Profiling

Con console.log()

1) Node.js Prof.

result-prof-bloq.txt

```
Statistical profiling result from .\isolate-000001C6B56A8C80-17000-v8.log, (4560 ticks, 1 unaccounted, 0 excluded).

 [Shared libraries]:
   ticks  total  nonlib   name
    3962   86.9%          C:\WINDOWS\SYSTEM32\ntdll.dll
     572   12.5%          C:\Program Files\nodejs\node.exe

 [JavaScript]:
   ticks  total  nonlib   name
       8    0.2%   30.8%  RegExp: [ \t]*<%_
       4    0.1%   15.4%  LazyCompile: *next C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\express\lib\router\index.js:176:16
       3    0.1%   11.5%  LazyCompile: *scanLine C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\ejs\lib\ejs.js:803:22
       2    0.0%    7.7%  LazyCompile: *resolve path.js:153:10
       1    0.0%    3.8%  RegExp: ^\/app\/?(?=\/|$)
       1    0.0%    3.8%  RegExp: [|\\{}()[\]^$+*?.]
       1    0.0%    3.8%  RegExp: ;(\s*$)
       1    0.0%    3.8%  LazyCompile: *nextTick internal/process/task_queues.js:103:18
       1    0.0%    3.8%  LazyCompile: *log C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\winston\lib\winston\transports\console.js:44:6
       1    0.0%    3.8%  LazyCompile: *isBelowBreakLength internal/util/inspect.js:1711:28
       1    0.0%    3.8%  LazyCompile: *hash C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\express-session\index.js:596:14
       1    0.0%    3.8%  LazyCompile: *Writable.write
       C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\winston-transport\node_modules\readable-stream\lib\_stream_writable.js:288:37

 [C++]:
   ticks  total  nonlib   name

 [Summary]:
   ticks  total  nonlib   name
      25    0.5%   96.2%  JavaScript
       0    0.0%    0.0%  C++
      12    0.3%   46.2%  GC
    4534   99.4%          Shared libraries
       1    0.0%          Unaccounted

 [C++ entry points]:
   ticks    cpp   total   name

 [Bottom up (heavy) profile]:
  Note: percentage shows a share of a particular caller in the total
  amount of its parent calls.
  Callers occupying less than 1.0% are not shown.
```

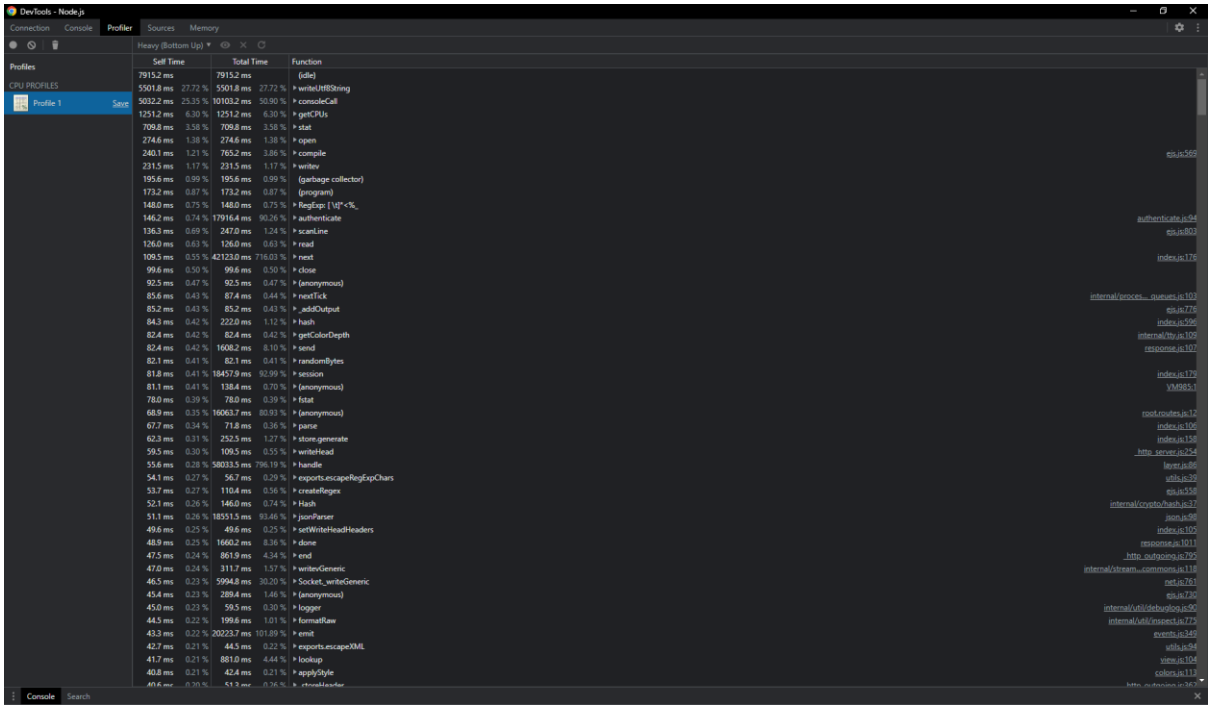
2) Node.js Inspect.

```
Yonathan Moncada@DESKTOP-C5A5PF7 MINGW64 /c/AppServ/www/curso-programacion-backend-coderhouse (main)
$ node benchmark.js
Running benchmarks in Parallel
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	360 ms	393 ms	460 ms	490 ms	394.85 ms	35.23 ms	501 ms

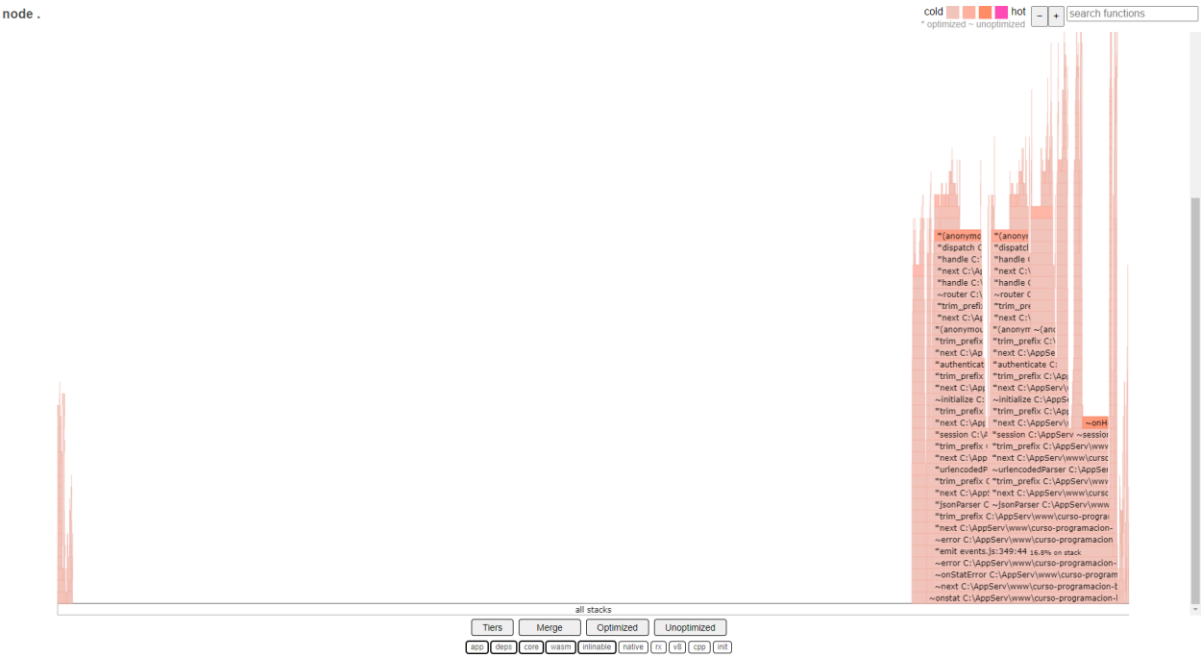
Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	214	214	251	271	251	15.22	214
Bytes/Sec	420 kB	420 kB	492 kB	531 kB	492 kB	29.9 kB	419 kB

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

5k requests in 20.06s, 9.84 MB read
```




```
1         const express = require('express');
2         const compression = require('compression');
3         const os = require('os');
4         const logger = require('../utils/logger.utils');
5
6         const rootRouter = express.Router();
7
8         rootRouter.get('/', function(req, res) {
9             res.redirect('/app/auth/login');
10        });
11
12   0.8 ms rootRouter.get('/info', (req, res) => {
13   1.6 ms     const info = {
14   2.5 ms         inputArguments: process.argv.slice(2),
15   3.0 ms         cpus: os.cpus().length,
16   1.4 ms         platformName: process.platform,
17   0.8 ms         nodeJsVersion: process.version,
18   1.3 ms         reservedTotalMemory: process.memoryUsage().rss,
19   0.7 ms         executionPath: process.execPath,
20   0.2 ms         processId: process.pid,
21   0.9 ms         projectFolder: process.cwd()
22              };
23
24  30.1 ms     console.log(info);
25
26  19.7 ms     res.render('pages/info.ejs', { info });
27   0.2 ms });
28
29        rootRouter.get('/info-compression', compression(), (req, res) => {
30            res.render('pages/info.ejs', { process: process, cwd: process.cwd(), rss: process.memoryUsage().rss, argv: process.argv.slice(2), cpus: os.cpus().length });
31        });
32
33        rootRouter.all('*', (req, res) => {
34            logger.write('warn', `La ruta [${req.method}] ${req.protocol + '://' + req.get('host') + req.originalUrl} es inexistente en el servidor.`);
35            res.json({status: false});
36        });
37
38        module.exports = rootRouter;
```

3) 0x.



Sin console.log()

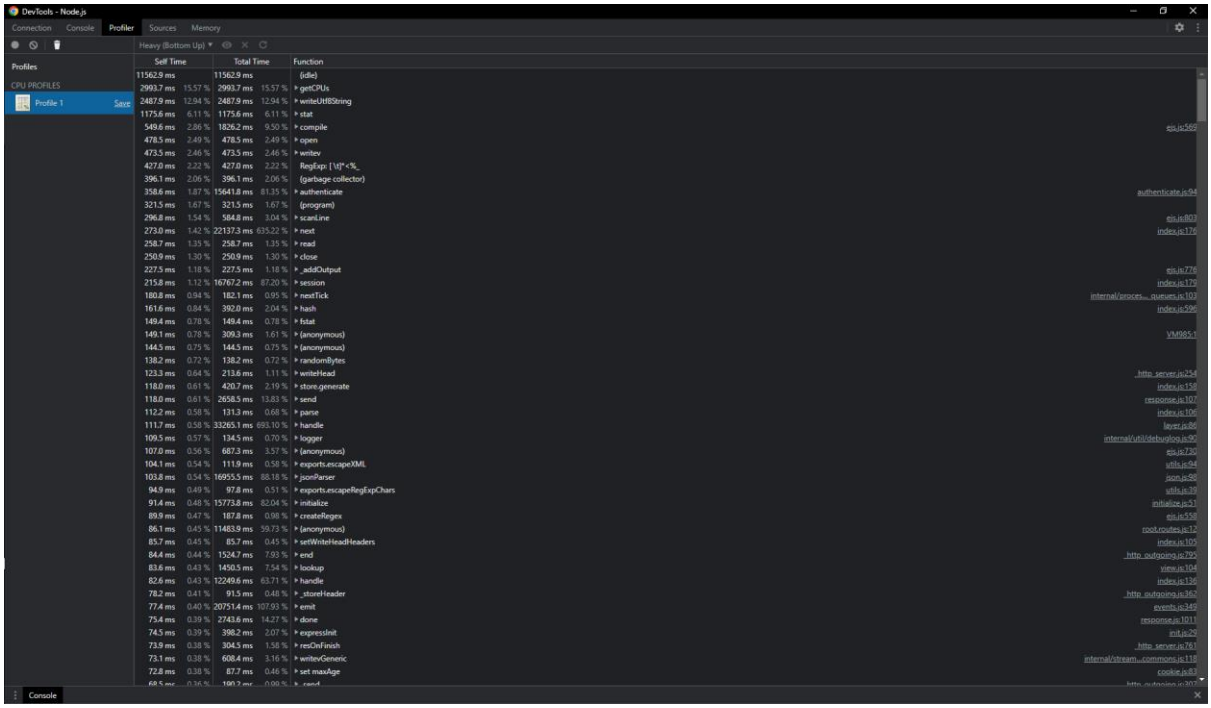
1) **Node.js Prof.**

result-prof-nobloq.txt

```
Statistical profiling result from .\isolate-000001FB0DB274C0-8856-v8.log, (3057 ticks, 3 unaccounted, 0 excluded).

 [Shared libraries]:
   ticks  total  nonlib   name
    2504   81.9%          C:\WINDOWS\SYSTEM32\ntdll.dll
     491   16.1%          C:\Program Files\nodejs\node.exe
       4    0.1%          C:\WINDOWS\System32\KERNELBASE.dll

 [JavaScript]:
   ticks  total  nonlib   name
      16    0.5%   27.6%  RegExp: [ \t]*<%_
       4    0.1%    6.9%  LazyCompile: *scanLine C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\ejs\lib\ejs.js:803:22
       2    0.1%    3.4%  RegExp: ('<%%|%%>|<%=|<%-|<%_|<%#|<%|%>|-%>|_%>)
       2    0.1%    3.4%  LazyCompile: *resolve path.js:153:10
       2    0.1%    3.4%  LazyCompile: *processTicksAndRejections internal/process/task_queues.js:67:35
       2    0.1%    3.4%  LazyCompile: *normalize path.js:299:12
       2    0.1%    3.4%  LazyCompile: *next C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\express\lib\router\index.js:176:16
       1    0.0%    1.7%  RegExp: ^[\^_`a-zA-Z\-0-9!#$%&'*+.|~]+$
       1    0.0%    1.7%  LazyCompile: *toNamespacedPath path.js:609:19
       1    0.0%    1.7%  LazyCompile: *stringifyFnReplacer C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\safe-stable-stringify\index.js:203:32
       1    0.0%    1.7%  LazyCompile: *send C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\express\lib\response.js:107:25
       1    0.0%    1.7%  LazyCompile: *resOnFinish _http_server.js:761:21
       1    0.0%    1.7%  LazyCompile: *readSync fs.js:577:18
       1    0.0%    1.7%  LazyCompile: *pipe C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\send\index.js:510:43
       1    0.0%    1.7%  LazyCompile: *parserOnHeadersComplete _http_common.js:75:33
       1    0.0%    1.7%  LazyCompile: *ondata C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\winston\node_modules\readable-stream\lib\_stream_readable.j
       1    0.0%    1.7%  LazyCompile: *lookup C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\express\lib\view.js:104:40
       1    0.0%    1.7%  LazyCompile: *getStatsFromBinding internal/fs/utils.js:495:29
       1    0.0%    1.7%  LazyCompile: *getHeader _http_outgoing.js:575:57
       1    0.0%    1.7%  LazyCompile: *getDefaultTriggerAsyncId internal/async_hooks.js:412:34
       1    0.0%    1.7%  LazyCompile: *exports.shallowCopyFromList C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\ejs\lib\utils.js:135:40
       1    0.0%    1.7%  LazyCompile: *exports.renderFile C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\ejs\lib\ejs.js:439:31
       1    0.0%    1.7%  LazyCompile: *etag C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\etag\index.js:70:15
       1    0.0%    1.7%  LazyCompile: *endReadableNT internal/streams/readable.js:1310:23
       1    0.0%    1.7%  LazyCompile: *compile C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\ejs\lib\ejs.js:569:21
       1    0.0%    1.7%  LazyCompile: *authenticate C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\passport\lib\middleware\authenticate.js:94:31
       1    0.0%    1.7%  LazyCompile: *allocate buffer.js:407:18
       1    0.0%    1.7%  LazyCompile: *afterWrite C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\winston-transport\node_modules\readable-stream\lib\_str
       1    0.0%    1.7%  LazyCompile: *_transform C:\AppServ\www\curso-programacion-backend-coderhouse\node_modules\winston\lib\winston\logger.js:275:13
       1    0.0%    1.7%  LazyCompile: *Readable.read internal/streams/readable.js:371:35
       1    0.0%    1.7%  LazyCompile: *<anonymous> C:\AppServ\www\curso-programacion-backend-coderhouse\src\routers\root.routes.js:12:25
       1    0.0%    1.7%  LazyCompile: *<anonymous> :1:20
```

2) Node.js Inspect.

```
Yonathan Moncada@DESKTOP-C5A5PF7 MINGW64 /c/AppServ/www/curso-programacion-backend-coderhouse (main)
$ npm test

> websockets@1.0.0 test C:\AppServ\www\curso-programacion-backend-coderhouse
> node benchmark.js

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


┌─────────┬────────┬────────┬────────┬────────┬──────────┬──────────┬────────┐
│ Stat    │ 2.5%   │ 50%    │ 97.5%  │ 99%    │ Avg      │ Stdev    │ Max    │
├─────────┼────────┼────────┼────────┼────────┼──────────┼──────────┼────────┤
│ Latency │ 100 ms │ 118 ms │ 207 ms │ 231 ms │ 125.18 ms │ 25.76 ms │ 250 ms │
└─────────┴────────┴────────┴────────┴────────┴──────────┴──────────┴────────┘
┌───────────┬─────────┬─────────┬─────────┬─────────┬─────────┬─────────┬─────────┐
│ Stat      │ 1%      │ 2.5%    │ 50%     │ 97.5%   │ Avg     │ Stdev   │ Min     │
├───────────┼─────────┼─────────┼─────────┼─────────┼─────────┼─────────┼─────────┤
│ Req/Sec   │ 543     │ 543     │ 823     │ 932     │ 794.25  │ 112.86  │ 543     │
├───────────┼─────────┼─────────┼─────────┼─────────┼─────────┼─────────┼─────────┤
│ Bytes/Sec │ 1.06 MB │ 1.06 MB │ 1.61 MB │ 1.83 MB │ 1.56 MB │ 221 kB  │ 1.06 MB │
└───────────┴─────────┴─────────┴─────────┴─────────┴─────────┴─────────┴─────────┘

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

16k requests in 20.03s, 31.1 MB read
```




```
1       const express = require('express');
2       const compression = require('compression');
3       const os = require('os');
4       const logger = require('../utils/logger.utils');
5
6       const rootRouter = express.Router();
7
8       rootRouter.get('/', function(req, res) {
9           res.redirect('/app/auth/login');
10      });
11
12  0.7 ms rootRouter.get('/info', (req, res) => {
13  1.6 ms     const info = {
14  3.8 ms         inputArguments: process.argv.slice(2),
15  4.4 ms         cpus: os.cpus().length,
16  3.9 ms         platformName: process.platform,
17  0.7 ms         nodeJsVersion: process.version,
18  1.9 ms         reservedTotalMemory: process.memoryUsage().rss,
19  0.8 ms         executionPath: process.execPath,
20  0.6 ms         processId: process.pid,
21  8.4 ms         projectFolder: process.cwd()
22      };
23
24          /* console.log(info); */
25
26 45.3 ms     res.render('pages/info.ejs', { info });
27  0.3 ms });
28
29      rootRouter.get('/info-compression', compression(), (req, res) => {
30          res.render('pages/info.ejs', { process: process, cwd: process.cwd(), rss: process.memoryUsage().rss, argv: process.argv.slice(2), cpus: os.cpus().length });
31      });
32
33      rootRouter.all('*', (req, res) => {
34          logger.write('warn', `La ruta [${req.method}] ${req.protocol + '://' + req.get('host') + req.originalUrl} es inexistente en el servidor.`);
35          res.json({status: false});
36      });
37
38      module.exports = rootRouter;
```

3) 0x.

Conclusión

Podemos notar en los resultados, que el console.log() implica que el proceso tenga un mayor consumo o una mayor carga, ya que esté es considerado un procedimiento bloqueante.