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' ^
--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.

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:

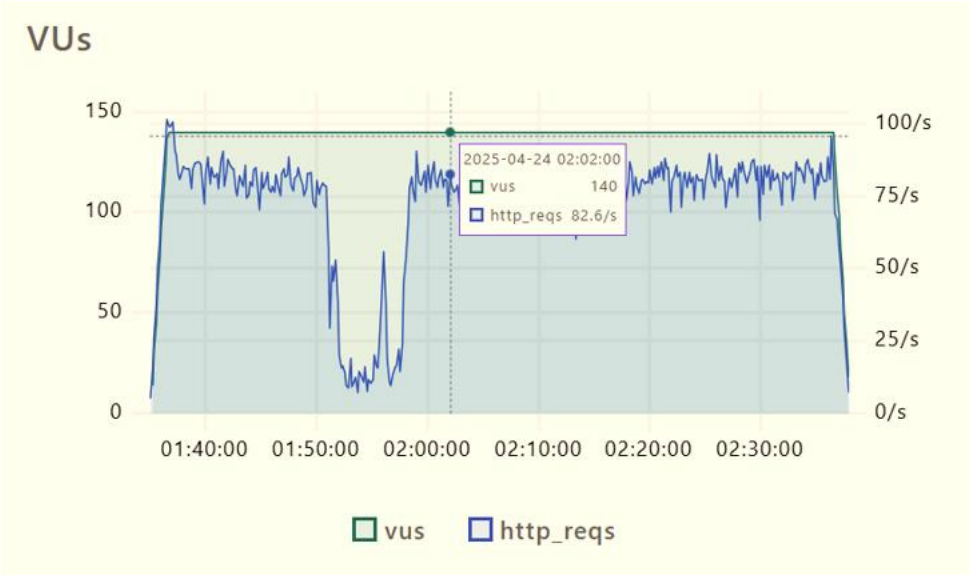
textSummary.txt

```
✗ App Transaction Balance response was OK
  ↳  97% — ✓ 269891 / ✗ 6759

  setup

✓ checks.........................: 97.55% ✓ 269891    ✗ 6759
  data_received..................: 842 MB 223 kB/s
  data_sent......................: 588 MB 156 kB/s
  failed_request.................: 2.44%  ✓ 6759      ✗ 269891
  http_req_blocked...............: avg=10.97µs  min=0s       med=0s       max=35.02ms p(90)=0s      p(95)=0s
  http_req_connecting............: avg=3.3µs    min=0s       med=0s       max=11.82ms p(90)=0s      p(95)=0s
✓ http_req_duration..............: avg=861.68ms min=191.86ms med=613.42ms max=29.93s  p(90)=1.28s   p(95)=1.57s
  ✓ { expected_response:true }...: avg=735.84ms min=244.92ms med=600.7ms  max=26.72s  p(90)=1.22s   p(95)=1.42s
✓ http_req_failed................: 2.44%  ✓ 6759      ✗ 269891
  http_req_receiving.............: avg=424.03µs min=0s       med=320.7µs  max=39.58ms p(90)=988.4µs p(95)=1.05ms
  http_req_sending...............: avg=43.22µs  min=0s       med=0s       max=31.25ms p(90)=0s      p(95)=517µs
  http_req_tls_handshaking.......: avg=7.36µs   min=0s       med=0s       max=27.02ms p(90)=0s      p(95)=0s
  http_req_waiting...............: avg=861.21ms min=191.86ms med=613.01ms max=29.93s  p(90)=1.28s   p(95)=1.57s
✓ http_reqs......................: 276650 73.176857/s
  iteration_duration.............: avg=1.86s    min=0s       med=1.61s    max=30.94s  p(90)=2.29s   p(95)=2.57s
✓ iterations.....................: 276650 73.176857/s
  vus............................: 2      min=2       max=140
  vus_max........................: 140    min=140     max=140
  y_failed_request_stage_0_HTTP5xx...: 1      0.000265/s
  y_failed_request_stage_1_HTTP4xx...: 769    0.203409/s
  y_failed_request_stage_1_HTTP5xx...: 5987   1.583625/s
  y_failed_request_stage_2_HTTP5xx...: 2      0.000529/s
```

Información Adicional:
El monitoreo de la prueba, arrojó el siguiente driagrama:



Que se debe entregar:
Un archivo **InformeResultados.doc** con los hallazgos, conclusiones y recomendaciones de la prueba realizada, incluyendo datos relevantes del diagrama obtenido de la relación entre los usuarios virtuales y el número de peticiones por segundo.