

# **INFORME DE ANÁLISIS DE RESULTADOS PRUEBA DE CARGA**

**PRUEBA DE PERFORMANCE – IMPLEMENTACIÓN Y ANÁLISIS DE  
RESULTADOS DOCUMENTADOS**

ELABORADO:

**ING.JORDY VINCIO CHILA AVILEZ**  
jordychila21@gmail.com

## **EVALUACIÓN TÉCNICA DE DESEMPEÑO Y RENDIMIENTO**

FECHA:

**ENERO 07 DEL 2026**

MARCO DE REFERENCIA: ISO/IEC/IEEE 29119

**Prueba API**

link del repositorio: <https://github.com/Jordy-Chila/prueba-carga-login-k6>

## **Contenido**

1.	Introducción .....	3
2.	Resumen del escenario ejecutado .....	3
3.	Análisis de resultados.....	3
3.1.	Tiempo de respuesta.....	3
3.2.	Errores .....	4
4.	Ánálisis del monitoreo (VUs vs Requests/s) .....	4
5.	Conclusiones.....	4
6.	Recomendaciones .....	4

## INFORME DE ANÁLISIS DE RESULTADOS – PRUEBA DE CARGA

### 1. Introducción

El presente informe corresponde al análisis de los resultados obtenidos en una prueba de carga realizada sobre el servicio de autenticación (login). El objetivo fue evaluar el comportamiento del sistema bajo concurrencia, considerando métricas de rendimiento, estabilidad y tasa de errores.

### 2. Resumen del escenario ejecutado

Durante la ejecución de la prueba se alcanzó un máximo de 140 usuarios virtuales concurrentes (VUs), generando un throughput promedio aproximado de 73 peticiones por segundo (TPS), con picos cercanos a 80 TPS. El escenario ejecutado superó el objetivo mínimo de 20 TPS definido inicialmente.

### 3. Análisis de resultados

```
textSummary.txt
1   x App Transaction Balance response was OK
2   ↳ 97% - ✓ 269891 / ✗ 6759
3
4   █ setup
5
6   ✓ checks.....: 97.55% ✓ 269891    x 6759
7   data_received...: 842 MB 223 kB/s
8   data_sent.....: 588 MB 156 kB/s
9   failed_request.: 2.44% ✓ 6759      x 269891
10  http_req_blocked.: avg=10.97µs min=0s med=0s max=35.02ms p(90)=0s p(95)=0s
11  http_req_connecting.: avg=3.3µs min=0s med=0s max=11.82ms p(90)=0s p(95)=0s
12  ✓ http_req_duration.: avg=861.68ms min=191.86ms med=613.42ms max=29.93s p(90)=1.28s p(95)=1.57s
13  ✓ { expected_response:true }.: avg=735.84ms min=244.92ms med=600.7ms max=26.72s p(90)=1.22s p(95)=1.42s
14  ✓ http_req_failed.: 2.44% ✓ 6759      x 269891
15  http_req_receiving.: avg=424.03µs min=0s med=320.7µs max=39.58ms p(90)=988.4µs p(95)=1.05ms
16  http_req_sending...: avg=43.22µs min=0s med=0s max=31.25ms p(90)=0s p(95)=517µs
17  http_req_tls_handshaking...: avg=7.36µs min=0s med=0s max=27.02ms p(90)=0s p(95)=0s
18  http_req_waiting...: avg=861.21ms min=191.86ms med=613.01ms max=29.93s p(90)=1.28s p(95)=1.57s
19  ✓ http_reqs.....: 276650 73.176857/s
20  | iteration_duration.: avg=1.86s min=0s med=1.61s max=30.94s p(90)=2.29s p(95)=2.57s
21  ✓ iterations.....: 276650 73.176857/s
22  vus.....: 2 min=2 max=140
23  vus_max.....: 140 min=140 max=140
24  y_failed_request_stage_0_HTTP5xx...: 1 0.000265/s
25  y_failed_request_stage_1_HTTP4xx...: 769 0.203409/s
26  y_failed_request_stage_1_HTTP5xx...: 5987 1.583625/s
27  y_failed_request_stage_2_HTTP5xx...: 2 0.000529/s
```

#### 3.1. Tiempo de respuesta

- Tiempo promedio de respuesta: ~861 ms
- Percentil 90 (p90): ~1.28 s
- Percentil 95 (p95): ~1.57 s

La mayoría de las solicitudes presenta tiempos de respuesta dentro del SLA de 1.5 segundos. No obstante, el percentil 95 supera ligeramente dicho umbral, lo que indica que bajo alta carga algunos usuarios experimentan mayores tiempos de respuesta.

### 3.2. Errores

- Tasa de error total: ~2.44%
- Predominio de errores HTTP 5xx

La presencia de errores 5xx sugiere fallos del lado servidor cuando el sistema opera cerca de su límite de capacidad.

## 4. Análisis del monitoreo (VUs vs Requests/s)



El gráfico evidencia una relación directa entre el incremento de usuarios virtuales y el throughput generado. Se observan caídas temporales en los VUs y en el número de peticiones por segundo, lo que indica episodios de inestabilidad asociados a saturación de recursos o timeouts del backend.

## 5. Conclusiones

- El sistema cumple el objetivo de throughput definido para la prueba.
- El tiempo de respuesta no se mantiene de forma consistente dentro del SLA bajo alta carga.
- Se evidencian problemas de estabilidad y capacidad del backend bajo carga sostenida.

## 6. Recomendaciones

- Optimizar el procesamiento del endpoint de login.
- Revisar la gestión de conexiones y recursos del servidor.
- Implementar monitoreo y alertas sobre errores HTTP 5xx.
- Ejecutar pruebas de estrés y pruebas de larga duración para identificar el límite real del sistema.