

## Informe de pruebas del laboratorio 32: Loggers-gzip-analisis-de-performance

Nombre: Guillermo Dante Matos Cuba

Luego de realizar las configuraciones de logging se ejecutaron las pruebas sobre la aplicación funcionando en modo fork como se observa en la imagen:

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_Coderhouse
\Parte2\Clases31-32\Clase32.desafio> nodemon .\b0.server.js
[nodemon] 2.0.16
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node .\b0.server.js`
debug: [productos.LDB.model.js] productosLDBModel debug [18-10-22 12:10:01.719 AM]
info: [indexDAO.js][FactoryDAO] typeDB mongoLDB [18-10-22 12:10:01.768 AM]
verbose: [indexDAO.js][FactoryDAO] typeDB Generate DAO with Mongo Ldb: mongoLDB [18-10-22 12:10:01.771 AM]
info: [b0.server.js] [Puerto] 8081 [18-10-22 12:10:02.064 AM]
info: [b0.server.js][Pruebas] [na] Test Info message [18-10-22 12:10:02.070 AM]
debug: [b0.server.js][Pruebas] [na] Test Debug: {"cookie":{"path":"/","_expires":"2022-09-30T17:35:03.710Z","originalMaxAge":60000,"httpOnly":true
,"secure":false},"passport":{"user":"yo"}} [18-10-22 12:10:02.072 AM]
verbose: [b0.server.js][Pruebas] [na] Test Verbose message [18-10-22 12:10:02.073 AM]
warn: [b0.server.js][Pruebas] [na] Test Warn message [18-10-22 12:10:02.075 AM]
error: [b0.server.js][Pruebas] [na] Test Error message [18-10-22 12:10:02.076 AM]
info: [b0.server.js] [listen] Server desplegado en http://127.0.0.1:8081 [18-10-22 12:10:02.087 AM]
warn: [b0.server.js] undefined PID WORKER 9184 [18-10-22 12:10:02.089 AM]
```

### 1 Perfilamiento del servidor con -prof

Los comandos usados fueron:

1. node -prof b0.server.js

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_Coderhouse
\Parte2\Clases31-32\Clase32.desafio> node --prof .\b0.server.js
```

2. artillery --count 50 -n 20 <http://localhost:8081/operaciones/info> > archive-destino.txt

```
PS C:\Users\GUILLERMO\artillery-results> artillery quick --count 50 -n 20 http://localhost:8081/operaciones/info > lab32-results_fork_2022-10-17T214323.txt
```

Luego de decodificar se generó el archivo r2-test-prof-ready.txt

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_Coderhouse
\Parte2\Clases31-32\Clase32.desafio> write.exe .\r2-test-prof-ready.txt
```

El resultado es:

```
[Summary]:
  ticks  total  nonlib   name
    80     0.1%   98.8%  JavaScript
     0     0.0%    0.0%    C++
    41     0.0%   50.6%     GC
 123790   99.9%
     1     0.0%
      1     0.0%   Unaccounted

[C++ entry points]:
  ticks  cpp  total   name

[Bottom up (heavy) profile]:
Note: percentage shows a share of a particular caller in the total
amount of its parent calls.
Callers occupying less than 1.0% are not shown.

  ticks parent  name
 121651   98.2%  C:\WINDOWS\SYSTEM32\ntdll.dll

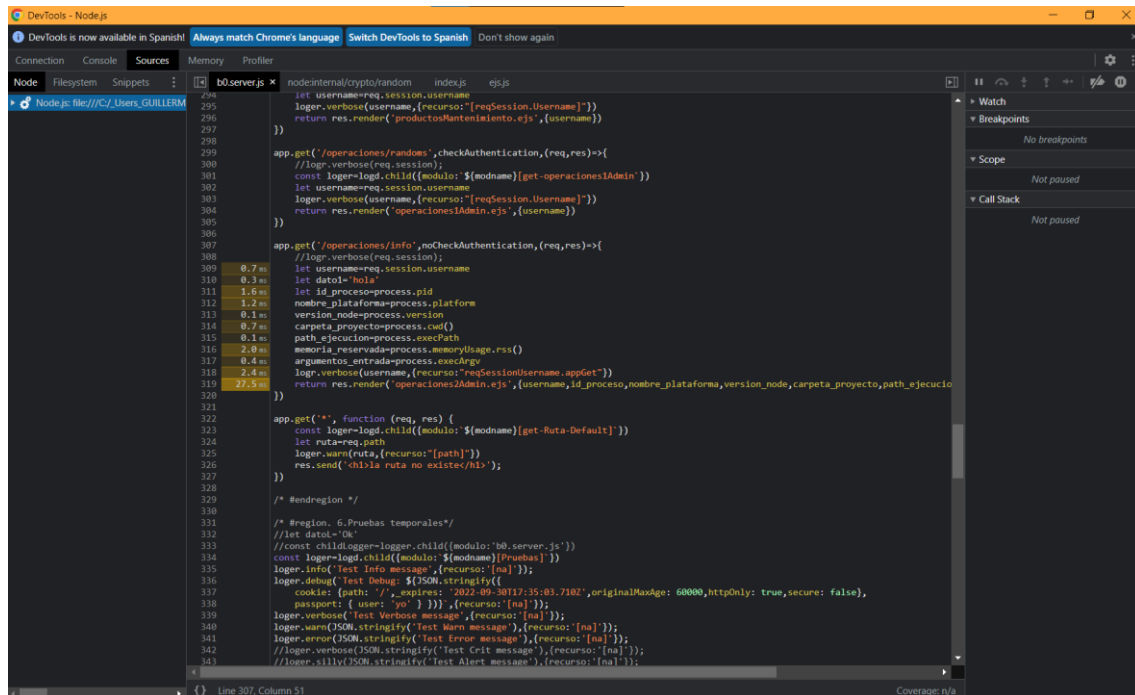
  2118     1.7%  C:\Program Files\nodejs\node.exe
```

## 2 Perfilamiento con en modo inspect inspect

El comando usado fue

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_coderhouse\Parte2\Clases31-32\Clase32.desafio> node --inspect .\b0.server.js
```

En la imagen del resultado (obtenido con chrome) se puede observar como la ejecución del archivo operaciones2Admin.ejs de la ruta /operaciones/info es uno de los procesos que toma más tiempo en ejecutarse con 27.5ms



## 3 Diagrama de flama

Luego de crear el archivo benchmark.js con el código para el autocannon se usaron los comandos:

1.npm start

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_coderhouse\Parte2\Clases31-32\Clase32.desafio> npm start
```

2.npm test

```
PS C:\Users\GUILLERMO\Desktop\3.Conocimiento tecnico especializado\Desarrollo_Software\1.HTML_CSS_JS_TS_DB\3.S2.NodeJS_TypeScript_DB\NodeJS_coderhouse\Parte2\Clases31-32\Clase32.desafio> npm test
```

En la imagen del resultado se puede observar que no hay procesos bloqueantes debido a que el color de los procesos es el mismo en la mayoría de ellos. Además los procesos de la parte más alta siempre se ejecutan rápidamente (su ancho de gráfico es pequeño comparado con los procesos que continúan en vertical hacia abajo)

```
node b0.server.js
```

