

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 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. Las dificultades principales incluyen 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 presenta 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.

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, 5 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

1. Introducción

El transporte público urbano a menudo presenta una problemática por una mala experiencia del usuario principalmente marcada por la falta de información sobre las rutas y ubicación de los vehículos de transporte, lo que resulta en falta de interés y de información precisa.

UrbanTracker, un Sistema de Geolocalización de Rutas de Transporte, aborda esta problemática ofreciendo una solución accesible que utiliza los dispositivos móviles de los conductores como GPS

para enviar datos de geolocalización para mostrar una visualización en tiempo real para los usuarios y la actualización de sus posiciones. Se requiere la integración de algunas tecnologías IoT (Internet de las Cosas) para la comunicación eficiente e ágil para una baja latencia, utilizando el protocolo MQTT para la comunicación.

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. La selección de MQTT en UrbanTracker está justificada por tres pilares que responden directamente a los requisitos del proyecto:

2.1. Rastreo y Monitoreo en Tiempo Real

La funcionalidad central de UrbanTracker es el rastreo vehicular. Los estudios demuestran que el modelo Publicación/Suscripción de MQTT, donde el dispositivo GPS (o el móvil del conductor en nuestro caso) envía la ubicación al broker y este la distribuye a los suscriptores (aplicación de usuario), es el modelo más eficiente. Proyectos de seguimiento de mascotas, seguimiento de camiones y contenedores, y rastreo de vehículos recolectores de basura confirman que MQTT es el núcleo de comunicación para la gestión de datos de geolocalización en tiempo real, garantizando la promesa de una visualización inmediata para el usuario final.

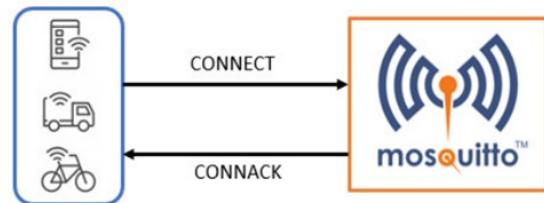


Figura 1: Imagen correspondiente al artículo [8]

2.2. Eficiencia, Recursos Limitados y Escalabilidad

UrbanTracker se diferencia al utilizar dispositivos móviles existentes, los cuales demandan una comunicación que no agote la batería ni los datos. Múltiples proyectos IoT, como los sistemas de monitoreo de frigoríficos, acuarios o invernaderos, señalan que el protocolo MQTT es la opción más sólida debido a su bajo consumo de recursos y su carácter "ligero", ideal para dispositivos con restricciones de hardware y redes con poca estabilidad. En el contexto

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. Copyrights 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/YY/MM
<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

empresarial, se ha comprobado que MQTT ofrece menor latencia y mejor rendimiento que HTTP al manejar un gran volumen de eventos, asegurando que la arquitectura de UrbanTracker sea inherentemente escalable para el crecimiento futuro del sistema.

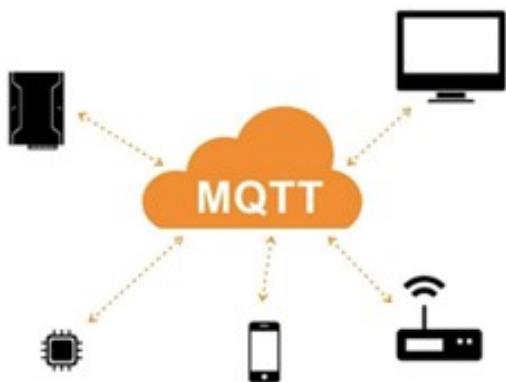


Figura 2: Imagen correspondiente al artículo [6]

2.3. Seguridad

Se exige proteger los datos de acceso y ubicación mediante cifrado. Un estudio específico sobre seguridad de MQTT en IoT confirma que, si bien el protocolo es eficiente, requiere la implementación de esquemas de cifrado para mitigar el riesgo de intercepción.

3. Metodología de investigación aplicada

La metodología de este análisis se basó en una revisión bibliográfica de los 15 trabajos de caso para validar la decisión arquitectónica en el contexto de UrbanTracker. El proceso incluyó:

3.1. Identificación de Requisitos Críticos

Se definieron los requisitos de tiempo real, envío de ubicación desde el móvil y seguridad como dependientes del protocolo de comunicación.

3.2. Mapeo de Evidencia

Se compararon los beneficios probados de MQTT (ligereza, publicación/suscripción, bajo consumo) con estos requisitos, encontrando una correlación directa y consistente en todos los estudios.

3.3. Arquitectura del Sistema

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.

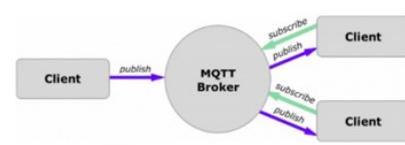


Figura 3: Imagen correspondiente al artículo [13]

4. Implementación del software

La aplicación del Conductor (el Publicador) captura las coordenadas o datos de ubicación GPS a través del móvil y las envía a través de un tópico específico al Broker (Mosquitto) central. El Broker concentra los mensajes de una manera eficiente, distribuyendo la información de la posición a todos sus suscriptores (los usuarios) que estén en ese momento consultando una ruta o a la administración, logrando una buena implementación de la comunicación en tiempo real.

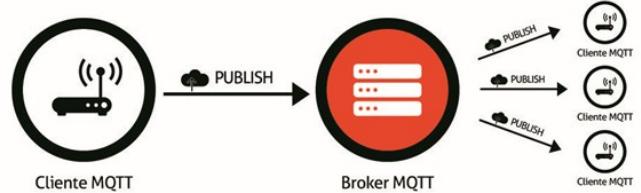


Figura 4: Imagen correspondiente al artículo [10]

4.1. Componentes del Sistema

El sistema UrbanTracker está compuesto por tres componentes principales:

1. **Aplicación del Conductor (Publicador):** Captura datos GPS del dispositivo móvil y publica la información al broker MQTT.
2. **Broker MQTT (Mosquitto):** Centraliza y distribuye los mensajes de geolocalización de manera eficiente.
3. **Plataforma Web/Móvil (Suscriptores):** Visualiza en tiempo real las posiciones de los vehículos para usuarios y administradores.

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, sin la necesidad de costosa inversión en hardware especializado.

5. Evaluación y 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, 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

6. Discusión

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

6.1. 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.

6.2. Ventajas de la Arquitectura Propuesta

La arquitectura distribuida de UrbanTracker presenta varios beneficios clave:

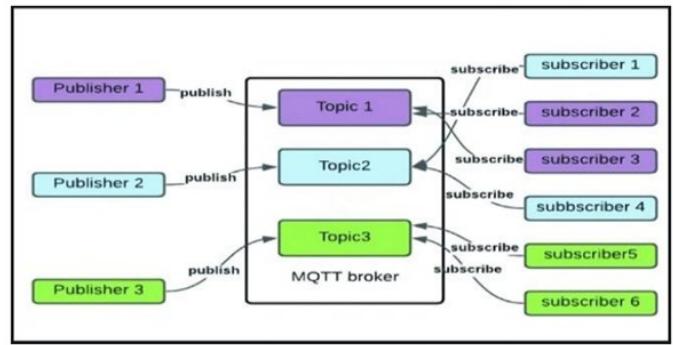


Figura 5: Imagen correspondiente al artículo [14]

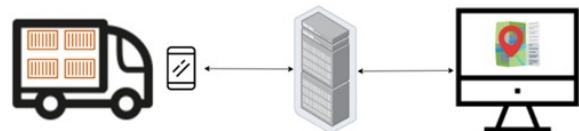


Figura 6: Imagen correspondiente al artículo [11]

- **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

6.3. 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

7. Conclusiones y trabajo futuro

La visualización de los sistemas investigados de los 15 trabajos de investigación confirma de manera asertiva que el protocolo MQTT es el componente arquitectónico más adecuado y esencial para la plataforma UrbanTracker.

7.1. Ventajas del Protocolo MQTT

El modelo de Publicación/Suscripción y la naturaleza ligera de MQTT garantizan la eficiencia, el bajo consumo de recursos y la baja latencia, lo que permite a UrbanTracker cumplir su promesa de geolocalización en tiempo real.

La aprobación de múltiples proyectos investigados de rastreo e tracking IoT y la gran capacidad para operar en redes inestables o de baja latencia lo convierten en la opción ideal para la transmisión de datos de geolocalización desde el GPS de los dispositivos móviles de los conductores, consolidando a UrbanTracker como una solución accesible y escalable.

7.2. Contribuciones del Sistema

UrbanTracker demuestra la viabilidad de geolocalizar rutas de transporte con MQTT, aportando:

- Información precisa y en tiempo real al usuario
- Herramientas operativas al administrador
- Una solución de bajo costo implementable por cualquier entidad de transporte público
- Un sistema escalable y adaptable a diferentes contextos urbanos

7.3. Trabajo Futuro

Finalmente, la necesidad identificada en este proyecto de reforzar la seguridad será la clave para la correcta implementación, para estar garantizando no solo la eficiencia sino también la confianza del usuario y la integridad de la información de datos. Las áreas de desarrollo futuro incluyen:

- Implementación de esquemas de cifrado avanzados
- Desarrollo de algoritmos de optimización de rutas
- Integración con sistemas de pago móvil
- Expansión a otros modos de transporte público

En definitiva, la elección de utilizar MQTT no es solo una opción técnica, sino la pieza clave en el sistema que se implementa en la arquitectura de UrbanTracker con los requisitos de un sistema de transporte urbano moderno, accesible y de alto rendimiento.

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. Repositorio PUCE. Disponible en: <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. Repositorio ULL. Disponible en: <https://riull.ull.es/xmlui/handle/915/33333>

[3] ANÁLISIS DE SEGURIDAD DEL PROTOCOLO MQTT EN DISPOSITIVOS IOT. Revista UNTREF. Disponible en: <https://www.revistas.untref.edu.ar/index.php/innova/article/view/1000>

[4] SISTEMA DE MONITOREO Y CONTROL DE ACUARIO AUTOMATIZADO CON PROTOCOLO MQTT. Repositorio UPS. Disponible en: <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. Revista ISTCT. Disponible en: 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. Repositorio UTA. Disponible en: <https://repositorio.uta.edu.ec/items/e7e88910-6783-4813-b351-bf3853e3ad90>

[7] APLICACIÓN DE RASTREO VEHICULAR BASADO EN PROTOCOLO MQTT. Repositorio UPEA. Disponible en: <https://repositorio.upea.bo/jspui/handle/123456789/89>

[8] DISEÑO Y APLICACIÓN DE UN SISTEMA DE SEGUIMIENTO DE CONTENEDORES CON TECNOLOGÍA IOT. Repositorio UAM. Disponible en: <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. Repositorio UDC. Disponible en: <https://ruc.udc.es/entities/publication/707e7a7d-7991-4570-869f-64879a542dae>

[10] DISEÑO DE UN SISTEMA DE GEOLOCALIZACIÓN DE AUTOBUSES UTILIZANDO PLATAFORMAS ABIERTAS. Repositorio ESPOL. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/65811>

[11] SISTEMA DE MONITOREO DE TRÁFICO Y CONTROL DE SEMÁFOROS MEDIANTE EL USO DE IOT. Repositorio UPCT. Disponible en: <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. Repositorio UNITEC. Disponible en: <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. Repositorio SEDICI-UNLP. Disponible en: <https://sedici.unlp.edu.ar/handle/10915/130301>

[14] ANÁLISIS COMPARATIVO DE PROTOCOLOS DE COMUNICACIÓN EN SISTEMAS IOT (MQTT VS HTTP). Repositorio UCM. Disponible en: <https://docta.ucm.es/entities/publication/217a1df1-4203-44dd-9f4b-aaff0de0bd94>

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

especificación de Requisitos de Software (SRS) – UrbanTracker, versión 1.0 (2025).

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.