Proyecto RA2: Ecosistema de Datos (Datalake & Datawarehouse) sobre Polymarket

1. Objetivo del Proyecto

Diseñar un pipeline de datos profesional que cubra todo el ciclo de vida del dato: Ingesta en **Delta Lake**, estructuración en **Data Warehouse (NeonDB)**, exposición mediante **API** y visualización estratégica en **Tableau** para la toma de decisiones de inversión.

2. Fase 1: Data Lake (Capa de Almacenamiento Crudo)

Cada alumno debe desarrollar un proceso de extracción en Python para obtener la totalidad de los datos de Polymarket:

- **Endpoints obligatorios:** `Tags`, `Events`, `Series` y `Markets`.
- **Tecnología:** Almacenamiento en un bucket de **AWS S3** utilizando el formato **Delta Lake** en el bucket **lasalle-bigdata-2025-2026**.
- **Entregable:** Un **Reporte de Volumetría** que detalle:
    - Cantidad de registros por entidad.
    - Distribución de mercados (activos vs. cerrados).
    - Análisis de relaciones (por ejemplo, cuántos mercados dependen de cada evento o serie).

3. Fase 2: Data Warehouse (Capa Gold - NeonDB)

Se debe realizar una carga completa desde el Data Lake hacia una base de datos **PostgreSQL en NeonDB**.

- **Modelado:** El esquema debe estar optimizado para consultas analíticas. Debe incluir obligatoriamente las dimensiones de `Mercado`, `Tiempo`, `Evento/Serie` y la jerarquía de `Tags`.
- **Integridad:** Los datos deben estar limpios y normalizados (conversión de tipos de datos, manejo de nulos y desanidado de precios).

4. Fase 3: Exposición de Datos (GitHub Compartido)

Además del repositorio individual con el código del pipeline, existirá un **GitHub compartido por toda la clase (https://github.com/lasalle-ai/apis)**. Cada alumno debe registrar y documentar los **Endpoints** de una API (desarrollada preferiblemente en FastAPI o Flask) que permita a terceros consultar datos de su Data Warehouse.

Ejemplos de endpoints que el alumno debe proveer:

- `GET /markets/top-volume`: Devuelve los 10 mercados con más volumen de su categoría.
- `GET /series/{id}/probability`: Devuelve la evolución de probabilidad media de una serie específica.
- `GET /tags/search?name=crypto`: Devuelve todos los eventos relacionados con un tag específico.
- `GET /events/closing-soon`: Lista de eventos que finalizan en las próximas 24-48 horas.

Son solo ejemplos, los endpoints ha crear deben ser propuestos por el alumno para que un servicio externo pueda consultar datos.

5. Fase 4: Análisis Estratégico en Tableau

El alumno debe actuar como un **Analista de Inversiones**. Utilizando los datos de su propio Data Warehouse, debe crear:

- **Mínimo 2 Dashboards** interactivos. Minimo 8 gráficos
- **Objetivo:** El diseño debe permitir a un inversor identificar oportunidades de arbitraje, detectar picos de liquidez o analizar tendencias de mercado para decidir dónde colocar su capital.
- *Nota: La elección de los KPIs y el tipo de gráficos queda a discreción del alumno, basándose en lo que considere "información relevante para inversión".*

Requisitos de Entrega

1. **GitHub Individual:** Código fuente de la ETL (Python), archivos de configuración y scripts DDL de la base de datos.
2. **GitHub de Clase:** Documentación de los endpoints y acceso (si aplica) para que otros alumnos puedan consultar sus datos.
3. **Documento PDF:** Reporte de volumetría e insights iniciales detectados en el Data Lake.
4. **Archivo Tableau:** Archivo `.twbx` que será presentado en la clase.