Entrega del Segundo Parcial

MVP técnico

Objetivo: demostrar un "end-to-end mínimo" funcionando que conecte *landing* → *Bronze* → *Silver* → *Gold* → *Serving (Cassandra)* con PySpark (Colab) y Structured Streaming, cumpliendo los requisitos obligatorios del proyecto.

Alcance mínimo requerido

1. **Batch a Bronze**

- Ingesta de al menos 3 maestros (p. ej., `customers_orgs.csv`, `users.csv`, `billing_monthly.csv`) a **Parquet particionado** con tipificación explícita, columnas técnicas (`ingest_ts`, `source_file`) y **dedupe** cuando aplique.

2. **Streaming a Bronze**

- Structured Streaming leyendo `usage_events_stream/*.jsonl` con **esquema explícito**, `withWatermark`, **dedupe por `event_id`** y manejo de *late data*. Checkpointing habilitado.

3. **Silver (mínimo)**

- Limpieza/conformance para **eventos + 1 maestro**, joins de enriquecimiento y **al menos 3 features** (p. ej., `daily_cost_usd`, `requests`, `genai_tokens` o `carbon_kg`).

- **Calidad de datos**: 3 reglas activas (ej.: `event_id` no nulo/único; `cost_usd_increment ≥ -0.01` + flag de anomalía; `unit` no nulo cuando `value` exista) y **quarantine** con muestras.

4. **Gold (mínimo)**

- Un mart **FinOps**: `org_daily_usage_by_service` (grano diario por org/servicio) con métricas y costos.

5. **Serving en Cassandra (AstraDB)**

- **Keyspace** creado y **1 tabla** modelada *query-first* para el mart anterior; carga desde Spark (`foreachBatch` o conector). Ejecutar **2 consultas mínimas** del enunciado (p. ej., #1 y #2) y adjuntar CQL + captura.

6. **Idempotencia y evidencias**

- Re-ejecución sin duplicar (mostrar conteos antes/después). Particionado sensato (evidencias de rutas y tamaños).

Artefactos a entregar

- **Repo** con notebooks/código reproducible (Colab o .py), **estructura de zonas** (Bronze/Silver/Gold en Parquet), **scripts CQL**, y **README (Quickstart)** con pasos exactos.

- **Diagrama** actualizado + breve **log de decisiones** (patrón Lambda/Kappa, particiones, claves Cassandra, umbrales).

Criterios de aceptación (checklist)

- Batch y streaming corren con datos provistos.

- Reglas de calidad y **quarantine** efectivas (ejemplos).

- Mart FinOps en Gold + tabla en Cassandra poblada.

- 2 consultas sobre AstraDB con resultados.

- Reprocesar no duplica (idempotencia OK).