# Long polling, websockets y server-sent events: comunicación para el envío de datos en tiempo real

# Long polling, websockets and server-sent events: communication for sending data in real time

**AUTORES:** Gracia Orejuela, Kevin José[1]
Gorozabel Bazurto, Melvin Bolívar[2]
Chango Sailema, Wilson Gustavo[3]

**Dirección para correspondencia:** kevin.gracia@pucese.edu.ec

**Fecha de recepción:** enero 10, 2024

**Fecha de aceptación:** septiembre 02, 2024

**DOI:** https://doi.org/10.61154/mrcm.v10i3.3272

**RESUMEN**

En diversas aplicaciones, desde sistemas de control industrial hasta juegos en línea y streaming de video, la comunicación en tiempo real ha ido ganando cada vez más importancia. Sin embargo, el envío de datos en tiempo real a través de redes presentó desafíos debido a la variabilidad en el ancho de banda y la latencia. Se llevó a cabo un estudio en el Departamento de Tecnologías de la Información de la Pontificia Universidad Católica del Ecuador con el objetivo de mejorar las comunicaciones de sus aplicaciones de backend. En la actualidad, estas aplicaciones utilizan diferentes tipos de protocolos de comunicación, pero no habían logrado solucionar la problemática de la comunicación en tiempo real. En el estudio, se analizaron distintos protocolos de comunicación mediante un chat en linea para determinar cuál sería el más adecuado para una arquitectura de backend escalable. Los protocolos de comunicación que se evaluaron fueron: Long

[1] Ingeniero y Magíster en Tecnologías de la Información por la Pontificia Universidad Católica del Ecuador, Sede Esmeraldas, Esmeraldas, Ecuador. E-mail: kevin.gracia@pucese.edu.ec Código ORCID: https://orcid.org/0009-0005-0308-4873

[2] Ingeniero en Sistemas y Computación, Magíster en Tecnologías de la Información por la Pontificia Universidad Católica del Ecuador, Sede Esmeraldas, Esmeraldas, Ecuador. E-mail: gorozabel13@gmail.com Código ORCID: https://orcid.org/0009-0001-9429-8818

[3] Ph.D en Computación Avanzada, Facultad de Informática y Electrónica, Docente de la Carrera Tecnologías de la Información, Escuela Superior Politécnica de Chimborazo, Sede Orellana. Orellana, Ecuador. E-mail: wilson.chango@espoch.edu.ec CÓDIGO ORCID: https://orcid.org/0000-0003-3231-0153

polling, WebSockets y Server-Sent Events. Los resultados mostraron que WebSockets era el protocolo de comunicación más adecuado para una arquitectura de backend escalable. Estos resultados tienen implicaciones importantes para el desarrollo de aplicaciones web. Al utilizar WebSockets, se espera que pueda mejorar la calidad de sus aplicaciones web y proporcionar a los usuarios una experiencia más fluida e interactiva. El estudio demostró que WebSockets era el protocolo de comunicación más adecuado para una arquitectura de backend escalable. Esto se debe a que WebSockets proporciona una conexión bidireccional entre el cliente y el servidor, permitiendo que los datos se envíen en tiempo real sin tener que recargar la página.

**PALABRAS CLAVE:** Comunicación en tiempo real; Aplicaciones de backend; Protocolos de comunicación; WebSockets; Arquitectura escalable; Experiencia interactiva; aplicación informática; comunicación. (Tesauro de la UNESCO)

## ABSTRACT

In a variety of applications, from industrial control systems to online gaming and video streaming, real-time communication has become increasingly important. However, sending real-time data over networks presented challenges due to bandwidth variability and latency. A study was carried out at the Department of Information Technology of the Pontificia Universidad Católica del Ecuador with the objective of improving the communications of its backend applications. Currently, these applications use different types of communication protocols, but they had not been able to solve the problem of real-time communication. In the study, different communication protocols were analyzed through an online chat to determine which would be the most suitable for a scalable backend architecture. The communication protocols evaluated were: Long polling, WebSockets and Server-Sent Events. The results showed that WebSockets was the most suitable communication protocol for a scalable backend architecture. These results have important implications for web application development. By using WebSockets, it is expected that you can improve the quality of your web applications and provide users with a more seamless and interactive experience. The study showed that WebSockets was the most suitable communication protocol for a scalable backend architecture. This is because WebSockets provides a bidirectional

connection between the client and the server, allowing data to be sent in real time without having to reload the page.

**KEYWORDS:** Real-time communication, Backend applications, Communication protocols, WebSockets, Scalable architecture, Interactive experience; Computer applications; communication. (UNESCO Thesaurus)

## INTRODUCCIÓN.

La comunicación en tiempo real se ha vuelto cada vez más crucial en una amplia variedad de aplicaciones debido a la creciente demanda de interactividad y respuesta inmediata. Desde sistemas de control industrial, donde la transmisión de datos en tiempo real es vital para garantizar la seguridad y eficiencia en procesos automatizados, hasta juegos en línea y servicios de streaming de video, donde una latencia mínima es fundamental para proporcionar una experiencia fluida y en tiempo real a los usuarios (Silva et al., 2018).

Sin embargo, el envío de datos en tiempo real a través de redes puede enfrentar desafíos significativos. Uno de los principales problemas es la variabilidad en el ancho de banda y la latencia (Hazra et al., 2023). Estos factores pueden generar retrasos y pérdida de paquetes, lo que afecta la eficiencia y la calidad de la comunicación en tiempo real (Gorton & Teja Rayavarapu, 2022).

Para abordar estos desafíos, es esencial contar con protocolos de comunicación adecuados que permitan transmitir datos de manera rápida y eficiente en un entorno de red variable (Bello & Zeadally, 2022). Es aquí donde entra en juego la investigación propuesta, que busca analizar y comparar tres protocolos de comunicación específicos: Long Polling, WebSockets y Server-Sent Events (Aung, 2023).

La PUCE sede Esmeraldas dispone de diversas aplicaciones tecnológicas orientadas a la gestión académica, como el sistema de notas y el correo institucional, que proporcionan servicios esenciales a los estudiantes. La comunicación eficiente entre estas aplicaciones es fundamental para garantizar una experiencia fluida y una interacción en tiempo real. Sin embargo, en la actualidad, la respuesta de estas aplicaciones no resulta óptima en términos de eficiencia y tiempo de respuesta, lo que afecta negativamente la calidad del servicio.

En este contexto, la implementación de tecnologías avanzadas de comunicación como Long Polling, WebSockets y Server-Sent Events (SSE) se presenta como una alternativa viable. Estas tecnologías permiten establecer una comunicación bidireccional o unidireccional en tiempo real entre el cliente y el servidor, mejorando significativamente la velocidad de actualización de datos y la capacidad de respuesta de las aplicaciones.

La relevancia de esta investigación radica en el hecho de que una arquitectura escalable y eficiente es esencial para cualquier aplicación o sistema en la actualidad. La capacidad de adaptarse a cambios y mantener un rendimiento óptimo a medida que aumenta la demanda de usuarios es crucial para el éxito y la competitividad del aplicativo informático. Una arquitectura escalable bien implementada permite una gestión más eficiente de los recursos y mejora la experiencia del usuario, lo que puede atraer a más usuarios y generar mayores ganancias para la institución (Sheridan et al., 2023).

Para alcanzar los objetivos de la investigación, se llevará a cabo una metodología que incluye entrevistas con el director del departamento de Tecnologías de la Información de la universidad. Estas entrevistas permitirán obtener información detallada sobre el estado actual de la infraestructura de comunicación en tiempo real, así como las necesidades y desafíos actuales del departamento.

La validación de la información obtenida se llevó a cabo mediante el coeficiente de fiabilidad Alfa de Cronbach, una medida estadística que evaluó la consistencia interna de un conjunto de datos.

En el artículo, se describió en detalle el análisis de los protocolos de comunicación utilizados en ese momento en la universidad para el envío de datos en tiempo real. Se evaluó la efectividad de estos protocolos y se identificaron sus limitaciones y oportunidades de mejora con el objetivo de lograr una transmisión de datos más rápida y eficiente.

En última instancia, la investigación proporcionó una visión más clara sobre los protocolos de comunicación más adecuados para el envío de datos en tiempo real y cómo implementar una arquitectura escalable de backend que mejorara la eficiencia de las aplicaciones en la Pontificia Universidad Católica del Ecuador Sede Esmeraldas. Los resultados de esta investigación pueden tener implicaciones prácticas significativas para mejorar la comunicación en tiempo real en

diversas aplicaciones y sistemas, brindando a la institución una ventaja competitiva en el mercado actual y futuros desarrollos tecnológicos.

## METODOLOGÍA

El estudio se centra en el análisis de protocolos de comunicación para mejorar el envío de datos en tiempo real mediante una arquitectura escalable de backend (Mehmood et al., 2016). Se realizaron entrevistas con el director del departamento de Tecnologías de la Información de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas para obtener información sobre el estado actual y las necesidades del departamento.

Las variables por considerar en el estudio son:

Eficiencia en tiempo real: capacidad de los protocolos para enviar datos de manera oportuna y precisa (Fan et al., 2022).
Latencia: tiempo de retraso entre el envío de un mensaje y su recepción (Tsindeliani et al., 2022).
Jitter: variación en el tiempo de retraso entre los mensajes (Schoos et al., 2023).
Escalabilidad: capacidad de los protocolos para soportar un aumento en el número de usuarios o la cantidad de datos (Vellela & Balamanigandan, 2022).
Tolerancia a fallos: capacidad de los protocolos para continuar funcionando en caso de fallos en la red o en los servidores (Liang et al., 2023).
Facilidad de implementación: facilidad con la que los protocolos pueden ser implementados y configurados(Hosney et al., 2022) .
El presente estudio se centra en evaluar el desempeño de una arquitectura de backend escalable, con el objetivo de identificar los cuellos de botella en la transmisión de datos en tiempo real y proponer soluciones basadas en protocolos de comunicación avanzados. Para ello, se llevó a cabo una evaluación exhaustiva de la infraestructura de red existente, incluyendo mediciones de rendimiento y entrevistas a expertos.

Se empleó una metodología mixta, combinando técnicas cuantitativas y cualitativas. En la fase cuantitativa, se realizaron pruebas de rendimiento para evaluar métricas como la latencia, el jitter, la tasa de pérdida de paquetes y la capacidad de procesamiento. En la fase cualitativa, se llevaron a cabo entrevistas semiestructuradas con el director del departamento de Tecnologías de la

Información para obtener una comprensión profunda de las necesidades y desafíos actuales. Los datos obtenidos de las entrevistas fueron analizados mediante técnicas de codificación abierta y axial, con el objetivo de identificar patrones y temas emergentes. Posteriormente, se realizó un análisis comparativo de los protocolos Long Polling, WebSockets y Server-Sent Events, considerando criterios como la eficiencia, la escalabilidad y la complejidad de implementación.

Se espera que los resultados de este estudio permitan identificar el protocolo de comunicación más adecuado para la implementación de una arquitectura de backend escalable y eficiente, capaz de soportar un alto volumen de tráfico en tiempo real. Además, se espera generar recomendaciones prácticas para la optimización de la infraestructura de red existente y el desarrollo de aplicaciones web en tiempo real.

Los hallazgos de esta investigación contribuirán a avanzar en el conocimiento sobre las mejores prácticas para el diseño y desarrollo de aplicaciones web en tiempo real, y podrán ser utilizados como referencia para futuros proyectos en el ámbito de las comunicaciones de datos.

## RESULTADOS

Para realizar una evaluación objetiva de los protocolos de comunicación utilizados en el contexto de aplicaciones web en la PUCE, se definió un escenario experimental detallado que abarca tanto el hardware como el software involucrado. El objetivo es asegurar que los resultados obtenidos sean representativos y replicables en un entorno controlado. A continuación, se describe el escenario y los parámetros utilizados en las pruebas, junto con el análisis contextual de estos.

### Escenario de realización de las pruebas

### Hardware

Las pruebas se llevaron a cabo en una computadora con las siguientes especificaciones técnicas (ver **Tabla 1**).

***Tabla 1***

*Propiedades del pc que se desarrollaron las pruebas*

| **Parámetros** | **Valor** |
|---|---|
| Procesador | Inside core i7 vpro |
| Ram | 16Gb |
| Tarjeta grafica | Intel UHD 5000 |
| Disco solido | SSD de 256 GB |
| Sistema operativo Windows | Pro versión 22H2 |

*Nota.* Características del computador para las pruebas realizadas en los experimentos de la investigación

Para evaluar de manera exhaustiva el rendimiento de los protocolos, se diseñó un conjunto de pruebas que simulaban condiciones de carga real. Se realizaron 2000 peticiones por protocolo, lo que permitió obtener una muestra representativa del comportamiento de cada uno. Al controlar variables como la configuración del hardware y el software, se garantizaron condiciones de prueba estables y reproducibles. Los resultados obtenidos proporcionaron información detallada sobre métricas clave como la latencia, el throughput y la tasa de éxito, lo que permitió realizar una evaluación comparativa rigurosa de los protocolos evaluados.

**Software**

Los protocolos evaluados fueron WebSocket, Long Polling, y Server-Sent Events (SSE). Cada uno de estos fue sometido a 2000 solicitudes para medir su rendimiento en un entorno controlado. Las pruebas incluyeron la creación de una instancia del servidor local para cada protocolo, simulando un entorno de comunicación web en tiempo real. Se configuraron los siguientes parámetros en el entorno de pruebas

**(Websocket)**

Los resultados obtenidos tras llevar a cabo 2000 peticiones por cada protocolo, con el consiguiente re-inicialización de la máquina después de cada prueba, proporcionan información valiosa sobre el rendimiento del protocolo WebSocket (ver **Tabla 2**).

**Tabla 2**

*Tiempo de respuesta a 2000 solicitudes con el protocolo WebSocket*

| Requests | Executions | | | Response Times (ms) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Label | #Samples | FAIL | Error % | Average | Min | Max | Median | 90th pct | 95th pct | 99th pct |
| Total | 2000 | 0 | 0.00 % | 2781.77 | 321 | 4848 | 2911 | 4090.9 | 4282.9 | 4815.98 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| http://localhost:3000/ | 2000 | 0 | 0.00 % | 2781.77 | 321 | 4848 | 2911 | 4090.9 | 4282.9 | 4815.98 |
| Test | 2000 | 0 | 0.00 % | 2781.77 | 321 | 4848 | 2911 | 4090.9 | 4282.9 | 4815.98 |

*Nota.* Evaluación del rendimiento de alguna aplicación o servicio web en el que se realizaron 2000 solicitudes. Los tiempos de respuesta varían desde 321 milisegundos hasta 4848 milisegundos, con un promedio de 2781.77 milisegundos

Se llevaron a cabo un total de 2000 solicitudes durante las pruebas. No se registraron ejecuciones fallidas, lo que indica que todas las solicitudes se completaron con éxito. Los tiempos proporcionaron información crucial sobre el rendimiento de las solicitudes. El tiempo promedio de respuesta de todas las solicitudes fue de 2781.77 milisegundos (2.78 segundos). Este valor indica el tiempo medio que tardaron las solicitudes en procesarse y recibir una respuesta.

El tiempo mínimo de respuesta fue de 321 milisegundos (0.32 segundos), representando el tiempo más corto que una solicitud tardó en completarse. El tiempo máximo de respuesta se registró en 4848 milisegundos (4.84 segundos), indicando la solicitud más lenta en el conjunto de pruebas.

La mediana del tiempo de respuesta se situó en 2911 milisegundos (2.91 segundos). La mediana es el valor que se encuentra justo en el medio de todos los tiempos de respuesta, lo que significa que el 50% de las solicitudes se completaron en menos de este tiempo y el 50% en más tiempo.

Los percentiles 90th, 95th y 99th indican los tiempos de respuesta que superaron el 90%, 95% y 99% de las solicitudes, respectivamente. Por ejemplo, el percentil 95th muestra que el 95% de las solicitudes se completaron en menos de 4282.9 milisegundos (4.28 segundos).

Los resultados demuestran que las solicitudes exhibieron tiempos de respuesta variables, con un promedio de aproximadamente 2.78 segundos. El hecho de que no se hayan registrado ejecuciones fallidas indica que el sistema pudo gestionar todas las solicitudes con éxito. Sin embargo, es importante destacar que se debe realizar una evaluación exhaustiva para determinar si estos tiempos de respuesta cumplen con los requisitos de rendimiento del sistema o si se requieren optimizaciones para mejorar la eficiencia y la velocidad de respuesta.

El rendimiento del sistema se midió en términos de Throughput, que representaba la cantidad de transacciones procesadas por segundo. Durante las pruebas globales, se registró un rendimiento promedio de aproximadamente 284.86 transacciones por segundo. Para la URL específica "http://localhost:3000/", el sistema demostró un rendimiento ligeramente superior, con un promedio de aproximadamente 289.9 transacciones por segundo (ver **Tabla 3**).

***Tabla 3***
*Rendimiento y la eficacia de las solicitudes realizadas*

| Requests | Executions | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|
| **Label** | **#Samples** | **FAIL** | **Error %** | **Transactions/s** | **Received** | **Sent** |
| **Total** | **2000** | **0** | **0.00%** | **284.86** | **67.88** | **116** |
| http://localhost:3000/ | 2000 | 0 | 0.00% | 284.86 | 67.88 | 116 |
| Test | 2000 | 0 | 0.00% | 289.9 | 69.08 | 118.05 |

*Nota.* Rendimiento exitoso de una aplicación web durante una prueba de carga, incluyendo el número de solicitudes, el rendimiento (transacciones por segundo) y el tráfico de red (datos recibidos y enviados)

Estos valores de rendimiento indicaban la capacidad del sistema para gestionar una carga de trabajo de solicitudes a un ritmo constante.

En cuanto a la medición de la red, se evaluó en términos de la cantidad de datos recibidos (Received) y enviados (Sent) en kilobytes por segundo (KB/sec). Durante las pruebas generales, se observó un promedio de alrededor de 67.88 KB/sec recibidos y aproximadamente 116 KB/sec enviados. Para la URL específica "http://localhost:3000/", se registraron valores ligeramente más altos, con un promedio de alrededor de 69.08 KB/sec recibidos y 118.05 KB/sec enviados. Estos datos proporcionaron información sobre la utilización de la red durante las pruebas y resultaron relevantes para evaluar el ancho de banda y la carga en la infraestructura de red.

Los resultados indicaron que el sistema evaluado fue capaz de gestionar las 2000 solicitudes sin experimentar fallos, lo que sugería una alta estabilidad y confiabilidad. El rendimiento, medido en términos de Throughput, se mantuvo sólido, superando un promedio de 284 transacciones por segundo. Además, los datos de red señalaron que se utilizó un ancho de banda razonable durante las pruebas. Estos resultados prometedores pueden ser de utilidad para evaluar la capacidad del sistema para escalar y cumplir con los requisitos de rendimiento en situaciones de carga similares en un entorno de producción (ver **Figura 1**).

**Long polling**

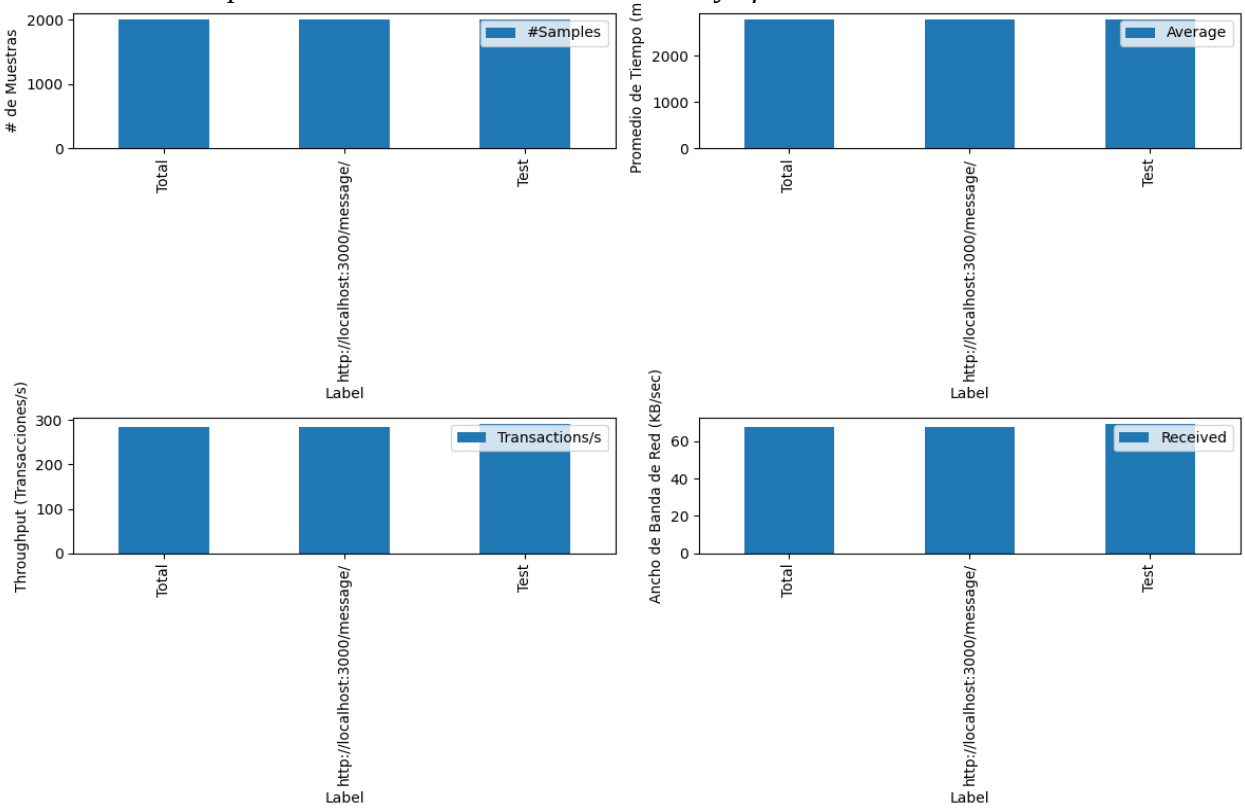
Se llevaron a cabo 2000 solicitudes, se registraron 138 fallos, lo que representó un 6.90% de error en las solicitudes. El tiempo promedio de respuesta fue de 304.87 ms. El tiempo mínimo fue de 11 ms, mientras que el tiempo máximo alcanzó los 2093 ms. La mediana se ubicó en 63 ms. Los

percentiles 90 y 95 se situaron en 982 ms y 1975 ms, respectivamente. El percentil 99 alcanzó los 2077.98 ms (ver

**Tabla *4***).

**Figura 1**

*Solicitudes con el protocolo WebSocket su Rendimiento y eficacia*




*Nota*. El grafico representa el rendimiento y la eficiencia de la utilización del protocolo de comunicación websocket en 2000 solicitudes.

**Tabla 4**

*Tiempo de respuesta a 2000 solicitudes con el protocolo Long polling*

| Requests | Executions | | | Response Times (ms) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Label** | **#Samples** | **FAIL** | **Error %** | **Average** | **Min** | **Max** | **Median** | **90th pct** | **95th pct** | **99th pct** |
| Total | 4000 | 176 | 4.40 % | 1461.34 | 11 | 4617 | 1124 | 3429.9 | 3865.95 | 4121.98 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| http://localhost:8080/publish | 2000 | 38 | 1.90 % | 2617.82 | 30 | 4617 | 2756 | 3865.9 | 3980 | 4168 |
| http://localhost:8080/subscribe?random=0.34135998343727825 | 2000 | 138 | 6.90 % | 304.87 | 11 | 2093 | 63 | 982 | 1975 | 2077.98 |
| Test | 2000 | 176 | 8.80 % | 2922.69 | 187 | 6695 | 2839 | 4084 | 5434.95 | 5989.82 |

*Nota*. rendimiento del sistema en términos de tiempo de respuesta y tasas de error. Las ejecuciones fallidas, así como los tiempos de respuesta más altos.

Estos resultados indican que el sistema fue sometido a una carga significativa, y se observó una proporción considerable de fallos en las solicitudes, especialmente en las pruebas bajo la etiqueta "Test". Además, se observó variabilidad en el tiempo promedio de respuesta entre las diferentes URL, siendo más alto en "Test". Los percentiles 90, 95 y 99 también proporcionaron información valiosa sobre la distribución de los tiempos de respuesta, lo cual es crucial para comprender el comportamiento del sistema bajo carga. Estos resultados sugieren que podría haber una necesidad de optimizar el sistema para mejorar el rendimiento y reducir la tasa de errores.

Se llevaron a cabo 2000 solicitudes a una URL específica. De estas 2000 solicitudes, 38 resultaron en fallos, lo que equivalía al 1.90% de error. El throughput promedio para esta URL fue de 347.64 transacciones por segundo. Además, la cantidad de datos recibidos en la red fue de 59.52 KB/sec, mientras que la cantidad enviada fue de 137.55 KB/sec. (ver **Tabla 5**).

**Tabla 5**

*Rendimiento y cantidad de datos recibidos en la red*

| **Requests** | **Executions** | | | **Throughput** | **Network (KB/sec)** | |
|---|---|---|---|---|---|---|
| **Label** | **#Samples** | **FAIL** | **Error %** | **Transactions/s** | **Received** | **Sent** |
| **Total** | **4000** | **176** | **4.40%** | **509.36** | **135.27** | **186.21** |
| http://localhost:8080/publish | 2000 | 38 | 1.90% | 347.64 | 59.52 | 137.55 |
| http://localhost:8080/subscribe?random=0.34135998343727825 | 2000 | 138 | 6.90% | 258.13 | 92.91 | 86.6 |
| Test | 2000 | 176 | 8.80% | 254.91 | 135.39 | 186.37 |

Nota. Evaluación el rendimiento y la estabilidad del sistema bajo carga, porcentajes de errores y las tasas de transacciones por segundo

Los resultados indican que se llevaron a cabo pruebas en un sistema y se recopilaron métricas relacionadas con solicitudes, errores, throughput y uso de la red. Es relevante notar que las tasas

de error variaron según la URL específica a la que se realizaron las solicitudes, lo que sugiere que algunas partes del sistema pudieron ser más propensas a errores que otras. También se observaron diferencias en el throughput y el uso de la red entre las distintas URL, lo que podría indicar disparidades en el rendimiento y la eficiencia de las diferentes partes del sistema sometido a prueba. Estos datos resultan fundamentales para la evaluación y mejora del rendimiento y la confiabilidad del sistema en cuestión (ver **Figura 2**).

**Server-sent event (SEE).**

Se realizaron un total de 2000 solicitudes durante las pruebas. No se registraron fallos en ninguna de las solicitudes, lo que indica que todas las solicitudes se completaron exitosamente. El porcentaje de errores fue del 0.00%, confirmando que no hubo fallos en las solicitudes.

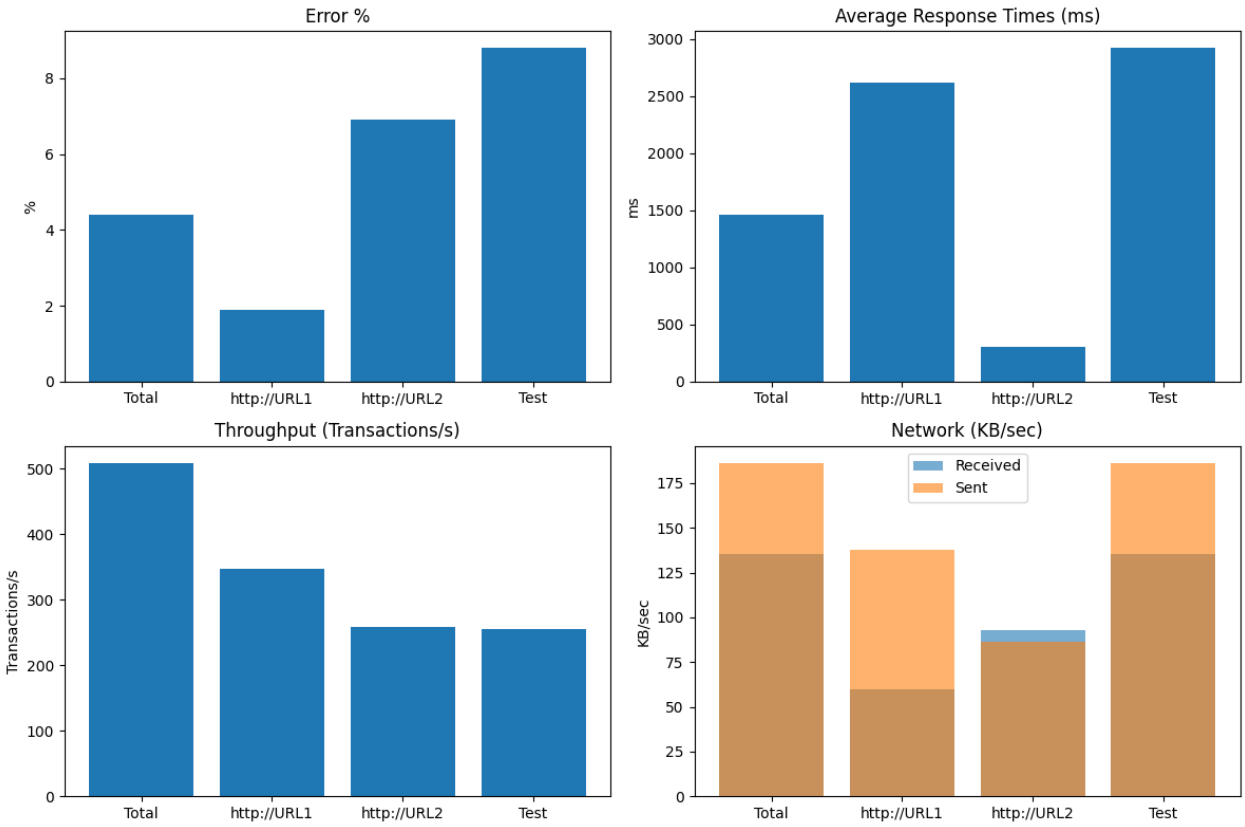
El tiempo promedio de respuesta fue de 1889.92 ms (milisegundos), representando el tiempo promedio requerido para completar una solicitud.

El tiempo mínimo registrado fue de 4 ms, mientras que el tiempo máximo alcanzó los 3983 ms.

La mediana (percentil 50) fue de 1937 ms, lo que significa que el 50% de las solicitudes tuvo un tiempo de respuesta igual o inferior a este valor.

**Figura 2**

*Error %, el tiempo promedio de respuesta, el rendimiento (Throughput) y la cantidad de datos recibidos y enviados*




Nota. *Promedio de rrendimiento y la eficiencia de las diferentes partes del sistema sometido a prueba*

Los percentiles 90, 95 y 99 indican que el 90%, 95% y 99% de las solicitudes, respectivamente, experimentaron tiempos de respuesta iguales o inferiores a los valores especificados.

Los valores de tiempo de respuesta mostraron una notable consistencia entre las diferentes etiquetas (Label), lo que sugiere que las pruebas produjeron resultados similares para todas las solicitudes evaluadas (ver **Tabla 6**).

**Tabla 6**

*Tiempo de respuesta a 2000 solicitudes con el protocolo Server-sent event*

| Requests | Executions | | | Response Times (ms) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Label** | **#Samples** | **FAIL** | **Error %** | **Average** | **Min** | **Max** | **Median** | **90th pct** | **95th pct** | **99th pct** |
| **Total** | **2000** | **0** | **0.00%** | **1889.92** | **4** | **3983** | **1937** | **2875** | **3117** | **3528.99** |
| http://localhost:9000/chat/ message | 2000 | 0 | 0.00% | 1889.92 | 4 | 3983 | 1937 | 2875 | 3117 | 3528.99 |
| Test | 2000 | 0 | 0.00% | 1889.92 | 4 | 3983 | 1937 | 2875 | 3117 | 3528.99 |

Nota. *Tiempos de respuesta promedio del sistema que responde en menos de 2 segundos, pero existen solicitudes que experimentan tiempos de respuesta significativamente más largos, como se refleja en los valores máximos y los percentiles 95 y 99.*

Los datos indican que las pruebas se realizaron con éxito, ya que no se registraron fallos y el porcentaje de errores fue del 0.00%. Los tiempos de respuesta promedio y los percentiles sugieren que el sistema o la aplicación bajo prueba mantuvo un rendimiento aceptable en términos de tiempo de respuesta durante las 2000 solicitudes realizadas.

No se registraron fallos en ninguna de las solicitudes, lo que indica que todas las solicitudes se completaron con éxito. El porcentaje de errores fue del 0.00%, confirmando que no hubo fallos en las solicitudes.

El rendimiento del sistema se midió en términos de Throughput, que representaba la cantidad de transacciones procesadas por segundo (Transactions/s). Durante las pruebas, el sistema alcanzó un rendimiento promedio de 399.68 transacciones por segundo, lo que señala su capacidad para manejar una carga de trabajo significativa (ver **Tabla 7**).

**Tabla 7**

*Rendimiento y cantidad de datos recibidos*

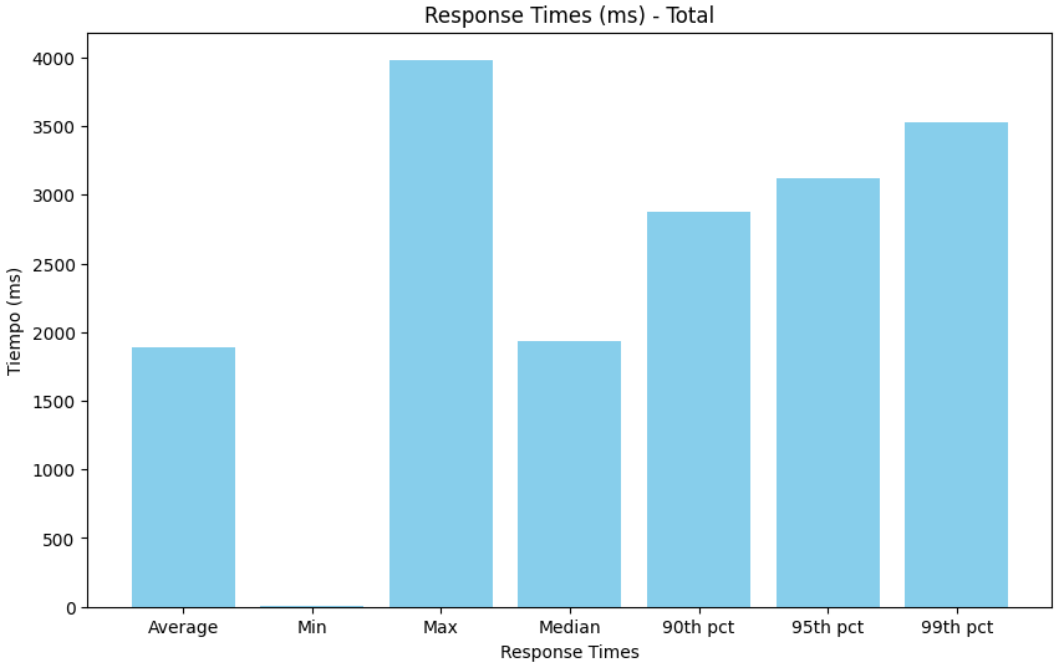
| Requests | Executions | | | Throughput | Network (KB/sec) | |
|---|---|---|---|---|---|---|
| **Label** | **#Samples** | **FAIL** | **Error %** | **Transactions/s** | **Received** | **Sent** |
| **Total** | **2000** | **0** | **0.00%** | **399.68** | **66.74** | **154.17** |
| http://localhost:9000/chat/message | 2000 | 0 | 0.00% | 399.68 | 66.74 | 154.17 |
| Test | 2000 | 0 | 0.00% | 399.6 | 66.73 | 154.14 |

Nota. *En el uso de la red, se reciben en promedio 66.74 KB/sec y se enviaron 154.17 KB/sec durante las pruebas. Estos valores indican la cantidad de datos que se movieron a través de la red durante las pruebas.*

Los datos indican que las pruebas se llevaron a cabo con éxito, ya que no se registraron fallos y el porcentaje de errores fue del 0.00%. El rendimiento del sistema, medido en Throughput, fue sólido, con un promedio de 399.68 transacciones por segundo, lo que sugiere que el sistema pudo manejar la carga de trabajo de 2000 solicitudes de manera efectiva. Además, los valores de uso de la red indican que se movió una cantidad razonable de datos a través de la red durante las pruebas. Estos resultados son prometedores y pueden ser útiles para evaluar la capacidad del sistema para cumplir con los requisitos de rendimiento en situaciones de carga similar en producción (ver **Figura 3**).

**Figura 3**

*Tiempos de respuesta y rendimiento*




## DISCUSIÓN

En primer lugar, se utilizaron los protocolos WebSocket para las pruebas, y se completaron las 2000 solicitudes sin fallos. Esto refleja una alta estabilidad y confiabilidad del sistema bajo estas condiciones, lo cual es un indicador crucial en evaluaciones de rendimiento. Los tiempos de respuesta promedio fueron de aproximadamente 2.78 segundos, con una variabilidad notable entre los tiempos mínimos (0.32 segundos) y máximos (4.84 segundos), lo que sugiere fluctuaciones en la capacidad de procesamiento de las solicitudes. El rendimiento, medido en términos de Throughput, fue consistente, alcanzando un promedio de 284 transacciones por segundo, lo que sugiere que el protocolo WebSocket gestionó de manera eficiente el volumen de solicitudes. En cuanto a la utilización de la red, los valores de KB por segundo recibidos y enviados fueron moderados, lo que implica que la carga de la infraestructura de red fue razonable y no representó un cuello de botella. Estos resultados indican que el protocolo WebSocket funcionó de manera sólida en este entorno de prueba, con tiempos de respuesta aceptables y un rendimiento estable, lo cual coincide con las observaciones de Liu et al. (2018), quienes subrayan la importancia de estos estudios como base fundamental para la investigación de protocolos en tiempo real.

En segundo lugar, se utilizó el protocolo Long Polling. En esta evaluación, se registraron 138 fallos en las 2000 solicitudes, lo que representa un 6.90% de errores. Este alto porcentaje de fallos, especialmente en la etiqueta "Test", revela una proporción considerable de fallos en la gestión de la carga, como lo evidencian Zhang et al. (2020) en sus investigaciones sobre la escalabilidad de este tipo de protocolos. Los tiempos de respuesta mostraron variaciones significativas, con un promedio de 304.87 ms, y un máximo que superó los 2 segundos, lo cual indica que el sistema experimentó problemas para mantener tiempos de respuesta consistentes bajo este protocolo. Los percentiles de tiempos de respuesta proporcionaron información crucial sobre cómo se comportaba el sistema bajo carga, lo que es especialmente relevante para la optimización de sistemas complejos, como destacan Bayılmıs¸a et al. (2022). El Throughput fue de 347.64 transacciones por segundo para una URL específica, pero las diferencias en el uso de la red entre las URL analizadas sugieren disparidades en el rendimiento, lo que podría estar relacionado con la estructura del sistema. Estos resultados muestran que el protocolo Long Polling experimentó una carga considerable y una tasa significativa de fallos, lo que apunta a la necesidad de optimización para mejorar el rendimiento y reducir los errores, tal como lo subrayan (Zohrabi et al., 2021).

Finalmente, el protocolo Server-Sent Event (SEE) se probó bajo las mismas condiciones, con 2000 solicitudes completadas sin fallos, lo que, al igual que con WebSocket, confirma la alta estabilidad y confiabilidad del sistema en estos experimentos. Los tiempos de respuesta promedio fueron de 1.89 segundos, con variabilidad entre el mínimo (4 ms) y el máximo (3.98 segundos). El Throughput fue sólido, alcanzando un promedio de 399 transacciones por segundo, lo que indica una capacidad robusta para manejar solicitudes en tiempo real, como lo destacan Guan et al. (2019) en sus estudios. La utilización de la red fue moderada, con valores de KB/sec razonables, lo que indica una carga moderada en la infraestructura de red y sugiere que SEE es eficiente en cuanto al uso de recursos.

## CONCLUSIONES

El estudio tenía como objetivo principal evaluar el rendimiento de tres protocolos de comunicación (WebSocket, Long polling y Server-sent event) a través de la realización de 2000 solicitudes en cada uno de ellos. Se buscaba comprender cómo estos protocolos se comportaban bajo carga y obtener métricas relacionadas con el tiempo de respuesta, el rendimiento y el uso de la red.

En el experimento con WebSocket, se completaron las 2000 solicitudes sin fallos. Los tiempos de respuesta promedio fueron de aproximadamente 2.78 segundos, con variabilidad en los tiempos mínimos y máximos. El rendimiento fue sólido, con un promedio de 284 transacciones por segundo.

En el experimento con Long polling, se registraron 138 fallos, lo que representó un 6.90% de error en las solicitudes. Los tiempos de respuesta variaron significativamente, y se observó una variabilidad en los tiempos promedio entre diferentes URL. El rendimiento y el uso de la red también mostraron variabilidad.

En el experimento con Server-sent event (SEE), no se registraron fallos en ninguna de las 2000 solicitudes. Los tiempos de respuesta promedio fueron de aproximadamente 1.89 segundos. El rendimiento fue sólido, con un promedio de 399.68 transacciones por segundo, y el uso de la red fue razonable.

En el caso de WebSocket, se observó un buen rendimiento y estabilidad, con tiempos de respuesta aceptables y un bajo porcentaje de errores.

Long polling experimentó una proporción significativa de fallos, lo que sugiere la necesidad de optimizar el sistema para mejorar el rendimiento y reducir la tasa de errores.

Server-sent event (SEE) demostró una alta estabilidad y buen rendimiento, con tiempos de respuesta aceptables y un rendimiento sólido.

Una limitación importante es que las pruebas se realizaron en un entorno controlado y no se tuvieron en cuenta factores externos que podrían haber afectado el rendimiento en un entorno de producción real.

Los resultados proporcionan información valiosa para la selección de protocolos de comunicación en función de los requisitos de rendimiento y la estabilidad deseados.

El estudio proporcionó una evaluación detallada del rendimiento de tres protocolos de comunicación en un entorno de pruebas controlado. Los resultados destacaron la importancia de seleccionar el protocolo adecuado en función de los requisitos de rendimiento y la estabilidad deseados, especialmente en el caso de WebSocket. Además, subrayaron la necesidad de optimizar el sistema en el caso de protocolos que experimentan una alta tasa de errores. Estos hallazgos son

fundamentales para mejorar el rendimiento y la confiabilidad del sistema en cuestión en situaciones de carga similares en un entorno de producción.

## REFERENCIAS BIBLIOGRÁFICAS

Aung, E. (2023). Comparison of Real-time Communication Performance between Web sockets using Socket.io and Long-Polling using Ajax. *Research & Creative Achievement Day*. https://openriver.winona.edu/rca/2023/schedule/15

Bayılmıs¸a, C., Bayılmıs¸a, B., Ebleme, M. A., Çavus¸o, Ü., Glu, Ç., Küçük, K., & Sevin, A. (2022). *A survey on communication protocols and performance evaluations for Internet of Things*. https://doi.org/10.1016/j.dcan.2022.03.013

Bello, O., & Zeadally, S. (2022). Internet of underwater things communication: Architecture, technologies, research challenges and future opportunities. *Ad Hoc Networks*, *135*, 102933. https://doi.org/10.1016/J.ADHOC.2022.102933

Fan, Y., Wang, X., Funk, T., Rashid, I., Herman, B., Bompoti, N., Mahmud, M. S., Chrysochoou, M., Yang, M., Vadas, T. M., Lei, Y., & Li, B. (2022). A Critical Review for Real-Time Continuous Soil Monitoring: Advantages, Challenges, and Perspectives. *Environmental Science and Technology*, *56*(19), 13546–13564. https://doi.org/10.1021/ACS.EST.2C03562/SUPPL_FILE/ES2C03562_SI_001.PDF

Gorton, I., & Teja Rayavarapu, V. (2022). Foundations of Scalable Software Architectures. *2022 IEEE 19th International Conference on Software Architecture Companion, ICSA-C 2022*, 233–236. https://doi.org/10.1109/ICSA-C54293.2022.00052

Guan, S., Hu, W., & Zhou, H. (2019). Real-time data transmission method based on websocket protocol for networked control system laboratory. *Chinese Control Conference, CCC*, *2019-July*, 5339–5344. https://doi.org/10.23919/CHICC.2019.8865690

Hazra, A., Rana, P., Adhikari, M., & Amgoth, T. (2023). Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges. *Computer Science Review*, *48*, 100549. https://doi.org/10.1016/J.COSREV.2023.100549

Hosney, E. S., Halim, I. T. A., & Yousef, A. H. (2022). An Artificial Intelligence Approach for Deploying Zero Trust Architecture (ZTA). *5th International Conference on Computing and Informatics, ICCI 2022*, 343–350. https://doi.org/10.1109/ICCI54321.2022.9756117

Liang, C., Ma, P., & Wu, S. (2023). Research on fault tolerance of communication network based on network structure. *Https://Doi.Org/10.1117/12.2682459*, *12715*, 106–110. https://doi.org/10.1117/12.2682459

Liu, Q., Yang, G., Zhao, R., & Xia, Y. (2018). Design and Implementation of Real-time Monitoring System for Wireless Coverage Data Based on WebSocket. *2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT)*, 63–67. https://doi.org/10.1109/CCIOT45285.2018.9032640

Mehmood, N. Q., Culmone, R., & Mostarda, L. (2016). A Flexible and Scalable Architecture for Real-Time ANT+ Sensor Data Acquisition and NoSQL Storage. *International Journal of Distributed Sensor Networks*, *2016*. https://doi.org/10.1155/2016/3651591

Schoos, K., Meschkov, S., Tahoori, M. B., & Gnad, D. R. E. (2023). JitSCA: Jitter-based Side-Channel Analysis in Picoscale Resolution. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, *2023*(3), 294–320. https://doi.org/10.46586/TCHES.V2023.I3.294-320

Sheridan, S., Sunderland, K., & Courtney, J. (2023). Swarm electrification: A comprehensive literature review. *Renewable and Sustainable Energy Reviews*, *175*, 113157. https://doi.org/10.1016/J.RSER.2023.113157

Silva, D. R. C., Oliveira, G. M. B., Silva, I., Ferrari, P., & Sisinni, E. (2018). Latency evaluation for MQTT and WebSocket Protocols: An Industry 4.0 perspective. *Proceedings - IEEE Symposium on Computers and Communications*, *2018-June*, 1233–1238. https://doi.org/10.1109/ISCC.2018.8538692

Tsindeliani, D., Povstyana, Y., Lishchyna, N., & Yashchuk, A. (2022). Latency Reduction in Real-time GPS tracking in Android and the Web-based GPS Monitoring System. *Proceedings of the 2022 IEEE 12th International Conference on Dependable Systems, Services and Technologies, DESSERT 2022*. https://doi.org/10.1109/DESSERT58054.2022.10018609

Vellela, S. S., & Balamanigandan, R. (2022). Design of Hybrid Authentication Protocol for High Secure Applications in Cloud Environments. *International Conference on Automation, Computing and Renewable Systems, ICACRS 2022 - Proceedings*, 408–414. https://doi.org/10.1109/ICACRS55517.2022.10029183

Zhang, Y., Li, Y., Diao, W., Jia, R., Tan, Z., & Du, Z. (2020). Design of multi-level 1553B bus test system for Complex Manned Spacecraft. *ACM International Conference Proceeding Series*, 108–111. https://doi.org/10.1145/3434581.3434600

Zohrabi, N., Martin, P. J., Kuzlu, M., Linkous, L., Eini, R., Morrissett, A., Zaman, M., Tantawy, A., Gueler, O., Islam, M. Al, Puryear, N., Kalkavan, H., Lundquist, J., Karincic, E., & Abdelwahed, S. (2021). OpenCity: An Open Architecture Testbed for Smart Cities. *2021 IEEE International Smart Cities Conference, ISC2 2021*. https://doi.org/10.1109/ISC253183.2021.9562813

©2024 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-No Comercial-Compartir Igual 4.0 Internacional (CC BY-NC-SA 4.0) ( https://creativecommons.org/licenses/by-nc-sa/4.0/).