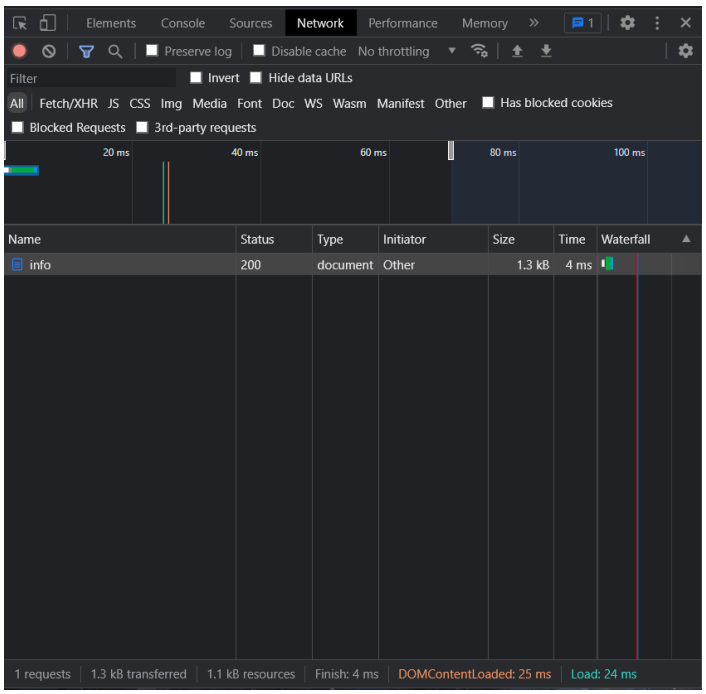
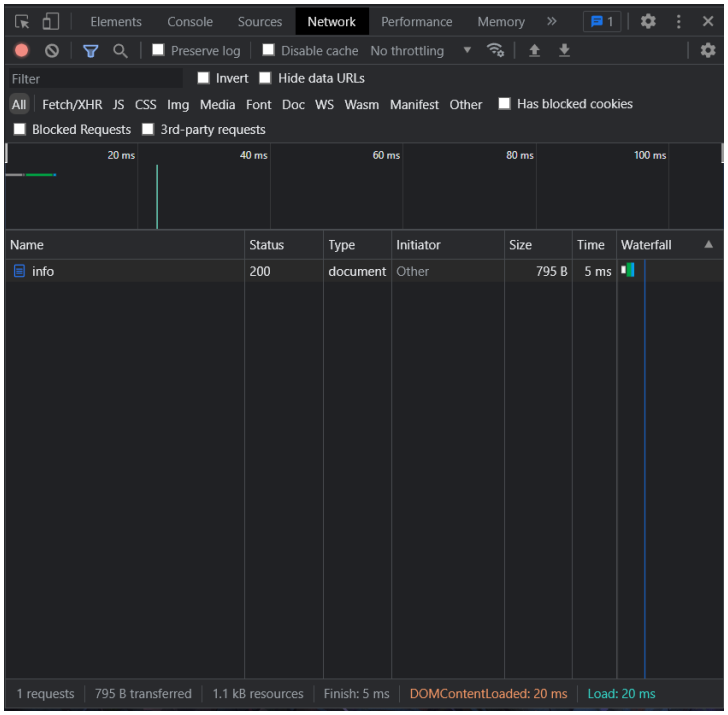
Desafío 16

- Info sin compresión:



- Info con compresión:



- --Prof con console.log:

```
[Summary]:
   ticks   total   nonlib   name
       4    0.1%  100.0%  JavaScript
       0    0.0%    0.0%  C++
       3    0.1%   75.0%  GC
    4983   99.9%          Shared libraries
```

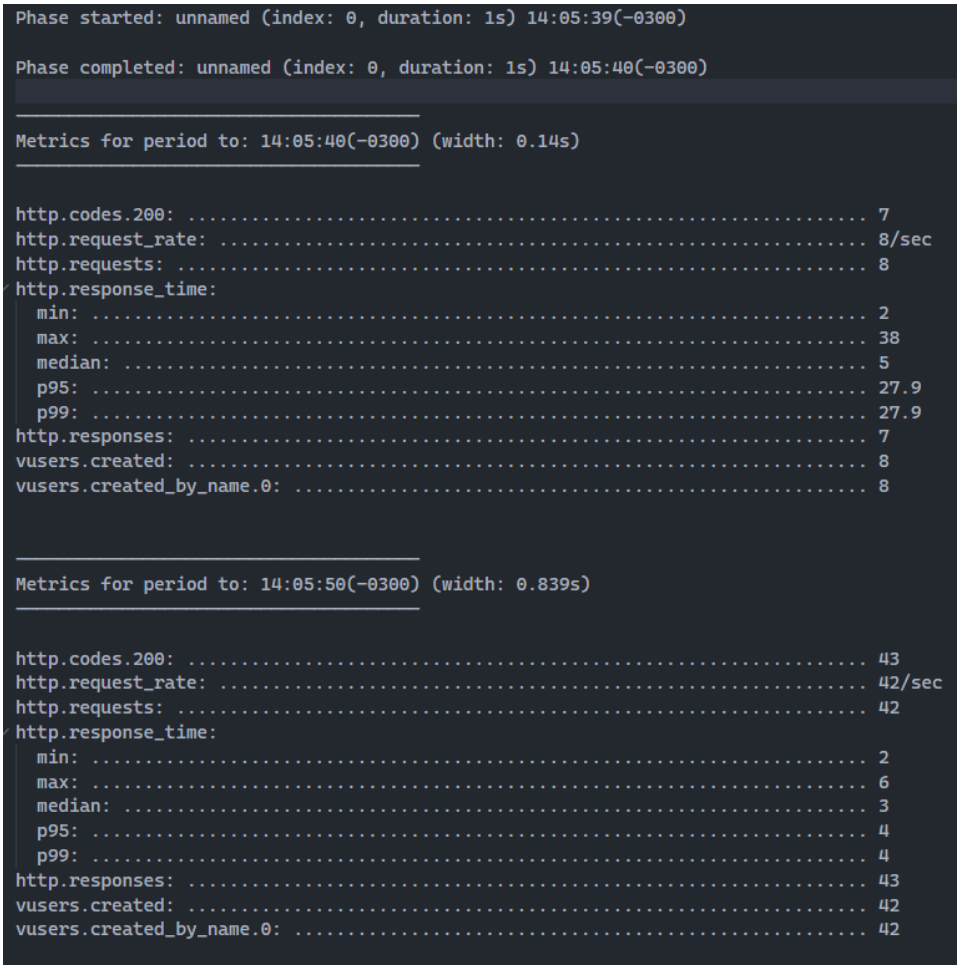
- --Prof sin console.log:

```
[Summary]:
   ticks   total   nonlib   name
       6    0.3%  100.0%  JavaScript
       0    0.0%    0.0%  C++
       5    0.2%   83.3%  GC
    2101   99.7%          Shared libraries
```

- Artillery con console.log:

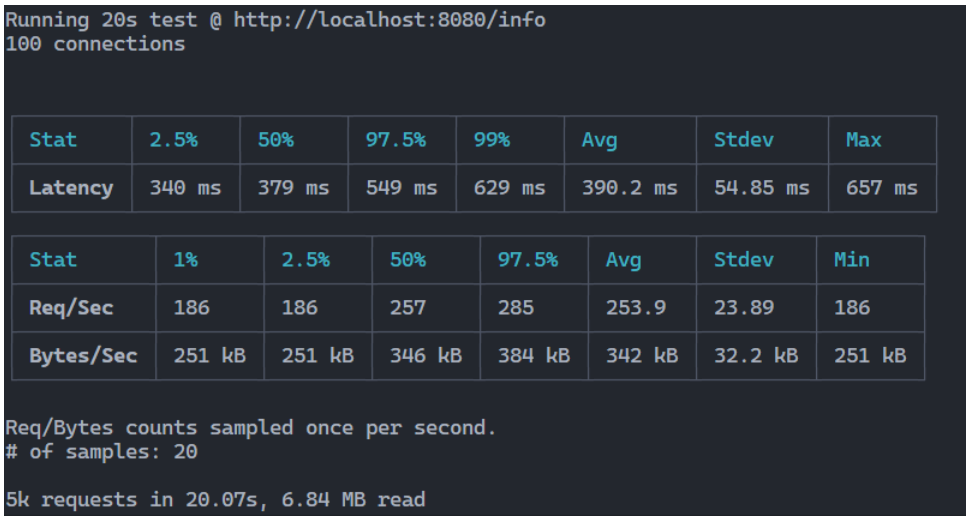
```
Phase started: unnamed (index: 0, duration: 1s) 14:50:39(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 14:50:40(-0300)

─────────────────────────────────────

Metrics for period to: 14:50:40(-0300) (width: 0.665s)
─────────────────────────────────────

http.codes.200: ..................................................... 34
http.request_rate: .................................................. 34/se
http.requests: ...................................................... 34
http.response_time:
  min: .............................................................. 5
  max: .............................................................. 13
  median: ........................................................... 5
  p95: .............................................................. 8.9
  p99: .............................................................. 10.1
http.responses: ..................................................... 34
vusers.created: ..................................................... 34
vusers.created_by_name.0: ........................................... 34


─────────────────────────────────────
Metrics for period to: 14:50:50(-0300) (width: 0.307s)
─────────────────────────────────────

http.codes.200: ..................................................... 16
http.request_rate: .................................................. 16/se
http.requests: ...................................................... 16
http.response_time:
  min: .............................................................. 4
  max: .............................................................. 8
  median: ........................................................... 5
  p95: .............................................................. 6
  p99: .............................................................. 6
http.responses: ..................................................... 16
vusers.created: ..................................................... 16
vusers.created_by_name.0: ........................................... 16
```

- Artillery sin console.log:



```
Phase started: unnamed (index: 0, duration: 1s) 14:05:39(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 14:05:40(-0300)

-------------------------------------
Metrics for period to: 14:05:40(-0300) (width: 0.14s)
-------------------------------------

http.codes.200: ..................................................... 7
http.request_rate: .................................................. 8/sec
http.requests: ...................................................... 8
http.response_time:
  min: .............................................................. 2
  max: .............................................................. 38
  median: ........................................................... 5
  p95: .............................................................. 27.9
  p99: .............................................................. 27.9
http.responses: ..................................................... 7
vusers.created: ..................................................... 8
vusers.created_by_name.0: ........................................... 8


-------------------------------------
Metrics for period to: 14:05:50(-0300) (width: 0.839s)
-------------------------------------

http.codes.200: ..................................................... 43
http.request_rate: .................................................. 42/sec
http.requests: ...................................................... 42
http.response_time:
  min: .............................................................. 2
  max: .............................................................. 6
  median: ........................................................... 3
  p95: .............................................................. 4
  p99: .............................................................. 4
http.responses: ..................................................... 43
vusers.created: ..................................................... 42
vusers.created_by_name.0: ........................................... 42
```

- Autocannon con console.log:



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

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	340 ms	379 ms	549 ms	629 ms	390.2 ms	54.85 ms	657 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	186	186	257	285	253.9	23.89	186
Bytes/Sec	251 kB	251 kB	346 kB	384 kB	342 kB	32.2 kB	251 kB

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

5k requests in 20.07s, 6.84 MB read

- Autocannon sin console.log:

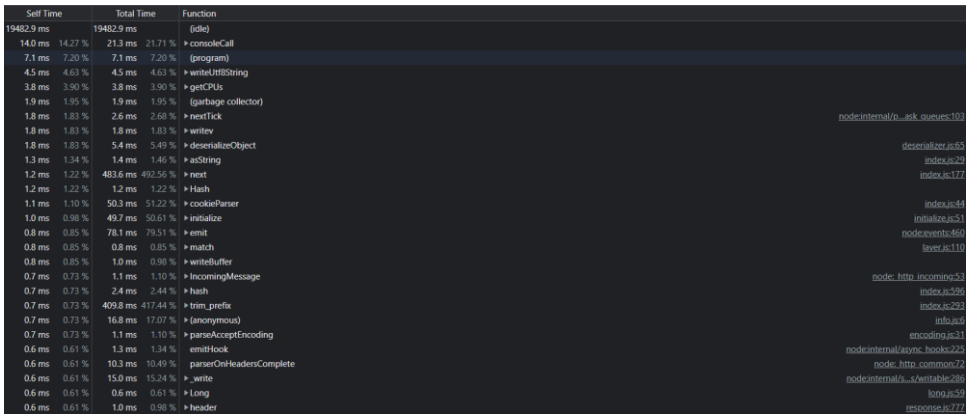
```
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	117 ms	136 ms	181 ms	193 ms	138.47 ms	17.05 ms	250 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	540	540	720	795	717.85	61.54	540
Bytes/Sec	727 kB	727 kB	969 kB	1.07 MB	966 kB	82.8 kB	727 kB

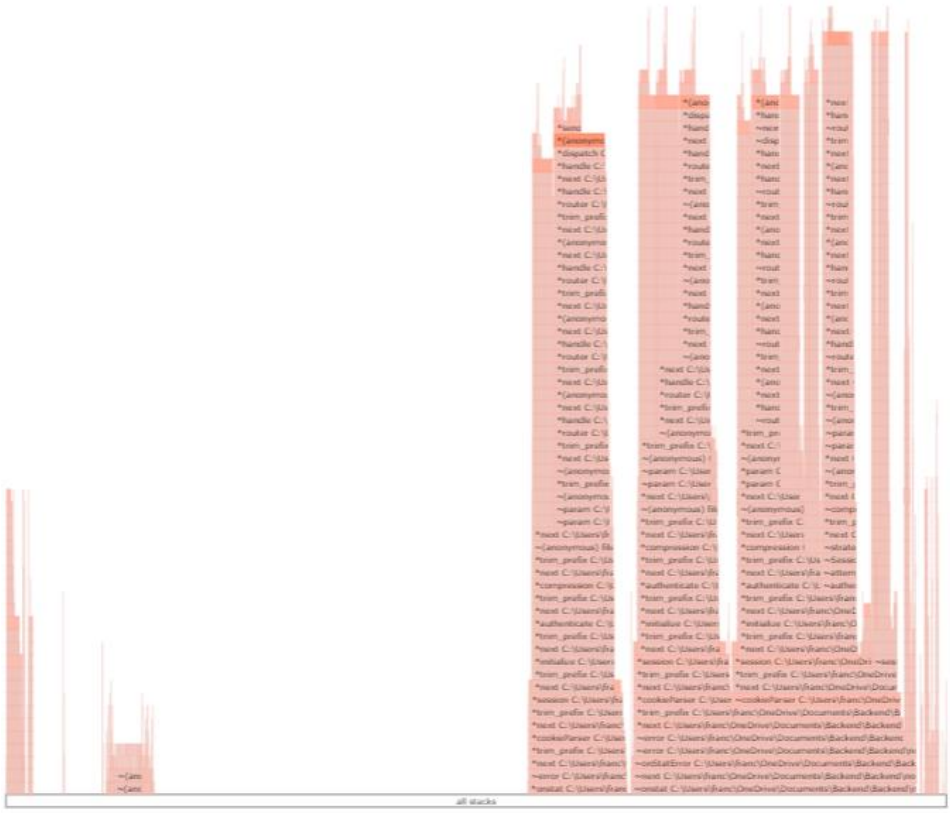
```
Req/Bytes counts sampled once per second.
# of samples: 20

14k requests in 20.05s, 19.3 MB read
```

- Inspector:

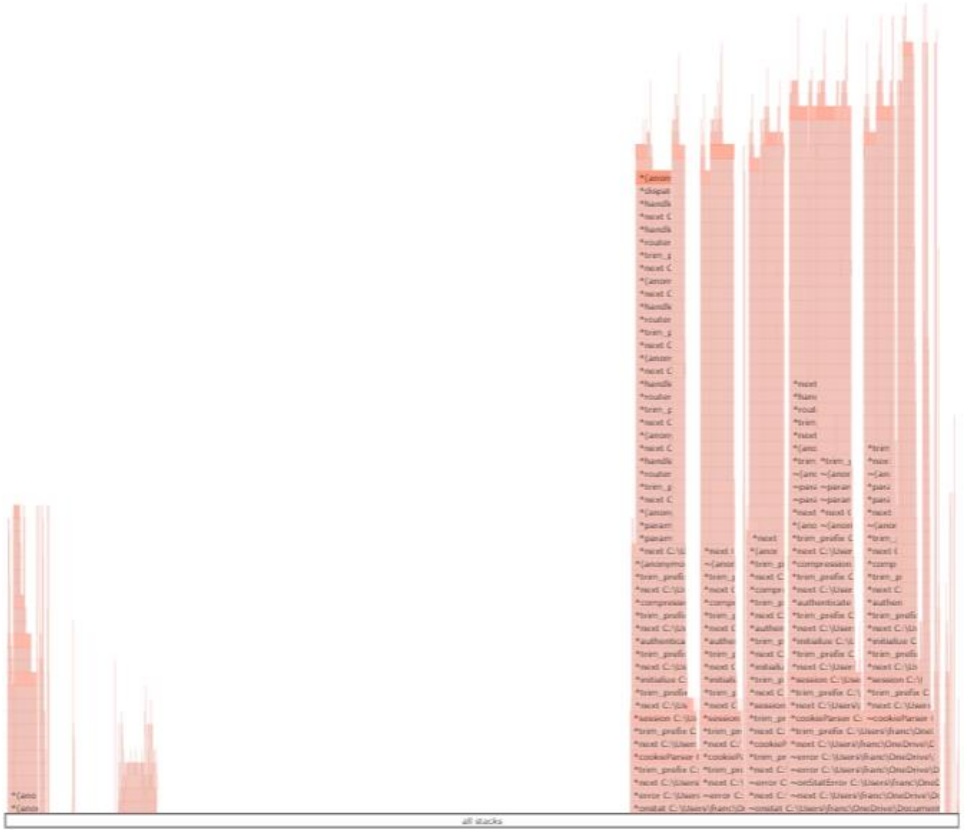


Self Time		Total Time		Function	
19482.9 ms		19482.9 ms		(idle)	
14.0 ms	14.27 %	21.3 ms	21.71 %	consoleCall	
7.1 ms	7.20 %	7.1 ms	7.20 %	(program)	
4.5 ms	4.63 %	4.5 ms	4.63 %	writeUtf8String	
3.8 ms	3.90 %	3.8 ms	3.90 %	getCPUs	
1.9 ms	1.95 %	1.9 ms	1.95 %	(garbage collector)	
1.8 ms	1.83 %	2.6 ms	2.68 %	nextTick	node:internal/p...ask_queues:103
1.8 ms	1.83 %	1.8 ms	1.83 %	writev	
1.8 ms	1.83 %	5.4 ms	5.49 %	deserializeObject	deserializer.js:65
1.3 ms	1.34 %	1.4 ms	1.46 %	asString	index.js:29
1.2 ms	1.22 %	483.6 ms	492.56 %	next	index.js:177
1.2 ms	1.22 %	1.2 ms	1.22 %	Hash	
1.1 ms	1.10 %	50.3 ms	51.22 %	cookieParser	index.js:44
1.0 ms	0.98 %	49.7 ms	50.61 %	initialize	initialize.js:51
0.8 ms	0.85 %	78.1 ms	79.51 %	emit	node:events:460
0.8 ms	0.85 %	0.8 ms	0.85 %	match	layer.js:110
0.8 ms	0.85 %	1.0 ms	0.98 %	writeBuffer	
0.7 ms	0.73 %	1.1 ms	1.10 %	IncomingMessage	node: http_incoming:53
0.7 ms	0.73 %	2.4 ms	2.44 %	hash	index.js:596
0.7 ms	0.73 %	409.8 ms	417.44 %	trim_prefix	index.js:293
0.7 ms	0.73 %	16.8 ms	17.07 %	(anonymous)	info.js:6
0.7 ms	0.73 %	1.1 ms	1.10 %	parseAcceptEncoding	encoding.js:31
0.6 ms	0.61 %	1.3 ms	1.34 %	emitHook	node:internal/async_hooks:225
0.6 ms	0.61 %	10.3 ms	10.49 %	parserOnHeadersComplete	node: http_common:72
0.6 ms	0.61 %	15.0 ms	15.24 %	_write	node:internal/s...s/writable:286
0.6 ms	0.61 %	0.6 ms	0.61 %	Long	long.js:59
0.6 ms	0.61 %	1.0 ms	0.98 %	header	response.js:777

- 0x con console.log:



- 0x sin console.log:



Conclusión: Siempre es recomendable utilizar funciones no bloqueantes, ya que éstas tienen un impacto negativo en el rendimiento del servidor.