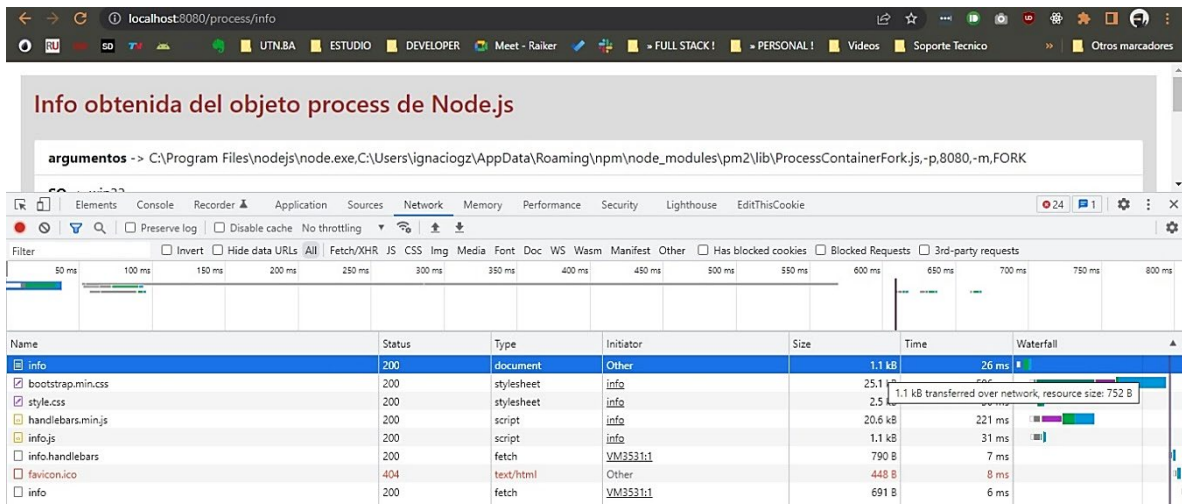
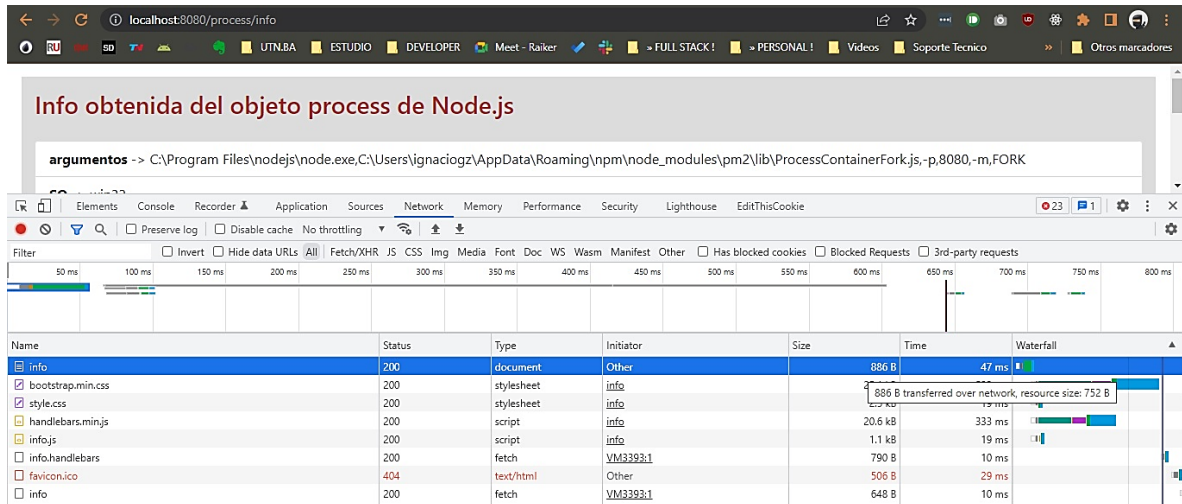


Ruta: /process/info [Sin gzip]



Ruta: /process/info [Con gzip]



Profiling: Node built-in

Ruta: /process/api/info

Test: Artillery

```
result-con-consolelog.txt U X
profiling > node-built-in > artillery > result-con-consolelog.txt
84
85 [Summary]:
86 ticks total nonlib name
87 107 0.9% 99.1% JavaScript
88 0 0.0% 0.0% C++
89 192 1.6% 177.8% GC
90 12077 99.1% Shared libraries
91 1 0.0% Unaccounted
92
```

```
result-sin-consolelog.txt U X
profiling > node-built-in > artillery > result-sin-consolelog.txt
75
76 [Summary]:
77 ticks total nonlib name
78 103 2.1% 100.0% JavaScript
79 0 0.0% 0.0% C++
80 109 2.3% 105.8% GC
81 4712 97.9% Shared libraries
82
```

El código sin console.log presenta menor consumo en shared libraries tanto en términos porcentuales como en ticks.

Profiling: Node built-in

Ruta: /process/api/info

Test: Autocannon en consola

```
result-con-consolelog.txt U X
profiling > node-built-in > autocannon > result-con-consolelog.txt
201
202 [Summary]:
203 ticks total nonlib name
204 407 3.2% 98.8% JavaScript
205 0 0.0% 0.0% C++
206 240 1.9% 58.3% GC
207 12454 96.8% Shared libraries
208 5 0.0% Unaccounted
209
```

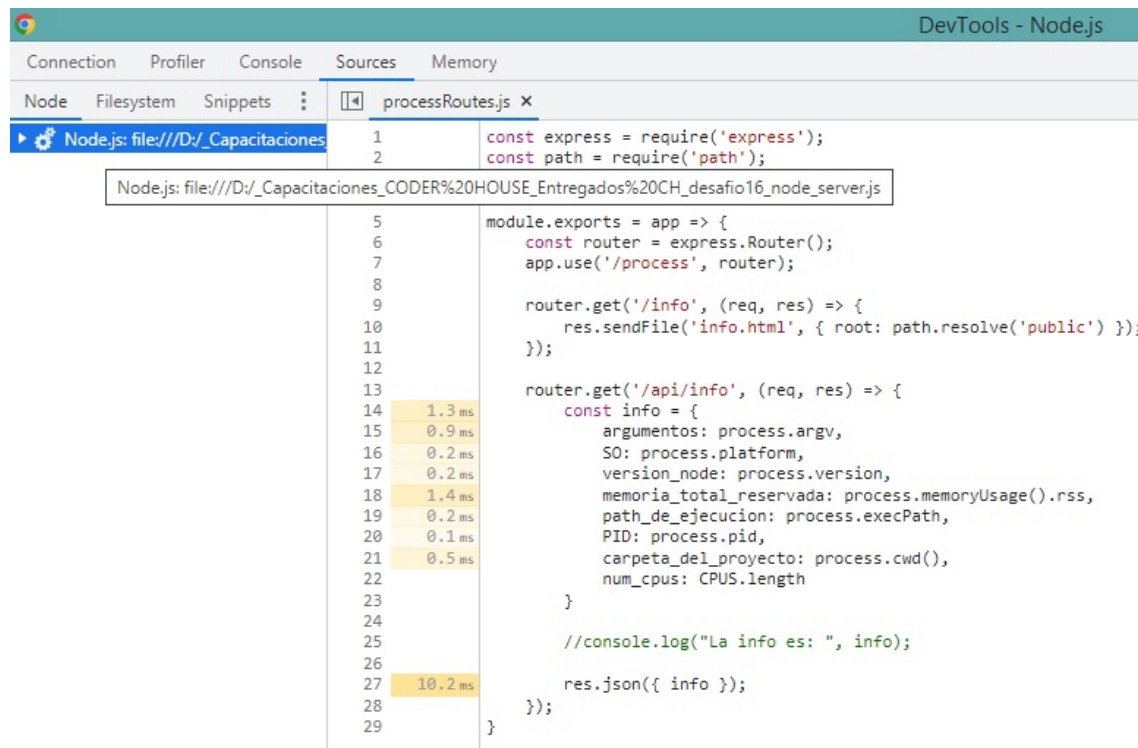
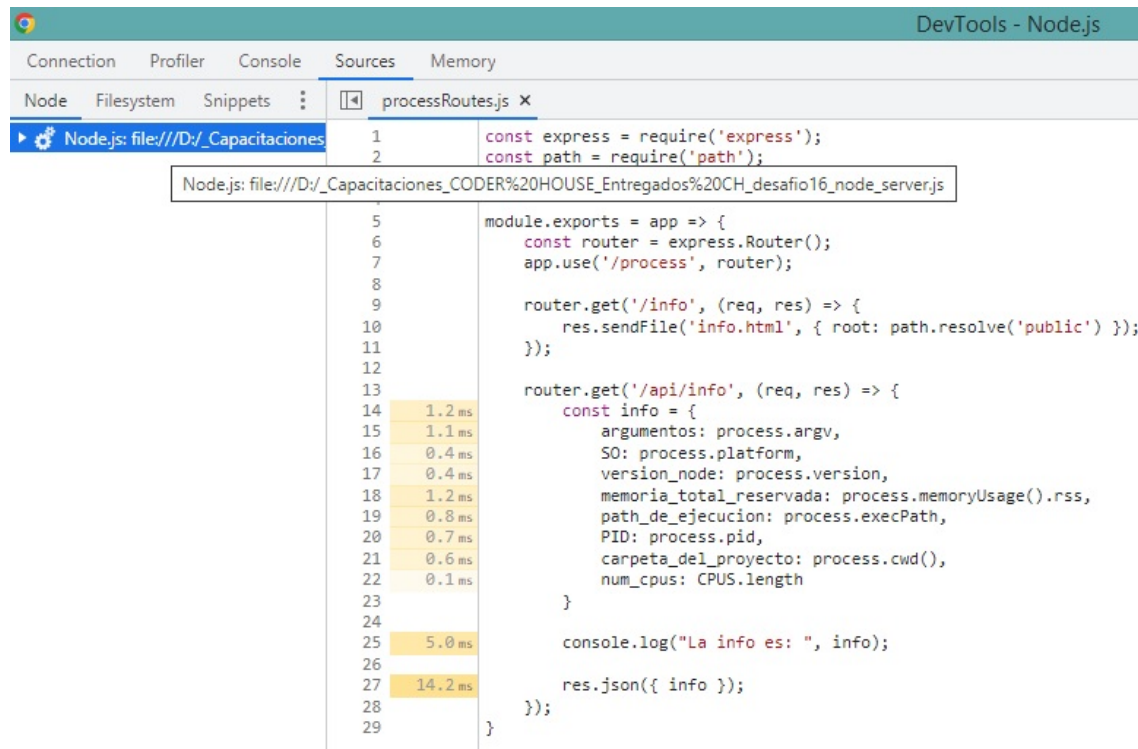
```
result-sin-consolelog.txt U X
profiling > node-built-in > autocannon > result-sin-consolelog.txt
266
267 [Summary]:
268 ticks total nonlib name
269 1022 6.2% 97.9% JavaScript
270 0 0.0% 0.0% C++
271 516 3.2% 49.4% GC
272 15335 93.6% Shared libraries
273 22 0.1% Unaccounted
274
```

El código sin console.log también presenta menor consumo en shared libraries tanto en términos porcentuales como en ticks.

Profiling: Node inspect

Ruta: /process/api/info

Test: Artillery

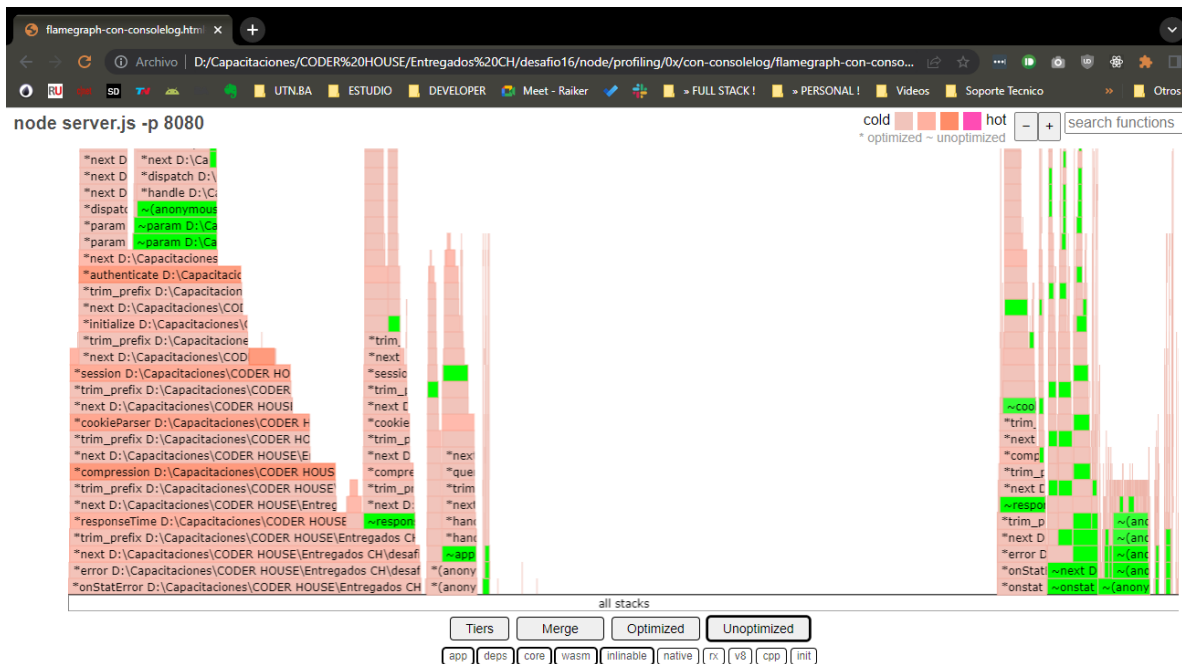


Como se puede apreciar en las imágenes anteriores, un solo console.log consume 5.8 ms (Cantidad elevada de tiempo de ejecución, si lo comparamos con el resto de línea), y al mismo tiempo, retrasa la ejecución de instrucciones posteriores.

Profiling: 0x

Ruta: /process/api/info

Test: Autocannon en benchmark.js



Conclusión:

La instrucción `console.log` al ser de tipo bloqueante, genera retrasos en la ejecución de instrucciones posteriores.

Por otro lado, su lógica en sí, consume muchos recursos ya que implica representar datos por salida estándar (En pantalla).

Se recomienda evitar su uso, reemplazándolo por sistemas de logs.