Universidad de los Andes
Facultad de Ingeniería

DESARROLLO DE SOLUCIONES CLOUD

Proyecto Entrega 2:

Análisis Pruebas de Carga

INTEGRANTES

Ricardo Andres Leyva Osorio - r.leyva@uniandes.edu.co

Edda Camila Rodriguez Mojica - ec.rodriguez@uniandes.edu.co

Cristian David Paredes Bravo - c.paredesb@uniandes.edu.co

Andrea Carolina Cely Duarte - a.celyd@uniandes.edu.co

Juan Carlos Martinez Muñoz - jc.martinezm1@uniandes.edu.co

MAESTRÍA EN ARQUITECTURAS DE TECNOLOGÍAS DE INFORMACIÓN

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

UNIVERSIDAD DE LOS ANDES

BOGOTA D. C.

Universidad de los Andes
Facultad de Ingeniería

Contenido

Universidad de los Andes
Facultad de Ingeniería

Lista de Figuras

Lista de Tablas

Universidad de los Andes
Facultad de Ingeniería

INTRODUCCIÓN

Este informe resume el resultado de las pruebas de humo y las pruebas de carga escalonadas ejecutadas sobre dos rutas criticas establecidas dentro del **Sistema de Video y Ranking**

Las pruebas sobre las rutas o escenarios I**nteractivo** y **Upload** permiten visualizar el comportamiento del sistema a medida que aumentan los usuarios concurrentes

Se comparan los resultados frente a los criterios del Plan de Pruebas

Universidad de los Andes
Facultad de Ingeniería

1. ENTORNO E INFRAESTRUCTURA

1.1. Infraestructura del Generador de Carga:

i712700H, 16 GB RAM, Windows 11 Pro 22H2.

1.2. Infraestructura de la solución (AWS):

La aplicación se ejecuta en AWS sobre instancias EC2 tipo **t3.small para el frontend** (SPA público) y el **backend**, un **t3.large** para los *workers* que procesan tareas en segundo plano, y otros servicios de apoyo (RabbitMQ, Redis, MinIO), con el fin de escalar horizontalmente y reducir cuellos de botella.

Front (t3.small):	Servidor web/SPA público.
Back (t3.small):	API/servicio de negocio, accesible desde Front y Worker.
worker (t3.large):	Procesa tareas en segundo plano; más CPU/RAM que el resto.
rabbitmq (t3.small):	Message broker para desacoplar Back y Worker.
redis-ec2 (t3.small):	Caché/cola rápida o sesión. Sin IP pública, para uso interno
MinIO (t3.small):	Almacenamiento de objetos S3-compatible; tiene IP pública (expuesto).
bastion (t3.micro):	"jump host" para SSH hacia las instancias privadas.

Tabla 1. Infraestructura de la solución

Universidad de los Andes
Facultad de Ingeniería

2. RUTAS CRITICAS

2.1. Escenario Interactivo:

Objetivo: Validar latencia y unicidad de voto con navegación realista.
Flujo: Realizar login, listar los videos disponibles para voto, ejecutar tres votaciones y por ultimo consultar el ranking.

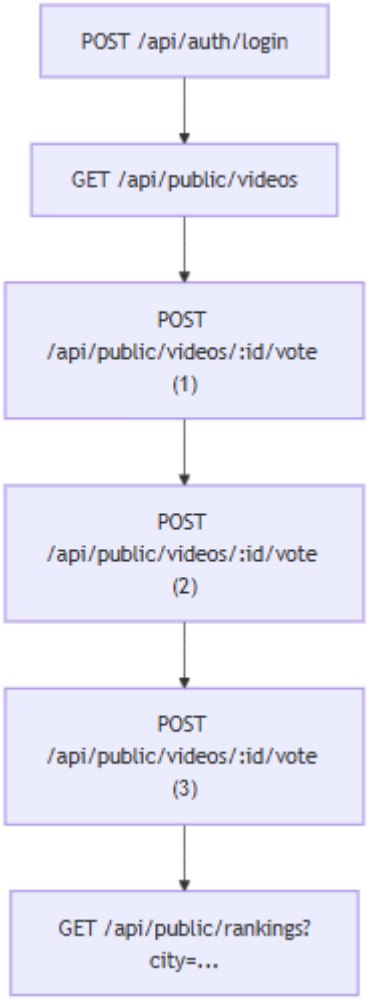

Figura 1. Flujo Escenario Interactivo

Universidad de los Andes
Facultad de Ingeniería

2.2. Escenario Carga/Asíncrono (Uploads)

Objetivo: Someter ingestión y pipeline asíncrono.
Flujo: Realizar login y carga del video (upload)

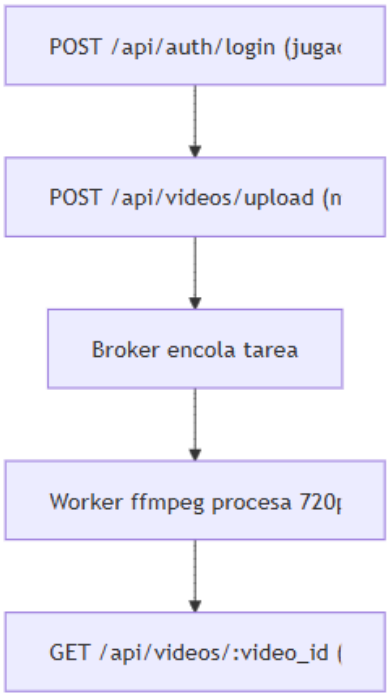

Figura 2. Flujo Escenario Upload

Universidad de los Andes
Facultad de Ingeniería

3. ESCENARIOS DE PRUEBAS

3.1. Prueba de Carga TG – Interactivo (login → listar → votar×3 → ranking)

Se plantea ejecutar una prueba de carga escalonada con las siguientes etapas:

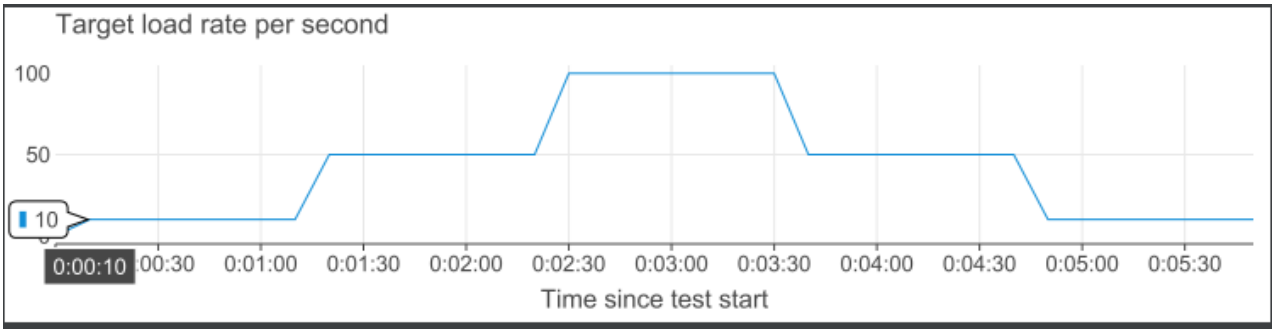

Figura 3. Escenario de carga flujo Interactivo

Inicio sin carga: rate(0/sec) durante 10 segundos.

Carga inicial baja: rate(10/sec) durante **1 minuto**.

Incremento moderado:

- Transición a rate(50/sec) en 10 segundos → Aumento de carga.
- Mantención de rate(50/sec) durante **1 minuto**.

Carga alta:

- Transición a rate(100/sec) en 10 segundos → Escalamiento a carga alta.
- Mantención de rate(100/sec) durante **1 minuto** → Se evalúa el rendimiento en condiciones exigentes.

Descenso progresivo:

- Reducción a rate(50/sec) en 10 segundos → Simulando disminución de tráfico.
- Mantención de rate(50/sec) durante **1 minuto.**.

Vuelta a carga baja:

- Reducción a rate(10/sec) en 10 segundos → Se simula el final de la jornada o baja demanda.

- Mantención de rate(10/sec) durante **1 minuto**.

3.2. Prueba de Carga TG – Upload (login → upload multipart 30–100 MB)

Se plantea ejecutar una prueba de carga escalonada con las siguientes etapas:

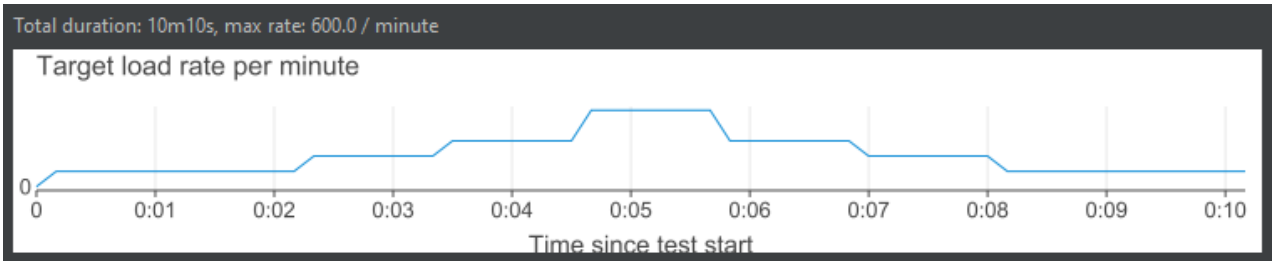

Figura 4. Escenario de carga flujo Upload

Inicio en reposo:

- **Tasa:** rate(0/sec)
- **Duración:** random_arrivals(10 sec)
- **Objetivo:** establecer un punto de partida sin carga.

Incremento gradual de carga:

- **Subida** a rate(2/sec) durante random_arrivals(2 min)
- **Luego** a rate(4/sec) durante random_arrivals(1 min)
- **Posteriormente** a rate(6/sec) durante random_arrivals(1 min)
- **Finalmente** a rate(10/sec) durante random_arrivals(1 min)

Descenso gradual de carga:

- **Reducción** a rate(6/sec) durante random_arrivals(1 min)
- **Luego** a rate(4/sec) durante random_arrivals(1 min)
- **Finalmente** a rate(2/sec) durante random_arrivals(2 min)

Universidad de los Andes
Facultad de Ingeniería

4. ESTRATEGIA Y CONFIGURACIÓN DE PRUEBAS

4.1. Configuración

Flujo	Etapas	Configuración
TG-Interactivo	**Humo**	(1 usuario, 1 min).
	Carga progresiva:	10 → 50 → 100 → 50 → 10 usuarios concurrentes (1 - 2 min por escalón).
	Estrés:	Subir hasta p95 > 1 s o error > 1%.
TG-Upload	**Humo**	(1 usuario, 1 min).
	Carga progresiva:	2 → 4 → 6 → 10 → 6 → 4 → 2 usuarios concurrentes (1 - 2 min por escalón).
	Estrés:	Subir hasta p95 > 1 s o error > 1%.

Tabla 2. Configuración de Pruebas

4.2. Definición de métricas

Flujo	Configuración
TG-Interactivo:	Foco en **p95** por endpoint; objetivo inicial ≤ 1000 ms.
TG-Upload (ingestión multipart):	Objetivo inicial **p95 ≤ 5 s**.

Tabla 3. Definición de métricas

Universidad de los Andes
Facultad de Ingeniería

5. RESULTADOS DE LAS PRUEBAS

5.1. Pruebas de Humo

Las pruebas de humo sirvieron para validar el correcto funcionamiento de los flujos antes de someterlos a cargas altas. En el flujo interactivo se realizaron 12 peticiones (login, listados, tres votos y ranking) y en el flujo de subida cuatro peticiones (login y subida). Los resultados muestran tiempos de respuesta moderados (entre 1 y 7 s en promedio), con **100 % de éxito** en la mayoría de los pasos excepto en algunos votos donde la API respondió con errores de unicidad (indicados como *failure*).

La tabla siguiente resume los resultados:

Escenario	Flujo/etiqueta	Nº peticiones	p50 (ms)	p95 (ms)	Tiempo máximo (ms)	Éxito
Flujo interactivo	Auth / Login (Interactivo)	2	3 696	6 805	7 151	100 %
	Public / List videos	2	1 705	3 161	3 323	100 %
	Public / Vote video (1)	2	1 800	3 335	3 506	0 %
	Public / Vote video (2)	2	4 472	6 097	6 277	0 %
	Public / Vote video (3)	2	1 280	2 274	2 384	0 %
	Ranking / Get rankings	2	5 732	6 658	6 761	100 %
Flujo de Subida (Upload)	Auth / Login (Upload)	2	202	249	254	100 %
	Videos / Upload (multipart)	2	44 440	45 258	45 349	100 %

Tabla 4. Resultados Pruebas de Humo

Nota: Para la operación **Vote**, se muestra éxito de 0% debido a que el video ya había sido votado por el usuario logueado.

Análisis: en el flujo de subida el endpoint /upload tarda ~45 s con archivos de 30–100 MB; en los flujos de voto se observaron fallos porque JMeter reintentó votar tres veces sobre el mismo vídeo y el backend, siguiendo las recomendaciones de unicidad, devuelve error de idempotencia. Estos errores no representan fallos de disponibilidad.

Universidad de los Andes
Facultad de Ingeniería

5.2. Pruebas de carga Escalonada – TG-Interactivo

Se procesaron **89 309 solicitudes** en el escenario interactivo. Cada registro incluye la etiqueta de la operación (login, listar, votar×3, ranking), la latencia en milisegundos y el número de hilos activos (allThreads).

A continuación se muestra un resumen estadístico de los tiempos de respuesta por etiqueta (p50/p90/p95/p99):

Etiqueta	Nº peticiones	p50 (ms)	p90 (ms)	p95 (ms)	p99 (ms)	Tasa de éxito
Auth / Login	15 329	21 030	36 765	43 429	125 723	22,5 %
Public / List videos	15 060	8 389	21 090	37 888	129 221	43,3 %
Public / Vote video (1)	14 804	716	21 041	21 052	39 971	5,05 %
Public / Vote video (2)	14 777	3 470	21 044	21 054	50 111	0,26 %
Public / Vote video (3)	14 714	3 313	21 043	21 051	46 299	0,16 %
Ranking / Get rankings	14 625	10 519	23 631	46 365	112 017	47,0 %

Tabla 5. Resultados Prueba de Carga Flujo Interactivo

Observaciones principales:

- **Latencias elevadas:** los tiempos p95 superan con creces los límites de aceptación del plan. El login tiene un **p95 ~ 43 s** y el listado de videos un **p95 > 37 s**, cuando se esperaba < 1000 ms. Las operaciones de voto y ranking también rebasan los objetivos.
- **Tasa de éxito baja:** la columna *éxito* indica el porcentaje de peticiones con campo success=true. La mayoría de las solicitudes de voto son marcadas como fallidas, posiblemente porque el backend devuelve códigos 409/422 (voto duplicado). Es necesario ajustar los datos de prueba para evitar reintentos que el backend rechaza y para interpretar correctamente los códigos 4xx/5xx.
- **Variación entre percentiles:** mientras que el p50 de las operaciones de voto (especialmente el primer voto) es relativamente bajo (<1 s), los percentiles altos se disparan a ~21 s. Esto sugiere saturación del backend o del cliente de pruebas cuando el número de hilos crece.

Para entender cómo afectan los hilos concurrentes al rendimiento, se agruparon las métricas por el campo allThreads y se calcularon el throughput (peticiones por segundo) y el p95 de latencia.

La siguiente gráfica muestra que a medida que aumenta la concurrencia el throughput se incrementa hasta cierto punto, pero el tiempo de respuesta p95 crece de forma no lineal, evidenciando saturación del servicio:

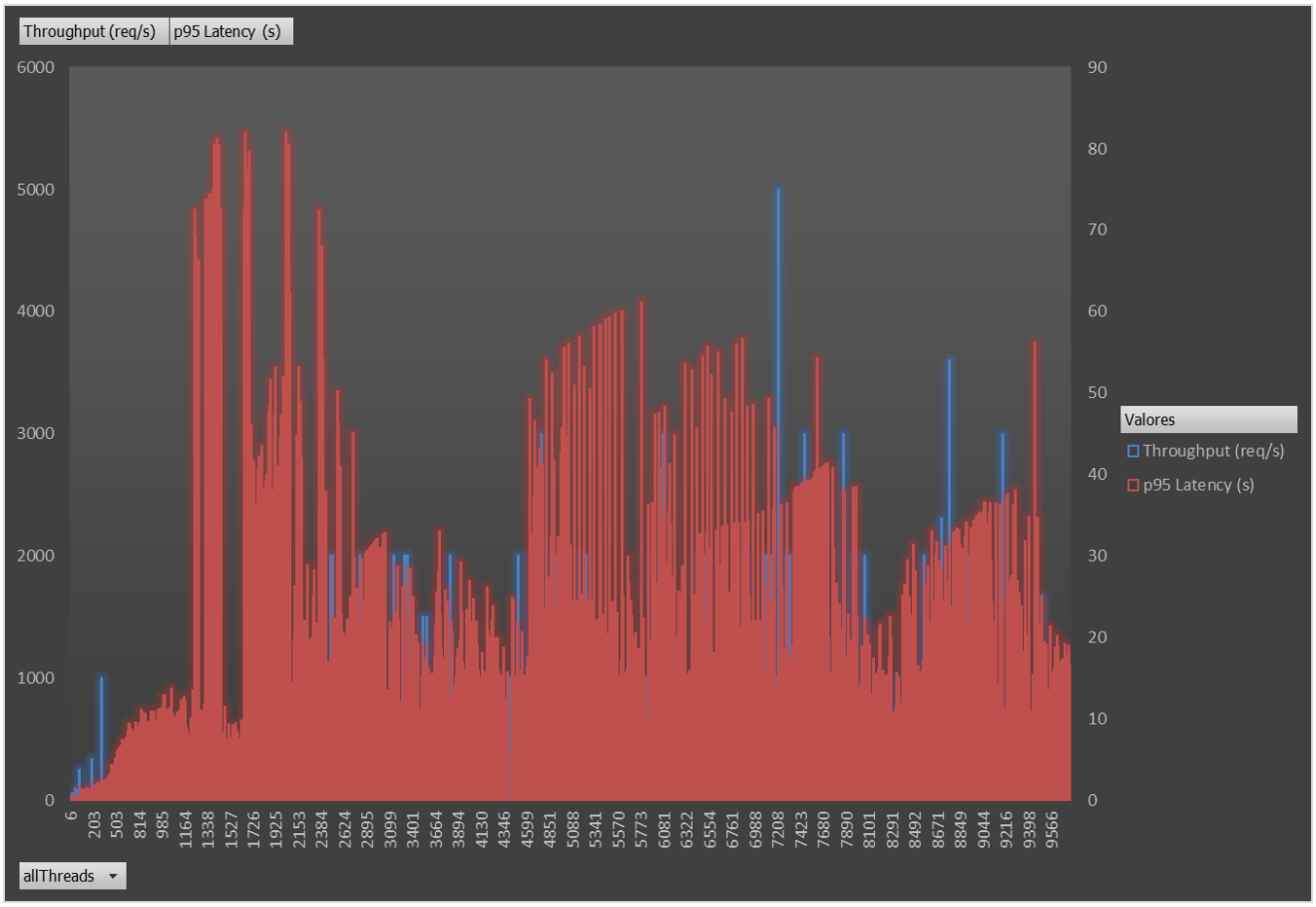

Figura 5. Throughput vs Latencia

En bajas concurrencias (10 hilos) se alcanzan hasta ~12 peticiones/s con p95 ~2 s; al incrementarse a 50 hilos el throughput crece, pero el p95 se eleva por encima de 30 s y finalmente supera los 50 s. Al validar la forma de la curva, la relación entre carga y métricas es lineal al inicio pero se vuelve no lineal cuando se saturan la CPU o el I/O.

Universidad de los Andes
Facultad de Ingeniería

Comportamiento de la infraestructura durante la prueba de carga:

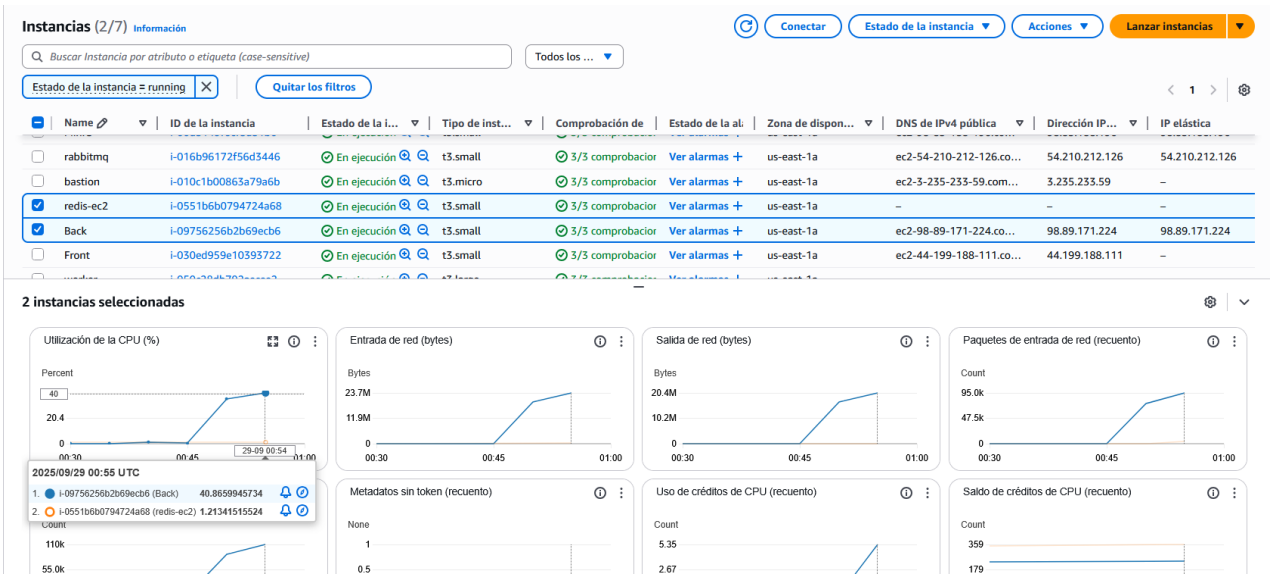

Figura 6. Comportamiento de la infraestructura durante la prueba de carga

Al finalizar la prueba de Carga:

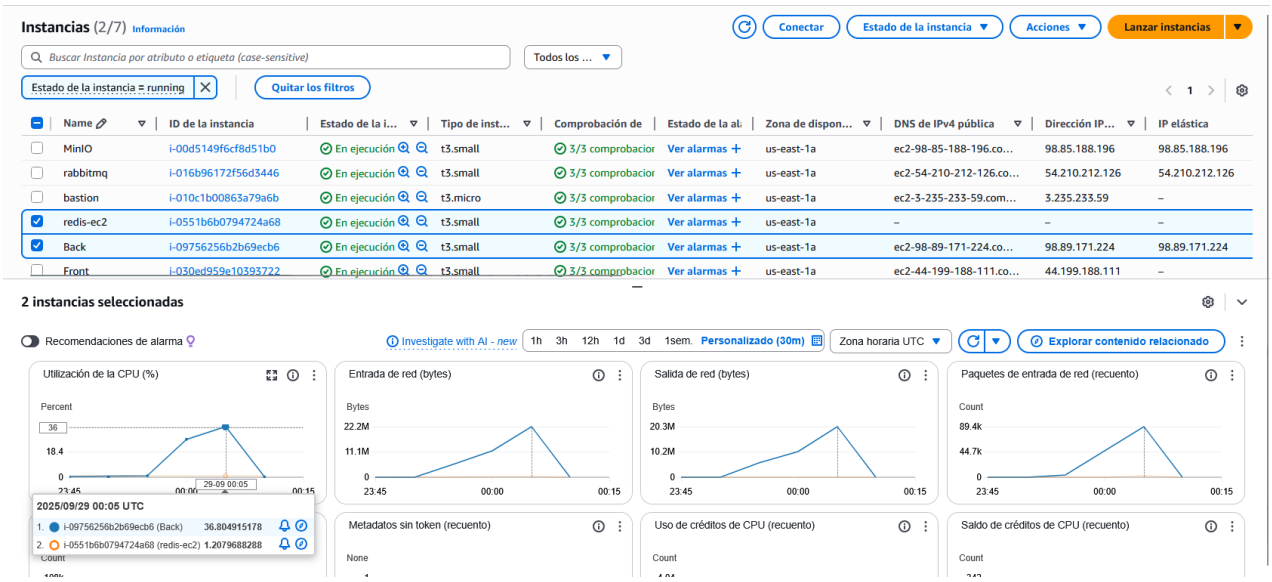

Figura 7. Comportamiento de la infraestructura finalizada la prueba de carga

5.3. Pruebas de carga Escalonada – TG-Upload

En el escenario de carga de subidas se procesaron **5 077 solicitudes** (logins y subidas). La subida de archivos generó latencias significativamente mayores que las operaciones del flujo interactivo.

Los resultados resumidos son:

Etiqueta	Nº peticiones	p50 (ms)	p90 (ms)	p95 (ms)	p99 (ms)	Tasa de éxito
Auth / Login	2 604	19 748	29 395	47 049	296 008	72 %
Videos / Upload (multipart)	2 473	142 357	284 162	312 809	338 083	1,6 %

Tabla 6. Resultados Prueba de Carga Flujo Upload

Las gráficas de throughput y p95 en función de la concurrencia evidencian una caída abrupta del rendimiento a medida que se incrementa la tasa de subida. El throughput por hilo se mantiene por debajo de 1 petición/s y el p95 de latencia de subida escala de ~150 s a más de 300 s, muy por encima del objetivo de 5 s.

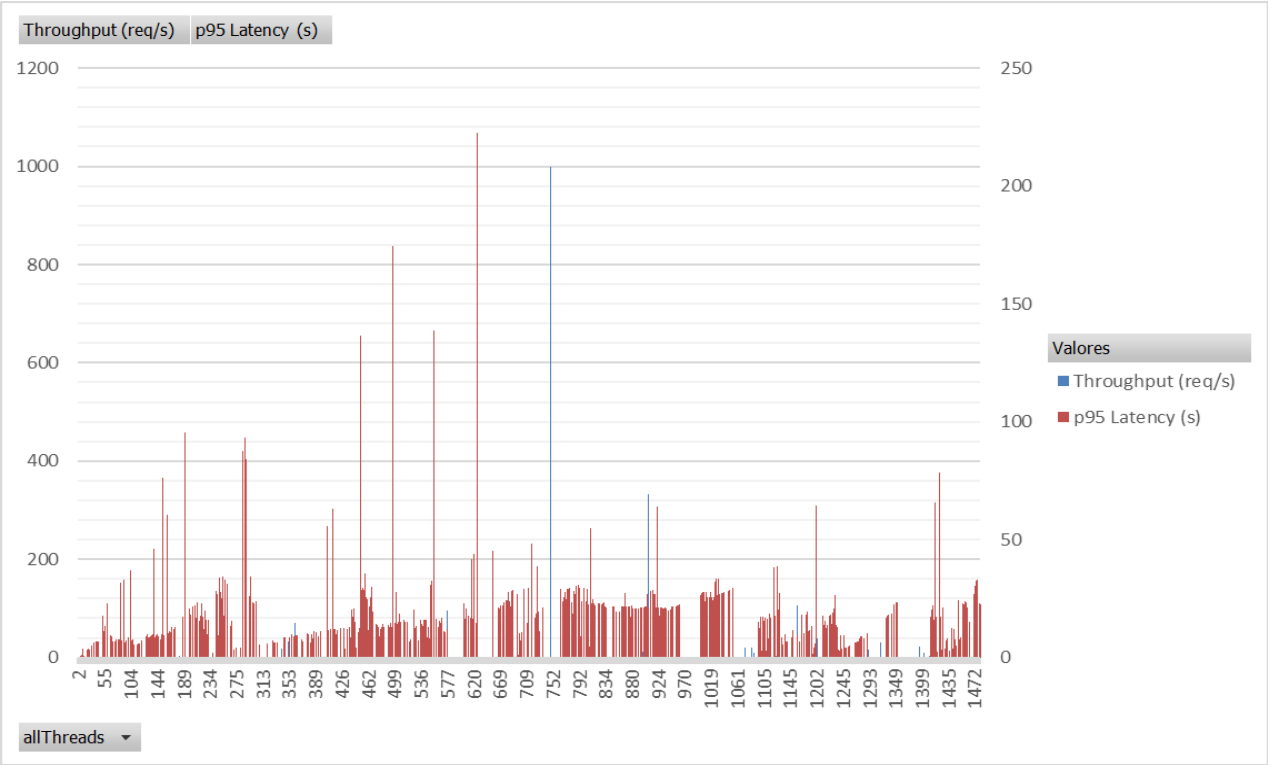

Figura 8. Throughput vs Latencia

Comportamiento de la infraestructura durante la prueba de carga:

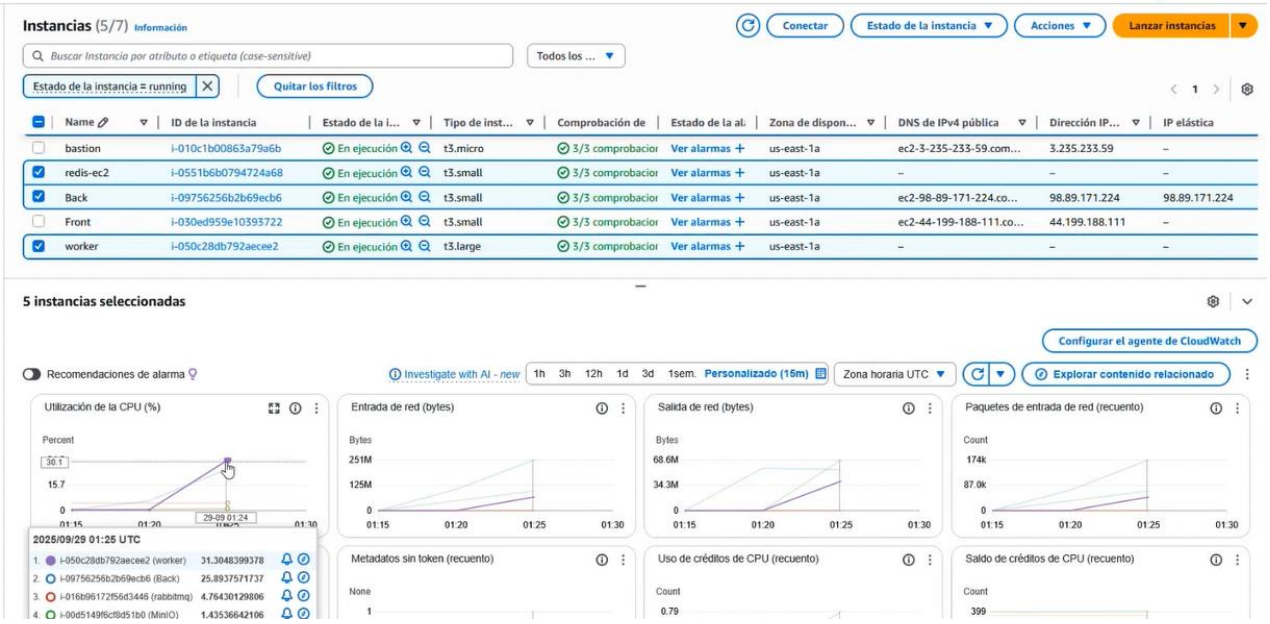

Figura 9. Comportamiento de la infraestructura durante la prueba de carga Upload

Interpretación:

- La infraestructura actual no logra mantener la tasa objetivo de 50 subidas/min. El **p95** de la subida se ubica en el orden de **5 – 6 minutos** al alcanzar 10 usuario concurrentes, podría deberse a que se llegó a la capacidad máxima de la memoria RAM o a una mala administración de la memoria
- La tasa de éxito de sólo **1,6 %** sugiere que la mayoría de las solicitudes de subida fallaron o expiraron. Esto puede deberse a timeouts en el cliente de pruebas (la prueba se diseñó con un límite de 10 minutos).
- El login en este escenario también presenta latencias elevadas, reflejando que el backend está saturado mientras procesa las subidas.

5.4. Pruebas de estrés

Teniendo en cuenta los resultados de las pruebas de carga, se desiste realizar la prueba de estrés ya que la tasa de éxito fue muy baja.

Universidad de los Andes
Facultad de Ingeniería

6. ANALISIS Y CONCLUSIONES

Los resultados obtenidos evidencian que el sistema, en su configuración actual, **no cumple con los criterios de aceptación** establecidos en el plan de pruebas. Los tiempos de respuesta p95 superan con amplitud los umbrales de 1000 ms para las operaciones web y 5 s para las subidas; además, la tasa de errores es alta.

A continuación se presentan las principales conclusiones y recomendaciones:

1. **Saturación de recursos**: la infraestructura utiliza instancias **t3.small** para el frontend y el backend. Estas instancias tienen un límite de 2 vCPU y 2 GB RAM, insuficiente para manejar 100 usuarios concurrentes. Es necesario **migrar a instancias más potentes** (p. ej., **t3.medium o t3.large** para el frontend/back**end** y **c6i.large o c6i.xlarge** para los workers).
2. **Escalado horizontal**: para cumplir con la carga objetivo se recomienda implementar un **Auto Scaling Group** para el backend. A 100 usuarios concurrentes, el plan especifica una degradación aceptable p95 < 1000 ms; esto solo se logra cuando la CPU y la memoria no se saturan. Se deben desplegar múltiples instancias del API detrás de un balanceador (ELB).
3. **Optimización de la base de datos**: los cuellos de botella en las operaciones de voto y ranking apuntan a la base de datos y a la lógica de unicidad. El plan aconseja utilizar **índices y un pool de conexiones**.
4. **Cliente de carga adecuado**: el equipo realizó las pruebas desde un portátil; esto limita el throughput que puede generar. Para futuras pruebas se recomienda utilizar una instancia EC2 dedicada para JMeter (c6i.xlarge o m6i.large según el presupuesto). Ello permitirá generar cargas realistas y obtener métricas más representativas.

Universidad de los Andes
Facultad de Ingeniería

7. RECOMENDACIONES PARA FUTURAS VERSIONES

Para que la **versión 2** del proyecto atienda a cientos de usuarios finales que subirán archivos y consumen el API de manera concurrente, se proponen las siguientes modificaciones:

1. **Incrementar la capacidad del backend**: escalar el servicio web a **t3.large** o **m6i.large** con 2–4 vCPU y 8 GB RAM; habilitar auto-scaling basado en métricas de CPU y latencia.
2. **Escalar los workers de procesamiento**: usar instancias con capacidad de cómputo (c6i.large/xlarge) y aumentar el número de workers proporcionalmente al número de vídeos concurrentes (4–8 workers para soportar 100 subidas simultáneas).
3. **Almacenar archivos en un servicio escalable**: Implementar pre-firmado (signed URLs) para que las cargas se hagan directamente al almacenamiento, descargando el backend.

Universidad de los Andes
Facultad de Ingeniería

8. CONCLUSIÓN

El análisis de capacidad revela que, bajo la configuración actual, el sistema no es capaz de sostener las cargas objetivo-definidas en el plan de pruebas. Las pruebas de humo mostraron que los endpoints funcionan, pero las pruebas de carga escalonada evidenciaron tiempos de respuesta mayores a 30 s e, incluso, a varios minutos.

Para alcanzar los criterios de aceptación y garantizar una experiencia fluida a cientos de usuarios concurrentes, es imprescindible dimensionar correctamente la infraestructura, optimizar la base de datos y dotar al proyecto de herramientas de escalado automático, para lo cual es necesario implementar herramientas de monitoreo que permitan visualizar el consumo de los recursos de cada componente (CPU, RAM)