

Design and Implementation of a Node Geolocation System for Fire Monitoring through LoRaWAN

Pedro Luna
Facultad de Ingeniería
Universidad Panamericana
Aguascalientes, México
pedro.luna@up.edu.mx

Sebastián Gutiérrez
Facultad de Ingeniería
Universidad Panamericana
Aguascalientes, México
jsgutierrez@up.edu.mx

Ricardo Espinosa
Facultad de Ingeniería
Universidad Panamericana
Aguascalientes, México
respinosa@up.edu.mx

Abstract—Maturity of LoRAWAN based solutions has led to the integration of innovative services in IoT systems, being geolocation a remarkable feature that enables functionalities as asset tracking. The before mentioned capability, combined with Real-time responsiveness and data analysis could derive to the possibility of smarter systems that significantly add value to applied technology. The current work contains the implementation of a technique for geolocation based on LoRaWAN which has been integrated into an IoT end to end (e2e) solution that monitors the probability of an existing fire. By adding this characteristic, the system can locate a fire incident at the right moment and prevent material, economic and environmental affectations.

Index Terms—Cloud computing, Fire monitoring, IoT, LoRaWAN, Reactive systems, Real-Time monitoring.

I. INTRODUCTION

La interacción con el software se ha transformado drásticamente durante la última década, el surgimiento de nuevas tecnologías y la mejora de las ya existentes han llevado a los usuarios de tecnología a cambiar sus hábitos de consumo [1]. La mejora en la conectividad mundial ha generado aumentos significativos en la velocidad, la cobertura y el acceso a Internet. En este contexto, el Internet de las cosas (IoT) y la computación en la nube han ganado relevancia e interés entre los desarrolladores de software y la academia. Ambos temas, con sus propios problemas y peculiaridades, enfrentan desafíos comunes debido a su naturaleza distribuida y basada en Internet, donde la disponibilidad, confiabilidad y el uso adecuado de los recursos son asuntos críticos.

Actualmente, las áreas de conocimiento como diseño de sistemas reactivos, ya han abordado una propuesta para hacer frente a estos escenarios. Aunque los sistemas de IoT se pueden diseñar y desarrollar bajo paradigmas de POO (Programación Orientada a Objetos), que tiene una gran variedad de patrones de diseño bien probados como GoF o MVC (Modelo Vista Controlador), el diseño reactivo tiene un enfoque particular en sistemas que esperan manejar miles de millones de clientes [2], un requisito que es imperativo para cumplir en cualquier sistema de IoT debido a la cantidad de dispositivos actualmente conectados a Internet [3]. Dada la diversidad en los dominios de aplicación de IoT [4], se requiere un tema concreto para probar las declaraciones antes mencionadas. En particular, se ha seleccionado el monitoreo de incendios,

debido a que la literatura existente [5], para crear una solución para la detección temprana y su alerta oportuna, requiere una cantidad asequible de recursos, y la prevención de incendios, significa un impacto económico, relevante en las circunstancias descritas en [6] [7] [8] [9] [10], muchos de los incidentes de incendios tienen lugar en áreas abiertas, donde la electricidad y las telecomunicaciones en muchas ocasiones no siempre están disponibles.

LoRa, es un protocolo de comunicación de largo alcance y con bajo consumo de energía que opera en la banda ISM. Utiliza como esquema de modulación de la señal una variación de la modulación DSSS de tipo Spread Spectrum denominada Chirp Spread Spectrum (CSS), permitiendo obtener conexiones de bajo costo y bajo consumo, demostrando su competencia en áreas suburbanas y rurales [11] [12] [13]. Como complemento, LoRaWAN es una especificación para redes de baja potencia que implementa una capa extra de seguridad para la privacidad, integridad y disponibilidad de los datos y, al mismo tiempo, vincula LoRa con la infraestructura de red clásica, lo que permite la posibilidad de utilizar los datos recopilados con otros servicios alojados en la nube [14] [15]. Por esta razón, LoRa y LoRaWAN han sido seleccionados como protocolos de comunicación para este proyecto.

Debido a la importancia de ubicar la posición de cada de los sensores para la detección temprana de incendios en áreas suburbanas y rurales, las soluciones basadas en GNSS (Sistema Global de Navegación por Satélite) actualmente han sido sustituidas por métodos innovadores para obtener la posición de varios sensores. En [16], se propone un análisis de datos generados por los propios sensores para su geolocalización a través de algoritmos de clasificación utilizando computación en la nube. Además en [17], se diseñó e implementó un sistema completo de seguimiento de IoT para presentar resultados de precisión utilizando la tecnología LoRa, se utiliza la diferencia del tiempo de los datos de llegada para calcular las coordenadas de las estaciones, en [18], se explica la técnica de localización TOA en combinación con TDOA para mejorar la precisión de localización de sensores. En [19], se presenta un protocolo de localización LoRa que a través de métodos de ángulo de llegada (AoA), TDA y TDOA permite localizar los sensores propuestos.

Para este proyecto se utilizó la API de la empresa Semtech

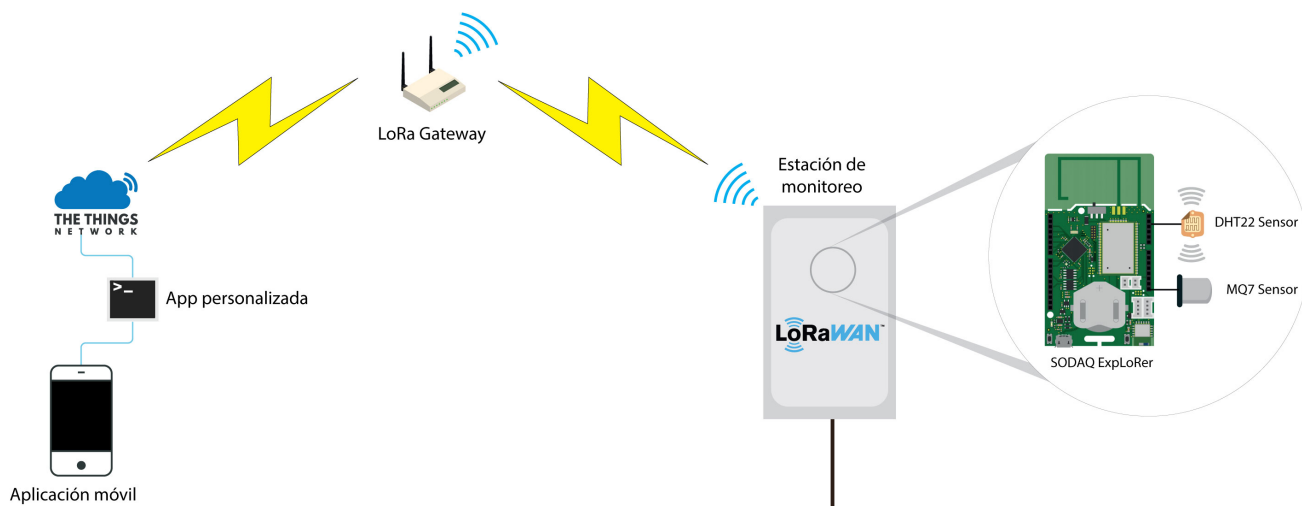


Fig. 1. Arquitectura del sistema basado en IoT.

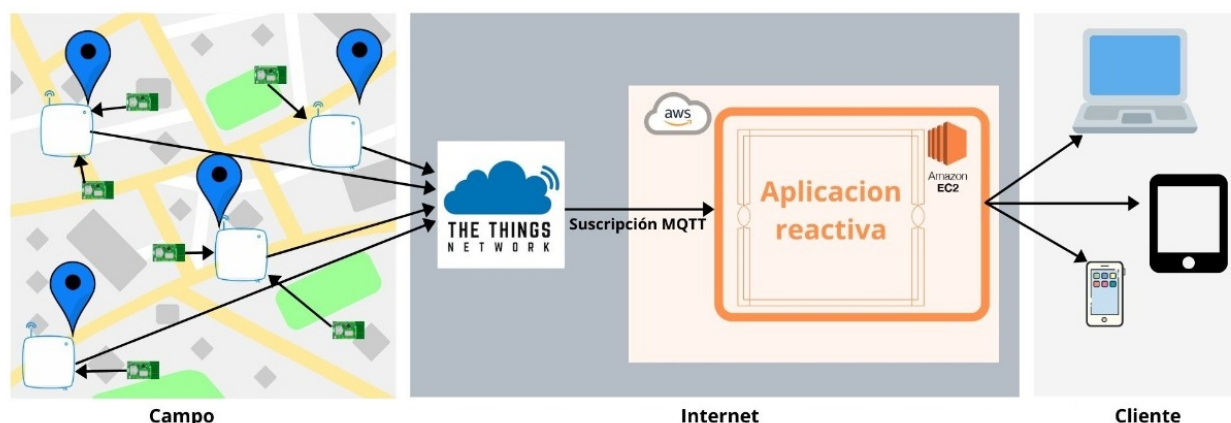


Fig. 2. Flujo de datos.

para realizar la geolocalización [20], este método calcula la posición utilizando el indicador de intensidad de señal recibida por sus siglas en inglés (RSSI), relación de ruido de señal (SNR), TDOA (en inglés, Time Difference of Arrival) y las coordenadas (Latitud, Longitud) de la gateways LoRa.

La propuesta de este proyecto de investigación presenta una solución en la integración de recolección de datos, procesamiento, geolocalización y visualización de datos a través de una aplicación móvil que muestra el estado y ubicación de los nodos de las estaciones de monitoreo de incendios a través de LoRaWAN. El documento está organizado de la siguiente manera: La Sección II describe los materiales y métodos desarrollados. La sección III ilustra los resultados. Finalmente en la sección IV se presentan las conclusiones.

II. MATERIALES Y MÉTODOS

En la Figura 1 se muestra la propuesta de la arquitectura basada en IoT la cual se ha diseñado utilizando la especificación LoRaWANTM 1.1 [21]. Todo el hardware relacionado

con la comunicación LoRa ha sido seleccionado, configurado y aprovisionado de acuerdo con los parámetros regionales LoRaWAN (RP002-1.0.1) [22]. La aplicación ha sido programada en lenguaje Go utilizando los requisitos establecidos por los patrones de diseño reactivo [2], y la integración de servicios y procesamiento de metadatos ha sido posible utilizando la información contenida en la especificación [23] de LoRaWAN Backend Interfaces 1.0. La Tabla 1 contiene un resumen de los materiales utilizados para implementar y probar la solución en Aguascalientes, centro de México (21.9 °N, -102.3 °E), utilizando el plan de frecuencia de 915 MHZ. Desde una perspectiva de flujo de trabajo, esta solución utiliza la canalización de datos descrita en la Figura 2, donde cada elemento procesa los datos de la siguiente manera:

- 1) Los nodos finales recopilan información del entorno y envían los datos a los gateways utilizando el protocolo LoRa. Cada nodo final consiste en una placa SODAQ ExpLoRer equipada con un sensor DHT22 para leer los niveles de temperatura y humedad, y un sensor

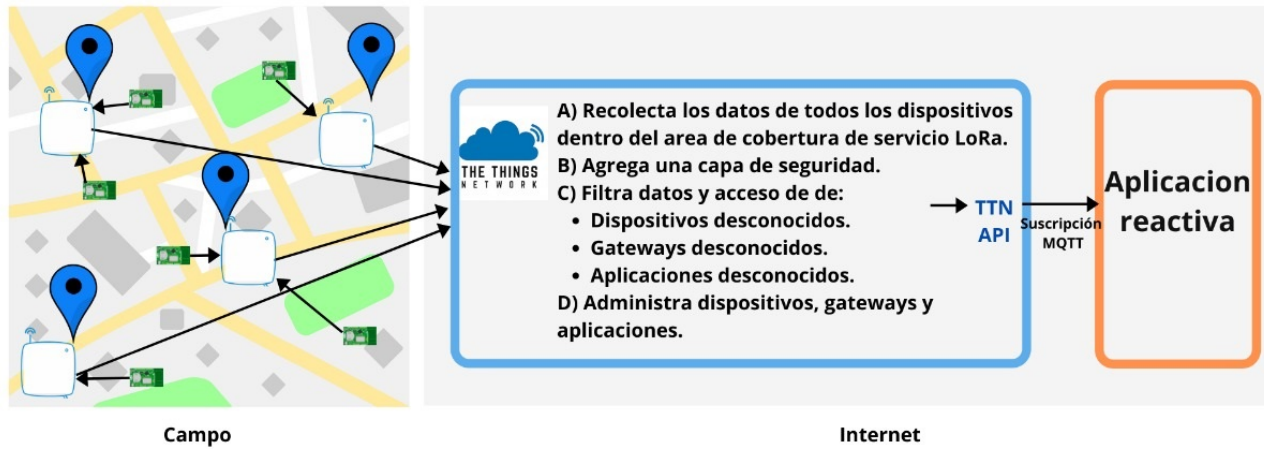


Fig. 3. Proceso de Geolocalización.

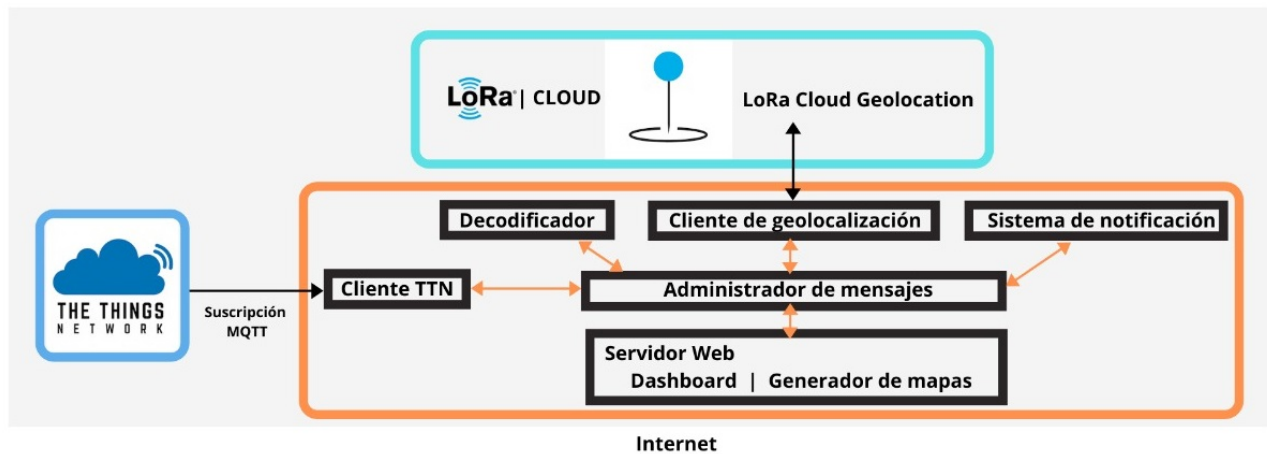


Fig. 4. Arquitectura de aplicación.

TABLE I
MATERIAL REQUERIDO PARA NUESTRA PROPUESTA DE ARQUITECTURA IoT

Componentes de la Solución			
Dominio	Componente	Modelo	Fabricante
Nodo Final	Tarjeta de prototipo	SODAQ	Microchip
		ExpLoRer	
	Temperatura	DHT22	—
	Humedad Sensor	MQ7	—
Concentrador	Gateway LoRa	Wirnet iFemtoCell 915	Kerlink
Servicios Red / Nube	LNS	Public Community Network	TTN
	Compute	AWS (Amazon Web Service)	Amazon
	Resources	EC2	
	Geolocalización solver	Geolocalización LoRa Cloud Services	Semtech
Aplicación	Custom Application*	—	—

*Desarrollado usando Go como lenguaje de programación, more details in section .

MQ7 para medir los niveles de CO. Se unen a la red LoRa mediante el método OTAA y operan en el modo Clase A, que permite comunicaciones bidireccionales en las que la transmisión de enlace ascendente de cada dispositivo final es seguida por dos ventanas de recepción de enlace descendente cortas. El intervalo de transmisión programado por el dispositivo final se basa en sus propias necesidades de comunicación con una pequeña variación basada en un tiempo aleatorio (tipo de protocolo ALOHA). Esta operación de Clase A es el sistema de dispositivo final de menor potencia para aplicaciones que solo requieren comunicación de enlace descendente desde el servidor poco después de que el dispositivo final haya enviado una transmisión de enlace ascendente. Las comunicaciones de enlace descendente desde el servidor en cualquier otro momento tendrán que esperar hasta el próximo enlace ascendente programado.

2) Los gateways recopilan los datos de cada nodo final en el rango y envían los paquetes a través de TCP-UDP a

un LNS (LoRaWAN Network Server) específico. Este proceso se ilustra del lado izquierdo de la Figura 3. Durante la configuración, las puertas de enlace iFemtoCell deben reconfigurarse para que apunten al extremo TTN más cercano, que en este caso es ttn-router-us-west.

- 3) El LNS recibe la información enviada desde varios gateways, descarta paquetes no autorizados y asocia los datos recolectados con la aplicación correspondiente como se describe en el lado derecho de la Figura 3. Por esta razón, todos los nodos finales y gateways deben ser dados de alta previamente en la plataforma TTN. En este punto los datos no se almacenan en ningún lugar, TTN carece de un mecanismo para hacerlo. Si bien cuentan con integraciones con plataformas de terceros que permite hacerlo, en este caso los datos no son necesarios para un análisis posterior, por lo que es más efectivo consumir los datos en tiempo real, esta tarea la realiza la aplicación personalizada utilizando TTN Data API que permite crear un puente entre el servicio de bróker TTN MQTT y nuestra aplicación.
- 4) Como se muestra en la Figura 4, una suscripción MQTT notifica a la aplicación cada vez que hay nuevos datos disponibles, por lo que puede obtener un flujo con formato JSON de un enlace ascendente LoRa que está estructurado según lo definido por [21] y [23]. Una vez que se han adquirido los datos, la aplicación activa dos operaciones ETL (Extraction Transformation Load) simultáneamente. Estas operaciones se detallan en 5) y 6). Esta aplicación fue desarrollada íntegramente en Go, aprovechando su naturaleza concurrente y multiparadigma. Lamentablemente, no existe una implementación del modelo de actor en lenguaje de programación Go, situación que tuvo que resolverse creando una arquitectura con un patrón de diseño de software singleton para cada suscripción y luego un observador para cualquier cliente que requiera obtener información de un monitor determinado. Sin embargo, la arquitectura de la aplicación sigue los principios reactivos que deben ser receptivos, resilientes, elásticos e impulsados por mensajes. Esto permite que la aplicación maneje conexiones masivas, garantice su propia disponibilidad y confiabilidad, y aumente y disminuya dinámicamente según la demanda del servicio.
- 5) Una de las operaciones ETL llamada decodificador de carga útil (en inglés payload decoder), consiste en extraer la carga útil del enlace ascendente LoRa y transformar su contenido mediante un decodificador, este decodificador convierte una secuencia de valores hexadecimales en diferentes tipos de valores, la carga útil también contiene un identificador que permite saber qué tipo de sensor ha generado el valor. Esta información se organiza y se carga en el servicio del panel de control de la aplicación como se muestra en la Figura 5.
- 6) La otra operación ETL usa los metadatos de uplink y extrae los valores requeridos de los servicios LoRa Cloud para calcular la geolocalización, estos valores se

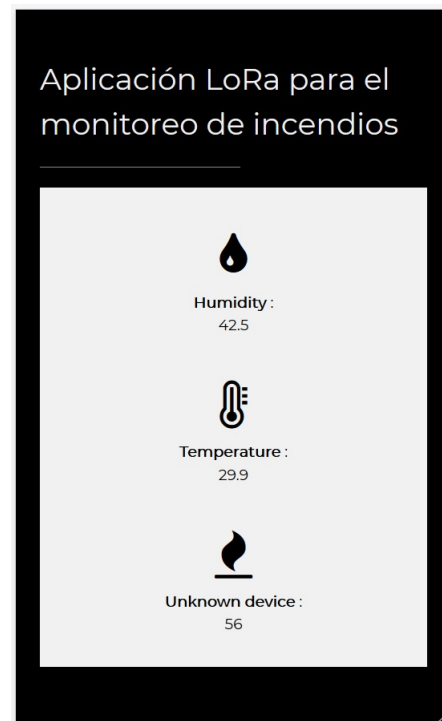


Fig. 5. Aplicación LoRa para el monitoreo de incendios.

transforman en una estructura JSON y se cargan en el servicio de geolocalización, como se muestra en la Figura 6.

- 7) Una vez que los resultados de ambas operaciones ETL están disponibles, los datos se evalúan utilizando la regla 30-30-30, algoritmo utilizado para la prevención de grandes incendios forestales, se ha demostrado su eficacia en [10].
- 8) Finalmente, si existiera la probabilidad de un incendio, la aplicación enviaría una alerta por correo electrónico con un mensaje de advertencia y las coordenadas del monitor que ha informado del incidente. De lo contrario, la información se almacena temporalmente y se descarta una vez que llegan nuevos datos. En el tablero de monitoreo de la aplicación, se agregó un botón para recuperar manualmente la posición de un monitor determinado, se agregó un segundo botón para llamar a la API de Google Maps y renderizar un mapa para tener una referencia visual de la posición del monitor, como se muestra en la Figura 7.

III. RESULTADOS

En esta aplicación, el trabajo previo relacionado con el monitoreo de incendios utilizando el protocolo de comunicación LoRa fue fácilmente replicado, confirmando que este protocolo es adecuado para resolver este tipo de problemas. El objetivo de integrar servicios para agregar valor a soluciones anteriores se ha logrado mediante la integración de geolocalización, notificaciones y una interfaz descriptiva que muestra correctamente los datos en tiempo real en una interfaz simple

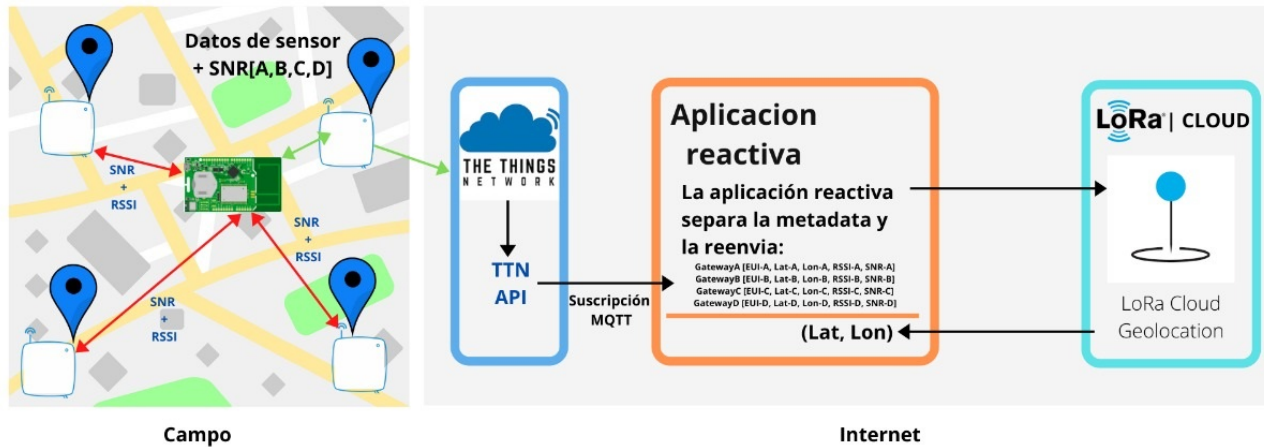


Fig. 6. Proceso de Geolocalización.

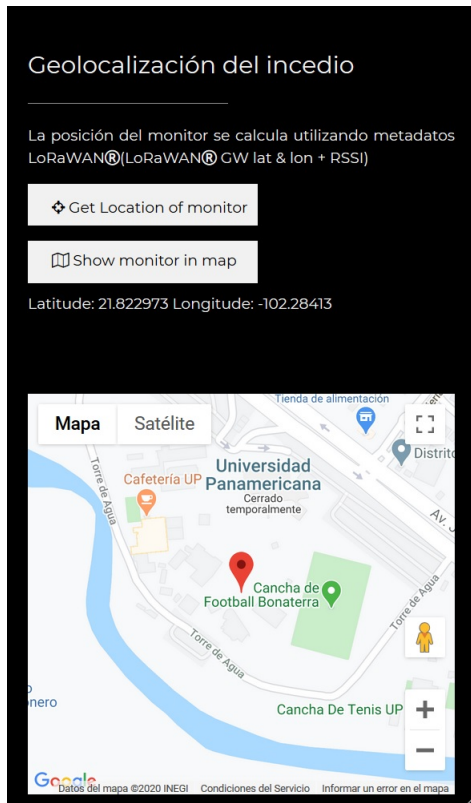


Fig. 7. Pantalla de geolocalización de incendios

y descriptiva. Durante el desarrollo de la aplicación se realizó la implementación de elementos sobre patrones de diseño reactivo, obtenido un comportamiento aceptable siguiendo las recomendaciones dadas por [2]. La aplicación pudo notificar satisfactoriamente el incidente de incendio como se ilustra en la Figura 8, esta prueba fue controlada fue realizada dentro de las instalaciones de la Universidad Panamericana, el cual, mediante una notificación por correo electrónico, se pudo

mostrar la localización del nodo monitor.

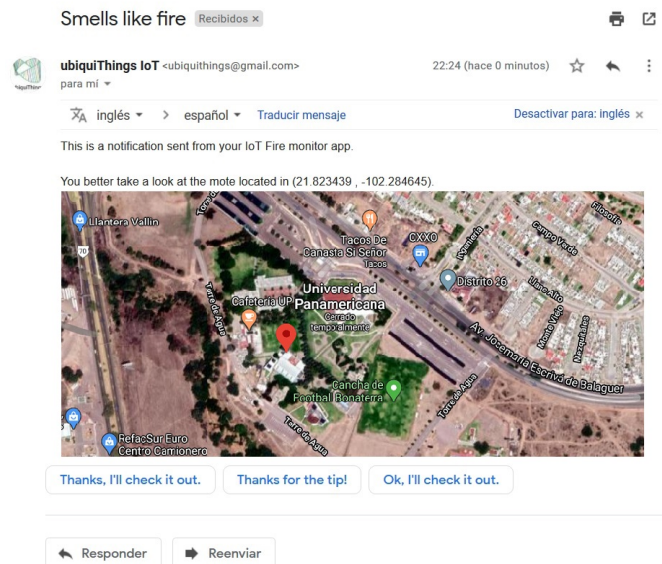


Fig. 8. Notificación de incendio via correo electronico.

IV. CONCLUSIONES

Para poder hacer frente a una solución que cumpla con los requisitos exigidos por los protocolos de IoT, aún es necesario integrar más elementos. Una mejora significativa para cumplir con los sistemas reactivos, se puede lograr utilizando recursos como Scala, lenguaje de programación creado para desarrollar sistemas reactivos, el cual cuenta con implementaciones de componentes requeridos para este tipo de aplicaciones. Otro recurso que podría utilizarse para mejorar este proyecto de investigación son los contenedores para crear microservicios. La novedad del presente estudio es demostrar la capacidad de integrar una solución LoRaWAN para monitorear la probabilidad de incendios existentes en

una ubicación determinada mediante la explotación de los beneficios que otorga el protocolo LoRa y su capacidad para interactuar con los servicios en la nube.

RECONOCIMIENTO

Un agradecimiento especial a la Universidad Panamericana por permitir el uso de su infraestructura LoRa y a todos los miembros del equipo de I+D de Semetech Corp. en Rapperswill, Suiza.

REFERENCES

- [1] J. Bonér, D. Farley, R. Kuhn, and M. Thompson, "The Reactive Manifesto," *Reactivemanifesto.Org*, 2014.
- [2] R. Kuhn, B. Hanafée, and J. Allen, *Reactive design patterns*. Manning Publications Company, 2017.
- [3] Gartner Says 5.8 Billion Enterprise and Automotive IoT End points Will Be in Use in 2020, Online: <https://www.gartner.com>, Last access: May 30, 2020
- [4] O. Vermesan et al., "Internet of Things: Strategic Research Roadmap," in *Internet of Things-Global Technological and Societal Trends*, 2011.
- [5] G. Saldamli, Et Al, Wildfire Detection using Wireless Mesh Network, Fourth International Conference on Fog and Mobile Edge Computing, 2019.
- [6] E. Surya, M. Rosmiati, F. Rizal, Integrating Forest Fire Detection with Wireless Sensor Network Based on Long Range Radio, International Conference on Control, Electronics, Renewable Energy and Communications, 2018.
- [7] S. Rizanov, A. Stoyanova, D. Todorov, System For Early Warning And Monitoring Of Wildfires, Proceedings of the XXVIII International Scientific Conference Electronics, 2019.
- [8] M. Antunes, Et Al, Low-Cost System for Early Detection and Deployment of Countermeasures Against Wild Fires, IEEE 5th World Forum on Internet of Things, 2019.
- [9] E. Abdul, Et Al, Modelling of wireless sensor networks for detection land and forest fire hotspot, TELKOMNIKA, Vol.17, No.6, pp.2772 2781, 2019.
- [10] R. Vega-Rodríguez, Et Al, Low Cost LoRa based Network for Forest Fire Detection, Sixth International Conference on Internet of Things: Systems, Management and Security, 2019.
- [11] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," 2016 IEEE Symp. Commun. Veh. Technol. Benelux, SCVT 2016, no. 2, pp. 0–4, 2016, doi: 10.1109/SCVT.2016.7797659.
- [12] M. Saari, A. Muzaffar Bin Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, "LoRa - A survey of recent research trends," 2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc., pp. 872–877, 2018, doi: 10.23919/MIPRO.2018.8400161.
- [13] L. Vangelista, "Frequency Shift Chirp Modulation: The LoRa Modulation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, 2017, doi: 10.1109/LSP.2017.2762960.
- [14] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A Survey of LoRaWAN for IoT: From Technology to Application," *Sensors (Basel)*, vol. 18, no. 11, 2018, doi: 10.3390/s18113995.
- [15] J. De Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino, "LoRaWAN - A low power WAN protocol for Internet of Things: A review and opportunities," in 2017 2nd International Multidisciplinary Conference on Computer and Energy Science, SpliTech 2017, 2017.
- [16] E. Estrada, M. P. M. Vargas, J. Gómez, A. P. P. Negron, G. L. López, and R. Maciel, "Smart cities big data algorithms for sensors location," *Appl. Sci.*, 2019, doi: 10.3390/app9194196.
- [17] B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANS," in *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, 2017, doi: 10.1109/GIOTS.2017.8016251.
- [18] A. A. A. Elsabaa, M. Ward, and W. Wu, "Hybrid localization techniques in lora-based WSN," in *ICAC 2019 - 2019 25th IEEE International Conference on Automation and Computing*, 2019, doi: 10.23919/ICAC.2019.8895196.
- [19] C. Rus, Et Al, "LoRa communication and geolocation system for sensors network", *MATEC Web of Conferences* 305, 00043, 2020
- [20] API v3 (LoRa® TAORSSI), Online: <https://www.loracloud.com/documentation/geolocation?url=v3.html>, Last access: June 11th 2020.
- [21] N. Sornin, A. Yegin, LoRaWANTM 1.1 Specification. LoRa Alliance, Inc., 2017.
- [22] LoRaWAN® Regional Parameters RP002-1.0.0, Online: <https://loralliance.org/resource-hub/lorawanr-regional-parameters-rp002-100>, June 11th 2020.
- [23] N. Sornin, A. Yegin, LoRaWAN Backend Interfaces 1.0 Specification. LoRa Alliance, Inc., 2017.