
LAB7
Presented By: Gerardo Pineda
https://github.com/Gerax5/Lab7_redes

SIMULACION

```
PS C:\Users\Gerax\OneDrive\Desktop\UVGG\4-2\Redes\Lab7_redes> & C:
sers/Gerax/OneDrive/Desktop/UVGG/4-2/Redes/Lab7_redes/producer.py
{"temperatura": 99.94, "humedad": 58, "direccion_viento": "E"}
{"temperatura": 51.61, "humedad": 34, "direccion_viento": "SO"}
{"temperatura": 73.27, "humedad": 19, "direccion_viento": "E"}
{"temperatura": 60.68, "humedad": 34, "direccion_viento": "NO"}
{"temperatura": 63.66, "humedad": 20, "direccion_viento": "S"}
{"temperatura": 29.75, "humedad": 46, "direccion_viento": "N"}
{"temperatura": 51.9, "humedad": 48, "direccion_viento": "E"}
{"temperatura": 67.86, "humedad": 17, "direccion_viento": "NE"}
{"temperatura": 79.31, "humedad": 15, "direccion_viento": "E"}
{"temperatura": 45.86, "humedad": 49, "direccion_viento": "NE"}
```

Se creo un sensor una simulación de un sensor en python que produjera la temperatura, humedad y dirección de viento

PREGUNTAS

¿A qué capa pertenece JSON/SOAP según el Modelo OSI y porque?

- JSON/SOAP esta en la capa de presentación 6 porque define formato de datos y codificación para intercambio.

¿Qué beneficios tiene utilizar un formato como JSON/SOAP?

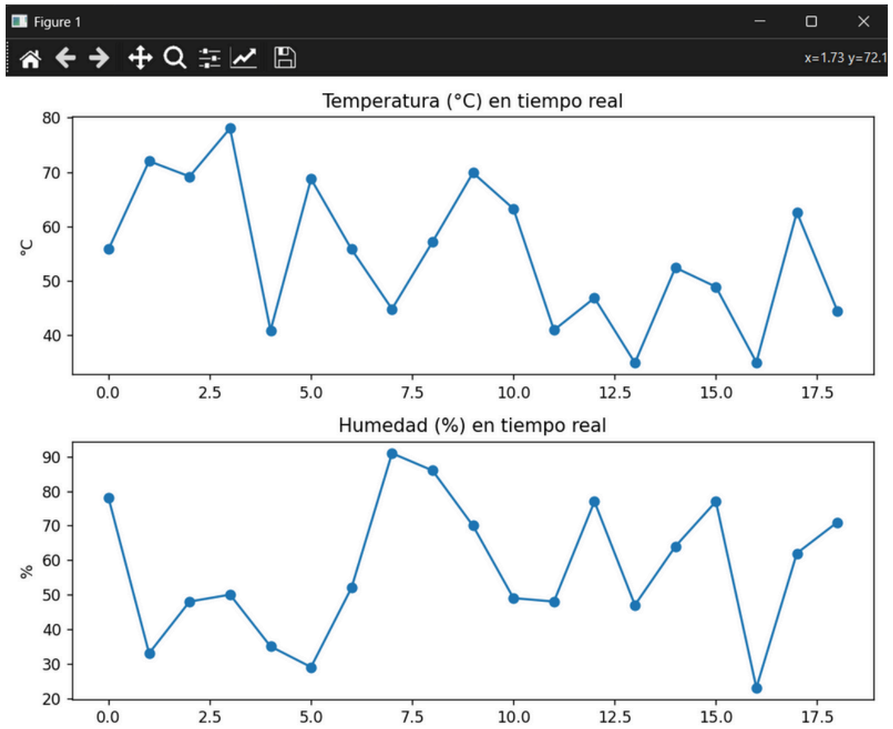
- Estos ofrecen una forma de estandarización, la cual es legible, independiente del lenguaje y plataforma de intercambio. Facilitan la interoperabilidad y depuración. Además, soportan validación y seguridad.

ENVIO DE DATOS

```
sers/Gerax/OneDrive/Desktop/UVGG/4-2/Redes/Lab7_redes/producer.py
Enviando: {"temperatura": 73.59, "humedad": 46, "direccion_viento": "S"}
Enviando: {"temperatura": 80.21, "humedad": 98, "direccion_viento": "S"}
Enviando: {"temperatura": 45.68, "humedad": 49, "direccion_viento": "S"}
Enviando: {"temperatura": 55.13, "humedad": 84, "direccion_viento": "N"}
Enviando: {"temperatura": 46.72, "humedad": 71, "direccion_viento": "O"}
Enviando: {"temperatura": 51.45, "humedad": 87, "direccion_viento": "E"}
Enviando: {"temperatura": 40.09, "humedad": 36, "direccion_viento": "SE"}
Enviando: {"temperatura": 59.24, "humedad": 71, "direccion_viento": "NO"}
Enviando: {"temperatura": 37.31, "humedad": 72, "direccion_viento": "NO"}
Enviando: {"temperatura": 44.49, "humedad": 26, "direccion_viento": "E"}
Enviando: {"temperatura": 80.93, "humedad": 69, "direccion_viento": "NE"}
Enviando: {"temperatura": 61.79, "humedad": 32, "direccion_viento": "E"}
Enviando: {"temperatura": 56.37, "humedad": 91, "direccion_viento": "NO"}
Enviando: {"temperatura": 52.05, "humedad": 19, "direccion_viento": "NO"}
Enviando: {"temperatura": 79.46, "humedad": 100, "direccion_viento": "O"}
Enviando: {"temperatura": 37.88, "humedad": 42, "direccion_viento": "N"}
Enviando: {"temperatura": 48.57, "humedad": 42, "direccion_viento": "N"}
Enviando: {"temperatura": 41.49, "humedad": 36, "direccion_viento": "NO"}
Enviando: {"temperatura": 25.77, "humedad": 31, "direccion_viento": "S"}
Enviando: {"temperatura": 105.79, "humedad": 18, "direccion_viento": "N"}
Enviando: {"temperatura": 60.79, "humedad": 57, "direccion_viento": "E"}
Enviando: {"temperatura": 63.71, "humedad": 51, "direccion_viento": "O"}
```

A la simulación se le añadio kafka producer para poder enviar los datos a el servidor con un lapso de 15 a 30 segundos para no sobrecargar el sistema y simular un ambiente real

CONSUMO DE DATOS



Con el consumidor se crearon graficas en tiempo real donde se ve la humidad y la temperatura enviada.

PREGUNTAS

¿Qué ventajas y desventajas considera que tiene este acercamiento basado en Pub/Sub de Kafka?

- El enfoque Pub/Sub con Kafka permite un sistema altamente escalable, tolerante a fallos y adecuado para procesamiento en tiempo real, ideal para IoT y telemetría. Sin embargo, implica complejidad, mayor consumo de recursos, una curva de aprendizaje considerable y puede ser excesivo para aplicaciones pequeñas o con conectividad limitada.

¿Para qué aplicaciones tiene sentido usar Kafka? ¿Para cuáles no?

- Kafka es ideal para sistemas de gran escala, IoT masivo, procesamiento en tiempo real, microservicios y pipelines de datos. No es adecuado para aplicaciones pequeñas, conexiones inestables, baja frecuencia de mensajes, baja latencia extrema o sistemas simples donde REST serían suficientes.

ENVIO DE DATOS

```
sers/Gerax/OneDrive/Desktop/UVGG/4-2/Redes/Lab7_redes/producer.py
Enviando: {"temperatura": 73.59, "humedad": 46, "direccion_viento": "S"}
Enviando: {"temperatura": 80.21, "humedad": 98, "direccion_viento": "S"}
Enviando: {"temperatura": 45.68, "humedad": 49, "direccion_viento": "S"}
Enviando: {"temperatura": 55.13, "humedad": 84, "direccion_viento": "N"}
Enviando: {"temperatura": 46.72, "humedad": 71, "direccion_viento": "O"}
Enviando: {"temperatura": 51.45, "humedad": 87, "direccion_viento": "E"}
Enviando: {"temperatura": 40.09, "humedad": 36, "direccion_viento": "SE"}
Enviando: {"temperatura": 59.24, "humedad": 71, "direccion_viento": "NO"}
Enviando: {"temperatura": 37.31, "humedad": 72, "direccion_viento": "NO"}
Enviando: {"temperatura": 44.49, "humedad": 26, "direccion_viento": "E"}
Enviando: {"temperatura": 80.93, "humedad": 69, "direccion_viento": "NE"}
Enviando: {"temperatura": 61.79, "humedad": 32, "direccion_viento": "E"}
Enviando: {"temperatura": 56.37, "humedad": 91, "direccion_viento": "NO"}
Enviando: {"temperatura": 52.05, "humedad": 19, "direccion_viento": "NO"}
Enviando: {"temperatura": 79.46, "humedad": 100, "direccion_viento": "O"}
Enviando: {"temperatura": 37.88, "humedad": 42, "direccion_viento": "N"}
Enviando: {"temperatura": 48.57, "humedad": 42, "direccion_viento": "N"}
Enviando: {"temperatura": 41.49, "humedad": 36, "direccion_viento": "NO"}
Enviando: {"temperatura": 25.77, "humedad": 31, "direccion_viento": "S"}
Enviando: {"temperatura": 105.79, "humedad": 18, "direccion_viento": "N"}
Enviando: {"temperatura": 60.79, "humedad": 57, "direccion_viento": "E"}
Enviando: {"temperatura": 63.71, "humedad": 51, "direccion_viento": "O"}
```

A la simulación se le añadio kafka producer para poder enviar los datos a el servidor con un lapso de 15 a 30 segundos para no sobrecargar el sistema y simular un ambiente real

ENCODERS

```python
def encode_payload(temp, hum, wind):
    # Escalar temperatura
    temp_scaled = int(temp * 100)  # (0–11000)

    # Mapear viento
    wind_map = ["N", "NO", "O", "SO", "S", "SE", "E", "NE"]
    wind_index = wind_map.index(wind)  # 0–7 (3 bits)

    # Empaquetar en un entero de 24 bits
    packed = (temp_scaled << 10) | (hum << 3) | wind_inde

    # Convertir a 3 bytes (big-endian)
    return packed.to_bytes(3, byteorder="big")
```

```python
def decode_payload(payload_bytes):
    wind_map = ["N", "NO", "O", "SO", "S", "SE", "E", "NE"]

    packed = int.from_bytes(payload_bytes, byteorder="big")

    # Extraer campos
    wind = packed & 0b111
    hum = (packed >> 3) & 0b1111111
    temp_scaled = (packed >> 10) & 0b11111111111111

    temp = temp_scaled / 100.0

    return {
        "temperatura": temp,
        "humedad": hum,
        "direccion_viento": wind_map[wind]
    }
```

La humedad se escala, la direccion del viento se pasa al index y por ultimo se multiplica la temperatura por 100 para que ocupe el resto

PREGUNTAS

¿Qué complejidades introduce el tener un payload restringido (pequeño)?

- Como ya no se puede enviar un json, ni un string, ahora se tiene que empezar a pensar en bits y compresión. Haciendo que aumente la complejidad del código y los riesgos de errores. Por mal planteamiento debido a que no hay depuración.

¿Cómo podemos hacer que el valor de temperatura quepa en 14 bits?

- 14 bits representan valores de 0 – 16383 y como no se puede enviar un float se multiplica por 100 los valores por ejemplo 110.00 → 11000 el cual cabe y en el receptor solo se divide el valor por la misma cantidad.

¿Qué sucedería si ahora la humedad también es tipo float con un decimal? ¿Qué decisiones tendríamos que tomar en ese caso?

- los 7 bits ya no bastarían, se podría hacer la misma técnica que con el otro pero esto requiere 10 bits. Se debería de tomar una decisión de perder precisión o sacrificar algo en otro lado para que entre en el protocolo. Se tiene que decidir que tan precisos queremos los datos.

PREGUNTAS

¿Qué parámetros o herramientas de Kafka podrían ayudarnos si las restricciones fueran aún más fuertes?

- kafka tiene varias herramientas de compresion en el productor mediante algoritmos. Tambien, se pueden ajustar el batching, que permite ajustar multiples mensajes pequeños en un solo bloque. Aunque niguna herramientas elimina la restriccion del payload a nivel de sensor, si opera de manera mas eficiente en entornos de recusos limitados.

REPO

https://github.com/Gerax5/Lab7_redes


THANK YOU
For Your Attention