



Persistencia de información de un parque eólico con InfluxDB

Grupo 2 - Iñigo Murga, Jon Cañadas y Mikel García



Reto abordado

Este proyecto consiste en la persistencia de datos del SCADA de una turbina de viento. Para ello utilizamos InfluxDB.

- Se introducen los datos del dataset del SCADA en InfluxDB.
- Se actualizan las marcas de tiempo de los datos.
- Se calculan diferentes promedios, maximos, etc..



¿Por qué InfluxDB?

InfluxDB está especialmente diseñada para manejar series temporales, como las que genera el SCADA de una turbina de viento.

Optimización para datos con marca temporal: InfluxDB maneja eficientemente grandes volúmenes de datos indexados por tiempo, ideal para lecturas frecuentes de sensores.

Lenguaje de consultas orientado a tiempo (Flux): Facilita agregaciones, downsampling, detección de anomalías y análisis temporal.



`docker-compose.yml`

```
1  ✓ services:
2  ✓   influxdb:
3      image: influxdb:2.7.11
4      container_name: influxdb
5      restart: unless-stopped
6  ✓   ports:
7      - "8086:8086"
8  ✓   volumes:
9      - influxdb_data:/var/lib/influxdb
10
11  ✓ volumes:
12      influxdb_data:
```

Función insertar_datos()

```

# Leer y enviar datos desde un archivo CSV
def insertar_datos(dataset_path, write_api):
    datos = []
    with open(dataset_path, mode='r+', encoding='utf-8-sig') as file:
        reader = csv.DictReader(file)
        for i, row in enumerate(reader):
            try:
                # Parsear fecha y sumar 7 años
                fecha_original = datetime.strptime(row["Date/Time"], "%d %m %Y %H:%M")
                fecha_modificada = fecha_original.replace(year=fecha_original.year + 7).replace(tzinfo=datetime.timezone.utc)

                dato = Point("wind_data") \
                    .time(fecha_modificada) \
                    .field("LV ActivePower (kW)", float(row["LV ActivePower (kW)"])) \
                    .field("Wind Speed (m/s)", float(row["Wind Speed (m/s)"])) \
                    .field("Theoretical Power Curve (KWh)", float(row["Theoretical Power Curve (KWh)"])) \
                    .field("Wind Direction", float(row["Wind Direction"]))

                datos.append(dato)

            except Exception as e:
                print(f"Error al escribir datos: {e}")
                logging.error(f"Error al escribir datos: {e}")

    # Escribir todos los puntos de una vez
    if datos:
        write_api.write(bucket=bucket, org=org, record=datos)
        print(f"{len(datos)} filas de datos insertadas correctamente.")
        logging.info(f"{len(datos)} filas de datos insertadas correctamente.")

```

Función ejecutar_consultas()

```

def ejecutar_consultas(client):
    query_api = client.query_api()

    # Promedio por hora - Potencia activa
    query1 = f'''
    from(bucket: "{bucket}")
    |> range(start: -30d)
    |> filter(fn: (r) => r._measurement == "wind_data" and r._field == "LV ActivePower (kW)")
    |> aggregateWindow(every: 1h, fn: mean)
    |> yield(name: "hourly_mean_power")
    ...

    # Promedio diario - Velocidad del viento
    query2 = f'''
    from(bucket: "{bucket}")
    |> range(start: -30d)
    |> filter(fn: (r) => r._measurement == "wind_data" and r._field == "Wind Speed (m/s)")
    |> aggregateWindow(every: 1d, fn: mean)
    |> yield(name: "daily_mean_wind")
    ...

    # Máximo semanal - Potencia teórica
    query3 = f'''
    from(bucket: "{bucket}")
    |> range(start: -90d)
    |> filter(fn: (r) => r._measurement == "wind_data" and r._field == "Theoretical_Power_Curve (KWh)")
    |> aggregateWindow(every: 1w, fn: max)
    |> yield(name: "weekly_max_theoretical_power")
    ...

    for i, query in enumerate([query1, query2, query3], start=1):
        print(f"\n--- Resultados consulta {i} ---")
        result = query_api.query(org=org, query=query)
        for table in result:
            for record in table.records:
                print(f"{record.get_time()} | {record.get_field()} | {record.get_value()}")

```



Problemas y Posibles futuras mejoras

- Al insertar los datos tras modificar el campo de fecha, se presentaron múltiples errores relacionados con el formato de fecha, que finalmente pudieron ser corregidos.
- Al ejecutar el script de volcado desde WSL, no se establecía conexión con InfluxDB. Fue necesario ejecutarlo directamente desde el sistema operativo Windows para que funcionara correctamente.
- ✓ Para evitar tener que modificar manualmente los parámetros del script en cada ejecución, se recomienda automatizar este proceso. Una opción sería exportar el token como variable de entorno y acceder a él mediante el script utilizando variables.



Alternativas

