



30 de septiembre del 2025

***Despliegue de bróker MQTT
con menor tiempo de latencia
en la transmisión de datos en
arquitecturas IoT: Caso de
estudio monitoreo de ruido***

Plan de Pruebas de Latencia para
Bróker MQTT



Thais Nicole Cartuche Peralta
UNIVERSIDAD NACIONAL DE LOJA

1. Objetivo del Plan de Pruebas

El propósito de este plan es **medir la latencia de transmisión de datos** en un entorno MQTT, simulando una arquitectura IoT de monitoreo de ruido. En este plan de pruebas se utiliza la herramienta **mqtt-stresser** para evaluar el desempeño y la latencia de distintos brokers MQTT. Este tipo de herramientas de benchmarking permiten variar parámetros como el número de clientes, la cantidad de mensajes, los niveles de QoS y el tamaño del payload, lo que asegura que las pruebas realizadas cumplan con los objetivos definidos en el proyecto.

El objetivo principal es identificar las condiciones de operación bajo las cuales se obtiene la menor latencia y documentar la replicabilidad de las pruebas.

2. Herramientas y Entorno

- **Herramienta:** mqtt-stresser
- **Broker MQTT:** Mosquitto / EMQX / VerneMQ
- **Infraestructura:** Servidor local / nube

3. Parámetros de Configuración y Justificación Técnica

Parámetro	Valores para evaluar	Justificación para medir latencia
Número de clientes (-c)	10, 50, 100, 200	Permite identificar el impacto de la concurrencia sobre la latencia.
Tamaño del mensaje (-s)	16 B, 128 B, 512 B, 1 KB	Determinar cómo el tamaño del paquete afecta los tiempos de transmisión.
Cantidad de mensajes (-n)	1000, 5000, 10000	Establece una base estadística para medir latencias consistentes.

4. Instrucciones y Comandos Definidos

Formato de los comandos

```
mqtt-stresser \  
-broker tcp://<IP_DEL_BROKER>:<PUERTO> \  
-num-clients <N_CLIENTES> \      # Número de clientes concurrentes  
-num-messages <N_MENSAJES> \    # Cantidad de mensajes por cliente  
-publisher-qos <QOS_PUBLICADOR> \ # Nivel QoS para el publicador  
-subscriber-qos <QOS_SUSCRIPTOR> \ # Nivel QoS para el suscriptor  
-topic-base-path <TOPICO_BASE> \ # Tópico base de publicación /  
suscripción  
-timeout <TIEMPO_MAXIMO> \      # Tiempo máximo de ejecución (ej. 60s)  
-log-level <NIVEL_LOG> \        # Nivel de detalle en logs (0-3)  
-no-progress \                  # (opcional) desactiva barra de progreso  
-constant-payload "<CADENA_PAYLOAD>" \ # Mensaje fijo o cadena de  
prueba  
-profile-cpu <ARCHIVO_CPU.prof> \ # Archivo de perfilado CPU  
-profile-mem <ARCHIVO_MEM.prof> \ # Archivo de perfilado Memoria  
> <ARCHIVO_RESULTADOS.txt>      # Archivo donde guardar resultados
```

Las siguientes instrucciones serán utilizadas para las pruebas. Cada comando permite variar un solo parámetro mientras los demás permanecen constantes para garantizar resultados comparables.

Prueba Base (Escenario controlado inicial)

```
mqtt-stresser
  -broker tcp://<IP_DEL_BROKER>:1883\
  -num-clients 1 \
  -num-messages 1000 \
  -publisher-qos 0 \
  -subscriber-qos 0 \
  -topic-base-path test/topic \
  -timeout 60s \
  -log-level 3 \
  -no-progress \
  -constant-payload "$(printf 'A%.0s' {1..128})" \
  -profile-cpu cpu_base.prof \
  -profile-mem mem_base.prof \
  > resultados_base.txt
```

Objetivo: Establecer una línea base mínima para medir la latencia con un único cliente y mensajes pequeños.

- **Variación de Clientes**

10 clientes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-
path test/topic -timeout 60s -log-level 3 -no-progress -constant-
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_c10.prof -
profile-mem mem_c10.prof > resultados_c10.txt
```

50 clientes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 50 -
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-
path test/topic -timeout 60s -log-level 3 -no-progress -constant-
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_c50.prof -
profile-mem mem_c50.prof > resultados_c50.txt
```

100 clientes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 100 -
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-
path test/topic -timeout 120s -log-level 3 -no-progress -constant-
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_c100.prof -
profile-mem mem_c100.prof > resultados_c100.txt
```

200 clientes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 200 -
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-
path test/topic -timeout 120s -log-level 3 -no-progress -constant-
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_c200.prof -
profile-mem mem_c200.prof > resultados_c200.txt
```

Objetivo: Identificar a partir de cuántos clientes simultáneos se empieza a degradar la latencia.

- Variación del Tamaño de Mensaje

16 bytes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-  
path test/topic -timeout 60s -log-level 3 -no-progress -constant-  
payload "$(printf 'A%.0s' {1..16})" -profile-cpu cpu_16.prof -  
profile-mem mem_16.prof > resultados_16B.txt
```

128 bytes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-  
path test/topic -timeout 60s -log-level 3 -no-progress -constant-  
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_128.prof -  
profile-mem mem_128.prof > resultados_128B.txt
```

512 bytes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-  
path test/topic -timeout 200s -log-level 3 -no-progress -constant-  
payload "$(printf 'A%.0s' {1..512})" -profile-cpu cpu_512.prof -  
profile-mem mem_512.prof > resultados_512B.txt
```

1024 bytes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 1000 -publisher-qos 0 -subscriber-qos 0 -topic-base-  
path test/topic -timeout 300s -log-level 3 -no-progress -constant-  
payload "$(printf 'A%.0s' {1..1024})" -profile-cpu cpu_1024.prof -  
profile-mem mem_1024.prof > resultados_1KB.txt
```

Objetivo: Identificar cómo afecta el tamaño del mensaje a la latencia en redes IoT.

- Variación de Cantidad de Mensajes

5000 mensajes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 5000 -publisher-qos 0 -subscriber-qos 0 -topic-base-  
path test/topic -timeout 300s -log-level 3 -no-progress -constant-  
payload "$(printf 'A%.0s' {1..128})" -profile-cpu cpu_5k.prof -  
profile-mem mem_5k.prof > resultados_5k.txt
```

10000 mensajes

```
mqtt-stresser -broker tcp://20.64.244.7:1883 -num-clients 10 -  
num-messages 10000 -publisher-qos 0 -subscriber-qos 0 -topic-  
base-path test/topic -timeout 600s -log-level 3 -no-progress -  
constant-payload "$(printf 'A%.0s' {1..128})" -profile-cpu  
cpu_10k.prof -profile-mem mem_10k.prof > resultados_10k.txt
```

Objetivo: Observar si con grandes volúmenes de tráfico se generan picos de latencia o cuellos de botella.

5. Justificación General del Diseño de las Pruebas

Cada conjunto de pruebas busca aislar una variable específica que puede influir en la latencia, permitiendo:

- Medir la eficiencia del bróker al escalar clientes.
- Determinar el impacto del tamaño de los mensajes.
- Evaluar el efecto del volumen de mensajes por sesión.