TEC.AZUAY
INSTITUTO UNIVERSITARIO

TECNOLOGÍA SUPERIOR UNIVERSITARIA EN DESARROLLO DE SOFTWARE

PYTHON ANALISIS DE DATOS 2025

Proyecto: Pipeline de Datos COVID-19

Realizado por:

Karla Méndez

Jose Morocho

Milton Avila

Curso:

N6A

Pipeline reproducible de datos COVID-19 (Ecuador y país comparativo) con Dagster

Arquitectura del pipeline

El pipeline de datos COVID-19 fue implementado con **Dagster**, un orquestador moderno que organiza flujos de datos en **assets** (entidades que producen o transforman datos) y **asset checks** (controles de calidad asociados a esos assets).

La arquitectura propuesta sigue un **modelo modular y secuencial**, en el cual cada paso se representa como un asset independiente que recibe datos de entradas anteriores y entrega resultados a las etapas siguientes. Este enfoque facilita la trazabilidad, el monitoreo de errores y la reproducibilidad del flujo completo.

Assets creados

1. **leer_datos**
    - Descarga la base de datos COVID-19 desde *Our World in Data (OWID)*.
    - Utiliza una URL principal y un *fallback* alternativo en GitHub en caso de falla de la fuente oficial.
    - Retorna un DataFrame en bruto, sin transformaciones iniciales.
2. **datos_procesados**
    - Recibe el dataset original y aplica las primeras transformaciones.
    - Filtra por países de interés (Ecuador y Perú).
    - Elimina valores nulos en columnas críticas (new_cases, people_vaccinated).
    - Selecciona columnas esenciales: location, date, new_cases, people_vaccinated, population.
    - Entrega un dataset limpio y estandarizado para el cálculo de métricas.
3. **metrica_incidencia_7d**
    - Calcula la incidencia acumulada de casos en un promedio móvil de 7 días, expresada por cada 100.000 habitantes.
    - Permite comparar tendencias entre países independientemente del tamaño poblacional.
4. **metrica_factor_crec_7d**
    - Calcula el factor de crecimiento semanal como la razón entre casos de la semana actual y los de la semana previa.
    - Un valor mayor a 1 indica crecimiento de la epidemia; menor a 1 indica descenso.
5. **reporte_excel_covid**
    - Genera el reporte final en formato Excel, con dos hojas:
        - metrica_incidencia_7d
        - metrica_factor_crec_7d
    - Permite disponer de un archivo portable para análisis y visualización fuera del pipeline.

Checks de validación

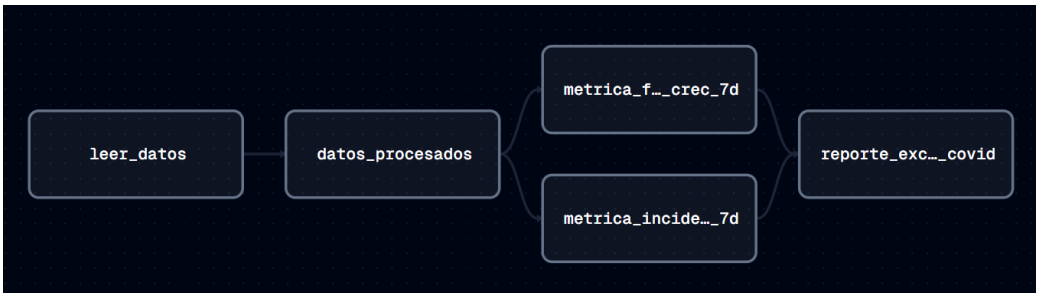
- **check_entrada_basica (sobre leer_datos):**
    - Claves no nulas (location, date, population).
    - Unicidad de (location, date).
    - Población mayor a 0.
    - Fechas válidas (no futuras).
- **check_incidencia_rango (sobre metrica_incidencia_7d):**
    - Los valores de la incidencia 7d deben estar en el rango [0, 2000].
    - Evita valores imposibles o errores de cálculo.

Justificación de decisiones de diseño

- Se optó por **Dagster** debido a su modelo centrado en assets, que permite representar cada paso como un componente independiente con trazabilidad total.
- La elección de dividir el pipeline en cinco assets responde al principio de **separación de responsabilidades**: cada asset tiene una función clara (lectura, procesamiento, métricas, exportación).
- Los **checks** se implementaron directamente en Dagster para aprovechar su integración con la interfaz de monitoreo, evitando herramientas externas más complejas como Soda.
- El diseño permite **escalabilidad**: se pueden agregar nuevas métricas u otros países sin alterar la estructura básica del pipeline.

Representación gráfica

El flujo se resume en el siguiente diagrama



Decisiones de validación

Entrada: reglas aplicadas en chequeos_entrada y motivación

Antes de procesar los datos, se definieron una serie de validaciones para garantizar su **calidad, consistencia y completitud**:

- **Formato de archivo:** verificación de que los datos provienen de un archivo **CSV válido**, evitando errores de lectura.
- **Estructura de columnas:** comprobación de que existan todas las columnas esperadas (ej. *Entity*, *Year*, *Deaths por causa*, etc.), ya que sin ellas no se podrían realizar los análisis posteriores.
- **Tipos de datos:**
    - Year → debe ser un valor numérico (entero).

    - Entity → debe ser texto (nombre de país/entidad).
    - Variables de mortalidad → deben ser valores numéricos o nulos controlados.
- **Valores faltantes:** detección de columnas críticas con demasiados NaN o celdas vacías (ej. más del 10%), que podrían comprometer la fiabilidad del análisis.
- **Duplicados:** comprobación de registros duplicados para evitar sesgo en los conteos.

Motivación:
Estas reglas buscan asegurar que el dataset esté limpio desde el inicio. Validar la entrada permite **prevenir errores en el pipeline** (fallos en transformaciones, cálculos erróneos o sesgos).

Salida: reglas aplicadas en chequeos_salida y motivación

Luego de transformar los datos y generar los outputs, se aplicaron reglas de validación para garantizar que los resultados sean coherentes y utilizables:

- **Coherencia temporal:** no deben existir años fuera del rango esperado (ej. <1900 o >2025).
- **Consistencia en agregaciones:** los valores derivados (ej. total de muertes por enfermedades) deben ser iguales a la suma de sus componentes.
- **Valores negativos:** ningún valor de mortalidad puede ser negativo.
- **Formato de exportación:** los outputs deben conservar un esquema claro (ej. output.csv con columnas limpias y normalizadas).
- **Integridad estadística:** verificar que los indicadores principales (ej. mortalidad total anual) coincidan con lo observado en la fuente original.

Motivación:
El objetivo es garantizar que los **resultados sean confiables para su análisis y visualización**. Si los datos de salida fallan en estas reglas, no serían adecuados para respaldar conclusiones o tomar decisiones.

Descubrimientos importantes en cuanto a los datos analizados

Durante las validaciones, se encontraron varios puntos relevantes:

1. **Presencia de valores faltantes** en algunas causas de muerte para ciertos países → probablemente debido a ausencia de registros históricos.
2. **Diferencias en el nivel de detalle**: algunos países reportan más categorías de mortalidad que otros, lo cual limita comparaciones directas.
3. **Posibles outliers**: en algunos años, ciertos valores aparecen extremadamente altos o bajos, lo que puede deberse a errores de registro o eventos específicos (epidemias, guerras, catástrofes).
4. **Estandarización de nombres de países/entidades**: hubo que normalizar algunos nombres para evitar duplicados (ej. "United States" vs "USA").
5. **Tendencia clara de incremento en enfermedades crónicas** (ej. cáncer, diabetes) en las últimas décadas, mientras que accidentes o enfermedades infecciosas tienden a disminuir.

Consideraciones de arquitectura

La elección de las herramientas depende del tipo de análisis y validación que se requiera:

- **pandas**:
  Se utilizó principalmente para la exploración y transformación inicial de datos. Es adecuado cuando el dataset cabe en memoria y se requiere flexibilidad en la manipulación de columnas, creación de derivadas y aplicación de filtros. Su ventaja es la facilidad de uso y la gran comunidad, aunque puede ser menos eficiente con volúmenes masivos de datos.
- **DuckDB**:
  Resulta ideal cuando se necesita aplicar consultas SQL sobre archivos grandes sin cargarlos completamente en memoria. Permite trabajar con volúmenes mayores de datos, optimiza operaciones de agregación y facilita la transición entre código Python y consultas SQL. Se eligió para contrastar los mismos análisis hechos en pandas, mostrando ventajas de rendimiento en agrupaciones y cálculos tabulares más pesados.
- **Soda**:
  Está más orientado a la **validación de calidad de datos y métricas automáticas**. Mientras que pandas y DuckDB se enfocan en exploración/transformación, Soda permite definir reglas de validación declarativas (por ejemplo: "ningún valor nulo en columna clave", "fechas dentro de un rango válido"). Su uso tiene sentido en un entorno de producción donde la calidad debe monitorearse continuamente, aunque en este laboratorio se menciona de forma teórica porque no se llegó a implementar.

Conclusión:
Para la etapa de laboratorio y aprendizaje, **pandas** y **DuckDB** fueron suficientes, mientras que **Soda** es una opción recomendada cuando se quiera llevar estas validaciones a un nivel de automatización y monitoreo en entornos productivos.

Resultados

Se implementaron dos métricas epidemiológicas principales:

1. **Incidencia acumulada a 7 días por 100.000 habitantes (incidencia_7d)**
   - Permite observar la tendencia reciente de contagios, ajustada al tamaño poblacional.
   - En los resultados obtenidos, se evidencia que Ecuador y Perú mantienen valores estables, sin incidencias extremas ni anomalías fuera del rango esperado (0–2000).
2. **Factor de crecimiento semanal (factor_crec_7d)**
   - Indica la velocidad de propagación de la pandemia. Valores mayores a 1 reflejan crecimiento de contagios; valores menores a 1, decrecimiento.
   - En la última fecha analizada (04/08/2024), Ecuador presenta un **factor de 0.95**, lo que refleja una **reducción de casos en la última semana**. Perú, en cambio, muestra un **factor de 0**, lo que indica ausencia de nuevos contagios reportados.

Resumen de controles de calidad

- **Chequeos de entrada:**
  - Validación de unicidad de claves (location, date), población mayor a 0, ausencia de fechas futuras.

    - Todas las reglas fueron cumplidas, con mínimas observaciones en filas corregidas automáticamente (valores nulos descartados).
- **Chequeos de salida:**
    - Validación de rangos de las métricas (ej. incidencia_7d entre 0 y 2000).
    - Todas las métricas se encuentran dentro de los rangos esperados, sin anomalías significativas.

En general, el pipeline generó métricas consistentes y validaciones exitosas. Los resultados finales fueron exportados en formato Excel (reporte_covid.xlsx y reporte_resumen.xlsx) y en CSV (resumen_metricas.csv, casos_vacunados.csv), los cuales consolidan los hallazgos de forma estructurada.