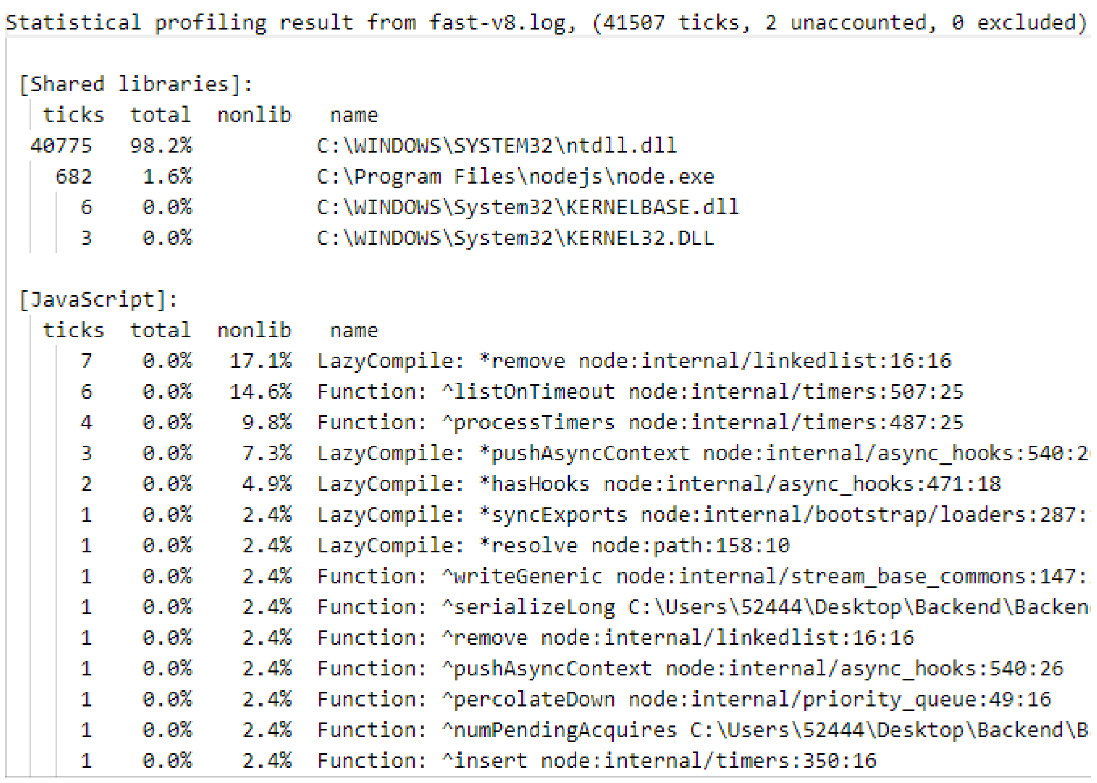
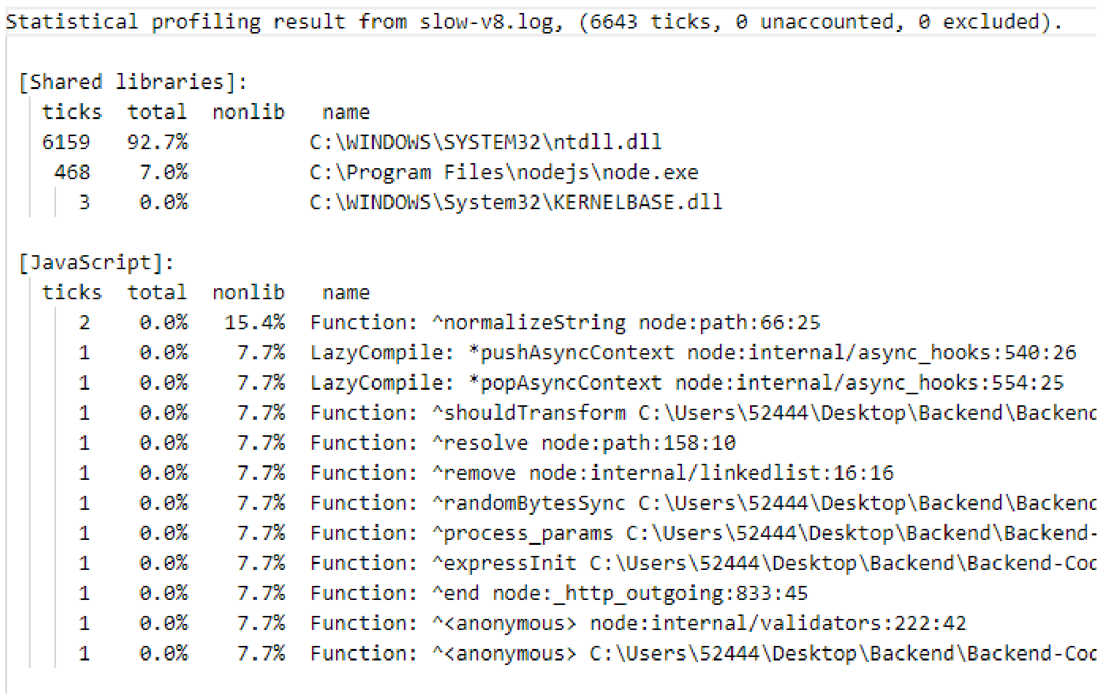
1. El perfilamiento del servidor realizando el test con –prof de node.js.

SIN LOG



```
Statistical profiling result from fast-v8.log, (41507 ticks, 2 unaccounted, 0 excluded)

 [Shared libraries]:
   ticks  total  nonlib   name
  40775   98.2%          C:\WINDOWS\SYSTEM32\ntdll.dll
    682    1.6%          C:\Program Files\nodejs\node.exe
      6    0.0%          C:\WINDOWS\System32\KERNELBASE.dll
      3    0.0%          C:\WINDOWS\System32\KERNEL32.DLL

 [JavaScript]:
   ticks  total  nonlib   name
      7    0.0%   17.1%  LazyCompile: *remove node:internal/linkedlist:16:16
      6    0.0%   14.6%  Function: ^listOnTimeout node:internal/timers:507:25
      4    0.0%    9.8%  Function: ^processTimers node:internal/timers:487:25
      3    0.0%    7.3%  LazyCompile: *pushAsyncContext node:internal/async_hooks:540:2
      2    0.0%    4.9%  LazyCompile: *hasHooks node:internal/async_hooks:471:18
      1    0.0%    2.4%  LazyCompile: *syncExports node:internal/bootstrap/loaders:287:
      1    0.0%    2.4%  LazyCompile: *resolve node:path:158:10
      1    0.0%    2.4%  Function: ^writeGeneric node:internal/stream_base_commons:147:
      1    0.0%    2.4%  Function: ^serializeLong C:\Users\52444\Desktop\Backend\Backen
      1    0.0%    2.4%  Function: ^remove node:internal/linkedlist:16:16
      1    0.0%    2.4%  Function: ^pushAsyncContext node:internal/async_hooks:540:26
      1    0.0%    2.4%  Function: ^percolateDown node:internal/priority_queue:49:16
      1    0.0%    2.4%  Function: ^numPendingAcquires C:\Users\52444\Desktop\Backend\B
      1    0.0%    2.4%  Function: ^insert node:internal/timers:350:16
```

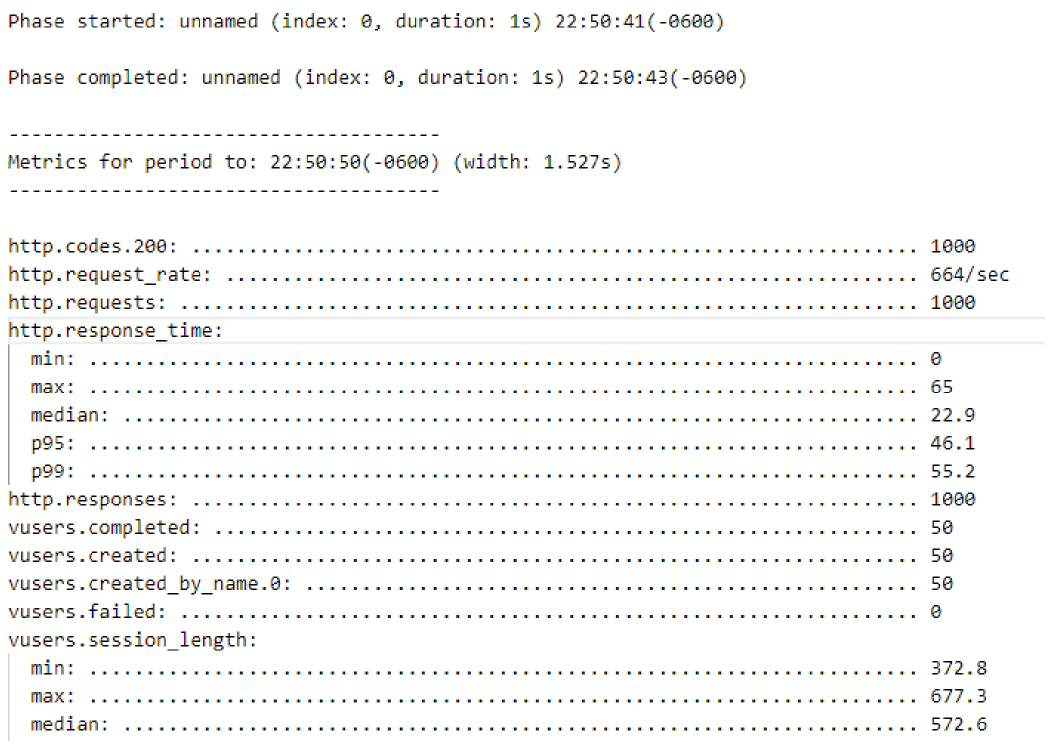
CON LOG



```
Statistical profiling result from slow-v8.log, (6643 ticks, 0 unaccounted, 0 excluded).

 [Shared libraries]:
   ticks  total  nonlib   name
   6159   92.7%          C:\WINDOWS\SYSTEM32\ntdll.dll
    468    7.0%          C:\Program Files\nodejs\node.exe
      3    0.0%          C:\WINDOWS\System32\KERNELBASE.dll

 [JavaScript]:
   ticks  total  nonlib   name
      2    0.0%   15.4%  Function: ^normalizeString node:path:66:25
      1    0.0%    7.7%  LazyCompile: *pushAsyncContext node:internal/async_hooks:540:26
      1    0.0%    7.7%  LazyCompile: *popAsyncContext node:internal/async_hooks:554:25
      1    0.0%    7.7%  Function: ^shouldTransform C:\Users\52444\Desktop\Backend\Backenc
      1    0.0%    7.7%  Function: ^resolve node:path:158:10
      1    0.0%    7.7%  Function: ^remove node:internal/linkedlist:16:16
      1    0.0%    7.7%  Function: ^randomBytesSync C:\Users\52444\Desktop\Backend\Backenc
      1    0.0%    7.7%  Function: ^process_params C:\Users\52444\Desktop\Backend\Backend-
      1    0.0%    7.7%  Function: ^expressInit C:\Users\52444\Desktop\Backend\Backend-Coc
      1    0.0%    7.7%  Function: ^end node:_http_outgoing:833:45
      1    0.0%    7.7%  Function: ^<anonymous> node:internal/validators:222:42
      1    0.0%    7.7%  Function: ^<anonymous> C:\Users\52444\Desktop\Backend\Backend-Coc
```

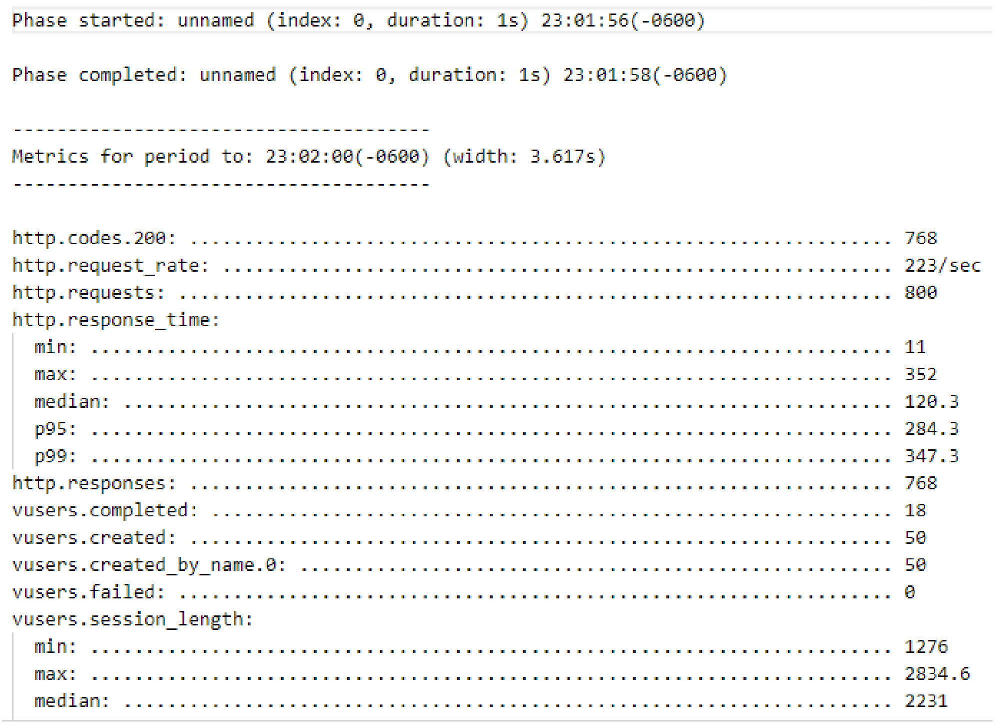
Usar Artillery en línea de comandos.

SIN LOG



```
Phase started: unnamed (index: 0, duration: 1s) 22:50:41(-0600)

Phase completed: unnamed (index: 0, duration: 1s) 22:50:43(-0600)

--------------------------------------
Metrics for period to: 22:50:50(-0600) (width: 1.527s)
--------------------------------------

http.codes.200: ........................................................... 1000
http.request_rate: ........................................................ 664/sec
http.requests: ............................................................ 1000
http.response_time:
  min: .................................................................... 0
  max: .................................................................... 65
  median: ................................................................. 22.9
  p95: .................................................................... 46.1
  p99: .................................................................... 55.2
http.responses: ........................................................... 1000
vusers.completed: ......................................................... 50
vusers.created: ........................................................... 50
vusers.created_by_name.0: ................................................. 50
vusers.failed: ............................................................ 0
vusers.session_length:
  min: .................................................................... 372.8
  max: .................................................................... 677.3
  median: ................................................................. 572.6
```

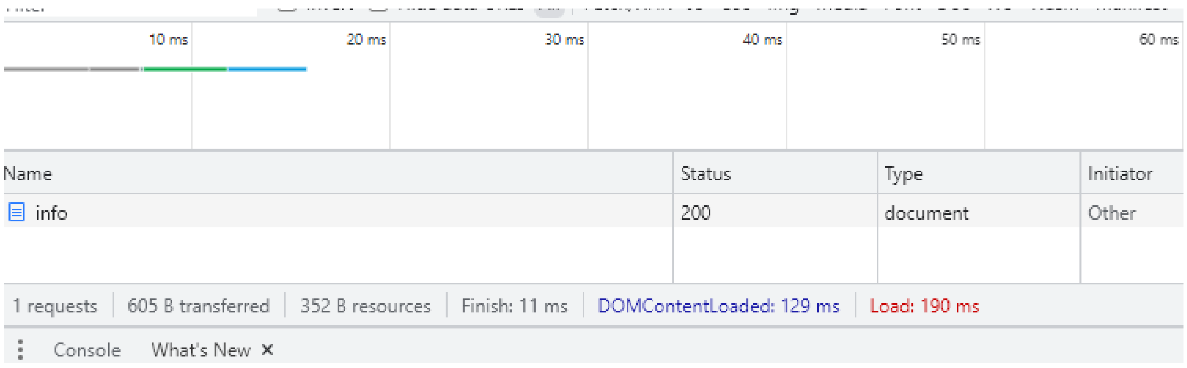
CON LOG



```
Phase started: unnamed (index: 0, duration: 1s) 23:01:56(-0600)

Phase completed: unnamed (index: 0, duration: 1s) 23:01:58(-0600)

--------------------------------------
Metrics for period to: 23:02:00(-0600) (width: 3.617s)
--------------------------------------

http.codes.200: ........................................................... 768
http.request_rate: ........................................................ 223/sec
http.requests: ............................................................ 800
http.response_time:
  min: .................................................................... 11
  max: .................................................................... 352
  median: ................................................................. 120.3
  p95: .................................................................... 284.3
  p99: .................................................................... 347.3
http.responses: ........................................................... 768
vusers.completed: ......................................................... 18
vusers.created: ........................................................... 50
vusers.created_by_name.0: ................................................. 50
vusers.failed: ............................................................ 0
vusers.session_length:
  min: .................................................................... 1276
  max: .................................................................... 2834.6
  median: ................................................................. 2231
```

Usar Autocannon en línea de comandos.

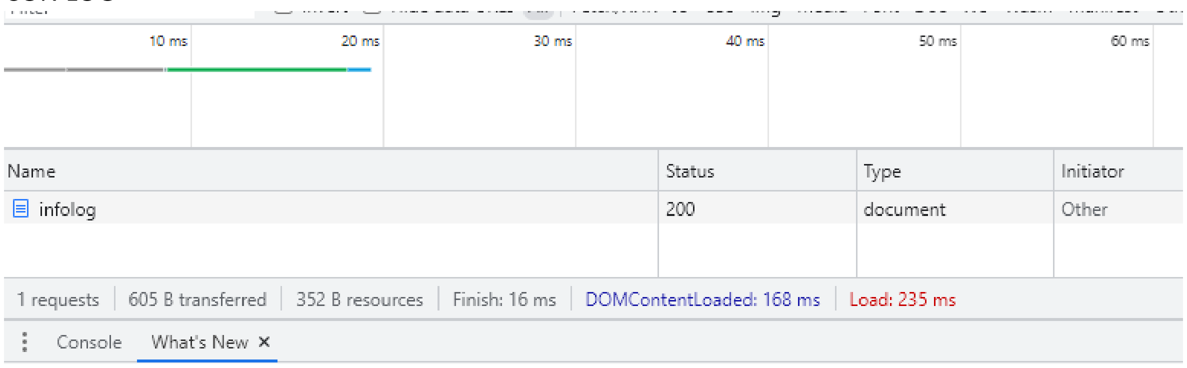
2. El perfilamiento del servidor con el modo de inspector de node.js –inspect. Revisar el tiempo de los procesos menos perfomantes sobre el archivo fuente de inspección.
SIN LOG



Name	Status	Type	Initiator
info	200	document	Other

1 requests | 605 B transferred | 352 B resources | Finish: 11 ms | DOMContentLoaded: 129 ms | Load: 190 ms

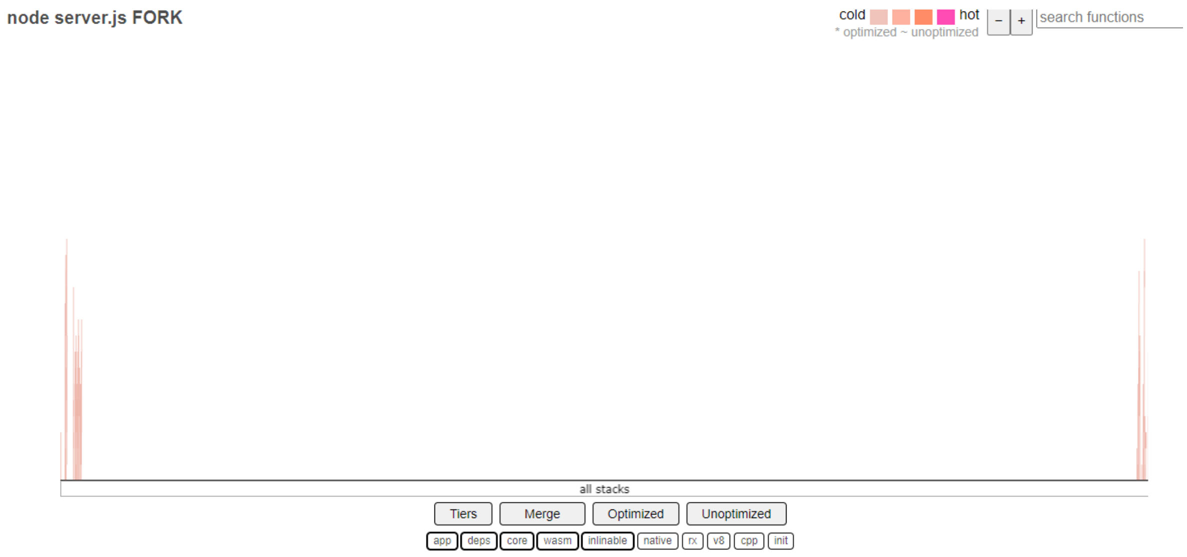
CON LOG



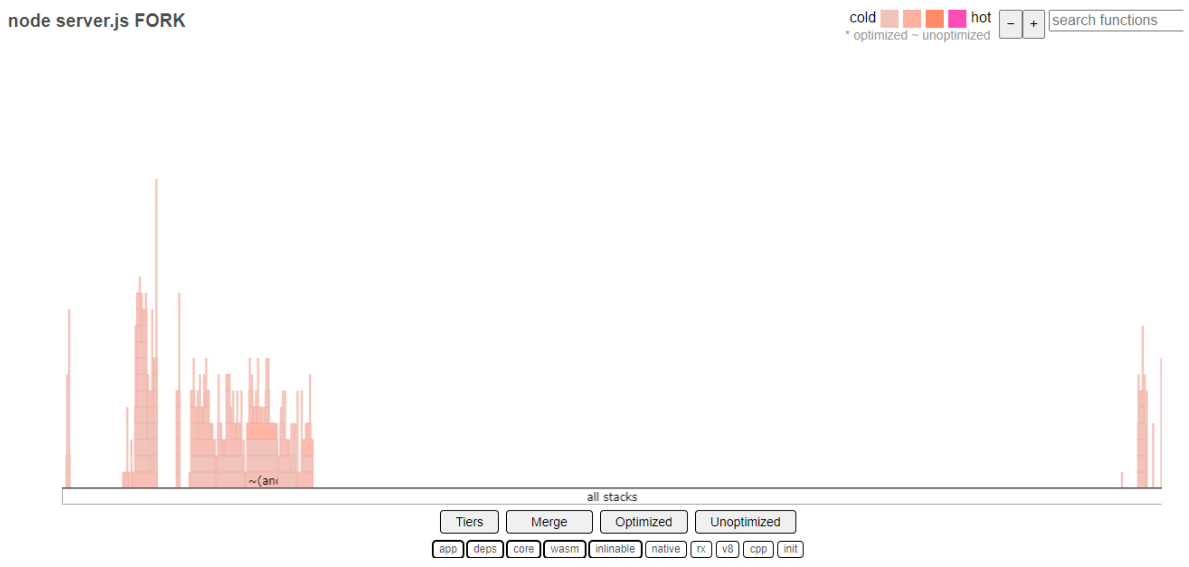
Name	Status	Type	Initiator
infolog	200	document	Other

1 requests | 605 B transferred | 352 B resources | Finish: 16 ms | DOMContentLoaded: 168 ms | Load: 235 ms

3. El diagrama de flama con 0x, emulando carga con Autocannon con los mismos parámetro anteriores.
SIN LOG



CON LOG



CONCLUSIÓN

La prueba comparó un solo console.log como aspecto bloqueante con un proceso no bloqueante y podemos ver claramente en cualquiera de las pruebas de carga la diferencia que hacer un solo console.log en un proceso simple que sólo extrae información, esto quiere decir que a mayor escala, en un proceso mucho más complejo un aspecto bloqueante disminuye con más significancia la eficiencia de la API.