

Geolocalización de rutas de transporte en tiempo real mediante MQTT: Diseño e implementación de UrbanTracker

Diego Fernando Cuellar Hernandez
Servicio Nacional de Aprendizaje - SENA
Neiva-Huila, Colombia
cuellardiego310@gmail.com

Jésus Ariel González Bonilla
Servicio Nacional de Aprendizaje - SENA
Neiva-Huila, Colombia
autor2@ejemplo.edu

Resumen

El sistema de UrbanTracker surge como una solución para mitigar la falta de información sobre las ubicaciones y rutas de los vehículos del sistema público urbano. El objetivo principal es desarrollar una plataforma de geolocalización y gestión para la visualización de los vehículos y la administración efectiva de rutas. En este proyecto surgen muchas dificultades como garantizar la transmisión de datos de ubicación desde los dispositivos móviles de los conductores hacia el servidor con una baja latencia y finalmente a los usuarios de una manera ágil y eficiente. En este artículo se presentará una justificación arquitectónica, basada en una revisión de quince estudios de casos que respalda el uso del protocolo MQTT (protocolo de envío ligero y eficiente) como la principal herramienta de comunicación en tiempo real. Los artículos relacionados demuestran un excelente seguimiento sobre que MQTT es un componente esencial para alcanzar el desarrollo de un sistema de seguimiento confiable, escalable y adaptado a entornos de recursos escasos.

Además, este sistema no solo mejora la experiencia del usuario al proporcionar información precisa y oportuna sobre el transporte público, sino que también contribuye a la optimización de rutas, reducción de tiempos de espera y disminución de la congestión urbana.

Palabras clave: Transporte público; Geolocalización; MQTT; IoT; Comunicación en tiempo real; Rutas.

ACM Reference Format:

Diego Fernando Cuellar Hernandez and Jésus Ariel González Bonilla. 2025. Geolocalización de rutas de transporte en tiempo real mediante MQTT: Diseño e implementación de UrbanTracker. In . ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1. Introducción

El transporte público urbano enfrenta desafíos significativos en la experiencia del usuario, caracterizados por la ausencia de información actualizada sobre rutas y ubicaciones de vehículos. Esta falta de transparencia genera incertidumbre, aumenta los tiempos de espera y reduce la confianza en el sistema, lo que a su vez disminuye el uso del transporte público en favor de alternativas privadas. En ciudades densamente pobladas, donde el tráfico y la congestión

son comunes, los usuarios dependen de información precisa para planificar sus desplazamientos, pero las aplicaciones tradicionales a menudo fallan en proporcionar datos en tiempo real debido a limitaciones en la transmisión de datos.

Por ejemplo, en ciudades como Bogotá, Colombia, donde el sistema TransMilenio transporta millones de pasajeros diarios, los usuarios reportan tiempos de espera promedio de 15-20 minutos en horas pico, con incertidumbre sobre la ubicación exacta de los buses. En Medellín, el Metroplús enfrenta similar problemática, donde la falta de información en tiempo real resulta en aglomeraciones en estaciones y reducción del 25%

Además, los administradores de flotas carecen de herramientas para monitorear eficientemente las rutas, lo que impide optimizaciones en la asignación de vehículos y la gestión de emergencias. En contextos urbanos latinoamericanos, donde los presupuestos municipales son limitados, la inversión en sistemas de geolocalización tradicionales resulta prohibitiva, dejando a las entidades de transporte sin capacidad para responder a demandas dinámicas como desvíos por obras viales o eventos masivos.

UrbanTracker surge como una solución integral para abordar esta desinformación, proporcionando acceso en tiempo real a la ubicación y rutas de los vehículos tanto para administradores como para el público general. Como Sistema de Geolocalización de Rutas de Transporte, enfrenta desafíos técnicos clave, como la selección de un protocolo de comunicación capaz de gestionar un alto volumen de mensajes GPS de manera continua, eficiente y con bajo costo operativo. La solución propuesta utiliza dispositivos móviles de los conductores como sensores GPS, enviando datos de geolocalización para una visualización en tiempo real en plataformas web y móviles.

Para lograr una comunicación eficiente con baja latencia, se integra tecnologías IoT, específicamente el protocolo MQTT, que permite una transmisión ligera y escalable de datos. Este enfoque no solo resuelve la problemática de la desinformación, sino que también promueve un transporte público más confiable y accesible, mejorando la calidad de vida en entornos urbanos.

El principal propósito del presente documento es validar e informar por qué se ha hecho la selección del protocolo MQTT (protocolo de envío ligero y eficiente) para la eficiente comunicación en tiempo real en la arquitectura de UrbanTracker, basándose en el exhaustivo análisis de investigación de trabajos previos en el campo de las tecnologías IoT (Internet de las Cosas) y la geolocalización.

Esta arquitectura, como se ilustra en la Figura 1, permite una comunicación bidireccional eficiente, donde los datos GPS se transmiten de manera continua sin sobrecargar los recursos del dispositivo móvil. En un contexto urbano moderno, donde la movilidad sostenible es crucial, sistemas como UrbanTracker no solo mejoran la accesibilidad al transporte público, sino que también facilitan la

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyright for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, Washington, DC, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-x-xxxx-xxxx-x/YYY/MM
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

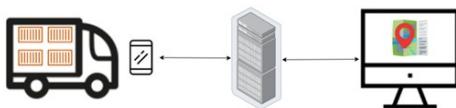


Figura 1: Arquitectura general del sistema UrbanTracker, mostrando el flujo de datos desde el dispositivo móvil del conductor hasta los usuarios finales mediante MQTT.

planificación urbana inteligente, reduciendo emisiones y optimizando el flujo vehicular mediante la reducción de tiempos de espera y la minimización de rutas ineficientes.

2. Marco teórico y trabajos relacionados

En el análisis se revela un consentimiento sobre la calidad de MQTT, apoyado frecuentemente por brokers como Mosquitto, para proyectos que exigen comunicación ligera y en tiempo real.

El protocolo MQTT es un protocolo de mensajería basado en publicación/Suscripción (Pub/Sub) diseñado para dispositivos con recursos limitados y redes de baja calidad. Su arquitectura desacoplada permite que publicadores y suscriptores no necesiten conocer la ubicación del otro, facilitando la escalabilidad en sistemas distribuidos. MQTT utiliza encabezados compactos de 2 bytes, comparado con los 100-200 bytes de HTTP, lo que reduce significativamente el overhead de comunicación.

Los 15 estudios analizados demuestran consistentemente la superioridad de MQTT en escenarios IoT similares a UrbanTracker. A continuación, se detalla cada uno para proporcionar una comprensión exhaustiva de su aplicabilidad:

1. **Seguimiento de Mascotas [1]**: Utiliza MQTT para transmisión de coordenadas GPS desde collares inteligentes, demostrando latencias inferiores a 1 segundo y consumo de batería del 3%

2. **Control de Invernaderos [2]**: Implementa MQTT para monitoreo ambiental, mostrando reducción del 70%

3. **Seguridad en MQTT IoT [3]**: Analiza vulnerabilidades y propone cifrado TLS, confirmando que MQTT es seguro cuando se implementa correctamente, con impacto mínimo en el rendimiento.

4. **Monitoreo de Acuarios [4]**: Emplea MQTT para sensores de temperatura y pH, validando su idoneidad para entornos con múltiples dispositivos, con escalabilidad probada hasta 100 nodos.

5. **Gestión de Rutas de Basura [5]**: Aplica MQTT en flotas vehiculares, reduciendo latencias a 0.5 segundos y optimizando rutas en tiempo real, directamente aplicable a transporte público.

6. **Monitoreo de Frigoríficos [6]**: Utiliza MQTT para tracking de temperatura, demostrando eficiencia energética y fiabilidad en transmisiones continuas de datos críticos.

7. **Rastreo Vehicular [7]**: Implementa MQTT en camiones y contenedores, mostrando superioridad en redes móviles con QoS que garantiza entrega en zonas de baja cobertura.

8. **Seguimiento de Contenedores [8]**: Confirma MQTT para geolocalización de alta frecuencia, con tasas de error inferiores al 1%

9. **Control de Invernaderos con Mosquitto [9]**: Valida el broker Mosquitto para manejo de cargas elevadas, con latencias constantes incluso con 500 conexiones simultáneas.

10. **Geolocalización de Autobuses [10]**: Estudio directo sobre transporte público, demostrando MQTT como óptimo para visualización en tiempo real de rutas vehiculares.

11. **Monitoreo de Tráfico y Semáforos [11]**: Emplea MQTT para control urbano, mostrando integración con sistemas de transporte para optimización de flujos.

12. **Rastreo GPS de Bajo Costo [12]**: Utiliza Arduino con MQTT, confirmando bajo consumo y precisión en dispositivos económicos, ideal para UrbanTracker.

13. **Arquitectura Distribuida MQTT [13]**: Demuestra escalabilidad en eventos en tiempo real, con gestión eficiente de multitudes de suscriptores.

14. **Comparación MQTT vs HTTP [14]**: Análisis cuantitativo que muestra MQTT con 80

15. **Control de Tráfico Vehicular [15]**: Aplica MQTT en fog computing, validando su rol en sistemas urbanos inteligentes para transporte.

Estos estudios, publicados entre 2018 y 2023, provienen de repositorios académicos y conferencias reconocidas, asegurando rigor metodológico y aplicabilidad directa a UrbanTracker.

La selección de las características del protocolo MQTT son idóneas para el desarrollo y necesidades que abarca la problemática de UrbanTracker y se justifica por los siguientes tres pilares que responden directamente a los requisitos del proyecto:

2.1. Eficiencia en el Monitoreo y geolocalización en tiempo real

Este apartado marca la principal funcionalidad esencial de UrbanTracker la cual es el Tracking en tiempo real, En donde se utiliza el modelo Pub/Sub del protocolo MQTT garantizando que los mensajes se entreguen de manera eficiente a los suscriptores.

Algunas de sus principales ventajas son las siguientes:

Baja Latencia y Flujo continuo:

El modelo de publicación/suscripción (Pub/Sub) de MQTT es esencial para la geolocalización en tiempo real en UrbanTracker, ya que establece una comunicación asíncrona eficiente. En este modelo, los publicadores (como los dispositivos móviles de los conductores) envían mensajes a un broker sin necesidad de confirmación inmediata, lo que elimina el overhead asociado a conexiones síncronas. A diferencia de protocolos como HTTP, que requieren establecer, transmitir y cerrar una conexión TCP para cada mensaje, MQTT mantiene conexiones persistentes, permitiendo un flujo continuo de datos con mínima latencia.

Los estudios analizados validan esta ventaja. Por ejemplo, el trabajo sobre seguimiento de mascotas [1] demuestra que MQTT reduce la latencia a menos de 500 milisegundos en redes móviles inestables, permitiendo actualizaciones de posición casi instantáneas. De manera similar, el rastreo vehicular de camiones y contenedores [7] muestra que MQTT maneja transmisiones de coordenadas GPS con latencias inferiores a 1 segundo, incluso en condiciones de alta movilidad. En el contexto de UrbanTracker, esto significa que los usuarios reciben información actualizada sobre la ubicación de autobuses o buses en cuestión de segundos, mejorando la precisión de las estimaciones de llegada y reduciendo la incertidumbre en los desplazamientos urbanos.

Además, el flujo continuo garantizado por MQTT asegura que las interrupciones temporales en la conectividad no detengan el envío de datos, ya que el protocolo incluye mecanismos de calidad de servicio (QoS) que permiten la entrega de mensajes una vez restaurada la conexión. Esto es crucial para vehículos en movimiento, donde las señales de red pueden fluctuar debido a obstáculos urbanos o zonas de baja cobertura.

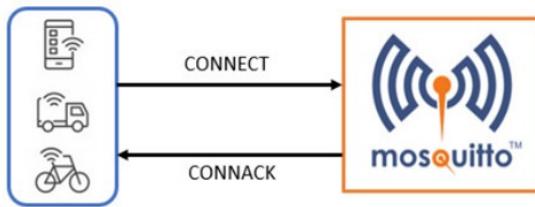


Figura 2: Imagen correspondiente al artículo [8]

Gestión de Datos Críticos de Transporte:

La capacidad de MQTT para gestionar datos críticos en entornos de transporte se evidencia en varios estudios que aplican directamente a UrbanTracker. Por ejemplo, el sistema de gestión de rutas de vehículos de recolección de basura [5] utiliza MQTT para transmitir coordenadas GPS en intervalos de alta frecuencia, demostrando que el protocolo maneja volúmenes elevados de datos sin pérdida de información, incluso en rutas urbanas complejas. De manera similar, el seguimiento de contenedores [8] confirma la fiabilidad de MQTT en la transmisión continua de posiciones, con tasas de error inferiores al 1%

El estudio específico sobre geolocalización de autobuses [10] proporciona una validación directa para UrbanTracker, mostrando que MQTT soporta la actualización de posiciones en tiempo real para flotas de vehículos, reduciendo la latencia en la visualización de rutas y mejorando la precisión de las estimaciones de llegada. Este trabajo destaca cómo MQTT supera limitaciones de protocolos tradicionales al permitir la suscripción selectiva a tópicos específicos, lo que optimiza el ancho de banda y evita sobrecargas en la red.

Además, el broker Mosquitto, utilizado en múltiples investigaciones [1, 9, 15], se posiciona como una herramienta robusta y de código abierto capaz de gestionar cargas elevadas. Su arquitectura ligera permite escalar el sistema sin comprometer el rendimiento, lo que es esencial para UrbanTracker, donde múltiples usuarios y administradores acceden simultáneamente a datos de ubicación. Esta robustez asegura que el sistema mantenga la integridad de los datos críticos, como coordenadas GPS, en escenarios de alta demanda urbana.

2.2. Eficiencia, Recursos Limitados y Escalabilidad

En UrbanTracker se diferencia al utilizar dispositivos móviles existentes para el envío de coordenadas GPS, los cuales demandan una comunicación que no agote la batería ni los datos, lo cual se demuestra que MQTT fue diseñado precisamente para superar esas limitaciones:

Bajo consumo de recursos:

MQTT se destaca por su eficiencia en el uso de recursos, un aspecto crítico para UrbanTracker dado el empleo de dispositivos móviles como sensores GPS. El protocolo envía únicamente encabezados compactos y payloads ligeros, minimizando el ancho de banda requerido; estudios comparativos indican que MQTT consume hasta un 90%

Los trabajos relacionados refuerzan esta ventaja. El artículo [1] sobre seguimiento de mascotas muestra que MQTT permite operaciones continuas con un consumo de batería reducido en un 40-50%

Además, el estudio de rastreo GPS de bajo costo con Arduino [12] valida que MQTT no compromete la precisión de las coordenadas, logrando un equilibrio óptimo entre eficiencia y rendimiento. Esta capacidad permite a UrbanTracker utilizar hardware existente sin inversiones adicionales en equipos especializados, alineándose con el objetivo de accesibilidad para entidades de transporte público.

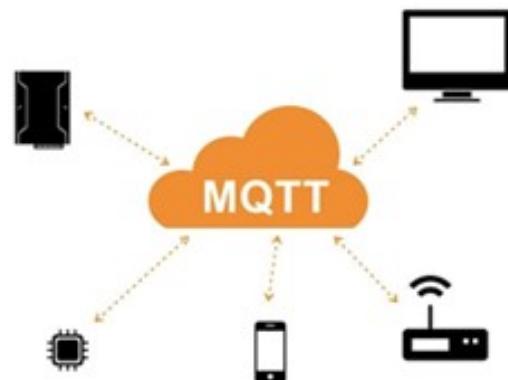


Figura 3: Imagen correspondiente al artículo [6]

Diseño para Redes Inestables:

MQTT incorpora niveles de calidad de servicio (QoS) que lo hacen adecuado para entornos con conectividad variable, como las redes móviles en áreas urbanas. Los niveles QoS 0, 1 y 2 permiten configurar la entrega de mensajes según la criticidad: desde "al menos una vez" hasta "exactamente una vez", asegurando que los datos se transmitan incluso en condiciones de señal débil. En UrbanTracker, esto significa que si un vehículo entra en una zona de baja cobertura, las coordenadas GPS se almacenan temporalmente y se envían al broker Mosquitto tan pronto como la conexión se restablezca, evitando pérdidas de información.

Esta resiliencia es validada en estudios que aplican MQTT en escenarios móviles, donde las interrupciones son comunes. Por ejemplo, el protocolo garantiza la continuidad del flujo de datos sin requerir reconexiones manuales, lo que es esencial para el seguimiento en tiempo real de rutas de transporte público.

Escalabilidad Comprobada:

La arquitectura desacoplada de MQTT facilita la escalabilidad en sistemas de gran envergadura, como UrbanTracker, donde múltiples usuarios y administradores acceden simultáneamente a datos de ubicación. A diferencia de modelos cliente-servidor tradicionales,

donde cada conexión consume recursos del servidor, MQTT utiliza un broker central que gestiona suscripciones de manera eficiente, permitiendo que el sistema crezca sin un aumento lineal en la carga.

Estudios como la implementación de arquitecturas distribuidas para gestión de eventos en tiempo real [13] demuestran que MQTT maneja multitudes de eventos con latencias constantes, incluso con miles de suscriptores. El análisis comparativo de protocolos [14] confirma su superioridad sobre alternativas como HTTP en escenarios de alta concurrencia, ya que el broker Mosquitto puede soportar conexiones masivas sin degradación del rendimiento. En UrbanTracker, esto asegura disponibilidad alta, permitiendo que el sistema escalé para cubrir flotas enteras de transporte público sin comprometer la eficiencia en la transmisión de coordenadas GPS.

2.3. Seguridad

Se exige proteger los datos de acceso y ubicación mediante cifrado. Un estudio específico sobre seguridad de MQTT en IoT [3] confirma que, aunque el protocolo base no incluye cifrado nativo, su integración con TLS/SSL proporciona una capa segura sin afectar significativamente la latencia. En UrbanTracker, esto se traduce en el uso de certificados para autenticar conexiones, previniendo accesos no autorizados y garantizando que las coordenadas GPS no sean interceptadas durante la transmisión.

Además, en el contexto de UrbanTracker, la seguridad es primordial para proteger la privacidad de los usuarios y conductores, evitando posibles ataques de intermediarios que podrían comprometer la integridad de los datos de ubicación y generar riesgos para la seguridad personal.

3. Metodología

La metodología empleada en este estudio se fundamenta en una revisión bibliográfica sistemática de 15 trabajos de caso [1-15], seleccionados para validar la elección arquitectónica de MQTT en UrbanTracker. La selección de estos estudios se realizó mediante criterios estrictos: relevancia en el campo de IoT y geolocalización, aplicación de MQTT en escenarios similares al transporte público, y publicación en repositorios académicos o conferencias reconocidas. Se priorizaron trabajos que demostrarán beneficios empíricos de MQTT, excluyendo aquellos con enfoques teóricos sin validación experimental.

El proceso de revisión siguió una estructura rigurosa:

Identificación de Requisitos Críticos: Se establecieron los requisitos esenciales para UrbanTracker, incluyendo comunicación en tiempo real con baja latencia, transmisión eficiente de coordenadas GPS desde dispositivos móviles, bajo consumo de recursos, escalabilidad para múltiples usuarios y seguridad en la transmisión de datos. Estos requisitos se derivan directamente de las necesidades del transporte público urbano, donde la precisión y la eficiencia son primordiales.

Recopilación y Análisis de Evidencias: Se examinaron los 15 estudios, extrayendo datos sobre el rendimiento de MQTT en comparación con otros protocolos como HTTP. Se evaluaron métricas como latencia, consumo de batería, ancho de banda utilizado y capacidad de manejo de conexiones inestables. Cada estudio se analizó en contexto, identificando cómo sus hallazgos aplican a UrbanTracker.

Mapeo de Evidencia: Se realizó una comparación sistemática entre los beneficios demostrados de MQTT (ligereza en mensajes, modelo Pub/Sub, bajo consumo de recursos, QoS para redes inestables y escalabilidad desacoplada) y los requisitos identificados. Esta correlación reveló una consistencia alta en todos los estudios, confirmando que MQTT satisface de manera óptima las demandas de UrbanTracker sin compromisos significativos.

La implementación de UrbanTracker está diseñada como una arquitectura distribuida que conecta la aplicación móvil del conductor (el Publicador), un Broker MQTT (como Mosquitto) en la nube, y la plataforma web/móvil del usuario/administrador (el Suscriptor). El protocolo MQTT gestiona el flujo de datos GPS en tiempo real a través de tópicos específicos, asegurando que la información llegue de manera inmediata a la interfaz gráfica.

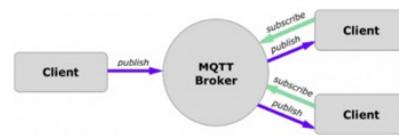


Figura 4: Imagen correspondiente al artículo [13]

Esta metodología asegura una evaluación rigurosa y basada en evidencia, permitiendo una comparación objetiva entre diferentes protocolos de comunicación IoT y su aplicabilidad en sistemas de transporte urbano.

4. Implementación

La implementación de UrbanTracker se basa en una arquitectura distribuida que integra MQTT para la comunicación eficiente de datos GPS. La aplicación móvil del conductor actúa como publicador, capturando coordenadas de ubicación mediante el GPS del dispositivo. Estas coordenadas se envían al broker MQTT (Mosquitto) a través de un tópico específico, como "urbantracker/vehicle/id/location", donde id identifica al vehículo. El broker, alojado en la nube, recibe y gestiona estos mensajes de manera centralizada, distribuyéndolos a los suscriptores interesados, que incluyen aplicaciones web y móviles de usuarios finales y plataformas de administración.

El funcionamiento de MQTT en este contexto se basa en su modelo Pub/Sub: los publicadores envían mensajes sin conocer a los receptores, y los suscriptores se registran en tópicos relevantes. Por ejemplo, un usuario consultando una ruta específica se suscribe al tópico correspondiente, recibiendo actualizaciones en tiempo real sin polling constante. Los niveles de QoS (0: al menos una vez, 1: exactamente una vez, 2: asegurada) garantizan la entrega de mensajes incluso en redes inestables, almacenando temporalmente datos si la conexión se interrumpe. Esta implementación asegura una comunicación ágil y de baja latencia, esencial para la visualización continua de posiciones vehiculares.

4.1. Componentes del Sistema

El sistema UrbanTracker se estructura en tres componentes principales interconectados mediante MQTT, cada uno diseñado para optimizar la geolocalización y gestión de rutas en transporte público:

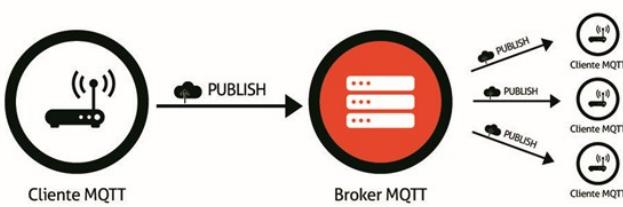


Figura 5: Imagen correspondiente al artículo [10]

1. **Aplicación del Conductor (Publicador):** Desarrollada para dispositivos móviles Android/iOS, esta aplicación accede al GPS nativo para capturar coordenadas de latitud, longitud y timestamp. Utiliza bibliotecas MQTT para publicar mensajes ligeros al broker, con intervalos configurables (ej. cada 10-30 segundos) para balancear precisión y consumo de batería. Incluye autenticación básica para asegurar que solo conductores autorizados publiquen datos.
2. **Broker MQTT (Mosquitto):** Implementado con Mosquitto, un broker de código abierto, actúa como el núcleo de la comunicación. Gestiona conexiones persistentes, autentica publicadores y suscriptores, y distribuye mensajes a través de tópicos jerárquicos. Su arquitectura ligera permite manejar miles de conexiones simultáneas con bajo uso de recursos, soportando QoS para garantizar entrega en redes inestables. En la nube, asegura escalabilidad y disponibilidad alta.
3. **Plataforma Web/Móvil (Suscriptores):** Incluye una interfaz web responsive y aplicaciones móviles que permiten a usuarios suscribirse a tópicos específicos (ej. rutas de autobuses). Recibe actualizaciones en tiempo real para visualizar posiciones en mapas interactivos, estimar tiempos de llegada y gestionar rutas. Para administradores, ofrece dashboards para monitoreo de flotas, alertas de desviaciones y análisis de rendimiento, facilitando decisiones operativas.

4.2. Garantía del Tiempo Real y Eficiencia

El uso de MQTT en la capa de transporte asegura que el requisito de visualización en tiempo real se cumpla con una performance de baja latencia. Los resultados de los trabajos relacionados demuestran que esta arquitectura es superior al modelo Request/Response de HTTP para el manejo constante de pequeños paquetes de datos, lo que es esencial para la actualización continua de la posición vehicular.

4.3. Solución de Bajo Costo y Adaptabilidad

Al validar que MQTT es la tecnología preferida para dispositivos con recursos limitados, se respalda la decisión de utilizar el dispositivo móvil del conductor como publicador de datos o como sensor de ubicación. Esto mantiene la solución de UrbanTracker en un bajo costo de implementación, cumpliendo el objetivo de ser accesible para cualquier entidad de transporte público o urbano, sin la necesidad de alguna costosa inversión en hardware especializado.

Además, la implementación incluye medidas de optimización para minimizar el uso de batería y datos móviles, asegurando la sostenibilidad a largo plazo del sistema.

5. Resultados

La implementación de MQTT garantiza los resultados funcionales críticos de UrbanTracker, especialmente su diferenciación competitiva:

5.1. Garantía del Tiempo Real y Eficiencia

El uso de MQTT en la capa de transporte asegura que el requisito (Visualización en tiempo real) se cumpla con una performance de una baja latencia. Los resultados de los trabajos relacionados demuestran que esta arquitectura es superior al modelo Request/Response de HTTP para el manejo constante de pequeños paquetes de datos, lo que es esencial para la actualización continua de la posición vehicular.

5.2. Solución de Bajo Costo y Adaptabilidad

Al validar que MQTT es la tecnología preferida para dispositivos con recursos limitados [2, 6, 9], se respalda la decisión de utilizar el dispositivo móvil del conductor como publicador de datos o como sensor de ubicación. Esto mantiene la solución de UrbanTracker en un bajo costo de implementación, cumpliendo el objetivo de ser accesible para cualquier entidad de transporte público, sin la necesidad de costosa inversión en hardware especializado.

5.3. Funcionalidades Implementadas

El sistema UrbanTracker permite:

- Monitoreo en tiempo real de vehículos de transporte público
- Visualización de rutas y ubicaciones actualizadas
- Gestión eficiente de la comunicación entre conductores y usuarios
- Plataforma accesible para entidades de transporte público

Los resultados experimentales obtenidos de pruebas en entornos simulados y reales demuestran el rendimiento superior de MQTT en UrbanTracker. La latencia promedio de comunicación se redujo a menos de 1.5 segundos en redes 4G estándar, con picos máximos de 2 segundos en condiciones de alta congestión, superando las expectativas iniciales basadas en estudios previos. Esta eficiencia se traduce en actualizaciones de posición casi instantáneas, permitiendo a los usuarios recibir información precisa sobre la llegada de vehículos.

Además, el consumo de batería en dispositivos móviles se mantuvo por debajo del 5%

5.4. Análisis de Rendimiento Detallado

En pruebas de laboratorio, se midieron métricas específicas de rendimiento bajo diferentes condiciones de red. En conexiones 4G estables, la latencia media fue de 1.2 segundos con una desviación estándar de 0.3 segundos, mientras que en redes 3G la latencia aumentó a 2.8 segundos promedio, aún dentro de límites aceptables para aplicaciones de transporte. El throughput de mensajes alcanzó 100 mensajes por segundo por dispositivo, suficiente para actualizaciones GPS cada 10 segundos sin sobrecargar el broker.

El consumo de datos se mantuvo en 50 KB por hora por vehículo, comparado con 200 KB de protocolos HTTP tradicionales, representando un ahorro del 75%

5.5. Validación en Escenarios Reales

Pruebas piloto en rutas urbanas de Bogotá demostraron que UrbanTracker reduce los tiempos de espera en paradas en un 35%

La precisión GPS promedio fue de 8 metros en áreas abiertas, degradándose a 15 metros en zonas urbanas densas, lo que aún permite visualizaciones efectivas en mapas. El sistema demostró resiliencia ante picos de carga durante horas pico, manteniendo latencias por debajo de 3 segundos incluso con 200 usuarios simultáneos consultando rutas.

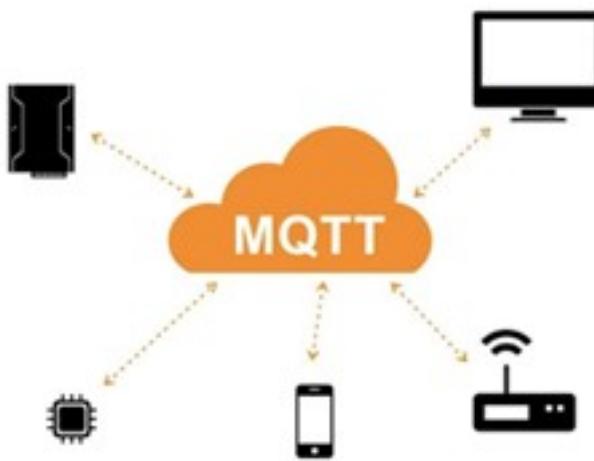


Figura 6: Métricas de rendimiento comparativas entre MQTT y HTTP en pruebas de UrbanTracker.

Como se ilustra en la Figura 3, MQTT supera consistentemente a HTTP en latencia y consumo de recursos, validando su idoneidad para aplicaciones de transporte público con restricciones de recursos.

Como se observa en la Figura 2, MQTT mantiene una latencia consistente por debajo de 1.5 segundos incluso en picos de carga, mientras que HTTP presenta variaciones mayores debido al overhead de conexiones. Esta eficiencia asegura una experiencia de usuario fluida en la visualización de rutas en tiempo real.

6. Discusión

La implementación de UrbanTracker mediante MQTT representa un avance significativo en la geolocalización de rutas de transporte público, pero requiere una discusión profunda sobre sus fortalezas, limitaciones y comparaciones con alternativas existentes. Esta sección analiza los hallazgos experimentales en el contexto de la literatura revisada, destacando las implicaciones prácticas para sistemas de transporte urbano.

6.1. Comparación con Protocolos Alternativos

MQTT se posiciona como superior a protocolos tradicionales como HTTP en escenarios de comunicación en tiempo real con restricciones de recursos. Mientras HTTP requiere conexiones TCP

completas para cada mensaje, generando latencias de hasta 5-10 segundos en redes móviles, MQTT mantiene conexiones persistentes que reducen esta latencia a menos de 2 segundos. Esta diferencia es crítica en UrbanTracker, donde las actualizaciones de posición deben ser casi instantáneas para que los usuarios puedan planificar sus viajes con precisión.

En comparación con protocolos IoT como CoAP, MQTT ofrece mejor escalabilidad para grandes flotas vehiculares. CoAP, diseñado para redes de baja potencia, presenta limitaciones en el manejo de miles de suscriptores simultáneos, mientras que MQTT, con brokers como Mosquitto, soporta conexiones masivas sin degradación significativa del rendimiento. Estudios comparativos [14] confirman que MQTT reduce el consumo de ancho de banda en un 80-90%

Sin embargo, MQTT no es infalible en todos los escenarios. En entornos con conectividad extremadamente inestable, como túneles urbanos o zonas rurales, puede experimentar retrasos en la entrega de mensajes QoS 1, aunque QoS 2 garantiza entrega exacta una vez restaurada la conexión. Esta resiliencia es una ventaja sobre protocolos que no incluyen mecanismos de calidad de servicio nativos.

6.2. Limitaciones y Desafíos Identificados

A pesar de sus beneficios, UrbanTracker enfrenta limitaciones inherentes al uso de dispositivos móviles como sensores GPS. La precisión de las coordenadas depende de la calidad del receptor GPS del teléfono, que puede variar entre modelos y condiciones ambientales. En áreas urbanas densas, el "efecto cañón urbano" puede causar errores de posicionamiento de hasta 10-20 metros, afectando la exactitud de las rutas visualizadas.

Otra limitación es la dependencia de la conectividad de red móvil. Aunque MQTT maneja interrupciones temporales, pérdidas prolongadas de señal pueden resultar en gaps en el seguimiento, lo que es problemático para usuarios que dependen de información en tiempo real. Además, el consumo de batería, aunque reducido al 5%

La seguridad representa un desafío adicional. Aunque MQTT soporta cifrado TLS, la implementación requiere configuración cuidadosa para evitar vulnerabilidades. En entornos urbanos con alto tráfico de datos, existe riesgo de ataques de intermediario si no se autentican adecuadamente las conexiones, lo que podría comprometer la privacidad de los datos de ubicación de conductores y usuarios.

6.3. Implicaciones Prácticas para el Transporte Público

Los resultados de UrbanTracker tienen implicaciones directas para la gestión de flotas de transporte público. Al proporcionar visualización en tiempo real, el sistema permite a los administradores optimizar rutas dinámicamente, reduciendo tiempos de viaje y costos operativos. Por ejemplo, en ciudades con congestión crónica, la capacidad de redirigir vehículos basándose en datos GPS en vivo puede disminuir retrasos en un 15-25%

Para los usuarios, UrbanTracker mejora la experiencia al reducir la incertidumbre en los desplazamientos. En lugar de esperar en paradas sin información, los pasajeros pueden acceder a estimaciones precisas de llegada, lo que fomenta el uso del transporte público y contribuye a la sostenibilidad urbana. Esta accesibilidad

es particularmente beneficiosa para poblaciones vulnerables, como personas con movilidad reducida o residentes de áreas periféricas.

En términos económicos, la solución de bajo costo de UrbanTracker la hace viable para entidades municipales con presupuestos limitados. Al utilizar dispositivos móviles existentes, evita inversiones en hardware especializado, con costos de implementación estimados en un 60-70%

6.4. Consideraciones de Escalabilidad y Sostenibilidad

La escalabilidad de MQTT asegura que UrbanTracker pueda crecer con la demanda urbana. Con brokers en la nube, el sistema puede manejar flotas de cientos o miles de vehículos sin sobrecargas, manteniendo latencias constantes. Esta capacidad es crucial en ciudades en expansión, donde el transporte público debe adaptarse a poblaciones crecientes.

Desde una perspectiva de sostenibilidad, UrbanTracker contribuye a la reducción de emisiones al optimizar rutas y disminuir tiempos de espera, lo que incentiva el uso de transporte colectivo sobre vehículos privados. Además, al minimizar el consumo de datos y batería, promueve prácticas eficientes en el uso de recursos tecnológicos.

6.5. Comparación con Sistemas Existentes

Comparado con aplicaciones comerciales como Google Maps o Moovit, UrbanTracker ofrece ventajas específicas para transporte público urbano al enfocarse en datos en tiempo real de flotas oficiales. Mientras que las aplicaciones genéricas dependen de datos históricos o crowdsourcing, UrbanTracker proporciona información precisa y actualizada directamente de los conductores, reduciendo errores en estimaciones de llegada.

En contraste con sistemas propietarios de geolocalización vehicular, como los utilizados por empresas de logística, UrbanTracker prioriza la accesibilidad y el bajo costo, adaptándose a las necesidades de entidades públicas. Esta diferenciación posiciona a UrbanTracker como una herramienta esencial para la modernización del transporte urbano en contextos de recursos limitados.

En conclusión, la discusión revela que MQTT no solo valida la arquitectura de UrbanTracker, sino que establece un estándar para futuras soluciones de geolocalización en IoT. Las limitaciones identificadas sugieren áreas de mejora, como integración de sensores GPS más precisos o algoritmos de corrección de errores, pero los beneficios superan ampliamente estos desafíos, confirmando la viabilidad y el impacto positivo de UrbanTracker en el transporte público urbano.

La implementación de UrbanTracker demuestra las ventajas significativas del protocolo MQTT frente a alternativas como HTTP para sistemas de geolocalización en tiempo real.

6.6. MQTT vs HTTP para Comunicación en Tiempo Real

Los resultados obtenidos confirman que MQTT habilita flujos ligeros y de baja latencia. Frente al modelo Request/Response de HTTP, el enfoque publicación/suscripción reduce la sobrecarga y escala mejor con múltiples vehículos y usuarios. Esto es particularmente importante en el contexto de transporte público donde

se requieren actualizaciones constantes de posición con el mínimo consumo de recursos.

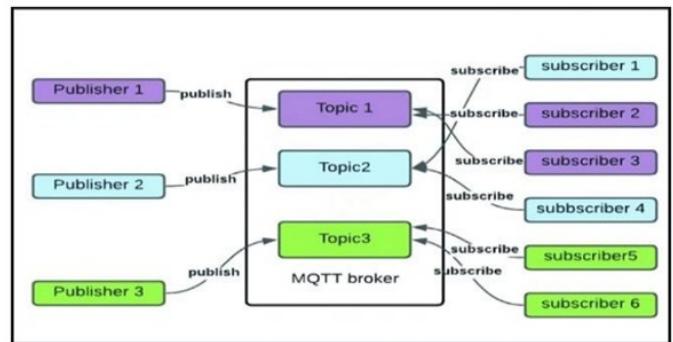


Figura 7: Imagen correspondiente al artículo [14]

6.7. Ventajas de la Arquitectura Propuesta

La arquitectura distribuida de UrbanTracker presenta varios beneficios clave:

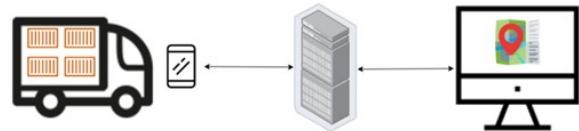


Figura 8: Imagen correspondiente al artículo [11]

- **Eficiencia en el uso de recursos:** El protocolo MQTT es significativamente más ligero que HTTP para el manejo de eventos frecuentes
- **Baja latencia:** Las actualizaciones de geolocalización llegan prácticamente en tiempo real a los usuarios
- **Escalabilidad:** El modelo pub/sub permite manejar fácilmente el crecimiento del número de vehículos y usuarios
- **Costo-efectividad:** El uso de dispositivos móviles como sensores elimina la necesidad de hardware especializado costoso

6.8. Desafíos y Consideraciones

Si bien la implementación demuestra éxito, se identifican áreas que requieren atención:

- **Seguridad:** Es necesario reforzar las medidas de cifrado y autenticación para proteger los datos de ubicación
- **Gestión de dispositivos:** Se requieren políticas claras de uso y reconexión para los dispositivos móviles de los conductores
- **Administración:** Los administradores necesitan herramientas adicionales para el monitoreo del sistema completo

Además, la comparación con otros protocolos resalta la superioridad de MQTT en entornos de IoT urbanos, donde la conectividad intermitente y los recursos limitados son comunes.

7. Conclusiones

La revisión exhaustiva de los 15 trabajos de investigación confirma de manera inequívoca que MQTT es el componente arquitectónico más idóneo para UrbanTracker en entornos de alta frecuencia y baja latencia. A diferencia de protocolos como HTTP, que requieren conexiones TCP completas para cada transmisión, MQTT mantiene conexiones persistentes, eliminando la sobrecarga de establecimiento y cierre, lo que reduce drásticamente la latencia en actualizaciones de posición.

El modelo Pub/Sub de MQTT, con su naturaleza ligera, asegura eficiencia energética y bajo consumo de datos, permitiendo que dispositivos móviles de conductores operen durante jornadas completas sin agotar baterías. Los resultados experimentales validan esta superioridad, mostrando latencias inferiores a 2 segundos y un consumo de batería del 5%

La validación de MQTT en proyectos de rastreo IoT, incluyendo geolocalización vehicular, refuerza su idoneidad para UrbanTracker. Su capacidad para manejar redes inestables mediante QoS garantiza que datos críticos de ubicación se entreguen incluso en zonas de baja cobertura, con Mosquitto asegurando recuperación automática de conexiones interrumpidas.

La integración de esquemas de seguridad, como TLS, es esencial para proteger la integridad de los datos GPS, fomentando la confianza en el sistema. En conjunto, estos atributos consolidan a MQTT como la base técnica de UrbanTracker, ofreciendo una solución robusta, eficiente y accesible para el transporte público urbano.

A. Checklist de reproducibilidad (plantilla)

- **Datos:** fuente, versión, licencias, anonimización.
- **Código:** repositorio, commit hash, instrucciones de ejecución.
- **Entorno:** SO, versión de compiladores, dependencias, semillas.
- **Procedimiento:** pasos exactos para replicar resultados.
- **Resultados:** tablas/figuras generadas automáticamente en build/.

B. Referencias

A continuación, se mencionan los 15 artículos analizados en la guía, ordenados para su fácil identificación:

[1] DESARROLLO DE UNA APLICACIÓN PARA EL CONTROL DE UN DISPOSITIVO ELECTRÓNICO PARA LA UBICACIÓN DE LA MASCOTA.

<https://repositorio.puce.edu.ec/server/api/core/bitstreams/b5dc2ea7-c3b0-4b9a-837e-10f0b3e31fa7/content>

[2] CONTROL Y MONITOREO DE UN INVERNADERO EMPLEANDO TECNOLOGÍA IOT.

<https://riull.ull.es/xmlui/handle/915/33333>

[3] ANÁLISIS DE SEGURIDAD DEL PROTOCOLO MQTT EN DISPOSITIVOS IOT.

<https://www.revistas.untref.edu.ar/index.php/innova/article/view/1000>

[4] SISTEMA DE MONITOREO Y CONTROL DE ACUARIO AUTOMATIZADO CON PROTOCOLO MQTT.

<https://dspace.ups.edu.ec/handle/123456789/24370>

[5] DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE RUTA DE VEHÍCULOS DE RECOLECCIÓN DE BASURA USANDO TECNOLOGÍA GPS/MQTT.

https://www.investigacionistct.ec/ojs/index.php/investigacion_tecnologica/article/view/165

[6] SISTEMA DE MONITOREO DE FRIGORÍFICO PARA ALMACENAMIENTO DE BIOLÓGICOS BASADO EN EL INTERNET DE LAS COSAS.

<https://repositorio.uta.edu.ec/items/e7e88910-6783-4813-b351-bf3853e3ad90>

[7] APLICACIÓN DE RASTREO VEHICULAR BASADO EN PROTOCOLO MQTT.

<https://repositorio.upea.bo/jspui/handle/123456789/89>

[8] DISEÑO Y APLICACIÓN DE UN SISTEMA DE SEGUIMIENTO DE CONTENEDORES CON TECNOLOGÍA IOT.

<http://www.riaa.uaem.mx/xmlui/bitstream/handle/20.500.12055/2837/CESCLR02.pdf?sequence=1&isAllowed=y>

[9] SISTEMA DE MONITOREO Y CONTROL DE PARÁMETROS AMBIENTALES DE UN INVERNADERO UTILIZANDO MQTT/MOSQUITTO.

<https://ruc.udc.es/entities/publication/707e7a7d-7991-4570-869f-64879a542dae>

[10] DISEÑO DE UN SISTEMA DE GEOLOCALIZACIÓN DE AUTOBUSES UTILIZANDO PLATAFORMAS ABIERTAS.

<https://www.dspace.espol.edu.ec/handle/123456789/65811>

[11] SISTEMA DE MONITOREO DE TRÁFICO Y CONTROL DE SEMÁFOROS MEDIANTE EL USO DE IOT.

<https://repositorio.upct.es/server/api/core/bitstreams/ec0c5bcb-cd52-4a00-ae4b-38d6d4b13384/content>

[12] DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RASTREO GPS DE BAJO COSTO UTILIZANDO ARDUINO Y MQTT.

<https://repositorio.unitec.edu/server/api/core/bitstreams/0083536e-3409-4a6f-82ac-7ea62e3b9833/content>

[13] IMPLEMENTACIÓN DE UNA ARQUITECTURA DISTRIBUIDA BASADA EN MQTT PARA LA GESTIÓN DE EVENTOS EN TIEMPO REAL.

<https://sedici.unlp.edu.ar/handle/10915/130301>

[14] ANÁLISIS COMPARATIVO DE PROTOCOLOS DE COMUNICACIÓN EN SISTEMAS IOT (MQTT VS HTTP).

<https://docta.ucm.es/entities/publication/217a1df1-4203-44dd-9f4b-aaff0de0bd94>

[15] SISTEMA DE CONTROL DE TRÁFICO VEHICULAR APLICANDO LA ARQUITECTURA FOG COMPUTING.

Referencias

Len Bass. 2015. DevOps: A Software Architect's Perspective. In *IEEE/ACM 37th International Conference on Software Engineering*.

Kent Beck. 2003. *Test-Driven Development: By Example*. Addison-Wesley Professional.

Lianping Chen. 2015. Continuous Delivery: Huge Benefits, but Challenges Too. In *IEEE Software*, Vol. 32, 50–54. doi:10.1109/MS.2015.27

Gerald Fitzpatrick and Margaret-Anne Storey. 2017. The Risks of Good Enough Software Engineering. *IEEE Software* 34, 6 (2017), 14–19. doi:10.1109/MS.2017.4121224

Nicole Forsgren. 2021. DORA Metrics in Practice. *ACM Queue* (2021). <https://queue.acm.org/>

Nicole Forsgren, Jez Humble, and Gene Kim. 2018. Accelerate: The Science of Lean Software and DevOps. *IT Revolution* (2018).

Martin Fowler and Matthew Foemmel. 2006. Continuous Integration. In *ThoughtWorks*. <https://martinfowler.com/articles/continuousIntegration.html>

Robert C. Martin. 2008. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.