PI M4: Diseño e implementación de un pipeline de datos ETLT escalable sobre un Data Lake en la nube

I. Objetivo:

Diseñar e implementar un **pipeline de tipo ETLT sobre una arquitectura de Data Lake** escalable que permita a la organización integrar información proveniente de diversas fuentes, con la finalidad de mejorar la toma de decisiones basada en datos confiables y actualizados.

II. Visión general de la arquitectura:

El proyecto implementa un **pipeline ETLT (Extract, Load, Transform, Load)** con el objetivo de construir un **Data Lake escalable y gobernado en AWS** para la integración, transformación y análisis de datos meteorológicos provenientes de diversas fuentes, tanto **batch** como **streaming**. La solución integra herramientas open source y servicios nativos de AWS bajo una **arquitectura medallion (Bronze, Silver, Gold)**, garantizando calidad, trazabilidad y flexibilidad analítica.

Flujo general del pipeline:

1. **Extracción:** obtención de datos desde la API de OpenWeather y fuentes manuales (archivos JSON históricos).
2. **Carga (L):** almacenamiento de datos crudos en Amazon S3, mediante Airbyte y Kafka.
3. **Transformación (T):** procesamiento y limpieza en Apache Spark (PySpark), ejecutado sobre EC2 dockerizada.
4. **Segunda Carga (L):** escritura de resultados en S3 Silver (curado) y S3 Gold (modelo dimensional).
5. **Orquestación:** mediante Apache Airflow, que programa y supervisa los flujos completos, ejecutado sobre EC2 dockerizada.
6. **Streaming:** integración de Kafka + Spark Structured Streaming para ingesta continua desde la ciudad de Irapuato, ejecutado sobre EC2 dockerizada.

III. Propósito de cada capa del data lake

Tabla 1: *Estructura del Data Lake*

Capa	Propósito	Rol en el flujo	Tecnologías clave
Bronze y raw-streaming layers	Almacena los datos crudos provenientes de las fuentes, en formato original (JSON/Parquet).	Punto de entrada del pipeline; mantiene trazabilidad y reproducibilidad.	S3, Airbyte, Kafka, Glue Catalog

Silver layer	Contiene datos validados, estandarizados y enriquecidos con cálculos adicionales (fechas locales, proxies solares, indicadores de viento, etc.).	Facilita la construcción de métricas consistentes y reduce la duplicidad en transformaciones posteriores.	Spark (PySpark), S3, EC2 Docker, Glue Catalog
Gold (Analytics Layer)	Reúne los datos listos para análisis, modelados con enfoque **dimensional (Kimball)** en tablas de hechos y dimensiones.	Sirve como fuente directa para notebooks, dashboards y consultas analíticas.	Spark, S3, notebooks (.ipynb)

IV. Justificación del Stack Tecnológico

Tabla 2: *Stack tecnológico*

Componente	Tecnología	Justificación
Ingesta batch	Airbyte (conector HTTP)	Open source, fácil de extender, compatible con OpenWeather API
Ingesta streaming	Apache Kafka	Permite capturar datos en tiempo real y desacoplar productores/consumidores, garantizando durabilidad y escalabilidad horizontal.
Almacenamiento	Amazon S3	Servicio altamente disponible, escalable y económico. Separa datos en capas (Bronze, Silver, Gold, Streaming).
Catálogo y gobernanza	AWS Glue Catalog + Lake Formation	Permite registrar metadatos, definir permisos y habilitar consultas vía Athena. Mejora el control de acceso y el lineage.
Procesamiento Batch	Apache Spark (PySpark) en EC2 Dockerizada	Framework distribuido para grandes volúmenes, permite transformaciones complejas (ETL) con alta velocidad y soporte para Python.

Procesamiento Streaming	Spark Structured Streaming	Integración nativa con Kafka y S3
Orquestación	Apache Airflow	Estándar industrial para programar y monitorizar DAGs, ejecuta tareas remotas en Spark (SSH) y gestiona dependencias entre etapas.
Visualización / Análisis	Google Colab / Notebooks (.ipynb)	Permite exploración interactiva de resultados Gold y elaboración de reportes analíticos y gráficos.
Infraestructura	Amazon EC2 + Docker	Facilita el aislamiento de entornos, control de dependencias y despliegue reproducible de servicios (Airflow, Spark, Kafka).

V. Fuentes de Datos y Relevancia Analítica

Tabla 3: *Origen de los datos y relevancia analítica*

Fuente	Tipo	Descripción	Valor Analítico
OpenWeather API	Streaming y Batch	Datos meteorológicos (temperatura, viento, precipitación, radiación solar, etc.) obtenidos por ciudad y frecuencia horaria.	Principal fuente para responder preguntas de negocio relacionadas con clima, energía solar y patrones de viento.
Archivos JSON manuales (históricos)	Batch	Datos climáticos descargados manualmente para realizar *backfill* y completar series temporales.	Permiten análisis comparativos interanuales y validación de datos recientes.
Kafka Topic openweather-topic	Streaming	Canal de ingestión en tiempo real desde la API hacia Spark Structured Streaming.	Permite análisis continuo

VI. Preguntas de negocio

El pipeline responde a un conjunto de preguntas clave de negocio relacionadas con energía renovable, patrones meteorológicos y desempeño ambiental, tales como:

Tabla 4: *Preguntas de negocio*

Sintaxis	Pregunta de negocio	Objetivo Analítico
q1_solar_hour_by_month	¿Cuáles son las horas con mayor potencial solar por ciudad y mes?	Identificar las franjas horarias más favorables para generación solar fotovoltaica.
q2_wind_patterns	¿Cuáles son las horas con mayor potencial eólico por ciudad y mes?	Identificar las franjas horarias más favorables para generación de energía eólica
q3_weather_main	¿Qué condiciones climáticas predominan durante los periodos observados?	Clasificar y cuantificar la frecuencia de condiciones principales (clear, cloudy, rain, etc.).
q4_today_vs_last_year	¿Cómo se comportan las predicciones meteorológicas actuales en comparación con las condiciones observadas en el pasado reciente?	Medir variación interanual de condiciones climáticas, apoyando análisis de tendencia.
q5_best_days_topk **q5_worst_days_topk**	¿Cuáles fueron los días con mayor y menor potencial energético en cada ubicación durante el periodo de análisis?	Comparar eficiencia entre ubicaciones para planificación de proyectos energéticos.
q6_wind_sector_topk	¿Cuál es el sector del viento predominante en cada ciudad y periodo?	Identificar direcciones dominantes de viento (N, NE, E, SE, etc.) útiles para ubicación de aerogeneradores.
q7_temp_extremes	¿Qué día del mes actual presenta los valores más altos y bajos de temperatura?	Detectar extremos térmicos diarios para evaluación ambiental y alertas de calor/frío.

VII. Método Kimball en la Capa Gold (Modelo Dimensional)

Principio: organizar datos en hechos y dimensiones con granularidad explícita, para responder preguntas del negocio de forma simple, rápida y estable**.**

a) **Granularidad**
   1. **fact_weather_hourly:** una fila por ciudad-hora.
   2. **fact_weather_daily:** una fila por ciudad-día.

b) **Hechos:**
   1. **fact_weather_hourly.**
   2. **fact_weather_daily.**

c) **Dimensiones (Dims)**
   1. **dim_date**: calendario (año, mes, día)
   2. **dim_city**: metadatos de la ubicación (nombre, país, lat/lon, zona horaria).
   3. **dim_weather_condition**: catálogo de weather_main/id (clear, clouds, rain…). *SCD Tipo 1* (estática).
   4. **dim_wind_sector**: discretización de wind_deg en 8/16 sectores (N, NE, …). *SCD Tipo 1* (estática).

d) **SCD (Slowly Changing Dimensions)**
   1. **Tipo 1 (sobrescribe):** para catálogos/atributos sin interés histórico

VIII. Gobernanza, catálogo de datos y ciclo de vida

La gobernanza se gestionará con AWS Lake Formation, que será responsable de centralizar los permisos a nivel de base, tabla o columna. Se registraron locations de los tres buckets (Bronze, Silver, Gold) en Lake Formation con Hybrid access mode y se crearon databases db_bronze, db_silver, db_gold en Glue Catalog.

Ciclo de vida:

1. **Bronze:**
   - Día 0: objetos cargados (clase Standard).
   - Día 90: transición a Glacier Instant Retrieval (GIR).
   - Día 365: transición a Glacier Deep Archive (GDA).

Justificación: Acceso esporádico para bronce ya que se consulta rara vez (solo para auditoría o reprocesos especiales). GIR reduce costo manteniendo recuperación casi inmediata si se necesitara rehacer Silver con nuevas reglas. GDA a 1 año conserva el histórico a costo ultra bajo.

2. **Silver:**
   - Día 0: objetos cargados (Standard).
   - Día 365: transición a Standard-IA.

Justificación: Silver se consulta en backfills y validaciones técnicas, no en dashboards diarios. Después de 365 se mueve a IA para ahorrar costos de mantenimiento con latencia de acceso baja (segundos) para recalcular Gold comparativos de un año a otro.

3. **Gold:**
    - Día 0: objetos cargados (Standard).
    - Día 180: transición a Standard-IA.

Justificación: Consumo activo los primeros meses (notebooks, validaciones) a los 6 meses baja la frecuencia de consulta se mueve a IA reduciendo costo sin afectar la experiencia (sigue siendo acceso casi inmediato para cualquier reporte histórico).

IX. Diagrama de la arquitectura del pipeline ETLT

Figura 1: Diagrama de la arquitectura del pipeline ETLT