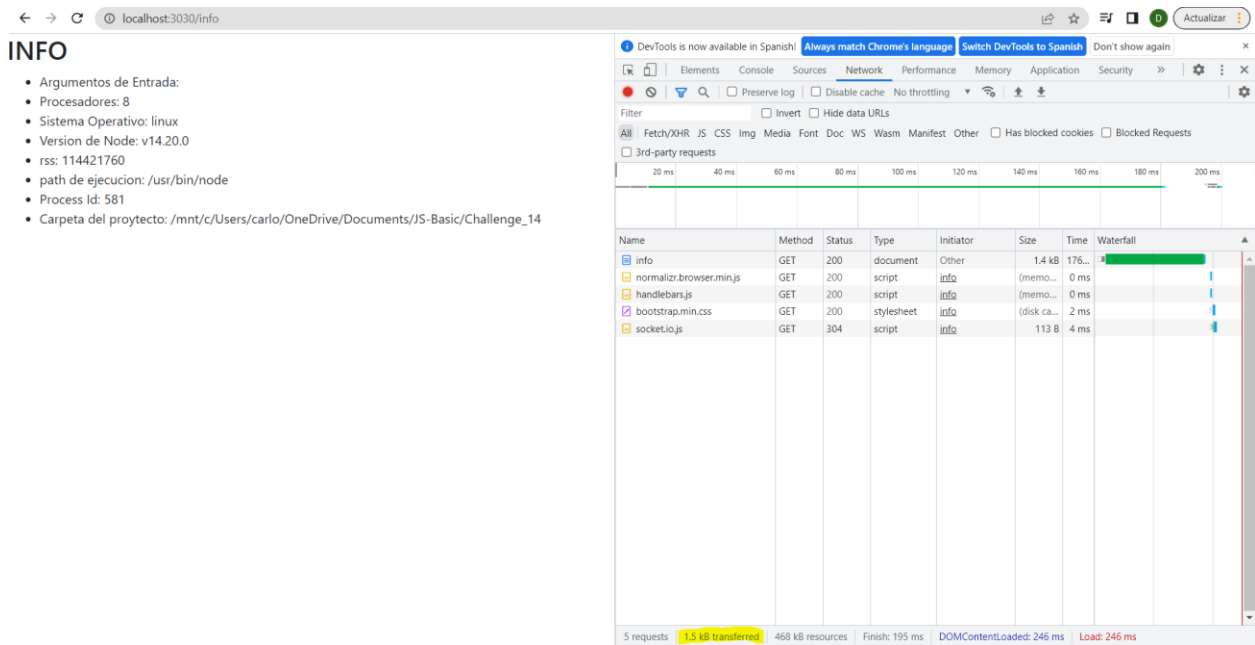
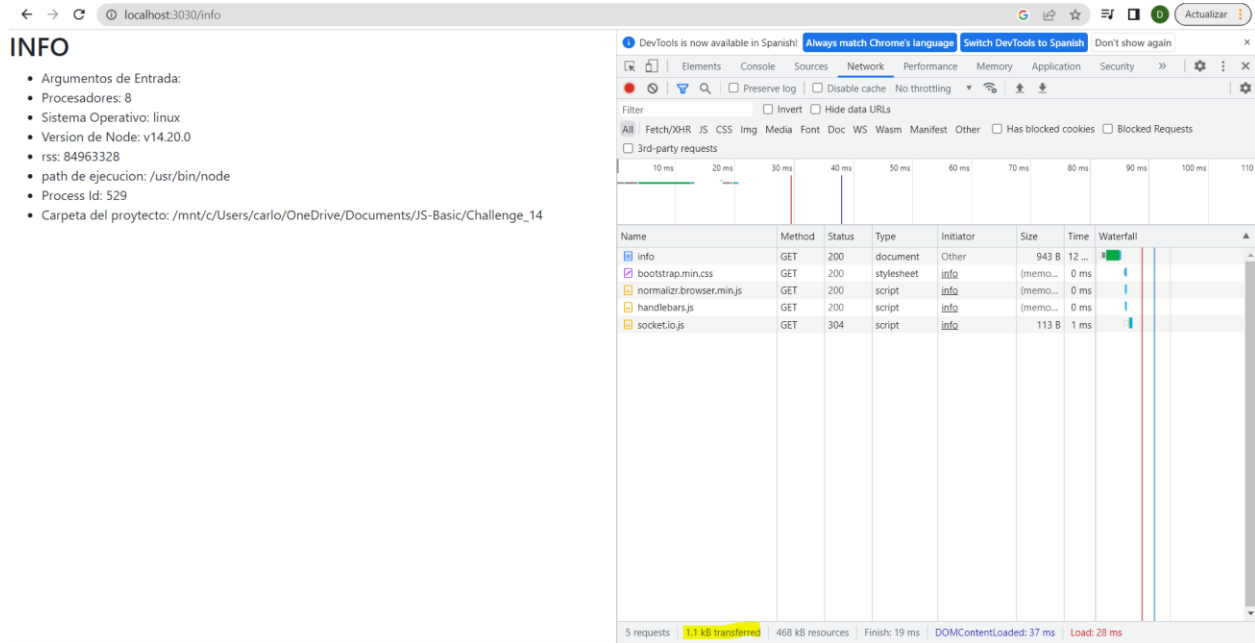


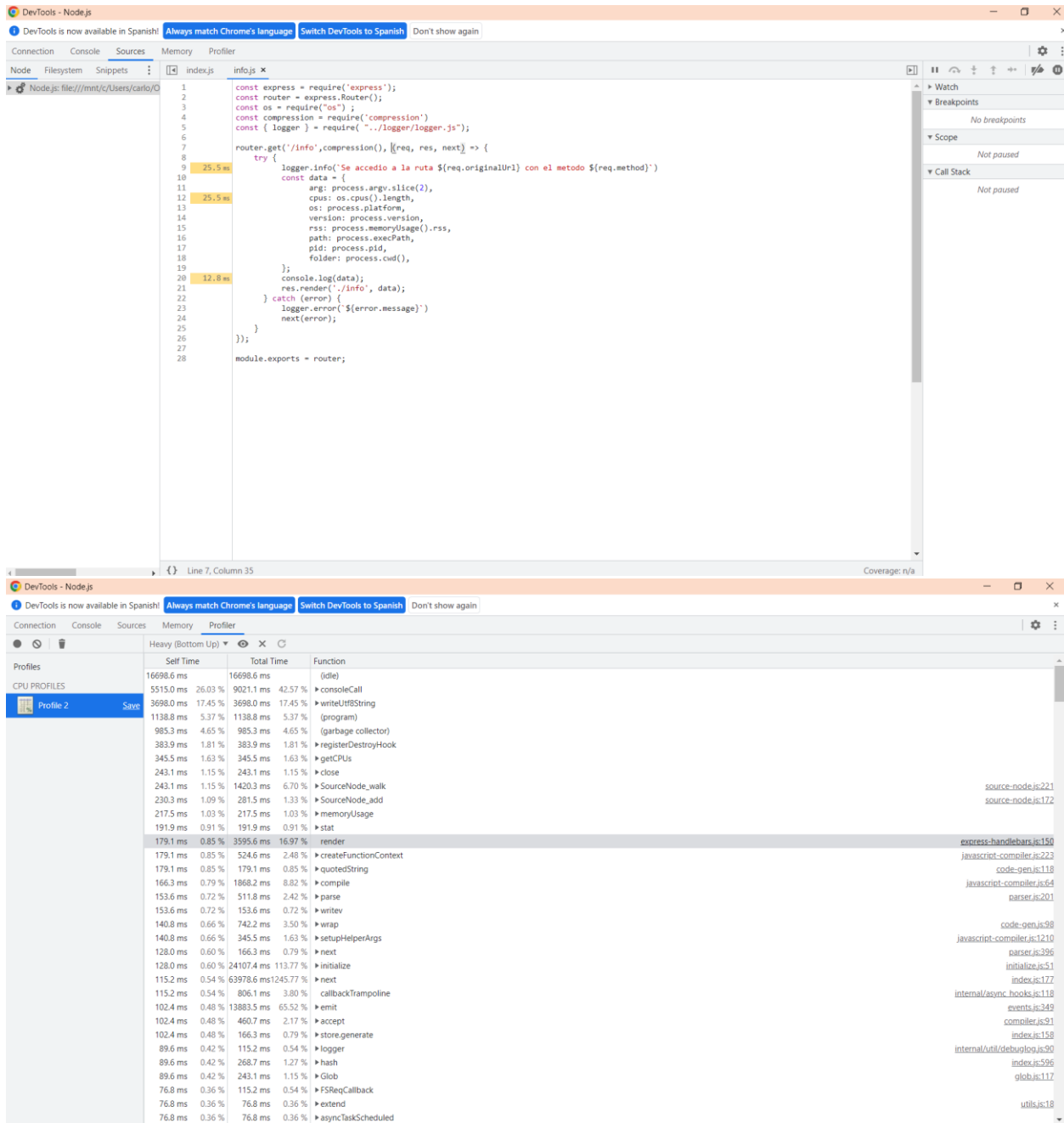
CHALLENGE 14

CODERHOUSE – BACKEND

- Fue verificado sobre la ruta /info la diferencia de bytes devueltos en los casos de compresion y sin compresion con gzip, donde se observa una diferencia de 0.4 kB.



- Se revisaron los tiempos de los procesos como el modo inspector de node.js –inspect, del cual se nota un aumento de 12.8 ms con la utilizacion del console.log en la ruta /info.



- Acontinuacion se muestran las graficas e informacion generada con 0x emulando la carga con autocannon:

Challenge_14 [main] npm run test

```
> challenge_14@1.0.0 test /mnt/c/Users/carlo/OneDrive/Documents/JS-Basic/Challenge_14
> node benchmark.js
```

Runnig all benchmarks in parallel...
Running 20s test @ http://localhost:3030/info
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	1731 ms	5453 ms	6554 ms	7355 ms	5145.5 ms	1151.9 ms	7672 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	147	185	139	42.13	100
Bytes/Sec	0 B	0 B	212 kB	268 kB	201 kB	60.9 kB	145 kB

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

4k requests in 20.14s, 4.02 MB read

Challenge_14 [main] npm run test

```
> challenge_14@1.0.0 test /mnt/c/Users/carlo/OneDrive/Documents/JS-Basic/Challenge_14
> node benchmark.js
```

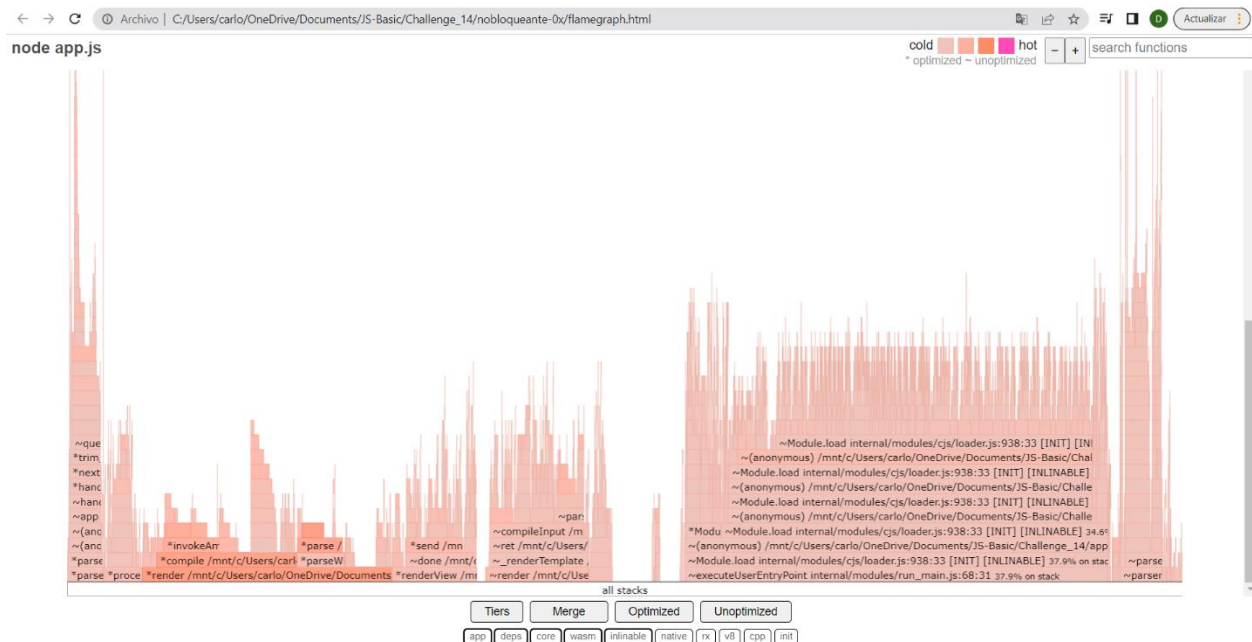
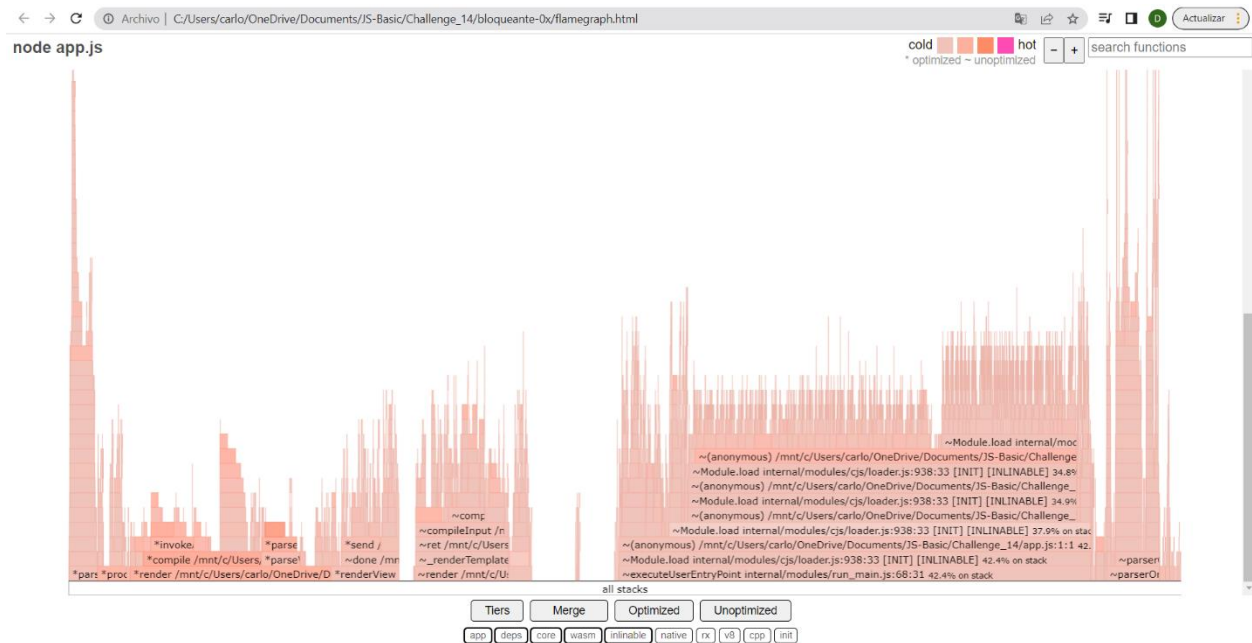
Runnig all benchmarks in parallel...
Running 20s test @ http://localhost:3030/info
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	3273 ms	11881 ms	18318 ms	18542 ms	11512.83 ms	4450.87 ms	18812 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	184	243	165.35	76.55	1
Bytes/Sec	0 B	0 B	266 kB	351 kB	239 kB	111 kB	1.45 kB

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

9k requests in 20.07s, 4.78 MB read
5k errors (0 timeouts)



De lo anterior se puede observar una amplitud en ambos procesos varios picos y una diferencia ligera en la amplitud de algunos procesos. Lo anterior confirma lo critico que es tratar de hacer nuestro código limpio y sin reprocesamientos innecesarios.