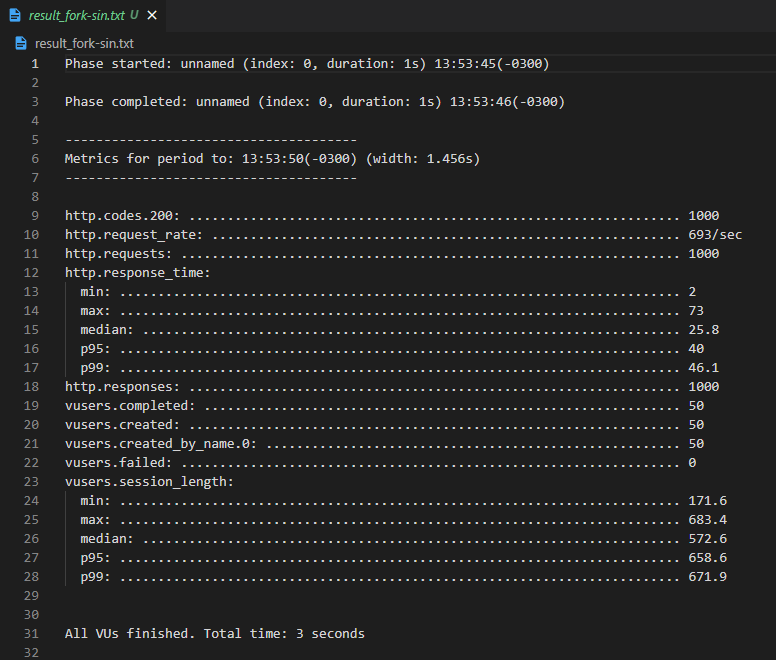
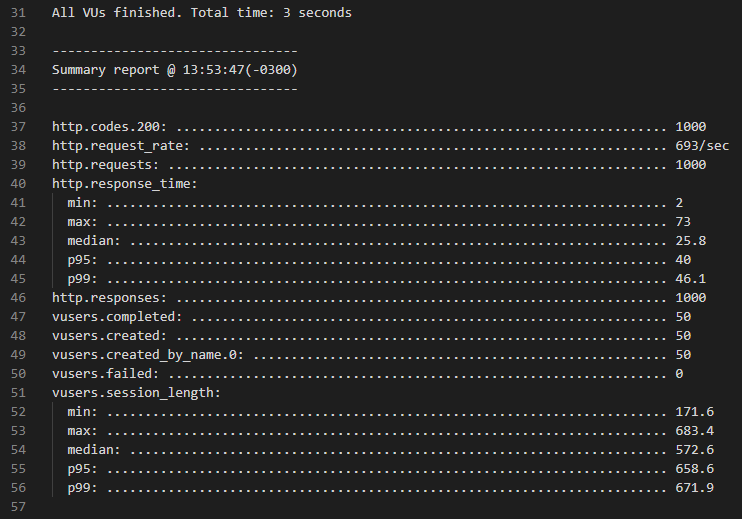
**Informe**

El primer test que realizamos fue con *artillery*, emulando un total de 50 conexiones con 20 *requests* en cada una de ellas. Lo hice sobre un *endpoint* que contiene (o no) un *console log* demandante, para ver la diferencia entre las respuestas. Para tal fin, utilicé el siguiente comando:

artillery quick --count 50 -n 20 http://localhost:8081/info-sin > result\_fork-sin.txt

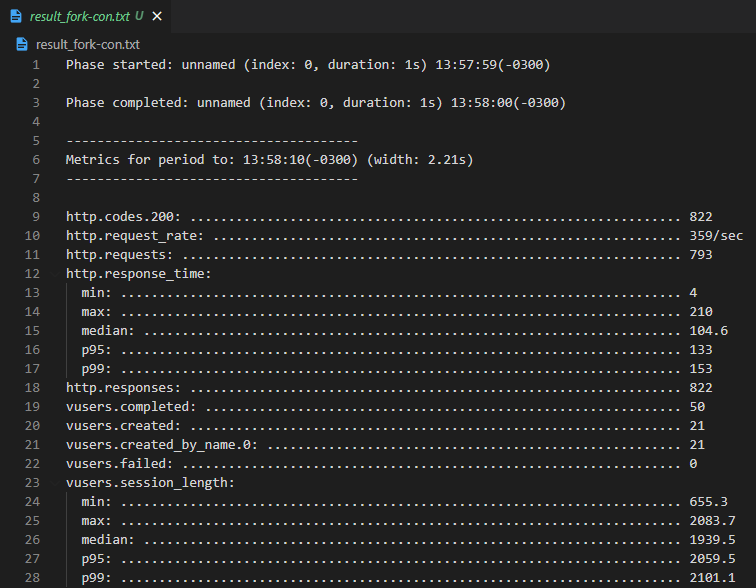
El resultado sobre la ruta sin console log se ve en las siguientes dos capturas.

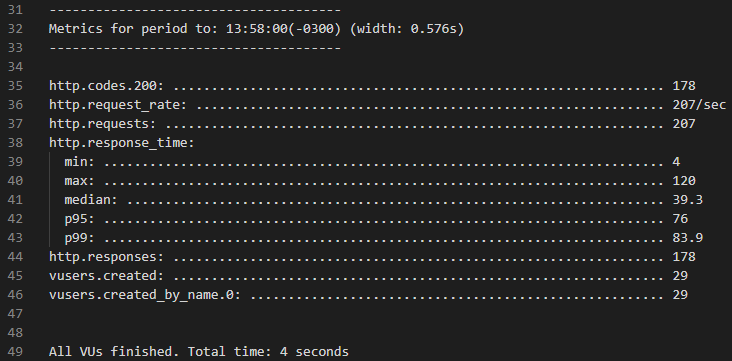


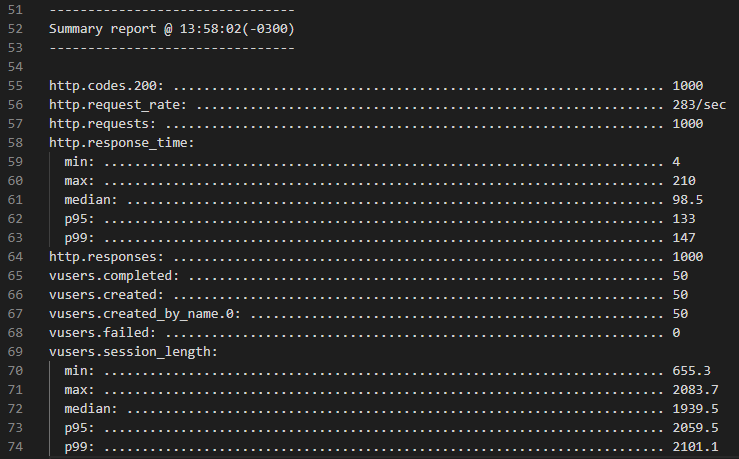


Luego repetí el proceso para la ruta que sí tiene el console log en su flujo. El resultado se ve en las siguientes tres capturas.

artillery quick --count 50 -n 20 http://localhost:8081/info-con > result\_fork-con.txt







Si comparamos ambos resultados, vemos que el valor de ***median*** (valor que corresponde a la mediana en la distribución de milisegundos de latencia) es menor en la ruta sin console log (**25.8**) que en la ruta con console log (**98.5**). Estos valores los extraigo del *summary report* en ambos casos. Eso indica que el proceso sin console log es más eficiente. A su vez, el valor de ***request\_rate*** fue de **693/s** para la ruta sin console log, y de **283/s** para la ruta con console log. Esto también me indica que la ruta sin console log es más eficiente, ya que este servidor despliega una mayor cantidad de respuestas por segundo.

Luego utilizamos el *Node built-in profiler*. Creamos el archivo isolate, necesario para el análisis. Corrimos nuevamente los tests con *artillery*, y se observaron resultados similares a los anteriores:

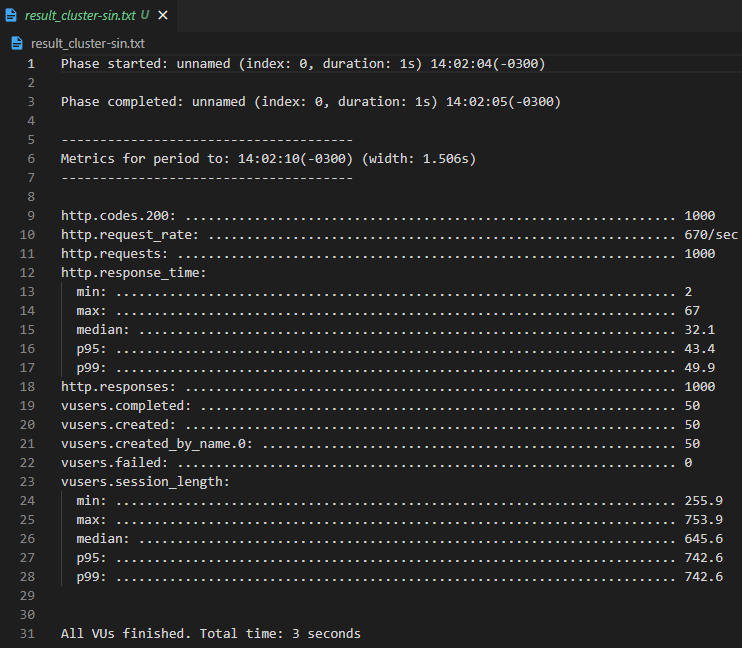
// NODE BUILT-IN PROFILER

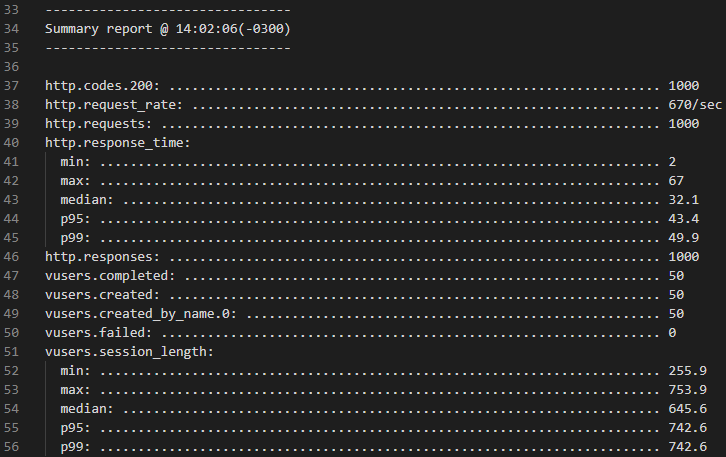
// node --prof server.js



/ EN OTRA CONSOLA:

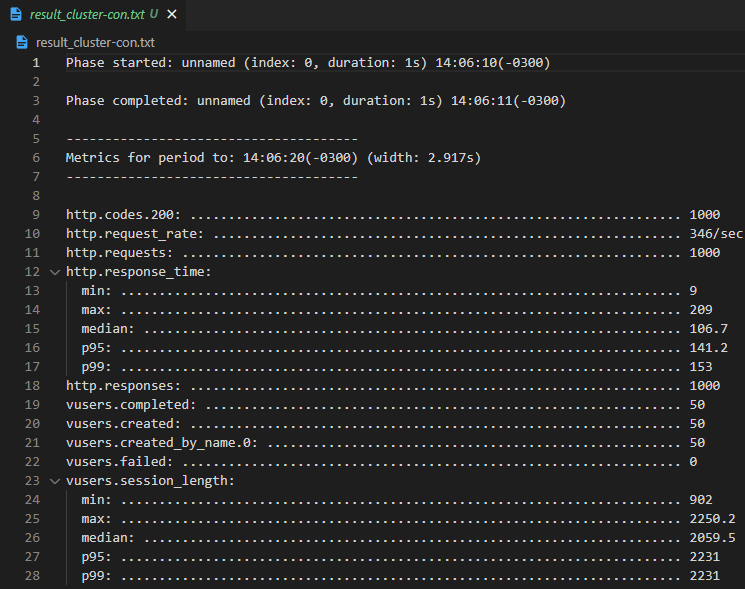
// artillery quick --count 50 -n 20 http://localhost:8081/info-sin > result\_cluster-sin.txt

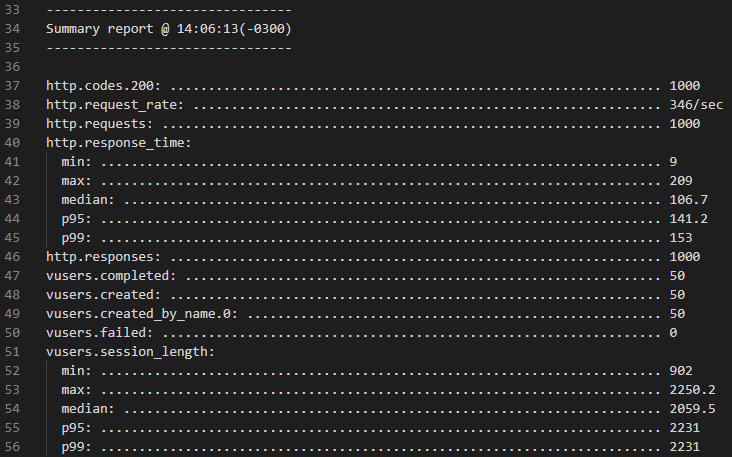




// EN OTRA CONSOLA:

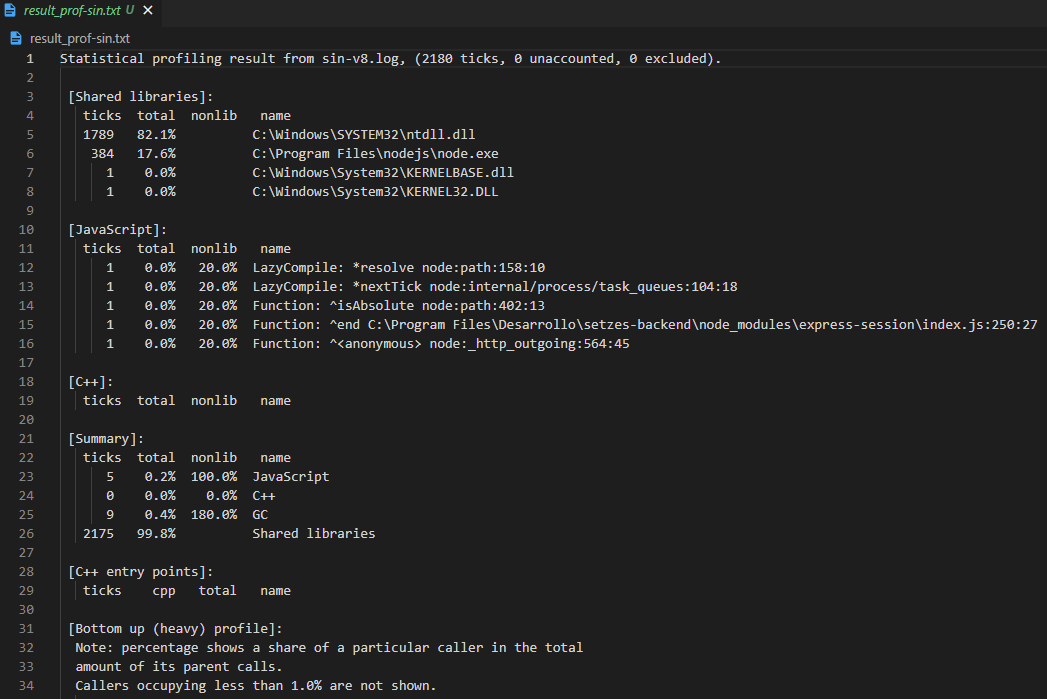
// artillery quick --count 50 -n 20 http://localhost:8081/info-con > result\_cluster-con.txt

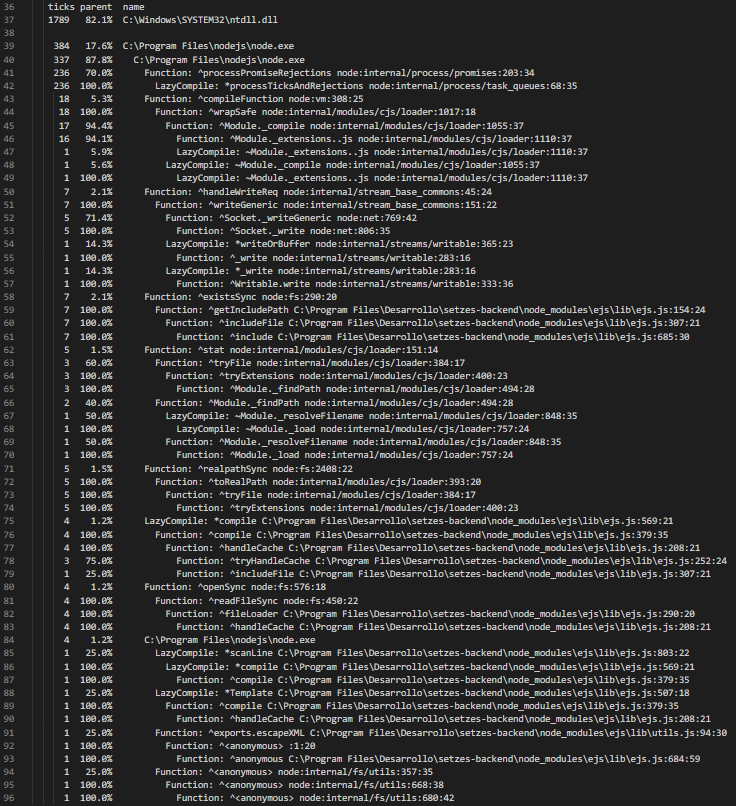




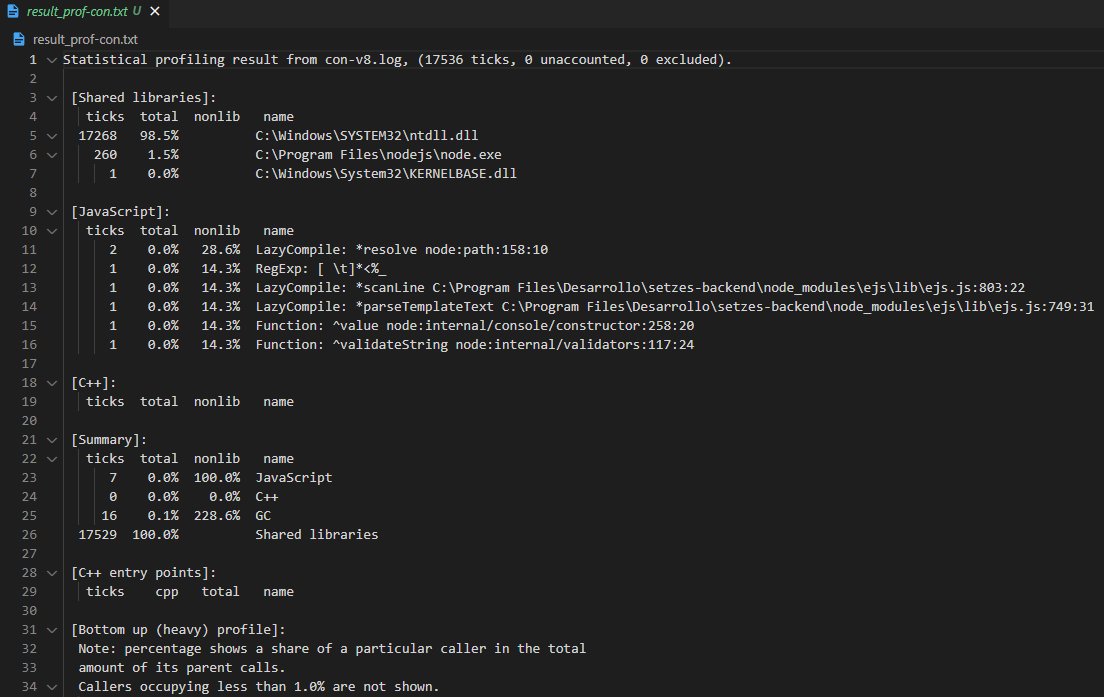
Con el siguiente commando, logramos obtener un perfil estadístico de ambas respuestas. En el mismo, analizamos los ticks que presentó cada una de las respuestas:

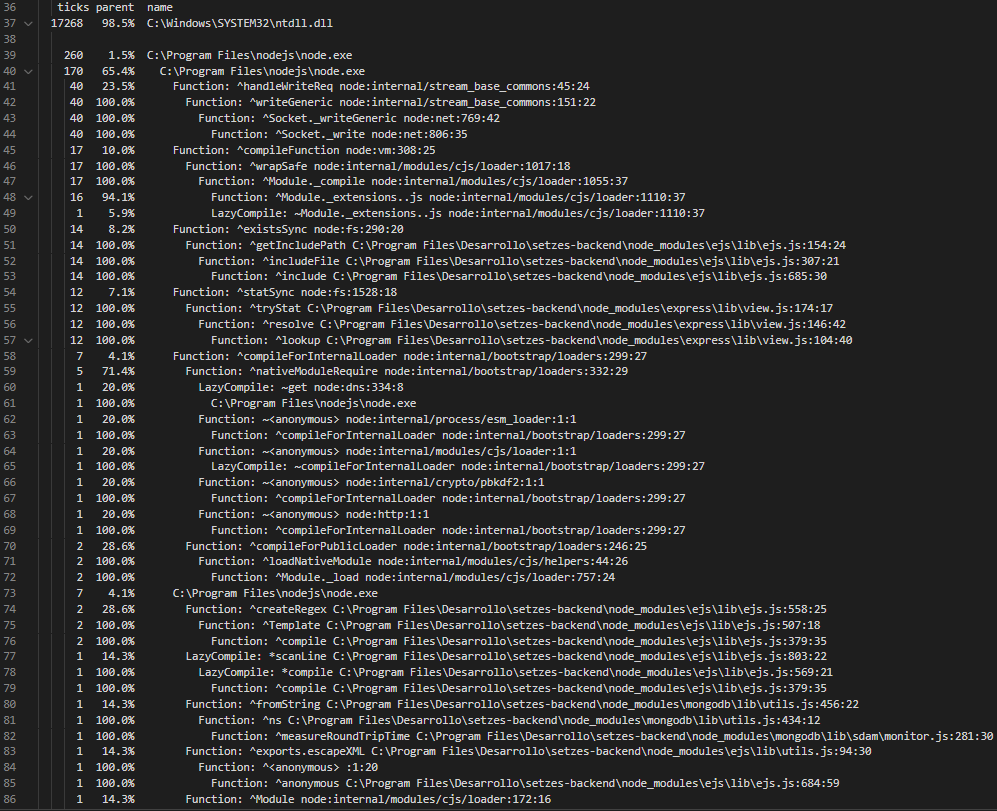
// node --prof-process sin-v8.log > result\_prof-sin.txt

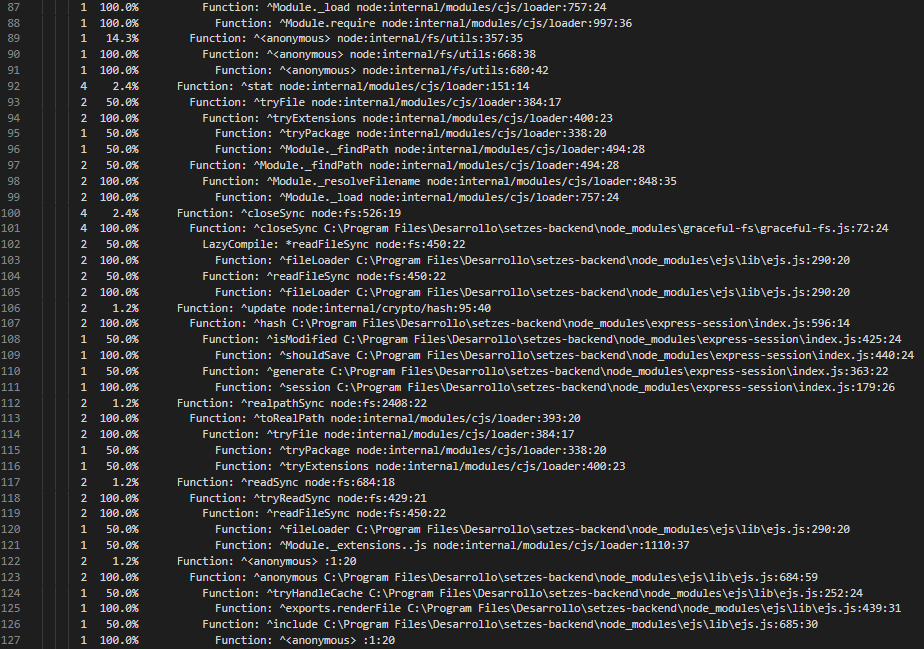




node --prof-process con-v8.log > result\_prof-con.txt







Como se aprecia en las capturas, la cantidad de *ticks* es mucho mayor en la ruta que sí tiene el console log (17529) en relación a la ruta que no lo tiene (2175). Esta información la extraigo del *summary report*, que se observa en las capturas para ambos casos. La menor cantidad de ticks en el proceso más “liviano” sigue la línea de lo que veníamos viendo en los procesos anteriores.

Por último, pruebo con *Autocannon*, una dependencia similar a *Artillery*. También usamos 0x, para perfilar y agregar más información al análisis. Utilizo un archivo llamado *benchmark js* para realizar el test, y modifico el *package json* de manera acorde.

// MODIFICO package.json //

// "scripts": {

//   "test": "node benchmark.js",

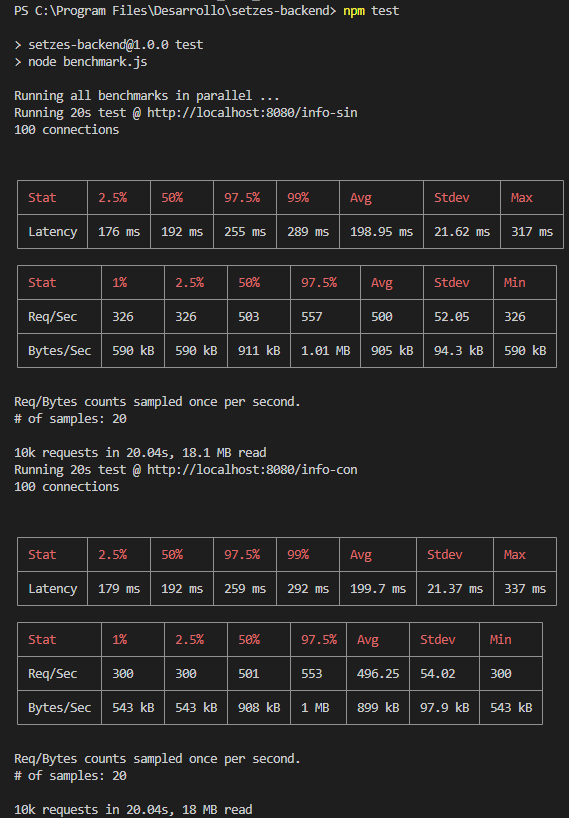
//   "start": "0x server.js"

// }

// npm start

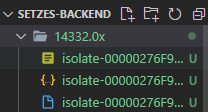
// EN OTRA CONSOLA:

// npm test



Nuevamente vemos que el tiempo promedio de latencia en milisegundos es mayor en el proceso con console log (199.7 ms) que en el proceso sin (198.95 ms). A su vez, la cantidad de respuestas por segundo es mayor en el proceso más liviano (500 en promedio) que en el proceso más pesado (496.25). Esto sigue la línea de lo observado anteriormente.

Mediante 0x, se generan estos tres archivos en una carpeta con nombre aleatorio:



Uno de ellos debería permitir el análisis del diagrama de flama en el navegador, pero yo no fui capaz de hacerlo con ninguno de ellos. Desconozco si es un problema de mi navegador o del proceso, pero no fui capaz de visualizarlo.

**Conclusión**

El proceso más liviano sin console log exhibió en todos los casos, en promedio, 1) un mayor número de respuestas por segundo, 2) un menor número de ticks y 3) un menor número de latencia. Esto es coherente con lo anticipado previo a la realización de todos los *tests*.