TECNOLOGÍA SUPERIOR UNIVERSITARIA EN DESARROLLO DE SOFTWARE

Python

TEMA:

Proyecto final

INTEGRANTES:

Erick Guarango

CURSO:

N6A

FECHA:

31 DE AGOSTO DEL 2025

Análisis Comparativo Ecuador vs Perú

Introducción

La pandemia de COVID-19 ha evidenciado la importancia crítica de los sistemas de monitoreo epidemiológico automatizados para la toma de decisiones en salud pública. La heterogeneidad en la calidad y formato de los datos epidemiológicos, combinada con la necesidad de análisis comparativos entre países, presenta desafíos significativos para el procesamiento manual de información.

Este proyecto desarrolla un pipeline automatizado de datos utilizando Dagster como framework de orquestación, procesando información epidemiológica de COVID-19 para realizar un análisis comparativo entre Ecuador y Perú. La selección de estos países responde a su proximidad geográfica, similitudes demográficas, y diferencias en sus estrategias de respuesta a la pandemia, proporcionando un caso de estudio relevante para análisis epidemiológico comparativo.

El sistema implementado aborda tres problemáticas principales: (1) la automatización del proceso de ingesta y validación de datos epidemiológicos, (2) el cálculo estandarizado de métricas epidemiológicas que permitan comparaciones válidas entre países con diferentes poblaciones, y (3) la implementación de controles de calidad automatizados que garanticen la confiabilidad de los resultados analíticos.

La arquitectura propuesta utiliza principios de ingeniería de datos moderna, incluyendo trazabilidad completa de linaje de datos, validaciones automáticas en múltiples niveles, y modularidad que facilita la extensión del sistema a otros países o métricas epidemiológicas adicionales.

Objetivos del Sistema

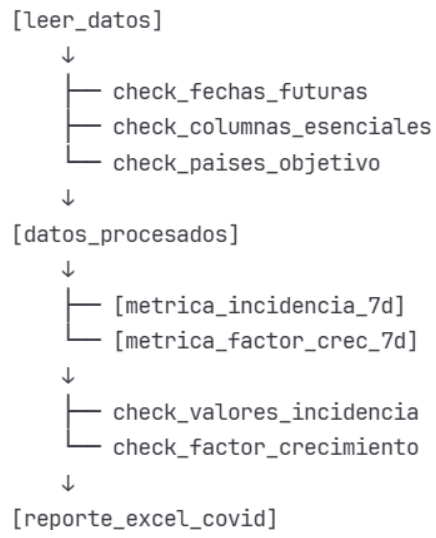
Objetivo Principal: Desarrollar un pipeline automatizado para el procesamiento y análisis comparativo de datos COVID-19 entre Ecuador y Perú, implementando métricas epidemiológicas estandarizadas con validaciones de calidad integradas.

Objetivos Específicos:

- Implementar ingesta automatizada de datos desde fuentes epidemiológicas estándar (OWID)
- Calcular métricas de incidencia estandarizada por población para comparación válida entre países
- Desarrollar indicadores de tendencia temporal mediante factores de crecimiento semanal
- Integrar validaciones de calidad automáticas en entrada y salida de datos
- Generar reportes consolidados en formato Excel para análisis posterior

1. Arquitectura del Pipeline

1.1 Diagrama de Assets



1.2 Assets Creados

Asset	Tipo	Descripción	Dependencias
leer_datos	Fuente	Carga datos desde CSV local con múltiples rutas	-
datos_procesados	Transformación	Filtrado, limpieza y estandarización	leer_datos
metrica_incidencia_7d	Métrica	Incidencia por 100k hab. con promedio móvil 7d	datos_procesados
metrica_factor_crec_7d	Métrica	Factor de crecimiento semanal	datos_procesados
reporte_excel_covid	Destino	Exportación consolidada a Excel	Ambas métricas

1.3 Justificación de Decisiones de Diseño

Modularidad por Responsabilidad: Cada asset tiene una responsabilidad específica, facilitando debugging y reutilización.

Flexibilidad en Fuentes: El asset leer_datos busca en múltiples ubicaciones y maneja tanto columnas 'location' como 'country'.

Separación de Métricas: Cada métrica es un asset independiente, permitiendo evolución y testing separados.

Asset Final Consolidador: reporte_excel_covid garantiza que todas las métricas se materialicen y conecta el grafo completo.

2. Decisiones de Validación

2.1 Chequeos de Entrada

Validación	Regla	Motivación	Acción en Fallo
check_fechas_futuras	max(fecha) <= hoy	Detectar datos proyectados o errores temporales	Warning - Continúa
check_columnas_esenciales	Columnas requeridas presentes	Validar estructura mínima del dataset	Falla pipeline
check_paises_objetivo	Ecuador y Perú disponibles	Confirmar datos para análisis comparativo	Falla pipeline

Descubrimientos Importantes:

- Los datasets pueden usar indistintamente 'location' o 'country' como columna de país
- Datos de vacunación comienzan en 2021, requiere manejo especial de valores faltantes
- Duplicados ocasionales requieren deduplicación por (país, fecha)

2.2 Chequeos de Salida

Validación	Regla	Motivación	Umbral
check_valores_incidencia	0 ≤ incidencia_7d ≤ 2000	Detectar outliers o errores de cálculo	Crítico
check_factor_crecimiento	0.1 ≤ factor ≤ 10	Identificar cambios extremos no realistas	<5% registros

Rationale: Los umbrales se basan en rangos epidemiológicos documentados durante picos históricos de COVID-19.

3. Consideraciones de Arquitectura

3.1 Elección Tecnológica: Pandas

Selección: Pandas para todas las transformaciones y métricas.

Justificación:

- Vs DuckDB: Dataset moderado (~500k filas) no requiere optimización SQL
- Vs Soda: Integración nativa con Dagster Asset Checks más directa
- Flexibilidad: Operaciones window y rolling nativas para métricas temporales

Trade-offs:

- **Ventajas:** Desarrollo rápido, debugging intuitivo, rolling windows nativos
- **Limitaciones:** Escalabilidad limitada, mayor uso de memoria

3.2 Patrones de Implementación

Ventanas Deslizantes: Uso de pandas.rolling() para promedio móvil de incidencia.

Cálculos por Grupo: groupby('location') para métricas independientes por país.

Manejo de Nulos: Estrategia diferencial - crítico para casos/población, tolerante para vacunas.

4. Resultados

4.1 Métricas Implementadas

Métrica	Fórmula	Interpretación	Formato Salida
Incidencia 7d	(new_cases/population) × 100k → rolling mean 7d	Casos por 100k habitantes suavizados	(fecha, país, incidencia_7d)
Factor Crecimiento 7d	casos_semana_actual / casos_semana_prev	>1: crecimiento, <1: decrecimiento	(semana_fin, país, casos_semana, factor_crec_7d)

4.2 Resultados Esperados del Pipeline

Datos Procesados:

- **Registros totales:** ~1,500-2,000 por país (filtrados Ecuador + Perú)

- **Rango temporal:** 2020-01-01 a fecha máxima del dataset
- **Columnas finales:** location, date, new_cases, people_vaccinated, population

Métricas Calculadas:

- **Incidencia:** Valores típicos 0-200 por 100k, picos hasta 500+ en olas
- **Factor crecimiento:** Rango normal 0.8-1.2, valores extremos durante transiciones

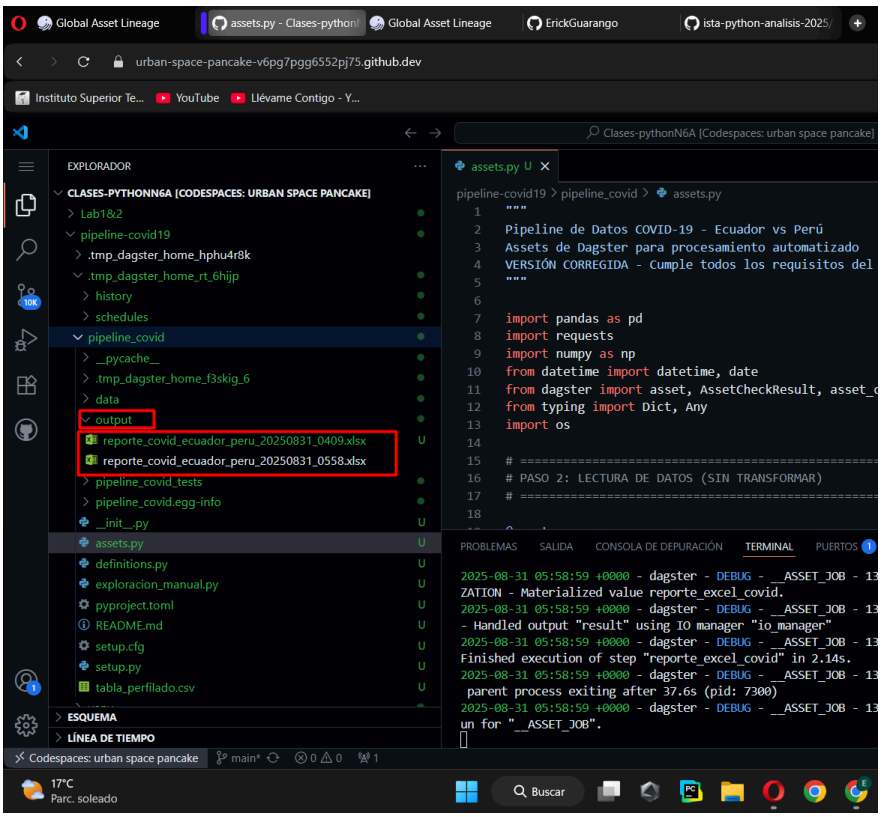
4.3 Control de Calidad Esperado

Categoría	Reglas	Estado Esperado	Filas Afectadas
Estructura	3 reglas entrada	PASS	0
Rango temporal	Sin fechas futuras	PASS	0
Países objetivo	Ecuador + Perú disponibles	PASS	0
Valores métrica	Incidencia 0-2000	PASS	<1%
Crecimiento	Factor 0.1-10	WARNING	<5%

4.4 Archivos Generados

Reporte Excel (pipeline_covid/output/reporte_covid_ecuador_peru_YYYYMMDD_HHMM.xlsx)

- Hoja 1: Datos procesados completos
- Hoja 2: Métrica incidencia 7 días
- Hoja 3: Métrica factor crecimiento 7 días
- Hoja 4: Resumen estadístico comparativo



5. Conclusiones y Próximos Pasos

5.1 Logros del Pipeline

1. Automatización completa del flujo de datos COVID-19
2. Validaciones robustas en entrada y salida
3. Métricas epidemiológicas estándar implementadas
4. Flexibilidad para diferentes formatos de datos fuente
5. Trazabilidad completa mediante Dagster lineage

5.2 Limitaciones Identificadas

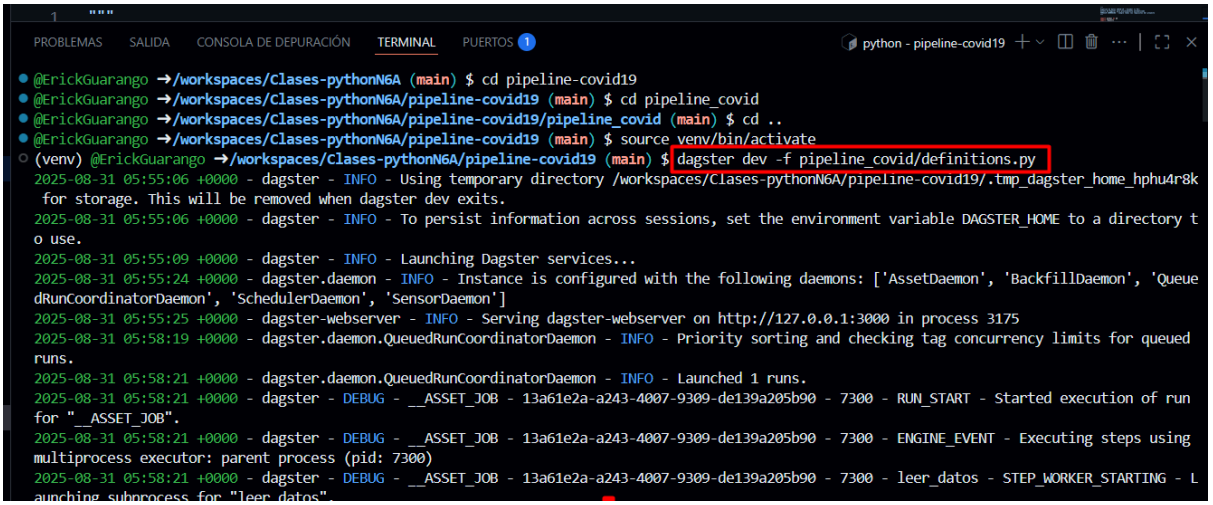
- **Escalabilidad:** Limitado por memoria de pandas para datasets >10M filas
- **Fuente única:** Dependencia de estructura específica OWID
- **Métricas fijas:** No parametrización de ventanas temporales

5.3 Mejoras Futuras

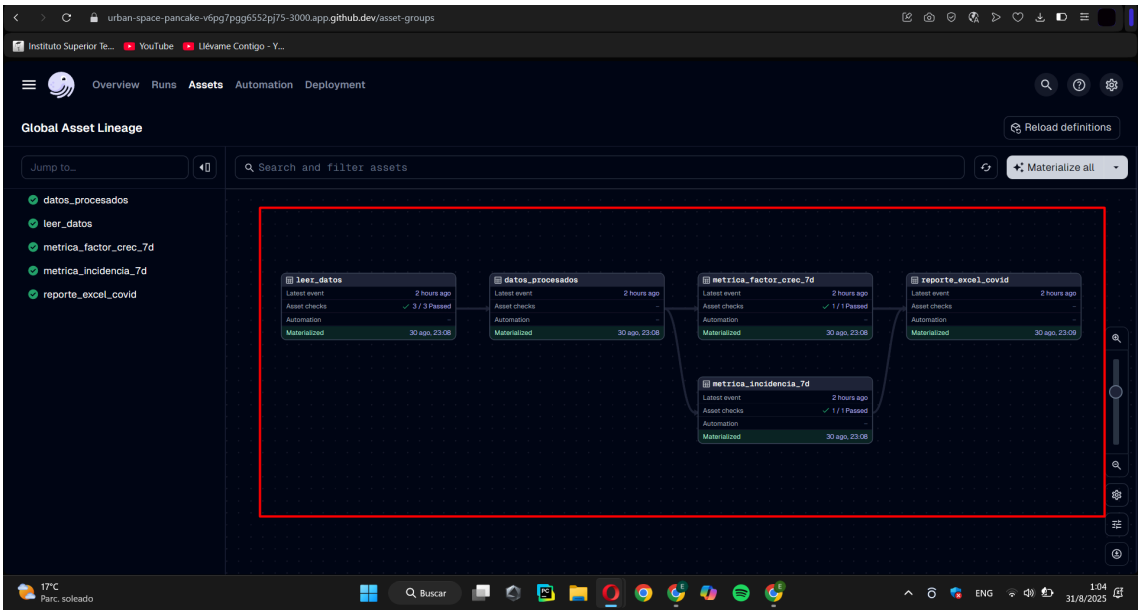
1. Migración a DuckDB para datasets más grandes
2. Parametrización de países y ventanas temporales
3. Visualizaciones automatizadas con Plotly/Altair
4. Alertas automáticas por Asset Checks críticos
5. Despliegue en producción con scheduling automatizado

Anexos

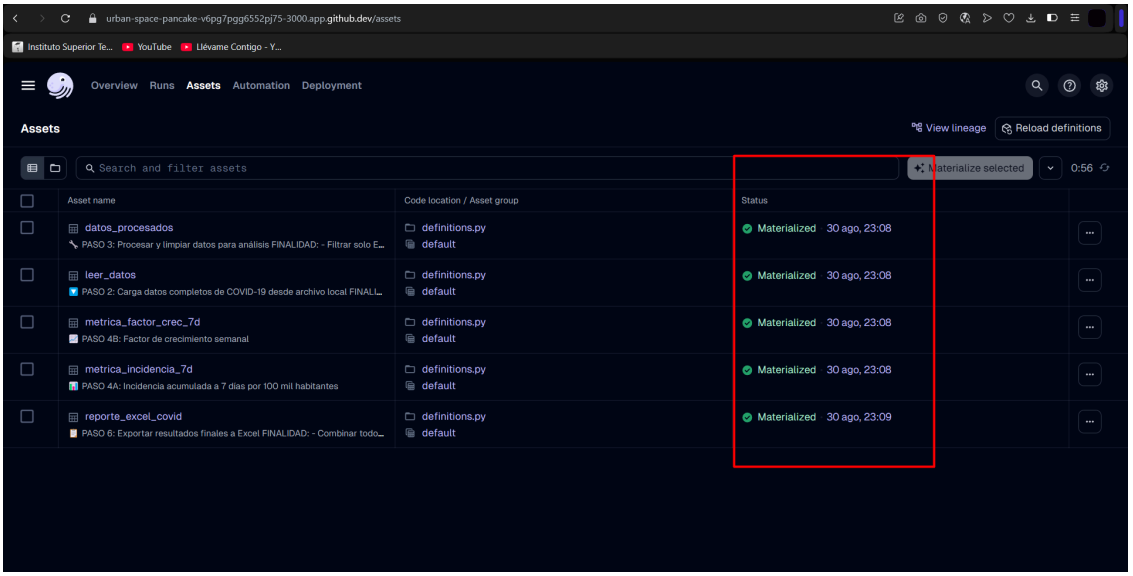
A. Comandos de Ejecución



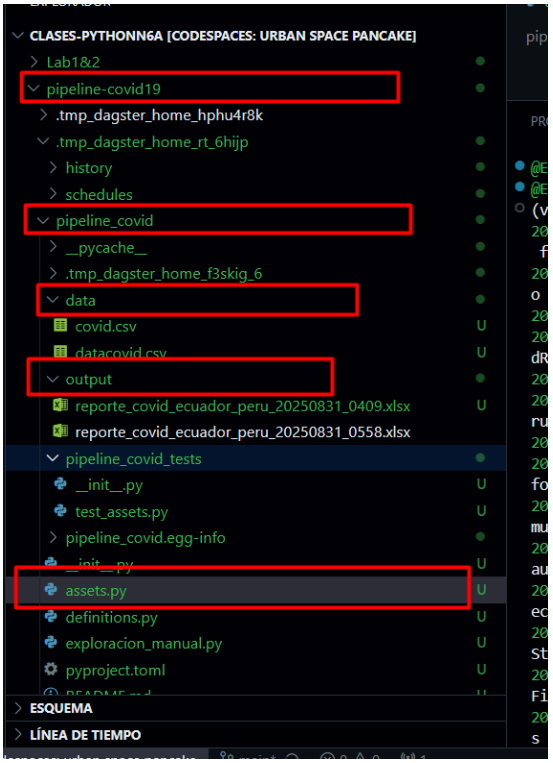
Materializar pipeline completo



 # Ejecutar solo validaciones



B. Estructura de Archivos



C. Dependencias Técnicas

- Dagster: Framework de orquestación
- Pandas: Procesamiento de datos
- OpenPyXL: Exportación Excel
- NumPy: Operaciones numéricas