>> Consigna: 1

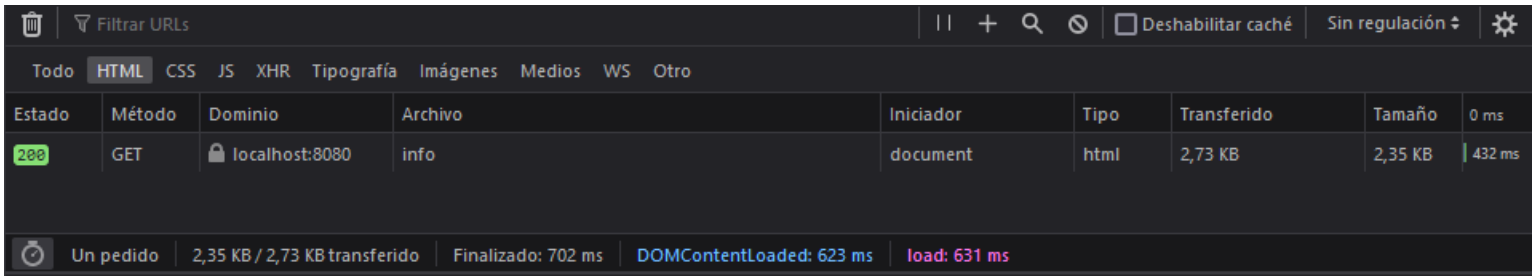
Incorporar al proyecto de servidor de trabajo la compresión gzip.

Verificar sobre la ruta /info con y sin compresión, la diferencia de cantidad de bytes devueltos en un caso y otro.

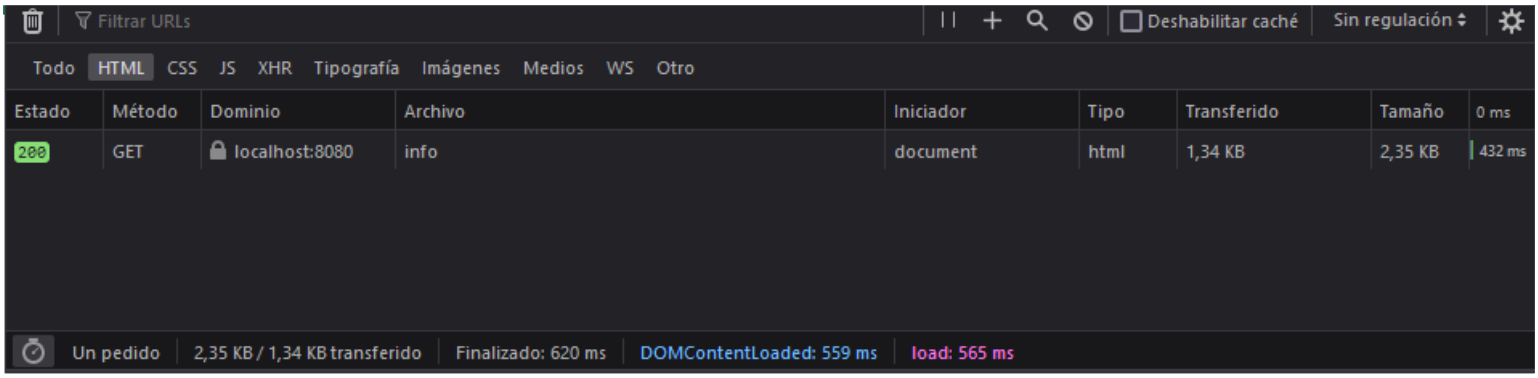
Ruta "/info", servidor sin y con "compression":

Firefox:

- Sin Compresión:

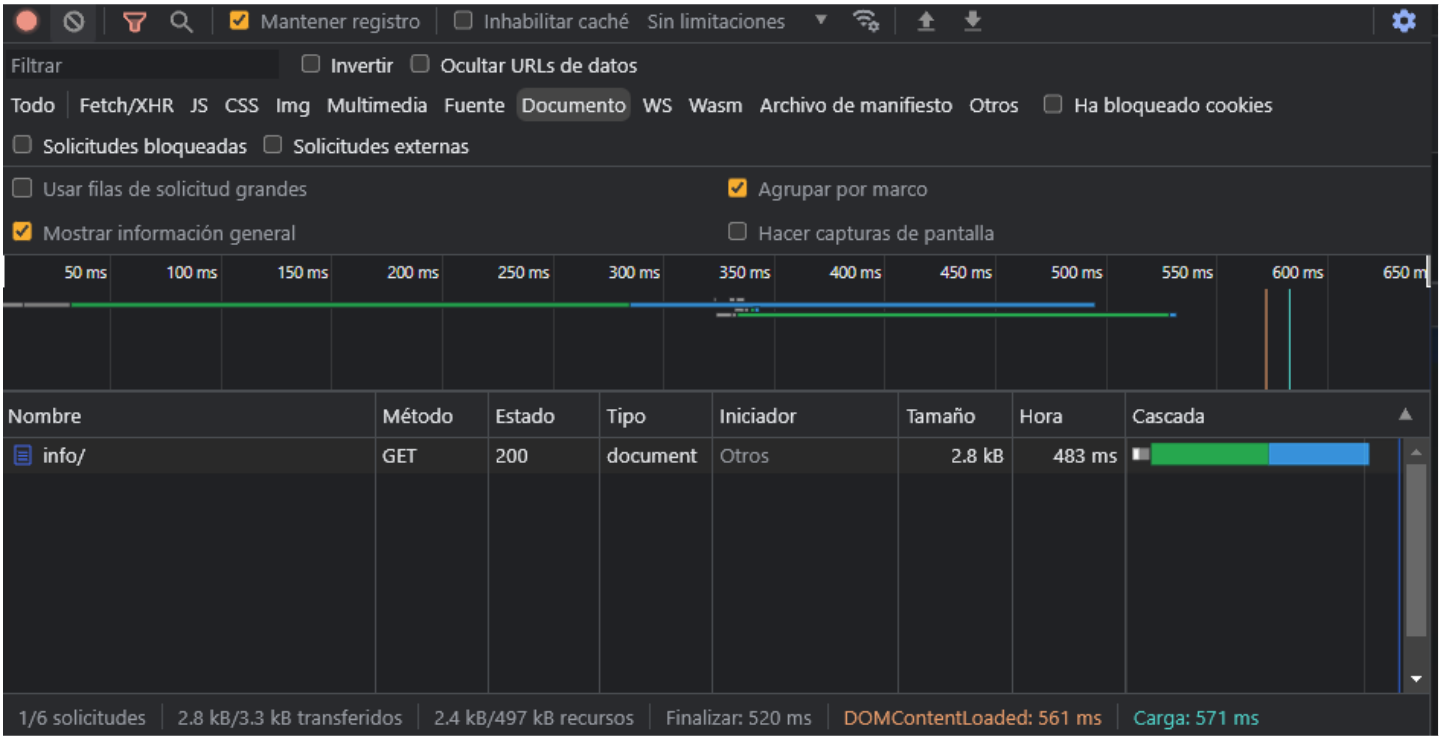


- Con Compresión:

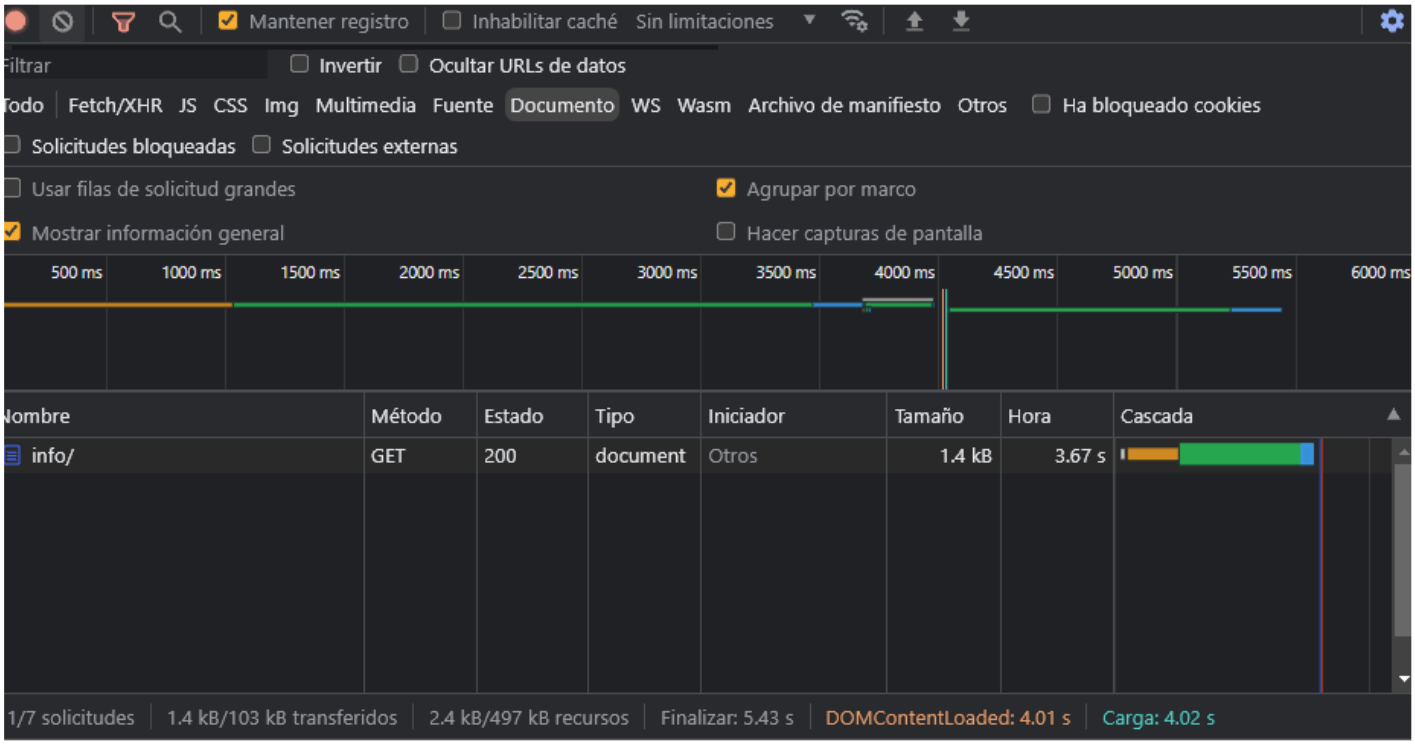


Chrome:

- Sin compresión:



- Con Compresión:



Luego implementar loggueo (con alguna librería vista en clase) que registre lo siguiente:

Winston: (https://www.npmjs.com/package/winston)

Ruta y método de todas las peticiones recibidas por el servidor (**info**)

- server.js línea 80

Ruta y método de las peticiones a rutas inexistentes en el servidor (**warning**)

- server.js línea 129

Errores lanzados por las apis de mensajes y productos, únicamente (**error**)

- server.js línea 123

Mensajes:

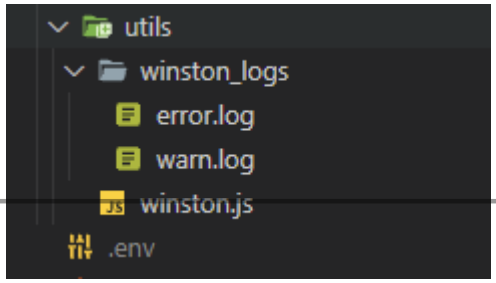
- */src/routes/router.mensajes.js*
- */src/controllers/contenedorMensajes.js*

Productos:

- */src/controllers/contenedorProductos.js*
- */src/routes/router.productos.js*

```
PROBLEMAS   SALIDA   CONSOLA DE DEPURACIÓN   TERMINAL

PS D:\CODERHOUSE\CODERHOUSE\BACKEND\DESAFIOS\Desafio-32> node server
YARGS: { port: 8080, debug: false, modo: 'FORK', pid: 14948, otros: [] }
info: Escuchando en el Puerto: 8080 - MODO: FORK - PID: 14948 - fyh: 15/8/2022, 6:55:53
info: Ruta: /info y Metodo: GET solicitado
info: Ruta: /api/mensajes y Metodo: GET solicitado
info: Ruta: /api/mensajes/344 y Metodo: GET solicitado
error: Error Metodo: GET Mensaje 344 no existe
info: Ruta: /api/asdas y Metodo: GET solicitado
warn: Ruta /api/asdas y metodo GET no implementada - fyh: 15/8/2022, 6:56:27
```



>> Consigna: 2

Profile con ruta “/info” con y sin console.log

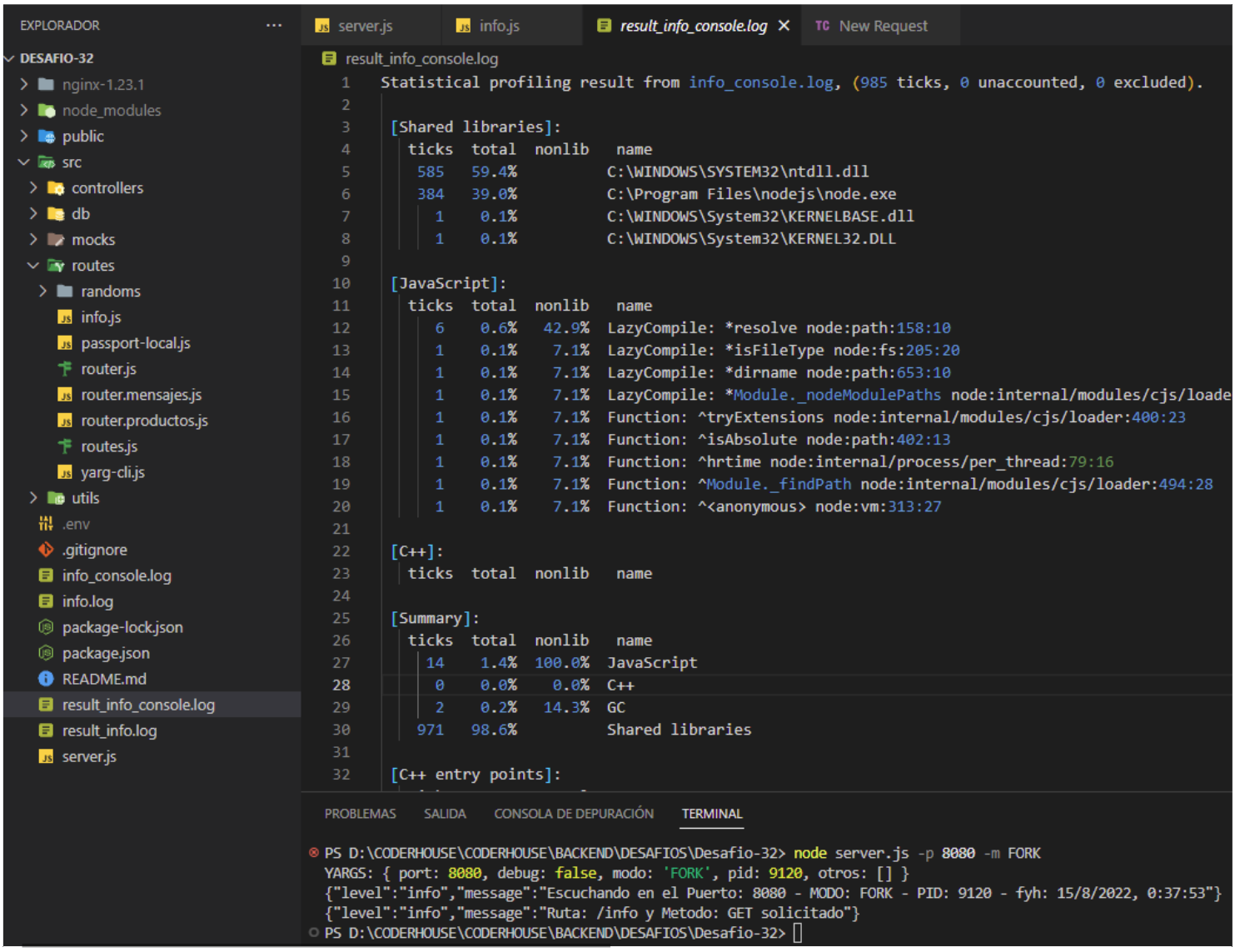
1) node --prof server.js

Get a “/info” con console, detener servidor, re nombrar archivo a “info_console.log”

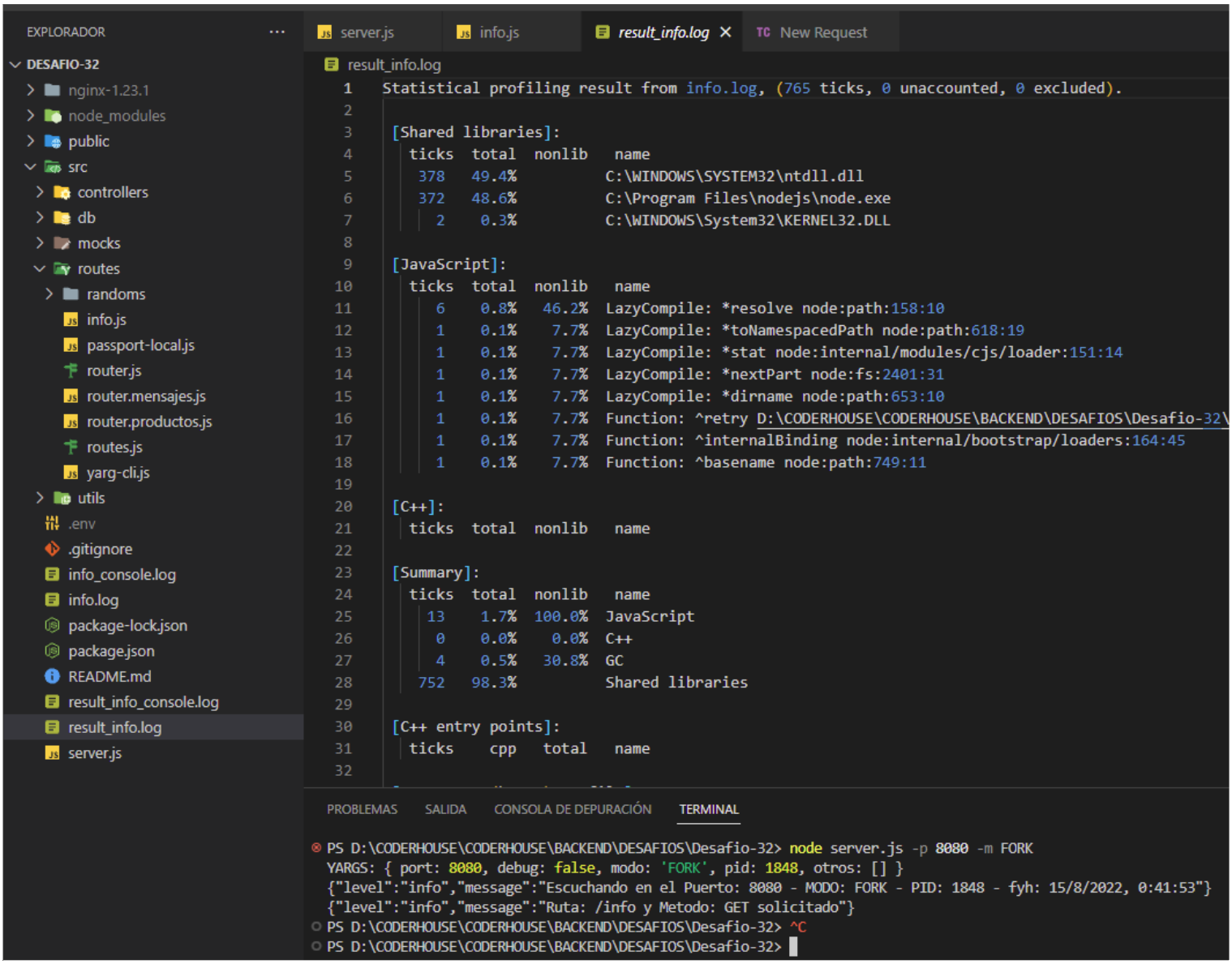
2) node --prof server.js

Get a “/info” , detener servidor, re nombrar archivo a “info.log”

3) node --prof-process info_console.log > result_info_console.log



4) node --prof-process info.log > result_info.log

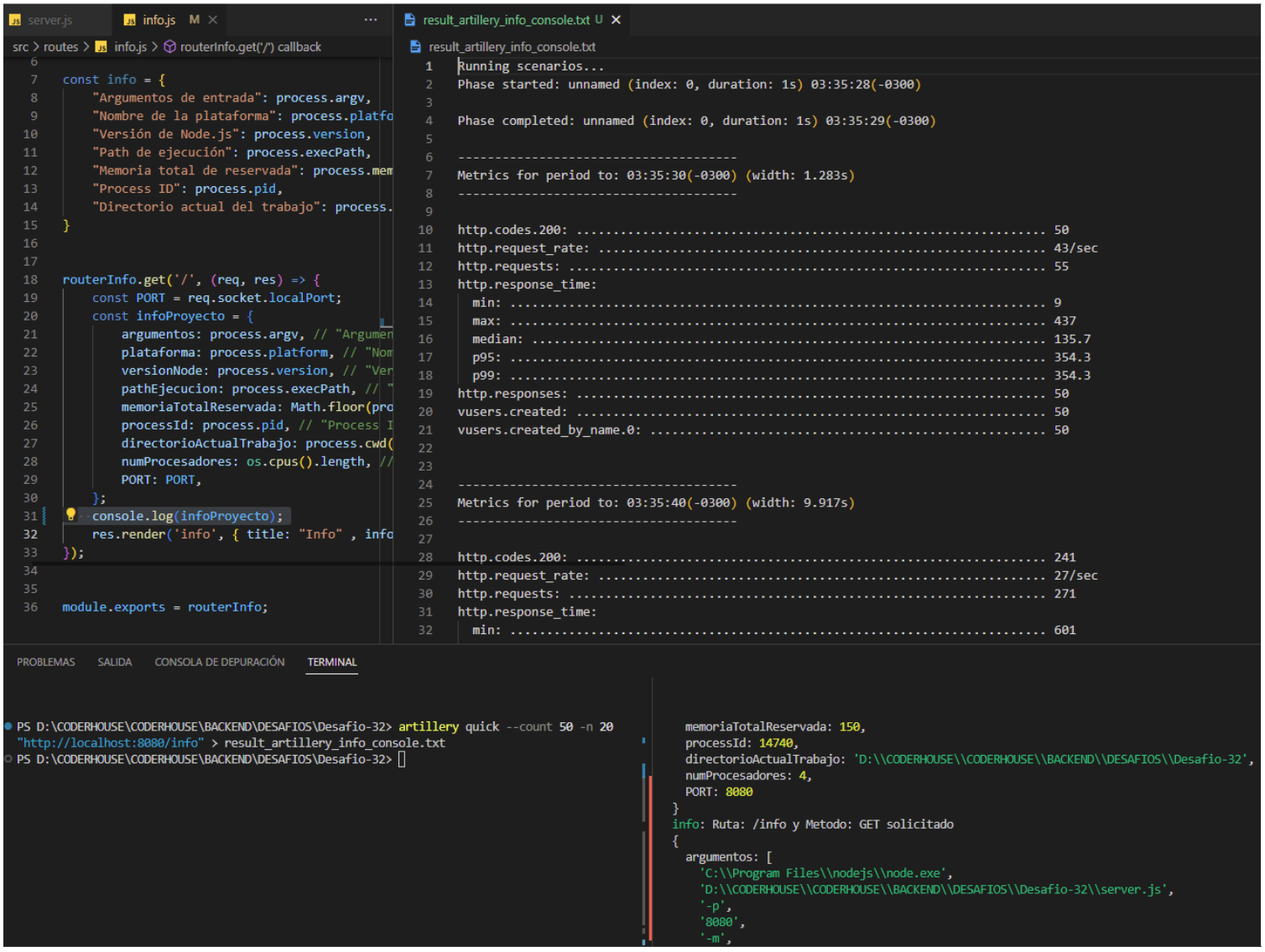


Utilizaremos como test de carga Artillery en línea de comandos, emulando 50 conexiones concurrentes con 20 request por cada una. Extraer un reporte con los resultados en archivo de texto.

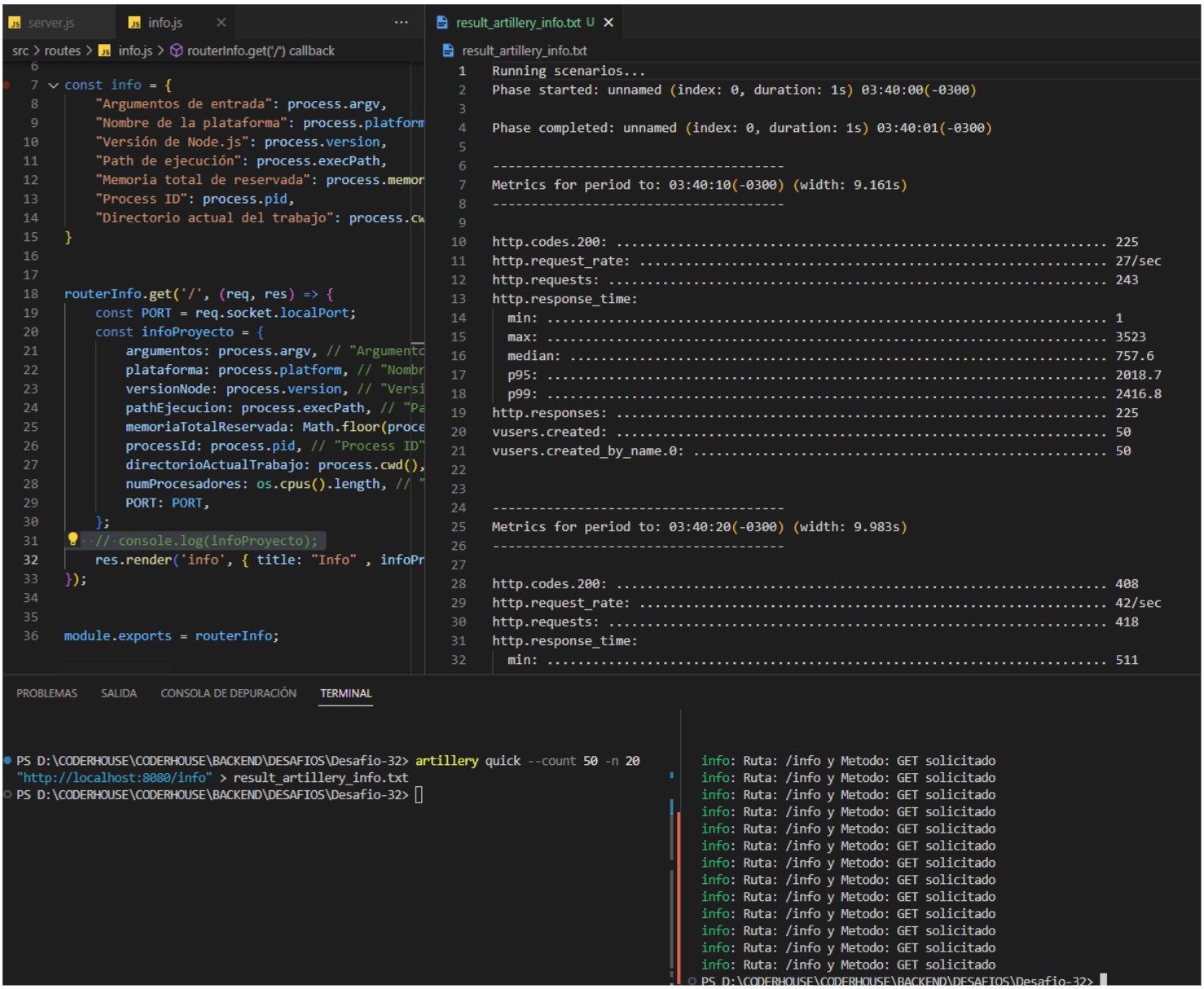
npm install -g artillery

1) node server.js -p 8080 -m FORK

2) artillery quick --count 50 -n 20 "http://localhost:8080/info" > artilley/result_artillery_info_console.txt



3) artillery quick --count 50 -n 20 "http://localhost:8080/info" > result_artillery_info.txt

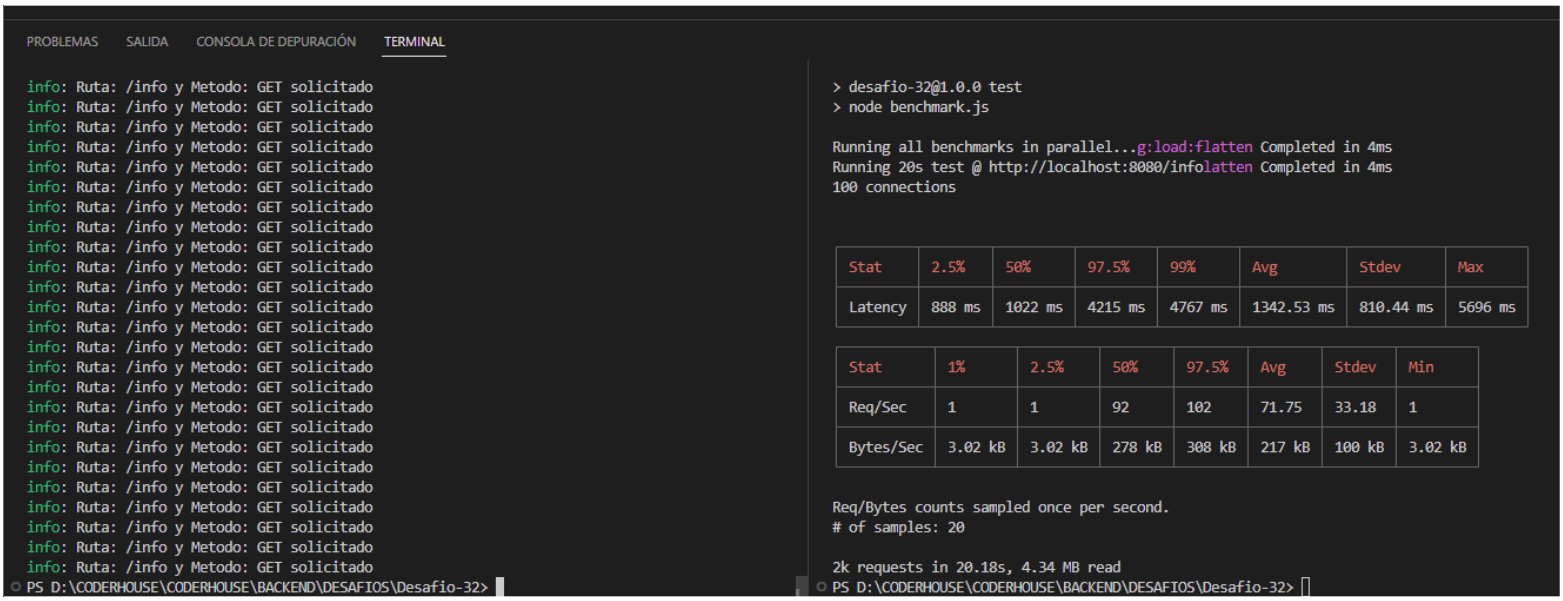


>> Consigna: 3

Luego utilizaremos Autocannon en línea de comandos, emulando 100 conexiones concurrentes realizadas en un tiempo de 20 segundos. Extraer un reporte con los resultados (puede ser un print screen de la consola)

1 - npm start

2 - npm run test (**sin** console.log)



```
> desafio-32@1.0.0 test
> node benchmark.js

Running all benchmarks in parallel...g:load:flatten Completed in 4ms
Running 20s test @ http://localhost:8080/infolatten Completed in 4ms
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	888 ms	1022 ms	4215 ms	4767 ms	1342.53 ms	810.44 ms	5696 ms

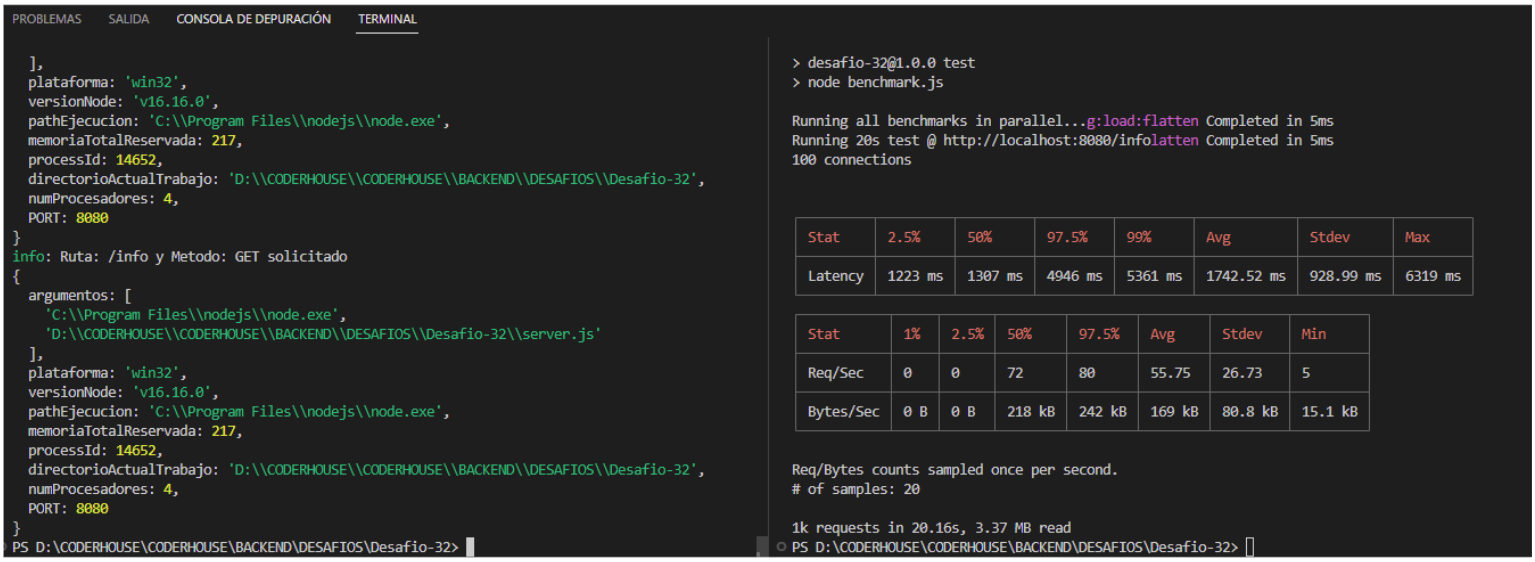
Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	1	1	92	102	71.75	33.18	1
Bytes/Sec	3.02 kB	3.02 kB	278 kB	308 kB	217 kB	100 kB	3.02 kB

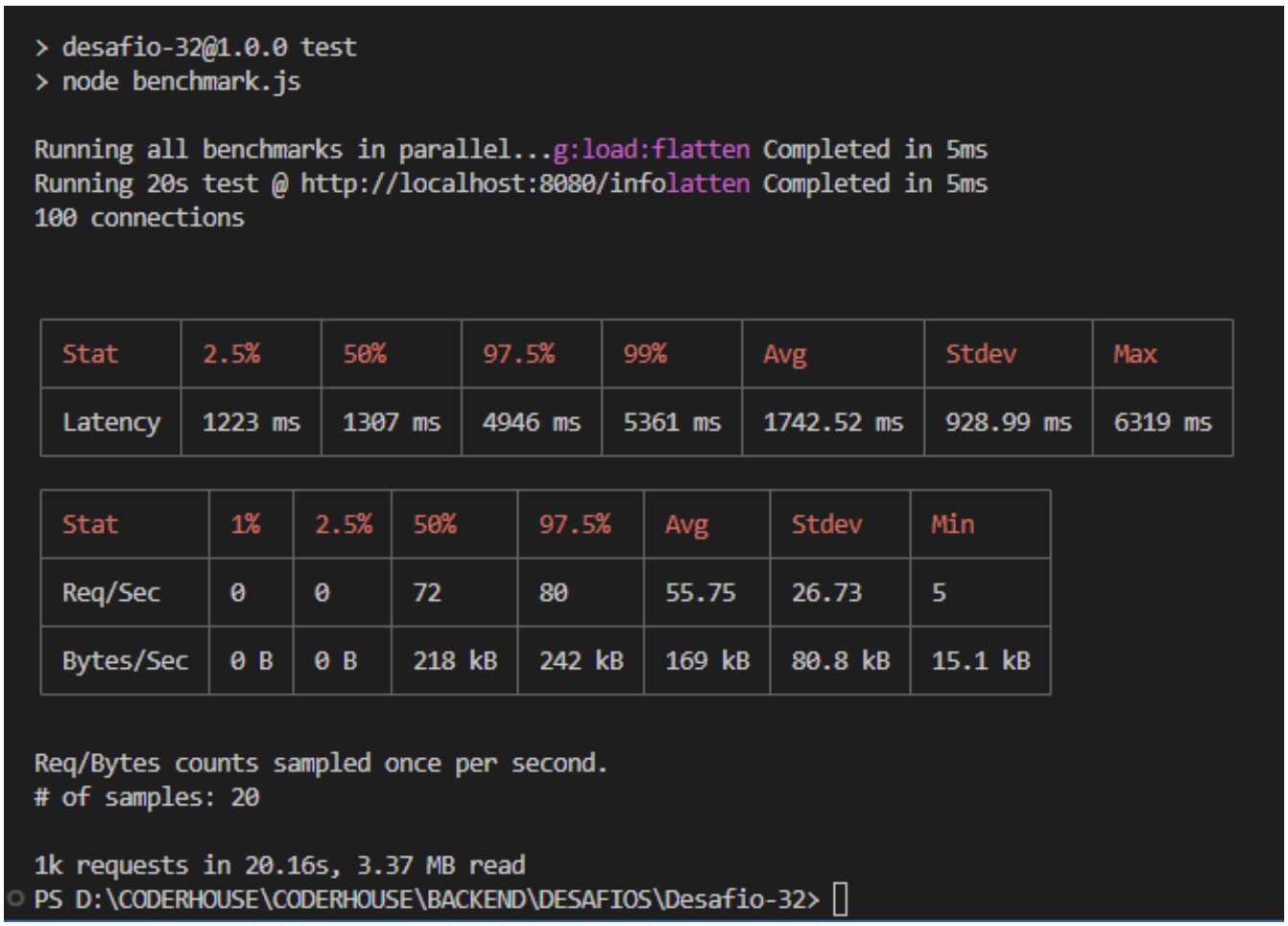
```
Req/Bytes counts sampled once per second.
# of samples: 20

2k requests in 20.18s, 4.34 MB read
PS D:\CODERHOUSE\CODERHOUSE\BACKEND\DESAFIOS\Desafio-32>
```

1 - npm start

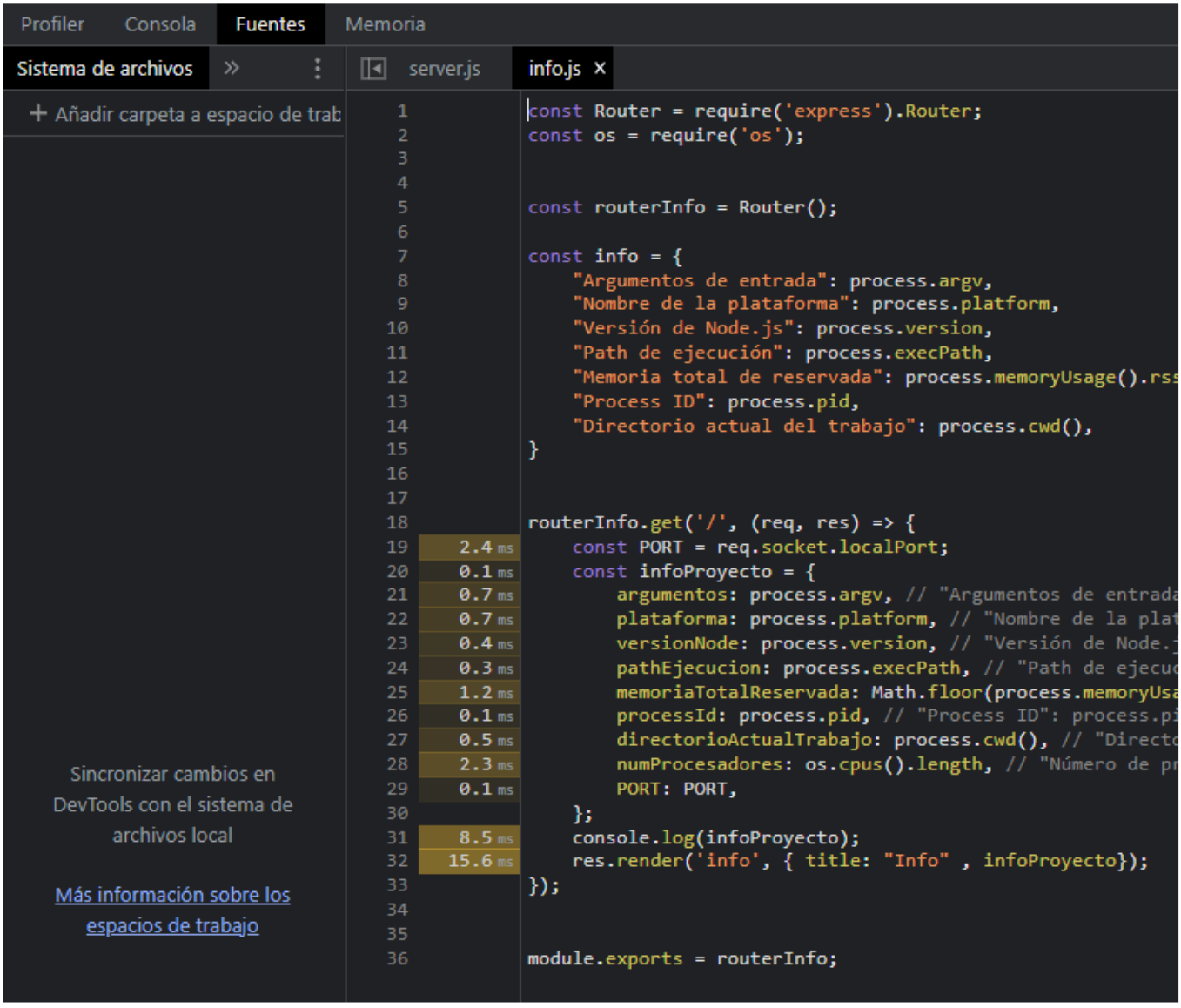
2 - npm run test (con console.log)



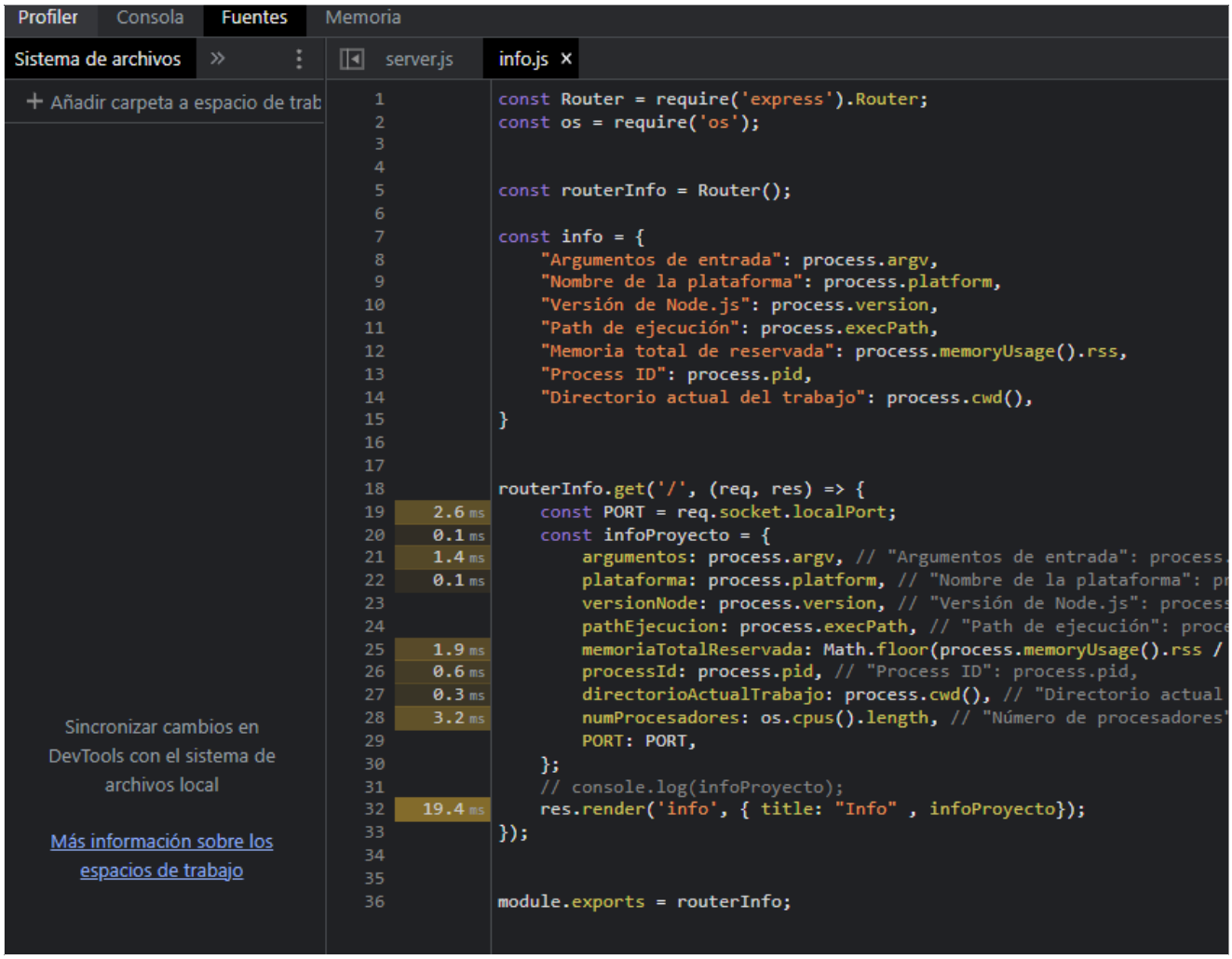


2) El perfilamiento del servidor con el modo inspector de node.js --inspect. Revisar el tiempo de los procesos menos performantes sobre el archivo fuente de inspección.

- Ruta “/info” con console.log:
  1) node --inspect server.js
  2) chrome://inspect/#devices



- Ruta “/info”:
  1) node --inspect server.js
  2) chrome://inspect/#devices



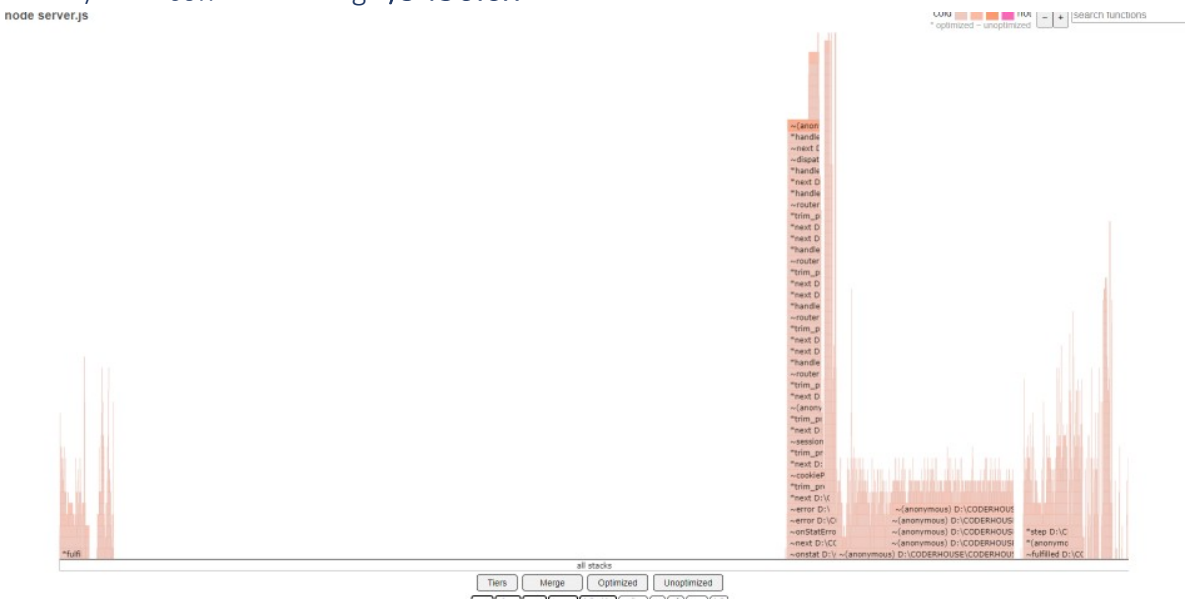
3) El diagrama de flama con 0x, emulando la carga con Autocannon con los mismos parámetros anteriores.

1- npm run ox (0x)

2 – npm run test (autocanon)

3 – Detener servidor

Ruta “/info” con console.log : /9456.0x



Ruta “/info” sin console.log : /12216.ox