ACTIVIDAD 4

LUIS CARLOS PACHÓN MENDOZA

CRISTHIAN MATEO PACHÓN MENDOZA

Infraestructura y arquitectura para Big Data

PROFESOR

Andres Felipe Callejas

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA

INGENIERÍA DE SOFTWARE Y DATOS

2025

Documentación del Proyecto de Big Data

1. Introducción y Descripción Global de la Arquitectura

Este documento describe la arquitectura del proyecto de procesamiento de datos, el cual se desarrolla en un entorno simulado de Big Data en la nube. El proyecto consta de tres fases principales:

1. **Ingesta de Datos:** Obtención y almacenamiento de datos desde la API RandomUser https://randomuser.me/ en una base de datos SQLite.
2. **Preprocesamiento y Limpieza:** Eliminación de duplicados, tratamiento de valores nulos y transformaciones necesarias.
3. **Enriquecimiento de Datos:** Integración con fuentes de datos adicionales en diferentes formatos para mejorar la calidad de la información.

Todas estas fases se encuentran automatizadas mediante GitHub Actions para asegurar la ejecución continua y la integridad de los datos.

---

2. Componentes Principales

2.1 Base de Datos Analítica

Se utiliza SQLite para almacenar la información obtenida en la fase de ingesta y los datos procesados. La base de datos contiene la siguiente tabla principal:

Tabla usuarios

- **id** (INTEGER, PRIMARY KEY)
- **nombre** (TEXT)
- **apellido** (TEXT)
- **email** (TEXT)
- **genero** (TEXT)
- **pais** (TEXT)
- **fecha_extraccion** (TEXT)

2.2 Scripts de Procesamiento

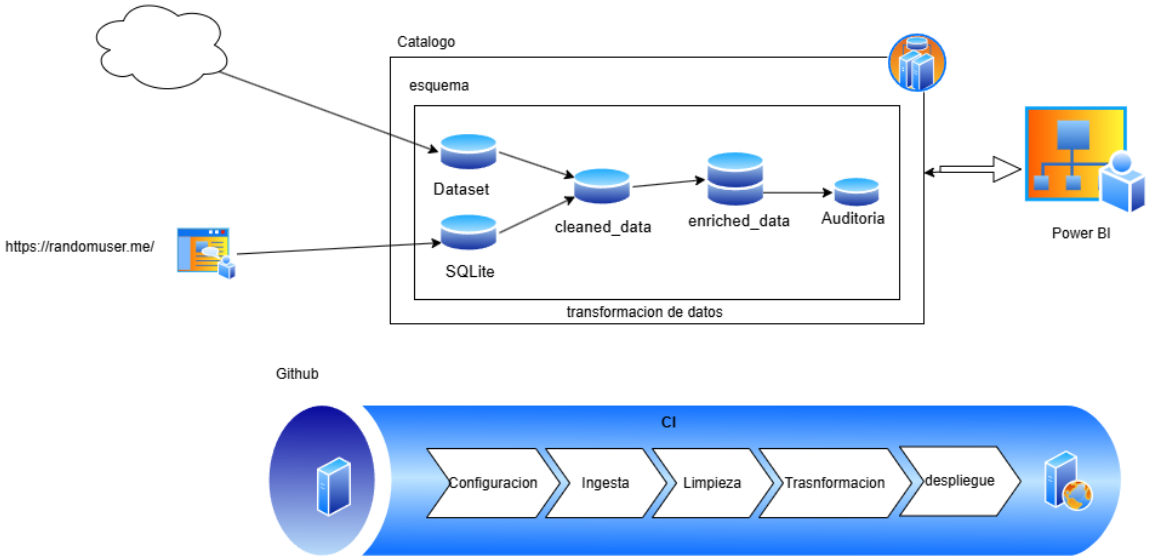
- `ingesta.py`: Obtiene datos desde la API y los almacena en la base de datos.
- `cleaning.py`: Realiza la limpieza y normalización de los datos almacenados.
- `transform.py`: Fusiona los datasets `dataset.csv` y `cleaned_data.csv`, eliminando redundancias y generando `enriched_data.csv`.

2.3 Mecanismo de Automatización

El flujo de procesamiento está automatizado mediante GitHub Actions, como se describe en el archivo test.yml, permitiendo la ejecución programada y el versionamiento automático.

3. Diagramas de Arquitectura

3.1 Diagrama de Flujo de Datos



3.2 Modelo de Datos

USUARIOS		
int	id	PK
string	nombre	
string	apellido	
string	email	
string	genero	
string	pais	
date	fecha_extraccion	

Relacion

ENRICHED_DATA		
int	id	PK
string	nombre	
string	apellido	
string	email	
string	genero	
string	pais	
date	fecha_enriquecimiento	

---

4. Justificación de Herramientas y Tecnologías

4.1 Elección de Herramientas

- **SQLite:** Permite almacenar y gestionar grandes volúmenes de datos de forma eficiente sin necesidad de un servidor de base de datos.
- **Pandas:** Facilita la manipulación y análisis de datos estructurados de manera eficiente.
- **GitHub Actions:** Automatiza el procesamiento de datos, asegurando ejecuciones consistentes y repetibles.

4.2 Simulación del Entorno Cloud

La base de datos SQLite y los scripts de procesamiento simulan un entorno en la nube, asegurando que el flujo de datos sea el adecuado para una futura migración a un sistema de Big Data real.

---

5. Flujo de Datos y Automatización

El flujo de datos sigue el siguiente proceso:

1. Se extraen datos desde la API RandomUser y se almacenan en SQLite.
2. Se ejecuta un proceso de limpieza y normalización.
3. Se integran fuentes de datos adicionales y se enriquece la información.
4. Se exportan los datos finales en formato CSV y se generan reportes de auditoría.
5. Todo este flujo se ejecuta automáticamente mediante GitHub Actions.

---

6. Conclusiones y Recomendaciones

6.1 Beneficios

- Automatización del procesamiento de datos.
- Integración de diversas fuentes para mejorar la calidad de los datos.
- Implementación escalable y flexible para futuras ampliaciones.

6.2 Limitaciones

- SQLite tiene limitaciones para entornos de Big Data a gran escala.
- El procesamiento puede optimizarse usando herramientas como PySpark o bases de datos distribuidas.

6.3 Recomendaciones

- Implementar almacenamiento en la nube con bases de datos distribuidas como BigQuery o Amazon Redshift.
- Escalar el procesamiento utilizando PySpark para manejar volúmenes mayores de datos.
- Mejorar la integración de nuevas fuentes de datos con APIs adicionales y fuentes externas.