

UrbanTracker

Autor: Carlos Javier Rodriguez Manchola

Afiliación: SENA (Proyecto académico)

Correo del autor de correspondencia: cjrodriguez801@soy.sena.edu.co

Fecha: 2025

Resumen (Abstract)

Objetivo: Presentar el diseño e implementación de UrbanTracker, una plataforma web y móvil para el seguimiento en tiempo real de buses, construida sobre un backend en Spring Boot y un frontends web en Next.js (sobre React) y app móvil en React Native, con comunicación en tiempo real mediante WebSockets/MQTT.

Métodos: Se describe la arquitectura del sistema; los módulos de seguridad basados en JWT; la **captura y transmisión** de posiciones desde la app móvil; y el funcionamiento de las aplicaciones cliente (web y móvil). El marco teórico se sustenta en trabajos sobre rastreo vehicular, arquitectura de software y desarrollo móvil.

Resultados: El proyecto integra servicios REST seguros, modelos de datos para flota y rutas, y una app móvil que publica ubicaciones periódicas hacia el broker de mensajería.

Conclusiones: UrbanTracker demuestra la viabilidad de una solución modular y extensible para el transporte público, alineada con prácticas modernas de ingeniería de software y con potencial de optimización operativa.

Palabras clave: geolocalización; transporte público; Spring Boot; React Native; MQTT; Socket.IO; JWT.

Introducción

El rastreo y la visualización de flotas de transporte urbano han sido líneas de trabajo recurrentes en la investigación previa, con propuestas que van desde prototipos experimentales hasta despliegues operativos. Estudios sobre sistemas de información para transporte público y aplicaciones basadas en GPS evidencian mejoras en puntualidad, seguridad y experiencia del usuario (CITY-RUTA, 2018; STOPBUS, 2016). A la vez, enfoques centrados en arquitectura de software y patrones de diseño muestran que la modularidad y la separación de responsabilidades facilitan la evolución y la mantenibilidad de plataformas de monitoreo en tiempo real (Patrones de diseño en software, 2022; Elementos reusables para estrategias y patrones de privacidad, 2023).

En paralelo, la adopción de marcos y tecnologías para desarrollo web y móvil ha consolidado opciones híbridas como React Native y Flutter, ampliamente empleadas tanto en prototipado como en productos en producción. La evidencia reporta ventajas de portabilidad y reducción del tiempo al mercado, junto con retos en rendimiento, acceso a sensores y gestión de permisos según el sistema operativo (Desarrollo híbrido con Flutter, 2022; Módulo de recomendación de patrones de diseño para EGPAT, 2022). En el caso de UrbanTracker, la aplicación móvil se implementó con React Native.

Adicionalmente, las API RESTful y la documentación interactiva mediante OpenAPI han estandarizado la exposición de datos de vehículos, rutas y eventos, favoreciendo la interoperabilidad con paneles y aplicaciones de terceros (API REST con Spring, 2022; Sistemas de software híbrido, 2022).

Métodos

Diseño general: UrbanTracker utiliza una arquitectura cliente-servidor con Backend (Spring Boot, PostgreSQL), Web-Admin y Web-Client en Next.js (basado en React), y una app móvil en React Native. La comunicación en tiempo real se implementa con WebSockets/MQTT para difundir eventos de ubicación. Este diseño sigue lineamientos de arquitecturas por capas y de captura y envío de datos de movilidad descritos en trabajos previos (Adaptación de Scrum al desarrollo de software cuántico, 2025; Prácticas de gestión de proyectos de software, 2024).

Backend: El servicio utiliza JPA para la persistencia y expone recursos protegidos mediante JWT; se definen rutas públicas, privadas y un filtro que valida el token en cada solicitud. El esquema de datos modela entidades como vehículos, recorridos y puntos de ruta. Estas decisiones de seguridad y manejo de sesión se alinean con recomendaciones actuales sobre autenticación basada en tokens y buenas prácticas de arquitectura de datos (Reutilización de código y seguridad en aplicaciones escalables, 2023; Bases de datos relacionales y NoSQL en el desarrollo, 2021).

Cliente móvil: La app del conductor solicita permisos de ubicación, obtiene coordenadas con alta precisión y publica posiciones a intervalos configurables cuando el modo de seguimiento está activo. Este patrón se apoya en prácticas de muestreo, ahorro de batería y gestión de permisos documentadas para entornos móviles híbridos y soluciones de seguimiento vehicular (Desarrollo híbrido con Flutter, 2022; STOPBUS, 2016).

Resultados

Funcionalidad principal.

El proyecto integra un inicio de sesión con JWT para asegurar el acceso, un conjunto de endpoints REST para manejar la información (por ejemplo, vehículos y rutas) y una app móvil que pide permisos de ubicación y envía la posición de forma periódica. Además, el backend cuenta con documentación interactiva del API para facilitar pruebas y la integración con otros sistemas. En conjunto, esto da una base sólida y fácil de escalar (API REST con Spring, 2022; Sistemas de software híbrido, 2022).

Evidencia en el cliente móvil.

El flujo de seguimiento es directo: la app solicita los permisos, comienza a tomar latitud/longitud cada cierto tiempo y manda esos datos al servidor; si el usuario niega permisos o el GPS falla, la app avisa y pausa el envío hasta que todo esté bien. Este comportamiento sigue prácticas comunes en apps de localización para transporte (STOPBUS, 2016; Desarrollo híbrido con Flutter, 2022).

Evidencia en el backend y la web.

El backend ofrece endpoints para vehículos, rutas y recorridos, y aplica reglas de seguridad claras: hay rutas públicas (por ejemplo, la documentación) y rutas protegidas que exigen token. Esto respalda un esquema de autorización robusto para el panel administrativo y los clientes. Por su parte, la interfaz web para administración y para usuarios está construida con Next.js (sobre React), lo que facilita el enrutamiento, el renderizado eficiente y la integración con el API (Patrones de diseño en software, 2022; Reutilización de código y seguridad, 2023).

Discusión

Elección tecnológica.

Optamos por un stack “conocido y rendidor”: Spring Boot + PostgreSQL en el backend; Next.js (sobre React) para las interfaces web (admin y usuarios); React Native para la app móvil; WebSockets/MQTT para la comunicación en tiempo real; y JWT para la autenticación, con REST/OpenAPI para documentar el API. Esta combinación acelera el desarrollo (componentes reutilizables, rutas claras y pruebas sencillas) y facilita la integración con otros sistemas gracias a contratos de API bien definidos (API REST con Spring, 2022; Patrones de diseño en software, 2022; Sistemas de software híbrido, 2022).

Implicaciones y limitaciones.

En uso real, la señal GPS, el consumo de batería y la conectividad pueden afectar la precisión y la latencia. Por eso conviene ajustar el intervalo de muestreo, usar distanceFilter cuando aplique y preferir brokers MQTT cercanos para reducir tiempos. En privacidad y seguridad, los tokens JWT y la separación de rutas públicas/privadas sientan una buena base. (STOPBUS, 2016; Reutilización de código y seguridad, 2023).

Conclusiones

UrbanTracker ofrece una arquitectura actual para geolocalización de flotas: autenticación sin estado con JWT, APIs REST, clientes web y móvil, y mensajería en tiempo real. En conjunto, el sistema es modular y escalable. (CITY-RUTA, 2018; Adaptación de Scrum al desarrollo de software cuántico, 2025).

Conflictos de interés / Declaración ética

El autor declara no tener conflictos de interés. Este trabajo no involucra datos personales identificables de usuarios reales; se siguieron prácticas éticas de manejo de datos.

Contribuciones de los autores (CRediT)

Conceptualización, Metodología, Software, Validación, Redacción – C. J. Rodriguez Manchola

Referencias

- [R1] CITY-RUTA: Aplicación para el servicio de transporte público. 2018
<https://ridum.umanizales.edu.co/server/api/core/bitstreams/8fdf5333-60bc-44bf-b576-a60183fdde47/content>
- [R2] Java Struts Framework. 2008.
https://personal.tuxe.es/universidad/phd/arquitecturas_ebusiness.pdf.
- [R3] Aplicación móvil apoyada en georreferenciación para transporte público (STOPBUS). 2016. <https://core.ac.uk/download/pdf/352368119.pdf>.
- [R4] Dispositivo/estrategias de geolocalización y transmisión con GPS/Wi-Fi 2016.
http://scielo.sld.cu/scielo.php?pid=S1815-59282016000200002&script=sci_abstract.
- [R5] SisMo: sistema de seguridad para motocicletas 2016.
https://web.archive.org/web/20180415091800id_/http://revistas.usc.edu.co/index.php/Ingeniur/article/viewFile/651/518.
- [R6] Diseño de aplicación móvil para monitoreo de buses 2025.
<https://revistareg.com/index.php/1/article/view/90/265>.
- [R7] Adaptación de Scrum al desarrollo de software cuántico: revisión metodológica. 2025.
<https://ciencialatina.org/index.php/cienciala/article/view/17963/25824>.
- [R8] Crecimiento de la industria del software en Colombia: análisis sistémico. 2015.
<http://www.scielo.org.co/pdf/eia/n23/n23a09.pdf>.
- [R9] EasyRest: Generador automático de API REST basado en Spring Framework y motor de plantillas Beutl. 2022. https://d1wqtxts1xzle7.cloudfront.net/111860414/18841-libre.pdf?1708974048=&response-content-disposition=inline%3B+filename%3DEasyRest_Generador_automatico_de_API_Res.pdf&Expires=1756126855&Signature=PfW2C2QHi7Wp4BuhUwi0rQpJlJgRsm1r7NunpWEpVpdoZ~FzGJeMnHsyJFaKgwo~gic~m77kUOvu2tVwmrWcNjykmhOcg4KZlxPdeft0U0pPvQXoJ~CSl~F3NRVO6MJdJm5yVyC3HeHVwNjVkuAUab6ouhUiZvXNqlOT8Mt-EXJW~FpmHQzXixQdeRsUMHJsIKjarLwrHjTk9hsK-a3hfYXiW8JQp3DksHZKQq16kkHgw-fvVu7A-4pbbP6xEP0J14ytVkbzxQ01WA~IWJMIhRp4werkbMab0~kBVJqb3OR4sHCJWMHWSJBPO8n6ASbDmAfKdZgC9LZIAvz64Lg_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA.
- [R10] GotenJS Framework para generar estructura base de una app web. 2021.
https://sedici.unlp.edu.ar/bitstream/handle/10915/140261/Documento_completo.pdf-PDFA.pdf?sequence=1.
- [R11] Incidencia de patrones de diseño de software en el aprendizaje orientado a objetos. 2022. <https://www.investigarmqr.com/ojs/index.php/mqr/article/view/886/3451>

[R12] El papel de la programación orientada a objetos en el desarrollo de software sostenible y escalable. 2023. <https://ve.scielo.org/pdf/uct/v27n121/2542-3401-uct-27-121-85.pdf>.

https://sedici.unlp.edu.ar/bitstream/handle/10915/144325/Documento_completo.pdf-PDFA.pdf?sequence=1 [R13] Aspectos de ingeniería de software, bases de datos relacionales, no relacionales y como servicios en la nube para el desarrollo de sistemas de software híbridos 2022. .

[R14] Machine learning aplicado al análisis del rendimiento de desarrollos de software 2022. <https://www.redalyc.org/journal/6078/607870799010/607870799010.pdf>.

[R15] Prácticas empíricas en la gestión de proyectos de desarrollo de software a partir de análisis del principio de adaptabilidad 2024. <https://dspace.ups.edu.ec/bitstream/123456789/28055/1/UPS-CT011453.pdf>.

[R16] Arquitectura de software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra 2022. <https://www.redalyc.org/journal/5337/533774788007/533774788007.pdf>

[R17] Desarrollo híbrido con flutter 2022. <https://ciencialatina.org/index.php/cienciala/article/view/2959/4350>

[R18] Análisis comparativo de entornos de desarrollo móvil 2022. <https://ciencialatina.org/index.php/cienciala/article/view/2950>

[R19] Módulo de recomendación de patrones de diseño para EGPat 2021. <http://scielo.sld.cu/pdf/rcci/v15n2/2227-1899-rcci-15-02-118.pdf>

[R20] Elementos Reusables para Experimentar con Metodologías Basadas en Estrategias y Patrones de Privacidad 2023. <http://scielo.senescyt.gob.ec/pdf/rpolit/v51n2/2477-8990-rpolit-51-02-00113.pdf>

Anexos

Anexo A. Tabla de referencias

#	TÉRMINO O FRASE	FECHA	URL	BASE DE DATOS	NOMBRE DEL ARTÍCULO	TIPO
1	"java" and "gps" and "software" and "colombia" and "transporte publico"	2018	https://ridum.umanizales.edu.co/server/api/core/bitstreams/8df5333-60bc-44bf-b576-a60183fdde47/content	Google Sholar	CITY-RUTA: APLICACIÓN PARA EL SERVICIO DE TRANSPORTE PÚBLICO	Artículo
2	"artículo" AND "JAVA" AND "framework"	2015	https://personal.tuxes.es/universidad/phd/arquitecturas_ebusiness.pdf	Google Sholar	Java Struts Framework	Artículo
3	"java" and "gps" and "software" and "colombia" and "transporte publico" and "artículo"	2016	https://core.ac.uk/download/pdf/352368119.pdf	Google Sholar	Aplicación móvil apoyada en georeferenciación que permita optimizar el uso del transporte público en la ciudad de Cúcuta (STOPBUS)	Artículo
4	"gps" and "aplicación" and "artículos" and "software"	2016	http://scielo.sld.cu/scielo.php?pid=S1815-59282016000200002&script=sci_abstract	Google Sholar	Dispositivo de adquisición y transmisión de la posición de un vehículo mediante GPS y Wi-Fi	Artículo
5	"artículos" and "gps" and "mqtt" and "java"	2016	https://web.archive.org/web/20180415091800id/http://revistas.usc.edu.co/index.php/ingenium/article/viewFile/651/518	Google Sholar	SisMo: sistema de seguridad para motocicletas	Artículo
6	"architecture" AND "GPS tracking" AND "MQTT" AND "java"	2025	https://revistareg.com/index.php/1/article/view/90/265	Google Sholar	Diseño de una Aplicación Móvil para el Monitoreo de los Buses de la UTELVT	Artículo
7	artículo AND "software" AND "metodologías ágiles" AND "scrum"	2025	https://ciencialatina.org/index.php/cienciala/articulo/view/17963/25824	Google Sholar	ADAPTACIÓN DE SCRUM AL DESARROLLO DE SOFTWARE CUÁNTICO: UN ARTÍCULO DE REVISIÓN METODOLÓGICA	Artículo
8	"artículo" AND "software" AND "COLOMBIA"	2015	http://www.scielo.org.co/pdf/eia/n23/n23a09.pdf	Google Sholar	EL CRECIMIENTO DE LA INDUSTRIA DEL SOFTWARE EN COLOMBIA: UN ANÁLISIS SISTÉMICO	Artículo
9	"artículo" AND "JAVA" AND "api-rest"	2022	https://d1wqtxts1xzle7.cloudfront.net/111860414/18841-libre.pdf?1708974048=&response-content-disposition=inline%3B+filename%3DEasyRest_Generador_automatizado_API_Rest.pdf&Expires=1756126855&Signature=PfW2C2QHi7Wp48UhUwi0RQpIUgRsm1r7NunpWEpVpd0Z~FzGJeMnHsyJFakewo~clic~m77kUOvu2tVwmrWcNiykmhOcg4K2LxPdef10U0pPvQXoJ~CSi~F3NRVO6Mjdjm5yVyc3HeHYwNjVkuAZJab6ouhUJZvXNqLO78Mt~EXjW~FpmHQzXixQdeRsuUMHJsKjarLwrHITk9hsk~a3hfYXIw8JQp3DksHZKQq16kkHgw-fvVu7A~4pbB6xEP0J14ytVkbzxQ0IWA~IWIJMIhRp4werkbMab0~kBVJqb30R4sHCJWMHWSJBPCQ8n6ASbDmAfKdZgc9LZiAvz64Lg_&Key-Pair-Id=APKAILOHF5GGSRLRBV4ZA	Google Sholar	EasyRest: Generador automático de API Rest basado en Spring Framework y motor de plantillas Beetl	Artículo
10	"artículo" AND "react" AND "jwt"	2021	https://sedici.unlp.edu.ar/bitstream/handle/10915/140261/Documento_completo.pdf-PDFA.pdf?sequence=1	Google Sholar	GotenJS: Framework para generar la estructura base de una aplicación web	Artículo
11	"patrones de diseño" and "software" and "artículo"	2024	https://www.investigarmqr.com/ojs/index.php/mqr/article/view/886/3451	Google Sholar	Incidencia de los patrones de diseño de software en la seguridad de aplicaciones web	Artículo
12	"reutilización de código" and "software" and "artículo"	2023	https://ve.scielo.org/pdf/uct/v27n121/2542-3401-uct-27-121-85.pdf	Google Sholar	El papel de la programación orientada a objetos en el desarrollo de software sostenible y escalable	Artículo
13	"motores de bases de datos" and "software" and "artículo"	2022	https://sedici.unlp.edu.ar/bitstream/handle/10915/144325/Documento_completo.pdf-PDFA.pdf?sequence=1	Google Sholar	ASPECTOS DE INGENIERÍA DE SOFTWARE, BASES DE DATOS RELACIONALES, NO RELACIONALES Y COMO SERVICIOS EN LA NUBE PARA EL DESARROLLO DE SISTEMAS DE SOFTWARE HÍBRIDOS.	Artículo
14	"pruebas de rendimiento" and "software" and "artículo"	2022	https://www.redalyc.org/journal/6078/607870799010/607870799010.pdf	Google Sholar	MACHINE LEARNING APLICADO AL ANÁLISIS DEL RENDIMIENTO DE DESARROLLOS DE SOFTWARE	Artículo
15	"proyectos" and "software" and "artículo"	2024	https://dspace.ups.edu.ec/bitstream/123456789/28055/1/UPS-CT011453.pdf	Google Sholar	Prácticas empíricas en la gestión de proyectos de desarrollo de software a partir de análisis del principio de adaptabilidad	Artículo
16	"arquitectura" and "software" and "artículo"	2022	https://www.redalyc.org/journal/5337/533774788007/533774788007.pdf	Google Sholar	Arquitectura de software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra	Artículo
17	"react native" and "software" and "artículo"	2022	https://ciencialatina.org/index.php/cienciala/articulo/view/2959/4350	Google Sholar	DESARROLLO HÍBRIDO CON FLUTTER	Artículo
18	"react native" and "software" and "artículo" and "movil"	2022	https://ciencialatina.org/index.php/cienciala/articulo/view/2950	Google Sholar	Análisis comparativo de entornos de desarrollo móvil	Artículo
19	"patrones de diseño" and "software" and "artículo"	2021	http://scielo.sld.cu/pdf/rccl/v15n2/2227-1899-rccl-15-02-118.pdf	Google Sholar	Módulo de recomendación de patrones de diseño para EGPat	Artículo
20	"patrones de diseño" and "software" and "artículo" and "implementación"	2023	http://scielo.senescyt.gob.ec/pdf/rpolit/v51n2/2477-8990-rpolit-51-02-00113.pdf	Google Sholar	Elementos Reusables para Experimentar con Metodologías Basadas en Estrategias y Patrones de Privacidad	Artículo