

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

Objetivo: Optimizar la movilidad urbana mediante geolocalización en tiempo real de vehículos de transporte, aprovechando IoT con protocolo MQTT para la transmisión de datos. Métodos: Se diseñó UrbanTracker, compuesto por una app móvil para conductores (GPS), un Backend con bróker MQTT y clientes web para visualizar rutas y buses. Resultados: El sistema permite monitoreo en vivo con baja latencia, gestión de rutas, conductores y vehículos.

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

1 Introducción

El transporte urbano Tiene una problemática sobre la desinformación sobre la llegada de autobuses y sus rutas. Las tecnologías IoT y protocolos ligeros como MQTT permiten construir soluciones de rastreo vehicular mediante un gps, con visualización en mapas para usuarios y administradores.

MARCO DE REFERENCIA – Geolocalización en transporte urbano: Según «Aplicación web para el control de desviaciones de rutas en el transporte público mediante IOT» (ESPOL, 2025), combinar GPS en buses y una plataforma web permite visualizar en tiempo real ubicación y trayectos de las unidades.

2 Marco teórico y trabajos relacionados

Aportes clave seleccionados del thesaurus y su relación con Urban-Tracker:

- DESARROLLO DE UN SISTEMA DE LOCALIZACIÓN BASADO EN GPS E IOT: UN ESTUDIO DE CASO EN QUITO (2025). Fuente: Google Sschoolar. https://www.investigacionistct.ec/ojs/index.php/investigacion_tecnologica/article/view/165
- SISTEMA DE GEOLOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APPLICANDO INTERNET DE LAS COSAS (2025). Fuente: Google Sschoolar. <https://repositorio.upea.bo/jspui/handle/123456789/89>

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/nnnnnnn.nnnnnnn>

• Aplicación web para el control de desviaciones de rutas en el transporte público mediante IOT (2025). Fuente: Google Sschoolar. <https://www.dspace.espol.edu.ec/handle/123456789/65811>

• Aplicación del protocolo MQTT y recolección de datos para aplicaciones IoT (2025). Fuente: Google Sschoolar. <https://repositorio.unitec.edu/it4bb8-4fd7-887a-d93f122933c5>

• Arquitectura orientada a eventos sobre protocolo MQTT (2025). Fuente: Google Sschoolar. <https://sedici.unlp.edu.ar/handle/10915/130301>

En conjunto, estos trabajos recomiendan una arquitectura de tres capas con GPS en el vehículo, un pub/sub –preferentemente MQTT– y cliente web para visualización. UrbanTracker adopta este enfoque y lo aplica a un contexto local.

MARCO DE REFERENCIA – MQTT: Según «Arquitectura orientada a eventos sobre protocolo MQTT» (SEDICI–UNLP, 2025), el paradigma publicación/suscripción de MQTT reduce latencia y uso de ancho de banda en telemetría vehicular.

3 Metodología de investigación aplicada

Arquitectura: (1) Adquisición de datos: la app del conductor obtiene coordenadas con el GPS del dispositivo móvil y publica JSON al bróker; (2) Backend: bróker MQTT (Mosquitto), almacenamiento y API de apoyo; (3) Presentación: apps web con mapas interactivos.

Comunicación: tópicos del tipo /ruta/vehículo que permiten separar productores y consumidores. Para la web se habilita MQTT sobre WebSocket.

Seguridad: autenticación por roles, JWT.

MARCO DE REFERENCIA – Diseño IoT para rastreo vehicular: Según «DESARROLLO DE UN SISTEMA DE LOCALIZACIÓN BASADO EN GPS E IOT: UN ESTUDIO DE CASO EN QUITO» (ISTCT, 2025), un dispositivo con GPS y conectividad GSM/4G envía datos a la nube para visualización en tiempo real.

MARCO DE REFERENCIA – Uso de smartphone como nodo IoT: Según «SISTEMA DE GEOLOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APPLICANDO INTERNET DE LAS COSAS» (UPEA, 2025), es viable sustituir hardware dedicado por dispositivos móviles con GPS para reducir costos de implementación.

4 Implementación del software

Describimos arquitectura, decisiones tecnológicas y *pipelines*. Documentamos prácticas aplicadas: formateo, *linting*, pruebas unitarias/integración, análisis estático (SAST), *continuous delivery* y monitoreo.

4.1 Arquitectura del sistema

Arquitectura: (1) Adquisición de datos: la app del conductor obtiene coordenadas con el GPS del dispositivo móvil y publica JSON al

bróker; (2) Backend: bróker MQTT (Mosquitto), almacenamiento API de apoyo; (3) Presentación: apps web con mapas interactivos.

4.2 Fragmento de código

Ejemplo de implementación de una función con tipado estático:

```

1 def suma(a: int, b: int) -> int:
2     """Suma dos numeros enteros.
3
4     Args:
5         a: Primer operando
6         b: Segundo operando
7
8     Returns:
9         La suma de a y b
10    """
11
12    return a + b

```

4.3 Comparación de tecnologías

La selección de tecnologías se basó en criterios objetivos. La 1 presenta una comparación detallada de los frameworks evaluados.

Cuadro 1: Comparación de Frameworks de Desarrollo Web

Framework	Lenguaje	Performance	Puntuación
React	JavaScript	Alta	9.2
Angular	TypeScript	Alta	8.7
Vue.js	JavaScript	Alta	8.9
Django	Python	Media	8.5
Spring Boot	Java	Alta	8.8
Laravel	PHP	Media	8.1
Express.js	JavaScript	Alta	8.3

5 Evaluación y resultados

Se implementó:

- (a) consulta de rutas y trazado;
- (b) visualización en tiempo real de los vehículos activos;
- (c) inicio/fin de recorrido desde la app del conductor;
- (d) administración de rutas, conductores y vehículos desde panel del administrador web.

MARCO DE REFERENCIA — Gestión y monitoreo web/móvil: Según «Aplicación web para el control de desviaciones de rutas en el transporte mediante IOT» (ESPOL, 2025), la visualización cartográfica en tiempo real mejora la toma de decisiones operativas y la experiencia del usuario.

6 Discusión

MQTT habilitó flujos ligeros y de baja latencia. Frente a HTTP, el enfoque pub/sub reduce la sobrecarga y escala mejor con múltiples vehículos y usuarios.

MARCO DE REFERENCIA — MQTT vs alternativas: Según «Arquitectura orientada a eventos sobre protocolo MQTT» (SEDICI-UNLP, 2025), la arquitectura orientada a eventos con bróker ofrece ventajas de desempeño para actualización continua.

Usar dispositivos móviles como punto de conexión IoT disminuyendo el costo, pero se requiere políticas de uso y reconexión.

7 Conclusiones y trabajo futuro

UrbanTracker demuestra la viabilidad de geolocalizar rutas de transporte con MQTT, aportando información precisa al usuario y herramientas operativas al administrador.

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

DESARROLLO DE UN SISTEMA DE LOCALIZACIÓN BASADO EN GPS E IOT: UN ESTUDIO DE CASO EN QUITO (2025). Google Sschoolar. Disponible en: https://www.investigacionistct.ec/ojs/index.php/investigacion_tecnologica/article/view/165

SISTEMA DE GEOLOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS (2025).

Google Sschoolar. Disponible en: <https://repositorio.upea.bo/jspui/handle/123456789/65811>

Aplicación web para el control de desviaciones de rutas en el transporte publico mediante IOT (2025). Google Sschoolar. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/65811>

Aplicación del protocolo MQTT y recolección de datos para aplicaciones IoT (2025). Google Sschoolar. Disponible en: <https://repositorio.unitec.edu.4bb8-4fd7-887a-d93f122933c5>

Arquitectura orientada a eventos sobre protocolo MQTT (2025).

Google Sschoolar. Disponible en: <https://sedici.unlp.edu.ar/handle/10915/130301>

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

Aplicación del protocolo mqtt y recolección de datos para aplicaciones iot. (2025). [Artículo]. Repositorio UNITEC. <https://repositorio.unitec.edu.4bb8-4fd7-887a-d93f122933c5>

Aplicación web para el control de desviaciones de rutas en el transporte publico mediante iot. (2025). [Tesis]. Repositorio Digital ESPOL. <https://www.dspace.espol.edu.ec/handle/123456789/65811>

Arquitectura orientada a eventos sobre protocolo mqtt. (2025). [Tesis]. SEDICI, Universidad Nacional de La Plata. <https://sedici.unlp.edu.ar/handle/10915/130301>

Desarrollo de un sistema de localización basado en gps e iot: un estudio de caso en quito. (2025). [Artículo]. Investigación Tecnológica (ISTCT). https://www.investigacionistct.ec/ojs/index.php/investigacion_tecnologica/article/view/165

Sistema de geolocalización de vehículos recolectores de basura aplicando internet de las cosas. (2025). [Artículo]. Repositorio UPEA. <https://repositorio.upea.bo/jspui/handle/123456789/89>

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.