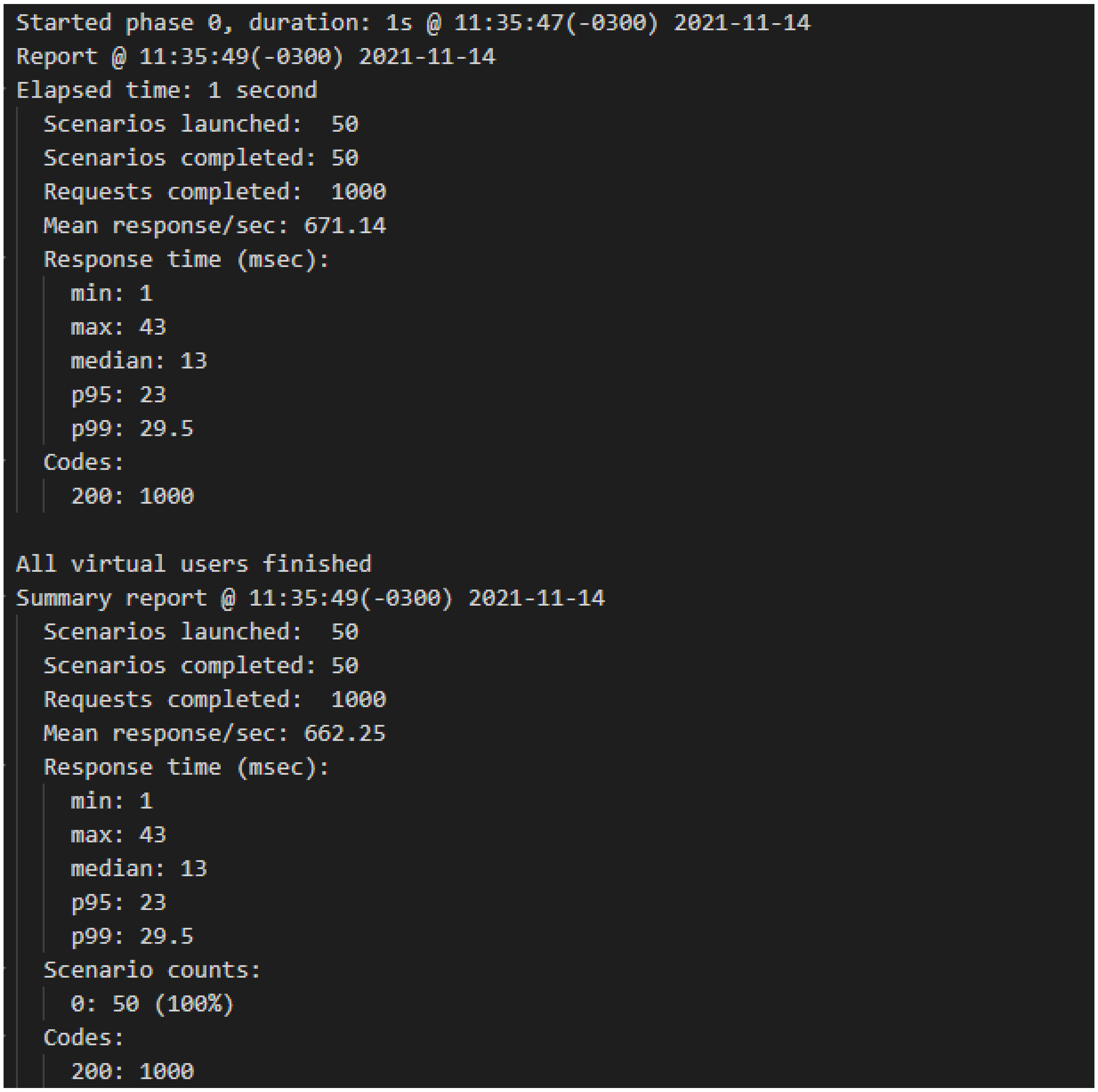
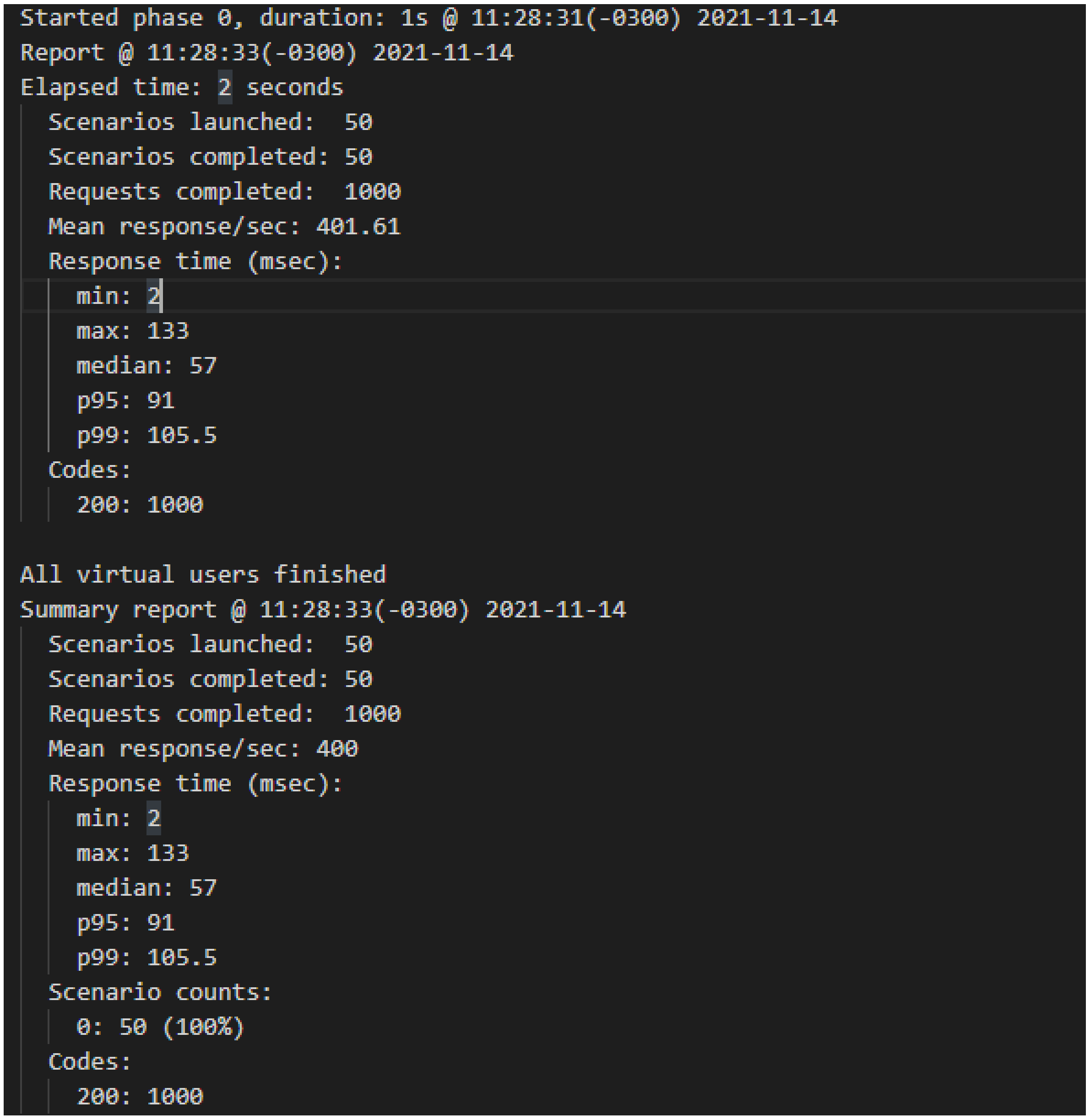
ARTILLERY

Con console.log



```
Started phase 0, duration: 1s @ 11:35:47(-0300) 2021-11-14
Report @ 11:35:49(-0300) 2021-11-14
Elapsed time: 1 second
  Scenarios launched:  50
  Scenarios completed: 50
  Requests completed:  1000
  Mean response/sec: 671.14
  Response time (msec):
    min: 1
    max: 43
    median: 13
    p95: 23
    p99: 29.5
  Codes:
    200: 1000

All virtual users finished
Summary report @ 11:35:49(-0300) 2021-11-14
  Scenarios launched:  50
  Scenarios completed: 50
  Requests completed:  1000
  Mean response/sec: 662.25
  Response time (msec):
    min: 1
    max: 43
    median: 13
    p95: 23
    p99: 29.5
  Scenario counts:
    0: 50 (100%)
  Codes:
    200: 1000
```

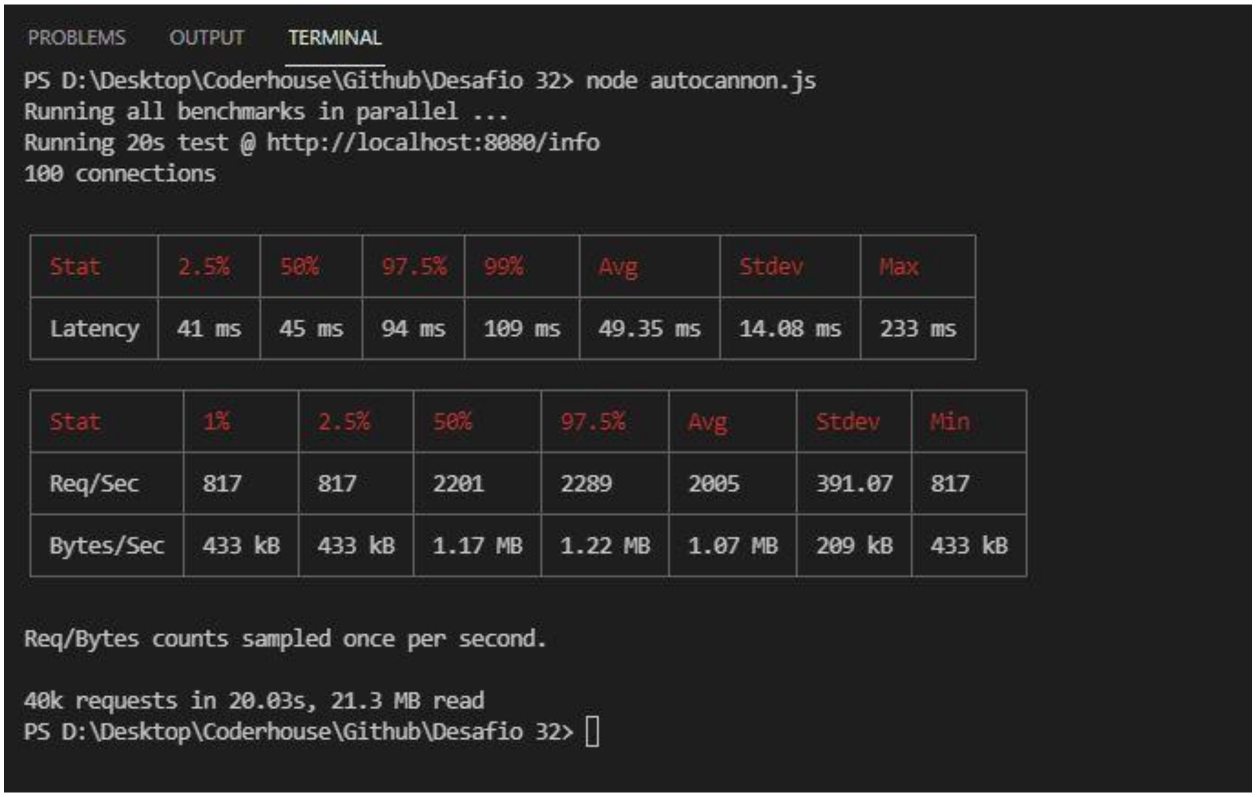
Sin console.log



```
Started phase 0, duration: 1s @ 11:28:31(-0300) 2021-11-14
Report @ 11:28:33(-0300) 2021-11-14
Elapsed time: 2 seconds
  Scenarios launched:  50
  Scenarios completed: 50
  Requests completed:  1000
  Mean response/sec: 401.61
  Response time (msec):
    min: 2
    max: 133
    median: 57
    p95: 91
    p99: 105.5
  Codes:
    200: 1000

All virtual users finished
Summary report @ 11:28:33(-0300) 2021-11-14
  Scenarios launched:  50
  Scenarios completed: 50
  Requests completed:  1000
  Mean response/sec: 400
  Response time (msec):
    min: 2
    max: 133
    median: 57
    p95: 91
    p99: 105.5
  Scenario counts:
    0: 50 (100%)
  Codes:
    200: 1000
```

AUTOCANNON

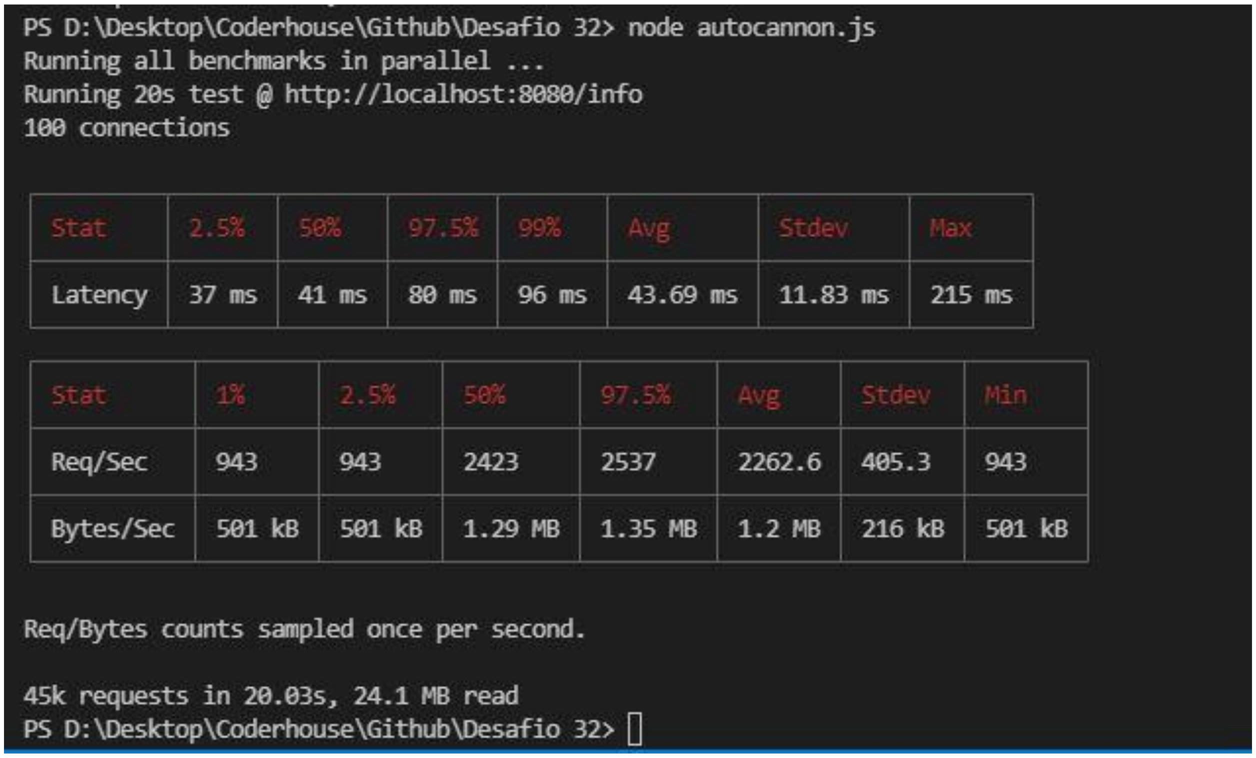
Con console.log



PROBLEMS OUTPUT TERMINAL

```
PS D:\Desktop\Coderhouse\Github\Desafio 32> node autocannon.js
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 │ 41 ms │ 45 ms │ 94 ms │ 109 ms │ 49.35 ms │ 14.08 ms │ 233 ms │
└─────────┴───────┴───────┴───────┴────────┴──────────┴──────────┴────────┘
┌───────────┬────────┬────────┬─────────┬─────────┬─────────┬────────┬────────┐
│ Stat      │ 1%     │ 2.5%   │ 50%     │ 97.5%   │ Avg     │ Stdev  │ Min    │
├───────────┼────────┼────────┼─────────┼─────────┼─────────┼────────┼────────┤
│ Req/Sec   │ 817    │ 817    │ 2201    │ 2289    │ 2005    │ 391.07 │ 817    │
├───────────┼────────┼────────┼─────────┼─────────┼─────────┼────────┼────────┤
│ Bytes/Sec │ 433 kB │ 433 kB │ 1.17 MB │ 1.22 MB │ 1.07 MB │ 209 kB │ 433 kB │
└───────────┴────────┴────────┴─────────┴─────────┴─────────┴────────┴────────┘

Req/Bytes counts sampled once per second.

40k requests in 20.03s, 21.3 MB read
PS D:\Desktop\Coderhouse\Github\Desafio 32> 
```

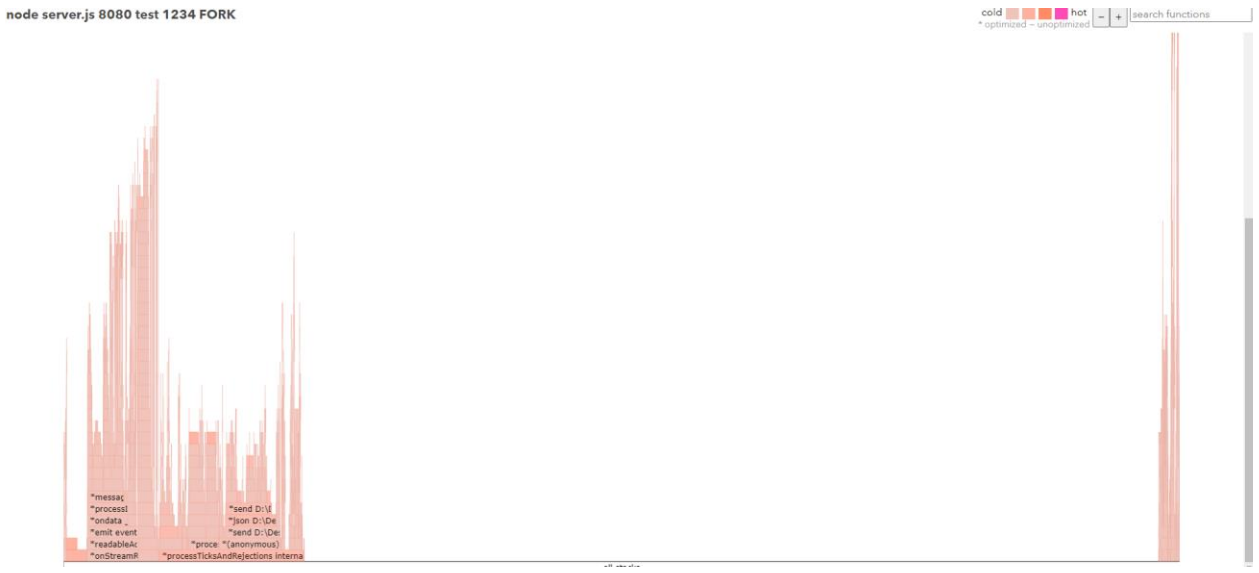
Sin console.log



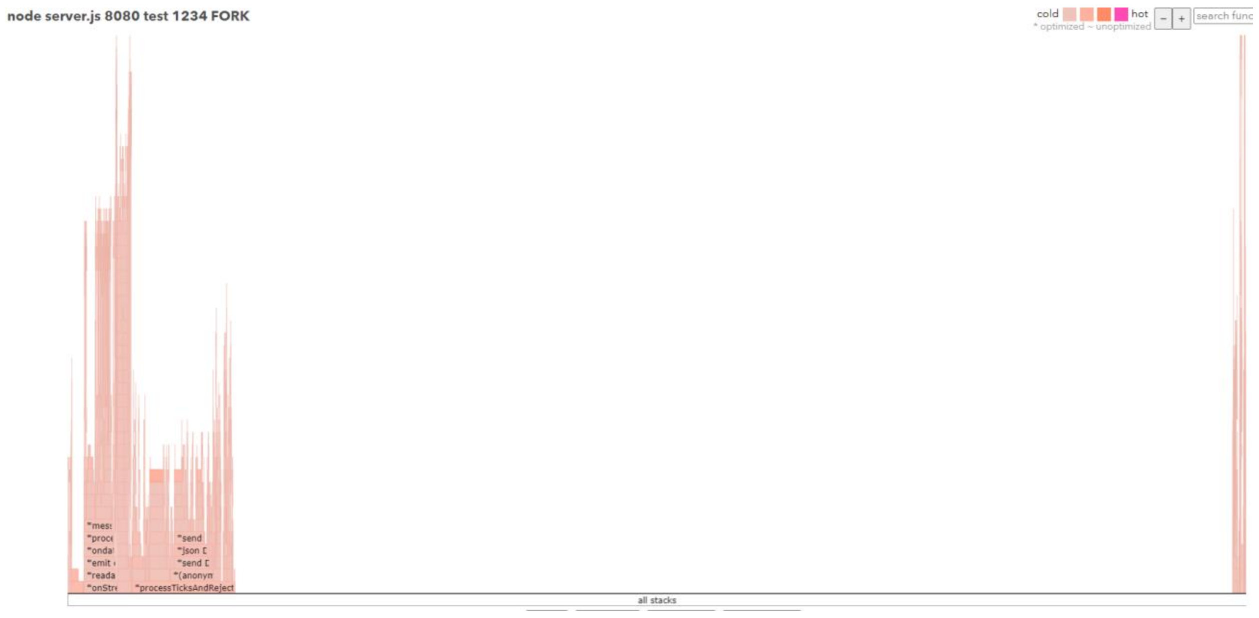
```
PS D:\Desktop\Coderhouse\Github\Desafio 32> node autocannon.js
Running all benchmarks in parallel ...
Running 20s test @ http://localhost:8080/info
100 connections

┌─────────┬───────┬───────┬───────┬───────┬──────────┬──────────┬────────┐
│ Stat    │ 2.5%  │ 50%   │ 97.5% │ 99%   │ Avg      │ Stdev    │ Max    │
├─────────┼───────┼───────┼───────┼───────┼──────────┼──────────┼────────┤
│ Latency │ 37 ms │ 41 ms │ 80 ms │ 96 ms │ 43.69 ms │ 11.83 ms │ 215 ms │
└─────────┴───────┴───────┴───────┴───────┴──────────┴──────────┴────────┘
┌───────────┬────────┬────────┬─────────┬─────────┬────────┬────────┬────────┐
│ Stat      │ 1%     │ 2.5%   │ 50%     │ 97.5%   │ Avg    │ Stdev  │ Min    │
├───────────┼────────┼────────┼─────────┼─────────┼────────┼────────┼────────┤
│ Req/Sec   │ 943    │ 943    │ 2423    │ 2537    │ 2262.6 │ 405.3  │ 943    │
├───────────┼────────┼────────┼─────────┼─────────┼────────┼────────┼────────┤
│ Bytes/Sec │ 501 kB │ 501 kB │ 1.29 MB │ 1.35 MB │ 1.2 MB │ 216 kB │ 501 kB │
└───────────┴────────┴────────┴─────────┴─────────┴────────┴────────┴────────┘

Req/Bytes counts sampled once per second.

45k requests in 20.03s, 24.1 MB read
PS D:\Desktop\Coderhouse\Github\Desafio 32> 
```

Resultados en 0x

Con console.log



Sin console.log



En todas las pruebas podemos ver, tal como analizamos en clase que el console.log al ser un proceso bloqueante, genera mas tiempo de demora en las requests.