

Servicio nacional de “Notificaciones push”

National “Push notifications” service

MSc. Joaquín Quintas Santiago^{1*}, Ing. Yuniesky Orlando Vasconcelo Mir²

Recibido: 12/2023 | Aceptado: 02/2024 | Publicado: 08/2024

Resumen

Las notificaciones son útiles y necesarias para disímiles propósitos: informar a los usuarios de eventos importantes para ellos, promociones, ofertas; mantenerlos al tanto de soluciones de problemas de seguridad y actualizaciones del sistema. En muchos casos, se requiere que estas notificaciones sean a tiempo casi real en aplicaciones que no necesitan estar activas para recibirlas. Para estos efectos, existen varias plataformas propietarias principalmente de Google, Huawei y Apple. Sin embargo, en varios casos, no es posible emplearlas debido a restricciones de los propietarios. Este trabajo se realizó para identificar si existe alguna alternativa de software libre y de código abierto para ofrecer este servicio a las instituciones y desarrolladores de aplicaciones nacionales. También para analizar las diferentes alternativas atendiendo a un conjunto de requisitos y determinar en qué grado se satisface cada uno y cuál es más factible de implementar para ofrecer un servicio masivo. Para ello, además de comparar un grupo de dichas alternativas, se realizaron un conjunto de pruebas para validar el adecuado funcionamiento en entornos de alta demanda y con una base de usuarios superior a los 100 000 usuarios conectados de forma simultánea. Como resultados de este trabajo, se identificó al menos una alternativa para ofrecer este servicio y ganar en independencia y soberanía tecnológica.

1* XETID, Cuba, jquintas@xetid.cu

2 XETID, Cuba, yovasconcelo@xetid.cu

Palabras clave: Notificaciones; push; software-libre; ntfy

Abstract

Notifications are useful and necessary for various purposes: informing users of events important to them, promotions, offers; keeping them updated on security problem solutions and system updates. In many cases, these notifications need to be almost real-time in applications that do not need to be active to receive them. For these purposes, there are several proprietary platforms mainly from Google, Huawei, and Apple. However, in several cases, it is not possible to use them due to owner restrictions. This work was carried out to identify if there is any free and open-source software alternative to offer this service to national institutions and application developers. Also, to analyze the different alternatives according to a set of requirements and determine to what degree each one is satisfied and which is more feasible to implement in a mass service environment. For this, in addition to comparing a group of these alternatives, a set of tests were carried out to validate the correct performance in high-demand environments and with a user base of over 100,000 users connected simultaneously. As a result of this work, at least one alternative was identified to offer this service and gain technological independence and sovereignty.

Keywords: Notificaciones; push; free-software; ntfy

Introducción

En un mundo cada vez más digital, las formas en que las empresas y las aplicaciones se comunican con los usuarios están evolucionando rápidamente. Una de las formas más efectivas de mantener a los usuarios comprometidos y actualizados es a través de las “Notificaciones push”. Estos mensajes instantáneos pueden proporcionar actualizaciones en tiempo real, promociones personalizadas y recordatorios útiles, todo directamente en el dispositivo de un usuario. Poseen, además, la definición de un objetivo claro para su activación, son fáciles de leer y se puede planificar su envío (IBM, 2023).

En Cuba, a consecuencia del bloqueo, no se tiene acceso a las principales plataformas que se incorporan de forma nativa con los diferentes sistemas operativos y fabricantes de dispositivos móviles. Por

tanto, se hizo necesario identificar herramientas de software libre para implementar este tipo de notificaciones, así como su valoración según las ventajas y posibilidades que ofrecen.

Además, se realizaron una serie de pruebas y, finalmente, se determinó el sistema más adecuado para este propósito. A través de este análisis, se espera proporcionar una solución nacional de notificaciones push para cualquier persona o entidad interesada en implementarlas en su propia aplicación o plataforma.

Materiales y métodos

El método utilizado en este estudio es cuantitativo, ya que se basa en la recopilación y análisis de información objetiva sobre las características del software. Para ello, se realizó una búsqueda sistemática en varios sitios web de software. Se establecieron criterios claros para la búsqueda, que incluían:

- Ser de software libre, de código abierto.
- Poseer soporte para las principales plataformas de teléfonos móviles (iOS y Android) y para aplicaciones web.
- Posibilidad de despliegue privado sin necesidad de emplear servicios de terceros: "auto-alojados".
- Capacidad de satisfacer la demanda de 100 000 usuarios de forma concurrente.
- Se evaluaron más de 10 opciones de software y se recopiló información objetiva sobre cada una de ellas. La evaluación se basó en la presencia o ausencia de las características específicas, lo que permitió una comparación objetiva de las opciones. De esta forma se decantaron los posibles candidatos y se seleccionó uno de ellos.

Además, se evaluó el rendimiento del candidato de manera objetiva y sistemática, utilizando herramientas y métricas específicas para medir el desempeño del software seleccionado bajo diferentes condiciones de carga. Para ello, además del software de prueba, se empleó la herramienta "K6" (de Grafana Labs) utilizada para simular la carga y el estrés en el sistema. Para ello también fue necesario montar un laboratorio con servidores, máquinas virtuales, redes, etc. y el conjunto de datos utilizados para alimentar las pruebas y simular el comportamiento de los usuarios.

Los objetivos que guiaron el diseño de las pruebas fueron:

- Identificar la capacidad de respuesta del servicio Ntfy.
- Obtener información sobre el consumo de recursos en entornos de alta exigencia.
- Validar la pertinencia o no de emplear el software seleccionado.

El entorno de pruebas fue configurado para alcanzar estos objetivos y proporcionar datos valiosos para la posterior toma de decisiones sobre:

- La capacidad del software seleccionado de comportarse de forma satisfactoria en el entorno de explotación esperado.
- Los recursos de hardware mínimos que deberán ser empleados para el despliegue en producción del servicio.
- En qué entornos se hace necesario escalar el servicio.

En principio, valoraciones iniciales hacen estimar que un servicio de notificaciones para una plataforma de software requerirá de poder contar con al menos 100 000 clientes conectados de forma concurrente. También, que el flujo de mensajes tenga un pico máximo de 500 mensajes por segundo.

No para todas las plataformas, pero en los casos de las que ofrecen servicios de telefonía o videoconferencia, deberán obtenerse latencias entre el envío y recepción de los mensajes menores a 1 segundo. Pues son aplicaciones que requieren las notificaciones casi a tiempo real.

Entorno de pruebas

Para lograr los volúmenes de clientes, conexiones y procesamiento de mensajes que se aproximen a los valores estimados y mencionados con anterioridad, se preparó el entorno de pruebas que se detalla a continuación.

Servidor de prueba Ntfy

Valorando que se intentará ofrecer servicios a 100 000 usuarios concurrentes y que cada conexión que establezcan los clientes deberá permanecer activa aunque no existan mensajes para ese cliente; se realizará un consumo intensivo de memoria RAM por parte del servicio.

También se valora que por la cantidad de mensajes por segundo (500 pico máximo), se requerirá un nivel no demasiado alto de empleo del CPU y que el sistema se beneficiará del procesamiento en paralelo.

Por tanto, para las pruebas se configuró un servidor con 16GB de memoria RAM y 20 núcleos de CPU. Se configuró¹ el Ntfy de la siguiente forma:

- No emplear caché de mensajes.
- No implementar el envío de ficheros.
- Deshabilitar los límites de conexiones posibles desde una dirección IP.
- Deshabilitar los límites de mensajes por segundo permitidos para un usuario.
- Deshabilitar los límites de ráfaga de mensajes por segundo (burst messages).

Herramienta de Benchmark

Para realizar las pruebas se empleó la herramienta K6 de Grafana Labs (K6, 2023). Se configuró un cluster Kubernetes con 6 nodos *workers* virtualizados sobre Proxmox VE 7.0. Cada *worker* fue equipado con 20 vCPU, 50GB de memoria RAM y una interfaz de red a 1 Gb/s. Se definió una afinidad y segmentación de los nodos para aislar cada componente de la forma siguiente: 5 nodos para los K6 distribuidos; 1 nodo para el Grafana y Prometheus.

Descripción de las pruebas

Las pruebas se diseñaron para ejecutarse de forma continua por el tiempo de 1 hora fuera del horario laboral, con el objetivo de minimizar la variación de los datos por incrementos del empleo de la red o fallas en la misma durante el proceso de pruebas.

Prueba 1

Para esta prueba se configuró la herramienta de pruebas para simular 100 000 usuarios de forma concurrente y el envío de 1 000 mensajes de texto por segundo.

Se podrá apreciar que el número de mensajes por segundo es el doble de la estimación de uso de servicio cuyo caso base valora que en momentos pico se enviarán 500 mensajes por segundo.

1 Muchos de los límites deshabilitados serán empleados en la puesta en marcha del servicio con objetivos de seguridad y de garantizar la disponibilidad del servicio. En el caso del entorno de pruebas, deshabilitarlos facilita la simulación de mayor cantidad de usuarios.

Prueba 2

Para esta segunda prueba se configuró la herramienta para simular 100 000 clientes concurrentes al igual que en la prueba anterior. Pero en este caso se simuló el envío de diferentes rangos de mensajes por segundo desde los 7 000 hasta los 15 000 por segundo.

Esto se realizó con el objetivo, ya no de validar si el servicio de notificaciones era capaz de soportar la carga de trabajo esperada si no, para intentar identificar los límites del servicio configurado.

Resultados y discusión

1 – Soluciones de software libre

Para solventar los problemas que se encontraron con los principales proveedores y ganar en soberanía e independencia tecnológica, se hizo un estudio de varias herramientas de software libre. Algunas de las más populares son (Mu, 2023)(Stringfellow, 2023):

1.1 – Push.js

Es una compacta biblioteca de JavaScript para notificaciones push en la web. Es gratuita y de código abierto. Es compatible con una amplia gama de navegadores, incluyendo Chrome, Firefox, Safari, Opera y Microsoft Edge (Push.js, 2023).

Es importante mencionar que las notificaciones push de la web tienen algunas limitaciones en comparación con las notificaciones push de las aplicaciones móviles. Por ejemplo, las notificaciones web requieren que el usuario esté en línea y que el navegador esté abierto (aunque no necesariamente en primer plano).

1.2 – Gotify

Un servidor simple para enviar y recibir mensajes en tiempo real. Es de código abierto y se puede auto-alojar en un servidor propio, lo que te da un control total sobre tus datos y elimina la dependencia de los servicios de terceros.

Proporciona una API RESTful que puede ser utilizada para enviar y recibir mensajes. Esto hace que sea compatible con una amplia gama de lenguajes de programación y plataformas (Gotify, 2023).

Permite a los usuarios crear aplicaciones y plugins para extender la funcionalidad del servidor. Soporta HTTPS y permite a los clientes autenticarse con un nombre de usuario y una contraseña.

1.3 – Gorush

Es una herramienta de notificaciones push de código abierto escrita en el lenguaje de programación Go. Permite a los desarrolladores enviar notificaciones push a dispositivos iOS, Android y Desktop. Gorush fue diseñado para ser eficiente y rápido, y es capaz de enviar miles de notificaciones en paralelo (Gorush, 2023).

Gorush en sí, es solo un proxy de notificaciones push y requiere de servicios como FCM -*Firestore Cloud Messaging*-, APNs -*Apple Push Notification service*- o HMS -*Huawei Mobile Services*- para enviar notificaciones a los dispositivos.

1.4 – Ntfy

En un servidor simple para enviar y recibir mensajes en tiempo real. Es de código abierto y se puede auto-alojar. Puede enviar notificaciones a una amplia gama de servicios y plataformas, incluyendo Pushover, Pushbullet, Slack, Telegram, Twitter, entre otros.

Posee control de acceso basado en roles y permite configurar paquetes de notificaciones para determinados clientes. Emplea HTTPS y puede emplearse autenticación básica de nombre de usuario y contraseña o empleando tokens de seguridad (Ntfy, 2023).

2 - Selección del servicio de notificaciones

A continuación se muestra un conjunto de tablas con el objetivo de proporcionar una visión clara y detallada de los aspectos analizados para seleccionar la plataforma que más se adecúe a nuestras necesidades.

	Web	Android	iOS
Push.js	X		
Gotify	X	X	
GoRush		X	X
Ntfy	X	X	X

Tabla 1: Plataformas soportadas

	SMS	Correo	Otros
Push.js			
Gotify			
GoRush			X
Ntfy	X	X	X

Tabla 2. Integración con otros servicios de notificación

	FE	D y S	Seguridad
Push.js	B	B	B
Gotify	B	B	B
GoRush	B	R	B
Ntfy	B	B	B

Tabla 3. Otros aspectos de interés

Nota: se recogen varios aspectos en esta tabla de facilidad de empleo (FE), documentación y soporte (D y S) y seguridad (Seguridad). Se califican de (B)ueno, (R)egular y (M)alo.

En cuanto a aspectos generales, todos los software poseen buena documentación, soporte y seguridad. Salvo la excepción del Gorush en el caso de la documentación. Atendiendo a las plataformas soportadas e integración con otros servicios, Ntfy es el que mejores características ofrece.

Aunque no se ofrece una tabla comparativa, Gorush queda descartado por no ser un servidor de notificaciones auto-alojado. Solo actúa de intermediario para una o varias plataformas como FCM, HMS o APNs. Pero es incapaz de funcionar si no se ha configurado previamente alguna de ellas.

Atendiendo a estos resultados, la selección a priori es: “Ntfy”.

3 – Resultados de las pruebas de carga

Con las pruebas realizadas se obtuvieron un conjunto de gráficos y resultados. Dichos gráficos, se recortaron en tramos donde la información era similar para evitar afectar lo menos posible la leyenda y las escalas.

3.1 – Prueba 1

En esta prueba de enviar 1 000 mensajes por segundo con 100 000 clientes concurrentes, según se muestra en la figura 1, se obtuvo:

- Promedio de latencia de los mensajes: 350 ms.
- Latencia máxima: 494 ms.
- Máximo ancho de banda empleado: 4 MB/s.
- Empleo de memoria RAM: 14 GB.
- Cantidad de CPUs empleados: 3 en promedio.

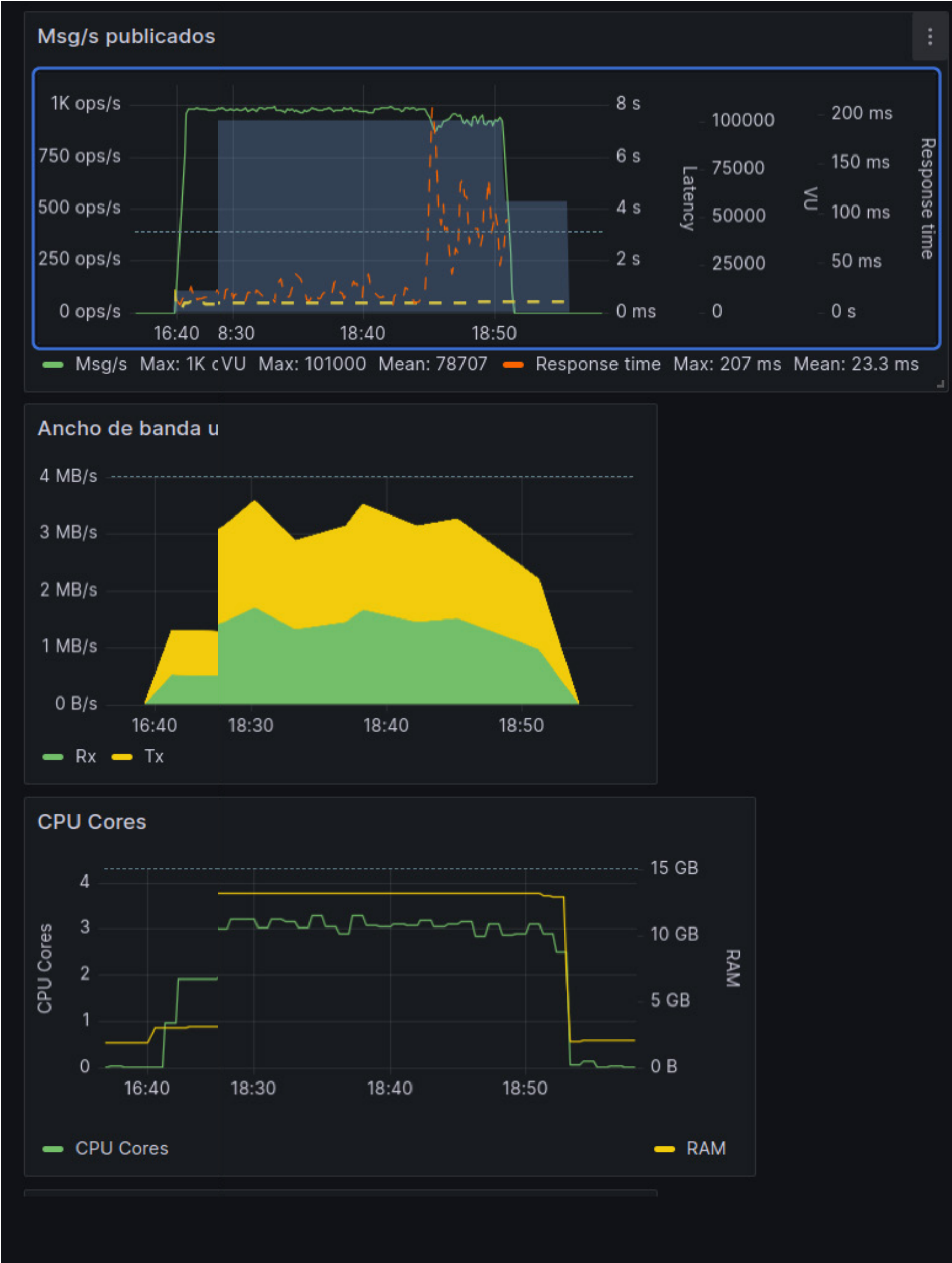


Figura 1. Gráfico del comportamiento para 1 000 mensajes por segundo.

3.2 – Prueba 2

- En esta prueba de enviar entre 7 000 y 15 000 mensajes por segundo con 100 000 clientes concurrentes, según se muestra en la figura 2, se obtuvo:
- Promedio de latencia de los mensajes se mantuvo: entre los 2.5 y 5.1 segundos.
- Máximo ancho de banda empleado: entre los 10 y 15 MB/s.
- Empleo de memoria RAM: 14 GB.
- Cantidad de CPUs empleados: entre 10 y 20 en promedio.

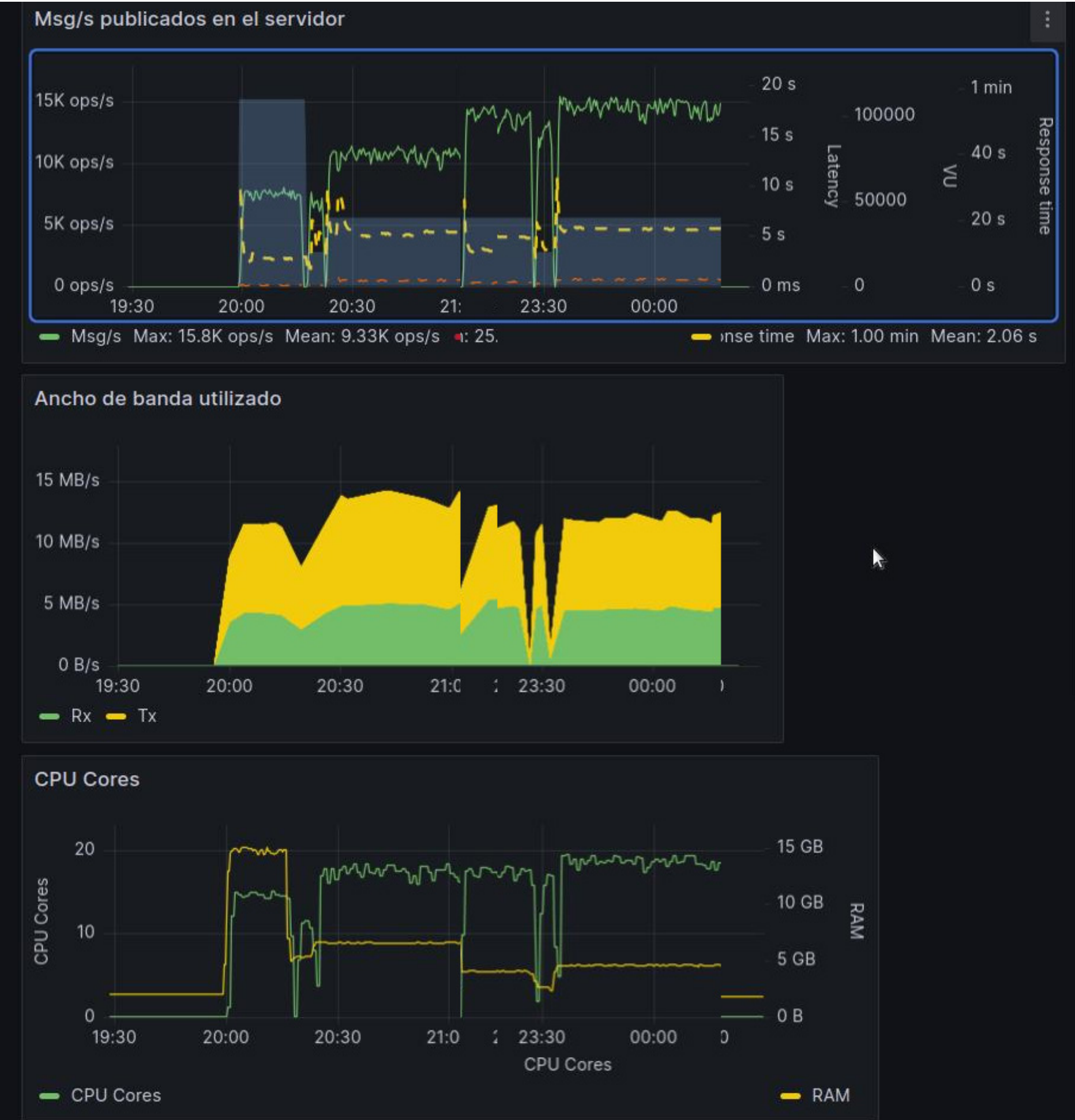


Figura 2. Gráfico del comportamiento para 7 000 hasta 15 000 mensajes por segundo.

Atendiendo a los resultados obtenidos podemos observar que con el incremento de la cantidad de mensajes por segundo, la latencia de

publicación y entrega de los mismos comienza a crecer. Esto ocurre incluso con el incremento de la cantidad de CPUs empleadas.

Con el incremento de la cantidad de CPUs, disminuye el empleo de memoria RAM. Lo cual puede deberse a que se requiere reservar menos memoria por un incremento en la rapidez de entrega de los mensajes.

El ancho de banda empleado, se mantiene relativamente invariable independiente de la cantidad de memoria y CPUs empleados a pesar de los diferentes rangos de mensajes por segundo. Esto podría sugerir que el ancho de banda está actuando como un cuello de botella y, por tanto, causando un incremento en las latencias y consumo de recursos.

Conclusiones

Una vez valorado un conjunto de software atendiendo a diferentes características y haber seleccionado el Ntfy y, a partir de los resultados de las pruebas realizadas se puede arribar a las siguientes conclusiones:

Ntfy cumple con un conjunto de criterios para ser seleccionado para el despliegue de un servicio de notificaciones.

Los resultados obtenidos en las pruebas demuestran que cumple satisfactoriamente para las condiciones de empleo de recursos y latencia de los mensajes en entornos de producción.

A futuro se requiere un análisis más detallado para mejorar el rendimiento del servicio.

Con estos resultados, será posible desplegar un servicio de "Notificaciones push" que beneficiará a un conjunto de aplicaciones de carácter nacional como: Platel, EnZona y Ticket. También podrá ofrecerse como servicio a los desarrolladores de aplicaciones que lo requieran o consideren beneficioso.

Referencias bibliográficas

IBM. (2023). What is a push notification? <https://ibm.com/topics/push-notifications/>

K6. (2023). Grafana Cloud k6 - Performance testing tool. <https://grafana.com/products/cloud/k6/>

Mu, Hamza. (2023). 15 Open-Source Push Notification Projects, Alternative to Apple and Google (Firebase) services. <https://medevel.com/15-os-push-notification/>

Gorush. (2023). A push notification micro server written in Go. <https://github.com/appleboy/gorush>

Gotify. (2023). Gotify · a simple server for sending and receiving messages. <https://gotify.net>

Push.js. (2023). Documentación Push.js. <https://pushjs.org/docs/introduction>

Ntfy. (2023). Push notifications made easy. <https://ntfy.sh/>

Stringfellow, Angela. (2023). App Push Notifications - 40 Best Open Source Notification Systems. <https://www.magicbell.com/blog/best-open-source-notification-systems>

