

Perfilamiento del servidor: Se puede observar que las Shared libraries en el proceso con console.log los ticks son mas de la mitad que en el proceso sin el.

```
result_prof-sinconsole.txt
35
36 [Summary]:
37   ticks  total  nonlib   name
38     64    0.4%   94.1%  JavaScript
39      0    0.0%    0.0%   C++
40     59    0.4%   86.8%    GC
41   15934   99.6%           Shared libraries
42      4    0.0%           Unaccounted
43
```

```
result_prof-conconsole.txt
35
36 [Summary]:
37   ticks  total  nonlib   name
38     53    0.1%   93.0%  JavaScript
39      0    0.0%    0.0%   C++
40     64    0.1%  112.3%    GC
41   43245   99.9%           Shared libraries
42      4    0.0%           Unaccounted
43
```

Artillery: Podemos observar que los milisegundos de latencia son mucho mas altos con console.log. Y que la media de respuestas por segundo es mucho mas alta sin console.log()

```
result_sinconsole.txt
17 All virtual users finished
18 Summary report @ 15:08:44(-0300) 2021-12-16
19   Scenarios launched: 20
20   Scenarios completed: 20
21   Requests completed: 1000
22   Mean response/sec: 195.31
23   Response time (msec):
24     min: 0
25     max: 163
26     median: 64
27     p95: 118
28     p99: 145
29   Scenario counts:
30     0: 20 (100%)
31   Codes:
32     200: 1000
```

```
result_conconsole.txt
32 All virtual users finished
33 Summary report @ 16:58:06(-0300) 2021-12-16
34   Scenarios launched: 20
35   Scenarios completed: 20
36   Requests completed: 1000
37   Mean response/sec: 56.15
38   Response time (msec):
39     min: 13
40     max: 906
41     median: 331.5
42     p95: 686
43     p99: 855
44   Scenario counts:
45     0: 20 (100%)
46   Codes:
47     200: 1000
```

Autocannon:

Sin console.log()

```
Running all benchmarks in parallel ...
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	132 ms	160 ms	284 ms	342 ms	173.64 ms	44.76 ms	493 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	341	341	574	709	573.21	112.27	341
Bytes/Sec	165 kB	165 kB	278 kB	343 kB	277 kB	54.3 kB	165 kB

Req/Bytes counts sampled once per second.

12k requests in 20.11s, 5.55 MB read

Con console.log

```
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	126 ms	156 ms	381 ms	544 ms	178.18 ms	71.77 ms	719 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	246	246	601	704	559.4	141.74	246
Bytes/Sec	119 kB	119 kB	291 kB	341 kB	271 kB	68.6 kB	119 kB

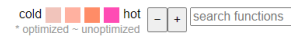
Req/Bytes counts sampled once per second.

11k requests in 20.19s, 5.41 MB read

Diagrama de flama: Lo que se puede diferenciar a simple vista es la horizontalidad, para console.log() es mucho mas largo, esto quiere decir que los procesos duran mas a diferencia de sin console.log().

Sin console.log():

```
node server.js
```



Con `console.log()`:

```
node server.js
```

