Technology Posts Analytics

Implementación de un Pipeline DataOps

Miguel Ángel Hernández Vargas

Escuela Colombiana de Ingeniería

5 de diciembre de 2025

Agenda

Contexto del Proyecto

Las redes sociales se han convertido en una de las fuentes principales para identificar tendencias tecnológicas. Sin embargo, trabajar con estos datos presenta desafíos importantes:

- **Volumen:** miles de interacciones por minuto.
- **Ruido:** información no estructurada.
- **Volatilidad:** picos de viralidad impredecibles.

¿Cómo convertir el flujo masivo de datos sociales en un sistema automatizado y preciso?

Se necesita un sistema que asegure calidad y confiabilidad en el monitoreo del desempeño.

Objetivos del Proyecto

Objetivo General

Diseñar e implementar un pipeline DataOps, aplicando principios de automatización, calidad y gobernanza, para analizar la correlación entre la actividad social digital y las métricas de desempeño en el sector tecnológico.

Objetivos Específicos

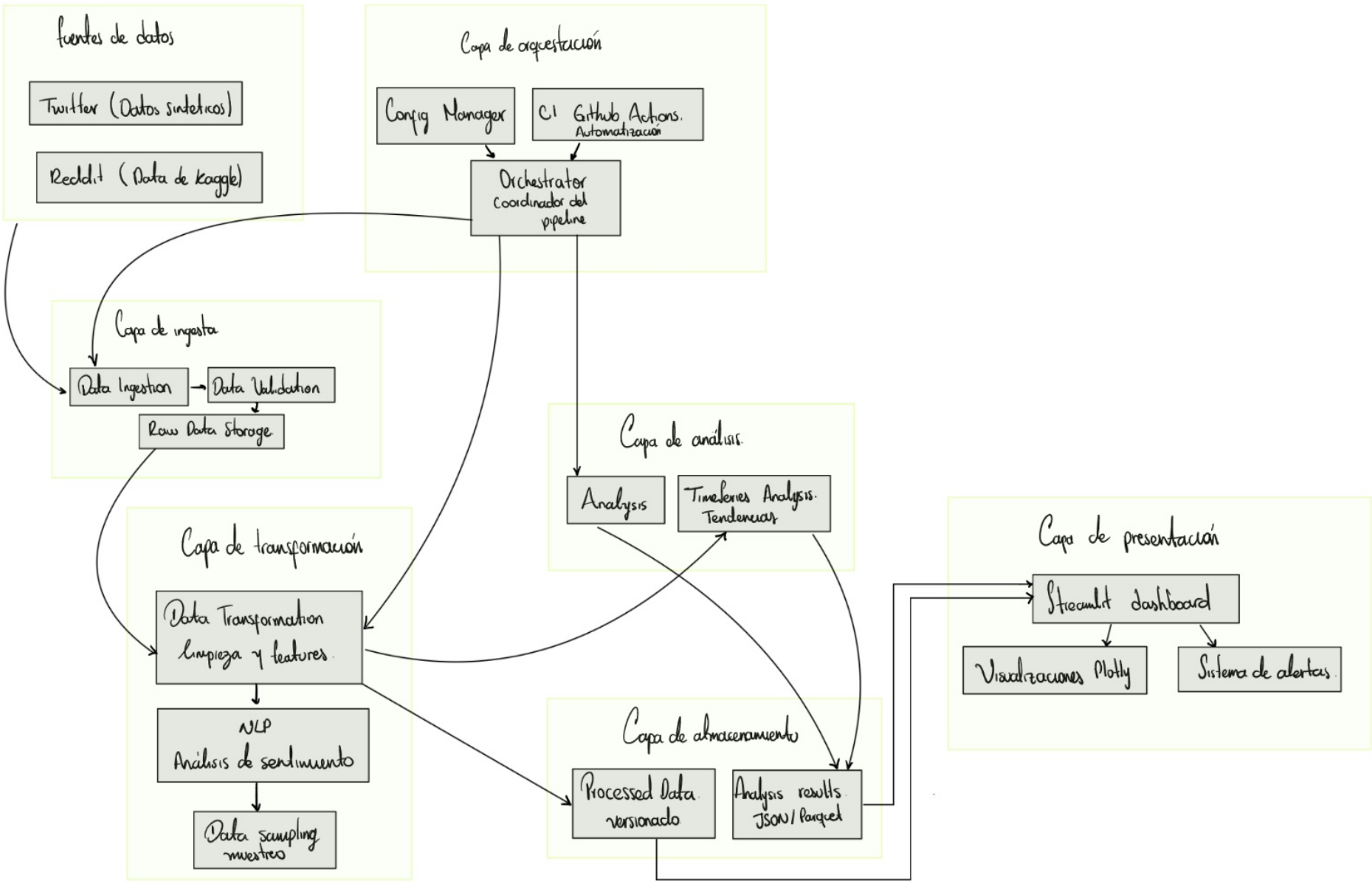
1. **Automatización:** Implementar procesos de ingesta, transformación y control de calidad de datos.
2. **Rigor Analítico:** Aplicar técnicas de análisis estadístico para validar posibles patrones y relaciones en los datos.
3. **Observabilidad:** Desarrollar herramientas interactivas para el monitoreo de tendencias y detección de anomalías.
4. **Gobernanza:** Asegurar la trazabilidad y reproducibilidad mediante versionamiento y CI.

Arquitectura Modular del Pipeline

El sistema orquesta 3 etapas:

1. **Ingesta:**
   - **Inicial:** Conexión automatizada a la API de Kaggle.
   - **Mecanismo de respaldo:** Detección automática de archivos locales en caso de fallo de red.
2. **Validación y Calidad:**
   - **Contratos de Esquema:** Verificación de columnas antes de procesar.
   - **Validaciones Estadísticas:** Detección de columnas con varianza cero.
   - **Control de Nulos:** Monitoreo de umbrales de completitud.
3. **Transformación y Enriquecimiento:**
   - **Estandarización Temporal:** Unificación de formatos dispares.
   - **Imputación:** Manejo estadístico de valores faltantes en métricas de influencia.
   - **NLP:** Scoring de sentimiento (VADER) y modelado de tópicos (LDA).

Arquitectura del Sistema



Motor de Análisis Adaptativo

Análisis Confiable y Automatizado

El pipeline no asume la naturaleza de los datos. Ejecuta un Motor de Decisión en tiempo de ejecución:

- **Paso 1:** Test de Normalidad.
- **Paso 2:** Selección de Algoritmo:
  - Si es Normal $\rightarrow$ Pearson
  - Si No es Normal $\rightarrow$ Spearman.

Resultado: Métricas confiables que se adaptan a la naturaleza de los datos.

Dashboard

Se desarrolló una interfaz para explorar resultados de manera interactiva:

Funcionalidades Implementadas

- **Alertas Automáticas:** Detección de cambios inusuales en el tráfico de datos y caídas importantes en el sentimiento de las publicaciones.
- **Visualización temporal:** Muestra datos históricos en una línea de tiempo.
- **Análisis Estadístico:** Permite ver correlaciones entre variables.
- **Exploración Interactiva:**
  - Filtrado por fecha y plataforma.
  - Visualización de longitud de texto vs sentimiento y temas principales (LDA).

Pipeline de Integración Continua (GitHub Actions)

Para asegurar la reproducibilidad y calidad del pipeline, se implementó un flujo de CI automatizado:

Workflow Automático

1. **Setup:** Instalación de dependencias y uso de caché para acelerar ejecuciones.
2. **Pruebas Unitarias:** Ejecución de la suite `pytest` (33 tests exitosos).
3. **Verificación Rápida:** Test completo del pipeline para asegurar que todos los componentes funcionen correctamente.

Conclusiones

- **Continuidad del Pipeline:** La arquitectura diseñada permite que el flujo de datos siga funcionando incluso si hay problemas con servicios externos, asegurando que el análisis no se interrumpa.
- **Análisis Estadístico Confiable:** Las métricas de interacción en redes muestran distribuciones no normales. Usar pruebas no paramétricas (Spearman) ayudó a obtener resultados más precisos y evitar interpretaciones erróneas.
- **Arquitectura Modular:** La separación en capas (Ingesta, Transformación, Análisis) facilitó la integración de controles de calidad y deja abierta la posibilidad de incorporar nuevas fuentes de datos de forma sencilla.

Gracias

Gracias

Repositorio: `https://github.com/Miguelhv27/technology-posts-pipeline`