Artillery report sin console.log:

```
--------------------------------
Summary report @ 20:35:39(-0300)
--------------------------------

http.codes.200: ..................................................... 1000
http.request_rate: ..................................................... 20/sec
http.requests: ..................................................... 1000
http.response_time:
  min: ..................................................... 3
  max: ..................................................... 1499
  median: ..................................................... 61
  p95: ..................................................... 804.5
  p99: ..................................................... 1224.4
http.responses: ..................................................... 1000
vusers.completed: ..................................................... 20
vusers.created: ..................................................... 20
vusers.created_by_name.0: ..................................................... 20
vusers.failed: ..................................................... 0
vusers.session_length:
  min: ..................................................... 21837.2
  max: ..................................................... 24510
  median: ..................................................... 22703.7
  p95: ..................................................... 23630.3
  p99: ..................................................... 23630.3
```

Artillery report con console.log:

```
--------------------------------
Summary report @ 20:38:33(-0300)
--------------------------------

http.codes.200: ..................................................... 1000
http.request_rate: ..................................................... 18/sec
http.requests: ..................................................... 1000
http.response_time:
  min: ..................................................... 4
  max: ..................................................... 1694
  median: ..................................................... 63.4
  p95: ..................................................... 837.3
  p99: ..................................................... 1224.4
http.responses: ..................................................... 1000
vusers.completed: ..................................................... 20
vusers.created: ..................................................... 20
vusers.created_by_name.0: ..................................................... 20
vusers.failed: ..................................................... 0
vusers.session_length:
  min: ..................................................... 22285.3
  max: ..................................................... 26005.4
```

median: 24107.7
p95: 25598.5
p99: 25598.5

Conclusión Artillery report: Se ve que la media agregando el console.log es más grande, por lo que agregarlo hace que precise más tiempo.

Profile con console:

[Summary]:

ticks	total	nonlib	name
551	7.2%	23.5%	JavaScript
1785	23.3%	76.3%	C++
300	3.9%	12.8%	GC
5335	69.5%		Shared libraries
4	0.1%		Unaccounted

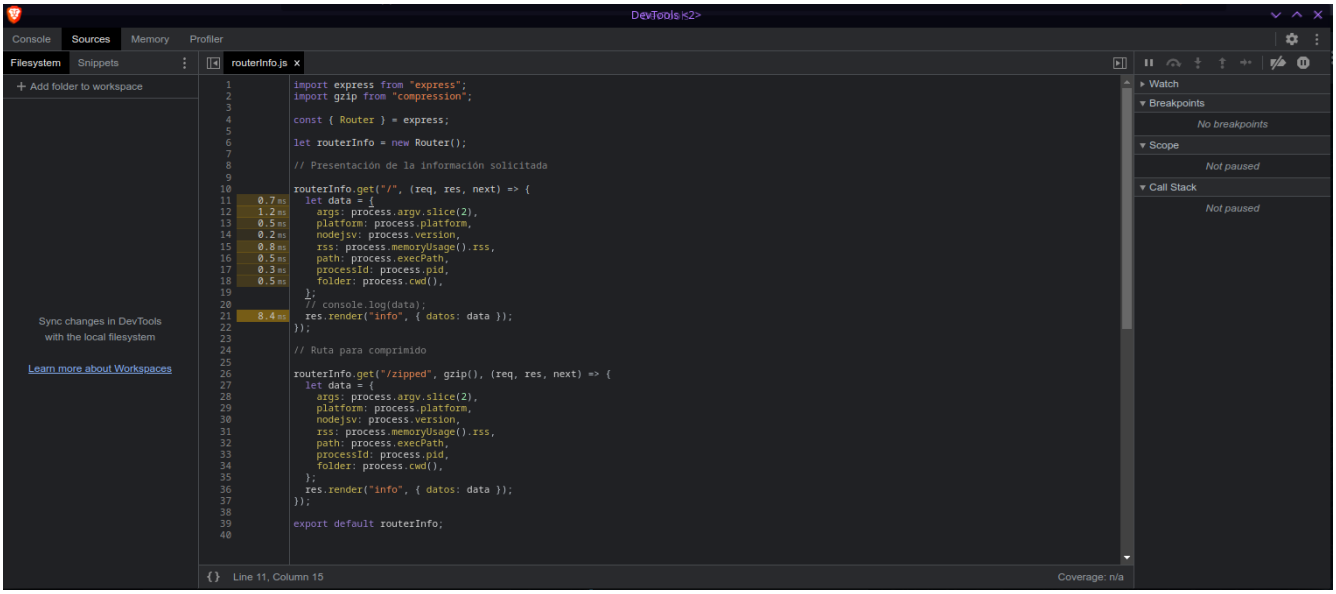
Profile sin console:

[Summary]:

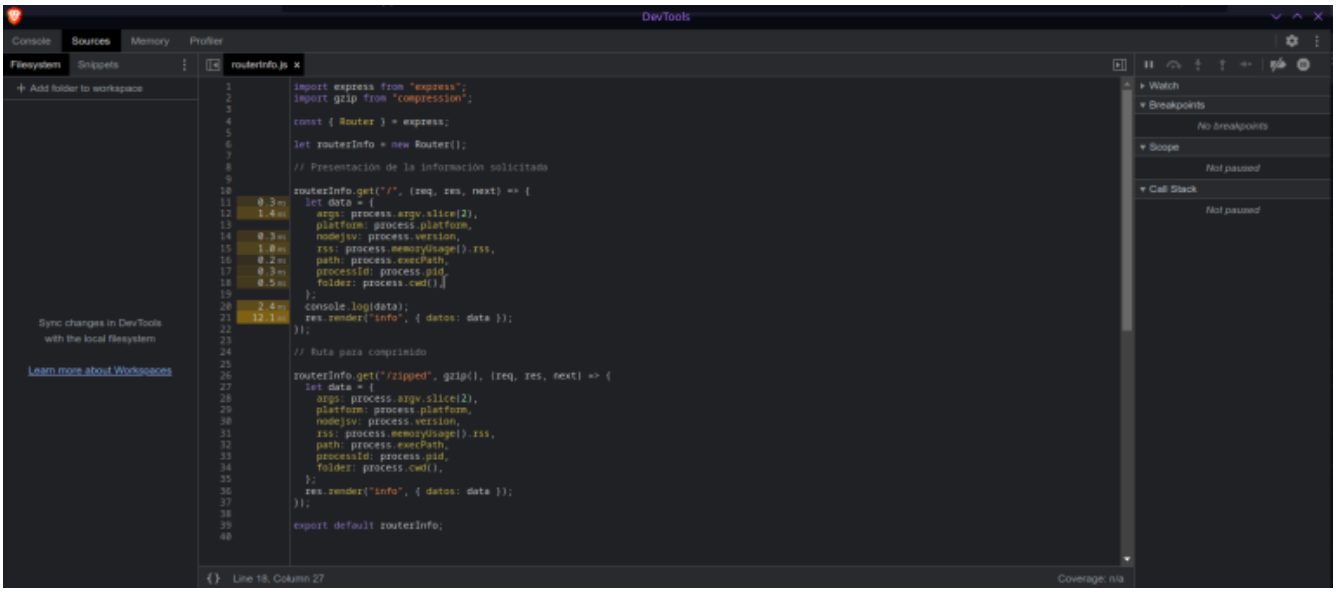
ticks	total	nonlib	name
536	7.6%	25.7%	JavaScript
1547	21.9%	74.1%	C++
317	4.5%	15.2%	GC
4990	70.5%		Shared libraries
6	0.1%		Unaccounted

Conclusión profile: No usar el console.log hace que se lleven menos ticks, el proceso es más rápido.

Inspect sin console.log:



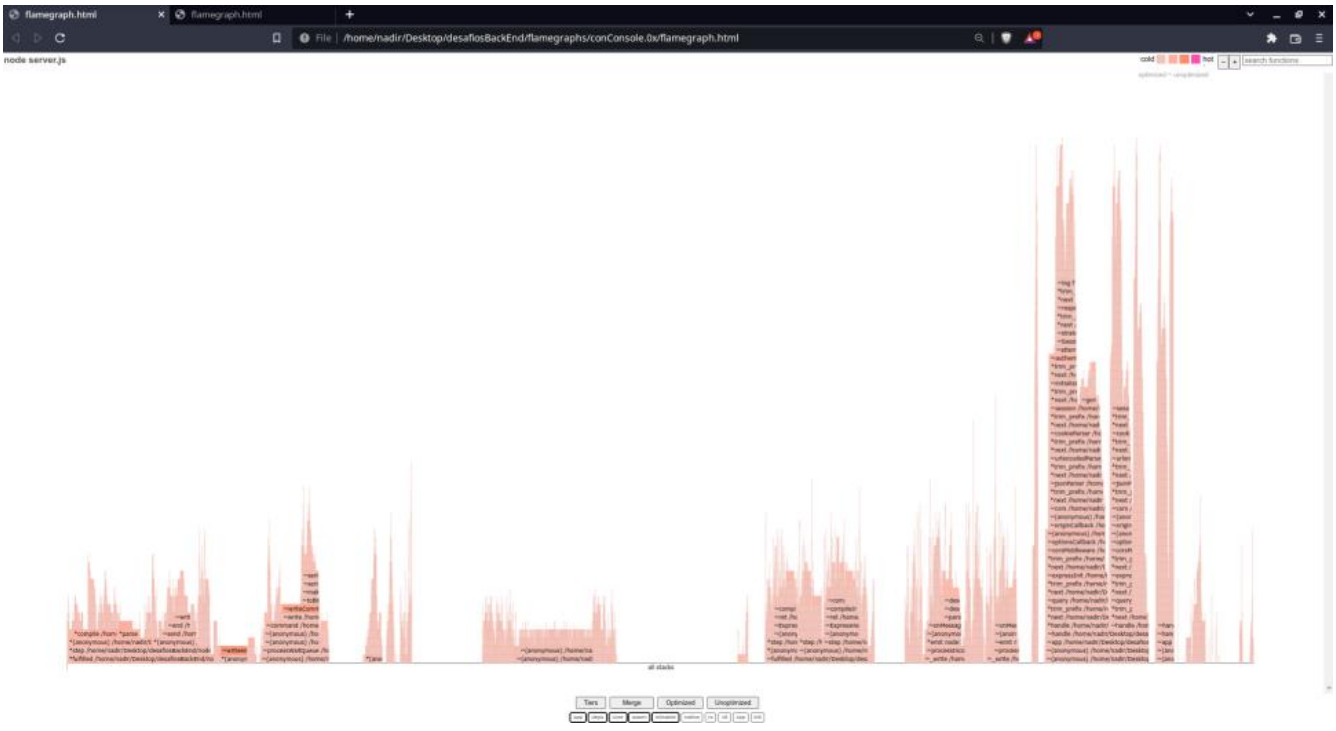
Inspect con console.log:



Conclusión inspect: se ve que con console.log lleva más tiempo

Flamegraph sin console:

Flamegraph con console:



Conclusión flamegraph: Con console hay más picos, por lo que se requieren más procesos, además, los procesos son más anchos, lo que quiere decir que requieren más tiempo.

Conclusión final: Con console.log se puede visualizar que hay más procesos, con más ticks y que además requieren más tiempo.