DOCUMENTACIÓN DE ARQUITECTURA Y MODELO DE DATOS

Taller referente a la unidad 3.

James Sànchez Toro

Patricia Franco Ruiz

Institución Universitaria Digital de Antioquia

06/04/2025

Andres Felipe Callejas Jaramillo

Infraestructura y arquitectura para Big Data

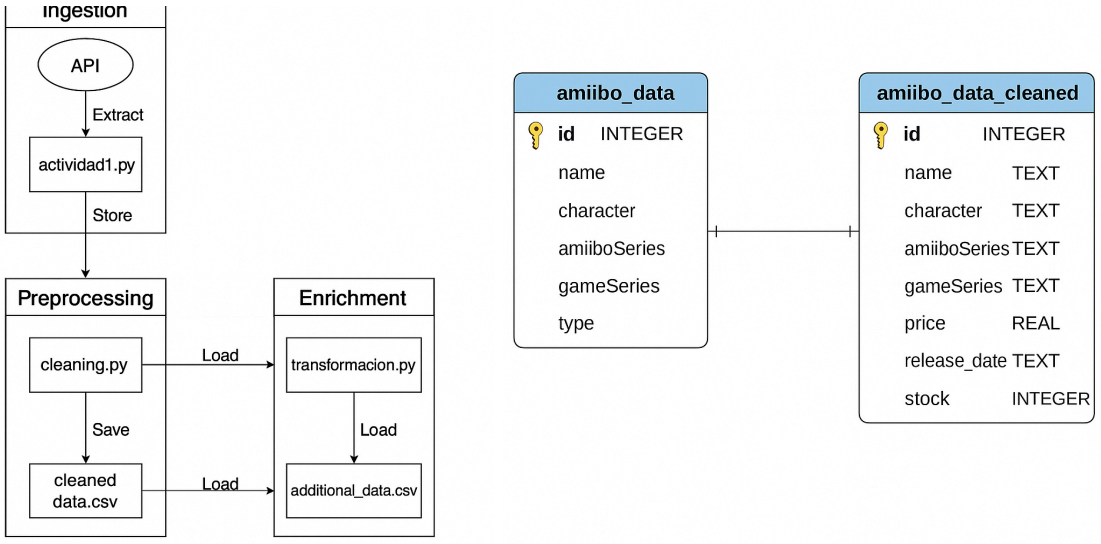
Introducción y Descripción Global de la Arquitectura

El presente proyecto simula un entorno de Big Data en la nube mediante la integración de tres fases fundamentales: Ingesta, Preprocesamiento y Enriquecimiento de datos. Estas fases se ejecutan de manera automatizada mediante GitHub Actions y emplean SQLite como base de datos analítica, junto con scripts desarrollados en Python utilizando librerías como Pandas y NumPy. La solución está orientada al procesamiento de datos provenientes de una API externa de Amiibos, con el fin de demostrar las capacidades de procesamiento, limpieza, enriquecimiento y automatización de datos en un entorno simulado.

Diagramas de Flujo y Arquitectura

Se incluyen diagramas de flujo que representan cada una de las etapas del proyecto:

- **Ingesta de datos**:
    a. Extracción de datos desde la API de Amiibo.
    b. Almacenamiento de los datos en una base SQLite.
    c. Generación de archivos CSV, Excel y un archivo de auditoría.
- **Preprocesamiento**:
    a. Introducción de errores (nulos y duplicados).
    b. Limpieza de los datos eliminando registros nulos y duplicados.
    c. Generación de un nuevo archivo limpio y reporte de limpieza.
- **Enriquecimiento**:
    a. Generación o carga de un dataset adicional (precio, stock, fecha).
    b. Unión del dataset limpio con el adicional.
    c. Exportación del dataset enriquecido y archivo de auditoría.



Modelo de Datos

Definición del Esquema

Se utilizan dos bases de datos SQLite:

- **big_data.db**:
    - Tabla: `amiibo_data`
        - `id` INTEGER (PK)
        - `name` TEXT
        - `character` TEXT
        - `amiiboSeries` TEXT
        - `gameSeries` TEXT
        - `type` TEXT
- **cleaned_data.db**:
    - Tabla: `amiibo_data_cleaned`
        - Igual estructura que la anterior, sin duplicados ni nulos
- **Datos adicionales**:
    - CSV: `additional_data.csv`
        - `id` INTEGER (FK)
        - `price` FLOAT
        - `release_date` TEXT
        - `stock` INTEGER

El modelo resultante permite el enriquecimiento del dataset original con nuevas variables como precio, disponibilidad y fecha de lanzamiento, facilitando el análisis y exploración de los datos.

Diagrama de Datos

El modelo consiste en una relación 1 a 1 entre las tablas `amiibo_data_cleaned` y `additional_data.csv` mediante el campo `id`. Esto permite mantener una estructura relacional clara y optimizada para análisis.

Justificación de Herramientas y Tecnologías

- **SQLite**: Base de datos ligera y sin servidor ideal para entornos simulados. Permite operaciones SQL y almacenamiento persistente.
- **Pandas**: Librería esencial para manipulación y análisis de datos tabulares.
- **NumPy**: Generación de datos aleatorios y cálculos eficientes.

- **GitHub Actions**: Permite la automatización continua del pipeline de datos desde la ingesta hasta la generación de reportes y auditorías.

Simulación del Entorno Cloud

La simulación se realiza mediante la automatización del workflow con GitHub Actions. Cada push al repositorio activa la ejecución de los scripts `actividad1.py`, `cleaning.py` y `transformation.py`, generando archivos intermedios y finales como parte de la arquitectura de procesamiento.

Flujo de Datos y Automatización

1. **Actividad 1 (Ingesta)**:
    - Obtiene datos desde la API.
    - Almacena en SQLite y genera auditoría.
2. **Actividad 2 (Preprocesamiento)**:
    - Ensucia los datos con valores nulos y duplicados.
    - Limpia los datos y los guarda en una nueva base.
    - Genera reporte de limpieza.
3. **Actividad 3 (Enriquecimiento)**:
    - Crea datos adicionales simulados.
    - Une los datos limpios con los adicionales.
    - Genera dataset enriquecido y auditoría.

Conclusiones y Recomendaciones

Beneficios:

- Estructura clara y modular.
- Pipeline reproducible y automatizado.

Limitaciones:

- SQLite no es escalable para entornos reales.
- La automatización está limitada a GitHub; en la nube real se debería usar Airflow o Databricks Workflows.

Recomendaciones:

- Migrar a una base de datos distribuida (como BigQuery).
- Escalar la automatización a servicios cloud nativos como AWS Lambda o GCP Cloud Functions.