

13 de octubre 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***

Informe de Evaluación de los  
Brokers y Justificación Técnica de  
la Selección del Óptimo

Thais Nicole Cartuche Peralta  
UNIVERSIDAD NACIONAL DE LOJA

## Tabla de Contenido

<b>Resumen .....</b>	2
<b>Metodología de evaluación.....</b>	2
1. Entorno de infraestructura y despliegue .....	2
2. Verificación de condiciones de red .....	3
3. Herramienta benchmarking y configuración de pruebas .....	4
4. Procedimiento de ejecución.....	5
<b>Resultados detallados por escenario .....</b>	5
<b>Análisis comparativo global.....</b>	6
1. Métricas de desempeño consolidadas.....	6
2. Análisis por dimensión de evaluación .....	6
2.1. Rendimiento bajo concurrencia.....	6
2.2. Impacto del Tamaño del Payload.....	7
2.3. Manejo de alto volumen.....	7
3. Consistencia y confiabilidad.....	7
4. Tasa de entrega .....	7
<b>Justificación técnica del broker seleccionado .....</b>	8
Principales Criterios Técnicos .....	8
Ventajas adicionales .....	8
Consideraciones sobre alternativas .....	8
<b>Conclusiones.....</b>	8
<b>Anexos.....</b>	9

## Resumen

El presente informe documenta de manera exhaustiva los resultados obtenidos de las pruebas de rendimiento realizadas sobre tres brokers MQTT de código abierto: **EMQX**, **VerneMQ** y **Mosquitto**, desplegados en infraestructura de nube Microsoft Azure. El estudio comparativo se enmarca en el desarrollo de una arquitectura IoT para el monitoreo de ruido vehicular en tiempo real, donde la minimización de la latencia en la transmisión de datos constituye un requisito crítico del sistema.

La evaluación contempló once escenarios de prueba meticulosamente diseñados para analizar el comportamiento de cada broker bajo diferentes condiciones operativas, incluyendo variaciones en el número de clientes concurrentes (10, 50, 100 y 200), tamaño de payload (16B, 128B, 512B y 1024B) y volumen de mensajes transmitidos (1000, 5000 y 10000 mensajes). Todas las pruebas se ejecutaron utilizando la herramienta especializada mqtt-stresser, garantizando condiciones controladas, reproducibles y científicamente válidas para la comparación objetiva del desempeño.

Los resultados demuestran que Mosquitto presenta el mejor equilibrio entre latencia promedio (1.046 ms), consistencia operativa (desviación estándar de 0.752 ms) y eficiencia bajo concurrencia, posicionándose como la solución óptima para aplicaciones IoT sensibles a la latencia. Este broker superó consistentemente a sus competidores en 8 de los 11 escenarios evaluados, demostrando particular fortaleza en condiciones de alta concurrencia y payloads de tamaño medio a grande, que representan los patrones de tráfico esperados en sistemas de monitoreo ambiental continuo.

El análisis incluye no solo métricas cuantitativas de latencia, sino también evaluaciones cualitativas de confiabilidad, tasa de entrega de mensajes y predictibilidad del comportamiento, proporcionando una visión integral que fundamenta técnicamente la selección del broker para el despliegue en producción del sistema de monitoreo de ruido.

## Metodología de evaluación

### 1. Entorno de infraestructura y despliegue

Las pruebas se realizaron desde un único equipo cliente local, conectado a una misma red física durante toda la ejecución experimental, garantizando así la eliminación de variables de confusión relacionadas con cambios en las condiciones de conectividad.

Cada broker fue desplegado en una máquina virtual independiente en Microsoft Azure, configurada con las siguientes especificaciones técnicas homogéneas:

- **Sistema Operativo:** Ubuntu Server 22.04 LTS
- **Tamaño de instancia:** Standard\_B1s (1 vCPU, 1 GB RAM)
- **Región:** West US 2
- **Red:** Red virtual única con subred predeterminada
- **IP pública:** Estática
- **Autenticación:** SSH Key
- **Puertos habilitados:** 22 (SSH), 1883 (MQTT sin cifrado)

Las tres máquinas virtuales se ubicaron dentro de la misma región geográfica de Azure (West US 2), garantizando condiciones equivalentes en términos de latencia de red,

distancia geográfica al cliente, infraestructura de datacenter y topología de red. Esta configuración permite atribuir las diferencias de rendimiento observadas exclusivamente a las características intrínsecas de cada broker, eliminando sesgos relacionados con la infraestructura subyacente.

Cada broker fue configurado para aceptar conexiones MQTT sin autenticación en el puerto estándar 1883, siguiendo las mejores prácticas documentadas en el *Manual de Configuración de Máquinas Virtuales*. Esta configuración, si bien no recomendada para entornos de producción, es apropiada para benchmarking, ya que elimina la sobrecarga adicional introducida por mecanismos de autenticación y cifrado.

## 2. Verificación de condiciones de red

Con el objetivo de asegurar la homogeneidad del entorno de red y validar que las diferencias de rendimiento observadas no se debieran a variaciones en la conectividad, se realizaron comprobaciones sistemáticas de conectividad mediante el comando *ping* desde el equipo cliente hacia cada una de las instancias de Azure antes de iniciar cada batería de pruebas. Todas las pruebas se llevaron a cabo bajo la misma red **Internet\_UNL**, ubicada en la Universidad Nacional de Loja, garantizando así la consistencia de las condiciones de acceso a la red durante todo el proceso experimental.

```
ping -c 74.179.58.252 #mosquitto
ping -c 20.64.248.245 #verneMQ
ping -c 20.64.244.7 #EMQX

C:\Windows\System32>ping 74.179.58.252

Haciendo ping a 74.179.58.252 con 32 bytes de datos:
Respuesta desde 74.179.58.252: bytes=32 tiempo=168ms TTL=39
Respuesta desde 74.179.58.252: bytes=32 tiempo=159ms TTL=39
Respuesta desde 74.179.58.252: bytes=32 tiempo=159ms TTL=39
Respuesta desde 74.179.58.252: bytes=32 tiempo=159ms TTL=39

Estadísticas de ping para 74.179.58.252:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 159ms, Máximo = 168ms, Media = 161ms

C:\Windows\System32>ping 20.64.248.245

Haciendo ping a 20.64.248.245 con 32 bytes de datos:
Respuesta desde 20.64.248.245: bytes=32 tiempo=156ms TTL=39
Respuesta desde 20.64.248.245: bytes=32 tiempo=155ms TTL=39
Respuesta desde 20.64.248.245: bytes=32 tiempo=156ms TTL=39
Respuesta desde 20.64.248.245: bytes=32 tiempo=155ms TTL=39

Estadísticas de ping para 20.64.248.245:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 155ms, Máximo = 156ms, Media = 155ms

C:\Windows\System32>ping 20.64.244.7

Haciendo ping a 20.64.244.7 con 32 bytes de datos:
Respuesta desde 20.64.244.7: bytes=32 tiempo=159ms TTL=39
Respuesta desde 20.64.244.7: bytes=32 tiempo=159ms TTL=39
Respuesta desde 20.64.244.7: bytes=32 tiempo=159ms TTL=39
Respuesta desde 20.64.244.7: bytes=32 tiempo=158ms TTL=39

Estadísticas de ping para 20.64.244.7:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 158ms, Máximo = 159ms, Media = 158ms

C:\Windows\System32>
```

Figura 1. Evaluación de condiciones de red

Los resultados arrojaron valores promedio de **Round-Trip Time (RTT)** estadísticamente equivalentes, confirmando que la conexión entre el cliente y los tres brokers mantenía condiciones de latencia de red comparables:

*Tabla 1. Resultados evaluación de condiciones de red*

Broker	IP Pública	RTT Mínimo (ms)	RTT Máximo (ms)	RTT promedio (ms)	Paquetes perdidos
EMQX	20.64.244.7	158	159	158	0 (0%)
VerneMQ	20.64.248.245	155	156	155	0 (0%)
Mosquitto	74.179.58.252	159	168	161	0 (0%)

La diferencia máxima de RTT promedio entre brokers fue de apenas 6 ms (3.87% de variación relativa), con todos los valores concentrados en el rango de 155-161 ms. Ninguna de las pruebas experimentó pérdida de paquetes (0% de pérdida en los tres casos), lo que confirma la estabilidad y calidad de la conexión.

Estos resultados demuestran que todas las pruebas se ejecutaron bajo condiciones de red esencialmente idénticas. Lo que permite concluir con alta confianza que las diferencias de latencia medidas en las pruebas MQTT son atribuibles exclusivamente al procesamiento interno de cada broker y no a factores de red externa.

De manera que esta prueba nos permite asegurar que los resultados reflejen genuinamente las diferencias de desempeño entre los brokers y no artefactos introducidos por variabilidad en las condiciones de conectividad o enrutamiento de red. La consistencia en los valores de RTT también evidencia la estabilidad temporal de la red durante el periodo de pruebas.

### 3. Herramienta benchmarking y configuración de pruebas

Se utilizó la herramienta mqtt-stresser, una aplicación de código abierto desarrollada en Go específicamente para realizar pruebas de estrés y medición de latencia en brokers MQTT. Esta herramienta fue seleccionada por su capacidad para:

- Simular múltiples clientes MQTT concurrentes
- Generar cargas de trabajo configurables y reproducibles
- Medir con precisión de microsegundos la latencia end-to-end
- Registrar métricas detalladas de rendimiento (latencia mínima, máxima, promedio, percentiles)
- Generar perfiles de CPU y memoria para análisis de recursos

Cada escenario de prueba fue ejecutado siguiendo el *Plan de Pruebas de Latencia* definido previamente, el cual establece una metodología de **aislamiento de variables**, donde se modifica un único parámetro mientras los demás permanecen constantes. Este diseño experimental permite identificar con precisión el impacto individual de cada factor sobre la latencia.

### **Parámetros constantes en todas las pruebas:**

- QoS de publicador: 0
- QoS de suscriptor: 0
- Tópico base: test/topic
- Nivel de log: 3 (detallado)

### **Variables evaluadas:**

- Número de clientes concurrentes: 1, 10, 50, 100, 200
- Tamaño de payload: 16B, 128B, 512B, 1024B
- Volumen de mensajes: 1000, 5000, 10000

Para cada configuración, se registraron las métricas de latencia (mínima, máxima, promedio, percentiles), tasa de mensajes por segundo, porcentaje de entrega exitosa y perfiles de uso de recursos. Los resultados fueron exportados a archivos de texto estructurados para su posterior análisis estadístico.

La elección de QoS 0 responde a la necesidad de medir el rendimiento base del broker sin la sobrecarga adicional de los mecanismos de confirmación y retransmisión de QoS 1 y 2. En aplicaciones de monitoreo de ruido en tiempo real, donde la pérdida ocasional de un dato individual tiene impacto limitado frente a la disponibilidad de un flujo continuo de información, QoS 0 representa el modo operativo más eficiente.

#### 4. Procedimiento de ejecución

Cada broker fue sometido a las once pruebas definidas en secuencia, con un periodo de enfriamiento de 3 minutos entre ejecuciones para permitir la estabilización del sistema. Antes de cada prueba, se verificó que el broker estuviera operativo y sin carga residual mediante inspección de los logs del sistema. Los resultados fueron documentados sistemáticamente en archivos individuales por escenario

### **Resultados detallados por escenario**

A continuación, se presenta la tabla resumen con la latencia promedio de cada bróker en cada escenario de prueba:

*Tabla 2. Resumen de resultados*

Prueba	Escenario	EMQX (ms)	VerneMQ(ms)	Mosquitto (ms)	Mejor rendimiento
1	Normal/base	0.574052813	0.45620438	0.427715997	Mosquitto
2	10 clientes	0.632511069	0.875656743	0.809716599	EMQX
3	50 clientes	3.367003367	2.923976608	0.941619586	Mosquitto
4	100 clientes	7.092198582	5.347593583	1.552795031	Mosquitto
5	200 clientes	12.65822785	8.333333333	2.277904328	Mosquitto
6	16 bytes	1.307189542	0.716332378	0.831255195	VerneMQ
7	128 bytes	1.166861144	0.714796283	0.582411182	Mosquitto
8	512 bytes	1.136363636	1.245330012	2.538071066	EMQX
9	1024 bytes	4.048582996	2.197802198	1.182033097	Mosquitto
10	5000 mensajes	1.216545012	0.716332378 (58%)*	0.205930807 (90%)*	EMQX

11	10000 mensajes	0.415627598	1.652892562 (33%)*	0.16 (80%)*	EMQX
----	-------------------	-------------	-----------------------	-------------	------

Nota: \* Porcentaje de mensajes entregados exitosamente. Los brokers que no muestran porcentaje alcanzaron 100% de entrega.

## Análisis comparativo global

### 1. Métricas de desempeño consolidadas

Tabla 3. Resultados globales

Broker	Latencia promedio global	Desviación Estándar	Entrega promedio (%)	Evaluación global
EMQX	3.055923964	3.596416545	100	Completó todas las pruebas con entrega total (100%), garantizando integridad de datos. Sin embargo, presentó la mayor latencia promedio y la más alta variabilidad entre escenarios, indicando sensibilidad a las condiciones de carga.
VerneMQ	2.289113678	2.346674925	90.09	Latencia intermedia con variabilidad moderada. Experimentó pérdida significativa de mensajes en escenarios de alto volumen (pruebas 10 y 11), con tasas de entrega del 58% y 33% respectivamente, comprometiendo la confiabilidad.
Mosquitto	1.0463139	0.75196421	97.27	Menor latencia promedio (66% inferior a EMQX) y menor variabilidad, demostrando alta consistencia y predictibilidad. Presentó pérdida de mensajes únicamente en escenarios de volumen extremo (10-20% en pruebas 10-11), manteniendo excelente desempeño en condiciones operativas típicas.

Nota: La desviación estándar cuantifica la variabilidad de las latencias de cada broker entre las distintas pruebas. Un valor menor (como el de Mosquitto: 0.752 ms) refleja un comportamiento más estable, consistente y predecible, característica deseable en sistemas de tiempo real donde la variabilidad puede ser tan crítica como la latencia promedio. La alta desviación de EMQX (3.596 ms) indica que su rendimiento es fuertemente dependiente de las condiciones específicas de carga, lo que dificulta la planificación de capacidad y el dimensionamiento del sistema.

### 2. Análisis por dimensión de evaluación

#### 2.1. Rendimiento bajo concurrencia

Los resultados evidencian diferencias arquitectónicas notables al aumentar el número de clientes concurrentes.

Mosquitto mantuvo las latencias más bajas y estables incluso con 200 clientes (2.278 ms), superando en más del 70% a EMQX y VerneMQ. Este comportamiento indica una arquitectura eficiente basada en *event loops* ligeros y gestión optimizada de conexiones. Su crecimiento de latencia fue prácticamente lineal ( $\approx 0.011$  ms/cliente), reflejando una excelente escalabilidad.

VerneMQ mostró un incremento moderado (8.333 ms con 200 clientes), mientras que EMQX presentó la mayor degradación (12.658 ms con 200 clientes). Dando como resultado que en el caso de uso con 20 - 50 sensores concurrentes, Mosquitto ofrece el mejor equilibrio entre eficiencia y capacidad de crecimiento.

## 2.2. Impacto del Tamaño del Payload

El desempeño frente a distintos tamaños de mensaje mostró optimizaciones específicas por broker.

Mosquitto destacó en mensajes medianos y grandes (128 B  $\rightarrow$  0.582 ms; 1 KB  $\rightarrow$  1.182 ms), siendo el más eficiente en los rangos típicos de telemetría IoT. VerneMQ se desempeñó mejor con mensajes pequeños (16 B  $\rightarrow$  0.716 ms), y EMQX tuvo un rendimiento más uniforme sin optimización destacada. La curva de crecimiento de Mosquitto fue la más gradual, evidenciando eficiencia en el procesamiento de datos voluminosos.

## 2.3. Manejo de alto volumen

En escenarios de alto volumen (5000 - 10000 mensajes), Mosquitto alcanzó las latencias más bajas (0.206 - 0.160 ms), aunque con pérdidas de mensajes en los escenarios más extremos (entrega 80 - 90%).

EMQX mantuvo 100% de entrega, pero con mayor latencia (1.217 - 0.416 ms), y VerneMQ mostró degradación severa en desempeño y confiabilidad. En el contexto del monitoreo de ruido, donde el volumen real por sensor es moderado, las pérdidas observadas en Mosquitto bajo cargas extremas no afectan su aplicabilidad práctica.

## 3. Consistencia y confiabilidad

La variabilidad en latencia determina la predictibilidad del sistema Mosquitto obtuvo la menor desviación estándar ( $\sigma = 0.752$  ms, CV = 71.9 %), reflejando un comportamiento altamente estable y predecible. VerneMQ presentó variabilidad moderada ( $\sigma = 2.347$  ms), y EMQX la mayor ( $\sigma = 3.596$  ms) La baja variabilidad de Mosquitto facilita el dimensionamiento de infraestructura y la configuración de *timeouts* confiables en sistemas de tiempo real.

## 4. Tasa de entrega

EMQX garantizó entrega completa (100%), mientras que Mosquitto alcanzó 97.27% de entrega promedio, con pérdidas solo bajo condiciones extremas. VerneMQ obtuvo 90.09%.

Dado que el sistema de monitoreo de ruido tolera pérdidas mínimas sin afectar el análisis global, la confiabilidad de Mosquitto es plenamente adecuada para la operación normal.

## Justificación técnica del broker seleccionado

Tras el análisis integral de latencia, consistencia, escalabilidad y confiabilidad, Mosquitto se selecciona como el broker óptimo para el sistema de monitoreo de ruido vehicular en tiempo real.

### Principales Criterios Técnicos

- **Latencia promedio global:** 1.046 ms (65 % menor que EMQX), asegurando transmisión casi instantánea de datos sensoriales.
- **Estabilidad y predictibilidad:** menor variabilidad ( $\sigma = 0.752$  ms), garantizando desempeño consistente.
- **Escalabilidad:** crecimiento lineal de latencia (0.011 ms/cliente), manteniendo eficiencia hasta 200 clientes concurrentes.
- **Optimización IoT:** desempeño óptimo en payloads de 128–1024 bytes, típico de datos de ruido con metadatos.
- **Confiabilidad:** 97–100 % de entrega en condiciones normales, suficiente para flujos continuos de datos.

### Ventajas adicionales

Mosquitto es un broker maduro y ligero, con bajo consumo de recursos, alta portabilidad y configuración simple. Su arquitectura monolítica minimiza la complejidad operativa y reduce el costo total de propiedad. Estas características lo hacen ideal para despliegues en infraestructuras de recursos limitados o entornos *edge*.

### Consideraciones sobre alternativas

- **EMQX:** sobresale en confiabilidad total (100%) y administración avanzada, pero su latencia elevada y complejidad arquitectónica lo hacen menos apropiado para sistemas donde el tiempo de respuesta es crítico.
- **VerneMQ:** eficiente con mensajes muy pequeños, pero pierde confiabilidad y estabilidad en cargas sostenidas.

## Conclusiones

- Los resultados confirman que Mosquitto obtuvo el mejor rendimiento entre los brokers evaluados, al mantener las latencias más bajas y un comportamiento estable en todos los escenarios. Su desempeño fue consistente tanto en condiciones de baja como de alta concurrencia, demostrando una respuesta rápida y escalable frente al incremento de clientes. En comparación, VerneMQ presentó un aumento progresivo de latencia y EMQX mostró los valores más altos y mayor variabilidad.
- En términos de confiabilidad, Mosquitto alcanzó una tasa de entrega elevada y un desempeño óptimo con los tamaños de mensajes típicos del sistema de monitoreo de ruido. Su equilibrio entre velocidad, estabilidad y eficiencia lo hace adecuado para aplicaciones de transmisión en tiempo real, garantizando datos oportunos sin requerir altos recursos de infraestructura. Aunque EMQX logró entrega total, su

mayor latencia limita su idoneidad para entornos sensibles al tiempo, mientras que VerneMQ mostró pérdidas bajo carga sostenida.

## Anexos

**Anexo A:** Creación y configuración de las máquinas virtuales

[Creación de máquinas virtuales, instalación y configuración de brokers](#)

**Anexo B:** Plan con los comandos ejecutados en cada uno de los escenarios

[Plan de pruebas](#)

**Anexo C:** Archivos de resultados

[Carpeta de archivos de resultados \(.txt\)](#)

**Anexo D:** Tablas de resultados

[Resultados \(.xlsx\)](#)