



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Laboratorio 7

Redes

Mauricio Julio Rodrigo Lemus Guzman – 22461

Evidencias

Producir normal en JSON

```
===== LABORATORIO 7 - IoT Estación =====
1) Producer normal (JSON)
2) Consumer normal (JSON)
3) Producer compacto (3 bytes)
4) Consumer compacto (3 bytes)
5) Salir

Elegí opción: 1

== Producer NORMAL (JSON) corriendo (7 mensajes) ==
[1/7] Enviado JSON: {'temperatura': 43.3, 'humedad': 81, 'direccion_viento': 'O'}
[2/7] Enviado JSON: {'temperatura': 94.83, 'humedad': 46, 'direccion_viento': 'NE'}
[3/7] Enviado JSON: {'temperatura': 31.2, 'humedad': 76, 'direccion_viento': 'S'}
[4/7] Enviado JSON: {'temperatura': 54.5, 'humedad': 35, 'direccion_viento': 'SE'}
[5/7] Enviado JSON: {'temperatura': 59.99, 'humedad': 39, 'direccion_viento': 'O'}
[6/7] Enviado JSON: {'temperatura': 0, 'humedad': 75, 'direccion_viento': 'NE'}
[7/7] Enviado JSON: {'temperatura': 73.71, 'humedad': 45, 'direccion_viento': 'S'}
✓ Producer NORMAL finalizado.
```

Consumer normal en JSON

```
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab 7> & C:/Users/rodle/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Escritorio/uvg/redes/lab 7/lab7.py"

===== LABORATORIO 7 - IoT Estación =====
1) Producer normal (JSON)
2) Consumer normal (JSON)
3) Producer compacto (3 bytes)
4) Consumer compacto (3 bytes)
5) Salir

Elegí opción: 2

== Consumer NORMAL (JSON) escuchando (7 mensajes) ==
[1/7] Recibido JSON: {'temperatura': 31.2, 'humedad': 76, 'direccion_viento': 'S'}
[2/7] Recibido JSON: {'temperatura': 54.5, 'humedad': 35, 'direccion_viento': 'SE'}
[3/7] Recibido JSON: {'temperatura': 59.99, 'humedad': 39, 'direccion_viento': 'O'}
[4/7] Recibido JSON: {'temperatura': 0, 'humedad': 75, 'direccion_viento': 'NE'}
[5/7] Recibido JSON: {'temperatura': 43.3, 'humedad': 81, 'direccion_viento': 'O'}
[6/7] Recibido JSON: {'temperatura': 94.83, 'humedad': 46, 'direccion_viento': 'NE'}
[7/7] Recibido JSON: {'temperatura': 73.71, 'humedad': 45, 'direccion_viento': 'S'}
```

Producer compacto 3 bytes

```
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab 7> & C:/Users/rodle/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/rodle/OneD
rive/Escritorio/uvg/redes/lab 7/lab7.py"

===== LABORATORIO 7 - IoT Estación =====
1) Producer normal (JSON)
2) Consumer normal (JSON)
3) Producer compacto (3 bytes)
4) Consumer compacto (3 bytes)
5) Salir

Elegí opción: 3

== Producer COMPACTO (3 bytes) corriendo (7 mensajes) ==
[1/7] Enviado compacto (3 bytes): b'\x00\xfa\xaa'
[2/7] Enviado compacto (3 bytes): b'\x00\x4u'
[3/7] Enviado compacto (3 bytes): b'\x01<z'
[4/7] Enviado compacto (3 bytes): b'\x01\x0e\xce'
[5/7] Enviado compacto (3 bytes): b'\x00\xaa\xbd'
[6/7] Enviado compacto (3 bytes): b'\x00\xfdY'
[7/7] Enviado compacto (3 bytes): b'\x00\xc8\xd9'
✓ Producer COMPACTO finalizado.
```

Consumer compacto 3 bytes

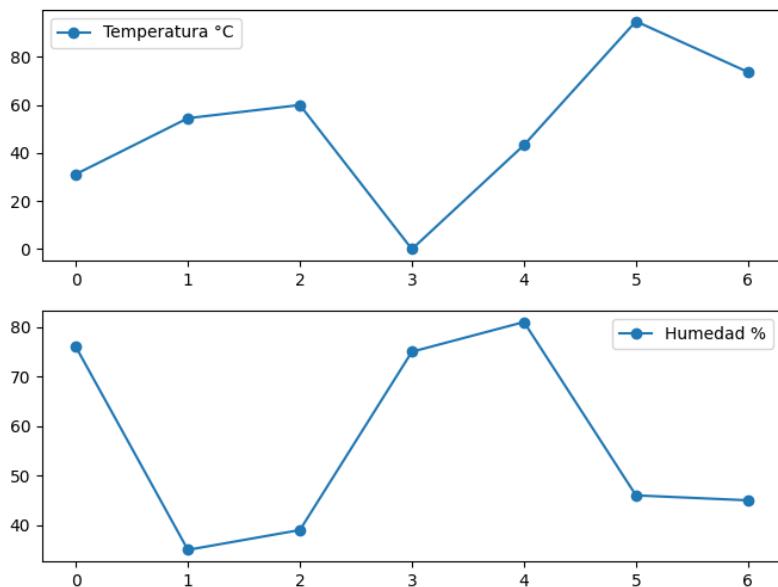
```
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab 7> & C:/Users/rodle/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/rodle/OneD
rive/Escritorio/uvg/redes/lab 7/lab7.py"

===== LABORATORIO 7 - IoT Estación =====
1) Producer normal (JSON)
2) Consumer normal (JSON)
3) Producer compacto (3 bytes)
4) Consumer compacto (3 bytes)
5) Salir

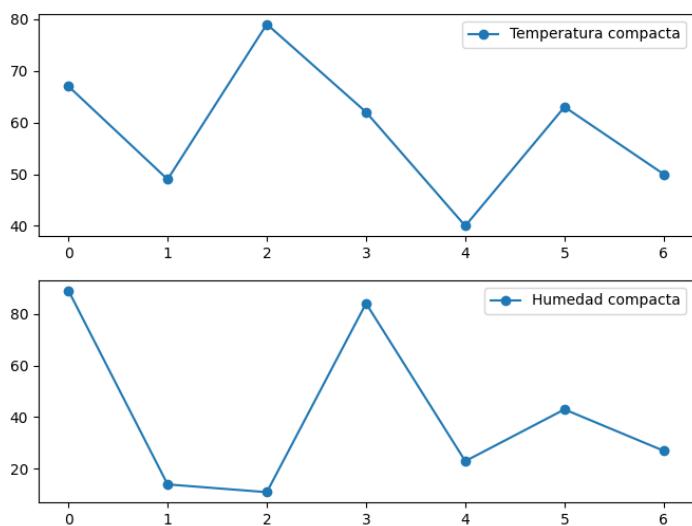
Elegí opción: 4

== Consumer COMPACTO escuchando (7 mensajes) ==
[1/7] Decodificado -> Temp:67 Hum:89 Viento:E
[2/7] Decodificado -> Temp:49 Hum:14 Viento:SE
[3/7] Decodificado -> Temp:79 Hum:11 Viento:O
[4/7] Decodificado -> Temp:62 Hum:84 Viento:NE
[5/7] Decodificado -> Temp:40 Hum:23 Viento:SE
[6/7] Decodificado -> Temp:63 Hum:43 Viento:NO
[7/7] Decodificado -> Temp:58 Hum:27 Viento:NO
```

Grafica del consumer normal



Grafica del consumer compacto



The screenshot shows a terminal window with the following details:

- File Explorer:** Shows a folder named "LAB 7" containing files: "lab7.py", "grafica_completo.png", "grafica_normal.png".
- Terminal:** Running "lab7.py".
 - Output:

```
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab > & C:/Users/rodle/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/rodle/OneDrive\Escritorio/uvg/redes/lab 7/lab7.py"
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab > & C:/Users/rodle/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/rodle/OneDrive\Escritorio/uvg/redes/lab 7/lab7.py"
```
 - Menu:
 - rive/Escritorio/uvg/redes/lab 7/Lab7.py"
 - Help:

```
===== LABORATORIO 7 - IoT Estación =====
1) Producer normal (JSON)
2) Consumer normal (JSON)
3) Producer compacto (3 bytes)
4) Consumer compacto (3 bytes)
5) Salir
```
 - User Input:

```
Elegí opción: 3
```
 - Output:

```
== Producer COMPACTO (3 bytes) corriendo (7 mensajes) ==
[1/7] Enviado compacto (3 bytes): b'\x00\xfa\x9a'
[2/7] Enviado compacto (3 bytes): b'\x00\xca\x4u'
[3/7] Enviado compacto (3 bytes): b'\x01\xCZ'
[4/7] Enviado compacto (3 bytes): b'\x01\x9e\xca'
[5/7] Enviado compacto (3 bytes): b'\x00\x90\xbd'
[6/7] Enviado compacto (3 bytes): b'\x00\xfdY'
[7/7] Enviado compacto (3 bytes): b'\x00\xc8\xd5'
✓ Producer COMPACTO finalizado.
```
 - Bottom status:

```
PS C:\Users\rodle\OneDrive\Escritorio\uvg\redes\lab >
```
- Plots:** Two line graphs are displayed side-by-side. The left graph is labeled "Producido compacto" and the right graph is labeled "Consumido compacto". Both graphs show data points over time steps 1 to 7.

Preguntas

1. ¿A qué capa pertenece JSON/SOAP según el modelo OSI?

JSON y SOAP pertenecen a la Capa 7, es decir, Aplicación, porque definen cómo las aplicaciones estructuran e intercambian datos, no cómo se transportan.

2. ¿Qué beneficios tiene usar un formato como JSON/SOAP?

Son estándares interoperables, legibles, fáciles de depurar y compatibles con múltiples lenguajes.

3. ¿Qué ventajas y desventajas tiene usar Kafka en un modelo Pub/Sub?

Ventajas: alto rendimiento, desacoplamiento, tolerancia a fallos y escalabilidad.

Desventajas: requiere más infraestructura y tiene una curva de aprendizaje mayor.

4. ¿Para qué aplicaciones tiene sentido usar Kafka?

Es ideal para IoT, telemetría, procesamiento en tiempo real y sistemas con alto volumen de eventos. No es adecuado para apps pequeñas o simples request/response.

5. ¿Qué complejidades surgen al tener un payload máximo de 3 bytes?

Se pierde precisión, se deben comprimir datos bit a bit y se limita la cantidad de información enviada.

6. ¿Cómo hacer que la temperatura quepa en 14 bits?

Representándola como un entero en el rango 0–110, que encaja completamente dentro de 14 bits.

7. ¿Qué pasaría si la humedad también fuera float con un decimal?

Habría que redondearla o escalarla para que quepa, lo cual reduce precisión o requiere más bits.

8. ¿Qué herramientas de Kafka ayudarían si las restricciones fueran mayores?

Compresión, batching, particiones y configuración de linger.ms para optimizar el envío y reducir tamaño.