Pruebas de Carga

En este documento detallamos las pruebas a realizar, el análisis de los resultados y la tecnología empleada.

Estructuración del plan de pruebas

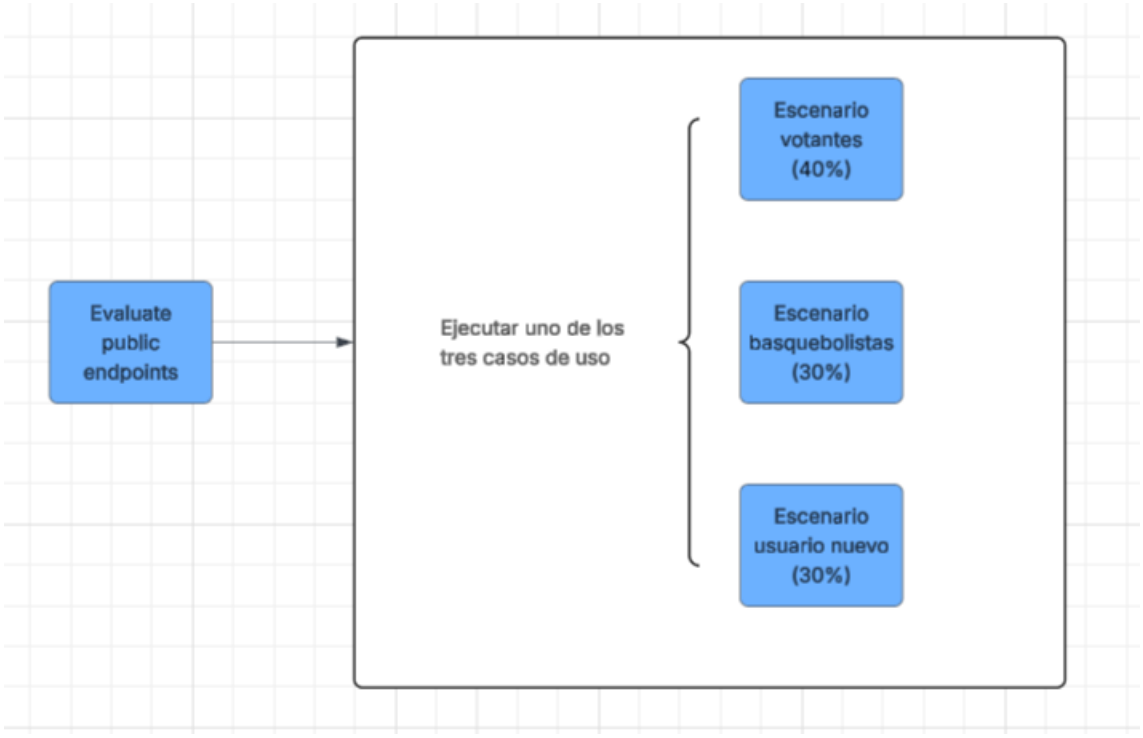
1. Evaluación básica a nivel de endpoints

Evaluar de forma aislada los tiempos de respuesta de cada endpoint.

2. Escenarios realistas de uso de la aplicación

Se evalúan varios escenarios que representan posibles casos de uso común para los usuarios de la aplicación.

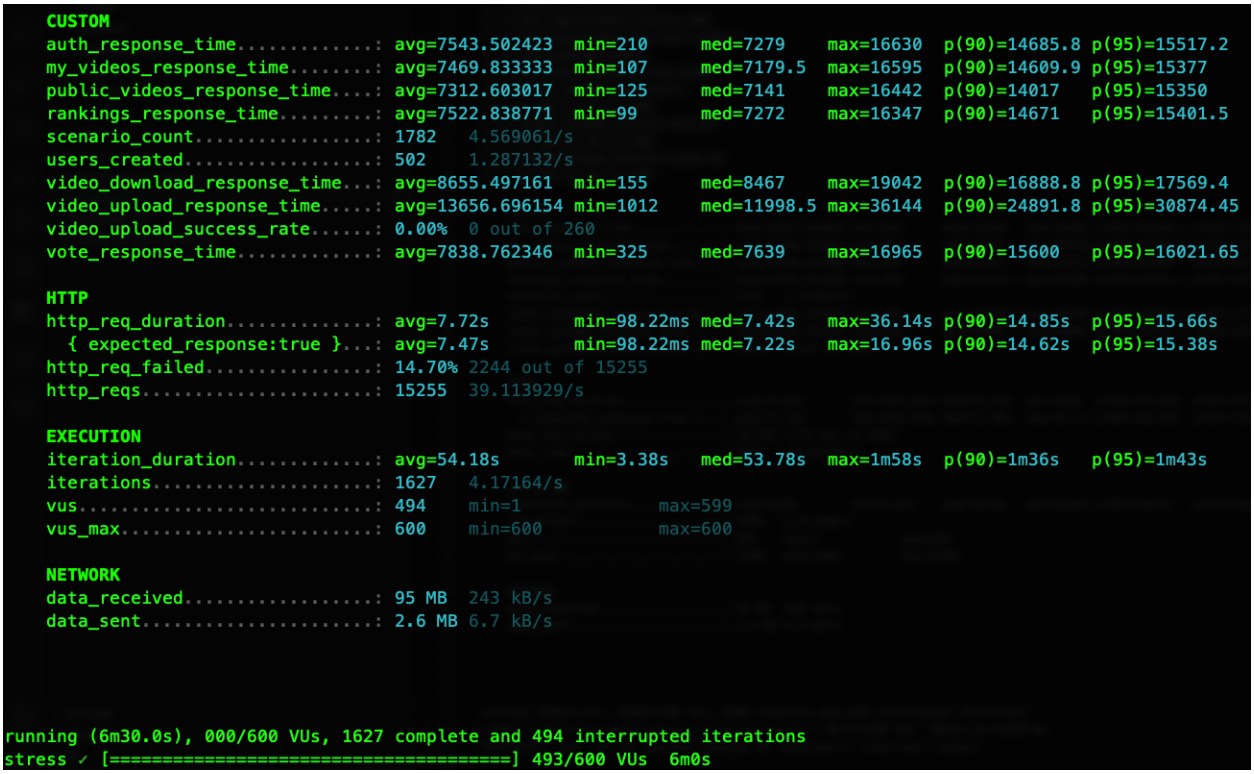
- Votantes: un usuario inicia sesión, mira el ranking, ve algunos videos y vota
- Uso de basquetbolistas: un usuario inicia sesión, sube un video y va a la lista de video subidos a mirar el estado de sus videos.
- Usuarios nuevos: un usuario crea una cuenta, inicia sesión y mira algunos videos



Carga progresiva

La prueba de carga se configuró con un esquema de incremento progresivo de usuarios virtuales (ramping). Durante seis etapas consecutivas de un minuto cada una, se agregan 100 usuarios por etapa, iniciando con 100 usuarios concurrentes y aumentando linealmente hasta alcanzar un máximo de 600 usuarios al sexto minuto. Este enfoque permite observar cómo responde el sistema bajo un crecimiento sostenido de la demanda, evaluando tanto su capacidad de escalamiento como la estabilidad del rendimiento a medida que aumenta la concurrencia.

Resultados



Análisis de los Resultados

Durante la evaluación básica a nivel de endpoints se identificó que la mayoría de las operaciones críticas presentaron tiempos de respuesta elevados bajo condiciones de carga progresiva. Los endpoints de autenticación (/api/auth/login), videos públicos (/api/public/videos), rankings (/api/public/rankings) y videos de usuario (/api/videos) tuvieron tiempos de respuesta promedio entre 7 y 8 segundos, con valores en el percentil 95

superiores a los 15 segundos. Esto evidencia que, conforme aumentó la concurrencia, el sistema no pudo mantener latencias aceptables, lo que impacta directamente la experiencia del usuario.

En los escenarios realistas de uso de la aplicación, se observaron los siguientes comportamientos:

- Votantes: los usuarios lograron iniciar sesión y consultar rankings y videos, pero con latencias superiores a lo esperado (≈7–8 segundos en promedio), lo cual afecta la fluidez de la interacción. Aunque el proceso de votación se ejecutó, en muchos casos la operación presentó demoras notables.
- Basquetbolistas: este escenario fue el más crítico. Las cargas de video tuvo tiempos promedio de 13.6 segundos y un pico máximo de 36 segundos. Este resultado evidencia que el sistema presenta debilidades para soportar operaciones pesadas de subida de archivos bajo alta concurrencia con la instancia unica.
- Usuarios nuevos: los procesos de creación de cuenta y exploración de videos sí se realizaron, pero nuevamente con tiempos de respuesta muy altos, comparables a los del resto de endpoints (7–8 segundos en promedio).

En términos globales, la prueba generó 15,255 solicitudes HTTP, de las cuales un 14.7% fallaron (2,244 errores). Asimismo, se registraron 494 iteraciones interrumpidas, lo cual indica que múltiples usuarios virtuales no pudieron completar su flujo debido a fallas o tiempos de espera excesivos.

El sistema alcanzó su punto de estrés a partir de ~493 usuarios concurrentes, momento en el cual la degradación del servicio se hizo evidente: los tiempos de respuesta aumentaron abruptamente, las tasas de error crecieron y varios escenarios dejaron de completarse correctamente. Esto significa que la plataforma es capaz de sostener un volumen de usuarios concurrentes cercano a los 500, pero colapsa al acercarse al objetivo de 600.

En conclusión, los resultados muestran que el sistema no logró sostener la carga de 600 usuarios concurrentes definida en la prueba. Las altas latencias y el porcentaje significativo de reflejan limitaciones tanto en el backend como posiblemente en la infraestructura de soporte. Antes de escalar a escenarios mayor uso, será necesario realizar optimizaciones en el procesamiento de peticiones, mejorar la eficiencia de operaciones críticas como la carga de videos y evaluar la capacidad de los recursos de infraestructura disponibles.

Tecnologia Empleada

Para la ejecucion de las pruebas se utilizo k6, que es una herramienta de pruebas de carga y estrés orientada a desarrolladores, diseñada para evaluar el rendimiento y la resiliencia de aplicaciones y servicios web. Permite definir escenarios de prueba mediante scripts en JavaScript, lo que facilita la creación de flujos realistas que simulan el comportamiento de los usuarios finales. Una de sus principales ventajas es su bajo consumo de recursos y su capacidad para ejecutar pruebas de alta concurrencia de manera eficiente, ya sea en entornos locales o en la nube. Además, proporciona métricas detalladas como latencias, tasas de error y percentiles de tiempo de respuesta, lo que la convierte en una opción sólida para detectar cuellos de botella y validar la capacidad de la infraestructura antes de enfrentar escenarios de uso real.