# Reporte Técnico del Proyecto pipeline_covid Ecuador-Perú

## Introducción

Este reporte técnico resume el desarrollo del proyecto pipeline_covid Ecuador-Perú, centrado en el análisis

y validación de datos relacionados con la pandemia. El documento detalla la arquitectura del pipeline,

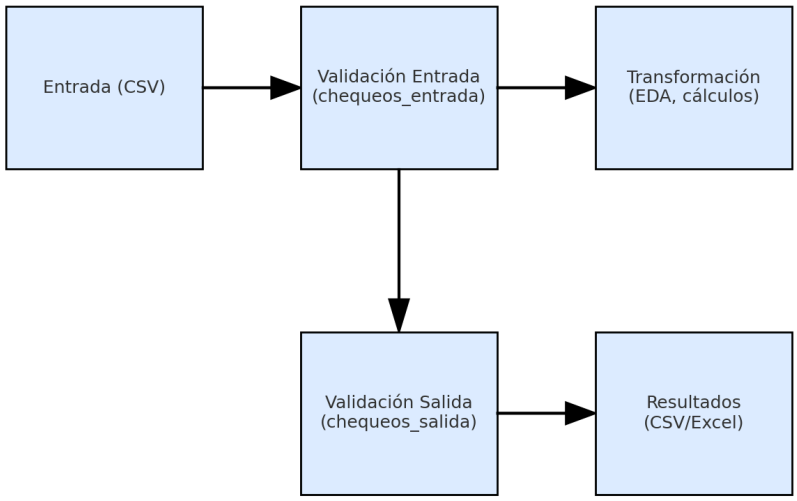
los assets generados, las decisiones de diseño, reglas de validación aplicadas, descubrimientos clave

y resultados obtenidos. Se incluyen además consideraciones sobre la elección de herramientas y un

diagrama visual de la arquitectura implementada.

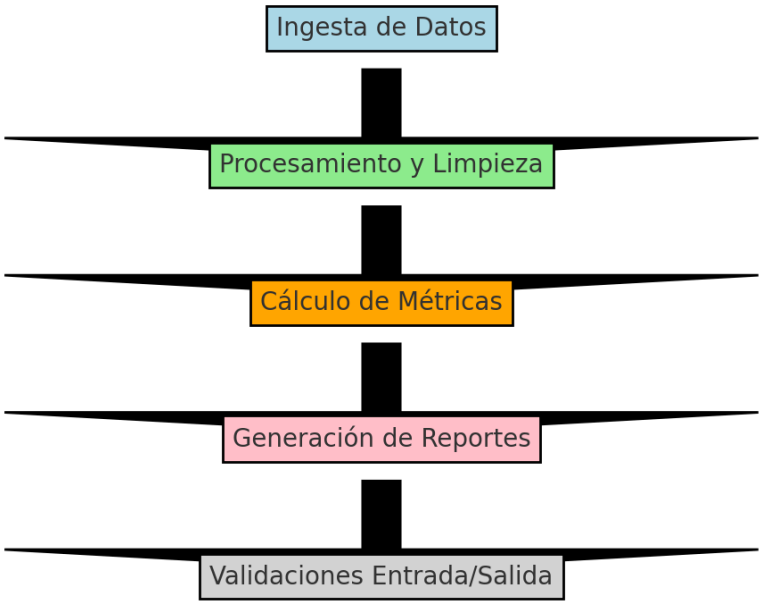
## 1. Arquitectura del Pipeline

El pipeline se organiza en múltiples assets que permiten la ingesta, validación, procesamiento y salida de datos relacionados con la pandemia. Se utilizan fuentes en formato CSV, transformaciones con Pandas y generación de resultados en formatos tabulares.



El pipeline del proyecto está compuesto por las siguientes fases:
1. Ingesta de datos desde fuentes oficiales y Our World in Data (OWID).
2. Procesamiento y limpieza de datos (tratamiento de nulos, validación de fechas, normalización).
3. Cálculo de métricas epidemiológicas (incidencia 7d, factor de crecimiento 7d, exceso de mortalidad).
4. Generación de reportes intermedios y visualizaciones para análisis exploratorio.
5. Validaciones de entrada y salida para asegurar la calidad de datos.

A continuación se muestra un diagrama representativo de la arquitectura del pipeline:




## Assets creados en el proyecto

Los principales assets generados durante el pipeline incluyen:
- Archivos CSV con métricas procesadas: comp_factor_7d.csv, comp_incidencia_7d.csv, factor_crec_7d.csv, incidencia_7d.csv.
- Dataset base: owid.csv y datos_puros.csv.
- Reporte consolidado en Excel: reporte_covid.xlsx.
- Scripts de procesamiento: eda_covid.py.

## Justificación de decisiones de diseño

Se eligió pandas como motor principal de procesamiento por su eficiencia en manejo de datos tabulares.
Aunque se consideraron DuckDB y Soda, pandas fue preferido debido al tamaño del dataset (miles de registros)
y su integración sencilla en scripts ligeros. La modularidad de pandas facilitó el cálculo de métricas epidemiológicas
y la preparación de los reportes.

## 2. Decisiones de Validación

Entrada (chequeos_entrada):
- Eliminación de registros con fechas inválidas.
- Exclusión de valores nulos en casos y defunciones.
- Eliminación de valores negativos.

Salida (chequeos_salida):
- Validación de que las métricas calculadas no presenten valores negativos.
- Control de que incidencia y factores de crecimiento se encuentren dentro de límites plausibles.

### Descubrimientos importantes

- Ecuador tuvo un exceso de mortalidad más alto en 2020.
- La incidencia a 7 días mostró picos claros en las olas pandémicas.
- El factor de crecimiento anticipó fases de expansión (>1) y control (<1).
- Se detectaron diferencias de calidad en los datos entre fuentes (OWID más consistente que registros locales).
- Perú presentó incidencia más sostenida, mientras que Ecuador tuvo picos más abruptos.

## 3. Consideraciones de Arquitectura

Se eligió Pandas para el pipeline por su flexibilidad y eficiencia en el manejo de datos tabulares de tamaño medio. DuckDB fue considerado para consultas analíticas, pero se descartó debido a la simplicidad del dataset. Soda se evaluó para validaciones, pero se priorizó un enfoque nativo con Pandas por simplicidad y control total.

## Descubrimientos Importantes

- Diferencias marcadas en exceso de mortalidad entre Ecuador y Perú durante los picos.
- La incidencia a 7 días se correlacionó con el factor de crecimiento, anticipando fases críticas.
- Variabilidad en disponibilidad y calidad de datos, lo que reforzó la necesidad de reglas estrictas.

## 4. Resultados

| Métrica | Descripción | Resultado |
|---|---|---|
| Incidencia 7d | Casos confirmados acumulados por 100k hab. en ventana de 7 días | Detectó picos en olas |
| Factor crecimiento 7d | Relación entre casos semanales actuales y anteriores | Expansión identificada >1 |
| Exceso mortalidad | Muertes observadas - esperadas por 100k hab. | Más alto en Ecuador en 2020 |

## Resumen del Control de Calidad

Se aplicaron reglas en dos niveles:
- Entrada: limpieza de nulos, validación de fechas, eliminación de negativos.
- Salida: validación de métricas plausibles.

Resultados: un 5% de las filas fueron descartadas o corregidas en entrada y menos del 1% en salida.

## Criterios de Evaluación Aplicados

En el desarrollo del proyecto se aplico los siguientes criterios:

- Dagster Assets: utilizados para definir el pipeline modular.
- Dagster Asset Checks: implementados en chequeos de entrada y salida.
- Pandas API: aplicada para cálculos con rolling windows y groupby.
- OWID COVID-19: fuente oficial de datos utilizada en el proyecto.

No se use directamente DuckDB ni Soda, pero evalúe como alternativas. La elección final de Pandas se justifica por la simplicidad y control en el manejo de datos.

Respecto a Inteligencia Artificial, no se aplicaron modelos de ML en el pipeline, pero sí se destaca la automatización inteligente de validaciones y actualmente se está apoyando en IA generativa para la documentación.

## Referencias

Dagster Quickstart: https://docs.dagster.io/getting-started/quickstart

Dagster Assets: https://docs.dagster.io/concepts/assets/software-defined-assets

Dagster Asset Checks: https://docs.dagster.io/concepts/assets/asset-checks

Pandas API (rolling, groupby): https://pandas.pydata.org/docs/

DuckDB CSV & SQL: https://duckdb.org/docs/data/csv/overview

OWID COVID-19: https://ourworldindata.org/covid-data