**Informe**

**Temas:** gzip, profiling, 0x, artillery, autocannon, inspect.

**Introducción:**

Los análisis se van a realizar en en dos rutas del proyecto, /info y /*api/*randmon.

En la ruta /*info se realizo análisis comparativo de la ruta con/*sin gzip(compress).

Ademas se realizaron dos análisis con profiling recibiendo solicitudes con artillery, agregando en un caso un console.log y en otro sin el console.log

En la ruta /*api/*randmon se realizaron dos análisis con 0x y con inspect, ambos recibiendo solicitudes con autocanon

**Localización:**

En en la carpeta **análisis** que esta en la raíz del proyecto, podemos encontrar 3 carpetas, el análisis de compres y una carpeta por cada ruta analizada, dentro de ellas estarán los reportes de todos los análisis

**Análisis:**

**compress:** se midió el peso de la respuesta de la ruta /info sin el middleware compres y luego con el midleware activado.

/info – sin compres: 2.5KB

/info – con compres: 1.4KB

**/info:** Detalle de los resultados de profiling.

**-Con console.log:**

Summary: 2611 99.6% Shared libraries

ticks total name

2158 82.3% C:\Windows\SYSTEM32\ntdll.dll

452 17.2% C:\Program Files\nodejs\node.exe

1 0.0% C:\Windows\System32\KERNELBASE.dll

**-Sin console.log:**

Summary: 6612 99.9% Shared libraries

ticks total name

6281 94.9% C:\Windows\SYSTEM32\ntdll.dll

330 5.0% C:\Program Files\nodejs\node.exe

1 0.0% C:\Windows\System32\KERNELBASE.dll

**/api/randmon:** Detalle de los resultados de profiling.

[Summary]:

ticks total name

5701 57.0% JavaScript

4307 43.0% Shared libraries

[JavaScript]:

ticks total name

5692 99.8% \controllers\random.controller.js:3:17

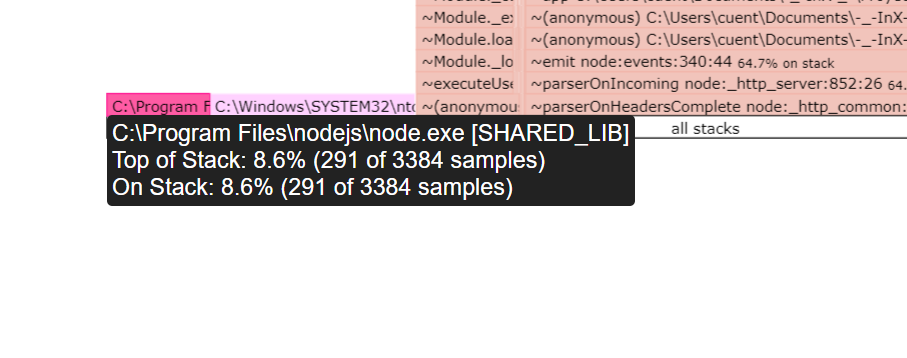
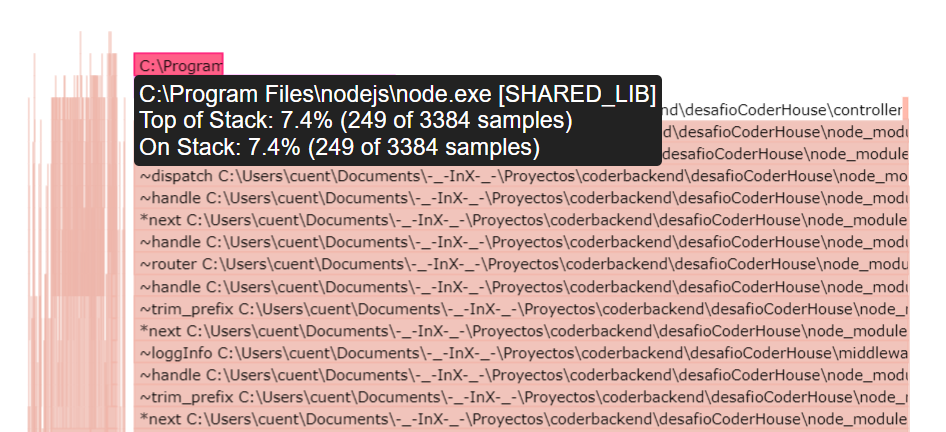
[Shared libraries]:

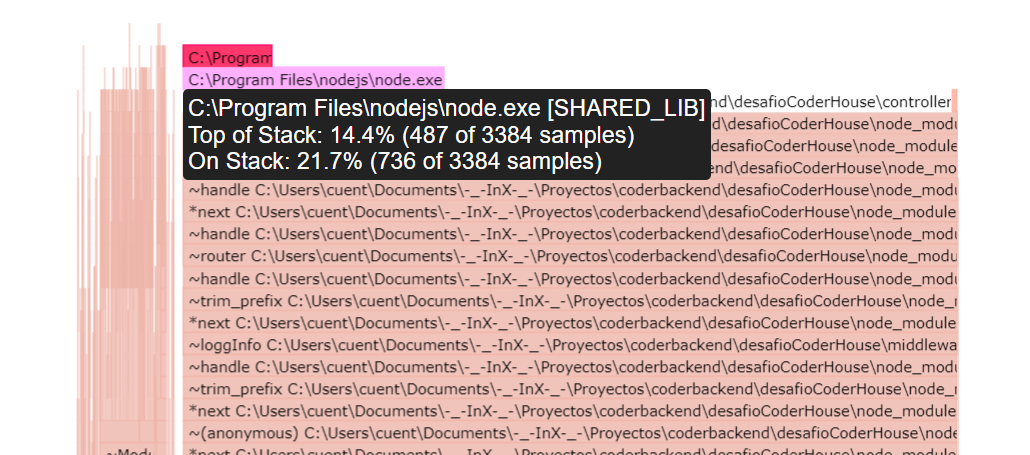
ticks total name

3642 36.4% C:\Program Files\nodejs\node.exe

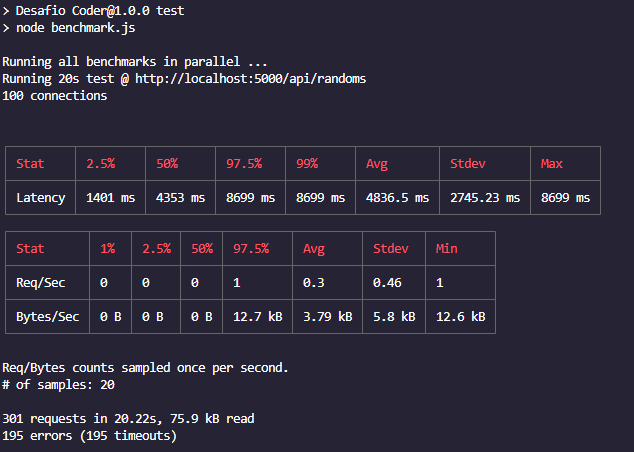
660 6.6% C:\Windows\SYSTEM32\ntdll.dll

**/api/randmon:** Detalle de los resultados de 0x.





**/api/randmon:** Resultados de autocannon.

****

**/api/randmon:** Resultados de inspect.



**Conclusión**:

**/info:**

En el caso de la ruta info podemos observar que hay muchos mas cantidad de golpes(ticks) en las pruebas sin console.log, aproximadamente 3 veces que con console.log.

Ademas en ambos casos vemos que los ticks, el 99,8% corresponde a librerías, a su vez casi el 100% de los ticks de la librerías corresponde a \SYSTEM32\ntdll.dll y en segundo lugar esta \nodejs\node.exe

**/api/randmon:**

En el caso del **profiling**, en esta ruta el total de ticks están divididos de manera pareja entre JavaScript y Librerías.

En el caso de JS el 99,8% de los ticks son a: \controllers\random.controller.js.

Y en el caso de las librerías, las dos mas notables son \nodejs\node.exe y \SYSTEM32\ntdll.dll

El análisis de **0x**, muestra que en la pila hay una demora critica en la librería node.exe

Por ultimo en el resultado de muestra **inspect**, se puede ver en los consumos de tiempo que express testa muy presente, y también se observa que los logs(realizado con winston), están presentes, no representan un gran consume de tiempo pero tiene un consumo de tiempo a tener en cuenta