2	73.21		146.42		C70616
V202400000078	2024-05-06T00:00:00.000Z	Limón	Zapatos Formales	Ropa	1	94.05		94.05		C37347
V202400000087	2024-02-12T00:00:00.000Z	Limón	Galletas Integrales	Alimentos	2	7.23		14.46		C57179
V202400000114	2024-10-22T00:00:00.000Z	Limón	Teclado Mecánico	Electrónica	3	128.79		386.37		C90974
V202400000118	2024-04-16T00:00:00.000Z	Limón	Aceite Oliva 1L	Alimentos	1	12.07		12.07		C98650

[Cancel](#) [Create table](#)

Hacemos el mismo paso con cada uno de los CSV de la Empresa

The screenshot shows the Databricks interface with the Catalog sidebar open. The main area displays the 'volumen_ventas' catalog overview, listing two schemas: 'default' and 'information_schema'. The 'default' schema contains 11 tables: ventas_alajuela, ventas_cartago, ventas_curridabat, ventas_desamparados, ventas_escazu, ventas_guadalupe, ventas_heredia, ventas_lemon, ventas_puntarenas, ventas_san_carlos, ventas_san_jose_centro, and ventas_san_pedro. A yellow box highlights this list of tables.

Catalog Explorer > **volumen_ventas**

Overview Details Permissions Policies Workspaces

Description

AI generate Add

Filter schemas 2 schemas

Name	Owner	Created at
default	dmorales@ucenfotec.ac.cr	Dec 01, 2025, 06:25 PM
information_schema	System user	Dec 01, 2025, 06:25 PM

About this catalog

Owner dmorales@ucenfotec.ac.cr

Tags Add tags

Policies New policy

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Ya tendríamos cargados todos los CSV en el Catalog

The screenshot shows the Databricks workspace interface. On the left, there is a dark sidebar with various navigation options: Home, Workspace (highlighted with a yellow circle labeled '1'), Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL (with sub-options: SQL Editor, Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses), Data Engineering (with sub-options: Job Runs and Data Ingestion), and AI/ML (with sub-options: Playground, Experiments, Features, Models, and Serving). The main area is titled 'Workspace' and shows a list of workspaces under the user 'dmoralesl@ucenfotec.ac.cr'. The 'Workspace' option in the sidebar is also highlighted with a yellow circle labeled '2'. The list includes:

Name	Type	Owner	Created at
Bakehouse Sales Starter Space	Genie space	dmoralesl@ucenfotec.ac.cr	Nov 26, 2025, 07:46 PM
ventas_alajuela 2025-12-01 19:04:43	Dashboard	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 07:04 PM
Workspace Usage Dashboard	Dashboard	dmoralesl@ucenfotec.ac.cr	Nov 26, 2025, 07:46 PM

At the top right, there are links for 'Send feedback', 'Share', and 'Create'. The top bar also features a search bar and a 'CTRL + P' shortcut.

Ahora procederemos a crear un folder para guardar nuestros notebooks

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace > Users >
dmoralesl@ucenfotec.ac.cr ☆

Send feedback Share Create

1 Create > Folder

2 File

Name

Type

Owner

Created at

Name	Type	Owner	Created at
lakehouse Sales Starter Space	Genie space	dmoralesl@ucenfotec.ac.cr	Nov 26, 2025, 07:46 PM
Entas_alajuela 2025-12-01 19:04:43	Dashboard	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 07:04 PM
Workshopage Dashboard	Dashboard	dmoralesl@ucenfotec.ac.cr	Nov 26, 2025, 07:46 PM

The screenshot shows the Databricks workspace interface. On the left sidebar, under the 'Workspace' section, there is a 'Create' button (circled with number 1) and a 'Notebook' option (circled with number 2). The main area displays a 'Project' workspace with a search bar and filters for 'Type', 'Owner', and 'Last modified'. A folder icon is shown with the message 'This folder is empty'. The right sidebar lists various creation options: Folder, Git folder, Notebook (highlighted with a yellow box), File, Query, Dashboard, Genie space, ETL Pipeline, Legacy Alert, Alert Preview, and MLflow experiment.

Ya cuando creamos nuestro folder empezaremos a crear nuestros Notebooks, empezamos primero por el Notebook de la etapa Bronze

Etap a Bronze

The screenshot shows the Databricks workspace interface. On the left, the sidebar contains various navigation links such as Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL, SQL Editor, Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses. Below these are sections for Data Engineering (Job Runs, Data Ingestion), AI/ML (Playground, Experiments, Features, Models, Serving), and a New button.

The main area displays a notebook titled "01_bronze_ingestion". The notebook's sidebar shows it is a Python notebook. The code in the notebook is as follows:

```
# Importamos la función para obtener la fecha/hora actual
from pyspark.sql.functions import current_timestamp

# =====
# 1. Lectura y consolidación de datos desde múltiples sucursales
# =====
# Usamos Spark SQL para leer las tablas de ventas de cada sucursal
# y unificarlas en un solo DataFrame con UNION ALL.
# Además, añadimos la columna 'archivo_fuente' usando _metadata.file_path
# para mantener trazabilidad del archivo origen de cada registro.

df_TechMart = spark.sql(
    """
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_alajuela
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_cartago
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_curridabat
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_heredia
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_desamparados
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_escazu
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_guadalupe
    UNION ALL
    SELECT *, _metadata.file_path AS archivo_fuente
    FROM volumen_ventas.default.ventas_limon
    """
```

Código de la etapa Bronze (En los comentarios del código se explica el código)

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

01_bronze_ingestion

Last edit was 7 days ago

1

Python

Run all

Serverless

Schedule

Share

Dec 01, 2025 (7s)

```
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_alajuela
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_cartago
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_curridabat
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_heredia
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_desamparados
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_escazu
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_guadalupe
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_lemon
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_san_carlos
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_puntarenas
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_san_jose_centro
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_san_pedro
"""
)
```

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

01_bronze_ingestion

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Dec 01, 2025 (7s)

```
FROM volumen_ventas.default.ventas_san_carlos
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_puntarenas
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_san_jose_centro
UNION ALL
SELECT *, _metadata.file_path AS archivo_fuente
FROM volumen_ventas.default.ventas_san_pedro
"""

)

# =====#
# 2. Añadir columna de auditoría
# =====#
# Insertamos la columna 'fecha_ingesta' con la marca de tiempo
# en que se realiza la ingesta, para trazabilidad y auditoría.

df_TechMart = df_TechMart.withColumn("fecha_ingesta", current_timestamp())

# =====#
# 3. Escritura en la capa Bronze
# =====#
# Guardamos el DataFrame consolidado en formato Delta Lake.
# Usamos modo 'append' para permitir múltiples cargas incrementales.
# La tabla resultante se llama 'DF_TechMart_Bronze_Ingestion'.

df_TechMart.write.format("delta") \
    .mode("append") \
    .saveAsTable("DF_TechMart_Bronze_Ingestion")
```

See performance (1)

df_TechMart: pyspark.sql.connect.DataFrame = [id_venta: string, fecha: timestamp ... 10 more fields]

[Shift+Enter] to run and move to next cell

Etapá Silver

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

02_silver_transformation +

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (1s)

```
# =====
# Lectura de la tabla Bronze
# =====
# Usamos spark.read.table para cargar la tabla Delta previamente creada
# en la capa Bronze. Esta tabla contiene los datos crudos consolidados
# de todas las sucursales, junto con columnas de auditoría como
# 'archivo_fuente' y 'fecha_ingesta'.

df_TechMart_SilverStage = spark.read.table("workspace.default.df_techmart_broze_ingestion")
```

df_TechMart_SilverStage: pyspark.sql.connect.dataframe.DataFrame = [id_venta: string, fecha: timestamp ... 10 more fields]

Dec 02, 2025 (<1s)

```
# =====
# Transformaciones sobre la tabla SilverStage
# =====

import pyspark.sql.functions as F

df_TechMart_SilverStage = (
    df_TechMart_SilverStage
    # 1. Convertir la columna 'fecha' a tipo timestamp
    # Se especifica el formato 'yyyy-MM-dd HH:mm:ss' para asegurar
    # que Spark interprete correctamente la fecha y hora.
    .withColumn(
        "fecha",
        F.to_timestamp("fecha", "yyyy-MM-dd HH:mm:ss")
    )
    # 2. Eliminar la columna 'fecha'
    # Aquí se elimina la columna recién creada, lo cual normalmente
    # no tendría sentido porque acabamos de transformarla.
    # (Probablemente se buscaba limpiar duplicados o redefinir la columna).
    .drop("fecha")
    # 3. Renombrar la columna 'fecha_ingesta' como 'fecha'
```

Ahora creamos el notebook de la etapa Silver, este es el código de la etapa Silver (En los comentarios del código se explica el código)

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

02_silver_transformation +

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

1

```
Dec 02, 2025 (1s)
df_TechMart_SilverStage = spark.read.table("workspace.default.dt_techmart_broze_ingestion")
df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, fecha: timestamp ... 10 more fields]
```

Python

Run all

Serverless

Schedule

Share

2

```
Dec 02, 2025 (<1s)
# =====
# Transformaciones sobre la tabla SilverStage
# =====

import pyspark.sql.functions as F

df_TechMart_SilverStage = (
    df_TechMart_SilverStage
    # 1. Convertir la columna 'fecha' a tipo timestamp
    # Se especifica el formato 'yyyy-MM-dd HH:mm:ss' para asegurar
    # que Spark interprete correctamente la fecha y hora.
    .withColumn(
        "fecha",
        F.to_timestamp("fecha", "yyyy-MM-dd HH:mm:ss")
    )
    # 2. Eliminar la columna 'fecha'
    # Aquí se elimina la columna recién creada, lo cual normalmente
    # no tendría sentido porque acabamos de transformarla.
    # (Probablemente se buscaba limpiar duplicados o redefinir la columna).
    .drop("fecha")
    # 3. Renombrar la columna 'fecha_ingesta' como 'fecha'
    # Esto reemplaza la columna de ingestión por el nombre genérico 'fecha',
    # dejando como referencia la fecha de carga en lugar de la fecha original.
    .withColumnRenamed("fecha_ingesta", "fecha")
)
```

3

```
df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 9 more fields]
```

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

02_silver_transformation

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Dec 02, 2025 (<1s)

2

```
.drop("fecha")
# 3. Renombrar la columna 'fecha_ingesta' como 'fecha'
# Esto reemplaza la columna de ingesta por el nombre genérico 'fecha',
# dejando como referencia la fecha de carga en lugar de la fecha original.
.withColumnRenamed("fecha_ingesta", "fecha")
```

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 9 more fields]

Dec 02, 2025 (<1s)

3

```
# =====
# Derivación de atributos de fecha en la capa SilverStage
# =====
```

```
df_TechMart_SilverStage = (
    df_TechMart_SilverStage
        # 1. Extraer el año desde la columna 'fecha'
        .withColumn("año", F.year("fecha"))
        # 2. Extraer el mes (valor numérico 1-12)
        .withColumn("mes", F.month("fecha"))
        # 3. Extraer el día del mes (1-31)
        .withColumn("dia", F.dayofmonth("fecha"))
        # 4. Obtener el nombre completo del día de la semana (ej. Monday, Tuesday)
        .withColumn("dia_semana", F.date_format("fecha", "EEEE"))
```

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (<1s)

4

```
# =====
# Normalización de texto en la capa SilverStage
```


 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

+ New

Workspace

Project

- 01_bronze_ingestion
- 02_silver_transformation**
- 03_gold_aggregations
- 04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (<1s) 4
.withColumn("producto", F.lower(F.col("producto")))
)

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (<1s) 5
=====
Eliminación de duplicados en la capa SilverStage
=====

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (<1s) 6
=====
Filtrado de registros inválidos en la capa SilverStage
=====

Usamos filter para conservar únicamente las filas donde la columna 'total' > 0.
Esto elimina registros con valores nulos, cero o negativos en el campo 'total',
garantizando que solo se mantengan ventas válidas en el DataFrame SilverStage.

df_TechMart_SilverStage = df_TechMart_SilverStage.filter(F.col("total") > 0)

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

02_silver_transformation +

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Dec 02, 2025 (<1s)

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (<1s)

```
# =====#
# Validación de campos obligatorios en la capa SilverStage
# =====#

# Definimos una lista con los nombres de los campos que deben ser obligatorios.
# Estos campos no pueden contener valores nulos, ya que son críticos para el análisis:
# - id_venta → identificador único de la transacción
# - fecha → fecha de la venta
# - sucursal → sucursal donde ocurrió la venta
# - producto → producto vendido
# - total → monto total de la transacción

campos_obligatorios = ["id_venta", "fecha", "sucursal", "producto", "total"]

# Iteramos sobre cada campo obligatorio y aplicamos un filtro
# para eliminar las filas donde ese campo sea nulo.
# De esta forma, garantizamos que el DataFrame SilverStage
# solo contenga registros válidos y completos.
for campo in campos_obligatorios:
    df_TechMart_SilverStage = df_TechMart_SilverStage.filter(F.col(campo).isNotNull())
```

df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (12s)

```
# =====#
# Escritura de la tabla Silver particionada
# =====#

# Guardamos el DataFrame SilverStage en formato Delta Lake.
# Se utiliza 'append' para permitir cargas incrementales sin sobrescribir datos previos.
# ... (resto del código)
```

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Workspace

Project

- 01_bronze_ingestion
- 02_silver_transformation**
- 03_gold_aggregations
- 04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (<1s) 7

```
df_TechMart_SilverStage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]
```

Dec 02, 2025 (12s) 8

```
# =====
# Escritura de la tabla Silver particionada
# =====

# Guardamos el DataFrame SilverStage en formato Delta Lake.
# Se utiliza 'append' para permitir cargas incrementales sin sobrescribir datos previos.
# La partición se realiza por la columna 'mes', lo que optimiza las consultas
# que filtran o agrupan por mes, ya que Spark puede leer solo las particiones necesarias.
# Finalmente, se registra la tabla en el catálogo con el nombre 'df_techmart_silver_partitioned'.

df_TechMart_SilverStage.write.format("delta") \
    .mode("append") \
    .partitionBy("mes") \
    .saveAsTable("df_techmart_silver_partitioned")
```

See performance (1)

[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette
[Esc H] to see all keyboard shortcuts

Optimize

Home

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Etapá Gold

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (1s) 1

```
# =====
# Lectura de la tabla Silver particionada para crear la capa Gold
# =====

# Usamos spark.read.table para cargar la tabla Delta 'df_techmart_silver_partitioned'
# que corresponde a la capa Silver ya limpia y particionada por mes.
# Esta lectura nos entrega un DataFrame Spark llamado 'df_TechMart_goldstage',
# el cual servirá como base para aplicar las transformaciones y agregaciones
# necesarias en la capa Gold (ej. métricas, tendencias, top productos).

df_TechMart_goldstage = spark.read.table("workspace.default.df_techmart_silver_partitioned")
```

df_TechMart_goldstage: pyspark.sql.connect.dataframe.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (4s) 2

```
# =====
# Cálculo de métricas globales en la capa Gold
# =====

import pyspark.sql.functions as F

# 1. Agregación de métricas principales
# - ventas_totales: suma de la columna 'total'
# - num_transacciones: conteo de registros por id_venta
# - ticket_promedio: promedio del monto por transacción
ventas_totales = (
    df_TechMart_goldstage
    .agg(
        F.sum("total").alias("ventas_totales"),
        F.count("id_venta").alias("num_transacciones"),
        (F.sum("total") / F.count("id_venta")).alias("ticket_promedio")
    )
)
# 2. Formateo de métricas monetarias
# Se convierten a string con símbolo $ y dos decimales
```

Ahora creamos el notebook de la etapa Gold, este es el código de la etapa Gold (En los comentarios del código se explica el código)

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

df_TechMart_goldstage: pyspark.sql.connect.DataFrame = [id_venta: string, sucursal: string ... 13 more fields]

Dec 02, 2025 (4s)

```
# =====
# Cálculo de métricas globales en la capa Gold
# =====

import pyspark.sql.functions as F

# 1. Agregación de métricas principales
# - ventas_totales: suma de la columna 'total'
# - num_transacciones: conteo de registros por id_venta
# - ticket_promedio: promedio del monto por transacción
ventas_totales = (
    df_TechMart_goldstage
    .agg(
        F.sum("total").alias("ventas_totales"),
        F.count("id_venta").alias("num_transacciones"),
        (F.sum("total") / F.count("id_venta")).alias("ticket_promedio")
    )
    # 2. Formateo de métricas monetarias
    # Se convierten a string con símbolo $ y dos decimales
    .withColumn("ventas_totales", F.format_string("$%.2f", F.col("ventas_totales")))
    .withColumn("ticket_promedio", F.format_string("$%.2f", F.col("ticket_promedio")))
)

# 3. Escritura en la capa Gold
# Guardamos la tabla en formato Delta Lake con modo 'append'
# para permitir cargas incrementales sin sobrescribir datos previos.
ventas_totales.write.format("delta").mode("append").saveAsTable("df_techmart_gold_ventas_totales")
```

See performance (1)

Optimize

ventas_totales: pyspark.sql.connect.DataFrame = [ventas_totales: string, num_transacciones: long ... 1 more field]

Dec 02, 2025 (3s)

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

02_silver_transformation 03_gold_aggregations +

ventas_totales: pyspark.sql.connect.DataFrame = [ventas_totales: string, num_transacciones: long ... 1 more field]

Dec 02, 2025 (3s) 3

```
# =====
# Cálculo de métricas por sucursal en la capa Gold
# =====

ventas_por_sucursal = (
    df_TechMart_goldstage
    # 1. Agrupamos por la columna 'sucursal'
    .groupBy("sucursal")
    # 2. Calculamos métricas principales por cada sucursal:
    # - ventas_totales: suma de la columna 'total'
    # - num_transacciones: conteo de registros por id_venta
    # - ticket_promedio: promedio del monto por transacción
    .agg(
        F.sum("total").alias("ventas_totales"),
        F.count("id_venta").alias("num_transacciones"),
        (F.sum("total") / F.count("id_venta")).alias("ticket_promedio")
    )
    # 3. Formateamos las métricas monetarias
    # Se convierten a string con símbolo $ y dos decimales
    .withColumn("ventas_totales", F.format_string("%.2f", F.col("ventas_totales")))
    .withColumn("ticket_promedio", F.format_string("%.2f", F.col("ticket_promedio")))
)

# 4. Escritura en la capa Gold
# Guardamos la tabla en formato Delta Lake con modo 'append'
# para permitir cargas incrementales sin sobrescribir datos previos.
ventas_por_sucursal.write.format("delta").mode("append").saveAsTable("df_techmart_gold_ventas_por_sucursal")
```

See performance (1)

Optimize

ventas_por_sucursal: pyspark.sql.connect.DataFrame = [sucursal: string, ventas_totales: string ... 2 more fields]

Dec 02, 2025 (4s) 4

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Dec 02, 2025 (3s) 3

See performance (1)

Optimize

ventas_por_sucursal: pyspark.sql.connect.DataFrame = [sucursal: string, ventas_totales: string ... 2 more fields]

Dec 02, 2025 (4s) 4

```
# =====#
# Cálculo del Top 10 productos más vendidos en la capa Gold
# =====#

top_productos = (
    df_TechMart_goldstage
    # 1. Agrupamos por la columna 'producto'
    .groupBy("producto")
    # 2. Calculamos la cantidad total vendida por producto
    .agg(F.sum("cantidad").alias("cantidad_vendida"))
    # 3. Ordenamos los productos de mayor a menor según la cantidad vendida
    .orderBy(F.desc("cantidad_vendida"))
    # 4. Limitamos el resultado a los 10 primeros productos
    .limit(10)
)

# 5. Escritura en la capa Gold
# Guardamos la tabla en formato Delta Lake con modo 'append'
# para permitir cargas incrementales sin sobrescribir datos previos.
top_productos.write.format("delta").mode("append").saveAsTable("df_techmart_gold_top_productos")
```

See performance (1)

Optimize

top_productos: pyspark.sql.connect.DataFrame = [producto: string, cantidad_vendida: long]

Dec 02, 2025 (4s) 5

```
# =====#
# Cálculo de métricas por categoría en la capa Gold
```

 **databricks**
Free Edition

workspace D

New

Workspace ▾

- Home
- Workspace
- Recents
- Catalog
- Jobs & Pipelines
- Compute
- Marketplace

SQL

- SQL Editor
- Queries
- Dashboards
- Genie
- Alerts
- Query History
- SQL Warehouses

Data Engineering

- Job Runs
- Data Ingestion

AI/ML

- Playground
- Experiments
- Features
- Models
- Serving

File Edit View Run Help Python ▾ Tabs: ON ▾ Last edit was 7 days ago

CTRL + P

02_silver_transformation 03_gold_aggregations +

Dec 02, 2025 (4s) See performance (1) top_productos: pyspark.sql.connect.DataFrame = [producto: string, cantidad_vendida: long] Optimize

Dec 02, 2025 (4s) 5

```
# =====
# Cálculo de métricas por categoría en la capa Gold
# =====

ventas_por_categoria = (
    df_TechMart_goldstage
    # 1. Agrupamos por la columna 'categoria'
    .groupBy("categoria")
    # 2. Calculamos métricas principales por cada categoría:
    # - ventas_totales: suma de la columna 'total'
    # - productos_vendidos: suma de la columna 'cantidad'
    .agg(
        F.sum("total").alias("ventas_totales"),
        F.sum("cantidad").alias("productos_vendidos")
    )
    # 3. Formateamos las métricas monetarias
    # Se convierten a string con símbolo $ y dos decimales
    .withColumn("ventas_totales", F.format_string("%.2f", F.col("ventas_totales")))
)

# 4. Escritura en la capa Gold
# Guardamos la tabla en formato Delta Lake con modo 'overwrite'
# para reemplazar completamente los datos previos con los nuevos cálculos.
ventas_por_categoria.write.format("delta").mode("overwrite").saveAsTable("df_techmart_gold_ventas_por_categoria")
```

See performance (1) ventas_por_categoria: pyspark.sql.connect.DataFrame = [categoria: string, ventas_totales: string ... 1 more field] Optimize

Dec 02, 2025 (4s) 6

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

02_silver_transformation

03_gold_aggregations

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (4s)

Cálculo de la tendencia mensual de ventas en la capa Gold

```
tendencia_mensual = (
    df_TechMart_goldstage
    # 1. Agrupamos por año y mes
    .groupBy("año", "mes")
    # 2. Calculamos métricas principales por cada período:
    # - ventas_totales: suma de la columna 'total'
    # - num_transacciones: conteo de registros por id_venta
    # - productos_vendidos: suma de la columna 'cantidad'
    .agg(
        F.sum("total").alias("ventas_totales"),
        F.count("id_venta").alias("num_transacciones"),
        F.sum("cantidad").alias("productos_vendidos")
    )
    # 3. Ordenamos cronológicamente por año y mes
    .orderBy("año", "mes")
    # 4. Formateamos las métricas monetarias
    # Se convierten a string con símbolo $ y dos decimales
    .withColumn("ventas_totales", F.format_string("%.2f", F.col("ventas_totales")))
)
```

5. Escritura en la capa Gold
Guardamos la tabla en formato Delta Lake con modo 'overwrite'
para reemplazar completamente los datos previos con los nuevos cálculos.
tendencia_mensual.write.format("delta").mode("overwrite").saveAsTable("df_techmart_gold_tendencia_mensual")

See performance (1)

tendencia_mensual: pyspark.sql.connect.DataFrame = [año: integer, mes: integer ... 3 more fields]

[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette

Creación y Ejecución del Job

Databricks Free Edition workspace D

New Home Workspace Recents Catalog Jobs & Pipelines Compute Marketplace SQL SQL Editor Queries Dashboards Genie Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion AI/ML Playground Experiments Features Models Serving

Search data, notebooks, recents, and more... CTRL + P

Catalog Serverless Starter Warehouse Serverless 2XS Type to search... For you All My organization workspace default datos_nulos df_techmart_broze_ingestion df_techmart_gold_tendencia_mensual df_techmart_gold_top_productos df_techmart_gold_ventas_por_categoria df_techmart_gold_ventas_por_sucursal df_techmart_gold_ventas_totales df_techmart_silver_partitioned empleado_mejor_pagado mayor_gasto_por_departamento rango_experiencia salario_anual salario_departamento salarios information_schema system volumen_ventas Delta Shares Received samples

Delta Sharing > External Data > Governed Tags > Add data >

Send feedback

Quick access Recents Favorites Catalogs Filter

Name	Last viewed	Type
mayor_gasto_por_departamento workspace.default	6 days ago	Table
empleado_mejor_pagado workspace.default	6 days ago	Table
df_techmart_silver_partitioned workspace.default	6 days ago	Table
df_techmart_gold_ventas_totales workspace.default	6 days ago	Table
df_techmart_gold_ventas_por_sucursal workspace.default	6 days ago	Table
df_techmart_gold_ventas_por_categoria workspace.default	6 days ago	Table
df_techmart_gold_top_productos workspace.default	6 days ago	Table
df_techmart_gold_tendencia_mensual workspace.default	6 days ago	Table
df_techmart_broze_ingestion workspace.default	6 days ago	Table
default workspace	6 days ago	Schema
workspace workspace	6 days ago	Catalog
salarios workspace.default	7 days ago	Table

Una vez cargadas todas las etapas en formato Delta Lake dentro del catalog, el siguiente paso corresponde a la creación y ejecución del Job que orquesta el pipeline

workspace  

+ New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Create new

Ingestion pipeline

ETL pipeline

Job

Send feedback

Jobs & pipelines Job runs

Filter by name or ID substring

All Jobs Pipelines Owned by me Accessible by me Favorites Tags Run as Create

Name	Type	Tags	Run as	Trigger	Recent runs
New Job Nov 26, 2025, 09:54 PM	Job		dmoreasl@ucenfotec.ac.cr	— — — ○ ○	< Previous Next >

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Nos dirigimos a la sección de “Jobs y Pipelines”

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Jobs & Pipelines

Create new

-  **Ingestion pipeline**
Ingest data from apps, databases and files
-  **ETL pipeline**
Build ETL pipelines using SQL and Python
-  **Job**
Orchestrate notebooks, pipelines, queries and more

Jobs & pipelines **Job runs**

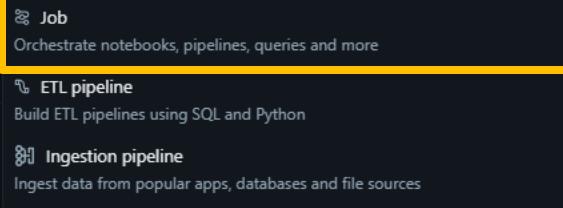
Filter by name or ID substring

All Jobs Pipelines Owned by me Accessible by me Favorites Tags Run as

Create

1 

Name	Type	Tags	Run as
New Job Nov 26, 2025, 09:54 PM	 Job		 dmorales@ucenfotec.ac.cr

2 
 **Job**
Orchestrate notebooks, pipelines, queries and more

 **ETL pipeline**
Build ETL pipelines using SQL and Python

 **Ingestion pipeline**
Ingest data from popular apps, databases and file sources

3 

4 

Creamos nuestro Job

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM Lakeflow Jobs UI: ON

Runs **Tasks**

Add your first task
Choose from your recently used ones...

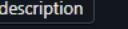
 **Notebook**
Run a notebook

or

+ Add another task type

Job details

Job ID: 169813361027417 
Creator: dmoralesl@ucenfotec.ac.cr
Run as: dmoralesl@ucenfotec.ac.cr 
Description: 
Lineage: 
Performance optimized: 

Schedules & Triggers

None 

Job parameters 

No job parameters are defined for this job 

Tags 



Job notifications 

Run now 

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

Home Workspace Recents Catalog Jobs & Pipelines Compute Marketplace SQL SQL Editor Queries Dashboards Genie Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion AI/ML Playground Experiments Features Models Serving

Unnamed task
Unspecified path

Task name* (Input field)

Type* Notebook (Dropdown)

Source* Workspace (Dropdown)

Path* (Input field) Select Notebook (Dropdown) **Path*** is highlighted with a yellow box.

Compute* Serverless (Dropdown) Autoscaling

Environment and Libraries* Select notebook to configure environment

Parameters (UI JSON) Key Value + Add

Cancel Create task

Job details

- Job ID: 169813361027417
- Creator: dmoralesl@ucenfotec.ac.cr
- Run as: dmoralesl@ucenfotec.ac.cr
- Description: Add description
- Lineage: No lineage information for this job. Learn more
- Performance optimized: Off

Schedules & Triggers

- None
- Add trigger

Job parameters

- No job parameters are defined for this job
- Edit parameters

Tags

- Add tag

Job notifications

Run now

Seleccionamos el notebook de la etapa Bronze para esta primer Task

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

Select Notebook

Workspace Recents

1

2

3

Job details

Job ID: 169813361027417

Creator: dmoralesl@ucenfotec.ac.cr

Run as: dmoralesl@ucenfotec.ac.cr

Description: Add description

Lineage: No lineage information for this job. Learn more

Performance optimized: Off

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Tags

Add tag

Job notifications

Run now

Repos Shared Users Laboratorio Proyect

01_bronze_ingestion 02_silver_transformation 03_gold_aggregations 04_analisis_resultados

Task name* Type* Source* Path* Compute* Environment and Libraries* Parameters* Key Value

Cancel Confirm

Cancel Create task

Databricks Free Edition workspace D

Search data, notebooks, recents, and more... CTRL + P

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion ...rkspace/Proyect/01_bronze_ingestion

Job details

- Job ID: 169813361027417
- Creator: dmorales@ucenfotec.ac.cr
- Run as: dmorales@ucenfotec.ac.cr
- Description: Add description
- Lineage: No lineage information for this job. Learn more
- Performance optimized: Off

Schedules & Triggers

- None
- Add trigger

Job parameters

- No job parameters are defined for this job
- Edit parameters

Tags

- Add tag

Job notifications

1

2

Task name* 01_bronze_ingestion

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/01_bronze_ingestion

Compute* Serverless Autoscaling

Environment and Libraries* Notebook Environment

Parameters

Key	Value

UI JSON

Cancel Create task

Le nombramos un nombre a nuestro Task y lo creamos

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion ...rkspace/Proyect/01_bronze_ingestion

+ Add task Add a new task to your job

Task name* 01_bronze_ingestion

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/01_bronze_ingestion

Compute* Serverless Autoscaling

Environment and Libraries* Notebook Environment

Edit the notebook's environment

Parameters

Key	Value

UI JSON

Cancel Save task Swap

Job details

Job ID 169813361027417

Creator dmoralesl@ucenfotec.ac.cr

Run as dmoralesl@ucenfotec.ac.cr

Description

Add description

No lineage information for this job. Learn more

Performance optimized

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Compute

Serverless

Añadimos un nuevo task a nuestro job

Añadimos un nuevo task a nuestro job

Databricks Free Edition

Search data, notebooks, recent, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion ...rkspace/Proyect/01_bronze

+ Add task

Task name* 01_bronze_ingestion

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/01_bronze_ingestion

Compute* Serverless

Environment and Libraries* Notebook Environment

Edit the notebook's environment

Parameters

Key	Value

UI JSON

Cancel Save task Swap

Add Task

search tasks

Ingestion pipeline Create a new ingestion pipeline to ingest data from SaaS and databases and more

Notebook Run a notebook

Python script Run a Python file

SQL query Run a SQL query

SQL file Run a SQL file

SQL alert Evaluate a SQL alert and notify users

Ingestion and Transformation

Run Ingestion pipeline

Job details

Job ID 169813361027417

Creator dmorales@ucenfotec.ac.cr

Run as dmorales@ucenfotec.ac.cr

Description

Add description

No lineage information for this job.

Learn more

Lineage

Performance optimized

Schedules & Triggers

None

Add trigger

Job parameters

No job parameters are defined for this job

Edit parameters

Compute

Serverless

Swap

The screenshot shows the Databricks interface for creating a new job. The 'Jobs & Pipelines' section is active. A modal window titled 'Add Task' is open, showing various task types like 'Ingestion pipeline', 'Notebook', 'Python script', etc. The 'Notebook' option is selected and highlighted with a yellow box. In the main job configuration area, the 'Type' field is set to 'Notebook'. The 'Compute' field is set to 'Serverless'. Other fields like 'Task name', 'Source', 'Path', and 'Environment and Libraries' are also visible. To the right, there are sections for 'Job details', 'Schedules & Triggers', 'Job parameters', and 'Compute' settings.

data bricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

+ New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion ...rkspace/Proyect/01_bronze_ingestion

Unnamed task Unspecified path

Task name* (1)

Type* Notebook

Source* Workspace

Path* (1) Select Notebook

Compute* (1) Serverless Autoscaling

Depends on 01_bronze_ingestion X

Run if dependencies (1) All succeeded

Environment and Libraries* (1) Select notebook to configure environment

Cancel Create task Swap

Job details

Job ID 169813361027417

Creator dmoralesl@ucenfotec.ac.cr

Run as dmoralesl@ucenfotec.ac.cr

Description Add description

Lineage (1) No lineage information for this job. Learn more

Performance optimized (1)

Schedules & Triggers

None Add trigger

Job parameters (1)

No job parameters are defined for this job Edit parameters

Compute

Serverless

Seleccionamos el notebook de la etapa Silver para este segundo Task

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON 

Runs Tasks

Select Notebook

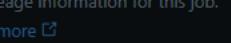
Job details

Job ID: 169813361027417 

Creator: dmorales@ucenfotec.ac.cr

Run as: dmorales@ucenfotec.ac.cr 

Description: 

Lineage: 

Performance optimized: 

Schedules & Triggers

None 

Job parameters

No job parameters are defined for this job 

Compute

Serverless 

Workspace Recents

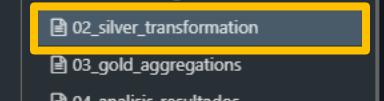
1  2 

Repos Shared Users Laboratorio 2 Proyect

01_bronze_ingestion 01_bronze_ingestion
02_silver_transformation
03_gold_aggregations
04_analisis_resultados

Task name* 
Type* 
Source* 
Path* 
Compute* 
Depends on 
Run if dependencies* 
Environment and Libraries*  Select notebook to configure environment

Cancel Confirm

1  2  3 

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion ...rkspace/Proyect/01_bronze_ingestion

02_silver_transformation ...ace/Proyect/02_silver_transformation

Job details

Job ID: 169813361027417

Creator: dmoralesl@ucenfotec.ac.cr

Run as: dmoralesl@ucenfotec.ac.cr

Description: Add description

Lineage: No lineage information for this job. Learn more

Performance optimized: On

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Compute

Serverless Swap

Task name*: 02_silver_transformation

Type*: Notebook

Source*: Workspace

Path*: /Workspace/Proyect/02_silver_transformation

Compute*: Serverless Autoscaling

Depends on: 01_bronze_ingestion

Run if dependencies: All succeeded

Environment and Libraries*: Notebook Environment

1

2

Cancel Create task

Le nombramos un nombre a nuestro Task y lo creamos

Databricks Free Edition

Search data, notebooks, recent, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

01_bronze_ingestion

02_silver_transformation

+ Add task

Task name* 02_silver_transformation

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/02_silver_transformation

Compute* Serverless Autoscaling

Depends on 01_bronze_ingestion

Run if dependencies* All succeeded

Environment and Libraries* Notebook Environment

Cancel Save task Swap

Job details

Job ID 169813361027417

Creator dmorales@ucenfotec.ac.cr

Run as dmorales@ucenfotec.ac.cr

Description Add description

Lineage No lineage information for this job. Learn more

Performance optimized

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Compute

Serverless

Añadimos nuestro ultimo task a nuestro job

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  D

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM Lakeflow Jobs UI: ON

Runs Tasks

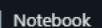
01_bronze_ingestion → 02_silver_transformation



Add Task

Search tasks

Ingestion pipeline
Create a new ingestion pipeline to ingest data from SaaS and databases
 and more

Recents


Notebook
Run a notebook

Code files

Notebook
Run a notebook

Python script
Run a Python file

SQL query
Run a SQL query

SQL file
Run a SQL file

SQL alert
Evaluate a SQL alert and notify users

Ingestion and Transformation

Run Ingestion pipeline

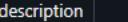
Cancel Save task Swap

Job details

Job ID: 169813361027417 

Creator: dmoralesl@ucenfotec.ac.cr

Run as: dmoralesl@ucenfotec.ac.cr 

Description: 

Lineage: 

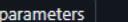
No lineage information for this job. [Learn more](#)

Performance optimized: 

Schedules & Triggers

None 

Job parameters

No job parameters are defined for this job 

Compute

Serverless

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

02_silver_transformation ...ace/Proyect/02_silver_transformation

Unnamed task Unspecified path

Job details

- Job ID: 169813361027417
- Creator: dmorales@ucenfotec.ac.cr
- Run as: dmorales@ucenfotec.ac.cr
- Description: Add description
- Lineage: No lineage information for this job. Learn more
- Performance optimized: On

Schedules & Triggers

- None Add trigger

Job parameters

- No job parameters are defined for this job Edit parameters

Compute

- Serverless Swap

Task name: Unnamed task

Type: Notebook

Source: Workspace

Path: Select Notebook (highlighted)

Compute: Serverless

Depends on: 02_silver_transformation

Run if dependencies: All succeeded

Environment and Libraries: Select notebook to configure environment

Cancel Create task

Seleccionamos el notebook de la etapa Gold para este ultimo Task

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Jobs & Pipelines >

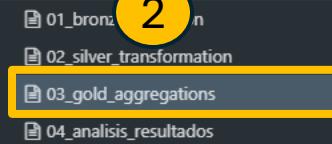
New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON 

Runs Tasks

Select Notebook

Workspace Recents

1 

2 

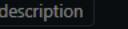
3 

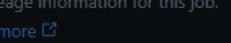
Job details

Job ID: 169813361027417 

Creator: dmoralesl@ucenfotec.ac.cr

Run as: dmoralesl@ucenfotec.ac.cr 

Description: 

Lineage: 

Performance: optimized 

Schedules & Triggers

None 

Job parameters

No job parameters are defined for this job 

Compute

Serverless 

Runs

Tasks

02_silver_transformation

01_bronze

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

Project

Task name* 

Type* 

Source* 

Path* 

Compute* 

Depends on 

Run if dependencies 

Environment and Libraries* 

Select notebook to configure environment

Cancel Create task

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

02_silver_transformation → 03_gold_aggregations

Add task

Task name* ① 03_gold_aggregations (1)

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/03_gold_aggregations

Compute* Serverless Autoscaling

Depends on 02_silver_transformation (X)

Run if dependencies All succeeded

Environment and Libraries* Notebook Environment

Cancel Save task Swap (2)

Job details

Job ID 169813361027417

Creator dmoralesl@ucenfotec.ac.cr

Run as dmoralesl@ucenfotec.ac.cr

Description Add description

Lineage No lineage information for this job. Learn more

Performance optimized

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Compute

Serverless

Le nombramos un nombre a nuestro Task y ejecutamos nuestro Job

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

02_silver_transformation → 03_gold_aggregations

+ Add task

Task name* 03_gold_aggregations

Type* Notebook

Source* Workspace

Path* /Workspace/Proyect/03_gold_aggregations

Compute* Serverless Autoscaling

Depends on 02_silver_transformation

Run if dependencies All succeeded

Environment and Libraries* Notebook Environment

Cancel Save task Swap

Triggered run in performance-optimized mode:
314937651646587
View run

Job details

Job ID 169813361027417

Creator dmoralesl@ucenfotec.ac.cr

Run as dmoralesl@ucenfotec.ac.cr

Description Add description

Lineage No lineage information for this job. Learn more

Performance optimized

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Compute

Serverless Swap

Abrimos el panel del estado de la ejecución del Job

Databricks Free Edition workspace D

Search data, notebooks, recents, and more... CTRL + P

New

Jobs & Pipelines > New Job Dec 10, 2025, 12:36 PM >

New Job Dec 10, 2025, 12:36 PM run

Lakeflow Jobs UI: ON Send feedback

Cancel job run Repair run

Graph Timeline List

Job run details

- Job ID [169813361027417](#)
- Job run ID 892055973938155
- Launched Manually
- Started Dec 10, 2025, 01:05 PM
- Ended -
- Duration 10s
- Execution time -
- Queue duration 9s
- Status Queued
 - Cancel
- Lineage No lineage information for this job.
 - Learn more
- Performance optimization Enabled

View run events

Compute

Serverless

```
graph LR; A[01_bronze_ingestion] --> B[02_silver_transformation]; B --> C[03_gold_aggregations]
```

01_bronze_ingestion
Queued · 7s
...rkspace/Proyect/01_bronze_ingestion

02_silver_transformation
Blocked · 0s
...ace/Proyect/02_silver_transformation

03_gold_aggregations
Blocked · 0s
...space/Proyect/03_gold_aggregations

Q
C
+

Esperémonos que se complete todo el proceso del Job

Databricks Free Edition workspace D

Search data, notebooks, recents, and more... CTRL + P

Jobs & Pipelines > New Job Dec 10, 2025, 12:36 PM >

New Job Dec 10, 2025, 12:36 PM run

Lakeflow Jobs UI: ON Send feedback

Delete job run Repair run

Graph Timeline List

Job run details

Job ID	169813361027417
Job run ID	892055973938155
Launched	Manually
Started	Dec 10, 2025, 01:05 PM
Ended	Dec 10, 2025, 01:08 PM
Duration	2m 57s
Execution time	1m 10s
Queue duration	1m 45s
Status	Succeeded
Lineage	No lineage information for this job.
Performance optimization	Learn more
Enabled	

View run events

Compute

Serverless Logs

01_bronze_ingestion Succeeded · 2m 8s ...rkspace/Proyect/01_bronze_ingestion

02_silver_transformation Succeeded · 20s ...ace/Proyect/02_silver_transformation

03_gold_aggregations Succeeded · 27s ...space/Proyect/03_gold_aggregations

Aquí ya se completo el proceso

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

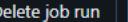
workspace  

New

Jobs & Pipelines > New Job Dec 10, 2025, 12:36 PM >

New Job Dec 10, 2025, 12:36 PM run

Lakeflow Jobs UI: ON  Send feedback

Delete job run  Repair run 

Timeline

Task name	Dec 10, 01:05 PM	Dec 10, 01:06 PM	Dec 10, 01:07 PM	Dec 10, 01:07 PM	Dec 10, 01:08 PM
> 01_bronze_ingestion		2m 8s			
> 02_silver_transformation				20.5s	
> 03_gold_aggregations					27.3s

Job run details

Job ID	169813361027417 
Job run ID	892055973938155 
Launched	Manually
Started	Dec 10, 2025, 01:05 PM
Ended	Dec 10, 2025, 01:08 PM
Duration	2m 57s
Execution time	1m 10s
Queue duration	1m 45s
Status	 Succeeded
Lineage	No lineage information for this job. Learn more 
Performance optimization	Enabled

View run events

Compute

 Serverless

Logs 

Home  Workspace  Recents  Catalog  Jobs & Pipelines  Compute  Marketplace  SQL  SQL Editor  Queries  Dashboards  Genie  Alerts  Query History  SQL Warehouses 

Job Runs  Data Ingestion  AI/ML  Playground  Experiments  Features  Models  Serving 

 **databricks**
Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace  

New

Jobs & Pipelines > New Job Dec 10, 2025, 12:36 PM >

New Job Dec 10, 2025, 12:36 PM run

Lakeflow Jobs UI: ON  Send feedback

Delete job run **Repair run**

Job run details

Job ID	169813361027417 
Job run ID	892055973938155 
Launched	Manually
Started	Dec 10, 2025, 01:05 PM
Ended	Dec 10, 2025, 01:08 PM
Duration ⓘ	2m 57s
Execution time ⓘ	1m 10s
Queue duration ⓘ	1m 45s
Status	 Succeeded
Lineage ⓘ	No lineage information for this job. Learn more 
Performance optimization ⓘ	Enabled

View run events 

Compute

 Serverless

Logs 

List

Filter task names starting with... Status Type

Status	Name	Type	Resource	Duration	Depends on
 Succeeded	01_bronze_ingestion	Notebook	 /Workspace/Proyect/01_bronze	2m 8s	
 Succeeded	02_silver_transformation	Notebook	 /Workspace/Proyect/02_silver	20s	01_bronze_ingestion
 Succeeded	03_gold_aggregations	Notebook	 /Workspace/Proyect/03_gold	27s	02_silver_transformation

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Jobs & Pipelines >

New Job Dec 10, 2025, 12:36 PM ★ Lakeflow Jobs UI: ON Send feedback

Runs Tasks

Include time in queue Started before ⌂ < Previous Next >

Run total duration: 2m 56s

Run total duration: 1m 28s

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

Dec 10

Go to the latest successful run

Start time	Run ID	Launched	Duration	Status	Error code	Run parameters	⋮
Dec 10, 2025, 01:05 PM	892055973938155	Manually	2m 57s	Succeeded			⋮

Cancel runs ⌂

Job details

Job ID: 169813361027417

Creator: dmoralesl@ucenfotec.ac.cr

Run as: dmoralesl@ucenfotec.ac.cr

Description: Add description

Lineage: 14 upstream tables, 7 downstream tables

Performance optimized: On

Schedules & Triggers

None Add trigger

Job parameters

No job parameters are defined for this job Edit parameters

Con la ejecución exitosa del Job en Databricks, todas las etapas del pipeline (Bronze, Silver y Gold) se completaron en estado “Succeeded” (verde). Esto confirma que el proceso de **Extracción, Transformación y Carga (ETL)** quedó implementado y funcionando correctamente, con los datos consolidados en formato Delta Lake dentro del catálogo y listos para análisis y reportería.

Cabe señalar que, al trabajar con la **versión gratuita de Databricks (Community Edition)**, no es posible configurar parámetros avanzados como el **tiempo de ejecución programada del Job**. Por lo tanto, la ejecución debe realizarse de manera manual cada vez que se requiera correr el pipeline.

Visualización de Resultados

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Workspace

Project

- 01_bronze_ingestion
- 02_silver_transformation
- 03_gold_aggregations
- 04_analisis_resultados

04_analisis_resultados

Last edit was 7 days ago

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Dec 02, 2025 (1s)

```
# =====
# Visualización: Ventas Totales por Sucursal
# =====

import matplotlib.pyplot as plt
import seaborn as sns
from pyspark.sql import functions as F

# 1. Lectura de la tabla Gold
# Cargamos la tabla 'df_techmart_gold_ventas_por_sucursal' desde el catálogo.
# Esta tabla contiene las métricas agregadas por sucursal.
df_sucursal = spark.read.table("workspace.default.df_techmart_gold_ventas_por_sucursal")

# 2. Conversión a Pandas
# Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib/seaborn.
pdf_sucursal = df_sucursal.toPandas()

# 3. Limpieza de formato monetario
# Si la columna 'ventas_totales' está en formato string con símbolo '$',
# la transformamos a tipo numérico (float) para poder graficar correctamente.
if pdf_sucursal["ventas_totales"].dtype == "object":
    pdf_sucursal["ventas_totales"] = pdf_sucursal["ventas_totales"].str.replace("$", "").astype(float)

# 4. Creación del gráfico de barras
plt.figure(figsize=(8,5))
sns.barplot(
    x="sucursal",
    y="ventas_totales",
    color="green",
    data=pdf_sucursal
)
plt.title("Ventas Totales por Sucursal") # Título del gráfico
plt.ylabel("Ventas ($)") # Etiqueta del eje Y
plt.xticks(rotation=45) # Rotamos etiquetas del eje X para mejor lectura
plt.tight_layout() # Ajustamos el layout para evitar solapamientos
plt.show() # Mostramos el gráfico
```

Se creó un notebook dentro del folder del proyecto destinado a generar visualizaciones con Pandas a partir de la capa Gold. El objetivo de estas gráficas es facilitar la interpretación de los datos consolidados y mostrar las conclusiones de negocio que la empresa podría obtener, como tendencias de ventas, productos más demandados o comparativas entre sucursales.

Databricks Free Edition workspace D

Search data, notebooks, recents, and more... CTRL + P

+ New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

04_analisis_resultados +

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Run all Serverless Schedule Share

Python

Dec 02, 2025 (1s)

```
# 1. Lectura de la tabla Gold
# Cargamos la tabla 'df_techmart_gold_ventas_por_sucursal' desde el catálogo.
# Esta tabla contiene las métricas agregadas por sucursal.
df_sucursal = spark.read.table("workspace.default.df_techmart_gold_ventas_por_sucursal")

# 2. Conversión a Pandas
# Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib/seaborn.
pdf_sucursal = df_sucursal.toPandas()

# 3. Limpieza de formato monetario
# Si la columna 'ventas_totales' está en formato string con símbolo '$',
# la transformamos a tipo numérico (float) para poder graficar correctamente.
if pdf_sucursal["ventas_totales"].dtype == "object":
    pdf_sucursal["ventas_totales"] = pdf_sucursal["ventas_totales"].str.replace("$", "").astype(float)

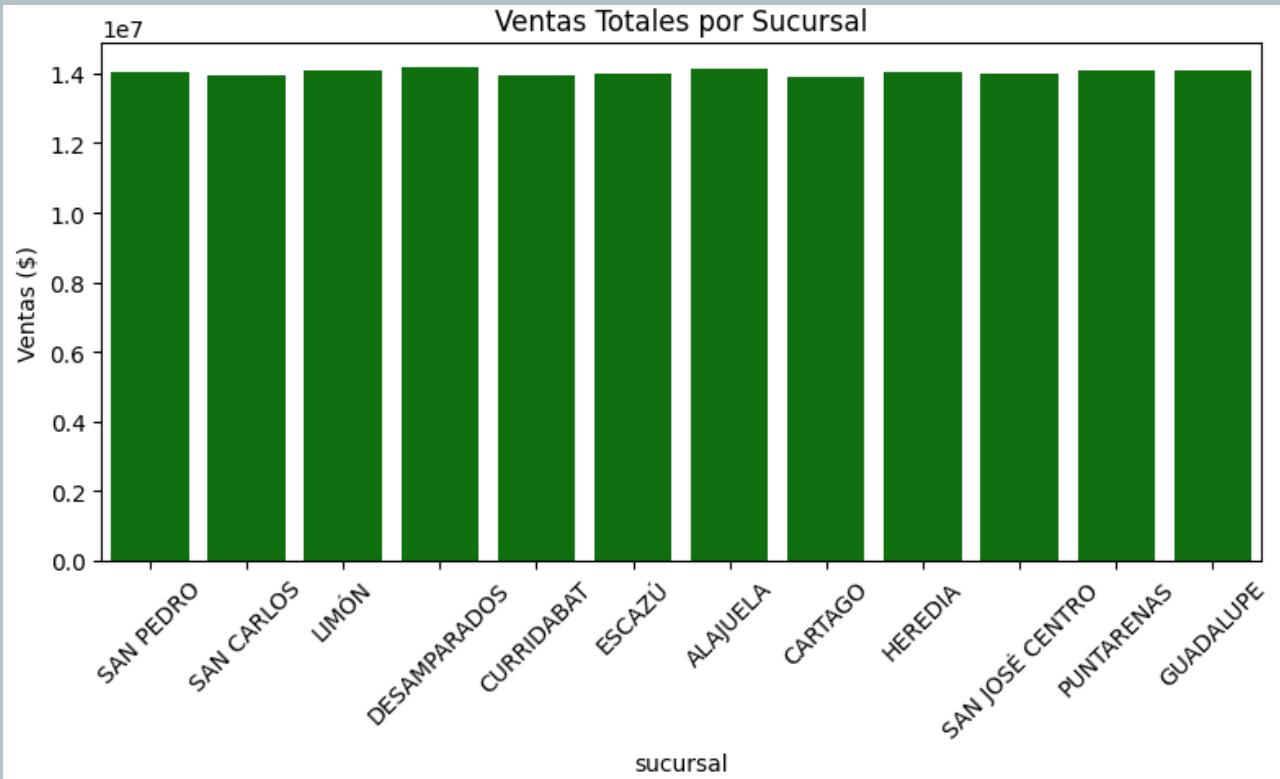
# 4. Creación del gráfico de barras
plt.figure(figsize=(8,5))
sns.barplot(
    x="sucursal",
    y="ventas_totales",
    color="green",
    data=pdf_sucursal
)
plt.title("Ventas Totales por Sucursal") # Título del gráfico
plt.ylabel("Ventas ($)") # Etiqueta del eje Y
plt.xticks(rotation=45) # Rotamos etiquetas del eje X para mejor lectura
plt.tight_layout() # Ajustamos el layout para evitar solapamientos
plt.show() # Mostramos el gráfico
```

See performance (1)

df_sucursal: pyspark.sql.connect.DataFrame = [sucursal: string, ventas_totales: string ... 2 more fields]

pdf_sucursal: pandas.core.frame.DataFrame = [sucursal: object, ventas_totales: float64 ... 2 more fields]

Optimize



El gráfico muestra las ventas totales en dólares por sucursal, utilizando datos consolidados de la capa Gold. Se observa que las 12 sucursales presentan niveles de ventas similares, sin diferencias marcadas entre ellas.

Esta visualización permite a la empresa concluir que, en términos de volumen de ventas, todas las sucursales mantienen un desempeño relativamente equilibrado, lo cual sugiere una distribución homogénea de la demanda a nivel nacional.

Este resultado puede ser útil para:

- Validar que no existen sucursales con bajo rendimiento crítico.
- Justificar una estrategia comercial uniforme en todas las regiones.
- Identificar oportunidades para potenciar ventas en zonas con mayor potencial, si se combinan estos datos con variables externas como población o competencia local.

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Serving

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

04_analisis_resultados +

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

CTRL + P

Dec 02, 2025 (1s)

=====
Visualización: Tendencia Mensual de Ventas
=====

1. Lectura de la tabla Gold
Cargamos la tabla 'df_techmart_gold_tendencia_mensual' desde el catálogo.
Esta tabla contiene las métricas agregadas por año y mes.
df_tendencia = spark.read.table("workspace.default.df_techmart_gold_tendencia_mensual")

2. Conversión a Pandas
Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib/seaborn.
pdf_tendencia = df_tendencia.toPandas()

3. Limpieza de formato monetario
Si la columna 'ventas_totales' está en formato string con símbolo '\$',
la transformamos a tipo numérico (float) para poder graficar correctamente.
if pdf_tendencia["ventas_totales"].dtype == "object":
 pdf_tendencia["ventas_totales"] = pdf_tendencia["ventas_totales"].str.replace("\$", "").astype(float)

4. Preparación de columnas de fecha
Convertimos 'mes' a entero para ordenar correctamente.
pdf_tendencia["mes"] = pdf_tendencia["mes"].astype(int)
Ordenamos cronológicamente por año y mes.
pdf_tendencia = pdf_tendencia.sort_values(["año", "mes"])
Creamos una columna 'periodo' con formato "año-mes" para el eje X.
pdf_tendencia["periodo"] = pdf_tendencia["año"].astype(str) + "-" + pdf_tendencia["mes"].astype(str)

5. Creación del gráfico de línea
plt.figure(figsize=(8,5))
sns.lineplot(
 x="periodo",
 y="ventas_totales",
 data=pdf_tendencia,
 marker="o"
)
plt.title("Tendencia Mensual de Ventas") # Título del gráfico
plt.ylabel("Ventas (\$)") # Etiqueta del eje Y

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

Dec 02, 2025 (1s) 2 Python

```
# 2. Conversion a Pandas
# Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib/seaborn.
pdf_tendencia = df_tendencia.toPandas()

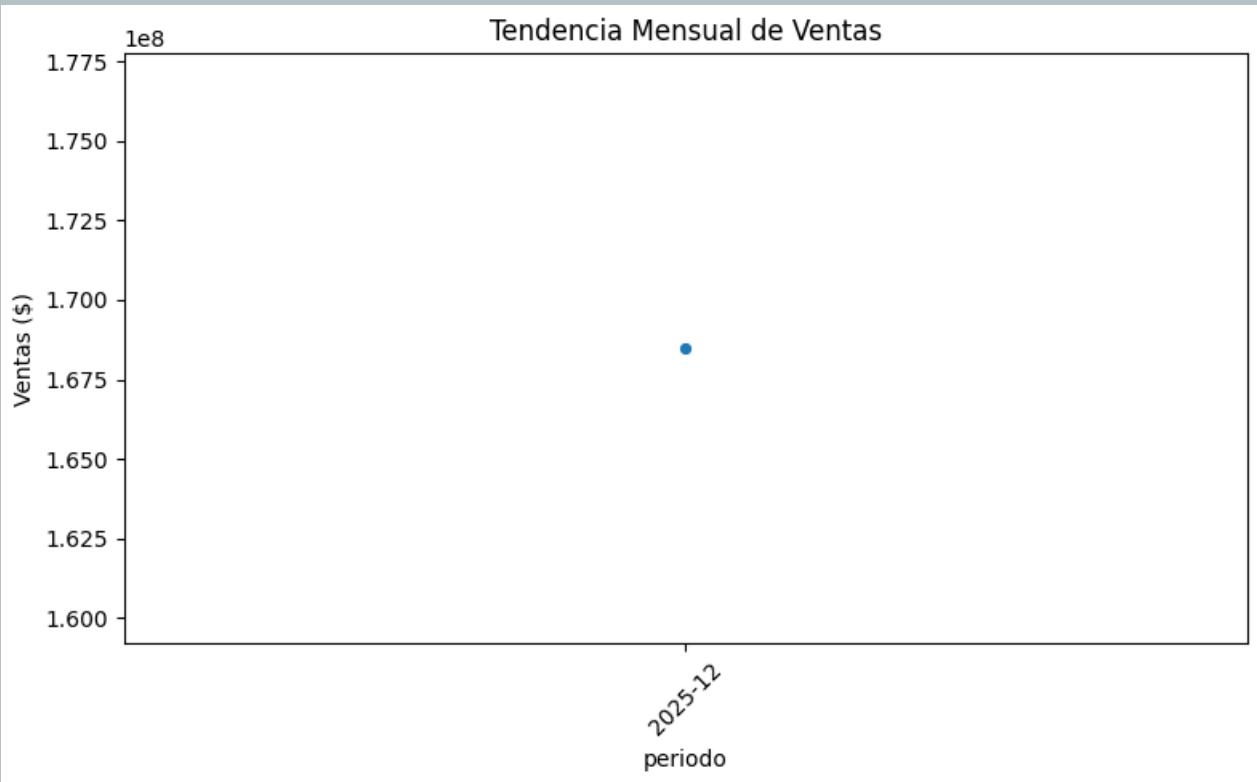
# 3. Limpieza de formato monetario
# Si la columna 'ventas_totales' está en formato string con símbolo '$',
# la transformamos a tipo numérico (float) para poder graficar correctamente.
if pdf_tendencia["ventas_totales"].dtype == "object":
    pdf_tendencia["ventas_totales"] = pdf_tendencia["ventas_totales"].str.replace("$", "").astype(float)

# 4. Preparación de columnas de fecha
# Convertimos 'mes' a entero para ordenar correctamente.
pdf_tendencia["mes"] = pdf_tendencia["mes"].astype(int)
# Ordenamos cronológicamente por año y mes.
pdf_tendencia = pdf_tendencia.sort_values(["año", "mes"])
# Creamos una columna 'periodo' con formato "año-mes" para el eje X.
pdf_tendencia["periodo"] = pdf_tendencia["año"].astype(str) + "-" + pdf_tendencia["mes"].astype(str)

# 5. Creación del gráfico de línea
plt.figure(figsize=(8,5))
sns.lineplot(
    x="periodo",
    y="ventas_totales",
    data=pdf_tendencia,
    marker="o"
)
plt.title("Tendencia Mensual de Ventas") # Título del gráfico
plt.ylabel("Ventas ($)") # Etiqueta del eje Y
plt.xticks(rotation=45) # Rotamos etiquetas del eje X para mejor lectura
plt.tight_layout() # Ajustamos el layout para evitar solapamientos
plt.show() # Mostramos el gráfico
```

See performance (1)

Optimize



2025-12
periodo

Databricks Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago periodo

Dec 02, 2025 (1s)

```
# =====
# Visualización: Top 10 Productos Más Vendidos
# =====

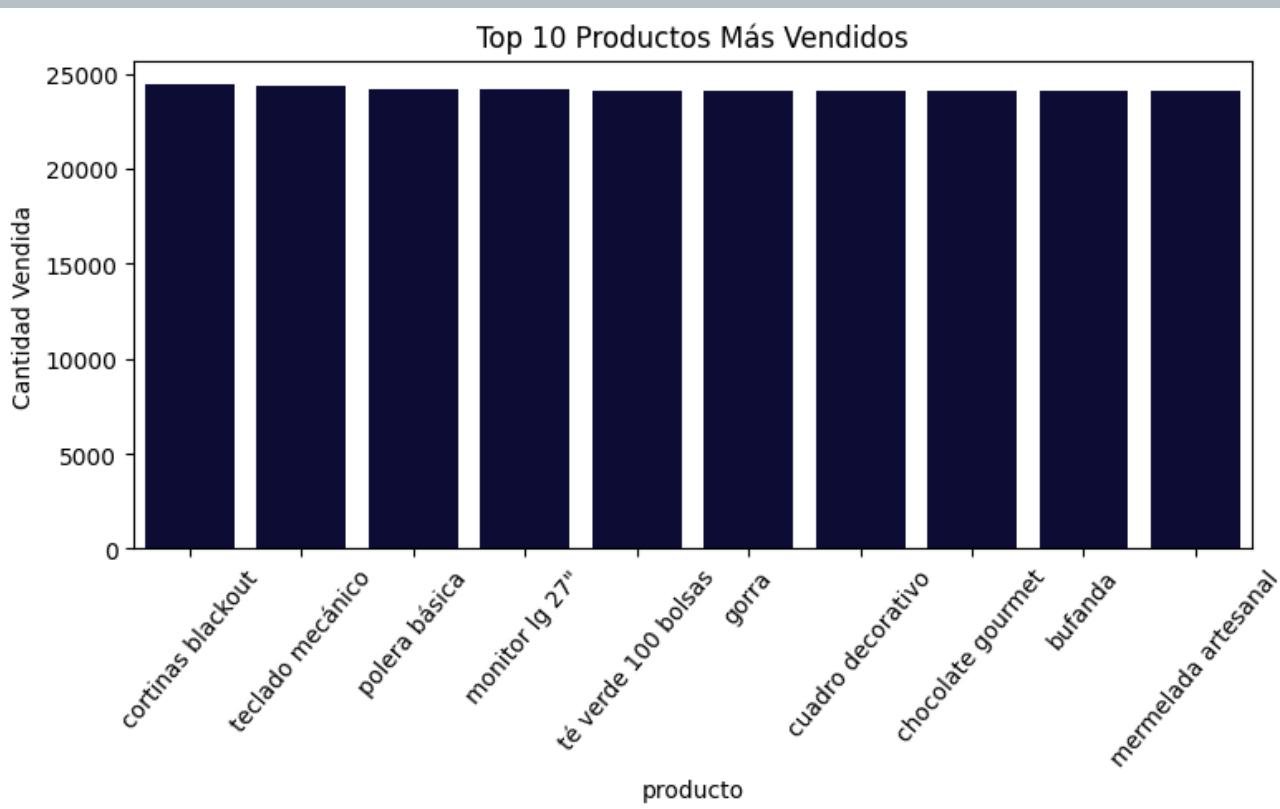
# 1. Lectura de la tabla Gold
# Cargamos la tabla 'df_techmart_gold_top_productos' desde el catálogo.
# Esta tabla contiene los 10 productos con mayor cantidad vendida.
df_top = spark.read.table("workspace.default.df_techmart_gold_top_productos")

# 2. Conversión a Pandas
# Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib/seaborn.
pdf_top = df_top.toPandas()

# 3. Creación del gráfico de barras
plt.figure(figsize=(8,5))
sns.barplot(
    x="producto",
    y="cantidad_vendida",
    color="#05053B", # Color azul oscuro personalizado
    data=pdf_top
)
plt.title("Top 10 Productos Más Vendidos") # Título del gráfico
plt.ylabel("Cantidad Vendida") # Etiqueta del eje Y
plt.xticks(rotation=50) # Rotamos etiquetas del eje X para mejor lectura
plt.tight_layout() # Ajustamos el layout para evitar solapamientos
plt.show() # Mostramos el gráfico
```

See performance (1)

Optimize

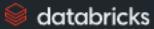


Este gráfico presenta los diez productos con mayor volumen de ventas, según los datos consolidados en la capa Gold. Se observa que todos los productos incluidos desde artículos tecnológicos como el teclado mecánico y el monitor LG 27", hasta productos de consumo como té verde, chocolate gourmet y mermelada artesanal muestran niveles de venta similares, lo que indica una diversificación equilibrada en la demanda.

Este resultado permite a la empresa concluir que:

- No existe una concentración excesiva en un solo producto, lo que reduce el riesgo comercial.
- La oferta actual satisface distintos perfiles de cliente (hogar, tecnología, moda, alimentación).
- Se puede mantener una estrategia de inventario balanceada, sin necesidad de sobrepriorizar un único artículo.

Además, esta visualización puede servir como base para identificar oportunidades de promoción cruzada entre categorías o reforzar el posicionamiento de productos con alto margen.

 **databricks**
Free Edition

workspace D

New

Home

Workspace

Recents

Catalog

Jobs & Pipelines

Compute

Marketplace

SQL

SQL Editor

Queries

Dashboards

Genie

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

AI/ML

Playground

Experiments

Features

Models

Workspace

Project

01_bronze_ingestion

02_silver_transformation

03_gold_aggregations

04_analisis_resultados

04_analisis_resultados

File Edit View Run Help Python Tabs: ON Last edit was 7 days ago

CTRL + P

Run all Serverless Schedule Share

Dec 02, 2025 (1s) 4

```
# =====
# Visualización: Distribución de Ventas por Categoría
# =====

# 1. Lectura de la tabla Gold
# Cargamos la tabla 'df_techmart_gold_ventas_por_categoria' desde el catálogo.
# Esta tabla contiene las métricas agregadas por categoría de producto.
df_categoria = spark.read.table("workspace.default.df_techmart_gold_ventas_por_categoria")

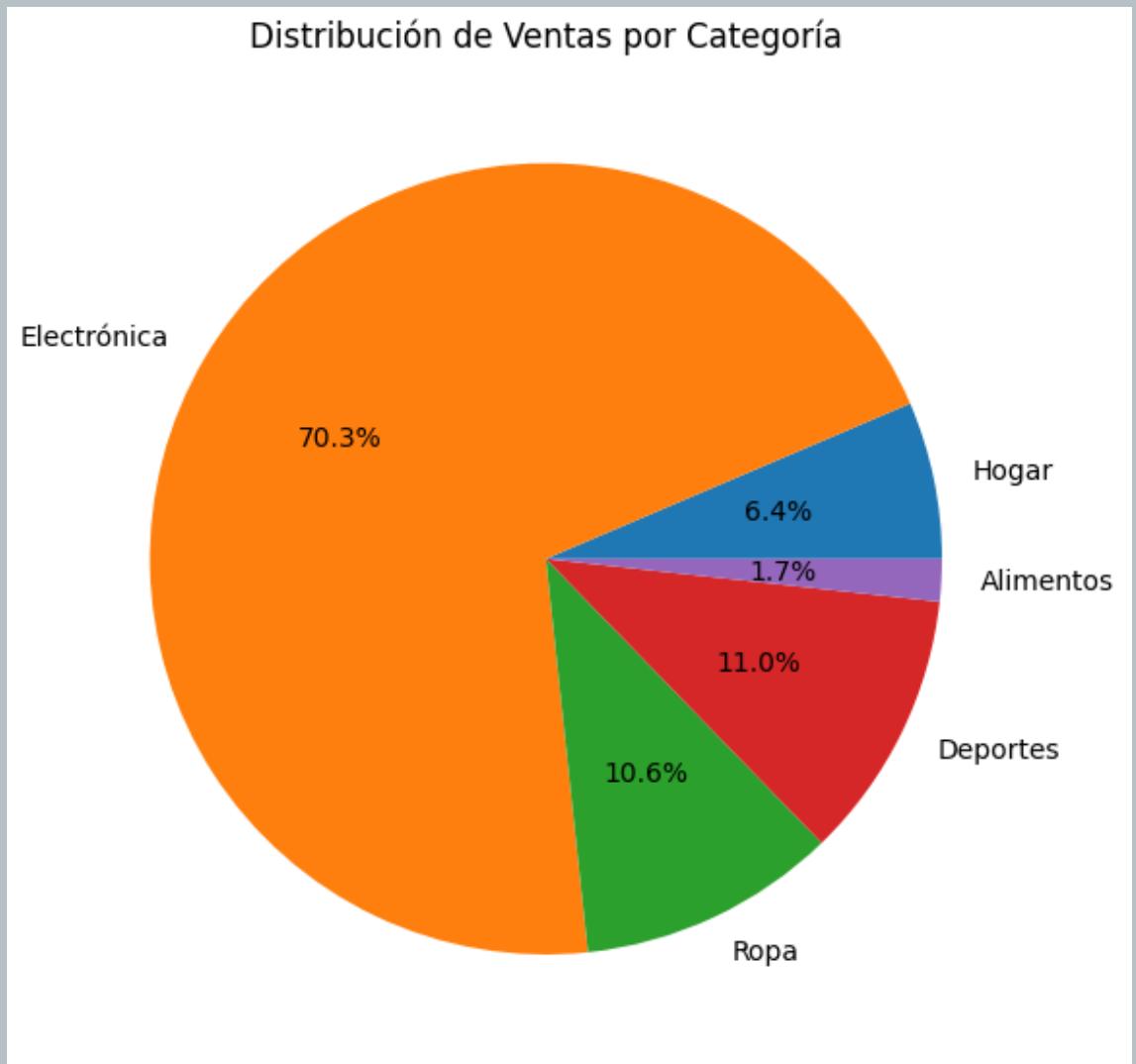
# 2. Conversión a Pandas
# Convertimos el DataFrame Spark a Pandas para poder graficar con matplotlib.
pdf_categoria = df_categoria.toPandas()

# 3. Limpieza de formato monetario
# Si la columna 'ventas_totales' está en formato string con símbolo '$',
# la transformamos a tipo numérico (float) para poder graficar correctamente.
if pdf_categoria["ventas_totales"].dtype == "object":
    pdf_categoria["ventas_totales"] = pdf_categoria["ventas_totales"].str.replace("$", "").astype(float)

# 4. Creación del gráfico de pastel
plt.figure(figsize=(6,6))
plt.pie(
    pdf_categoria["ventas_totales"],
    labels=pdf_categoria["categoria"],
    autopct="%1.1f%%" # Muestra el porcentaje con un decimal
)
plt.title("Distribución de Ventas por Categoría") # Título del gráfico
plt.tight_layout() # Ajustamos el layout para evitar solapamientos
plt.show() # Mostramos el gráfico
```

See performance (1)

Optimize



Este gráfico muestra cómo se distribuyen las ventas totales entre las distintas categorías de productos, utilizando datos consolidados de la capa Gold. Se observa una clara concentración en la categoría de Electrónica, que representa el 70.3% del total de ventas, mientras que las demás categorías Ropa, Deportes, Hogar y Alimentos tienen una participación significativamente menor.

Este resultado permite a la empresa concluir que:

- El portafolio actual está fuertemente orientado hacia productos electrónicos, lo que puede implicar una dependencia comercial de esa categoría.
- Las categorías con menor participación podrían representar oportunidades de crecimiento si se refuerzan con estrategias de marketing, surtido o posicionamiento.
- Es recomendable evaluar el margen de contribución por categoría para determinar si la concentración en Electrónica es también rentable o si conviene diversificar.

Esta visualización es clave para decisiones de inventario, campañas promocionales y análisis de riesgo comercial.

The screenshot shows the Databricks workspace interface. On the left sidebar, under the 'Workspace' section, there is a folder named 'Proyect' which is currently selected. This folder contains four items: '01_bronze_ingestion', '02_silver_transformation', '03_gold_aggregations', and '04_analisis_resultados'. All four items are of type 'Notebook' and are owned by 'dmoralesl@ucenfotec.ac.cr'. The last modification dates are Dec 01, 2025, 08:56 PM; Dec 02, 2025, 08:22 PM; Dec 02, 2025, 10:02 PM; and Dec 02, 2025, 11:35 PM respectively.

Name	Type	Owner	Created at
01_bronze_ingestion	Notebook	dmoralesl@ucenfotec.ac.cr	Dec 01, 2025, 08:56 PM
02_silver_transformation	Notebook	dmoralesl@ucenfotec.ac.cr	Dec 02, 2025, 08:22 PM
03_gold_aggregations	Notebook	dmoralesl@ucenfotec.ac.cr	Dec 02, 2025, 10:02 PM
04_analisis_resultados	Notebook	dmoralesl@ucenfotec.ac.cr	Dec 02, 2025, 11:35 PM

Al final del proyecto nuestro folder quedaría así

Conclusiones

La implementación del proyecto en Databricks bajo la arquitectura Medallion (Bronze, Silver y Gold) permitió estructurar el flujo de datos de manera clara y escalable. Cada capa cumplió su función:

- Bronze: almacenamiento crudo y confiable de la información.
- Silver: limpieza y estandarización para asegurar calidad.
- Gold: datos listos para análisis y visualización.

El Job en Databricks se ejecutó correctamente, con todas las tareas en estado “Succeeded” (verde), lo que confirma que el ETL quedó operativo y funcional. Cabe destacar que, al trabajar en la versión gratuita (Community Edition), la ejecución debe realizarse manualmente, ya que no es posible configurar tiempos de ejecución programada ni triggers automáticos.

Las visualizaciones generadas con Pandas a partir de la capa Gold aportan hallazgos clave para la empresa:

- Ventas por sucursal: desempeño equilibrado entre sedes, sin diferencias críticas.
- Tendencia mensual: diciembre 2025 muestra un volumen de ventas elevado, base para análisis estacionales.
- Top 10 productos más vendidos: demanda diversificada, sin concentración excesiva en un solo artículo.
- Distribución por categoría: predominio de la electrónica (70.3%), lo que evidencia fortaleza comercial pero también dependencia de un único segmento.

En conjunto, el proyecto demuestra que un pipeline bien diseñado no solo garantiza la integridad y disponibilidad de los datos, sino que también habilita la toma de decisiones estratégicas en áreas como inventario, marketing y expansión comercial.