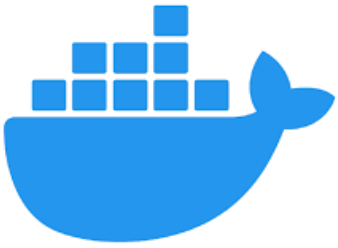
Deusto

Proyecto final

Fases del ciclo de vida del dato

Desarrollo de aplicaciones para internet de las cosas



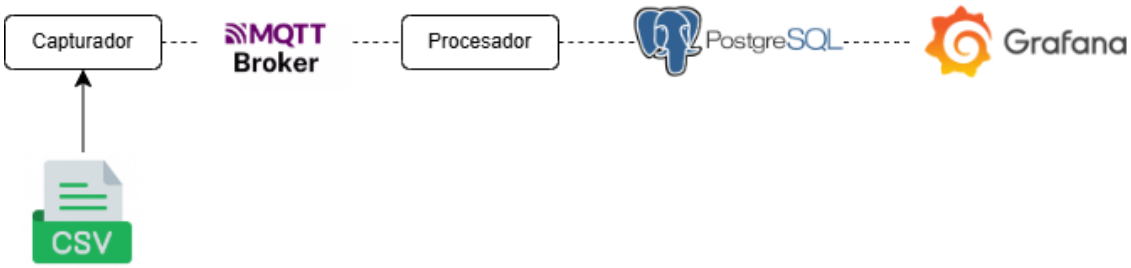
Eneko Sáez,
Manel Diaz y
Rubén Alsasua

Índice

Introducción

En este proyecto tiene como objetivo capturar datos desde un archivo CSV, publicarlos mediante MQTT y procesarlos para almacenarlos en una base de datos PostgreSQL. Por último, los datos obtenidos son analizados y posteriormente visualizados utilizando Grafana. Todo el sistema se integra en contenedores Docker y docker-compose.

Arquitectura del Proyecto



- Capturador: Publica datos al broker MQTT
- Procesador: Valida los datos y los guarda en PostgreSQL
- Broker MQTT: Middleware de mensajería
- PostgreSQL: Almacena los datos de calidad de aire
- Grafana Visualiza y analiza los datos históricos de calidad de aire

App “Capturador”

Fase de Captura

La fase de captura del Capturador se centra en leer datos históricos de la calidad del aire desde un archivo CSV, simulando la obtención de datos de un sensor. Para poder realizar el trabajo de lectura del archivo global_air_pollution_data.csv, se hace uso de la librería pandas de Python.

Fase de Envío (1)

La fase de envío del Capturador se encarga de convertir cada fila del CSV en un objeto JSON y publicarlo mediante MQTT. Para conseguirlo se ha utilizado la librería paho-mqtt para publicar mensajes en el tópico calidad_aire/ciudades.

El destino sería el Broker MQTT de mosquitto que actúa como intermediario.

App “Procesador”

Fase de Envío (2)

El procesador hace de suscriptor en la arquitectura publicador suscriptor, es decir se encarga de suscribirse al tópico indicado para recibir los datos esperados. De esta manera, recibe los datos enviados por el Capturador a través del Broker. Para poder recibir los mensajes hace uso de la librería paho-mqtt.

Fase de Procesamiento

En la parte de procesamiento de los datos, se encarga de validar los tipos de datos, campos requeridos y formatos de los datos entrantes. Para poder cumplir con su responsabilidad hace uso de la librería pydantic. En caso de existir algún dato erróneo se descarta el dato y se muestra un mensaje de error.

Fase de Persistencia

En la parte de Persistencia, se encarga de insertar los datos validados por pydantic en la base de datos PostgreSQL. Para poder realizar el guardado de datos se realiza una conexión a PostgreSQL vía librería de Python psycopg2. Además, hace uso de la imagen postgres:15.

Además, existe un archivo init.sql que se encarga de crear la tabla en la que se guardan los datos en caso de que no existiera.

Grafana

Fase de Visualización

En cuanto a la fase de visualización, se ha creado un dashboard visual en el que se muestran a tiempo real los datos almacenados en la base de datos. Para conseguir la visualización, se ha utilizado Grafana y PostgreSQL como fuente de datos.

Se han creado diferentes visualizaciones relevantes entre las que destaca un mapa geográfico.

Análisis y Limpieza

Fase de Analítica

Por último, para la fase de analítica, se ha creado un Notebook para visualizar diferentes datos de las columnas del csv, como, por ejemplo, cuántos valores nulos tiene, o duplicados, para finalizar visualizando varios gráficos utilizando la librería “matplotlib”.

Ejecución Paso a Paso

Clonar el Repositorio

Mediante el siguiente comando es posible clonar el proyecto: "*git clone https://github.com/EnekoDeusto/Proyecto_final.git*"

Luego es necesario abrir la carpeta en la que se encuentra el archivo docker-compose.yml.

Levantar contenedores

Mediante el siguiente comando es posible levantar todos los contenedores para el correcto funcionamiento del proyecto: *"docker-compose up –build-d"*

Esto iniciará:

- Broker MQTT
- Capturador de datos
- Procesador/ validador/ almacenamiento
- Base de datos PostgreSQL
- Panel Grafana

Verificar el funcionamiento

- Para comprobar que los contenedores están corriendo es posible utilizar el siguiente comando: *"docker ps"*
- Para comprobar el correcto funcionamiento de los contenedores de Capturador y precesador, es posible visualizar los logs de validación y el almacenamiento en las tablas mediante los siguientes comandos: *"docker exec -it Postgres psql -U postgres -d calidad_aire"* y *"SELECT * FROM calidad_aire LIMIT 5;"*
- Para comprobar de forma precisa la visualizacion se puede acceder a la siguiente URL mediante el navegador: http://localhost:3000. Iniciando sesión con usuario *"admin"* y contraseña *"admin"*, es posible ver todos los gráficos.