

En este reporte se documentan las pruebas de carga realizadas con la herramienta Siege sobre un entorno local. El objetivo fue evaluar el comportamiento de la aplicación ante distintas cantidades de usuarios concurrentes, duraciones y patrones de acceso.

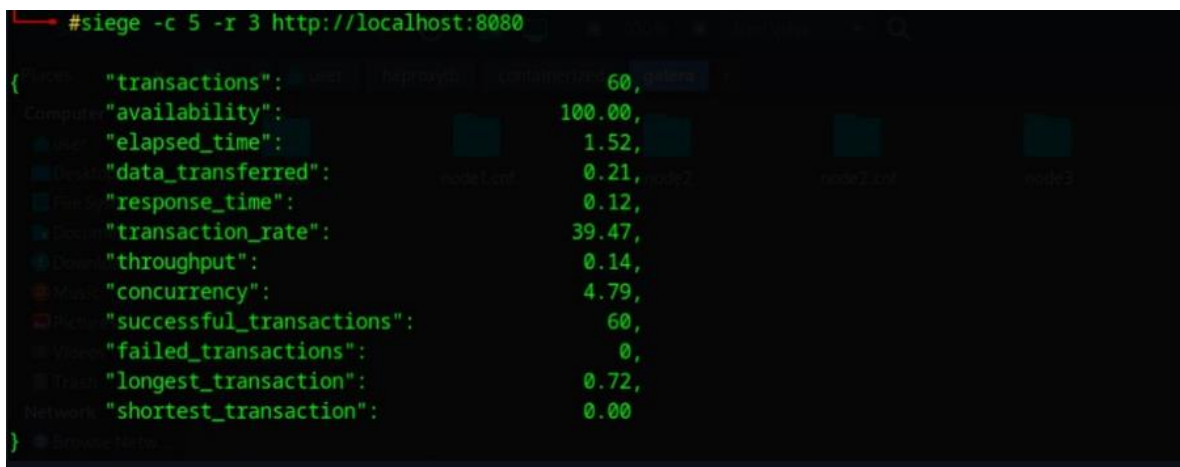
Las pruebas se hicieron antes y después de aplicar mejoras, para observar posibles diferencias en rendimiento. A lo largo del reporte se analizan métricas clave como transacciones, tiempos de respuesta, throughput y tasa de errores.

## Herramienta Utilizada

Siege: Utilizada para realizar pruebas de carga HTTP.

Entorno: localhost:8080 (aplicación corriendo en contenedor Docker)

Sistema Operativo: Parrot OS.



```
#siege -c 5 -r 3 http://localhost:8080
{
  "transactions": 60,
  "availability": 100.00,
  "elapsed_time": 1.52,
  "data_transferred": 0.21,
  "response_time": 0.12,
  "transaction_rate": 39.47,
  "throughput": 0.14,
  "concurrency": 4.79,
  "successful_transactions": 60,
  "failed_transactions": 0,
  "longest_transaction": 0.72,
  "shortest_transaction": 0.00
}
```

### 1. Prueba rápida básica:

siege -c 5 -r 3 http://localhost:8080

Esta prueba ejecutó 3 peticiones por cada uno de los 5 usuarios concurrentes, totalizando 60 solicitudes. Es una prueba muy básica, útil para validar que el servidor responde correctamente y de forma rápida.

Resultados:

- Transacciones: 60 exitosas
- Tiempo total: 1.52 s
- Tiempo de respuesta promedio: 0.12 s
- Disponibilidad del 100%

```
[root@parrot]~[/home/user/naproxy1b/containerized]
#siege -c 10 -t 1m http://localhost:8080

{
  "transactions": 2337,
  "availability": 100.00,
  "elapsed_time": 59.13,
  "data_transferred": 8.29,
  "response_time": 0.25,
  "transaction_rate": 39.52,
  "throughput": 0.14,
  "concurrency": 9.85,
  "successful_transactions": 2337,
  "failed_transactions": 0,
  "longest_transaction": 4.03,
  "shortest_transaction": 0.00
}
```

## 2. Prueba corta de concurrencia:

Bash

```
siege -c 10 -t 1m http://localhost:8080
```

Se simularon 10 usuarios concurrentes realizando peticiones durante 1 minuto continuo. Esta prueba evalúa la estabilidad bajo una carga baja pero sostenida.

Resultados:

- Transacciones: 2337
- Tiempo de respuesta promedio: 0.25 s
- Disponibilidad: 100%
- Throughput: 0.14 MB/s

```
-[root@parrot]-[/home/user/haproxy1b/containerized]  
#siege -c 5 -t 30s http://localhost:8080
```

```
"transactions":          2788,  
"availability":         100.00,  
"elapsed_time":         29.71,  
"data_transferred":     9.88,  
"response_time":        0.05,  
"transaction_rate":     93.84,  
"throughput":           0.33,  
"concurrency":          4.83,  
"successful_transactions": 2788,  
"failed_transactions":   0,  
"longest_transaction":  1.34,  
"shortest_transaction":  0.00
```

### 3. Prueba de usuarios limitados:

```
siege -c 5 -t 30s http://localhost:8080
```

Se simularon 5 usuarios durante 30 segundos. Es una prueba liviana para verificar el rendimiento a corto plazo con pocos usuarios.

- Transacciones: 2788
- Tiempo de respuesta promedio: 0.05 s (excelente)
- Throughput: 0.33 MB/s

```
-[root@parrot]-[/home/user/haproxy1b/containerized]  
— #siege -b -c 5 -t 45s http://localhost:8080
```

```
"transactions":          3148,  
"availability":          100.00,  
"elapsed_time":          44.93,  
"data_transferred":      11.15,  
"response_time":         0.07,  
"transaction_rate":      70.06,  
"throughput":            0.25,  
"concurrency":           4.94,  
"successful_transactions": 3148,  
"failed_transactions":    0,  
"longest_transaction":   1.64,  
"shortest_transaction":  0.00
```

#### 4. Prueba sin delay:

siege -b -c 5 -t 45s http://localhost:8080

Prueba en modo benchmark, es decir, sin esperar entre peticiones. Los usuarios atacan el servidor a máxima velocidad durante 45 segundos.

Resultados:

- Transacciones: 3148
- Tiempo de respuesta promedio: 0.07 s
- Throughput: 0.25 MB/s

```
[root@parrot]-[/home/user/haproxy1b/containerized]
#siege -c 3 -r 2 http://localhost:8080/wp-admin/

{
  #Host: "transactions": 66,
  #Host: "availability": 100.00,
  #Host: "elapsed_time": 1.78,
  #Host: "data_transferred": 0.59,
  #Host: "response_time": 0.08,
  #Host: "transaction_rate": 37.08,
  #Host: "throughput": 0.33,
  #Host: "concurrency": 2.90,
  #Host: "successful_transactions": 66,
  #Host: "failed_transactions": 0,
  #Host: "longest_transaction": 0.53,
  #Host: "shortest_transaction": 0.00
}
```

### 1. Prueba de páginas específicas:

siege -c 3 -r 2 http://localhost:8080/wp-admin/

Prueba específica sobre el endpoint /wp-admin/. Ejecuta 2 peticiones por usuario con 3 usuarios.

Resultados:

- Transacciones: 66
- Tiempo de respuesta: 0.08 s
- Throughput: 0.33 MB/s

```
[root@parrot]~/home/user/haproxy/b/containerized]
#siege -i -c 5 -t 1m http://localhost:8080
```

```
{
  "transactions": 6540,
  "availability": 100.00,
  "elapsed_time": 59.35,
  "data_transferred": 23.17,
  "response_time": 0.04,
  "transaction_rate": 110.19,
  "throughput": 0.39,
  "concurrency": 4.94,
  "successful_transactions": 6540,
  "failed_transactions": 0,
  "longest_transaction": 0.50,
  "shortest_transaction": 0.00
}
```

## 2. Simulación de usuario real:

siege -i -c 5 -t 1m http://localhost:8080

Prueba en modo aleatorio (-i), útil para simular navegación real. Se usaron 5 usuarios durante 1 minuto.

Resultados:

- Transacciones: 6540
- Tiempo de respuesta: 0.04 s
- Throughput: 0.39 MB/s

```
[root@parrot]~/home/user/haproxy1b/containerized]
#siege -c 7 -d 1 -t 45s http://localhost:8080

{
  "transactions": 1680,
  "availability": 100.00,
  "elapsed_time": 44.62,
  "data_transferred": 5.95,
  "response_time": 0.05,
  "transaction_rate": 37.65,
  "throughput": 0.13,
  "concurrency": 1.84,
  "successful_transactions": 1680,
  "failed_transactions": 0,
  "longest_transaction": 0.70,
  "shortest_transaction": 0.00
}
```

### 3. Prueba con retardo mínimo:

siege -c 7 -d 1 -t 45s http://localhost:8080

7 usuarios con un delay de 1 segundo entre peticiones. Útil para simular carga moderada pero más realista.

Resultados:

- Transacciones: 1680
- Tiempo de respuesta: 0.05 s
- Throughput: 0.13 MB/s