



PIPELINE DE INTEGRACIÓN Y ANÁLISIS DE DATOS

CLIMÁTICOS PARA BOLIVIA CLIMA 360

INTEGRANTES:

- GEORGINA QUIROZ MENDOZA
- CARLOS ALMARAZ ESCOBAR
- DILAN CONDORI ALEJO
- ALVARO LUIS JURADO ALFARO
- JAIME MONTECINOS MARQUEZ

RESUMEN

El proyecto Clima360 tiene como objetivo construir un pipeline de datos climáticos completo y automatizado, integrando fuentes históricas y en tiempo real, aplicando limpieza, validación, enriquecimiento y análisis de la información meteorológica. Se implementa un esquema ETL con Apache Airflow para la orquestación de tareas, garantizando reproducibilidad, trazabilidad y escalabilidad de los procesos. El proyecto incorpora APIs (OpenWeather y Meteostat), datasets históricos (FAOSTAT y Environment Temperature Change de Kaggle) y produce reportes automáticos de calidad de datos.

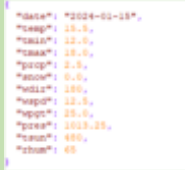
1. OBJETIVO

Construir un pipeline de datos para la ingesta, limpieza, integración y análisis de información climática, utilizando múltiples fuentes de datos (APIs y archivos históricos), control de calidad automatizado y almacenamiento estructurado.

2. ALCANCE

- Integración de APIs externas: OpenWeather (datos climáticos en tiempo real) y Meteostat (datos históricos).
- Incorporación de conjuntos históricos: FAOSTAT y Environment Temperature Change de Kaggle.
- Estandarización, validación y clasificación de variables meteorológicas.
- Generación automática de reportes de calidad y diccionario de datos.
- Implementación de arquitectura orquestada con Airflow.

3. FUENTES DE DATOS

Fuente	Tipo	Descripción
OpenWeather API	Tiempo real	Servicio: Empresa en el Reino Unido que ofrece datos meteorológicos en tiempo real. Datos: temperaturas promedio diarias en formato Json. Puntos de referencia en Bolivia: La Paz, Santa Cruz de la Sierra, Cochabamba, Oruro, Beni, Potosí, Tarija, Pando, Sucre
Meteostat API	Histórica	Registros meteorológicos diarios por estación. Librería: meteostat (Python).  Datos: temperaturas promedio diarias en formato Json. Puntos de referencia en Bolivia: La Paz, Santa Cruz de la Sierra, Cochabamba, Oruro, Beni, Potosí, Tarija, Pando, Sucre.
Environment Temperature Change de Kaggle	csv	Variación de temperatura histórica por países y meses. Tiene 66 columnas (una por cada año, desde 1961 a 2019). Formato ancho (wide format): cada año es una columna (Y1961, Y1962, ...). Datos históricos globales y regionales de temperatura promedio.
FAOSTAT de Kaggle	csv	

Justificación: La combinación de fuentes garantiza consistencia, diversidad y profundidad temporal en el análisis climático.

4. INGESTA DE DATOS

APIs (OpenWeather / Meteostat):

- Consumo mediante requests.
- Conversión de JSON a DataFrame.
- Manejo de errores y reconexión ante fallos.

CSVs históricos:

- Lectura con `pandas.read_csv()`.
- Normalización de fechas y unidades.
- Validación de estructura, duplicados y valores nulos.
- Uso de la clase `Daily` de Meteostat para extraer registros entre 2021–2025.
- Los datos se transformaron en promedios mensuales.
- Se consolidaron en un DataFrame con las fuentes históricas.

5. CALIDAD Y TRANSFORMACIÓN DE DATOS

5.1. Profiling

Se realizaron las siguientes verificaciones:

- **Estructura de datos** (.info()).
- **Valores nulos** (.isnull().sum()).
- **Distribución de meses** (.value_counts()).

5.2. Diccionario de Datos

Columna	Descripción	Tipo	Ejemplo
area	País o región	String	Bolivia
meses	Mes del registro	String	Enero
year	Año del registro	Entero	2001
unit	Unidad de medida	String	°C
value	Temperatura media	Float	13.4
temp_categoria	Clasificación: Baja, Media o Alta temperatura	String	Media

5.3. Limpieza

- Eliminación de caracteres extraños como "â".
- Normalización de meses en inglés → español.
- Eliminación de duplicados, promediando valores.

6. PROCESAMIENTO Y CLASIFICACIÓN

- Cálculo de anomalías climáticas respecto a la media histórica de 9 ciudades bolivianas.
- Clasificación de variables:
 - Temperatura → frío, templado, cálido, caluroso.
 - Viento → calma, brisa, ventoso, fuerte.
- Derivación de variables para análisis comparativo y enriquecimiento del dataset.
- Se creó la función temp_categoria para clasificar los valores: **Baja:** < 5°C **Media:** 5°C – 15°C **Alta:** > 15°C.

Esto permite análisis comparativos y visualizaciones más intuitivas.

7. ALMACENAMIENTO DE DATOS

- **CSV procesado:** /data/processed/dataset_clima_limpio.csv

- **Base de datos SQLite:** `weather.db` con tabla `fact_weather`.
- **Reportes:** HTML / Excel para análisis colaborativo.
- Tablas listas para consultas posteriores con SQL o análisis en Python.

Justificación: Formatos estructurados permiten acceso rápido, integración con herramientas analíticas y mantenimiento del historial de versiones.

8. ARQUITECTURA Y PIPELINE ORQUESTADO

Arquitectura Actual – Batch Processing (24h)

- Los datos climáticos cambian cada poca hora y requieren un flujo programado y estable.
- Pipeline Orquestado con Airflow:

1. Extracción:

APIs: OpenWeather, Meteostat

CSV históricos: FAOSTAT, Environment Temperature Change

2. Transformación:

T-Environment: limpieza CSV Environment Temperature Change

T-FAO: limpieza CSV FAOSTAT

T-OpenWeather: transformación y validación JSON OpenWeather

T-Meteostat: transformación y validación JSON Meteostat

3. Carga:

Almacenamiento final en SQLite DB y CSV limpio

4. Reportes:

Generación automática de Diccionario de Datos y Data Quality Reports

9. DATA QUALITY REPORT Y DOCUMENTACIÓN

- **Documentación Técnica:** Pasos de ingesta, limpieza y almacenamiento, Diccionario de datos y definiciones, Registro de ejecución y dependencias en Airflow
- **Compleitud:** se detectaron y eliminaron valores nulos.
- **Consistencia:** se unificaron unidades en °C y nombres de meses.
- **Validez:** se filtraron registros con anomalías (ej. “Met” en meses).
- **Unicidad:** duplicados fueron consolidados con promedios.
- **Exactitud:** se cruzaron datos FAOSTAT y Meteostat para validar tendencias.

10. RESULTADOS

- Pipeline funcional y reproducible desde múltiples fuentes.
- Dataset limpio, validado y enriquecido.
- Reporte de calidad y documentación técnica generados automáticamente.
- Arquitectura orquestada y escalable.

11. LIMITACIONES

- **OpenWeather API:** límite diario de requests y restricciones en endpoints avanzados.
- **Dependencia de Meteostat:** interrupciones o brechas en estaciones meteorológicas.

12. CONCLUSIONES Y MEJORA CONTINUA

Se implementó un ETL robusto, documentado y automatizado. Se cumplieron los siete lineamientos clave: ingesta, calidad, procesamiento, orquestación, almacenamiento, arquitectura y reporte. El pipeline orquestado con Airflow garantiza ejecución controlada, escalabilidad y mantenimiento programado.

Futuras Mejoras:

1. Integrar un Data Warehouse (PostgreSQL / BigQuery)
2. Crear dashboards interactivos (Power BI o Grafana)
3. Extender a procesamiento en tiempo real (streaming) con Kafka
4. Implementar Docker y CI/CD para despliegue continuo
5. Mantener un ciclo de mejora continua con monitoreo de rendimiento y optimización de recursos.