

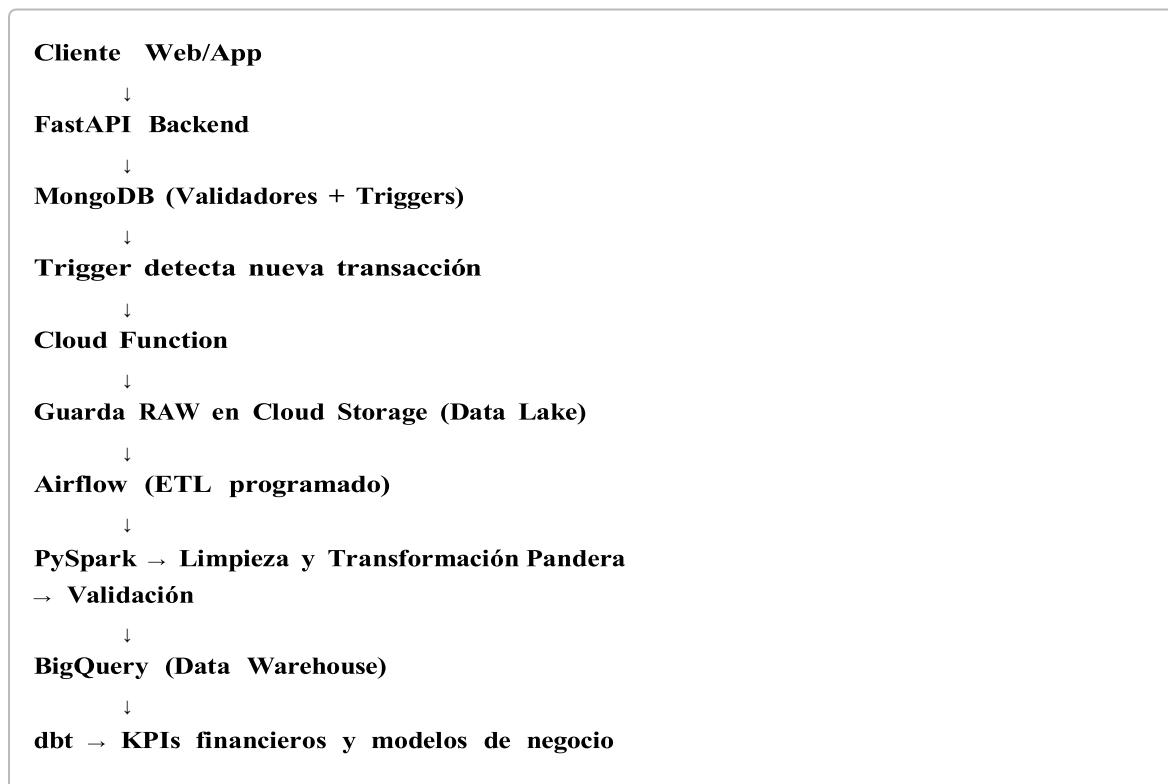
# Proyecto: Plataforma de Datos para Casa de Cambios (Arquitectura con MongoDB + Triggers + GCP)

## 1. Objetivo

Construir una plataforma integral para una casa de cambios digital utilizando:

- **MongoDB (transaccional + triggers + validadores)**
- **FastAPI** para la API
- **Cloud Functions** para orquestación
- **Cloud Storage como Data Lake**
- **Airflow + PySpark + Pandera** para ETL batch
- **BigQuery** como Data Warehouse
- **dbt-core** para modelos analíticos
- **CI/CD (GitHub Actions o Jenkins)**
- **Buenas prácticas ISO 27001**

## 2. Arquitectura General



## 3. Flujo de Datos

### 3.1 Ingesta

- Usuario realiza transacción desde la web.
- FastAPI recibe los datos.
- Se guardan en **MongoDB**.

- MongoDB aplica **validadores** (schemas, límites, tipos, monedas válidas).

### 3.2 Automatización

- Un **trigger** de MongoDB detecta el *insert*.
- Envía el documento completo a una **Cloud Function**.

### 3.3 Data Lake

- La Cloud Function escribe el archivo RAW en **Cloud Storage**.
- Estructura sugerida:

`raw/transactions/YYYY/MM/DD/tx_123.json`

### 3.4 ETL (Airflow + PySpark + Pandera)

- Airflow programa un DAG diario u horario.
- PySpark limpia, transforma y normaliza la data.
- Pandera valida tipos y reglas de negocio.

### 3.5 Warehouse

- Los datos limpios se cargan en **BigQuery**.
- dbt crea modelos:
  - `fact_transactions`
  - `dim_currency`
  - `kpi_exchange_rate`
  - `volume_by_hour`
  - `fraud_signals`

## 4. Tecnologías Incluidas

- **MongoDB**: triggers, validators, funciones.
- **Python**: Pandas, PySpark, PyMongo, Pandera.
- **FastAPI**: API transaccional.
- **GCP**: Cloud Storage, Cloud Functions, BigQuery, Cloud Run.
- **CI/CD**: GitHub Actions o Jenkins.
- **Airflow, dbt-core, PySpark**: orquestación + transformación.
- **Gobernanza**: prácticas alineadas a ISO 27001.

## 5. Caso de Uso Principal

Registrar transacciones de compra/venta de divisas en tiempo real, validarlas, almacenarlas de manera confiable y generar analíticas financieras para la operación.

## 6. Próximos Pasos Potenciales

1. Crear el **diagrama visual**.
  2. Crear la **estructura del repositorio**.
  3. Generar **código base del FastAPI**.
  4. Generar **trigger + función Cloud Function**.
  5. Crear **primer DAG de Airflow**.
  6. Definir **modelos en dbt**.
  7. Integrar **CI/CD**.
-