Sin console log

```console

$ node benchmark.js
Running tests
Running 20s test @ http://localhost:3030/test/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	131 ms	160 ms	293 ms	424 ms	173.93 ms	51.78 ms	665 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	257	257	600	700	572.15	113.28	257
Bytes/Sec	154 kb	154 kb	361 kb	421 kb	344 kb	68.2 kb	154 kb

```
Req/Bytes counts sampled once per second.
# of samples: 20

12k requests in 20.07s, 6.88 MB read

```
Resultado del archivo Artillery:

```console

http.response_time:
  min:
.....................................................................
4
  max:
.....................................................................
393
  median:
.....................................................................
141.2
  p95:
.....................................................................
210.6
  p99:
.....................................................................
308
```

```

Con console Log

```console

└─>$ node benchmark.js
Running tests
Running 20s test @ http://localhost:3030/test/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	173 ms	220 ms	446 ms	562 ms	238.81 ms	70.81 ms	830 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	218	218	427	540	415.35	81.88	218
Bytes/Sec	131 kb	131kb	257 kb	325 kb	2540 kb	49 kb	131 kb

```
Req/Bytes counts sampled once per second.
# of samples: 20

8k requests in 20.07s, 5 MB read

```

Resultado del archivo Artillery:

```console

http.response_time:
  min:
.....................................................................
15
  max:
.....................................................................
461
  median:
.....................................................................
172.5
  p95:
.....................................................................
228.2
```
```

```
  p99:
.....................................................................
376.2

```

Las pruebas realizadas indican que en el caso de loguear por consola el tiempo de respuesta es mayor y que se pueden manejar menos requests en el mismo tiempo.