Análisis de rendimiento:

Node profiler:

```
$> node --prof server.js
```

Inserción de requests con Artillery:

```
$> artillery quick --count 20 -n 50 'http://localhost:8080/info' >
result info prof.txt
```

Archivo result_info_prof.txt fue adjuntado al repositorio:

Se observa que todas las respuestas fueron satisfactorias, y que el tiempo medio de respuesta fue de 141,2

```
      Summary report @ 20:26:46(-0300)

      http.codes.200:
      1000

      http.request_rate:
      26/sec

      http.response_time:
      11

      max:
      926

      median:
      141.2

      pys:
      045.0

      p99:
      685.5

      http.responses:
      1000

      vusers.completed:
      20

      vusers.created:
      20

      vusers.reated:
      20

      vusers.session_length:
      0

      min:
      18621.5

      max:
      20305.7

      median:
      19737.6

      p95:
      20136.3

      p99:
      20136.3
```

A continuación se decodificaron el archivo de logs del profiler:

\$> node --prof-process isolate-0000023E169545A0-18216-v8.log > result_prof_process_info.txt

El archive *result_prof_process_info.txt* Se adjunta en el repositorio.

```
[Summary]:

ticks total nonlib name

58 0.4% 96.7% JavaScript

0 0.0% 0.0% C++

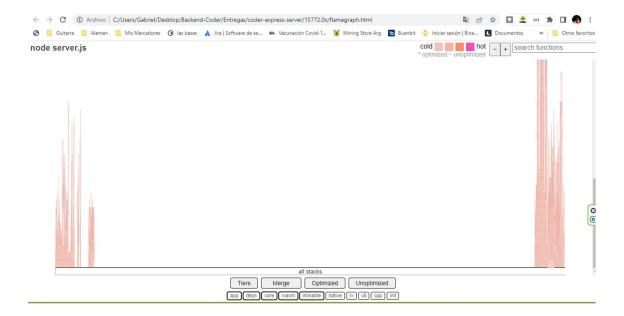
42 0.3% 70.0% GC

16185 99.6% Shared libraries

2 0.0% Unaccounted
```

Autocannon:

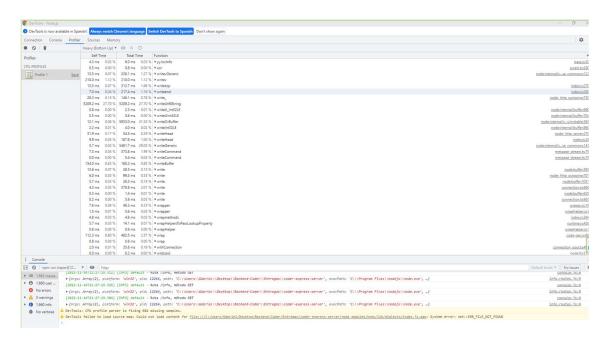
Grafico de Flama:



Se adjunta en el repositorio la carpeta 15772.0x con el gráfico de flama.

En el grafico se observan en forma muy clara 2 llamadas al benchmark y en cada una de ellas los picos de ejecución.

Node inspect:



En el grafico se muestras las operaciones más pesadas.