**Ejercicio 1 – Script de Pruebas de Carga**

**EJERCICIO 1:**

**INSTRUCCIONES:**

1. Para la resolución del ejercicio, se debe utilizar una de las siguientes herramientas: K6 o JMeter.

2. Al finalizar el ejercicio se debe subir en un repositorio github público. El ejercicio debe contener archivos, scripts, reportes y cualquier otro insumo que sustente su implementación para que pueda ser reproducido por la persona a revisar.

**Adicional debe contener:**

Un archivo **readme.txt** con las instrucciones paso a paso de ejecución (debe especificar las versiones de las tecnologías a usar).

Un archivo **conclusiones.txt** con los hallazgos y conclusiones del ejercicio

**EJERCICIO:**

Realizar una prueba de carga del servicio de login, para efectos del ejercicio, se brindará el siguiente CURL:

curl --location --max-time 60 '[https://fakestoreapi.com/auth/login'](https://fakestoreapi.com/auth/login%27) ^  
--header 'Content-Type: application/json' ^  
--data '{  
  "username": "user",  
  "password": "passwd"  
}'

Los datos de entrada que se deben parametrizar desde un archivo '.csv' son:

user,passwd

donero,ewedon

kevinryan,kev02937@

johnd,m38rmF$

derek,jklg\*\_56

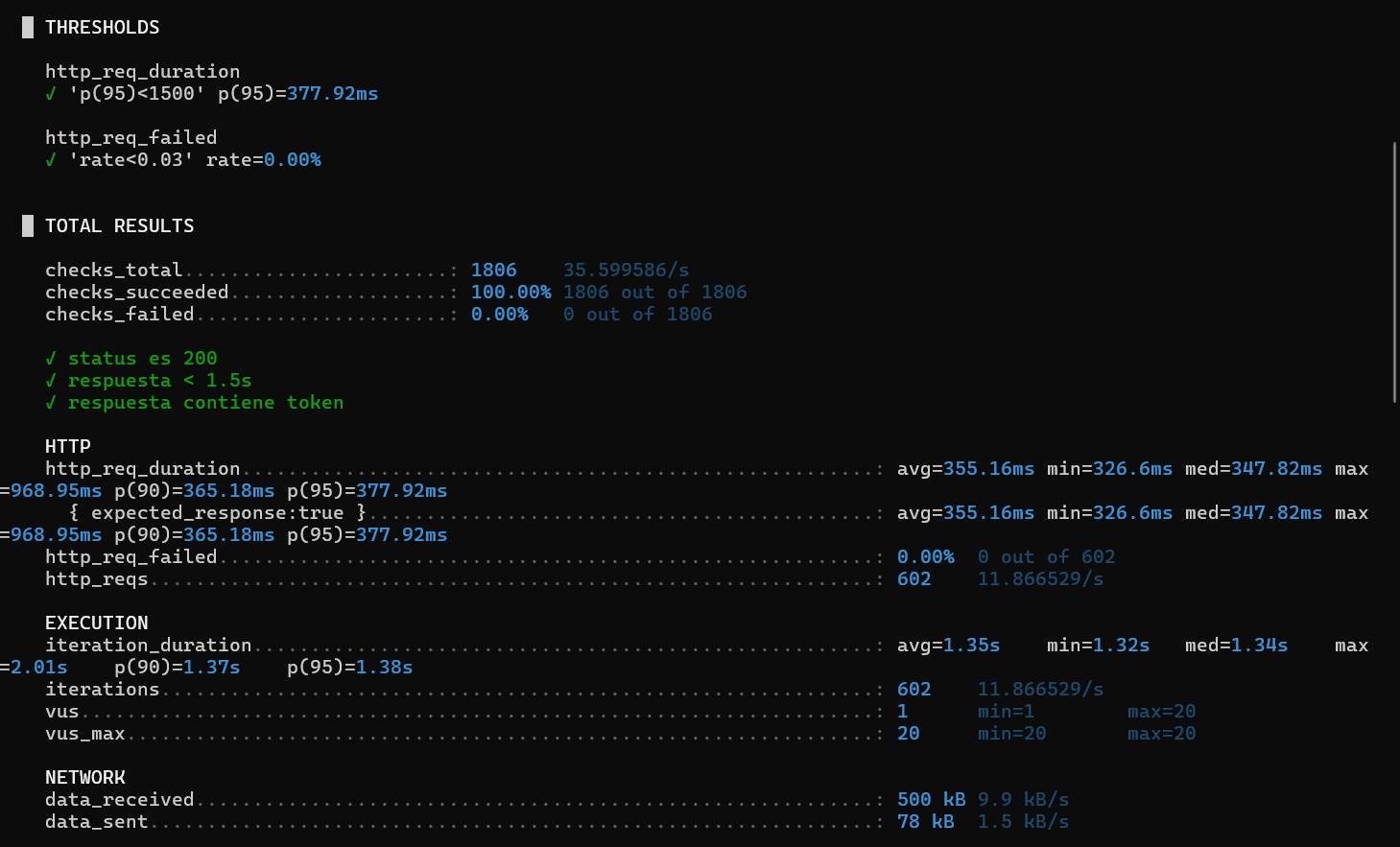
mor\_2314,83r5^\_

El escenario de la prueba al menos debe alcanzar los 20 TPS y debe tener las siguientes validaciones:

• El tiempo de respuesta permitido es de máximo 1,5 segundos.

• Tasa de error aceptable, menor al 3% del total de peticiones.

Conclusiones:



La prueba simuló 1806 peticiones, con una tasa de 35.6 peticiones/segundo. Todas las respuestas fueron exitosas, con código 200.

El p95 de http\_req\_duration fue de 377.92 ms, por debajo del umbral fijado en <1500 ms, lo cual indica buen rendimiento. No hubo peticiones fallidas (http\_req\_failed=0%). La duración de cada iteración (iteration\_duration) fue de 1.35 s y se mantiene estable.

La respuesta cumple con los requisitos esperados:

status=200

respuesta < 1.5 s

El uso de red fue mínimo: 500 kB recibidos y 78 kB enviados en total

No hubo fallos en las peticiones, lo que muestra un comportamiento confiable de la API bajo la carga actual, existe estabilidad.

Y las latencias están bien controladas (<400 ms para el p95), lo que es bastante aceptable para la mayoría de APIs RESTful, hay buen rendimiento.

**Ejercicio de Análisis de Resultados :**

**EJERCICIO 2**

**INSTRUCCIONES:**

1. Dado el siguiente cuadro de resultados de una prueba de carga detallados con los valores obtenidos luego de la ejecución de la prueba (Archivo: textSummary.txt), necesitamos realizar un análisis de los resultados encontrados:

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.

**Información Adicional:**

**El monitoreo** de la prueba, arrojó el siguiente driagrama:

Imagen que contiene Gráfico

El contenido generado por IA puede ser incorrecto.

Conclusión:

El 97% de respuestas fueron correctas (269.891 peticiones) y el 2.44% de respuestas fallidas (6759 peticiones), esto indica un alto grado de éxito en las pruebas y las cifras reflejan una buena eficiencia en transmisión para el volumen total de las solicitudes.

En los tiempos se puede observar que la mayoría del tiempo se consume esperando la respuesta del servidor (861.21ms) también no hay cuellos de botella en la red o conexión TLS (la mayoría tienen 0s).

En los errores http, 769 errores 4xx (cliente) y 5.987 errores 5xx (servidor), concentrados en la primera etapa de la prueba, lo que señala un problema en el backend al manejar carga inicial.

Tiene una concurrencia máxima: 140 usuarios virtuales simultáneos.

El sistema estable hasta cierto punto, con un 97% de éxito, pero no cumple criterios estrictos de alta disponibilidad (>99%).

En el segundo gráfico refleja que se mantuvieron 140 usuarios de manera constante, sin embargo las solicitudes HTTP por segundo (http\_reqs) experimentaron una caída abrupta alrededor de las 01:50 y se mantuvieron muy bajas hasta aproximadamente las 02:00, a pesar de que el número de usuarios no cambió, posteriormente las http\_reqs se recuperaron y mantuvieron un rendimiento cercano a 80 solicitudes por segundo hasta el final de la prueba, aunque los usuarios se mantuvieron activos indica que el servidor dejó de responder su capacidad de atender solicitudes en ese intervalo, confirmando un problema de escalabilidad o una interrupción temporal en el backend que coincide con el aumento de carga de usuarios.

Tiempos de respuesta aceptables, aunque el percentil 95 supera 1.5 s, lo cual puede ser percibido como lento por algunos usuarios.

Una recomendación se puede optimizar el backend, especialmente en el procesamiento en la etapa inicial de las solicitudes.