Coder House
Backend, Comision 28855
Bini, Raul Leopoldo

# Pruebas de estrés:

## Esquema general:

- Ruta: “/info”.
- Escenarios:
    A. Incorporando un log por consola en el servidor, de los datos devueltos al cliente en la ruta especificada.
    B. Sin implementar el log por consola en el servidor, de los datos devueltos al cliente en la ruta especificada.
- Pruebas realizadas:
    A. **Test de Carga**
        - Artillery: 20 req  / 50 usuarios.
        - Autocannon**:** 100 req / 20Seg Recurrencia
    B. **Perfilamiento**
        - --Prof de node.js
        - --Inspect de node.js
    **C. Diagrama de Flama ox**

## Test de Carga:

### Artillery:

**Con Console.Log:**

```
Summary report @ 16:47:04(-0300)
--------------------------------
http.codes.200: ..................................................... 1000
http.request_rate: .................................................. 64/sec
http.requests: ...................................................... 1000
http.response_time:
  min: .............................................................. 16
  max: .............................................................. 233
  median: ........................................................... 138.4
  p95: .............................................................. 190.6
  p99: .............................................................. 223.7
http.responses: ..................................................... 1000
vusers.completed: ................................................... 20
vusers.created: ..................................................... 20
vusers.created_by_name.0: ........................................... 20
vusers.failed: ...................................................... 0
vusers.session_length:
  min: .............................................................. 6907.5
  max: .............................................................. 7277.2
  median: ........................................................... 7260.8
  p95: .............................................................. 7260.8
  p99: .............................................................. 7260.8
```

**Sin ConsoleLog**

```
Summary report @ 16:56:06(-0300)
--------------------------------
http.codes.200: ..................................................... 1000
http.request_rate: .................................................. 114/sec
http.requests: ...................................................... 1000
```

http.response_time:
  min: ..... 4
  max: ..... 119
  median: ..... 77.5
  p95: ..... 100.5
  p99: ..... 111.1
http.responses: ..... 1000
vusers.completed: ..... 20
vusers.created: ..... 20
vusers.created_by_name.0: ..... 20
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 3458.8
  max: ..... 3772.2
  median: ..... 3678.4
  p95: ..... 3752.7
  p99: ..... 3752.7

## Autocannon:

**Con ConsoleLog:**

```
leopoldc@LAPTOP-Leopoldo:~/CoderHouse/backend/backend_entregables/clase 16 - Logs debug profiling$ npm run benchmark

> primeraentrega-proyectofinal@1.0.0 benchmark
> node ./benchmark.js

Starting tests...
Running 20s test @ http://localhost:8080/info
100 connections


| Stat | 2.5% | 50% | 97.5% | 99% | Avg | Stdev | Max |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Latency | 267 ms | 299 ms | 498 ms | 796 ms | 319.75 ms | 73.74 ms | 815 ms |

| Stat | 1% | 2.5% | 50% | 97.5% | Avg | Stdev | Min |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Req/Sec | 100 | 100 | 302 | 398 | 309.9 | 65.46 | 100 |
| Bytes/Sec | 214 kB | 214 kB | 646 kB | 851 kB | 663 kB | 140 kB | 214 kB |

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

6k requests in 20.05s, 13.3 MB read
leopoldc@LAPTOP-Leopoldo:~/CoderHouse/backend/backend_entregables/clase 16 - Logs debug profiling$
```

**Sin ConsoleLog:**

```
leopoldc@LAPTOP-Leopoldo:~/CoderHouse/backend/backend_entregables/clase 16 - Logs debug profiling$ npm run benchmark

> primeraentrega-proyectofinal@1.0.0 benchmark
> node ./benchmark.js

Starting tests...
Running 20s test @ http://localhost:8080/info
100 connections


| Stat | 2.5% | 50% | 97.5% | 99% | Avg | Stdev | Max |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Latency | 203 ms | 216 ms | 316 ms | 735 ms | 231.48 ms | 62.12 ms | 766 ms |

| Stat | 1% | 2.5% | 50% | 97.5% | Avg | Stdev | Min |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Req/Sec | 100 | 100 | 438 | 499 | 429.65 | 92.78 | 100 |
| Bytes/Sec | 214 kB | 214 kB | 937 kB | 1.07 MB | 919 kB | 199 kB | 214 kB |

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

9k requests in 20.04s, 18.4 MB read
leopoldc@LAPTOP-Leopoldo:~/CoderHouse/backend/backend_entregables/clase 16 - Logs debug profiling$
```

# Profiling:

## --Prof de Node.js:

**Con ConsoleLog:**

[Summary]:

| ticks | total | nonlib | name |
|---|---|---|---|
| **678** | 8.4% | 33.0% | JavaScript |
| **1360** | 16.9% | 66.3% | C++ |
| **357** | 4.4% | 17.4% | GC |
| **6015** | 74.6% | | Shared libraries |
| **14** | 0.2% | | Unaccounted |

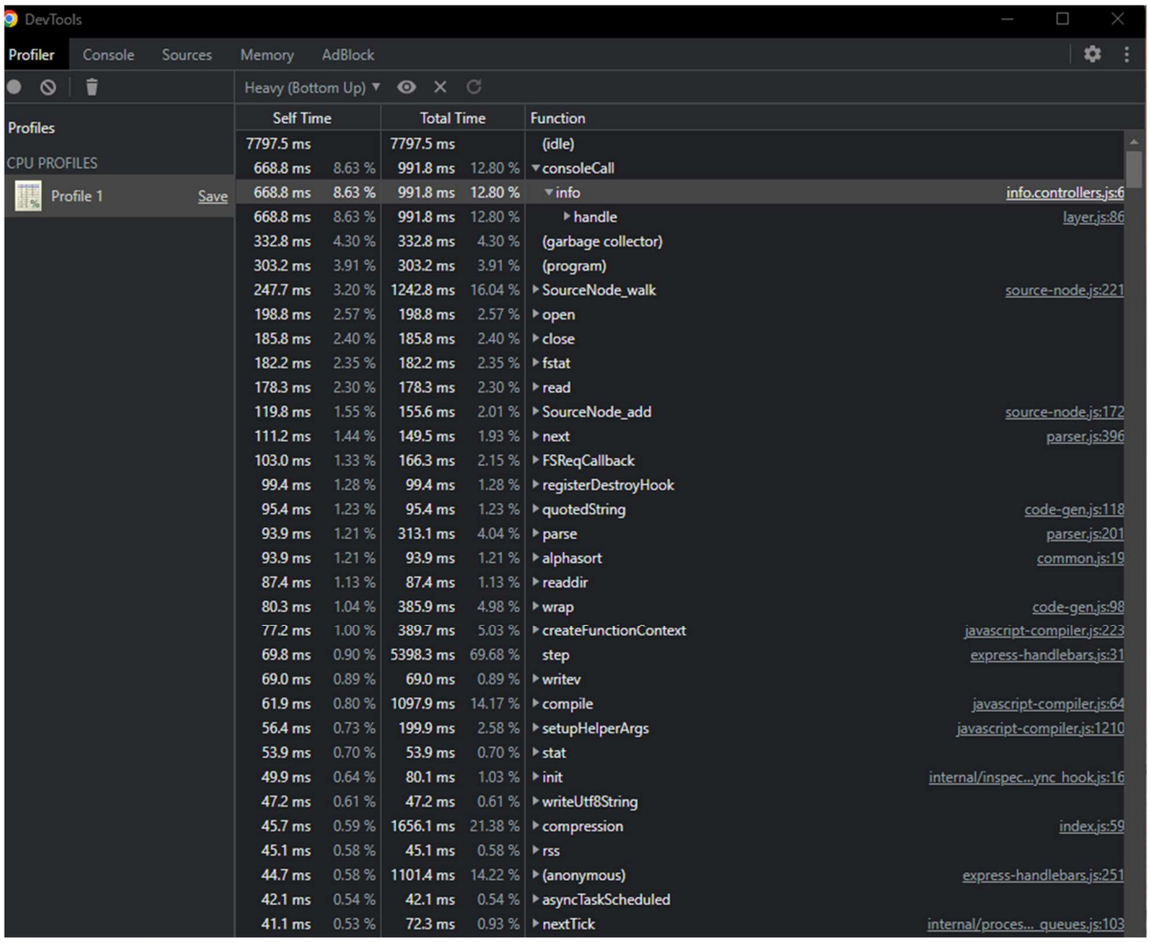
**Sin ConsoleLog:**

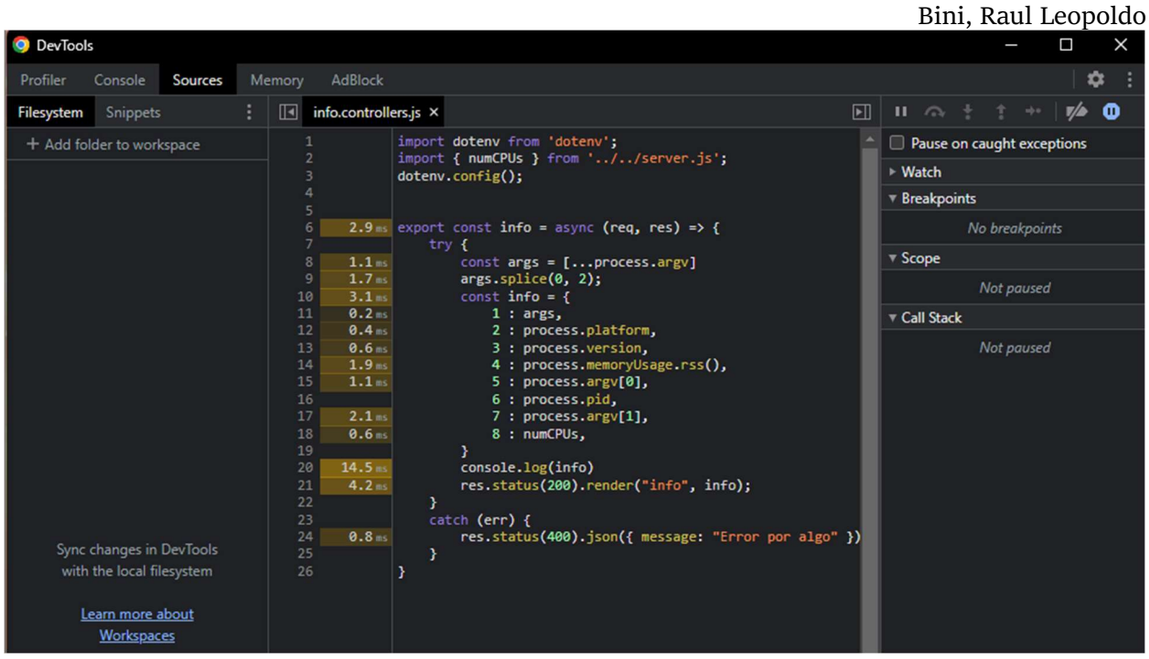
[Summary]:

| ticks | total | nonlib | name |
|---|---|---|---|
| **423** | 8.5% | 38.1% | JavaScript |
| **677** | 13.5% | 61.0% | C++ |
| **145** | 2.9% | 13.1% | GC |
| **3894** | 77.8% | | Shared libraries |
| **10** | 0.2% | | Unaccounted |

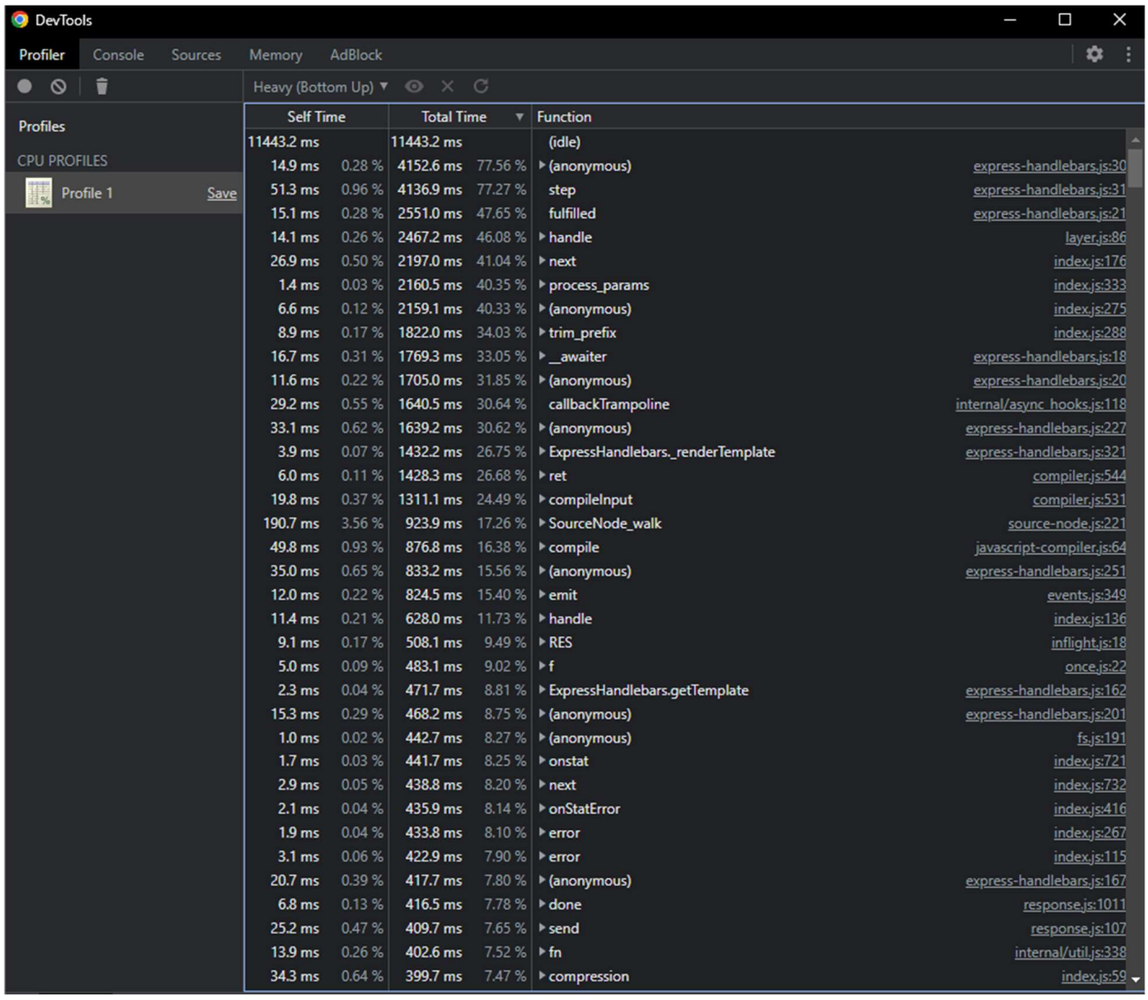
## --Inspect de Node.js:

**Con ConsoleLog:**






**Sin ConsoleLog:**



## Diagrama de Flamas:

Los diagramas de Flamas para ambos escenarios estan en la carpeta “Diagrama de Flamas” , dentro del directorio “zPruebasClase16” del repositorio de GitHub de la entrega.

## Conclusiones:

La conclusión evidente que podemos sacar de los testeos realizados es que nunca hay que olvidarse logs de consola en el código de ningún proyecto. Te destruyen la performance.

En los las distintas pruebas hechas, la diferencia que se puede notar en rendimiento es de 2 a 1  a favor del escenario en donde se prescinde de los console.logs.