

Performance Documentation

1. Alcance de las pruebas de performance

URL base: <https://petstore.swagger.io/v2>

Endpoints evaluados y razones de selección

- **GET** /pet/findByStatus
- **GET** /pet/{petId}
- **POST** /pet
- **DELETE** /pet/{petId}
- **POST** /store/order
- **GET** /store/order/{orderId}
- **DELETE** /store/order/{orderId}
- **POST** /user
- **GET** /user/{username}
- **DELETE** /user/{username}

La cobertura de las pruebas y endpoints se determino con base a los siguientes criterios:

- La importancia del endpoint en la store respecto a su funcionamiento en el manejo de la store.
- La frecuencia en que se podrían usar los endpoints, es decir, es posible que se consulten mas los pets que hay en la store en relación a que se suban imagenes de cada pet.
- La experiencia del usuario que maneja la tienda con el fin de que no se vea muy afectado en caso de que alguno de los endpoints principales fallen.

Tipos de pruebas realizadas:

- **Prueba de carga:** Simulación de múltiples usuarios concurrentes.
- **Prueba de estrés:** Envío de peticiones más allá de la capacidad esperada.
- **Prueba de pico:** Aumento inesperado en la cantidad de usuarios concurrentes.

2. Performance framework

Se implementó la herramienta k6 teniendo en cuenta el stack tecnologico de la compañía. Se tuvo en cuenta que esta herramienta se maneja con el lenguaje de javascript lo que facilita su implementación.

Configuración:

1. Tener configurado el ambiente de node e instalar k6.
2. Instalar k6-html-reporter.

Ejecución:

1. Definir que test se va a ejecutar y correr el comando k6 run tests/nombreEndpoint/{nombreTest}.js . Los tests se encuentran en la carpeta tests/ y se debe elegir un archivo para correr el test.
2. Generar reporte html con el comando node generateReport.js
3. Revisar reporte generado en la carpeta report.html.

Parámetros clave

Estos parametros son variables a lo largo de la prueba y la idea es que se puedan modificar cuando se va a correr el test para analizar diferentes comportamientos.

- **VUs:** 50 - 200 dependiendo del endpoint.
- **Duración:** 10s - 60s.
- **Tasa de solicitudes:** Dependiendo del test.

3. Análisis de resultados

Por temas prácticos se van a analizar los resultados con el endpoint post/user.

POST /user

Tipo de prueba: Es una prueba de carga escalonada en la que indica diferentes VUs durante tiempos diferentes.

Ejecución:

La prueba se ejecuta 3 veces con diferentes valores de tiempos y VUs para mirar su comportamiento.

Ejecuciones	Tiempo	VUs
1	10s, 20s, 20s, 10s	10, 50, 100, 0
2	20s, 30s, 30s, 20s	50, 100, 100, 0
3	5s, 5s, 5s, 5s	50, 100, 200, 0

Resultados:

En el proyecto se encuentran los tres reportes de las tres ejecuciones de las pruebas al endpoint, estos comienzan con el nombre postUser.html.

- En las tres ejecuciones se evidenció que las solicitudes tuvieron una respuesta de 200 o 201, por lo que se concluyó que el endpoint responde sin fallos.
- El tiempo de respuesta es bastante estable en las tres ejecuciones con un promedio de 88.94 ms, 89.26 ms y 88.52 ms respectivamente, lo que indica que la carga no afectó significativamente el rendimiento del servidor.
- El percentil 95 del tiempo de respuesta indica que la mayoría de solicitudes responde en menos de 105.62 ms lo que indica buena estabilidad del endpoint.
- En la tercera ejecución hubo mayor cantidad de UVs que fue 23, y el mayor tiempo de bloqueo se dio en este intervalo. Por lo tanto, podemos concluir que el endpoint tuvo mayor espera antes de enviar las solicitudes, probablemente debido a la mayor cantidad de solicitudes simultáneas generadas en esta prueba.
- La cantidad de datos enviados y recibidos aumentó en la segunda ejecución respecto a la primera, pero disminuyó en la tercera. Se recibieron 1,202,435 bytes en la primera prueba, 2,094,666 bytes en la segunda y 1,399,945 bytes en la tercera, lo que indica que el volumen de datos puede variar dependiendo del tráfico y la concurrencia sin afectar la estabilidad del endpoint.
- El tiempo de handshake TLS (http_req_tls_handshaking) mostró un incremento en la tercera ejecución con un promedio de 21.78 ms, comparado con 7.50 ms en la primera y 3.49 ms en la segunda, lo que sugiere que el aumento de usuarios concurrentes pudo haber afectado levemente la negociación de seguridad en la conexión.