Hecho por: Dylan Mateo Llano Jaramillo

Informe de pruebas de performance

Este documento describe los resultados de las pruebas de rendimiento realizadas sobre el servicio Artifact Clue Detector, junto con una proyección técnica de escalabilidad desde entorno local y en infraestructura en la nube (Render).

De acuerdo a los criterios definidos en el enunciado el problema, la API debería estar en la capacidad de recibir de 100 a 1'000.000 de peticiones por segundo.

Veamos si el servicio está en la capacidad de recibir esto:

1. Se ejecutará pruebas de performance en localhost, para identificar el mínimo requerido para escalar y acercarse al objetivo (100 TPS -> 1 M TPS).
2. Se ejecutará pruebas de performance en un servicio de nube llamado Render, para identificar el mínimo requerido para escalar y acercarse al objetivo (100 TPS -> 1 M TPS).

Infraestructura:

Servidor	Recursos	Localhost	Cloud (Render)
	RAM	32 GB RAM	512
	CPU	AMD Ryzen 5 5600G, 6 núcleos / 12 hilos	0.5
	DISCO	1 TB SSD	
Base de datos	**Servidores Virtuales**	1	1
	Conexiones	21 / 100	24 / 100
	RAM		0.25 GB
	CPU		0.1 CPU

Entorno: Localhost
Transacciones:

- Tx_Clue → POST /clue/
- Tx_Stats → GET /stats

Prueba de línea base: Es la primera prueba que se realiza y se considera fundamental. Consiste en aplicar una carga inicial del 10% del total previsto para verificar que el sistema funcione adecuadamente y cumpla con los requisitos básicos.

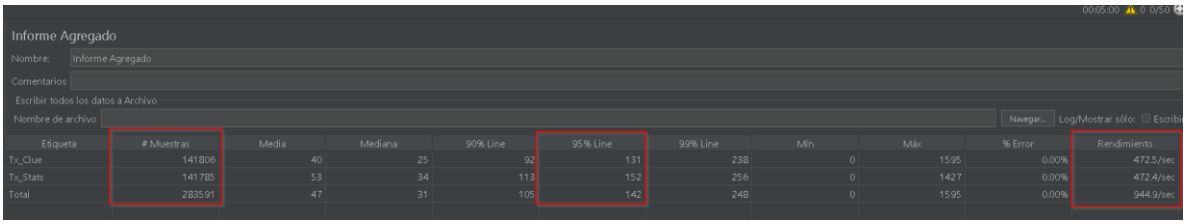
Objetivo: Asegurarse de que el sistema pueda manejar una carga mínima y cumplir con los requisitos no funcionales como el tiempo de respuesta. Si la prueba de línea base falla, no se puede avanzar a las pruebas siguientes.

Resultado Real:

Se define la siguiente configuración en Jmeter:

Threads	10
Rampup	60 segundos = 1 Minuto
Duration	300 segundos = 5 Minutos

Resultados Obtenidos

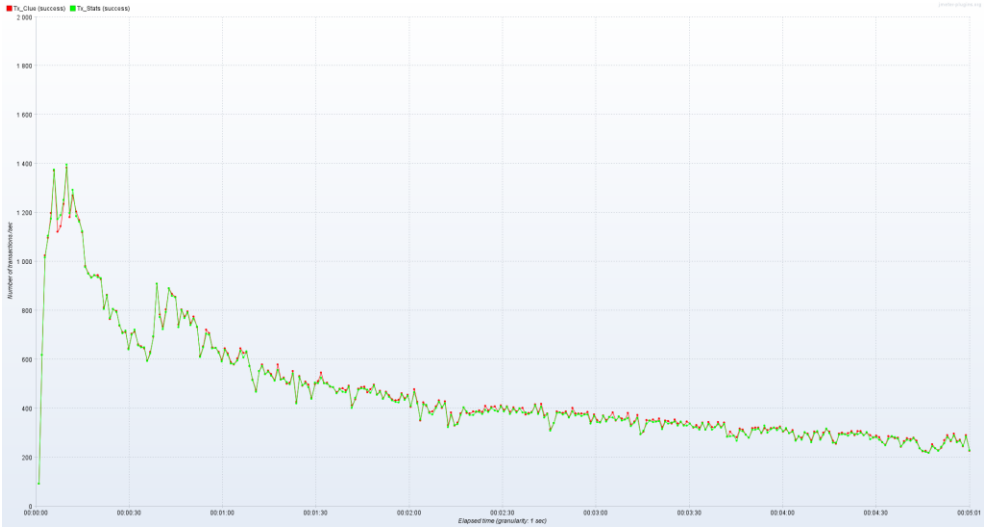


- El sistema mantiene ~945 transacciones por segundo estables combinando lecturas y escrituras.
- Esto es excelente para un entorno local monolítico con base de datos relacional.
- La media y los percentiles (P90–P99) están muy por debajo de 200 ms, lo que significa baja latencia y alta eficiencia del código.

Grafica Tiempos de Respuesta:

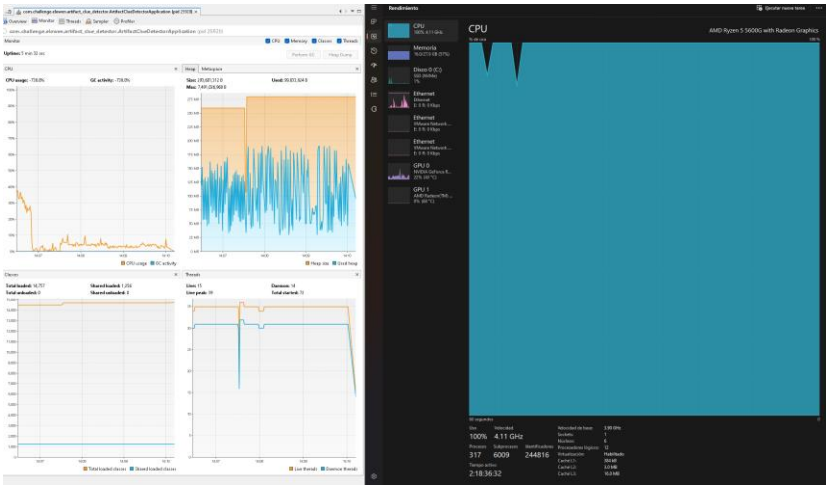
- Incremento progresivo del tiempo a medida que avanza el test → comportamiento natural al aumentar la carga y acumulación de registros.
- La distancia entre /clue y /stats se mantiene constante (~20 ms), lo que confirma balance de carga uniforme.

Grafica Transacciones por segundo (TPS)



- Pico inicial de ~1300 TPS, estabilizándose luego en torno a 900–950 TPS.
- Caída progresiva leve, coherente con saturación de CPU (100%).
- Sin caídas abruptas ni errores, excelente estabilidad.

Visual VM – CPU



- La CPU se mantiene al **100% de uso sostenido**, lo cual indica que el sistema **está limitado por procesamiento**, no por I/O o la base de datos

Prueba de Carga: Esta prueba se enfoca en verificar que el sistema pueda manejar la carga esperada en condiciones normales de operación.

Objetivo: Confirmar que el sistema puede soportar el número esperado de transacciones por segundo y cumplir con los tiempos de respuesta establecidos bajo condiciones normales.

Resultado Teórico – Alcanzar 100 TPS

PLANTILLA CALCULO CONCURRENCIA

Variable	Descripción	Valores de Ejemplo
Nte	Numero de Transacciones	100
Md	Unidad de tiempo del Nte	1
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Ttrx	**Total Transacciones Esperadas en un periodo Te**	30,000
Tps	**Transacciones por Segundo**	100

Variable	Descripción	Valores de Ejemplo
Ttrx	Total Transacciones Esperadas en un periodo Te	30,000
Top	Tiempo de duración promedio de una (1) operación (Segundos)	0,002
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Uc	**Usuarios Concurrentes**	1

	USUARIOS CONCURRENTES	TOTAL TRANSACCIONES
FORMULAS	$Uc = \frac{Top * Ttrx * O}{Te}$	$Ttrx = \frac{Nte * Te}{Md * O}$

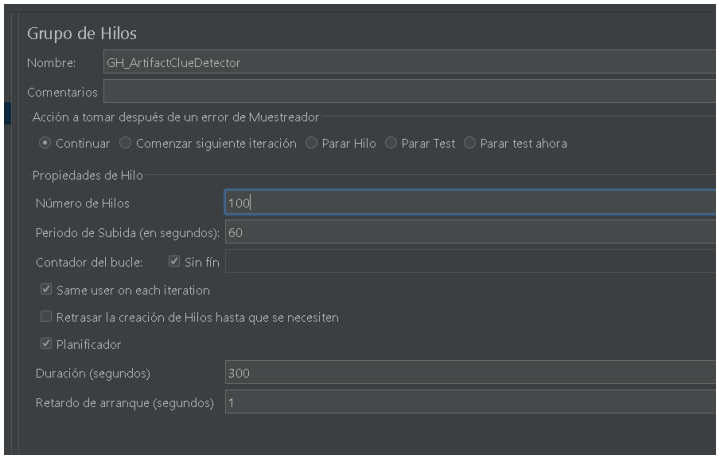
Variable	Descripción
Nte	Transacciones esperadas por el clientes
Md	Unidad de tiempo dada para las transaccione esperadas por el clientes (Segundo, Minuto, Hora, Dia)
Te	Tiempo de duración de la prueba de performance
Ttrx	Transacciones totales que se espera generar durante el escenario de prueba
Top	Tiempo de duración promedio de una (1) operación (Segundos)
Uc	Usuarios concurrentes

Guía para política de carga			
PRUEBA	Línea Base	Carga	Estrés (125%)
Uc	0	1	1
Tps	10	100	125

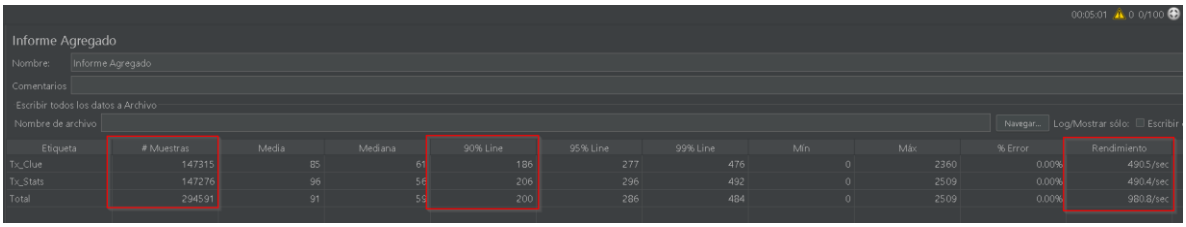
Ejecución #1 – Alcanzar 100 TPS:

Se define la siguiente configuración:

Threads	100
Rampup	60 segundos = 1 Minuto
Duration	300 segundos = 5 Minutos

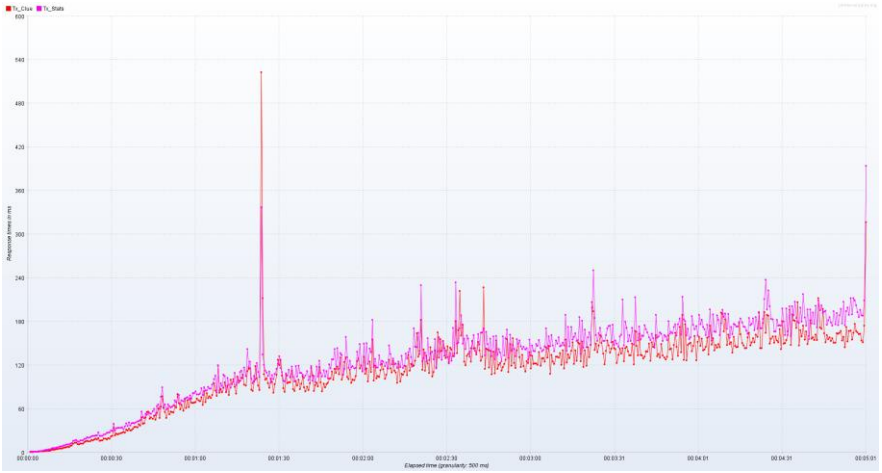


Resultados Obtenidos



- Durante los 5 minutos de ejecución, la aplicación procesó ~980 transacciones por segundo combinadas de manera estable, con tiempos promedio inferiores a 100 ms y sin errores.
- El sistema mantuvo una respuesta consistente en ambas transacciones, sin degradación notable ni saturación de memoria.

Grafica Tiempos de Respuesta:

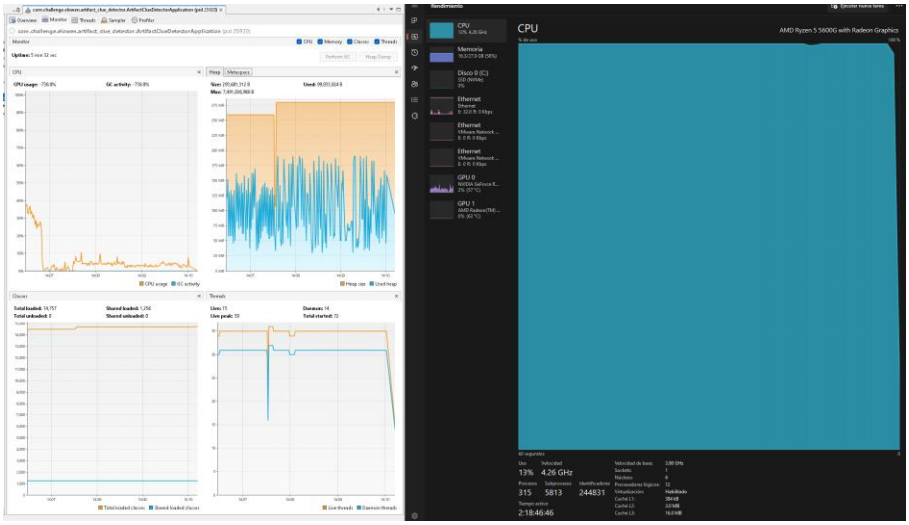


- Los tiempos de respuesta muestran un comportamiento estable y controlado durante toda la ejecución.
- El percentil 90 (P90) se mantiene alrededor de los 200 ms, lo que representa una latencia baja y consistente incluso bajo carga sostenida.
- Se observan ligeras fluctuaciones hacia el final de la prueba, asociadas a la saturación de CPU, sin afectar el rendimiento global.
- Ambas transacciones (/clue y /stats) presentan curvas similares, indicando balance adecuado de recursos.

Grafica Transacciones por segundo (TPS):

- El gráfico de TPS muestra un pico inicial cercano a 2000 TPS, estabilizándose luego en torno a 900–1000 TPS.
- El throughput se mantiene constante durante los 5 minutos de ejecución, sin caídas abruptas ni fallos.

Visual VM – CPU



- Durante toda la ejecución, la CPU se mantuvo en 100% de uso constante, lo que evidencia que la prueba alcanzó el límite físico de procesamiento del entorno local.
- El Heap mostró un consumo estable (alrededor de 100 MB usados), sin indicios de fugas de memoria.
- La actividad del Garbage Collector (GC) fue mínima, con pausas cortas y sin impacto en los tiempos de respuesta.

Conclusión Ejecución #1: Con 100 hilos, 60 segundos de ramp-up y 5 minutos de ejecución, el servicio alcanzó ~980 transacciones por segundo combinadas con P90 = 200 ms y 0% de error.

El rendimiento alcanzado representa el límite de la capacidad de hardware local, mostrando una aplicación eficiente, estable y con uso óptimo de recursos.

Resultado Teórico – Alcanzar 10.000 TPS:

PLANTILLA CALCULO CONCURRENCIA

Variable	Descripción	Valores de Ejemplo
Nte	Numero de Transacciones	10,000
Md	Unidad de tiempo del Nte	1
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Ttrx	**Total Transacciones Esperadas en un periodo Te**	3,000,000
Tps	**Transacciones por Segundo**	10,000

Guía para política de carga			
PRUEBA	Línea Base	Carga	Estrés (125%)
Uc	30	300	375
Tps	1,000	10,000	12,500

Variable	Descripción	Valores de Ejemplo
Ttrx	Total Transacciones Esperadas en un periodo Te	3,000,000
Top	Tiempo de duración promedio de una (1) operación (Segundos)	0.03
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Uc	**Usuarios Concurrentes**	300

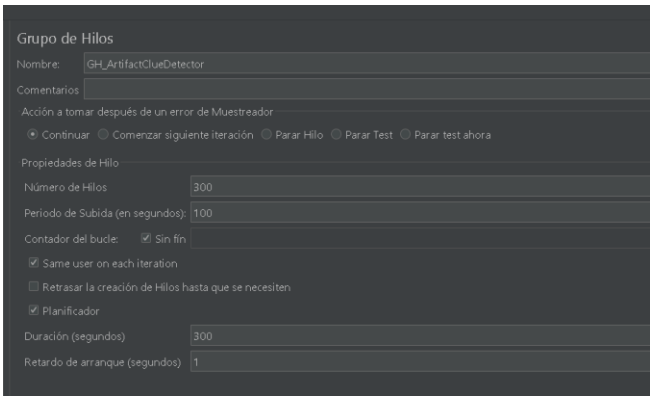
	USUARIOS CONCURRENTES	TOTAL TRANSACCIONES
FORMULAS	$Uc = \frac{Top * Ttrx * O}{Te}$	$Ttrx = \frac{Nte * Te}{Md * O}$

Variable	Descripción
Nte	Transacciones esperadas por el clientes
Md	Unidad de tiempo dada para las transaccione esperadas por el clientes (Segundo, Minuto, Hora, Dia)
Te	Tiempo de duración de la prueba de performance
Ttrx	Transacciones totales que se espera generar durante el escenario de prueba
Top	Tiempo de duración promedio de una (1) operación (Segundos)
Uc	Usuarios concurrentes

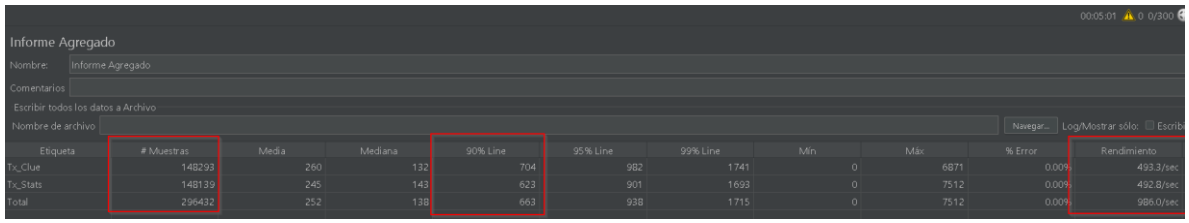
Ejecución #2 – Alcanzar 10.000 TPS:

Se define la siguiente configuración:

Threads	300
Rampup	100 segundos = 1,6 Minutos
Duration	300 segundos = 5 Minutos

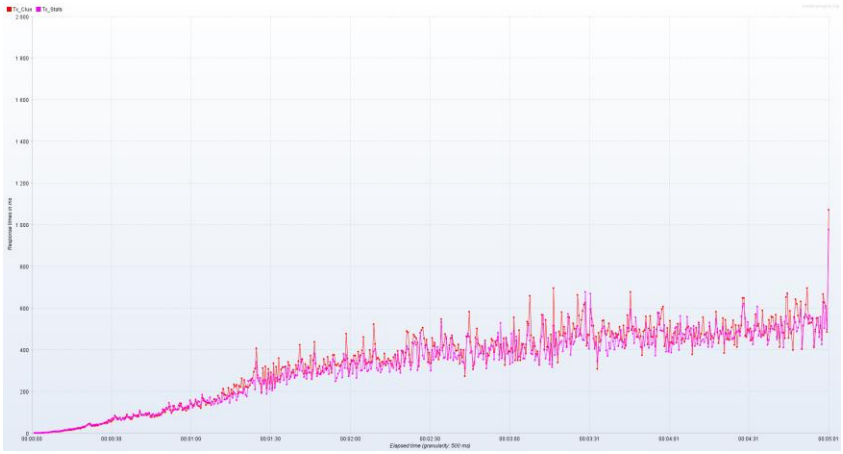


Resultados Obtenidos



- Durante la ejecución con 300 hilos concurrentes, la aplicación mantuvo un rendimiento total estable de aproximadamente 986 transacciones por segundo combinadas, sin errores y con latencias aceptables.
- El incremento de carga respecto a la prueba anterior (100 hilos) no generó un aumento significativo en throughput, lo que confirma que el sistema ha alcanzado el límite máximo de procesamiento de la CPU local (100%).

Grafica Tiempos de Respuesta:

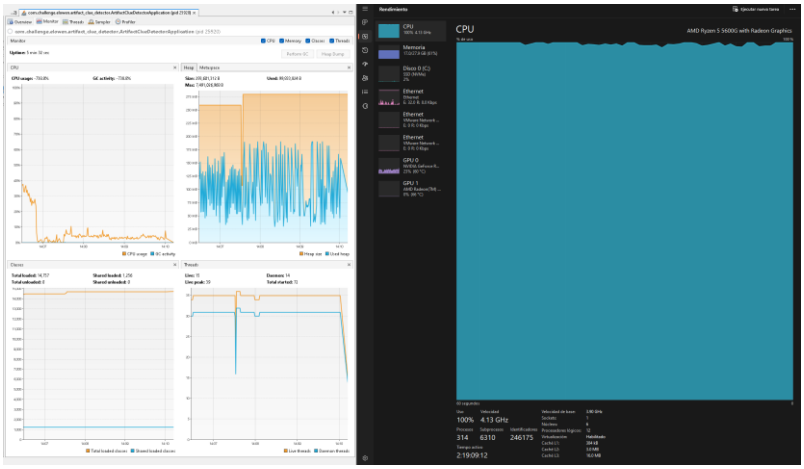


- Los tiempos de respuesta se mantuvieron por debajo de 700 ms (percentil 90) y 1,7 s en el percentil 99, mostrando un leve incremento respecto a pruebas anteriores debido al aumento de concurrencia.
- Se observa estabilidad general en las curvas de ambas transacciones (/clue y /stats), con mayor dispersión en momentos de saturación de CPU.
- El comportamiento es predecible y controlado, sin picos extremos ni errores de timeout.

Grafica Transacciones por segundo (TPS)

- El gráfico muestra un pico inicial cercano a 2200 TPS, estabilizándose posteriormente entre 900 y 1000 TPS durante la mayor parte de la prueba.
- El rendimiento promedio se mantuvo constante, sin caídas bruscas ni errores.

Visual VM – CPU



- La CPU permaneció al 100% de uso constante durante toda la ejecución, evidenciando que el sistema opera en su máxima capacidad física.
- La actividad del Garbage Collector (GC) fue ligera, sin pausas notables ni afectación al rendimiento.

Conclusión Ejecución #2: Con una configuración de 300 hilos, 100 segundos de ramp-up y 5 minutos de duración, el servicio alcanzó un rendimiento promedio de 986 TPS combinadas, con P90 = 663 ms, P99 ≈ 1.7 s, y 0% de error.

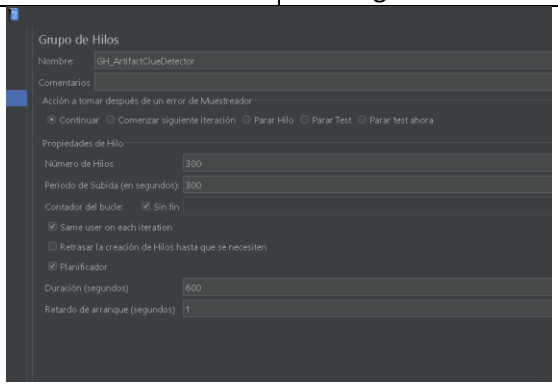
La aplicación mantiene estabilidad y consistencia bajo una carga 3 veces superior a pruebas anteriores, aunque el throughput no aumenta debido a la saturación completa de CPU.

Ejecución #3 – Alcanzar 10.000 TPS:

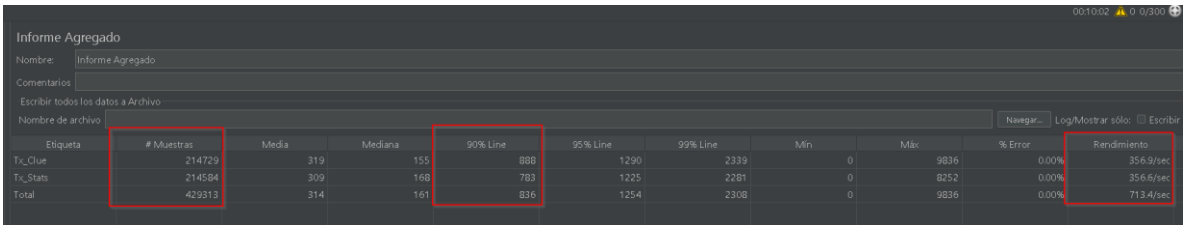
- Se aumenta el Ramp Up a 5 minutos
- Se aumenta el tiempo de la prueba a 10 minutos.

Se define la siguiente configuración:

Threads	300
Rampup	300 Segundos = 5 Minutos
Duration	600 Segundos = 10 Minutos

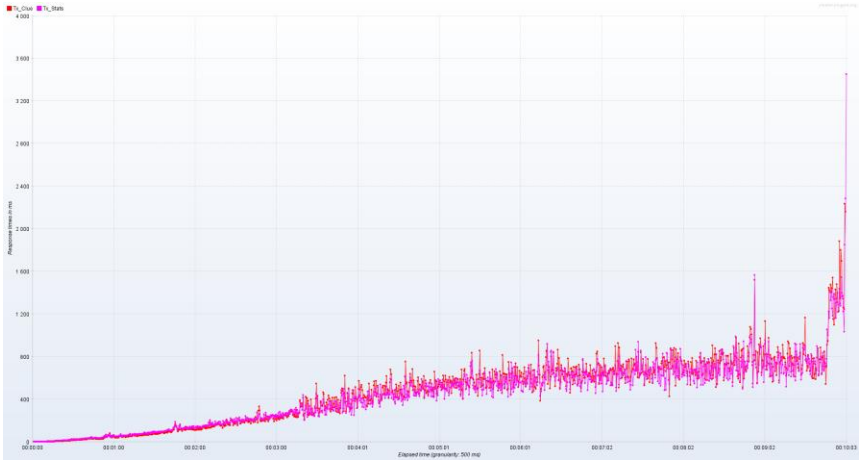


Resultados Obtenidos



- Durante la ejecución extendida con 300 hilos y ramp-up de 5 minutos, el sistema procesó ~713 transacciones por segundo combinadas con 0% de error, manteniendo tiempos de respuesta medios en torno a 300 ms.
- A diferencia de la prueba anterior, el aumento del ramp-up y la duración permitió una estabilización más gradual de la carga, reduciendo picos abruptos al inicio, aunque el rendimiento total se mantuvo limitado por el uso completo de CPU (100%).

Grafica Tiempos de Respuesta:

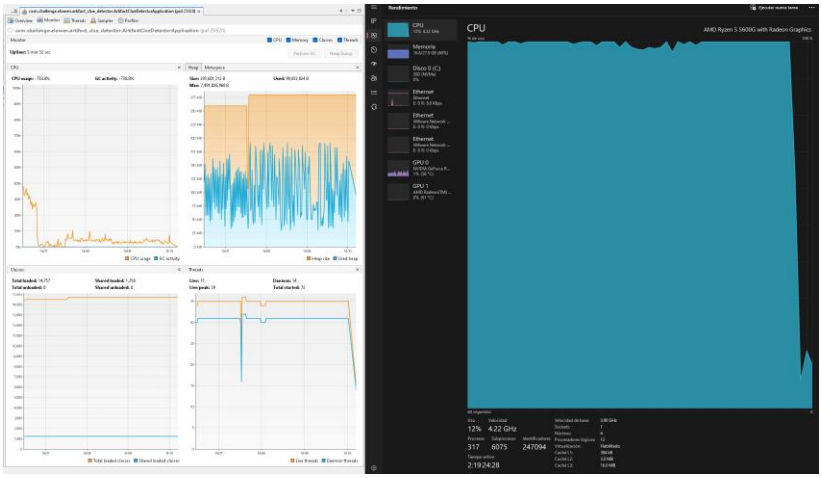


- Los tiempos de respuesta muestran un comportamiento más estable durante los primeros minutos, con una pendiente gradual en el incremento de latencia.
- A partir del minuto 8, se observa un incremento progresivo hasta alcanzar picos cercanos a 3.5 segundos en el percentil más alto.

Transacciones por segundo (TPS)

- El throughput inicial alcanzó picos de 1600–1800 TPS, estabilizándose luego entre 600 y 750 TPS durante los 10 minutos.
- La tasa de transacciones exitosa permaneció constante y sin errores.
- La gráfica refleja una curva descendente natural de estabilización, lo que sugiere que el sistema mantuvo un ritmo sostenible bajo carga constante.

Visual VM – CPU



La CPU se mantuvo al 100% de uso constante durante toda la prueba, confirmando nuevamente que el límite de rendimiento está determinado por el procesamiento del servidor.

El Garbage Collector (GC) presentó actividad regular y controlada, sin pausas significativas.

Conclusión Ejecución #3: Con un ramp-up de 5 minutos y duración total de 10 minutos, el sistema mantuvo un rendimiento promedio de 713 TPS combinadas, con P90 = 836 ms, P99 ≈ 2.3 s y 0% de error.

El incremento gradual de carga permitió una transición más estable sin impactos negativos en el rendimiento general, aunque la saturación de CPU persiste como el principal factor limitante.

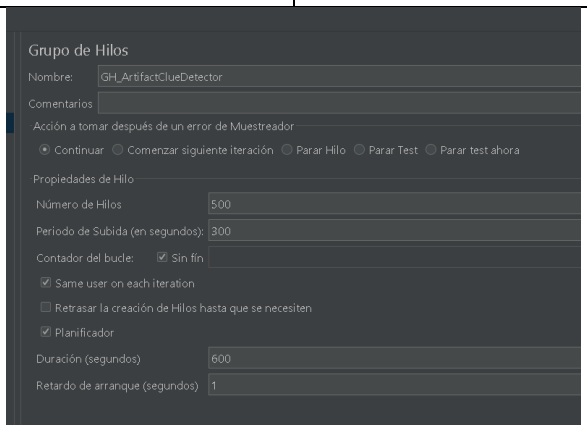
La aplicación se mantiene estable, sin errores, con buen manejo de memoria y concurrencia, pero requiere mayor capacidad de procesamiento o escalamiento horizontal para superar el umbral de 700–1000 TPS.

Ejecución #4 – Alcanzar 10.000 TPS:

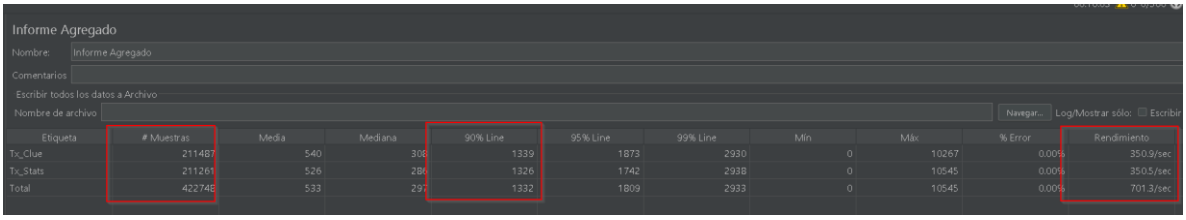
- Se aumenta la cantidad de hilos a 500

Se define la siguiente configuración:

Threads	500
Rampup	300
Duration	600

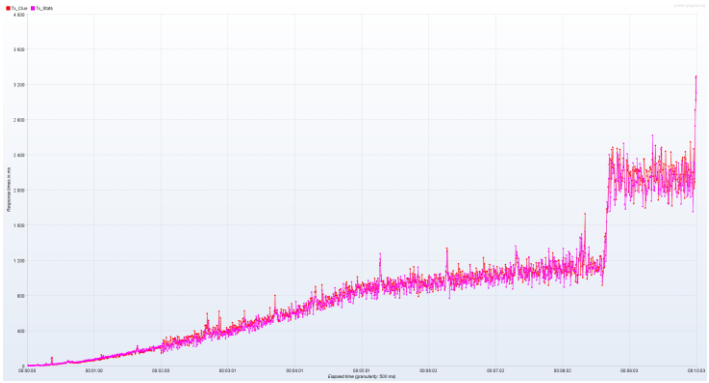


Resultados Obtenidos



- Al incrementar la carga concurrente a 500 hilos, el sistema alcanzó un rendimiento total promedio de ~701 transacciones por segundo, con 0% de errores y una latencia media cercana a los 530 ms.
- Sin embargo, los tiempos de respuesta se incrementaron notablemente en los percentiles altos (P90 ≈ 1.3 s, P99 ≈ 2.9 s), evidenciando que el sistema ha llegado al punto de saturación de CPU.

Grafica Tiempos de Respuesta:

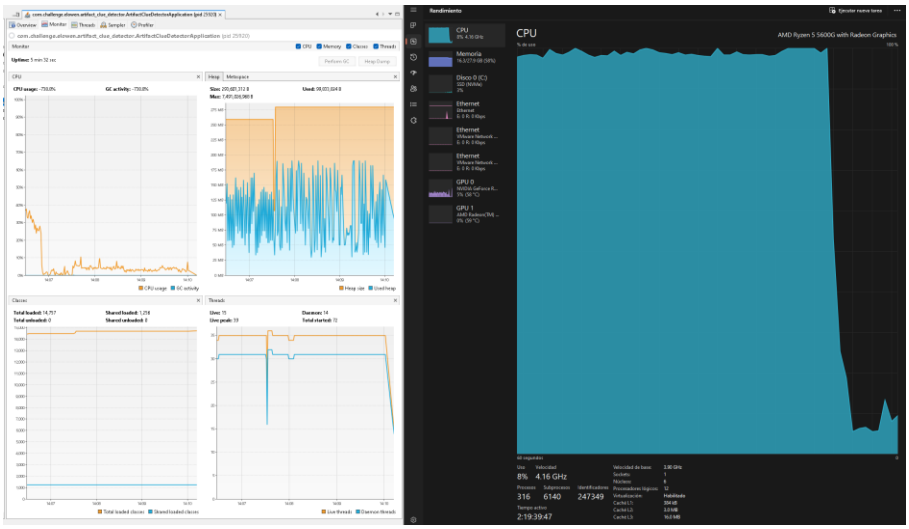


- Las latencias muestran un incremento sostenido a lo largo del tiempo, con valores que comienzan estables durante los primeros minutos y luego aumentan gradualmente.
- A partir del minuto 8, las respuestas superan el segundo de demora en más del 10% de las transacciones, y se observan picos aislados de hasta 3.5 segundos.
- Este comportamiento indica una sobrecarga progresiva, coherente con la cantidad de hilos activos y el límite físico de procesamiento.

Grafica Transacciones por segundo (TPS)

- El throughput alcanzó picos iniciales de 1800 TPS, estabilizándose luego entre 600 y 750 TPS.
- El comportamiento descendente y luego estable es consistente con las pruebas anteriores: el sistema responde correctamente pero se mantiene limitado por la CPU.
- Ambas transacciones (/clue y /stats) muestran curvas prácticamente idénticas, lo que refleja balance de carga equitativo y estabilidad en la ejecución concurrente.

Visual VM – CPU



- El uso de CPU permaneció en 100% constante durante toda la ejecución.
- La memoria Heap se mantuvo estable (~100 MB usados), sin evidencia de fugas de memoria.
- La actividad de GC fue estable y sin pausas perceptibles.

Conclusión Ejecución #4:

Con 500 hilos concurrentes, un ramp-up de 5 minutos y duración total de 10 minutos, el sistema procesó en promedio 701 transacciones por segundo, manteniendo 0% de errores.

Los tiempos de respuesta aumentaron en todos los percentiles, con P90 = 1332 ms y P99 ≈ 2.9 s, mostrando una tendencia clara hacia la saturación del CPU.

El comportamiento es consistente con un entorno que ha alcanzado su límite de capacidad física: la aplicación sigue estable, pero no puede escalar más en un solo nodo local.

Este límite **no se debe al servicio en sí**, sino al entorno local:

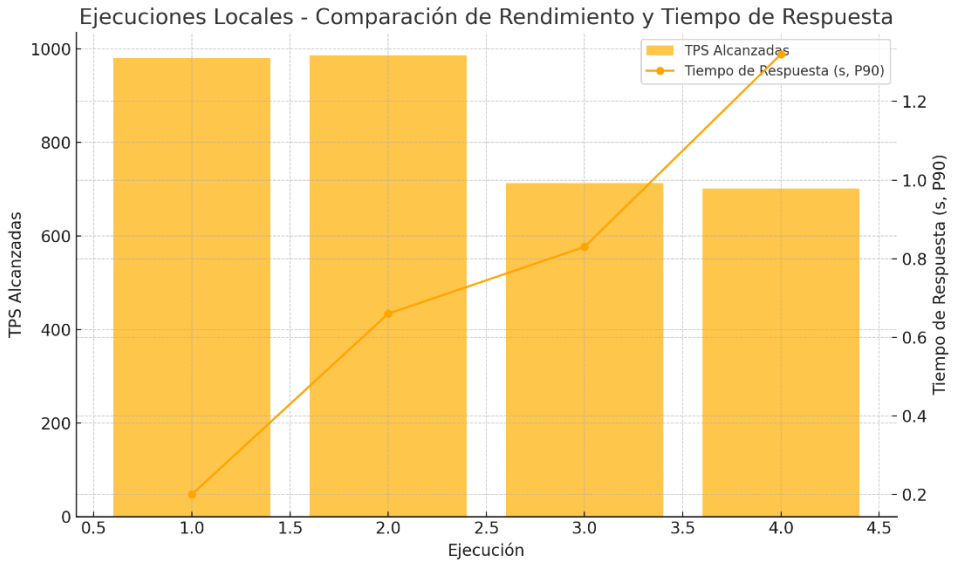
- CPU compartida entre JMeter (client) y Spring Boot (server).
- Loopback network (localhost) → no refleja latencia real.
- Limitaciones de hilos del sistema operativo y de la JVM.

Conclusiones Generales

1. En todas las ejecuciones se observó 0% de errores, lo cual indica que la aplicación mantiene una correcta gestión de concurrencia y estabilidad funcional bajo carga.
2. El sistema Artifact Clue Detector demuestra estabilidad, confiabilidad y buena gestión de memoria, incluso bajo alta concurrencia.
3. No obstante, su rendimiento en entorno local está limitado por la capacidad de CPU, alcanzando su punto máximo de throughput (~700–1000 TPS) sin poder escalar más con el hardware actual.
4. La aplicación mantiene una latencia controlada (P90 ≈ 800–1300 ms) y sin errores, lo cual indica una arquitectura sólida.

Resumen de ejecuciones:

Ejecución	Hilos	RampUp (seg)	Duration (seg)	TPS Alcanzadas	Tiempo de Respuesta (seg) (90 Percentile)
1	100	60	300	980	0,2
2	300	100	300	986	0,66
3	300	300	600	713	0,83
4	500	300	600	701	1,32



- Las TPS alcanzadas (barras) se mantienen estables y altas en las dos primeras ejecuciones (≈980), pero caen de forma notable en las pruebas más largas o con mayor carga (≈700 TPS).
- El tiempo de respuesta (línea naranja) crece de forma progresiva con el aumento de hilos y duración, pasando de 0.2 s a 1.32 s en el percentil 90.
- Esto evidencia que el sistema mantiene buen rendimiento inicial, pero su latencia aumenta proporcionalmente a la concurrencia, lo que marca el límite de su capacidad en pruebas locales.

Entorno: Cloud (Render)
Transacciones:

- Tx_Clue → POST /clue/
- Tx_Stats → GET /stats

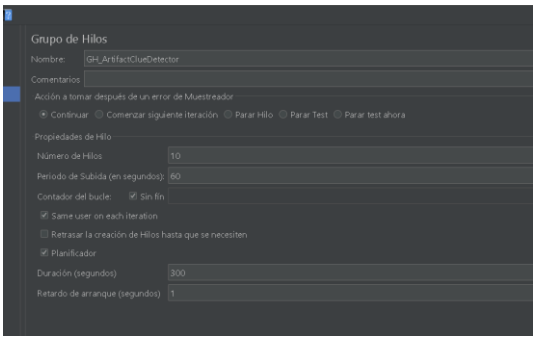
Prueba de línea base: Es la primera prueba que se realiza y se considera fundamental. Consiste en aplicar una carga inicial del 10% del total previsto para verificar que el sistema funcione adecuadamente y cumpla con los requisitos básicos.

Objetivo: Asegurarse de que el sistema pueda manejar una carga mínima y cumplir con los requisitos no funcionales como el tiempo de respuesta. Si la prueba de línea base falla, no se puede avanzar a las pruebas siguientes.

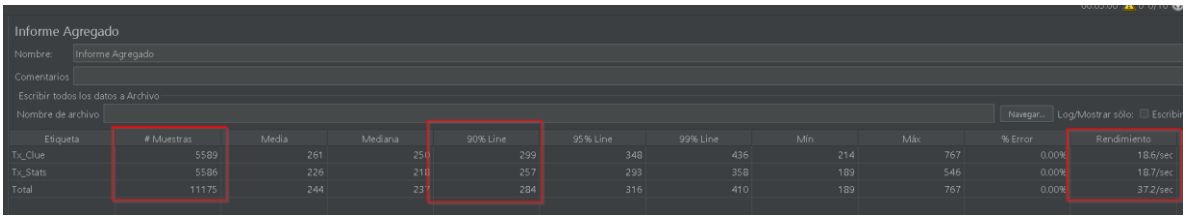
Resultado Real:

Se define la siguiente configuración en Jmeter:

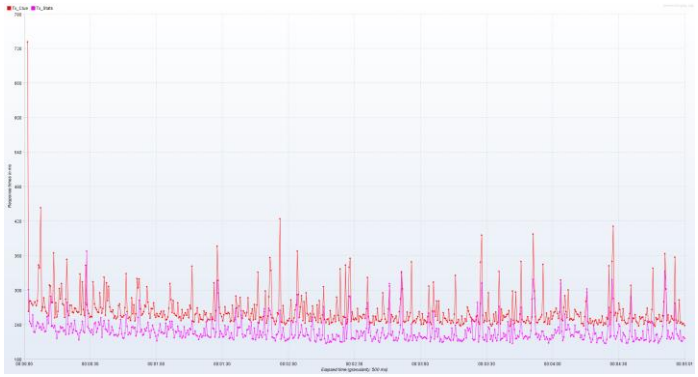
Threads	10
Rampup	60 segundos = 1 Minuto
Duration	300 segundos = 5 Minutos



Resultados Obtenidos

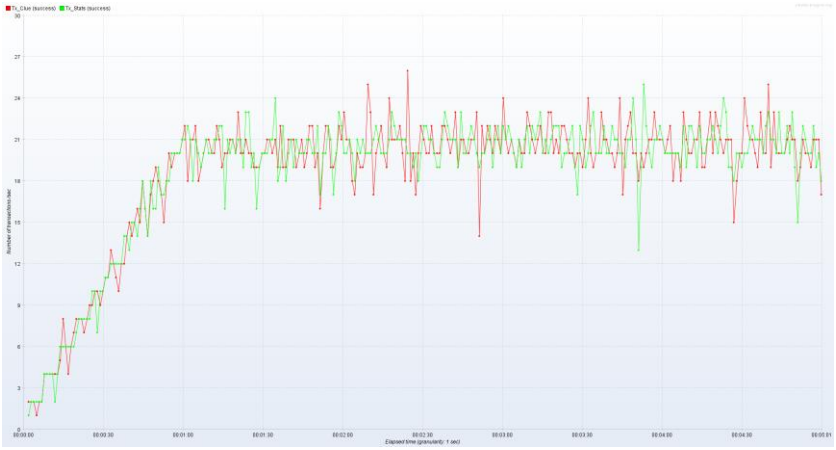


Grafica Tiempos de Respuesta:



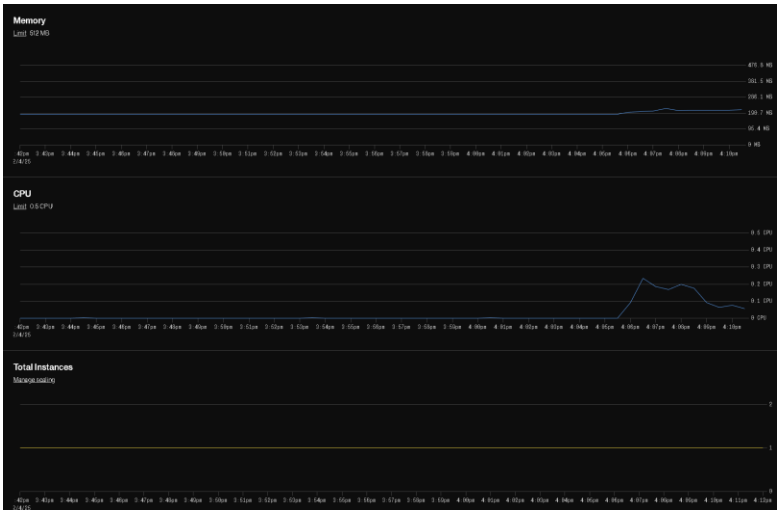
- En comparación con las ejecuciones locales, Render presenta una latencia media más alta, pero más estable, sin degradación con el tiempo.
- Esto indica un ambiente más regulado y limitado por recursos, donde la estabilidad está garantizada, aunque la capacidad máxima sea baja.

Grafica Transacciones por segundo (TPS)



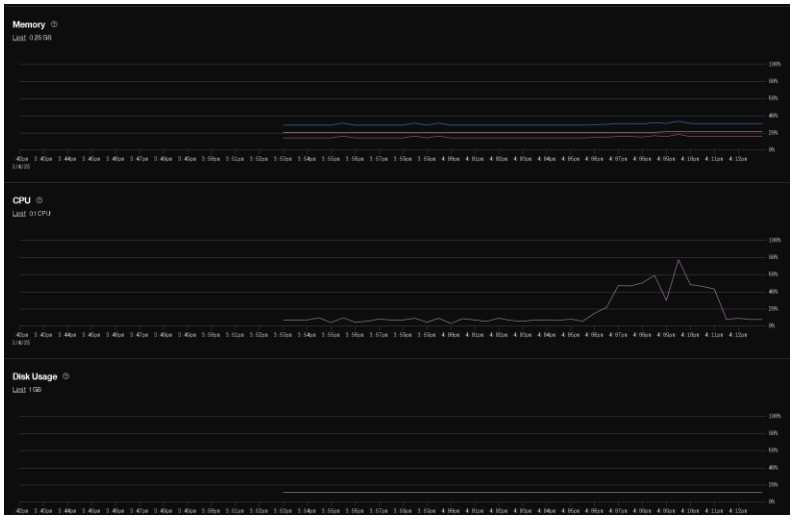
Render mantiene un throughput constante gracias a su gestión de recursos compartidos.

Métricas Servicio (CPU, MEMORIA, Instancias totales)



- La aplicación mantuvo estabilidad y no superó el umbral crítico de CPU ni de memoria, lo que indica que el dimensionamiento actual es adecuado para cargas ligeras.
- No se detectaron reinicios ni escalamiento automático, lo cual valida la estabilidad del servicio.

Métricas Base de Datos (CPU, MEMORIA, Uso de disco)



- La base de datos trabaja con un consumo de memoria muy controlado y sin necesidad de alcanzar su límite.
- La CPU responde dinámicamente al aumento de carga, procesando múltiples consultas concurrentes sin llegar al 100%, lo que demuestra que el motor de base de datos tiene suficiente capacidad para manejar la concurrencia actual sin saturarse.
- El hecho de que la CPU suba al 80% y luego vuelva a bajar indica una correcta liberación de recursos y ausencia de bloqueos o procesos pendientes.

Prueba de Carga: Esta prueba se enfoca en verificar que el sistema pueda manejar la carga esperada en condiciones normales de operación.

Objetivo: Confirmar que el sistema puede soportar el número esperado de transacciones por segundo y cumplir con los tiempos de respuesta establecidos bajo condiciones normales.

Resultado Teórico – Alcanzar 100 TPS

PLANTILLA CALCULO CONCURRENCIA

Variable	Descripción	Valores de Ejemplo
Nte	Numero de Transacciones	100
Md	Unidad de tiempo del Nte	1
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Ttrx	**Total Transacciones Esperadas en un periodo Te**	30,000
Tps	**Transacciones por Segundo**	100

Variable	Descripción	Valores de Ejemplo
Ttrx	Total Transacciones Esperadas en un periodo Te	30,000
Top	Tiempo de duración promedio de una (1) operación (Segundos)	0,002
Te	Tiempo de Ejecución de la prueba (Segundos)	300
O	Numero de operaciones en un período Md realizadas por un usuario	1
Uc	**Usuarios Concurrentes**	1

	USUARIOS CONCURRENTES	TOTAL TRANSACCIONES
FORMULAS	$Uc = \frac{Top * Ttrx * O}{Te}$	$Ttrx = \frac{Nte * Te}{Md * O}$

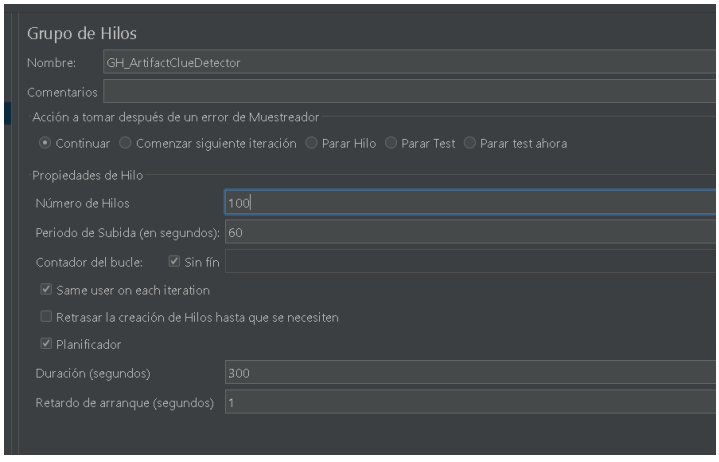
Variable	Descripción
Nte	Transacciones esperadas por el clientes
Md	Unidad de tiempo dada para las transaccione esperadas por el clientes (Segundo, Minuto, Hora, Dia)
Te	Tiempo de duración de la prueba de performance
Ttrx	Transacciones totales que se espera generar durante el escenario de prueba
Top	Tiempo de duración promedio de una (1) operación (Segundos)
Uc	Usuarios concurrentes

Guía para política de carga			
PRUEBA	Línea Base	Carga	Estrés (125%)
Uc	0	1	1
Tps	10	100	125

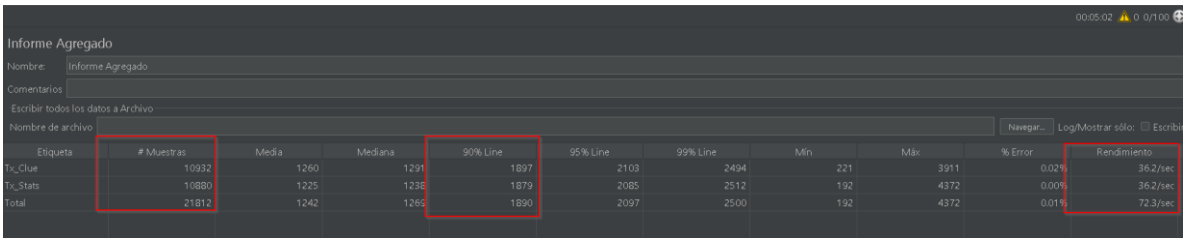
Ejecución #1 – Alcanzar 100 TPS:

Se define la siguiente configuración:

Threads	100
Rampup	60 segundos = 1 Minuto
Duration	300 segundos = 5 Minutos

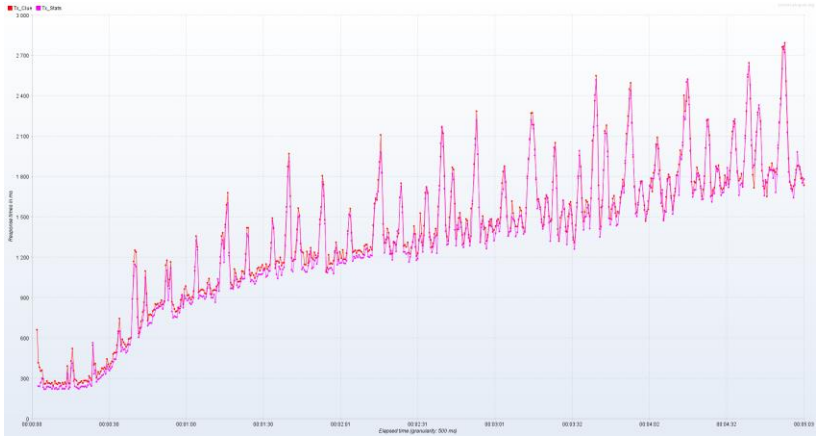


Resultados Obtenidos



- El rendimiento global alcanzó 72 transacciones por segundo (TPS) con tiempos medios de respuesta cercanos a 1.2 segundos, manteniendo errores mínimos (0.01%).
- Comparado con la primera ejecución (10 hilos), el rendimiento total se duplicó (de 37 a 72 TPS), aunque con un incremento esperado en latencia (de 250 ms a 1,200 ms promedio).

Grafica Tiempos de Respuesta:



- El gráfico muestra un aumento progresivo de latencia conforme los hilos alcanzan el máximo de concurrencia.
- Se observan picos entre 1.8 s y 2.7 s en los percentiles superiores (90%-99%).

Grafica Transacciones por segundo (TPS):

- El throughput aumenta rápidamente durante los primeros 30 segundos, alcanzando picos cercanos a 65–70 transacciones por segundo.
- Luego se estabiliza con oscilaciones menores hasta el final de la prueba.

Métricas Servicio (CPU, MEMORIA, Instancias totales)

- Se mantuvo entre **350–470 MB**, sin alcanzar el límite de 512 MB.
- Picos de uso entre **0.1 y 0.2 CPU**.

Métricas Base de Datos (CPU, MEMORIA, Uso de disco)

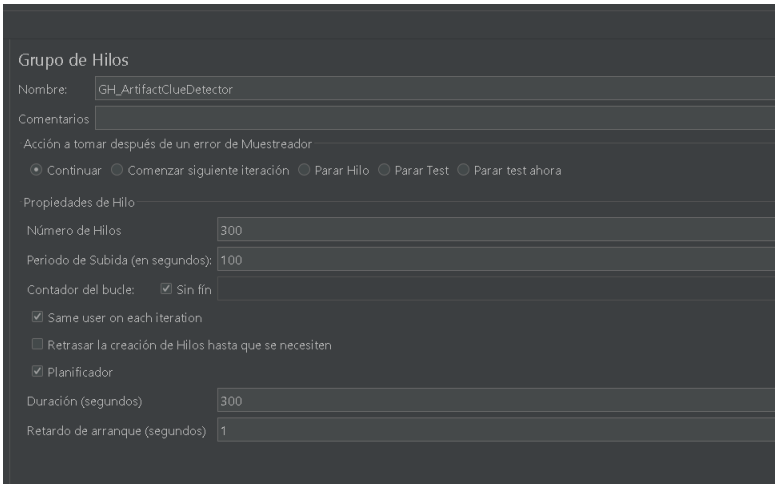
- Memoria promedio de uso entre 30–40%
- Comportamiento de memoria estable y sin crecimiento sostenido.
- Incremento de CPU durante el pico de carga, alcanzando 90–100% de utilización temporalmente.
- Descenso progresivo de CPU después de los 4:30 p.m., una vez reducida la presión de consultas concurrentes.

Conclusión Ejecución #1: Durante la segunda ejecución con 100 hilos concurrentes, el sistema mantuvo alta estabilidad operativa y sin errores, aunque con un aumento notable de latencia.

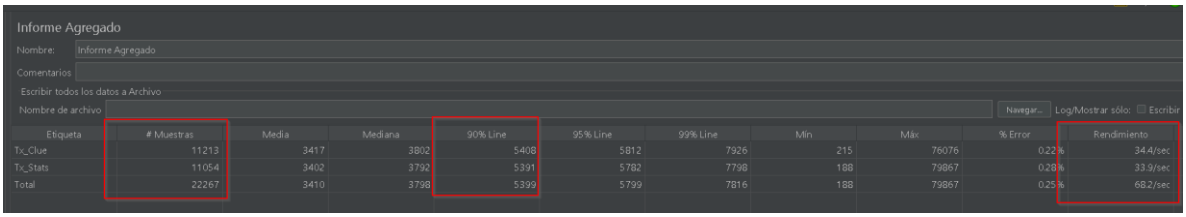
Ejecución #2 – Alcanzar 100 TPS:

Se define la siguiente configuración:

Threads	300
Rampup	100 segundos = 1,6 Minutos
Duration	300 segundos = 5 Minutos

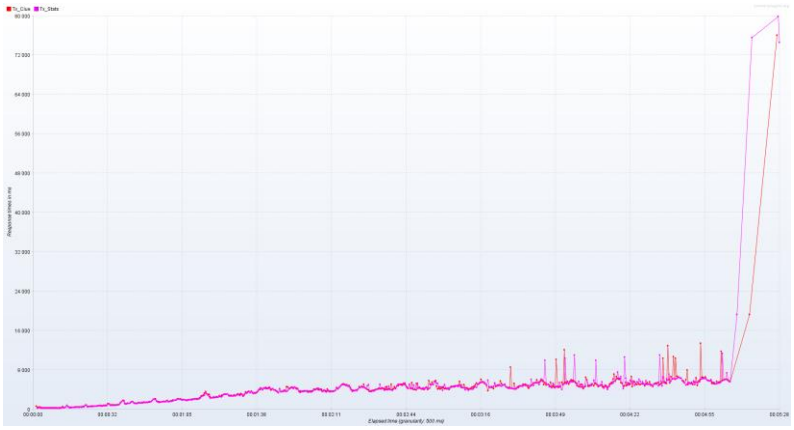


Resultados Obtenidos



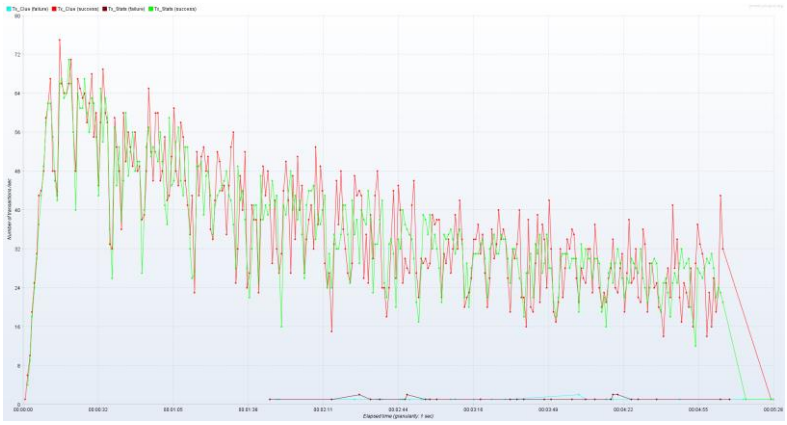
- A pesar del aumento a 300 hilos (3× más carga que en la ejecución anterior), el sistema mantuvo un rendimiento global estable de 68 TPS, solo un 5% menor al obtenido con 100 hilos (72 TPS).
- Sin embargo, los tiempos medios de respuesta aumentaron de 1.2 s a 3.4 s, con picos de hasta 7.9 s en el percentil 99.

Grafica Tiempos de Respuesta:



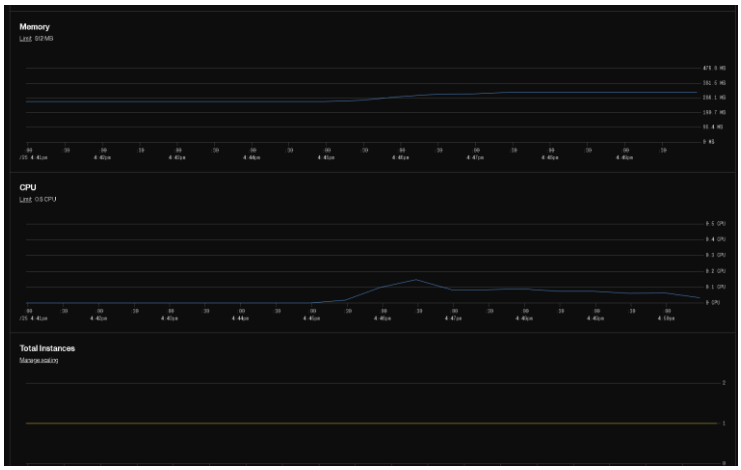
Los tiempos presentan una tendencia ascendente sostenida, con incrementos graduales durante la prueba y un pico al final que supera los 70,000 ms (70 s) en algunos casos extremos.

Grafica Transacciones por segundo (TPS):



Este comportamiento confirma un punto de saturación dinámica: el sistema llega a su límite de procesamiento (~70 TPS) y se autorregula disminuyendo el ritmo de respuesta para evitar fallos.

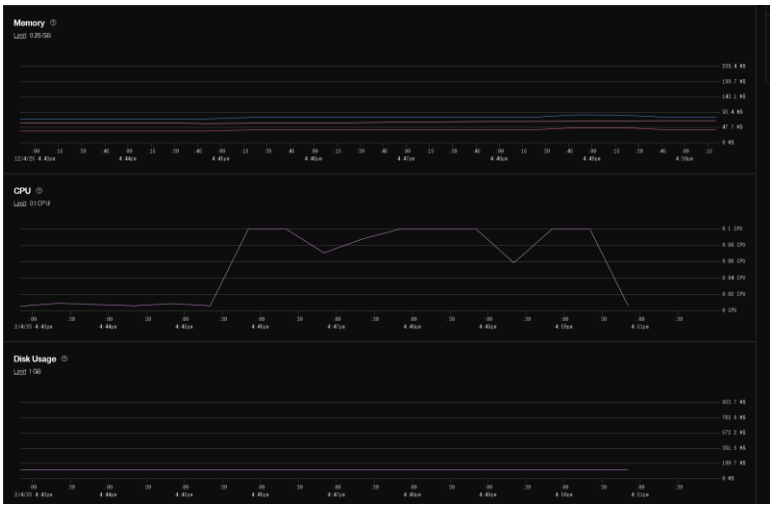
Métricas Servicio (CPU, MEMORIA, Instancias totales)



- El uso de memoria se mantiene **estable entre 280 MB y 380 MB** durante toda la ejecución.

- La CPU inicia en niveles bajos (<0.05 CPU) y muestra un incremento progresivo durante el ramp-up.

Métricas Base de Datos (CPU, MEMORIA, Uso de disco)



- La base de datos fue el componente más exigido del sistema, alcanzando su máximo nivel de procesamiento.
- Esto sugiere que las consultas concurrentes de escritura y lectura (/clue, /stats) saturan la CPU disponible.
- Aunque no hubo fallos ni bloqueos, este comportamiento confirma que la BD es el cuello de botella principal.

Conclusiones Ejecución #2: Durante la prueba con 300 hilos concurrentes, el servicio Render mostró un comportamiento estable y sin fallos, manteniendo uso de memoria eficiente (380–476 MB) y uso de CPU alto pero controlado (~90% del límite).

El sistema operó correctamente sin reinicios, errores o caídas, aunque alcanzó su máximo punto de procesamiento (CPU-bound), lo que explica la aumentada latencia promedio (~3.4 s) observada en los resultados de JMeter.

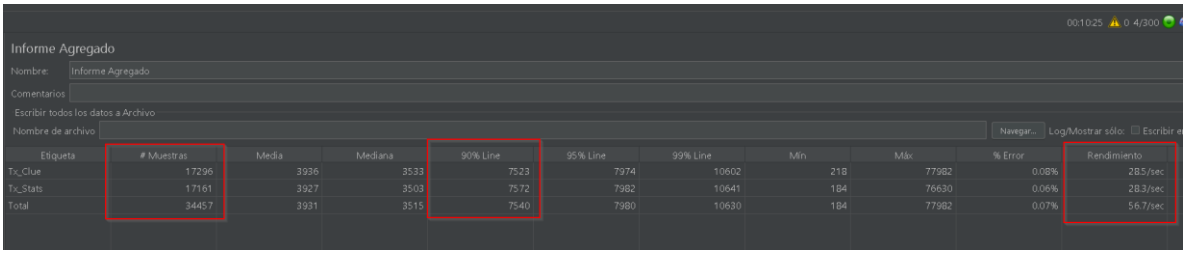
Ejecución #3 – Alcanzar 100 TPS:

- Se aumenta el Ramp Up a 5 minutos
- Se aumenta el tiempo de la prueba a 10 minutos.

Se define la siguiente configuración:

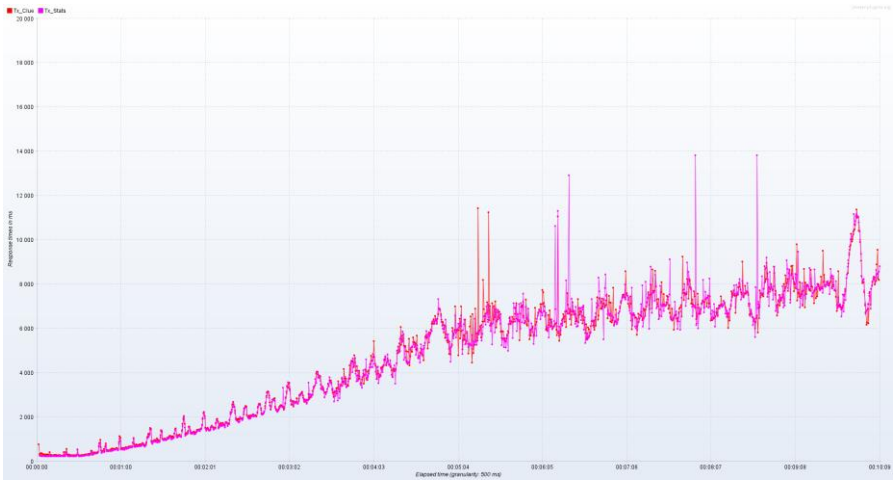
Threads	300
Rampup	300 Segundos = 5 Minutos
Duration	600 Segundos = 10 Minutos

Resultados Obtenidos



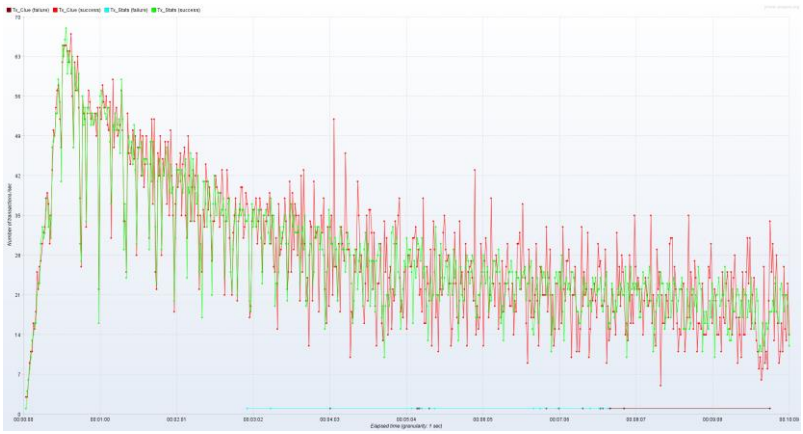
- A pesar de extender la duración de la prueba a 10 minutos y de mantener 300 usuarios concurrentes, el sistema conservó estabilidad y baja tasa de errores (0.07%), lo que demuestra resistencia y robustez del entorno Render.
- El rendimiento global se estabilizó en ≈56 TPS, menor que los 68 TPS observados en la prueba corta, lo cual indica un efecto de fatiga del sistema por uso sostenido de CPU y acumulación de tareas concurrentes.
- Los tiempos de respuesta promedio aumentaron a ≈3.9 s, con picos de hasta 10 s (percentil 99), lo que confirma una saturación controlada de los recursos asignados.

Grafica Tiempos de Respuesta:



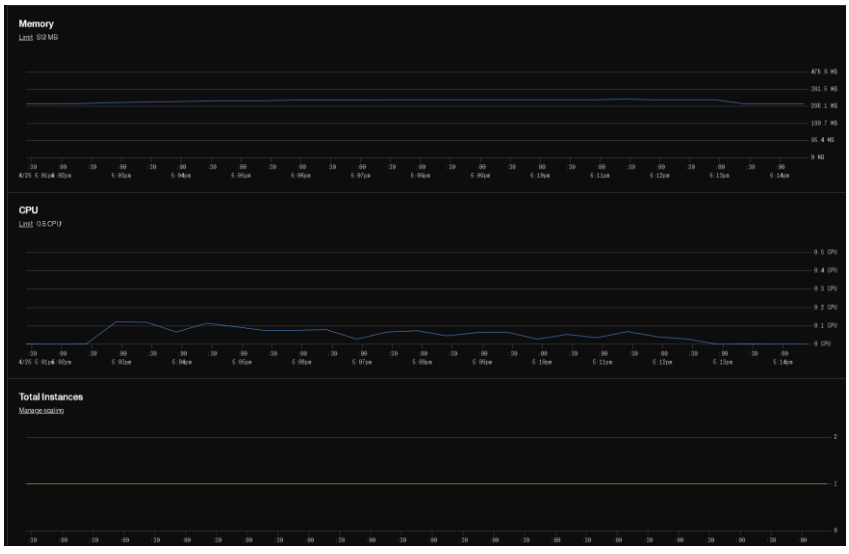
Los tiempos de respuesta muestran una tendencia ascendente gradual, con pequeñas fluctuaciones a lo largo de los 10 minutos

Grafica Transacciones por segundo (TPS):



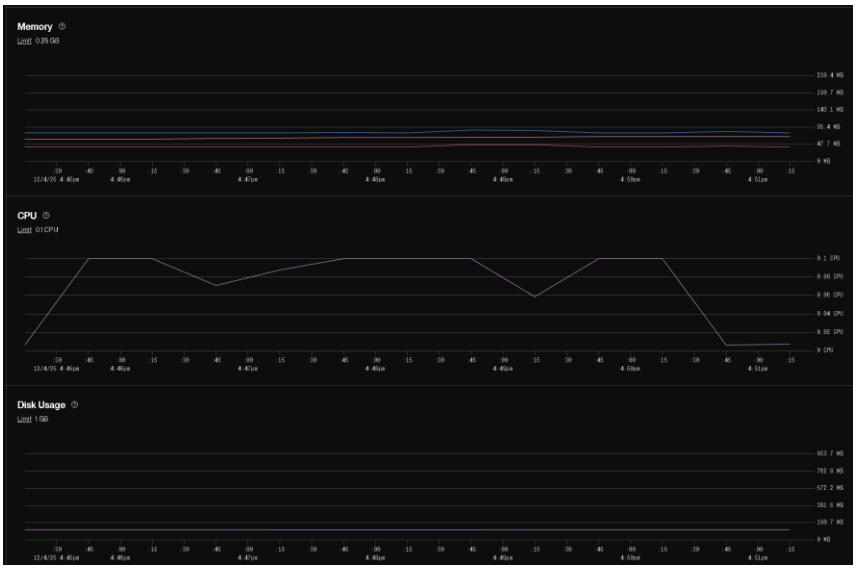
- El sistema alcanza su pico máximo de throughput durante los primeros 2–3 minutos, pero luego entra en una meseta de rendimiento donde mantiene estabilidad sin caídas críticas.
- Esto evidencia una buena gestión de concurrencia, pero también un límite natural de procesamiento debido al plan actual (0.5 CPU).

Métricas Servicio (CPU, MEMORIA, Instancias totales)



- El consumo de RAM fue constante y seguro, mostrando que la aplicación maneja bien la carga concurrente sin degradar el rendimiento.
- La aplicación trabaja de forma eficiente y estable, con suficiente capacidad para procesar la carga sin saturar la CPU.

Métricas Base de Datos (CPU, MEMORIA, Uso de disco)



- La base de datos mantiene un uso constante de memoria dentro del rango esperado; no hay indicios de saturación ni de uso excesivo del caché.
- Se observa que la CPU alcanza el 100% de uso durante la mayor parte de la prueba, lo que demuestra que la base de datos fue el componente más exigido.

Conclusiones Ejecución #3: La base de datos alcanzó su límite de CPU, actuando como el cuello de botella principal, mientras que la aplicación mostró buena estabilidad y uso eficiente de memoria.

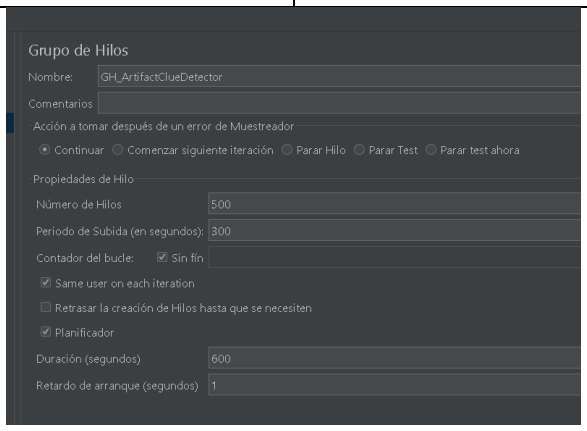
Esto confirma que el entorno es robusto, pero requiere más capacidad de CPU o autoescalado para mejorar tiempos de respuesta bajo alta concurrencia.

Ejecución #4 – Alcanzar 100 TPS:

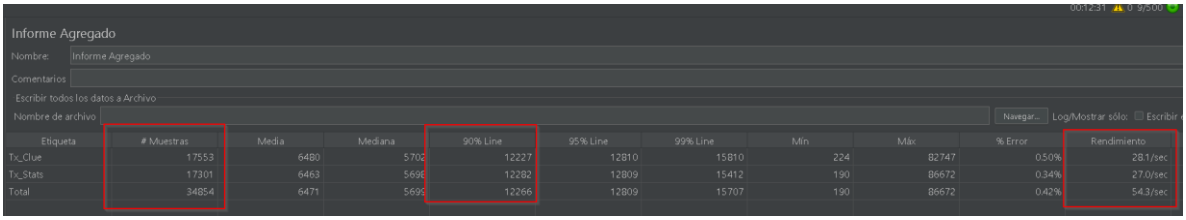
- Se aumenta la cantidad de hilos a 500

Se define la siguiente configuración:

Threads	500
Rampup	300
Duration	600

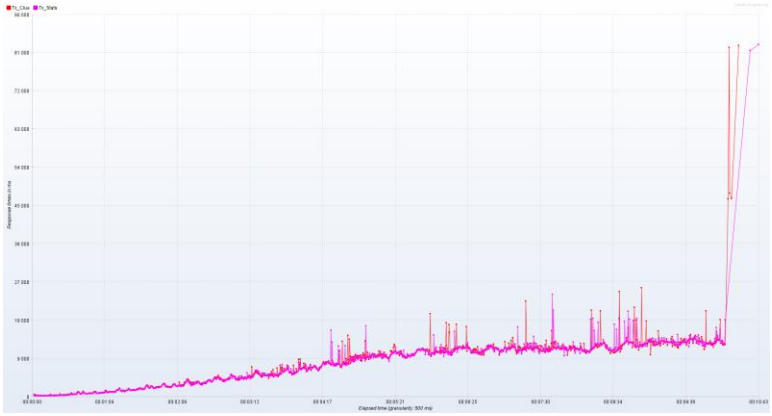


Resultados Obtenidos



- El throughput (TPS) bajó ligeramente de 56.7 a 54.3 transacciones por segundo, a pesar de aumentar un 66% los hilos.
- Esto indica que el sistema alcanzó su capacidad límite de procesamiento concurrente.
- Es decir, aunque puede manejar 500 usuarios, ya no escala linealmente: el incremento de carga no se traduce en más rendimiento.
- Los tiempos de respuesta aumentan de manera casi proporcional al aumento de concurrencia, lo cual es esperado, pero la magnitud del cambio indica saturación en el backend o base de datos.
- Pasar de 3.9 s a 6.5 s en promedio implica que cada usuario experimenta una demora adicional del 65%.

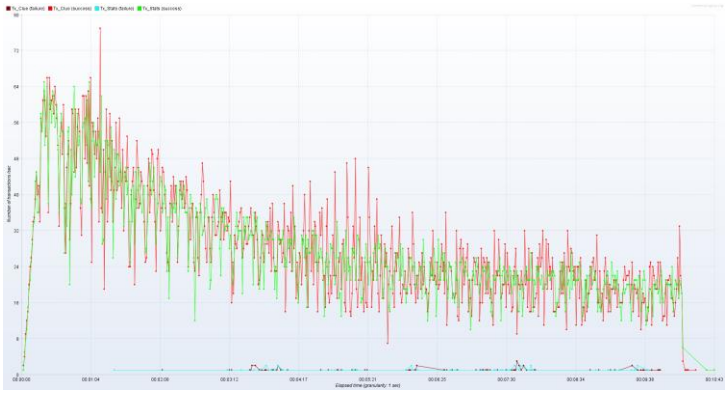
Grafica Tiempos de Respuesta:



El sistema mantuvo estabilidad inicial, pero a medida que los 500 usuarios se ejecutaron simultáneamente, los tiempos de respuesta se incrementaron exponencialmente, indicando saturación del sistema y pérdida de capacidad de procesamiento.

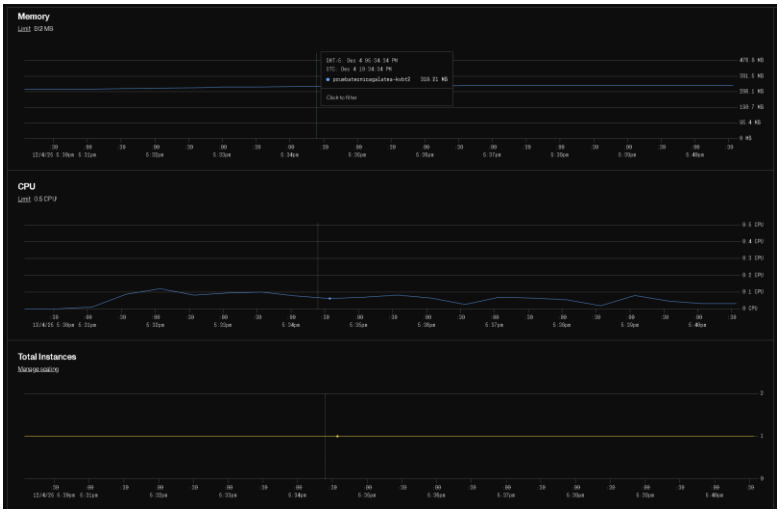
Esto evidencia que el sistema alcanzó su límite máximo de concurrencia sostenible.

Grafica Transacciones por segundo (TPS):



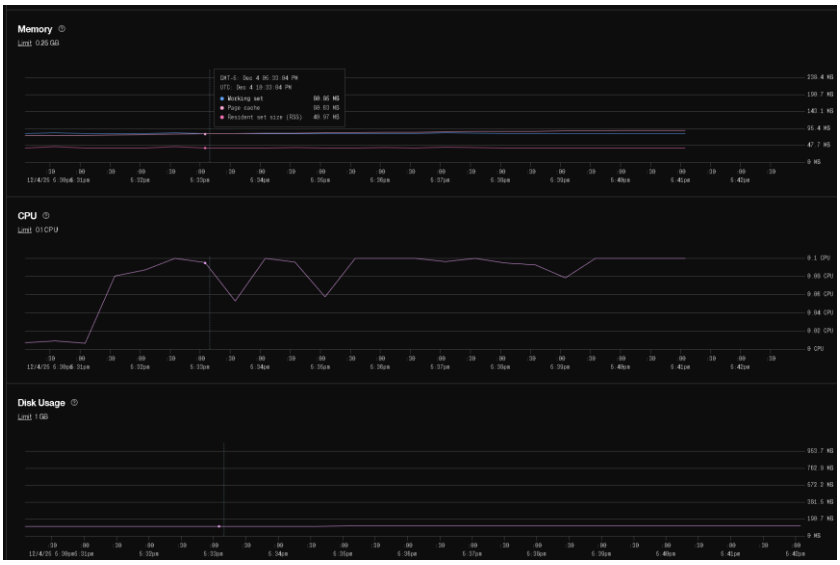
El sistema logra alcanzar un buen nivel de rendimiento al inicio, pero pierde estabilidad y rendimiento sostenido bajo cargas altas prolongadas.

Métricas Servicio (CPU, MEMORIA, Instancias totales)



- El servicio backend no presenta problemas de recursos locales; su consumo de CPU y memoria es estable.
- Sin embargo, al no contar con más instancias, toda la carga de los 500 usuarios concurrentes recae sobre una única instancia, lo cual limita la escalabilidad.
- La aplicación no es la causa del cuello de botella, pero podría mejorar su rendimiento con escalado horizontal o balanceo de carga.

Métricas Base de Datos (CPU, MEMORIA, Uso de disco)



- La base de datos opera constantemente al 100% de la capacidad de CPU asignada, lo que la convierte en el cuello de botella principal del sistema.
- Aunque el consumo de memoria y disco es estable, la limitación de CPU restringe el tiempo de respuesta general de las transacciones.
- Es necesario incrementar la capacidad de CPU o aplicar optimizaciones de consulta para mejorar el rendimiento.

Conclusiones Ejecución #4:

Durante la prueba con 500 usuarios concurrentes y duración de 14 minutos, el sistema mantuvo estabilidad operativa, pero presentó una degradación significativa en los tiempos de respuesta y el throughput conforme se incrementó la carga.

El análisis de recursos indica que el componente más afectado fue la base de datos, cuya CPU trabajó al límite durante toda la ejecución, provocando demoras generalizadas.

Conclusiones generales

- El sistema demostró ser estable, pero no completamente escalable bajo cargas concurrentes altas.
- La degradación observada se debe principalmente a limitaciones en la capacidad de la base de datos (CPU saturada) y a la ausencia de escalado horizontal en la capa de aplicación.
- A 300 hilos, el rendimiento es aceptable y consistente.
- A 500 hilos, el sistema sigue siendo funcional, pero los tiempos de respuesta aumentan entre un 60 % y 70 %, comprometiendo la experiencia de usuario bajo alta concurrencia.
- En términos generales, el sistema ha alcanzado su límite operativo estable con la configuración actual.

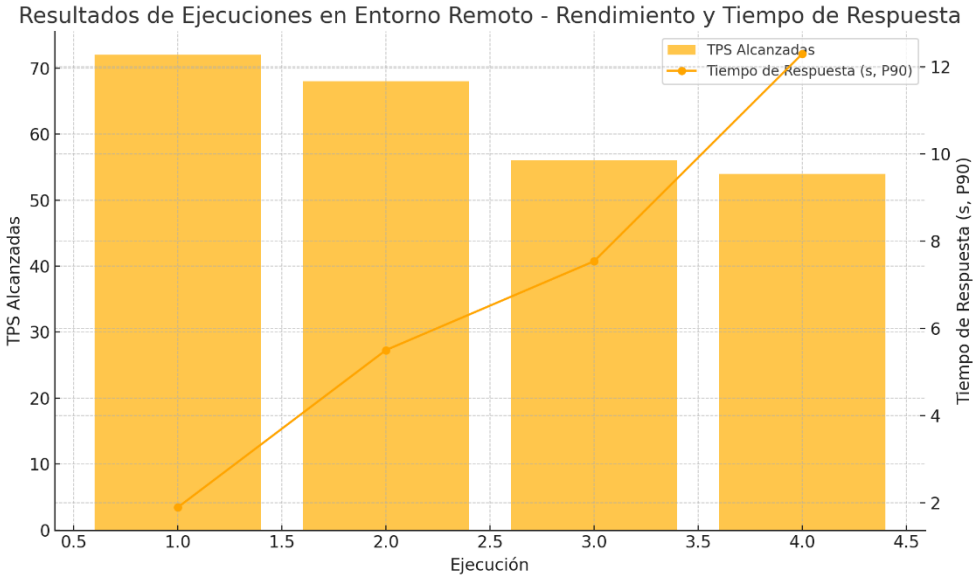
Recomendaciones Técnicas

- Aumentar la capacidad de CPU asignada
- Implementar auto escalado horizontal en la aplicación backend, de modo que se generen nuevas instancias cuando la carga exceda el 70 % de CPU.

- Evaluar la incorporación de un balanceador de carga que distribuya el tráfico entre instancias.

Resumen de ejecuciones:

Ejecución	Hilos	RampUp (seg)	Duration (seg)	TPS Alcanzadas	Tiempo de Respuesta (seg) (90 Percentile
1	100	60	300	72	1.89
2	300	100	300	68	5.5
3	300	300	600	56	7.54
4	500	300	600	54	12.3



- El rendimiento (TPS) disminuye progresivamente conforme aumenta la carga (de 72 TPS con 100 hilos a 54 TPS con 500 hilos).
- En contraposición, el tiempo de respuesta (línea naranja) crece de forma exponencial: pasa de 1.89 s a 12.3 s, lo que evidencia saturación del sistema ante mayores niveles de concurrencia.
- La tendencia indica que el entorno no escala linealmente y presenta limitaciones de procesamiento cuando se superan los 300 hilos, con un aumento notable de la latencia.