

INFORME DE PRUEBAS DE RENDIMIENTO

Institución: Coderhouse

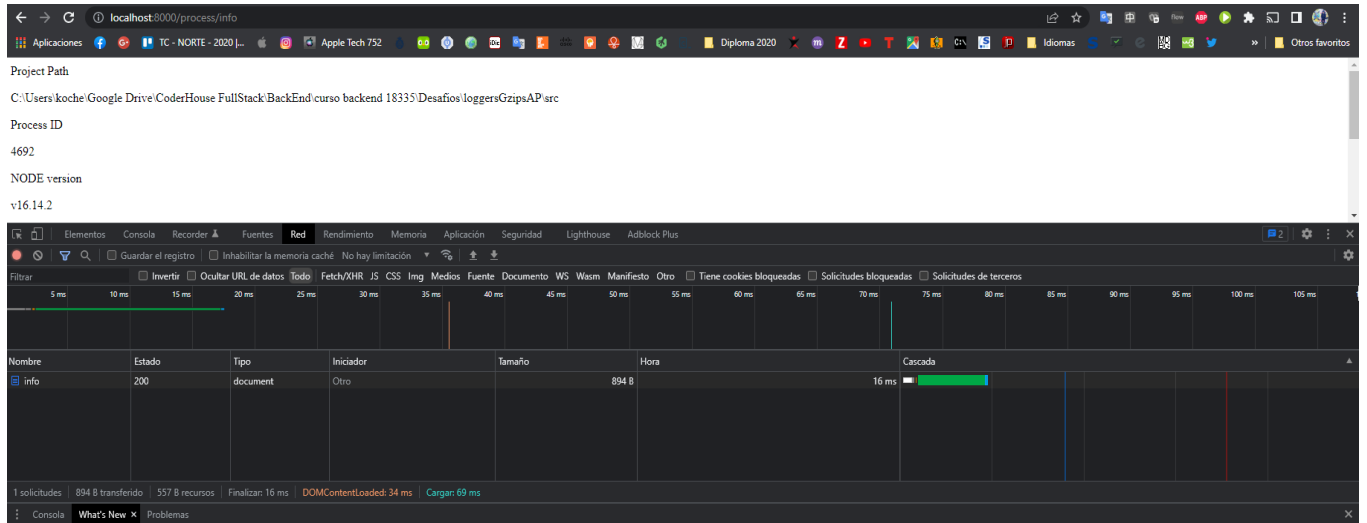
Curso: Programación Backend

Comisión: 18335

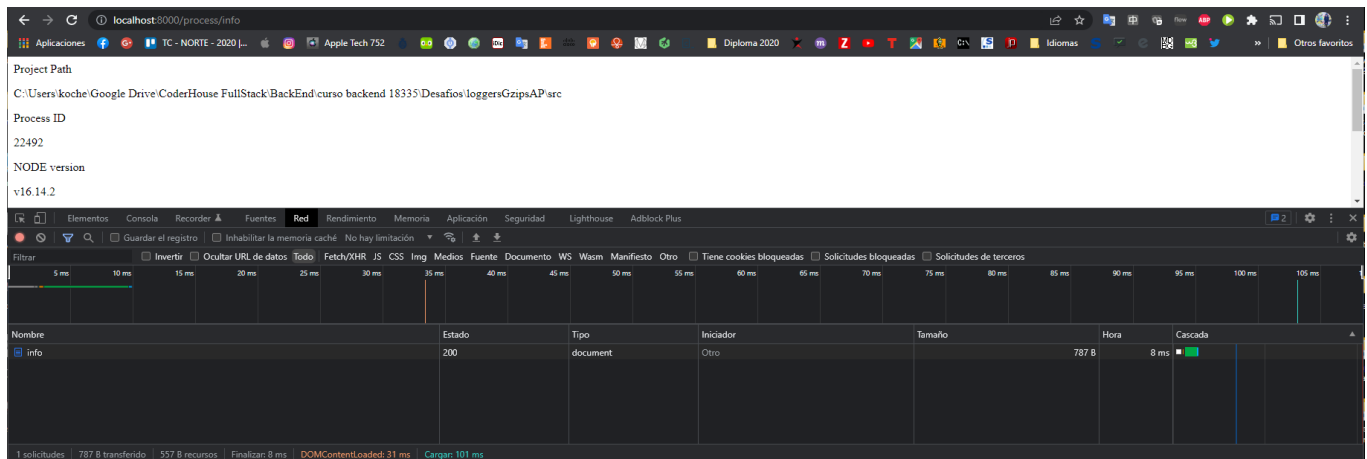
Alumno: José Carrizo

Año: 2022

Ruta: /process/Info [sin Gzip]



Ruta: /process/Info [con Gzip]



Profiling: Node built-in	
Ruta:/process/api/Info	Test: Autocannon, 20 peticiones, 100 conexiones

```
PS C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\src> node benchmark
Running test with autocannon...
Running 20s test @ http://localhost:8000/process/api/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	0 ms	0 ms	0 ms	0 ms	0 ms	0 ms	0 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	0	0	0	0	0
Req/Sec	749	749	1024	1050	997.05	73.38	749
Bytes/Sec	577 kB	577 kB	788 kB	809 kB	768 kB	56.6 kB	576 kB

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

20k requests in 20.04s, 15.4 MB read

Profiling: Node build-in	
Ruta: /process/api/Info	Test: OX

node app.js -p

search functions

~Long.negative C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\bson\lib\long.js:612:38 0% on stack, 0.04% stack top

~Long.negative C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\bson\lib\long.js:618:35 0% on stack

~Long.fromNumber C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\bson\lib\long.js:130:32 0% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\bson\lib\parser\deserializer.js:1:1 0% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\bson\lib\bson.js:1:1 0.1% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\mongodb\lib\bson.js:1:1 0.1% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\mongodb\lib\utils.js:1:1 0.1% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\mongodb\lib\operations\insert.js:1:1 0.1% on stack, 0.04% stack top

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\mongodb\lib\admin.js:1:1 0.2% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\mongodb\lib\index.js:1:1 0.4% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\connect-mongo\build\main\lib\MongoStore.js:1:1 0.4% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\node_modules\connect-mongo\build\main\lib\index.js:1:1 0.4% on stack

~(anonymous) C:\Users\koche\Google Drive\CoderHouse FullStack\BackEnd\curso backend 18335\Desafios\loggersGzipAP\src\app.js:1:1 4.5% on stack, 0.09% stack top

all stacks

Tiers Merge Optimized Unoptimized

node deps core module minimize native fs zlib crypto util

Profiling: Node built-in	
Ruta: /process/api/Info	Test: Artillery

```

-----
Metrics for period to: 13:24:10(-0300) (width: 1.404s)
-----

http.codes.200: ..... 1000
http.request_rate: ..... 719/sec
http.requests: ..... 1000
http.response_time:
  min: ..... 1
  max: ..... 25
  median: ..... 10.1
  p95: ..... 16.9
  p99: ..... 19.1
http.responses: ..... 1000
-----
Summary report @ 13:24:02(-0300)
-----

http.codes.200: ..... 1000
http.request_rate: ..... 360/sec
http.requests: ..... 1000
http.response_time:
  min: ..... 1
  max: ..... 25
  median: ..... 10.1
  p95: ..... 16.9
  p99: ..... 19.1
http.responses: ..... 1000
vusers.completed: ..... 20
vusers.created: ..... 20
vusers.created_by_name.0: ..... 20
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 246.9
  max: ..... 668.4
  median: ..... 539.2
  p95: ..... 658.6
  p99: ..... 658.6

```

Conclusión:

La instrucción console.log es de tipo bloqueante; por lo tanto, genera retrasos en la ejecución de instrucciones posteriores puesto que consume muchos recursos.

Se recomienda evitar su uso y aplicar un sistema de logs como Winston u otro alternativo.