ANÁLISIS DE CAPACIDAD

Objetivo

Ejecutar escenarios de prueba que permitan medir la capacidad máxima que pueden soportar algunos componentes del sistema. El resultado debe expresarse en el documento de análisis de capacidad con el que se simula el acceso, la carga, el estrés y la utilización de la aplicación.

Lugar y formato de entrega

Para cada una de las entregas del proyecto se debe entregar los siguientes recursos:

- Documento de análisis de capacidad, adicionando la ejecución de las pruebas de estrés, análisis de resultados y conclusiones.
- Entregar toda la documentación vía GitHub o GitLab.

Infraestructura requerida para la ejecución de las pruebas de carga

Para la ejecución de las pruebas, se recomienda utilizar una instancia de cómputo independiente de la cuenta y del proyecto en AWS en los que se despliega la aplicación final.

En particular, se sugiere realizar las pruebas de carga desde su equipo o entorno local en lugar de utilizar su cuenta de AWS Academy. Esto permitirá evitar bloqueos o suspensiones de la cuenta causados por prácticas que el servicio pueda considerar inapropiadas.

Asegúrese de crear una instancia de cómputo que cumpla con los requisitos definidos en el diseño de su plan de pruebas.

Escenarios

Escenario 1 - Capacidad de la capa Web (usuarios concurrentes)

Determinar el número de usuarios concurrentes (y RPS asociado) que la API de subida soporta cumpliendo SLOs, sin estar limitado por la capa asíncrona.

Estrategia de implementación

- **Desacoplar la capa worker**: en endpoints de carga, devolver 202 y redirigir a un mock de cola que acepte mensajes en memoria y responda instantáneamente.
- **Simular carga real de archivos**

Escenarios de prueba

- **Sanidad (Smoke):** 5 usuarios durante 1 minutos para validar que todo responde y la telemetría está activa.
- **Escalamiento rápido (Ramp):** iniciar en 0 y aumentar hasta **X** usuarios en 3 minutos; mantener 5 minutos. Repetir con X creciente (p. ej., 100 → 200 → 300) hasta observar degradación.
- **Sostenida corta:** ejecutar 5 minutos en el **80% de X** (el mejor nivel previo sin degradación) para confirmar estabilidad.

Criterios de éxito/fallo

- **Capacidad máxima,** mayor número de usuarios concurrentes que cumple:
  - **p95** de endpoints ≤ **1 s**
  - **Errores** (4xx evitables/5xx) ≤ **5%**.
  - Sin **resets/timeouts** anómalos ni **throttling** del almacenamiento.
- Si se supera **capacidad máxima**, registrar el **primer KPI** que se degrada (CPU del API, ancho de banda, etc) y usarlo como guía de mejora.

Herramientas sugeridas

- **Generador**: Locust o JMeter
- **Observabilidad**: Prometheus/Grafana + APM (OpenTelemetry)

Salidas esperadas

- Curva usuarios→latencia/errores.
- RPS sostenido a **capacidad máxima** (Soporta 450 usuarios concurrentes con 320 RPS manteniendo p95 1,0 s).
- *Bottlenecks* con evidencias (CPU 90% en API, saturación de ancho de banda de subida, etc.).

Plan B — Rendimiento de la capa Worker (videos/min)

Medir cuántos videos por minuto procesa el/los worker(s) a distintos niveles de paralelismo y tamaños de archivo.

Estrategia de implementación

- **Bypass de la web**: inyectar directamente mensajes en la cola (script/productor) con payloads realistas (rutas a archivos en storage de pruebas).

Diseño experimental

- Tamaño de video: 50 MB, 100 MB.
- Concurrencia de worker: 1, 2, 4 procesos/hilos por nodo.

Para cada combinación:

- Ejecutar **pruebas de saturación**: subir la cantidad de tareas progresivamente en la cola
- Ejecutar **pruebas sostenidas**: mantener un numero fijo de archivos en la cola que no la sature

Métricas y cálculos

- Throughput observado: X = videos procesados por / minuto.
- Tiempo medio de servicio: S = tiempo_proceso_promedio por video.

Criterios de éxito/fallo

- Capacidad nominal: (videos/min)
- Estabilidad: cola no crece sin control (tendencia ~0) durante la prueba

Herramientas sugeridas

- Productor de mensajes: scripts Python/Go contra Redis/RabbitMQ.
- Trazabilidad de jobs: IDs correlacionados (enqueue→start→end
- Perfilado del worker: métricas (CPU, IO, red, etc).

Salidas esperadas

- Tabla de capacidad por tamaño y configuración (1 nodos × 4 hilos → 18.5 videos/min a 200 MB).
- Puntos de saturación y cuellos de botella (CPU, decodificación, ancho de banda, *temp disk*).

El nombre del documento que describe el plan de pruebas refinado y los resultados de su ejecución deberá ser consignados en formato markdown dentro de su repositorio con base a los lineamiento definidos en el enunciado principal.