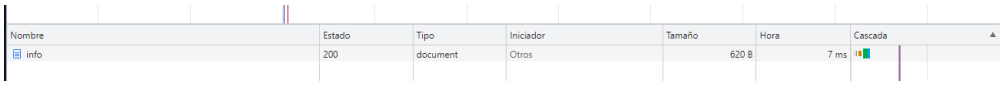
Ruta /info con y sin compresión

Sin compresión

Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Cascada
info	200	document	Otros	9.7 kB	15 ms	

Con compresión



Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Cascada
info	200	document	Otros	620 B	7 ms	

Perfilamiento con –prof (se adjuntan archivos en /performace)

(el console.log() se agrega en la línea 28 de /routes/infoRouter.js)

1

En el profConConsole.txt node.exe ocupa 6.1% de los ticks
En el profSinConsole.txt node.exe ocupa 5.3% de los ticks

En artilleryConConsole.txt se observan 390 req./seg. y respuesta promedio 55.2ms
En artillerySinConsole.txt se observan 674 req./seg y respuesta promedio de 24.8ms

AUTOCANON

Con console.log()

```
Tabare@DESKTOP-1SLHCNS MINGW64 ~/Desktop/Full_Stack/Backend_40845/desafio13 (main)
$ node ./node_modules/autocannon/autocannon -c 100 -d 20 http://localhost:8080
Running 20s test @ http://localhost:8080
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	118 ms	168 ms	204 ms	211 ms	167.28 ms	18.98 ms	257 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	467	467	598	651	595.71	44.06	467
Bytes/Sec	333 kB	333 kB	426 kB	465 kB	425 kB	31.6 kB	333 kB

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

12k requests in 20.09s, 8.5 MB read
```

Sin console.log()

```
Tabare@DESKTOP-1SLHCNS MINGW64 ~/Desktop/Full_Stack/Backend_40845/desafio13 (main)
$ node ./node_modules/autocannon/autocannon -c 100 -d 20 http://localhost:8080/info
Running 20s test @ http://localhost:8080/info
100 connections
```

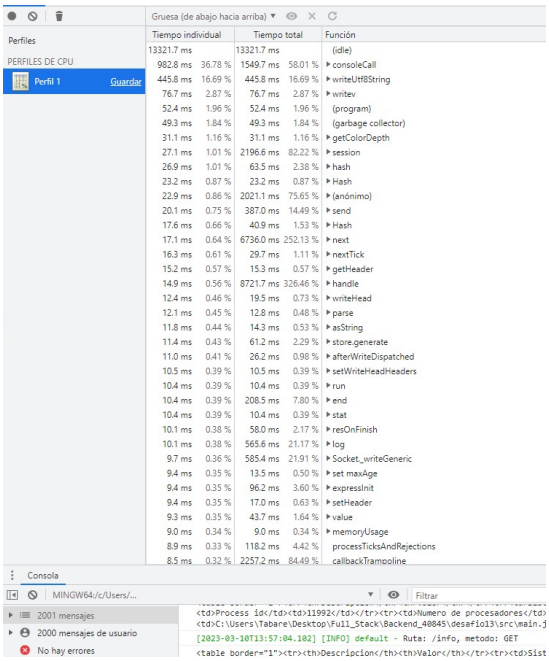
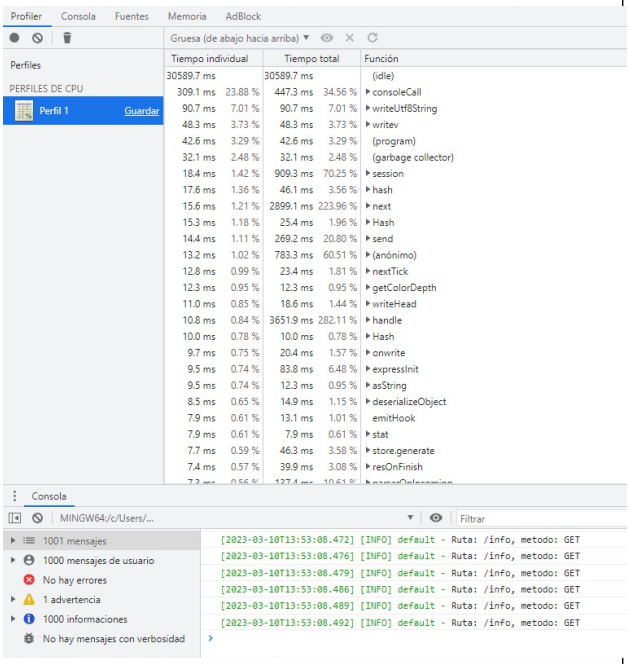
Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	59 ms	65 ms	124 ms	138 ms	74.79 ms	19.96 ms	165 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	921	921	1355	1485	1328.9	148.14	921
Bytes/Sec	657 kB	657 kB	968 kB	1.06 MB	949 kB	106 kB	657 kB

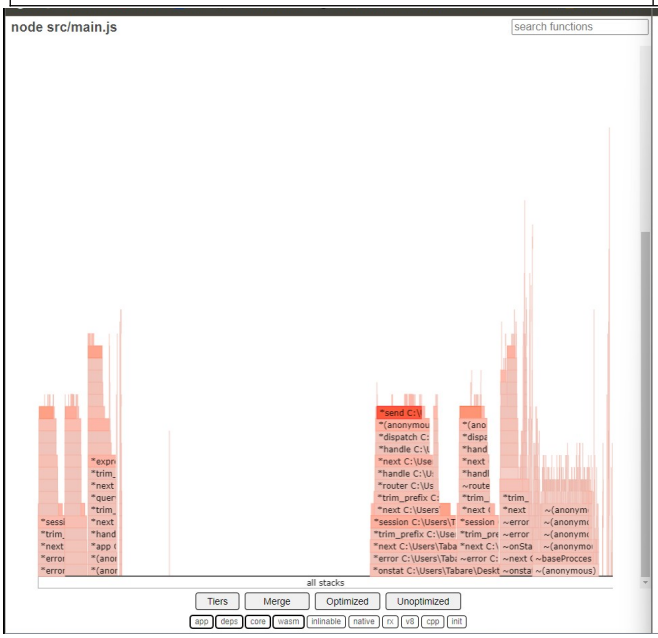
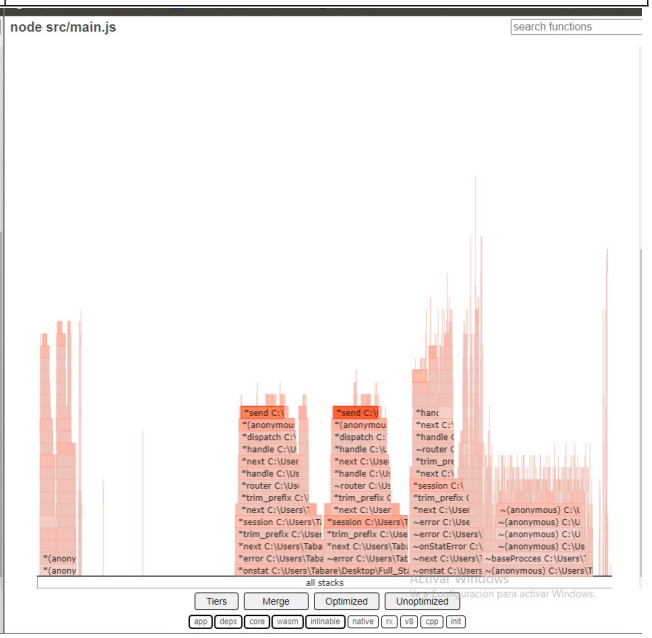
```
Req/Bytes counts sampled once per second.
# of samples: 20

27k requests in 20.08s, 19 MB read
```

2 Autocanon - INSPECT

Inspect	
Con console.log()	Sin console.log()
	

3 Autocanon - 0x

Diagrama flama 0x	
Sin console.log()	Con console.log()
	

CONCLUSIÓN

Después de realizar múltiples pruebas con las herramientas de Artillery y Autocannon, y analizar los resultados con las herramientas –prof, el paquete 0x e inspect en el navegador, se ha podido evidenciar una mejora significativa en el tiempo de respuesta y la cantidad de solicitudes realizadas al quitar la función console.log() de nuestro código. Estos resultados sugieren que la eliminación de esta función puede tener un impacto positivo en el rendimiento de nuestras aplicaciones.