



Sistema de monitoreo de monóxido de carbono en tiempo real en el hogar como aplicación de Internet de las Cosas

Real-time home monitoring system of carbon monoxide as an application of Internet of Things

Julio César Martínez Zárata ^a, & Oscar Alonso Durango Román ^b

^a Magíster en Ingeniería - Politécnico Colombiano Jaime Isaza Cadavid, analista de aplicaciones y servicios, Institución Universitaria Pascual Bravo, Medellín - Colombia, julio.martinez@pascualbravo.edu.co

^b Tecnólogo en Desarrollo de software - Universidad abierta y a distancia, asistente de aplicaciones y servicios, Institución Universitaria Pascual Bravo, Medellín - Colombia, oscar.durango@arrobamedellin.edu.co

Recibido: mayo 9, 2019. Aceptado: diciembre 26, 2019.

Resumen

En este artículo, se diseña un sistema para monitoreo de monóxido (CO) en tiempo real y de bajo costo como una aplicación al Internet de las cosas (IoT) en el hogar. Para ello, en primer lugar, se realiza una exploración de conceptos relacionados con la realización del sistema. Luego, se realiza la implementación de un prototipo mediante el uso de tecnologías emergentes para Back-end y Front-end tales como: C++, Spring Boot, ReactJS, Bootstrap material design, Eclipse Paho con protocolo MQTT, Servidor Broker + Websockets sobre contenedor Docker, MySQL. Se lleva a cabo una arquitectura message broker, con el uso del patrón Publish / subscribe. Un dispositivo Arduino se conecta (a través de un API) con un cliente MQTT, envía información que captura hacia el servidor Broker, y otro cliente en un navegador web podrá acceder desde cualquier lugar del mundo y visualizar la información en tiempo real y sobre de cualquier dispositivo. Se realizan las pruebas de funcionamiento a través del prototipo para la medición del CO en el hogar, con énfasis en el envío de alertas (vía e-mail y notificaciones push), cuando los niveles alcancen un tope máximo establecido tolerable por el ser humano. El presente artículo hace énfasis en el uso de herramientas software para su implementación.

Palabras Clave: Circuitos, Internet de las Cosas, software de aplicaciones, servicios web, salud y seguridad.

Abstract

This paper presents the design of a real-time, low-cost monoxide (CO) monitoring system as a home application to the Internet of Things (IoT). At first, the research of concepts related to the realization of the system is carried out. Then, the implementation of a prototype is done using emerging technologies for backend and front end, such as C++, Spring Boot, ReactJS, Bootstrap material design, Eclipse Paho with MQTT protocol MQTT on Docker container, MySQL. The prototype employs a message broker architecture with the use of the Publish / Subscribe pattern. An Arduino device is connected (through an API) with an MQTT client, sending information to be captured with the Broker server. A client in a web browser can access from anywhere in the world and watch real-time information on any device. The performance tests were applied with the prototype for the measurement of CO at home, focused on sending send alerts (via e-mail and push notifications) when the levels reach a maximum established limit tolerable by people. This paper focuses on software tools.

Keywords: Circuits, Internet of Things, Application software, web services, health and safety.

NOMENCLATURA

CO: Monóxido de carbono

IoT: Internet de las cosas

TI: tecnologías de la información.

1 INTRODUCCIÓN

El término “Internet de las Cosas” (IoT, por su acrónimo en inglés) es relativo, no existe una definición única que sea aceptable para la comunidad mundial de usuarios. De hecho, hay muchos grupos diferentes que incluyen

Citar como:

Julio César Martínez Zárata, & Oscar Alonso Durango Román. “Sistema de monitoreo de monóxido de carbono en tiempo real en el hogar como aplicación de Internet de las Cosas”. Revista CINTEX, Vol. 24(2), pp. 25-32. 2019.

académicos, investigadores, profesionales, innovadores, desarrolladores y personas corporativas que han definido el término, aunque su uso inicial se ha atribuido a Kevin Ashton, un experto en innovación digital [1], que define el IoT como: "Una red abierta y completa de objetos inteligentes que tienen la capacidad de auto-organizarse, compartir información, datos y recursos, reaccionando y actuando ante situaciones y cambios en el entorno". Ha sido pronosticado que el IoT tomará de 5 a 10 años para la adopción del mercado [2]. Los sistemas de internet de las cosas se han aplicado cada vez más en distintas áreas, incluso la industria, en la agricultura, el bienestar, salud y educación [3].

En el presente artículo, se propone un sistema para la detección de monóxido de carbono (CO) en tiempo real en el hogar, aplicado al Internet de las Cosas, integrado con un aplicativo software diseñado para brindar distintos beneficios a la hora de realizar el monitoreo y el control del hardware. El Sistema software utiliza una arquitectura Message Broker [4] que brinda servicios para transmisión de datos en tiempo real mediante websockets [5].

El monóxido de carbono es un gas tóxico dañino para los humanos, además es difícil detectar por los sentidos, puesto que no tiene olor, color, ni sabor; y las personas dentro de su hogar pueden estar expuestas fácilmente, puesto que se hace presente en distintos aparatos al interior, como por ejemplo en estufas a gas natural, calentadores a gas, automóviles dentro de los garajes, entre otros [6]. Un sistema de bajo costo, y sencillo de implementar en los hogares puede detectar con facilidad los niveles de CO en tiempo real, y emitir alertas tempranas a cualquier sitio donde se encuentren los habitantes, cuando los niveles propasen los permitidos y tolerables por las personas y de esta forma se podrían evitar calamidades.

Este artículo, está organizado de la siguiente manera: en la sección 2, se presenta el desarrollo del artículo en general, el cual incluye: los trabajos relacionados, la metodología utilizada para dicho desarrollo, el diseño e implementación del sistema propuesto; donde se muestran las herramientas tanto software como hardware utilizadas para tal fin, y por último la implementación del prototipo. En la sección 3, la metodología. Finalmente, se presentan los trabajos futuros y la conclusión.

2 DESARROLLO DEL ARTÍCULO

A continuación, se muestra la parte más importante del presente artículo. Se inicia enunciando algunos trabajos relacionados, exponiendo las diferencias y semejanzas entre estos y el presente. Luego, se presenta la metodología, donde se detalla qué y cómo se llevó a cabo este proyecto, y otros elementos importantes.

2.1 TRABAJOS RELACIONADOS

Se citan algunos trabajos relacionados con IoT usando Arduino. En [7], se propone un sistema que hará que todo el procedimiento de reserva de cilindros de gas licuado de petróleo (GLP) se automatice sin intervención humana, utilizando un Arduino como controlador, el sistema mide continuamente el peso del cilindro y una vez que alcanza el umbral mínimo enviará automáticamente un mensaje al agente de GLP autorizado para que pueda entregar el cilindro de GLP a tiempo. En el caso de [8], se propone un sistema de control del hogar y control ambiental flexible con Arduino Mega. Emplea un servidor de micro - web integrado en el microcontrolador Arduino Mega 2560, con conectividad IP para acceder y controlar dispositivos de forma remota, aunque este sistema no requiere de un servidor, como es el caso del presente artículo, ofrece un protocolo de comunicación para monitorear y controlar el entorno doméstico con más que solo la funcionalidad de conmutación; también emplea sensores de gas. En [9] se realiza un trabajo empleando Arduino UNO, haciendo propuesta de un sistema inteligente de alertas para el despacho de basura dando una señal de alerta al servidor web municipal; este sistema, al igual que el propuesto en este artículo, utiliza un servidor web. Por su parte, en [10] los autores presentan un diseño de sistema de aprendizaje basado en proyectos (PBL, Project-Based Learning) sobre IoT donde emplean Spring Boot [11], Arduino y otras tecnologías con el patrón Publish / Subscribe, como es el caso del presente artículo.

En [12] se propone una metodología para la implementación de la técnica de simulación HIL (hardware-in-the-loop) para desarrollar una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes en UCI usando la metodología HIL y estimadores de estado; en [13] se plantea el diseño de un AIO (Amplificador Operacional Instrumental), un sistema para la adquisición de señales electromiográficas provenientes de los músculos bíceps y tríceps, así como la metodología seguida para obtener los resultados obtenidos. Además, en [14] se habla sobre el diseño y simulación de un sistema de medición de temperatura con el objetivo de fortalecer el aprendizaje aplicativo, donde dicho sistema registra la variable física de temperatura por medio de un sensor, cuya señal de voltaje analógica es convertida a una señal digital por medio de un microcontrolador y, finalmente, este valor digitalizado de la temperatura se visualiza en un display LCD. Por su parte, en [15] se hace propuesta de un sistema autosostenible de alarma temprana en corrientes fluviales, donde se realiza un prototipo de bajo costo para realizar pruebas y verificar funcionalidad y eficiencia de estaciones de monitoreo; también se utiliza el sistema de alarma con mensajes de texto mediante telefonía celular, en el caso

del sistema de éste artículo, es mediante correo electrónico, vía internet, que puede llegar a un móvil mientras el dispositivo se encuentre conectado a una red con internet y teniendo instalada una aplicación cliente de correos electrónicos. Otro trabajo relacionado es el de [16], donde se hace utiliza el concepto de internet de las cosas aplicado a los procesos logísticos de máquinas dispensadoras; en ese trabajo se utiliza la recolección de información mediante dispositivos hardware, para toma de decisiones sobre el estudio en tratamiento.

2.2 METODOLOGÍA

En este trabajo inicialmente se realizó un estudio exploratorio, sobre los distintos conceptos pertinentes y relacionados con la investigación. Después de ejecutar dicho estudio, comprender cada uno de los conceptos y utilidades, la relación y la aplicación en el proyecto relacionado y haber comparado diferentes arquitecturas y tecnologías que se pueden emplear, se seleccionaron las tecnologías pertinentes: Se seleccionó la arquitectura Message Broker con el patrón Publish / Subscribe por sus ventajas para implementación de sistemas IoT. La imagen siguiente (Fig. 1) muestra una descripción general del diseño del modelo de mensajería de publish /subscribe para visualizar la descripción general.

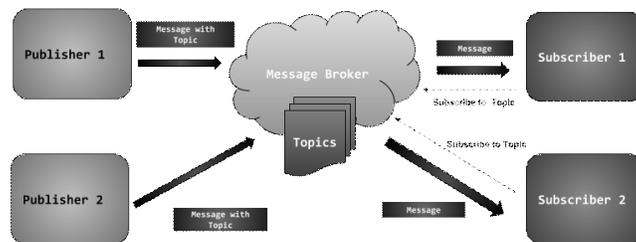


Figura 1. Modelo publish / subscribe

Se selecciona como hardware la plataforma Arduino Uno, un sistema útil para el propósito del trabajo y de bajo costo. Y para su integración, tecnologías software de desarrollo ágil como reactjs [17], springboot, y contenedores Docker [18].

Finalmente, se lleva a cabo el experimento con un prototipo funcional del sistema de IoT aplicado en el hogar, que será abordado en las secciones subsiguientes. Una vez construido el sistema e implementado se realizan pruebas con usuarios remotos.

2.3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

El esquema general del sistema se muestra en la Figura 2. Para el sistema se emplean herramientas tanto hardware como software.

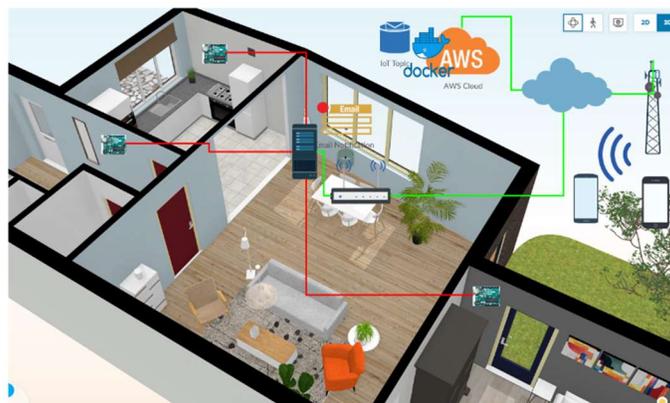


Figura 2. Esquema general del sistema IoT

2.3.1 HERRAMIENTAS DE HARDWARE

En la Tabla 1, se detalla el hardware utilizado en el experimento:

TABLA 1. HARDWARE UTILIZADO PARA EL SISTEMA HOME DE IOT

Item	Cantidad	Descripción
1	3	Arduino UNO R3 Mega328 Atmega 16u2.
2	3	Sensor MQ-7 Arduino Rasperry Gas Humo CO
3	9	Cables Jumpers Arduino 20cm
4	1	Infraestructura Amazon web service EC2

AWS (Amazon web services) cuenta con un servicio llamado EC2 (Elastic Compute Cloud), que cuenta con ventajas como las mencionadas por el proveedor Amazon, como proporción de capacidad informática en la “nube” de tamaño modificable. Los servicios en la nube poseen una gran ventaja para los desarrolladores de software, desde garantía de seguridad, hasta la fácil y rápida implementación (relativamente) de este tipo de sistemas, porque poseen una sencilla interfaz para obtener y configurar capacidad con una fricción mínima, diseñados para simplificación a escala web. Presentan servicios desde PaaS (plataforma como servicio), IaaS (infraestructura como servicio), hasta SaaS (software como un servicio). Para este proyecto se utilizaron los dos primeros servicios en mención. Como plataforma se instala un contenedor (Docker), y dentro de dicho contenedor, los servicios con Message Broker. En la Figura 3, se puede visualizar el esquema del circuito Arduino con el sensor MQ.

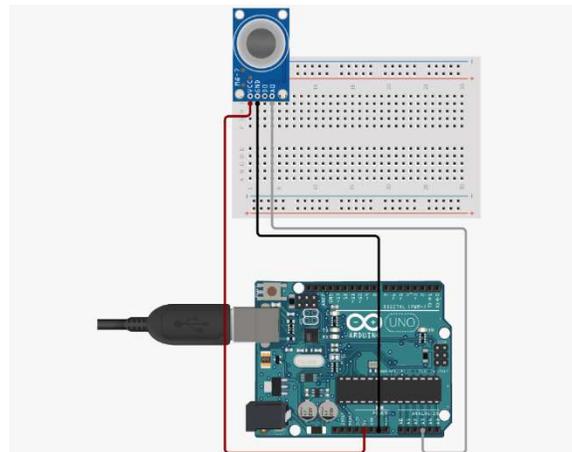


Figura 3. Esquema general del circuito Arduino – sensor MQ

2.3.2 HERRAMIENTAS DE SOFTWARE

En la Tabla 2, se detallan las tecnologías software utilizado en el experimento.

TABLA 2. SOFTWARE UTILIZADO PARA EL SISTEMA HOME DE IOT

Item	Cantidad	Descripción
1	BackEnd	Spring Boot con el lenguaje de programación Java.
2		IDE Arduino con el lenguaje de programación C++.
3		Docker con servidor MQTT sobre servidor AWS EC2
4		Bases de datos con MySQL
5	FrontEnd	ReactJS con el lenguaje Javascript y Bootstrap Material design

Los “contenedores” presentan un sinnúmero de ventajas al momento de querer desplegar servicios web. Para este caso, en el contenedor de Docker se realiza la implementación de un servidor MQTT, para la transmisión de mensajería en tiempo real, conectando aquí los clientes o suscriptores. El contenedor fue implementado en los servicios en la nube de Amazon, como ya se mencionó.

2.3.3 IMPLEMENTACIÓN DEL PROTOTIPO

Una vez armados los circuitos con Arduino y ubicados en los lugares del hogar con riesgo y para monitoreo de CO, se proceder a su implementación.

Existe una gran variedad de sensores MQ. Cada modelo está diseñado para detectar una o más sustancias, pensadas para un uso específico, como por ejemplo detección gases inflamables, calidad del aire o detección de alcohol en aire respirado. Para la lectura de concentración del gas (CO), se toma como referencia la guía oficial del dispositivo MQ7, y con la gráfica mostrada en Fig. 4.

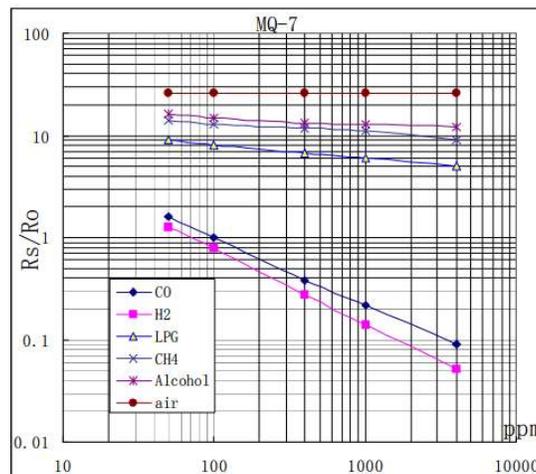


Figura 4. Características de sensibilidad del sensor MQ-7

A partir del gráfico anterior se detallan los niveles mínimos y máximos de Rs/R0 vs las partes por millón (ppm) medidas. Se emplea la ecuación (1) para hallar la concentración de CO, calibrada con resistencia.

$$Concentracion = 10^{A \log(Rs/R) + B} \quad (1)$$

Donde,

A, es la pendiente de la recta en la gráfica de la Fig. 4,

B, es el término independiente de dicha recta de la Fig. 4.

Rs/R, el cociente de resistencia mostrado en la Fig. 4.

El código en C++ es embebido en el circuito de Arduino con el uso de la herramienta Arduino IDE. Se realiza la conexión vía USB de los dispositivos Arduino hacia un equipo de cómputo cliente que recibirá la señal de los sensores de CO, teniendo en cuenta la limitación de estos cables (5 metros). Esta información es transmitida hacia el servidor Broker instalado en Docker desde el cliente MQTT en mención, el cual es desarrollado en Java con el *framework Spring Boot*. Y dicha información es transmitida a un cliente MQTT *ReactJS* [17], a donde se conectan los usuarios finales desde cualquier tipo de dispositivo para monitorear los niveles de CO (Fig. 5). Cuando se presentan niveles máximos establecidos por el usuario, se emite una alerta vía email y notificación *push* (Fig. 6).

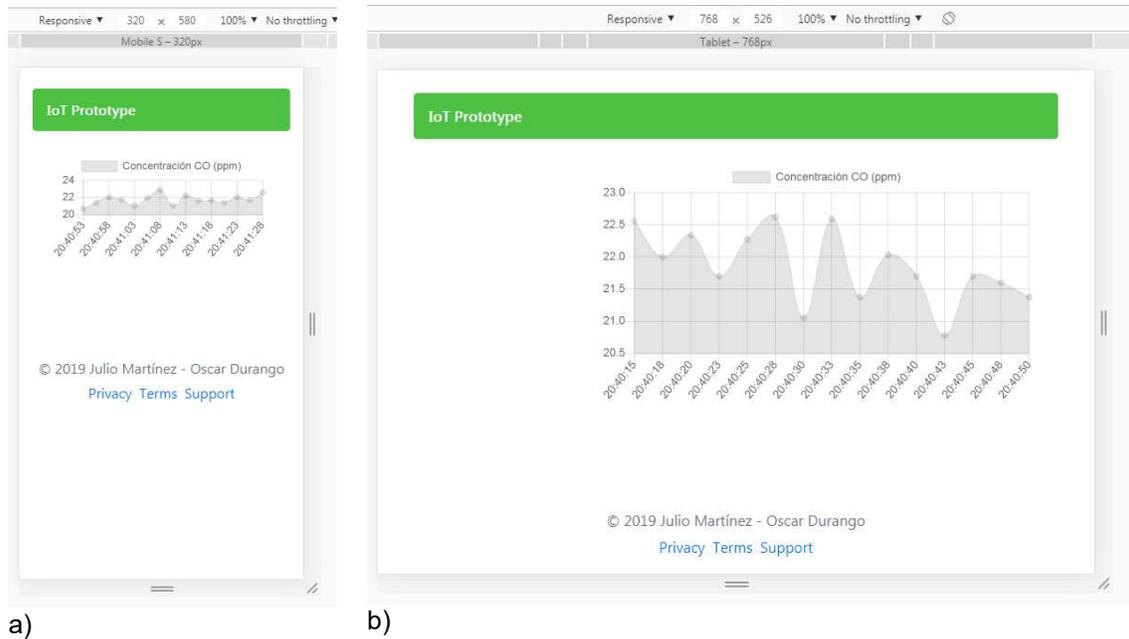


Figura 5. Pantalla de monitoreo vista desde: teléfono móvil (a), y Tablet (b).

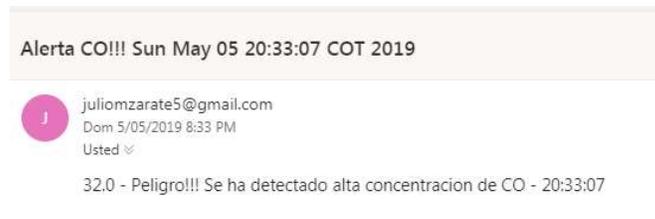


Figura 6. Notificación vía e-mail de altos niveles de CO

3 RESULTADOS Y TRABAJOS FUTUROS

Como resultado, se obtuvo un prototipo del sistema que posee un sensor de monóxido de carbono (CO) (Arduino UNO) conectado de forma cableada a un servidor local, transmitiendo una señal de forma analógica al cliente MQTT (Spring boot) que también se encuentra en el servidor mencionado, aquí se interpretan los niveles del gas, y se envían al servidor MQTT instalado en el contenedor que está en los servicios de la “nube”. Constantemente se está enviando la información (en tiempo real) al cliente (ReactJS). El cliente MQTT (Spring boot) también realiza tareas de servidor de correo electrónico; cuando el nivel de CO “sobrepasa los límites”, este envía un mensaje de correo electrónico registrado en la base de datos, y también una notificación push al aplicativo del lado del cliente (ReactJS) por medio del servidor MQTT como alerta.

El prototipo del circuito se ubica en lugares propensos a emisión de considerable cantidad de CO en el hogar (Fig. 7). Como trabajo futuro, se puede mejorar el prototipo, colocando un actuador tipo “brazo mecánico” para cerrar las llaves de control de flujo de gas, que sea controlado remotamente desde la aplicación de monitoreo o automáticamente reaccione ante las alertas de niveles altos de gas CO. Otro trabajo futuro es emplear comunicación de Arduino de forma inalámbrica, usando otro modelo con estas características. Además, se puede añadir al sistema actual otro tipo de sensores para mantener y reforzar la seguridad en los hogares.



Figura 7. Ubicación dispositivo cercano a estufa a gas

4 CONCLUSIONES

En este trabajo se presentó un sistema Internet de las cosas de bajo costo y de rápida implementación, para el monitoreo y cuidado en el hogar contra la concentración de monóxido de carbono, en donde se puede detectar de forma temprana la presencia alta de este gas dañino para la salud de las personas, mediante alertas en tiempo real.

El internet de las cosas es una tecnología emergente en la que se pueden encontrar un sinnúmero de aplicaciones para el beneficio de las personas. Entre esas aplicaciones, se pudo ver el presente sistema implementado, para este caso en el hogar. Se obtienen resultados de las tecnologías software y hardware que están a la mano para desarrollar este tipo de proyectos de forma “rápida” y de bajo costo, proponiendo mejorar la calidad de vida en los hogares, con la protección de sus habitantes contra algún tipo de contaminantes que puede hacer presencia y con el que se puede estar conviviendo.

AGRADECIMIENTOS

Este trabajo ha sido realizado en el marco del proyecto: “Prototipo de monitoreo de CO en tiempo real, utilizando tecnologías emergentes: una aplicación a smartcities”, vinculado al evento de ArduinoDay 2019, realizado en la Institución Universitaria Pascual Bravo.

Los autores reconocen las contribuciones de Mg. Julián Galeano Echeverry y del Ing. Sergio Ramírez Álvarez.

REFERENCIAS

- [1] S. Madakam, R. Ramaswamy, y S. Tripathi, «Internet of Things (IoT): A Literature Review», *J. Comput. Commun.*, vol. 03, n.º 05, pp. 164-173, 2015, doi: 10.4236/jcc.2015.35021.
- [2] J. Gubbi, R. Buyya, S. Marusic, y M. Palaniswami, «Internet of Things (IoT): A vision, architectural elements, and future directions», *Future Gener. Comput. Syst.*, vol. 29, n.º 7, pp. 1645-1660, sep. 2013, doi: 10.1016/j.future.2013.01.010.
- [3] J. J. Castro-Maldonado, J. A. Patiño-Murillo, A. E. Florian-Villa, y O. E. Guadrón-Guerrero, «Application of computer vision and low-cost artificial intelligence for the identification of phytopathogenic factors in the agro-industry sector», *J. Phys. Conf. Ser.*, vol. 1126, p. 012022, nov. 2018, doi: 10.1088/1742-6596/1126/1/012022.
- [4] S. P. Brydon y I. Singh, *Web services message broker architecture*. Google Patents, 2010.
- [5] V. Pimentel y B. G. Nickerson, «Communicating and Displaying Real-Time Data with WebSocket», *IEEE Internet Comput.*, vol. 16, n.º 4, pp. 45-53, jul. 2012, doi: 10.1109/MIC.2012.64.
- [6] J. Téllez, A. Rodríguez, y Á. Fajardo, «Contaminación por monóxido de carbono: un problema de salud ambiental», *Rev. Salud Pública*, vol. 8, n.º 1, pp. 108–117, 2006.

- [7] A. Macker, A. K. Shukla, S. Dey, y J. Agarwal, «ARDUINO Based LPG Gas Monitoring ... Automatic Cylinder Booking with Alert System», en *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, 2018, pp. 1209-1212, doi: 10.1109/ICOEI.2018.8553840.
- [8] N. David, A. Chima, A. Ugochukwu, y E. Obinna, «Design of a home automation system using arduino», *Int. J. Sci. Eng. Res.*, vol. 6, n.º 6, pp. 795–801, 2015.
- [9] N. S. Kumar, B. Vuayalakshmi, R. J. Prarthana, y A. Shankar, «IOT based smart garbage alert system using Arduino UNO», en *2016 IEEE Region 10 Conference (TENCON)*, Singapore, 2016, pp. 1028-1034, doi: 10.1109/TENCON.2016.7848162.
- [10] H. Chishiro, Y. Tsuchiya, Y. Chubachi, M. S. Abu Bakar, y L. C. De Silva, «Global PBL for Environmental IoT», en *Proceedings of the 2017 International Conference on E-commerce, E-Business and E-Government*, New York, NY, USA, 2017, pp. 65–71, doi: 10.1145/3108421.3108437.
- [11] P. Webb *et al.*, «Spring boot reference guide», *Part IV Spring Boot Featur.*, vol. 24, 2013.
- [12] C. Hernandez, D. Velez, y J. A. Isaza, «Diseño de una plataforma de prueba de sensores virtuales para el sistema glucosa-insulina de pacientes UCI usando la técnica HIL», *Rev. CINTEX*, vol. 23, n.º 2, pp. 61-75, dic. 2018, doi: 10.33131/24222208.318.
- [13] E. D. Aguirre Mesa, D. A. Garcia Arango, y C. F. Henao Villa, «Diseño y simulación de un amplificador “Instrumental Operacional” como apoyo en la lectura y transmisión de señales mioeléctricas por un solo canal con bajo ruido», *Rev. CINTEX*, vol. 22, n.º 1, pp. 11–23, 2017.
- [14] V. Srdanovic *et al.*, «Diseño y simulación de un sistema de medición de temperatura orientado hacia el aprendizaje aplicativo», *Rev. CINTEX*, vol. 19, pp. 9–22, 2014.
- [15] J. A. Ocampo Toro, J. P. Alzate Sanchez, Á. D. Chancy Villa, y S. Valencia Cardona, «Sistema autosostenible de alarma temprana en corrientes fluviales», *Rev. CINTEX*, vol. 22, n.º 2, dic. 2017.
- [16] D. Ramírez, «Integración del internet de las cosas en los procesos logísticos de máquinas dispensadoras», *Rev. CINTEX*, vol. 23, n.º 1, pp. 25-30, oct. 2018, doi: 10.33131/24222208.309.
- [17] C. Gackenheimer, *Introduction to React*. Berkeley, CA: Apress, 2015.
- [18] B. I. Ismail *et al.*, «Evaluation of Docker as Edge computing platform», en *2015 IEEE Conference on Open Systems (ICOS)*, Melaka, Malaysia, 2015, pp. 130-135, doi: 10.1109/ICOS.2015.7377291.