Profiling

Sin console.log()

Node -prof:

```
[Summary]:
  ticks  total  nonlib   name
     54    4.0%    4.1%  JavaScript
   1248   91.7%   95.1%  C++
     82    6.0%    6.3%  GC
     49    3.6%          Shared libraries
     10    0.7%          Unaccounted
```

Artillery:

```
--------------------------------
Summary report @ 23:47:03(-0300)
--------------------------------

http.codes.200: ..................................................................... 1000
http.request_rate: .................................................................. 981/sec
http.requests: ...................................................................... 1000
http.response_time:
  min: .............................................................................. 1
  max: .............................................................................. 34
  median: ........................................................................... 7.9
  p95: .............................................................................. 13.1
  p99: .............................................................................. 16
http.responses: ..................................................................... 1000
vusers.completed: ................................................................... 50
vusers.created: ..................................................................... 50
vusers.created_by_name.0: ........................................................... 50
vusers.session_length:
  min: .............................................................................. 46.8
  max: .............................................................................. 229.5
  median: ........................................................................... 175.9
  p95: .............................................................................. 219.2
  p99: .............................................................................. 223.7
```

Autocannon

```
> entrega_02@1.0.0 test
> node benchmark.js

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

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	42 ms	46 ms	70 ms	81 ms	49.14 ms	8.41 ms	149 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	1324	1324	2057	2165	2012.7	186.14	1324
Bytes/Sec	2.48 MB	2.48 MB	3.86 MB	4.06 MB	3.77 MB	349 kB	2.48 MB

```
Req/Bytes counts sampled once per second.

40k requests in 20.02s, 75.4 MB read
```

:

0x (flama):

Con console.log()

Node -prof:

```
[Summary]:
  ticks  total  nonlib   name
     82   5.4%    5.6%  JavaScript
   1369  89.9%   93.1%  C++
    153  10.1%   10.4%  GC
     51   3.4%          Shared libraries
     20   1.3%          Unaccounted
```

Artillery:

```
--------------------------------
Summary report @ 23:44:29(-0300)
--------------------------------

http.codes.200: ..................................................................... 1000
http.request_rate: .................................................................. 898/sec
http.requests: ...................................................................... 1000
http.response_time:
  min: .............................................................................. 1
  max: .............................................................................. 40
  median: ........................................................................... 12.1
  p95: .............................................................................. 21.1
  p99: .............................................................................. 37
http.responses: ..................................................................... 1000
vusers.completed: ................................................................... 50
vusers.created: ..................................................................... 50
vusers.created_by_name.0: ........................................................... 50
vusers.session_length:
  min: .............................................................................. 142
  max: .............................................................................. 321.3
  median: ........................................................................... 278.7
  p95: .............................................................................. 314.2
  p99: .............................................................................. 320.6
```

Autocannon:

```
> entrega_02@1.0.0 test
> node benchmark.js

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

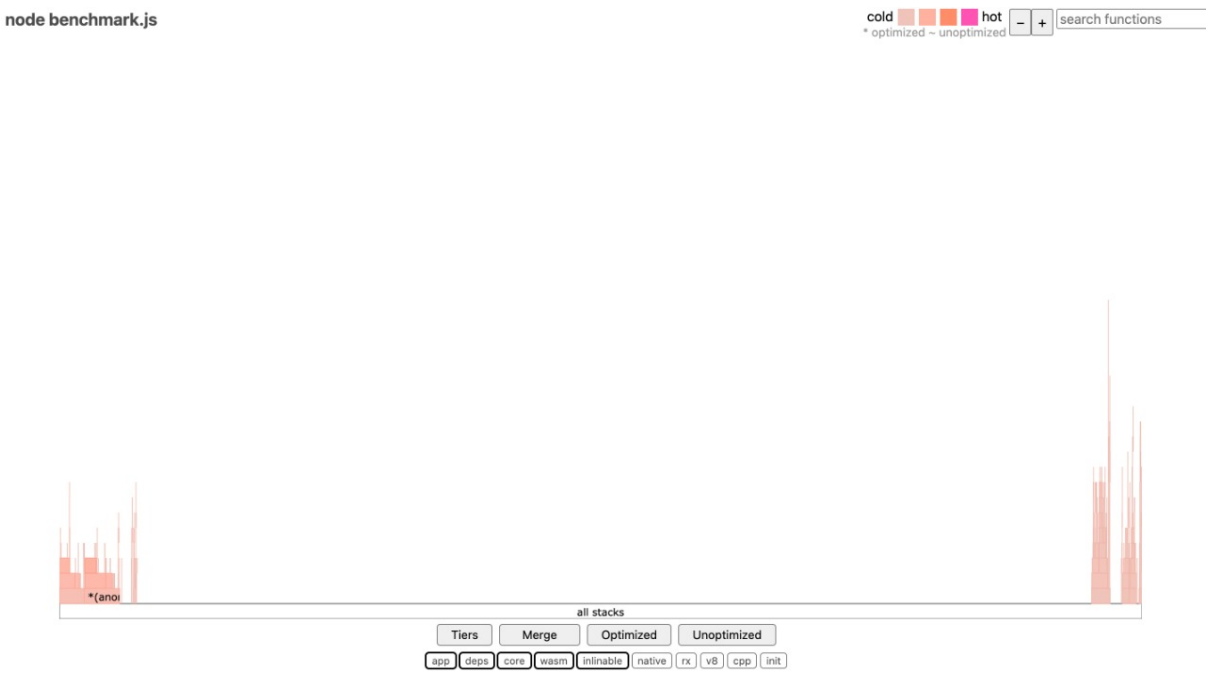
Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	42 ms	45 ms	68 ms	81 ms	48.01 ms	7.74 ms	114 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	1382	1382	2101	2211	2060.7	179.16	1382
Bytes/Sec	2.59 MB	2.59 MB	3.94 MB	4.14 MB	3.86 MB	335 kB	2.59 MB

```
Req/Bytes counts sampled once per second.

41k requests in 20.02s, 77.2 MB read
```

0x (flama):



Conclusiones:

No sé si se debe a algún problema con mi máquina, pero en los profiling de node y artillery pasó lo esperado: que el console.log() bajara la performance. En cambio, a mí parecer tanto en 0x como en autocannon parece ser al revés, quizás estoy haciendo una mala lectura.

Cabe destacar que no pude usar el inspector de Chrome dado que continuamente fallaba porque otro proceso estaba tomando el puerto 9229 que debía utilizar el inspector, y a pesar de pasearme por todos los posts de stack overflow e issues de github, no encontré solución al problema.