

Informe Técnico Análisis de Datos Con Dagster

El informe respectivo detalla a breves rasgos la estructura del proyecto y la solución aplicada de modo que se pueda interpretar y explicar a detalle la solución:

Assets			
	Asset name	Code location / Asset group	Status
<input checked="" type="checkbox"/>	datos_procesados	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	grafica_factor_crec_7d	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	grafica_incidencia_7d	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	leer_datos	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	metrica_factor_crec_7d	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	metrica_incidencia_7d	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	reporte_excel_covid	└ covid_pipeline └ default	↳ Materializing... 7acc7beb
<input checked="" type="checkbox"/>	tabla_perfilado	└ covid_pipeline └ default	↳ Materializing... 7acc7beb

- **covid_pipeline/assets/**
 - datos.py : ingesta y procesamiento inicial (leer_datos, datos_procesados)
 - metricas.py: cálculos de metrica_incidencia_7d y metrica_factor_crec_7d
 - reportes.py: exportación a Excel (reporte_excel_covid)
 - graficas.py: generación de gráficos (grafica_incidencia_7d, grafica_factor_crec_7d)
 - eda.py: asset de exploración (tabla_perfilado)
 - checks.py: asset checks (entrada/salida)
 - definitions.py: orquestador, carga todos los assets y checks
- **reportes/**
 - reporte_covid.xlsx: artefacto final con 3 hojas (datos, incidencia, factor crecimiento)
 - En este segmento también se generar
- **Archivos raíz**
 - .gitignore: asegura que no se suba __pycache__, venv, etc.
 - requirements.txt: dependencias (dagster, pandas, matplotlib, etc.).
 - pyproject.toml, setup.py, setup.cfg: metadatos del proyecto (opcionales pero correctos).
 - tabla_perfilado.csv: artefacto ligero con el EDA básico (obligatorio para entrega).
 - Informe_Covid.md: aquí puedes redactar el informe en Markdown y luego exportarlo a PDF.
 - Descripción de assets creados para los pasos. Recomendado incluir diagrama de las secciones respectivas.
 - Justificación de decisiones de diseño basadas en el modelado y lista de **assets** creados.

Dentro de la arquitectura **Assets/checks.py** se mantienen definidas funciones de verificación, cada función está decorada con `@asset_check` y se ejecuta sobre un asset en particular (ej. `leer_datos`, `metrica_incidencia_7d`).

Estas funciones devuelven un objeto AssetCheckResult que Dagster muestra en su interfaz como (pasó o falló).

En **Assets/definitions.py** se encuentra load_assets_from_modules([...]), se puede considerar una función que llama a cada uno de los archivos datos.py, métricas.py y recoge todas las funciones decoradas con @asset. De modo que no se necesita importarlas una por una.

- De datos.py trae leer_datos y datos_procesados.
- De metricas.py trae metrica_incidencia_7d, metrica_factor_crec_7d.
- De eda.py trae tabla_perfilado.
- De graficas.py trae grafica_incidencia_7d, grafica_factor_crec_7d.
- De reportes.py trae reporte_excel_covid.

En la estructura Assets/eda.py (Exploración / Perfilado): Recibe el DataFrame leer_datos

Es la salida del asset de ingesta (leer_datos), que descarga y normaliza los datos de COVID-19 de Our World in Data.

- Construye un perfil descriptivo en varias partes:
- Columnas y tipos (tipos): lista las columnas disponibles y sus tipos de datos (object, float64, etc.).
- Mínimo y máximo de new_cases (min_max): identifica el rango de casos diarios reportados.
- Porcentaje de nulos (nulos): calcula el % de valores faltantes en new_cases y people_vaccinated.
- Rango de fechas (fechas): obtiene la fecha más antigua y la más reciente en la columna date.

Concatena toda la información en una sola tabla (perfil).

Exporta un artefacto ligero tabla_perfilado.csv

- Se guarda en la raíz del proyecto.
- Este archivo es parte de la entrega obligatoria (artefacto versionable en Git).

Devuelve el DataFrame perfil así Dagster puede mostrarlo en la interfaz y usarlo en otros assets si fuera necesario.

2. Decisiones de validación

- Leer datos: Aseguran que los datos descargados de OWID sean consistentes antes de procesarlos:
 1. max_date_not_future: Verifica que la fecha máxima en el dataset no esté en el futuro. Ejemplo: si aparece 2025-12-31, levanta advertencia.
 2. keys_not_null: Revisa que las columnas clave (location, date, population) no tengan valores nulos. Ejemplo: detecta si hay filas donde location está vacío.
 3. unique_loc_date: Chequea que no haya duplicados en (location, date). Ejemplo: si para Ecuador el mismo día aparecen 2 registros, falla.

4. population_positive: Valida que population > 0. Ejemplo: si algún país aparece con población = 0, se reporta.
5. new_cases_nonnegative: Asegura que new_cases >= 0. (En OWID a veces hay negativos por revisiones: en tu código puedes marcar como *warning* y no bloquear el run).
6. Salida: reglas aplicadas en chequeos_salida y motivación.

3. Consideraciones de arquitectura

- En este proyecto se optó por **pandas** junto con los **Dagster Asset Checks** para todas las transformaciones y validaciones de datos.
- **pandas** se utilizó para:
 - Cargar y limpiar el dataset de OWID (leer_datos, datos_procesados).
 - Calcular métricas como la **incidencia acumulada a 7 días por 100k** y el **factor de crecimiento semanal** mediante operaciones de groupby, rolling y transformaciones vectorizadas.
 - Generar artefactos ligeros (tabla_perfilado.csv, reporte_covid.xlsx) y gráficos comparativos.
- **Dagster Asset Checks** se emplearon para:
 - Validar la calidad de los datos de entrada (max_date_not_future, keys_not_null, unique_loc_date, population_positive, new_cases_nonnegative).
 - Controlar que las métricas generadas fueran razonables (incidencia_en_rango, factor_crec_valido).
 - Mostrar los resultados de estas validaciones directamente en la **UI de Dagster**, facilitando la observabilidad y trazabilidad del pipeline.

Observaciones: No se utilizó DuckDB ni Soda, ya que el tamaño del dataset y los requerimientos académicos son manejables con pandas, y Dagster ya ofrece las capacidades de validación necesarias a través de Asset Checks.

4. Resultados

4.1 Métricas implementadas

1. Incidencia acumulada 7d por 100k hab.

- Permite comparar contagios en Ecuador y Perú de manera proporcional a su población.
- En la gráfica se observa que Perú tuvo picos de incidencia más altos (ej. >140 casos/100k en 2022), mientras que Ecuador se mantuvo en niveles más bajos, aunque con tendencias similares en los picos pandémicos.
- Interpretación: la evolución de brotes es similar en ambas curvas, pero con mayor intensidad relativa en Perú.

2. Factor de crecimiento 7d

- Mide la velocidad del brote.
- Valores >1 indican crecimiento, <1 descenso.

- En la gráfica se aprecia que ambos países alternaron fases de crecimiento y decrecimiento. Al inicio de la pandemia los factores fueron muy altos (>10 en Ecuador, >7 en Perú), debido a la baja base de casos semanales. Posteriormente se estabilizaron en rangos entre 0.5 y 2.
- Interpretación: tras los primeros meses, las oscilaciones reflejan olas de contagio, pero con una dinámica parecida entre países.

3. Tabla de perfilado (EDA)

- Reveló que:
 - new_cases tiene valores mínimos de 0 y máximos muy altos (coherente con reportes de olas).
 - Existe un % de nulos en people_vaccinated (debido a cobertura irregular de reportes).
 - El rango temporal incluye registros hasta fechas ligeramente futuras (advertencia, pero no crítico).

4.2 Resumen del control de calidad

Regla	Estado	Filas afectadas	Comentario
<code>max_date_not_future</code>	Warning	1 registro	El dataset trae fechas futuras de OWID; no invalida análisis.
<code>keys_not_null (location, date, population)</code>	Warning	~16,958 filas	Valores faltantes en claves (población o fecha) para ciertos países con cobertura incompleta.
<code>unique_loc_date</code>	Passed	0	No hubo duplicados en (location, date).
<code>population_positive</code>	Passed	0	Todas las poblaciones >0.
<code>new_cases_nonnegative</code>	Passed	0	No se detectaron casos negativos.
<code>incidencia_en_rango [0–2000]</code>	Passed	0	Todas las incidencias están dentro de límites razonables.
<code>factor_crec_valido</code>	Warning	Ventanas iniciales por país	Los primeros 7 días generan divisiones inválidas (esperado).

Conclusión del control de calidad:

- Todas las validaciones críticas pasaron (no hay duplicados, poblaciones correctas, casos no negativos).

Nombre: Bryam Romero

Curso: Sexto Ciclo

Email: bryam.romero.est@tecazuay.edu.ec

- Las advertencias corresponden a situaciones típicas del dataset OWID: registros con fechas futuras por actualizaciones, valores nulos en algunos países y factores inválidos en ventanas iniciales.
- El pipeline es confiable, ya que los problemas detectados no afectan la interpretación de las métricas finales.

