**Reporte Desafío 16**

1. Incorporar al proyecto de servidor de trabajo la compresión gzip. Verificar sobre la ruta /info con y sin compresión, la diferencia de cantidad de bytes devueltos en un caso y otro.

Respuesta sin compresión

Graphical user interface, text, application, email

Description automatically generated

Respuesta utilizando compresión

Graphical user interface, text, application, email

Description automatically generated

**Conclusión:** Se observa una disminución de 0,05 KB en el tamaño de la respuesta del método info al utilizar compression.

2) El perfilamiento del servidor, realizando el test con --prof de node.js. Analizar los resultados obtenidos luego de procesarlos con --prof-process. Utilizaremos como test de carga Artillery en línea de comandos, emulando 50 conexiones concurrentes con 20 request por cada una. Extraer un reporte con los resultados en archivo de texto.

Resultado método info sin console log:

A screenshot of a computer

Description automatically generated with low confidence

Resultado método info con console log:

Text

Description automatically generated with low confidence

Conclusión: Se observa que el método info que realiza el console.log tiene un mayor número de ticks que la versión que no realiza el console.log. Inicialmente se esperaba que se diera la situación contraria dado en la segunda configuración el código ejecuta una operación adicional.

3) Luego utilizaremos Autocannon en línea de comandos, emulando 100 conexiones concurrentes realizadas en un tiempo de 20 segundos. Extraer un reporte con los resultados (puede ser un print screen de la consola)

Resultado método info sin console log:

**A screenshot of a computer

Description automatically generated with medium confidence**

Resultado método info con console log:

**A screenshot of a computer

Description automatically generated with medium confidence**

Conclusión: Se observa que el método info que tiene la operación adicional console.log puede atender menos requests por segundos y puede trasmitir más información por segundo en línea con lo que se esperaba. Así también tiene una mayor latencia como se esperaba.

4) El perfilamiento del servidor con el modo inspector de node.js --inspect. Revisar el tiempo de los procesos menos performantes sobre el archivo fuente de inspección.

Resultado método info sin console log:

**Text

Description automatically generated**

Resultado método info con console log:

**Text

Description automatically generated**

**Conclusión:** Se observa que el código que más demora en ejecutar lo hace en menos tiempo en el caso donde se imprime por consola el resultado del método info. No se observa/identifica dentro del listado del este profile un tiempo significativo a la impresión del resultado en el segundo caso.

5) El diagrama de flama con 0x, emulando la carga con Autocannon con los mismos parámetros anteriores.

Resultado método info sin console log:

A picture containing graphical user interface

Description automatically generated

Resultado método info con console log:

A picture containing website

Description automatically generated

**Conclusión:** Luego de una inspección visual no se observan mayores diferencias entre el código que imprime en consola el resultado del método info con el que no lo hace. Se hubiese esperado ver alguna diferencia en la cantidad de peaks o la duración del stack total.