INSTITUTO SUPERIOR TECNOLOGICO DEL AZUAY

Tecnología Superior Universitaria En Desarrollo De Software

Nombres: Edwin Morocho

Curso: N6A

1. Arquitectura del Pipeline

1.1 Diagrama de Assets y Dependencias

leer_datos

- ├── chequeos_entrada
- └── datos_procesados (depende también de chequeos_entrada)
  - ├── metrica_incidencia_7d
  - ├── metrica_factor_crec_7d
  - ├── tabla_perfilado
  - └── chequeos_salida (depende de las métricas)
    - └── reporte_excel_covid (depende de todo)

1.2 Assets Implementados

ASSET	TIPO	PROPOSITO	SALIDA
leer_datos	Ingesta	Descarga CSV desde OWID con URLs de respaldo	DataFrame crudo (~300k filas)
chequeos_entrada	Validación	Verifica calidad de datos crudos	DataFrame con 4 reglas de validación
datos_procesados	Transformación	Limpia y filtra datos para Ecuador/Colombia	DataFrame filtrado (~8k filas)
metrica_incidencia_7d	Métrica	Calcula incidencia por 100k habitantes (promedio móvil 7d)	DataFrame con series temporales
metrica_factor_crec_7d	Métrica	Calcula factor de crecimiento semanal de casos	DataFrame con ratios de crecimiento
chequeos_salida	Validación	Verifica rangos válidos en métricas calculadas	DataFrame con 2 reglas de validación
tabla_perfilado	Reporte	Genera estadísticas descriptivas básicas	CSV para committear
reporte_excel_covid	Exportación	Consolida todos los resultados	Archivo Excel con 6 hojas

1.3 Justificación de Decisiones de Diseño

Separación de Responsabilidades: Cada asset tiene una función específica, facilitando debugging y reutilización.

Detección Automática de Columnas: El pipeline es robusto ante cambios en el esquema del dataset, buscando automáticamente columnas equivalentes (location, country, entity).

URLs de Respaldo: Implementamos múltiples fuentes de datos para garantizar disponibilidad:

- GitHub Raw (principal)
- Catálogo OWID (respaldo)
- Dominio COVID OWID (respaldo)

Validaciones como Assets: En lugar de Asset Checks (por compatibilidad), las validaciones se implementaron como assets normales que generan reportes estructurados.

2. Decisiones de Validación

2.1 Chequeos de Entrada (chequeos_entrada)

Regla	Motivación	Implementación
fechas_no_futuras	Detectar errores de carga o inconsistencias temporales	max(date) ≤ hoy
columnas_clave_validas	Asegurar que existan campos esenciales para el análisis	Verificar location, date, population
unicidad_location_date	Prevenir duplicados que distorsionen métricas	duplicated(location, date).sum() == 0
poblacion_positiva	Validar denominadores para cálculos per cápita	population > 0

Resultado de Implementación: El sistema detectó automáticamente las columnas correctas y no encontró anomalías críticas en los datos de OWID.

2.2 Chequeos de Salida (chequeos_salida)

Regla	Motivación	Rango Válido
incidencia_7d_rango_valido	Detectar valores extremos irrealistas en incidencia	[0, 2000] casos por 100k habitantes
factor_crecimiento_valido	Identificar divisiones por cero o valores infinitos	> 0 y finito

Decisión de Rangos: El límite de 2000 para incidencia se basó en los picos históricos más altos registrados durante la pandemia.

2.3 Descubrimientos Importantes

1.- Calidad de Datos OWID: Los datos presentaron alta calidad con mínimos valores faltantes para Ecuador y Colombia.

2.- Detección Automática: El pipeline detectó exitosamente las siguientes columnas:

- Location: location
- Date: date
- Cases: new_cases
- Vaccination: people_vaccinated_per_hundred
- Population: population

3.- Cobertura Temporal: Los datos abarcan desde enero 2020 hasta agosto 2025, proporcionando una serie temporal completa.

3. Consideraciones de Arquitectura

3.1 Elección Tecnológica: Pandas vs. DuckDB vs. Soda

Decisión: Pandas

Ventajas:

- Ecosistema maduro y familiar para análisis de datos
- Funciones rolling() nativas para promedios móviles
- Integración directa con Dagster
- Capacidad de procesamiento suficiente para el volumen de datos (~8k filas filtradas)

Alternativas Consideradas:

DuckDB: Ofrecería mejor rendimiento para datasets grandes (>1M filas), pero introduce complejidad innecesaria

Soda: Excelente para validaciones complejas, pero requiere configuración adicional y el enfoque de "validaciones como assets" es más transparente

4. Resultados

4.1 Métricas Implementadas

Métrica	Fórmula	Interpretación	Ventana Temporal
Incidencia 7d	(new_cases / population) * 100000 → promedio móvil 7d	Casos por 100k habitantes, suavizado	Diaria, promedio 7 días
Factor Crecimiento 7d	casos_semana_actual / casos_semana_anterior	>1: crecimiento, <1: decrecimiento	Semanal, con desfase 7 días

4.2 Interpretación de Resultados

Incidencia 7 días:

- Estandariza por población, permitiendo comparación directa Ecuador vs. Colombia
- Promedio móvil reduce el ruido de variaciones diarias (reportes weekends, días festivos)
- Valores típicos: 0-50 (endemia), 50-200 (epidemia), >200 (crisis)

Factor de Crecimiento 7 días:

- Factor = 1.0: casos estables
- Factor > 1.2: crecimiento acelerado, requiere atención
- Factor < 0.8: decrecimiento sostenido

4.4 Archivos Generados

1.- tabla_perfilado.csv - Estadísticas descriptivas para commitear

2.- reporte_covid_ecuador.xlsx - Reporte consolidado:

- Datos procesados
- Métricas de incidencia
- Métricas de crecimiento
- Chequeos de entrada
- Chequeos de salida
- Resumen ejecutivo

5. Conclusiones y Recomendaciones

Logros del Pipeline:

- Robustez: Sistema tolerante a cambios en esquema de datos
- Transparencia: Validaciones explícitas y documentadas
- Escalabilidad: Arquitectura modular permite agregar nuevos países/métricas
- Calidad: Control de calidad integral en entrada y salida

El pipeline proporciona una base sólida para monitoreo epidemiológico automatizado, con métricas válidas y controles de calidad rigurosos. La comparación Ecuador-Colombia permite insights sobre políticas públicas diferenciadas y sus efectos en la propagación viral.