Informe de Rendimiento y Escalabilidad

Proyecto: TRANSPUBLI CALI

Versión: 1.0

Integrantes: Juan Camilo Minota Peña

Karen Jhulieth Lucumi Mosquera

1. Introducción

El presente informe describe los resultados de las pruebas de carga realizadas sobre la API REST del sistema TRANSPUBLI CALI, así como las recomendaciones para mejorar su rendimiento y escalabilidad. El objetivo principal es garantizar tiempos de respuesta óptimos ante el aumento de usuarios concurrentes, especialmente en situaciones críticas como alertas de emergencia o consultas de rutas en tiempo real.

2. Herramientas Utilizadas

- **Artillery.io**: Herramienta de pruebas de carga en entorno local sobre la API desarrollada en Node.js con Express.
- **Postman**: Para validación manual de los endpoints.
- **Google Chrome Developer Tools**: Para observar tiempos de respuesta en ambiente cliente.

3. Metodología

Se configuró una simulación de hasta 200 usuarios concurrentes, accediendo a los siguientes endpoints críticos:

- GET /api/rutas
- POST /api/emergencias
- GET /api/usuarios

Cada prueba fue ejecutada por un periodo de 60 segundos y se registraron:

- Tiempos promedio de respuesta (ms)
- Tasa de peticiones por segundo
- Porcentaje de errores

4. Resultados de Pruebas de Carga

Endpoint	Tiempo promedio	Peticiones/segundo	% Errores
GET /api/rutas	180 ms	45	0%
POST /api/emergencias	220 ms	38	0%
GET /api/usuarios	160 ms	48	0%

Los resultados demuestran que la API mantiene un rendimiento aceptable bajo condiciones de carga moderada. No se registraron errores en la entrega de respuestas.

5. Análisis

- **Escalabilidad horizontal**: El backend puede desplegarse en instancias paralelas para distribuir carga, mediante balanceadores tipo NGINX.
- **Caché**: Endpoints como `GET /api/rutas` pueden beneficiarse de almacenamiento temporal en memoria (Redis o in-memory cache).
- **Optimización de Base de Datos**: Crear índices sobre campos de búsqueda frecuente (por ejemplo, nombre de ruta o ubicación).
- **Logs asincrónicos**: Las operaciones de registro (como POST /emergencias) deben procesarse sin bloquear la respuesta al usuario.

6. Recomendaciones de Mejora

- Implementar Redis como sistema de caché para respuestas de rutas y paradas frecuentes.
- Utilizar PM2 para escalabilidad de procesos Node.js en ambiente de producción.

- Incorporar NGINX como proxy inverso y balanceador de carga.
- Optimizar consultas SQL con uso de índices y relaciones adecuadas.
- Automatizar pruebas de carga en cada sprint utilizando scripts de Artillery.

7. Conclusión

La API de TRANSPUBLI CALI demuestra un comportamiento estable en condiciones de carga de hasta 200 usuarios concurrentes. Las optimizaciones sugeridas permitirán escalar el sistema de manera efectiva para uso real en entornos urbanos, especialmente en contextos sensibles como el apoyo a personas con discapacidad visual.

Este informe forma parte de los entregables del Proyecto Integrador del curso Desarrollo de Software I.