

Databricks Lakehouse Project

Medallion Pipeline (Bronze / Silver / Gold)

PySpark • Delta Lake • Unity Catalog • Jobs Orchestration • Star Schema (Gold)

Portfolio summary

End-to-end ingestion, cleansing, and analytics publishing pipeline over the Iowa Liquor Sales dataset. Silver applies strict data-quality rules; Gold publishes a star schema for BI/KPIs.

Designed for repeatable runs (MERGE-based loads) and operationalized with Jobs + orchestrator.

Highlights

- Medallion architecture with layer-by-layer jobs and end-to-end orchestrator (scheduled twice daily).
- Gold model: dim_time, dim_store, dim_item (Type I) + fact_sales partitioned by (year, month).
- Idempotency & dedup strategy documented (MERGE / analytics-only dedup view).

Andres Olguin

Data Engineer — Python / SQL / PySpark / Databricks

github.com/Andy47UTN/databricks-medallion-lakehouse-pipeline

Tabla de contenido

- 1.** Versionamiento
- 2.** Introducción
- 3.** Objetivo general
- 4.** Objetivos específicos
- 5.** Diagrama de arquitectura
- 6.** Diagrama lógico de la solución
- 7.** Naming convention
- 8.** Diseño técnico de la solución
 - Tecnologías utilizadas
 - Componentes
 - Archivos requeridos
 - Capas y objetos
 - Tabla de Hechos (Fact)
 - Dimensiones
 - Automatización (Jobs)
 - Periodicidad (Triggers)
- 9.** Manual de uso
- 10.** Anexos (Notebooks y Queries)

1) Versionamiento

- Versión: 1.0
- Fecha: 2025-10-25
- Autor: Andres Olguin
- Versión inicial para entrega del Proyecto Final

2) Introducción

Solución de ingesta, depuración, modelado y publicación analítica sobre Databricks/Delta Lake para el dataset de ventas de licores de Iowa. Se utiliza arquitectura medalla (Bronze/Silver/Gold), modelo estrella en Gold y automatización con Jobs y orquestador programado. El foco es asegurar calidad, trazabilidad y performance para responder consultas de negocio.

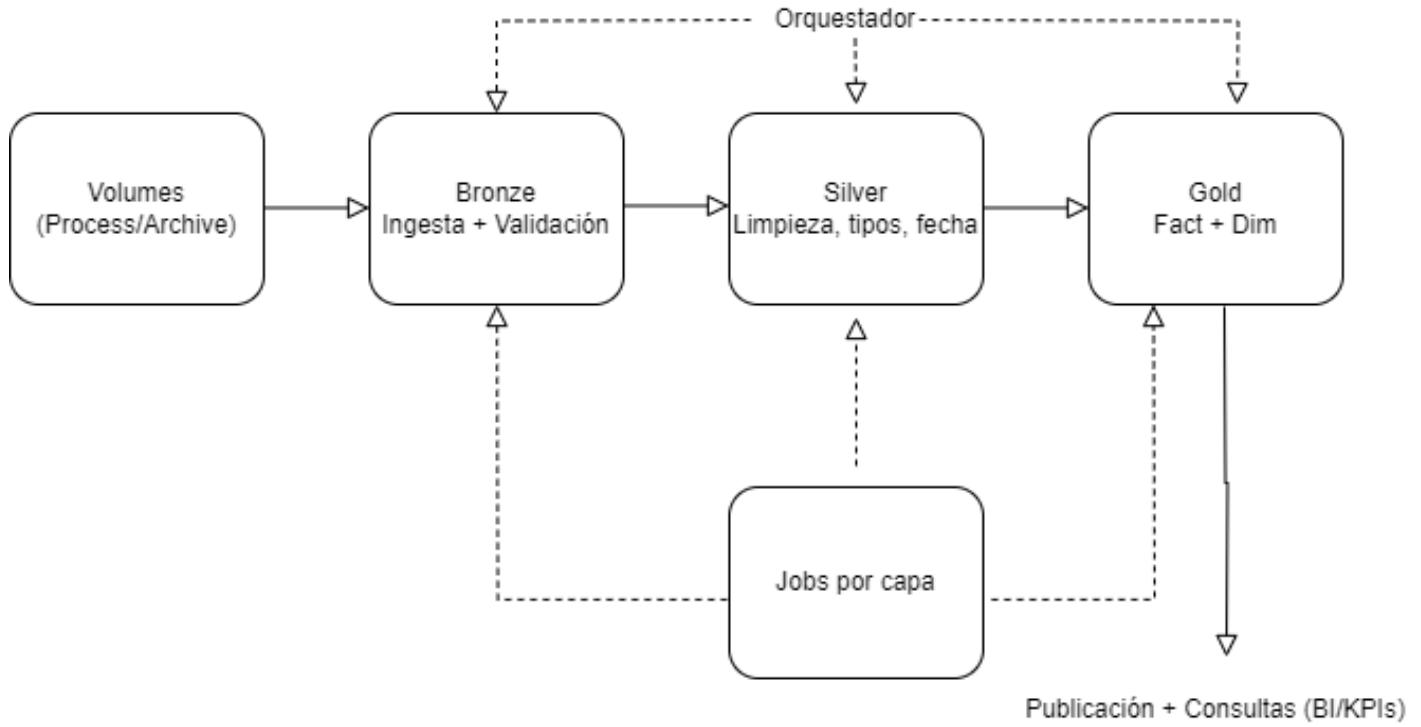
3) Objetivo general

Implementar un pipeline productizable que permita analizar ventas (tendencias, ranking por tienda/condado, categorías y rentabilidad) con datos confiables y procesos repetibles.

4) Objetivos específicos

1. Estandarizar la ingesta y el archivado de archivos con control de *naming* (_yyyyMMdd).
2. Construir Silver con reglas de calidad estrictas (anti-desplazamiento, tipificación, fechas válidas).
3. Publicar Gold con dimensiones Tipo I y fact particionada por year, month.
4. Exponer consultas de negocio (tendencias, ranking, crecimiento por categoría, margen bruto).
5. Automatizar la ejecución con Jobs por capa y orquestador con triggers L-V 06:00/18:00.

5) Diagrama de Arquitectura



6) Diagrama lógico de la solución (modelo estrella)

Gold:

```
dim_time(date_key PK, date, year, quarter, month, month_name, day,  
week_of_year, day_of_week, day_name, is_weekend) [Tipo I]
```

```
dim_store(store_id PK, store_name, address, city, zipcode,  
county_number, county) [Tipo I]
```

```
dim_item(item_no PK, item_desc, category, category_name,  
vendor_no, vendor_name, pack_int, bottle_volume_ml_d) [Tipo I]
```

```
fact_sales(invoice_line_no PK, date_key FK, store_id FK, item_no FK,  
sale_bottles, sale_dollars, sale_liters, sale_gallons,  
state_bottle_cost, state_bottle_retail, year, month, updated_at)
```

PARTITIONED BY (year, month) — Delta.

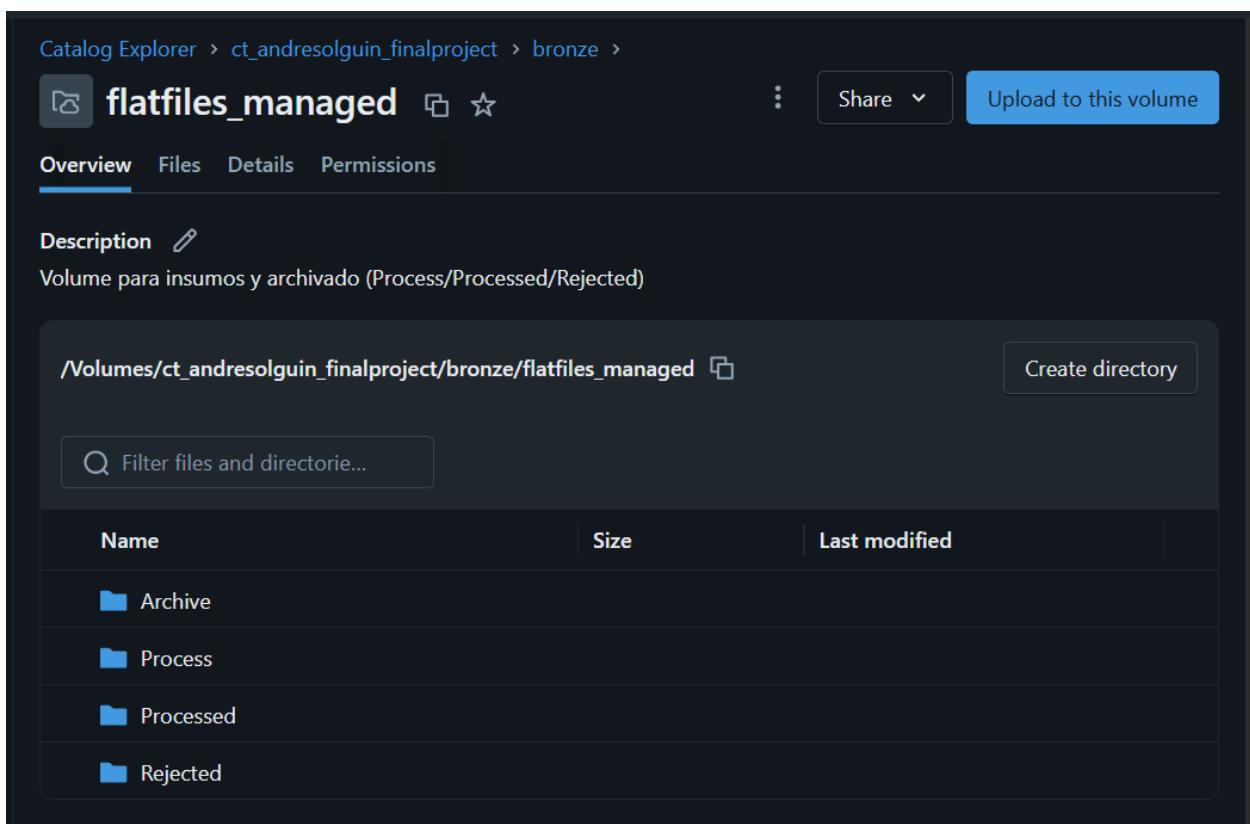
Catalog	Column	Type	Comment	Tags	Column masking ...
● Serverless Starter Warehouse Serverless 2XS	invoice_line_no	string			
Type to search...	sale_date	date			
My organization	year	int			
workspace	month	int			
system	store_id	int			
ct_andresolguin_finalproject	store_name	string			
bronze	address	string			
default	city	string			
gold	zipcode	string			
dim_item	county_number	int			
dim_store	county	string			
dim_time	item_no	int			
fact_sales					
qa_guardrail_2025_10					
v_sales_base					
information_schema					
silver					

7) Naming convention

- **Catálogo:** ct_andresolguin_finalproject
- **Esquemas:** bronze, silver, gold
- **Volumen:** flatfiles_managed
- **Rutas:**
/Volumes/{catalog}/{bronze}/flatfiles_managed/{Process|Archive}/{country}/yyyy=YYYY/mm=M
M/dd=DD/
- **Archivos de entrada:** iowa_dataset_YYYYMMDD.csv
- **Split en 4:** iowa_dataset_YYYYMMDD_part-{1..4}.csv
- **Tablas**

Silver: silver.iowa_sales_clean, silver.iowa_clean_v2_strict

Gold: gold.dim_time, gold.dim_store, gold.dim_item, gold.fact_sales
- **Jobs:** CT_Bronze_Ingest, CT_Silver_Clean, CT_Gold_Publish, CT_Pipeline_Orchestrator
- **Notebooks (carpetas en Workspace):** /bronze, /silver, /gold, /queries, /workflows, /docs



The screenshot shows the Catalog Explorer interface for the volume 'flatfiles_managed'. The top navigation bar includes 'Catalog Explorer > ct_andresolguin_finalproject > bronze > flatfiles_managed'. Below the navigation is a toolbar with icons for file operations and a 'Share' button. A blue button on the right says 'Upload to this volume'. The main area has tabs for 'Overview' (which is selected), 'Files', 'Details', and 'Permissions'. The 'Description' section contains the text: 'Volume para insumos y archivado (Process/Processed/Rejected)'. Below this is a path bar showing '/Volumes/ct_andresolguin_finalproject/bronze/flatfiles_managed' with a 'Create directory' button. A search bar with the placeholder 'Filter files and directorie...' is present. The file list table has columns for 'Name', 'Size', and 'Last modified'. The contents are: 'Archive', 'Process', 'Processed', and 'Rejected'. Each item has a small blue folder icon.

The screenshot shows the Databricks Catalog Explorer interface. The top navigation bar displays 'Catalog Explorer > ct_andresolguin_finalproject > bronze > flatfiles_managed'. Below the navigation is a header with a file icon, the volume name 'flatfiles_managed', a share icon, a star icon, a three-dot menu, a 'Share' dropdown, and a 'Upload to this volume' button. A horizontal menu bar below the header includes 'Overview' (which is underlined), 'Files', 'Details', and 'Permissions'. The main content area is titled 'Description' with an edit icon. The description text reads: 'Volume para insumos y archivado (Process/Processed/Rejected)'. Below this is a path bar showing '/Volumes/ct_andresolguin_finalproject/bronze/flatfiles_managed / Archive / usa / yyyy=2025 / mm=10 / dd=18'. To the right of the path bar is a 'Create directory' button. A search bar labeled 'Filter files and directorie...' is located below the path bar. A table follows, displaying a single file: 'iowa_dataset_20251018.csv' (Size: 4.32 GB, Last modified: 20 hours ago). At the bottom of the page is a section titled 'About this volume' with an 'Owner' field containing 'andres.olguin@clear-tech.com' and an edit icon.

8) Diseño técnico de la solución

○ 8.1 Tecnologías utilizadas

Databricks (Notebooks, Jobs, Volumes, Unity Catalog), Delta Lake (ACID, MERGE), PySpark/Spark SQL, Lakehouse (arquitectura de medallas).

```

Ksql
USE CATALOG ct_andresolguin_finalproject;
USE SCHEMA gold;

/* Repoblar GOLD desde el silver estricto (819 filas esperadas) */
CREATE OR REPLACE TABLE fact_sales
USING DELTA
PARTITIONED BY (year, month)
AS
SELECT * FROM ct_andresolguin_finalproject.silver.lowa_clean_strict;

/* Verificación rápida */
WITH gc AS (
    SELECT COUNT(*) n FROM gold.fact_sales WHERE year=2025 AND month=10
),
sc AS (
    SELECT COUNT(*) n FROM ct_andresolguin_finalproject.silver.lowa_clean_strict
),
g_Inv AS (
    SELECT DISTINCT regexp_extract(invoice_line_no,'^(0-9)+',0) inv
    FROM gold.fact_sales WHERE year=2025 AND month=10
),
r_Inv AS (
    SELECT DISTINCT regexp_extract(invoice_line_no,'^(0-9)+',0) inv
    FROM ct_andresolguin_finalproject.silver.invoice_rejected WHERE year=2025 AND month=10
),
overlap AS (SELECT COUNT(*) n FROM g_Inv g JOIN r_Inv r USING (inv))
SELECT 'gold_rows_2025_10' AS metric, (SELECT n FROM gc) AS n
UNION ALL
SELECT 'clean_strict_rows_total', (SELECT n FROM sc)
UNION ALL
SELECT 'invoices_overlap_gold_vs_rejected_2025_id', (SELECT n FROM overlap);

> See performance \(4\)

```

metric	n
gold_rows_2025_10	819
clean_strict_rows_total	819
invoices_overlap_gold_vs_rejected_2025_id	0

○ 8.2 Componentes

Volumes (landing y archive), tablas Delta por capa, Notebooks parametrizados con widgets, Jobs por capa, Orquestador, Triggers, queries de negocio.



○ 8.3 Archivos requeridos

- iowa_dataset_20251018.csv (fuente)
- iowa_dataset_20251018_part-1..4.csv (split en Bronze)

The screenshot shows the 'Catalog Explorer' interface with the path 'Catalog Explorer > ct_andresolguin_finalproject > bronze > flatfiles_managed'. The volume name 'flatfiles_managed' is displayed with a 'Share' button and a 'Upload to this volume' button. Below the volume name are tabs for 'Overview' (selected), 'Files', 'Details', and 'Permissions'. The 'Description' field contains the text 'Volume para insumos y archivado (Process/Processed/Rejected)'. A path bar at the top shows '/Volumes/ct_andresolguin_finalproject/bronze/flatfiles_managed / Processed / usa / yyyy=2025 / mm=10 / dd=18'. A 'Create directory' button is visible. A search bar says 'Filter files and directorie...'. The file list table has columns 'Name', 'Size', and 'Last modified'. It lists two items: 'iowa_dataset_20251018.csv' (4.32 GB, 1 day ago) and 'split4' (a folder).

Name	Size	Last modified
iowa_dataset_20251018.csv	4.32 GB	1 day ago
split4		

○ 8.4 Capas y objetos

Bronze

- Validación de naming (*_YYYYMMDD.csv), split en 4 archivos, archivado pos-proceso.

Silver

- iowa_sales_clean: normalización básica.
- iowa_clean_v2_strict: reglas de calidad (fecha válida, rangos de pack y bottle_volume_ml, sale_bottles entero y rango, precios > 0, retail ≥ cost, etc.).

Gold

- dim_time (rango dinámico entre MIN/MAX(sale_date) en Silver).
- dim_store, dim_item → MERGE Tipo I (sin duplicar al reprocesar).
- fact_sales → MERGE idempotente desde silver.iowa_clean_v2_strict, partition (year, month).

The screenshot shows the Catalog Explorer interface for the asset 'fact_sales'. The top navigation bar includes 'Catalog Explorer', 'ct_andresolguin_finalproject', 'gold', and 'fact_sales'. The main tabs are 'Overview', 'Sample Data', 'Details', 'Permissions', 'History', 'Lineage' (which is selected and underlined), 'Insights', and 'Quality'. Below the tabs are filters: 'Filter lineage', 'All assets', 'Up and Downstream', 'Last 3 months', and a 'See lineage graph' button. The main table lists the lineage relationships:

Name	Direction	Type	Last activity
CT_Pipeline_Orchestrator	↓ Downstream	Job	anteayer
iowa_clean_v2_strict ct_andresolguin_finalproject.silver	↑ Upstream	Table	anteayer
CT_Pipeline_Orchestrator	↑ Upstream	Job	anteayer
iowa_clean.strict ct_andresolguin_finalproject.silver	↑ Upstream	View	anteayer
qa_guardrail_2025_10 ct_andresolguin_finalproject.gold	↓ Downstream	View	anteayer
CT_Gold_Publish	↓ Downstream	Job	anteayer
CT_Gold_Publish	↑ Upstream	Job	anteayer
nb_gold_publish	↓ Downstream	Notebook	anteayer
nb_gold_publish	↑ Upstream	Notebook	anteayer
dim_store ct_andresolguin_finalproject.gold	↑ Upstream	Table	hace 4 días

○ 8.5 Tabla de Hechos (Fact)

gold.fact_sales

- **PK lógico:** invoice_line_no
- **FK:** date_key → dim_time, store_id → dim_store, item_no → dim_item
- **Medidas:** sale_bottles INT, sale_dollars DECIMAL(14,2), sale_liters DECIMAL(12,3), sale_gallons DECIMAL(12,3)
- **Atributos de costo/precio:** state_bottle_cost DECIMAL(12,2), state_bottle_retail DECIMAL(12,2)
- **Particiones:** year INT, month INT

The screenshot shows a data catalog interface with a sidebar and a main table view.

Left Sidebar (Catalog):

- Serverless Starter Warehouse Serverless 2XS
- Type to search...
- My organization
 - > workspace
 - > system
 - > ct_andresolguin_finalproject
 - > bronze
 - > default
 - > gold
 - fact_sales
 - qa_guardrail_2025_10
 - v_sales_base
 - > information_schema
 - > silver
 - > dev
 - > prod
 - Delta Shares Received
 - > samples

Main View:

Column	Type	Comment	Tags	Column masking rule
invoice_line_no	string			
sale_date	date			
year	int			
month	int			
store_id	int			
store_name	string			
address	string			
city	string			
zipcode	string			
county_number	int			
county	string			
item_no	int			
item_desc	string			
category	int			
category_name	string			
vendor_no	int			

Right Panel:

- Popularity
- Tags
 - Add tags
- Row filter
 - Add filter
- Insights
 - Top users
 - andres.olguin@clear-tech.com
 - Top joins
 - ct_andresolguin_finalproject.gold.dim_item
 - ct_andresolguin_finalproject.gold.dim_store
 - Related assets
 - nb_gold_publish
 - nb_queries_negocio_sales
 - nb_bronze_ingest
 - Untitled Notebook 2025-10-21 19:59:14
- All insights

○ 8.6 Dimensiones

- **dim_time** (PK date_key INT yyyyMMdd) — Tipo I.
- **dim_store** (PK store_id INT) — Tipo I.
- **dim_item** (PK item_no INT) — Tipo I.

Todas cargadas por MERGE (upsert) sin duplicados al reprocesar

The screenshot shows the 'dim_item' table details in the Catalog Explorer. The table has 10 columns:

Column	Type	Comment	Tags	Column masking rule
item_no	int			
item_desc	string			
pack	int			
bottle_volume_ml	int			
category_id	int			
category_name	string			
vendor_no	int			
vendor_name	string			
updated_at	timestamp			

○ 8.7 Automatización (Jobs)

- **CT_Bronze_Ingest** - ingestión/validación/archivado.
- **CT_Silver_Clean** - construcción de iowa_clean_v2_strict.
- **CT_Gold_Publish** - creación/actualización de dim_* + fact_sales.
- **CT_Pipeline_Orchestrator** - orquesta 3 tareas en secuencia.

The screenshot shows the 'Jobs & Pipelines' section of a data platform. At the top, there are three cards: 'Ingestion pipeline' (ingest data from popular apps, databases and file sources), 'ETL pipeline' (Build ETL pipelines using SQL and Python), and 'Job' (Orchestrate notebooks, pipelines, queries and more). Below these are two tabs: 'Jobs & pipelines' (selected) and 'Job runs'. A search bar and filter buttons ('All', 'Jobs', 'Pipelines', 'Owned by me', 'Accessible by me', 'Favorites', 'Tags', 'Run as') are present. The main table lists four jobs:

Name	Type	Tags	Run as	Trigger	Recent runs
CT_Bronze_Ingest	Job		andres.olguin@cl...	— — — —	Run now
CT_Gold_Publish	Job		andres.olguin@cl...	— — — —	Run now
CT_Pipeline_Orchestrator	Job		andres.olguin@cl...	Scheduled	Run now
CT_Silver_Clean	Job		andres.olguin@cl...	— — — —	Run now

○ 8.8 Periodicidad (Triggers)

- Orquestador programado dos veces al día (L-V): 06:00 y 18:00.
- Todos los jobs permiten ejecución manual (“Run now”).

The screenshot shows the details for the 'CT_Pipeline_Orchestrator' job. It includes sections for 'Runs' (listing recent runs with durations of 37m 55s and 18m 58s, and tasks: Bronze_Ingest, Silver_Clean, Gold_Publish), 'Run as' (set to andres.olguin@clear-tech.com), 'Description' (Add description), 'Lineage' (15 upstream tables, 11 downstream tables), 'Performance optimized' (toggle switch), and 'Schedules & Triggers' (triggered at 06:00 AM and 06:00 PM, Monday through Friday (UTC-03:00)). Buttons for 'Edit trigger', 'Pause', and 'Delete' are also visible.

9) Manual de uso

Ejecución por capa

1. Abrir el job de la capa y Run now.
2. Verificar/ajustar parámetros (widgets): catalog, schema, process_date, etc.
3. Revisar Runs → Logs si hace falta.

Ejecución orquestada

1. Abrir CT_Pipeline_Orchestrator y Run now (o esperar al trigger).
2. Verificar Gantt de tasks (todo en Succeeded).

Reprocesos / Idempotencia

- Dimensiones y fact usan MERGE; es seguro reprocesar sin duplicar.
- fact_sales particionada por (year, month) optimiza pruning y cargas.

Recuperación de errores

- Revisar logs del run fallido.
- Re-lanzar la task afectada o el orquestador completo.
- Los insumos quedan archivados por fecha de proceso (trazabilidad).

10) Anexos

A. Notebooks

- /bronze/nb_bronze_ingest
- /silver/nb_silver_clean
- /gold/nb_gold_publish

B. Queries (resumen)

- Tendencias diarias/mensuales/anuales (con YoY donde aplica).
- Ranking tiendas y condados (botellas y ventas).
- Categorías Top y mayor crecimiento YoY (último año completo).
- Margen por item, categoría y vendor; panel precio-volumen (elasticidad y sugerencia de precio).