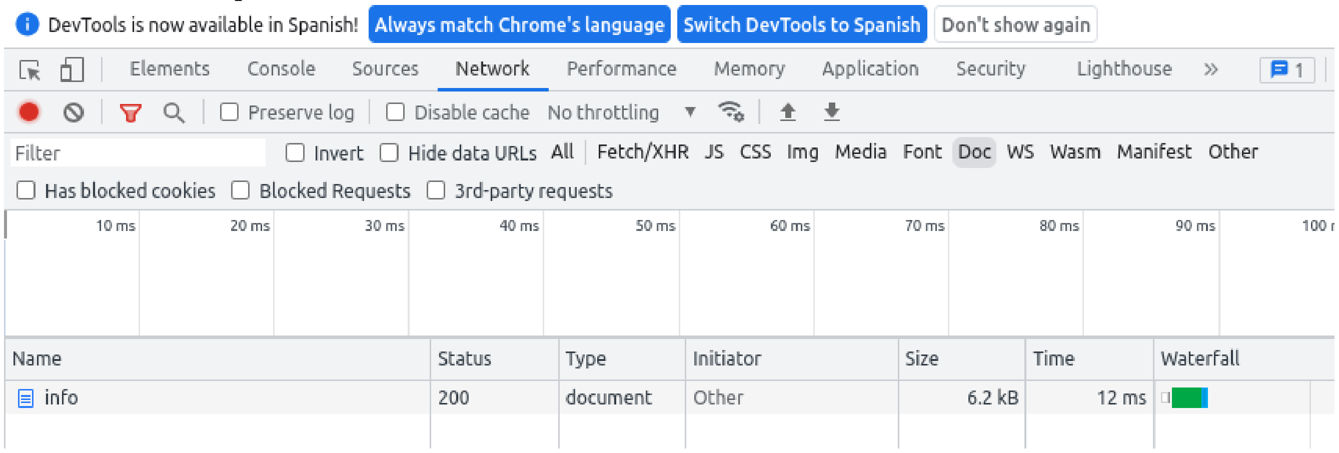
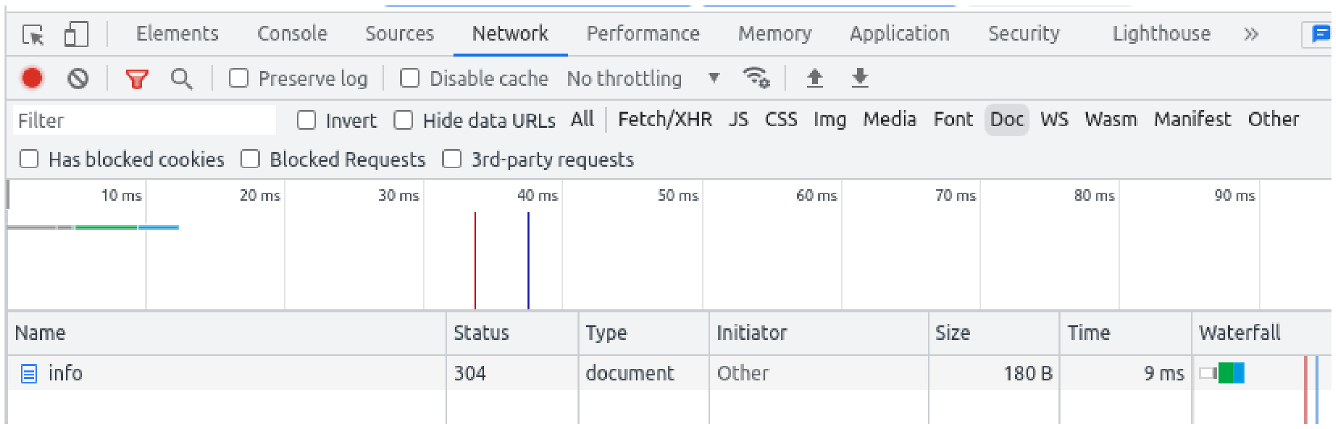
Usando GZIP

Ruta Info sin compresión:



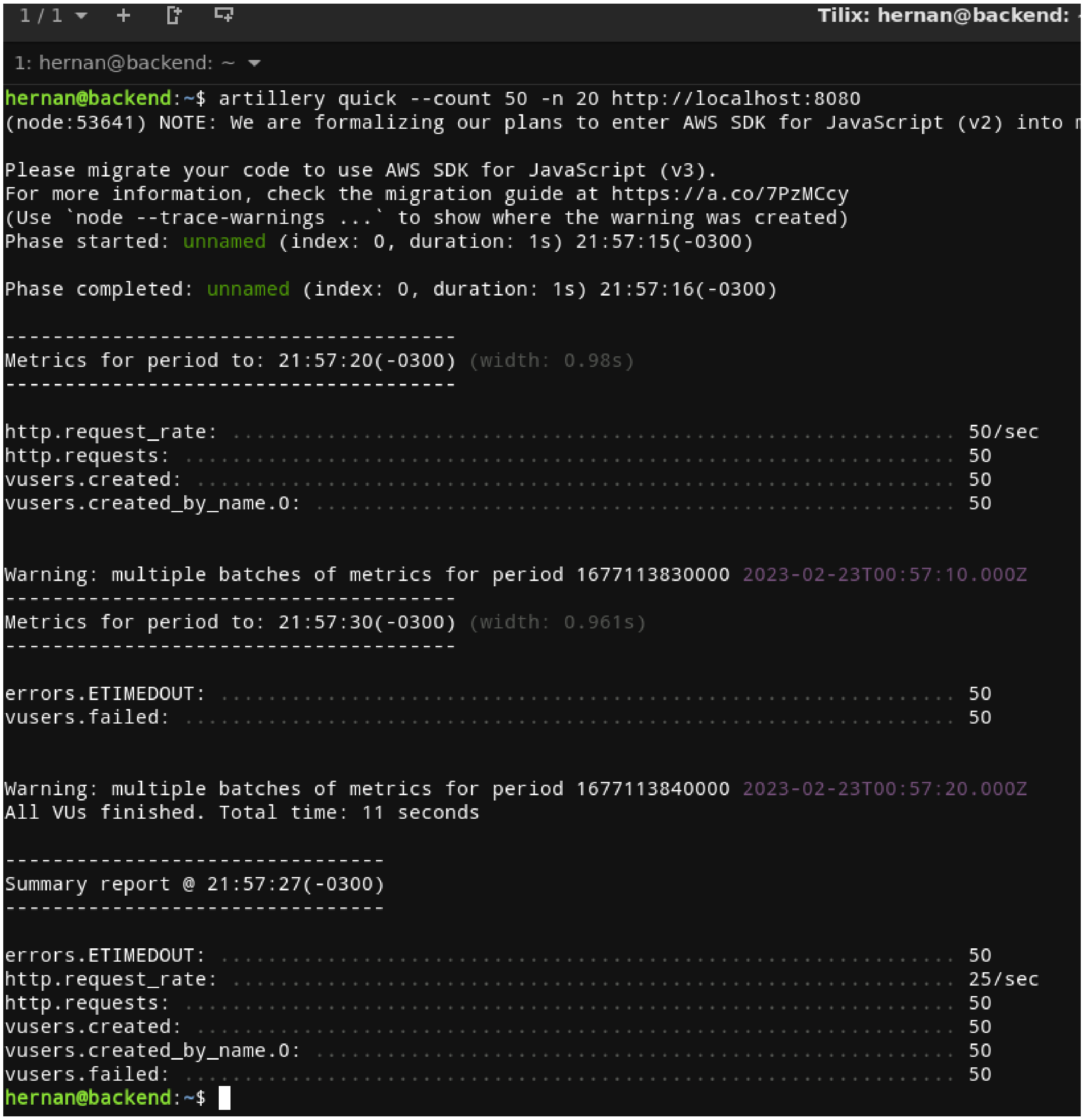
Name	Status	Type	Initiator	Size	Time	Waterfall
info	200	document	Other	6.2 kB	12 ms	



Name	Status	Type	Initiator	Size	Time	Waterfall
info	304	document	Other	180 B	9 ms	

Usando compresión bajó desde los 6.2 KB a los 180 B.

<u>Test de carga 50 conexiones con 20 request por cada una</u>



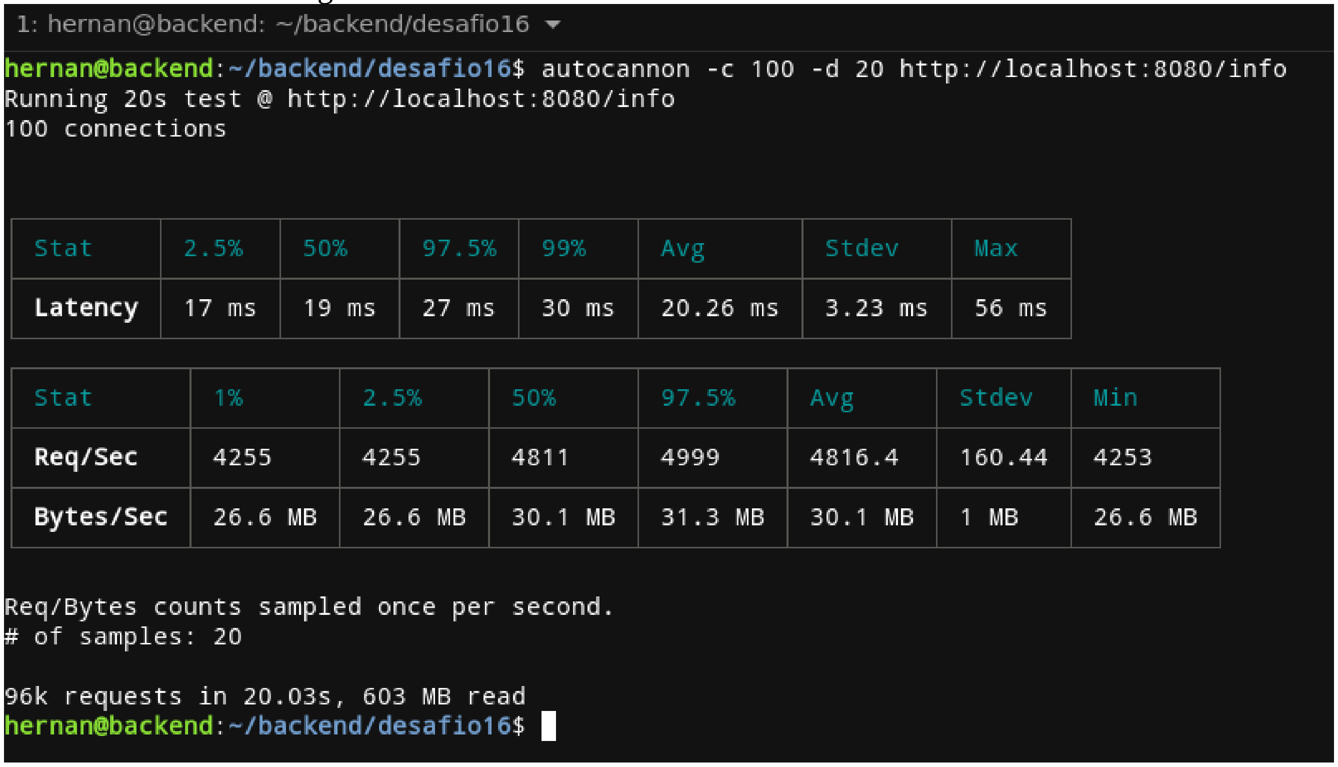
1 / 1 Tilix: hernan@backend:

1: hernan@backend: ~

```
hernan@backend:~$ artillery quick --count 50 -n 20 http://localhost:8080
(node:53641) NOTE: We are formalizing our plans to enter AWS SDK for JavaScript (v2) into m

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the migration guide at https://a.co/7PzMCcy
(Use `node --trace-warnings ...` to show where the warning was created)
Phase started: unnamed (index: 0, duration: 1s) 21:57:15(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 21:57:16(-0300)

--------------------------------------
Metrics for period to: 21:57:20(-0300) (width: 0.98s)
--------------------------------------

http.request_rate: ............................................................. 50/sec
http.requests: ................................................................. 50
vusers.created: ................................................................ 50
vusers.created_by_name.0: ...................................................... 50


Warning: multiple batches of metrics for period 1677113830000 2023-02-23T00:57:10.000Z
--------------------------------------
Metrics for period to: 21:57:30(-0300) (width: 0.961s)
--------------------------------------

errors.ETIMEDOUT: .............................................................. 50
vusers.failed: ................................................................. 50


Warning: multiple batches of metrics for period 1677113840000 2023-02-23T00:57:20.000Z
All VUs finished. Total time: 11 seconds

--------------------------------
Summary report @ 21:57:27(-0300)
--------------------------------

errors.ETIMEDOUT: .............................................................. 50
http.request_rate: ............................................................. 25/sec
http.requests: ................................................................. 50
vusers.created: ................................................................ 50
vusers.created_by_name.0: ...................................................... 50
vusers.failed: ................................................................. 50
hernan@backend:~$
```

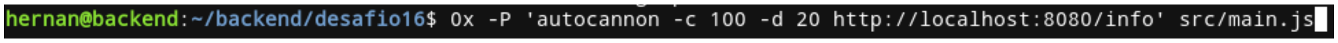
Uso de autocannon:

100 conexiones en 20 segundos:



```
1: hernan@backend: ~/backend/desafio16
hernan@backend:~/backend/desafio16$ autocannon -c 100 -d 20 http://localhost:8080/info
Running 20s test @ http://localhost:8080/info
100 connections


┌─────────┬───────┬───────┬───────┬───────┬──────────┬─────────┬───────┐
│ Stat    │ 2.5%  │ 50%   │ 97.5% │ 99%   │ Avg      │ Stdev   │ Max   │
├─────────┼───────┼───────┼───────┼───────┼──────────┼─────────┼───────┤
│ Latency │ 17 ms │ 19 ms │ 27 ms │ 30 ms │ 20.26 ms │ 3.23 ms │ 56 ms │
└─────────┴───────┴───────┴───────┴───────┴──────────┴─────────┴───────┘
┌───────────┬─────────┬─────────┬─────────┬─────────┬─────────┬────────┬─────────┐
│ Stat      │ 1%      │ 2.5%    │ 50%     │ 97.5%   │ Avg     │ Stdev  │ Min     │
├───────────┼─────────┼─────────┼─────────┼─────────┼─────────┼────────┼─────────┤
│ Req/Sec   │ 4255    │ 4255    │ 4811    │ 4999    │ 4816.4  │ 160.44 │ 4253    │
├───────────┼─────────┼─────────┼─────────┼─────────┼─────────┼────────┼─────────┤
│ Bytes/Sec │ 26.6 MB │ 26.6 MB │ 30.1 MB │ 31.3 MB │ 30.1 MB │ 1 MB   │ 26.6 MB │
└───────────┴─────────┴─────────┴─────────┴─────────┴─────────┴────────┴─────────┘

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

96k requests in 20.03s, 603 MB read
hernan@backend:~/backend/desafio16$
```

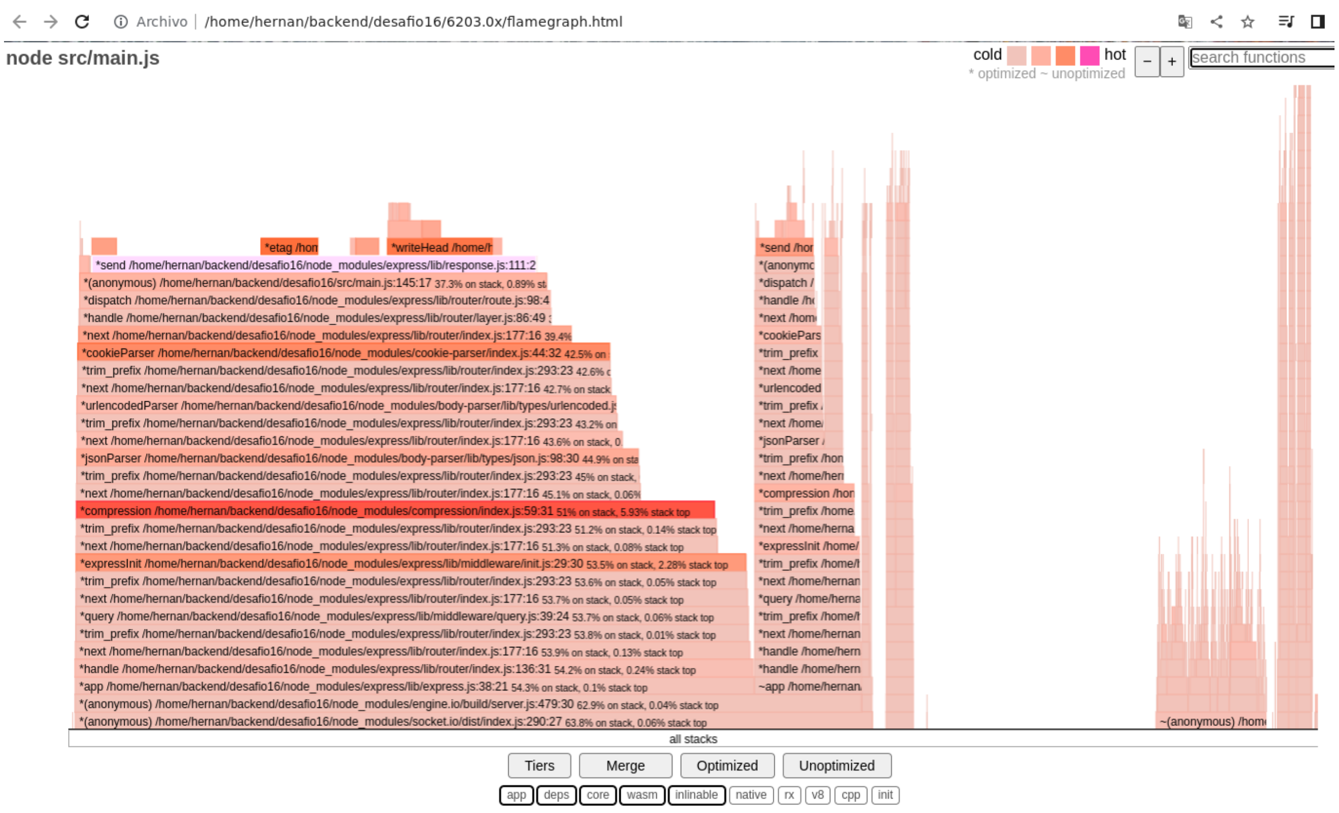
El diagrama de flama con 0x, emulando la carga con Autocannon con los mismos parámetros anteriores



```
hernan@backend:~/backend/desafio16$ 0x -P 'autocannon -c 100 -d 20 http://localhost:8080/info' src/main.js
```

Comando para ejecutar x0 y autocanon desde la linea de comandos: Fuente https://morioh.com/p/d250a81c79e9

0x -P 'autocannon -c 100 -d 20 http://localhost:8080/info' src/main.js



```
*etag /hon
*writeHead /home/h
*send /home/hernan/backend/desafio16/node_modules/express/lib/response.js:111:2
*(anonymous) /home/hernan/backend/desafio16/src/main.js:145:17 37.3% on stack, 0.89% st
*dispatch /home/hernan/backend/desafio16/node_modules/express/lib/router/route.js:98:4
*handle /home/hernan/backend/desafio16/node_modules/express/lib/router/layer.js:86:49
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 39.4%
*cookieParser /home/hernan/backend/desafio16/node_modules/cookie-parser/index.js:44:32 42.5% on
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 42.6%
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 42.7% on stack
*urlencodedParser /home/hernan/backend/desafio16/node_modules/body-parser/lib/types/urlencoded.js
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 43.2% on
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 43.6% on stack, 0
*jsonParser /home/hernan/backend/desafio16/node_modules/body-parser/lib/types/json.js:98:30 44.9% on sta
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 45% on stack,
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 45.1% on stack, 0.06%
*compression /home/hernan/backend/desafio16/node_modules/compression/index.js:59:31 51% on stack, 5.93% stack top
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 51.2% on stack, 0.14% stack top
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 51.3% on stack, 0.08% stack top
*expressInit /home/hernan/backend/desafio16/node_modules/express/lib/middleware/init.js:29:30 53.5% on stack, 2.28% stack top
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 53.6% on stack, 0.05% stack top
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 53.7% on stack, 0.05% stack top
*query /home/hernan/backend/desafio16/node_modules/express/lib/middleware/query.js:39:24 53.7% on stack, 0.06% stack top
*trim_prefix /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:293:23 53.8% on stack, 0.01% stack top
*next /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:177:16 53.9% on stack, 0.13% stack top
*handle /home/hernan/backend/desafio16/node_modules/express/lib/router/index.js:136:31 54.2% on stack, 0.24% stack top
*app /home/hernan/backend/desafio16/node_modules/express/lib/express.js:38:21 54.3% on stack, 0.1% stack top
*(anonymous) /home/hernan/backend/desafio16/node_modules/engine.io/build/server.js:479:30 62.9% on stack, 0.04% stack top
*(anonymous) /home/hernan/backend/desafio16/node_modules/socket.io/dist/index.js:290:27 63.8% on stack, 0.06% stack top
```

```
*send /hor
*(anonymo
*dispatch /
*handle /h
*next /hom
*cookiePars
*trim_prefix
*next /home
*urlencoded
*trim_prefix
*next /home
*jsonParser
*trim_prefix /hon
*next /home/her
*compression /hor
*trim_prefix /home
*next /home/herna
*expressInit /home/
*trim_prefix /home/
*next /home/hernan
*query /home/herna
*trim_prefix /home/
*next /home/hernan
*handle /home/hern
*handle /home/hern
~app /home/hernan
~(anonymous) /hom
```

Podemos concluir que la librería socket.io es la que mas demanda a la aplicación.