

Programando redes inalámbricas de sensores aplicadas al Internet de las Cosas (IoT): Un análisis sistemático

Programming wireless sensor networks to apply in the Internet of Things (IoT): systematic review

ARIAS-HURTADO, Alma^{†*}, RUIZ-IBARRA, Joel, ESPINOZA-ZALLAS, Edgar y VALENZUELA-MENDOZA, Filiberto

Universidad Estatal de Sonora

ID 1^{er} Autor: *Alma, Arias-Hurtado* / ORC ID: 0000-0002-4003-5588, arXiv Author ID: alma_isabela, CVU CONACYT ID: 824394

ID 1^{er} Coautor: *Joel, Ruiz-Ibarra* / ORC ID: 0000-0002-4932-2006, arXiv Author ID: JoelRuizIbarra, CVU CONACYT ID: 165149

ID 2^{do} Coautor: *Edgar, Espinoza-Zallas* / ORC ID: 0000-0001-7667-5782, arXiv Author ID: edgarez, CVU CONACYT ID: 620815

ID 3^{er} Coautor: *Filiberto, Valenzuela-Mendoza* / ORC ID: 0000-0002-2786-8577, arXiv Author ID: fivame, CVU CONACYT ID: 356875

Recibido Junio 30, 2018; Aceptado Septiembre 30, 2018

Resumen

En este artículo se presenta una revisión sistemática del estado del arte en programación de redes inalámbricas de sensores orientadas a la interconectividad de dispositivos del hogar o industriales, conocido el concepto como Internet de las Cosas (IoT) e Internet Industrial de las Cosas (IIoT). Se presentan las características, las ventajas, los retos de implementación, aspectos de energía, así como los parámetros de desempeño de una red inalámbrica de sensores, que funciona sobre diferentes protocolos como WiFi, IEEE802.15.4, Bluetooth, entre otros. El objetivo de este análisis es facilitar el desarrollo de un entorno de programación amigable para el usuario, que le permita validar el óptimo funcionamiento de una red bajo el esquema IoT, según las necesidades industriales y del hogar. Se proponen métodos para el manejo de la Interfaz de Programación de Aplicaciones (APIs) que proveen los fabricantes de radio transceptores, de modo que la comunicación de capas inferiores del modelo abierto de interconexión de sistemas abiertos (OSI) sea transparente para el usuario, y a su vez le dé amplia manejabilidad de la configuración de la red y sus recursos.

Desarrollo de Redes Inalámbricas de Sensores, Internet de las Cosas, Apis

Abstract

This article presents a state of the art systematic review about Wireless Sensor Networks applied to the interconnection of industrial or home appliances devices. This concept of interconnection among home appliances is well known as Internet of Things (IoT), also the interconnection among industrial devices is known as Industrial Internet of Things (IIoT). We present a review of the WSN-IoT characteristics such as its benefits, implementation challenges, energy issues, within others performance parameters for several protocols for WSN: WiFi, IEEE802.15.4, and Bluetooth. The main goal of this analysis is to facilitate the development of a WSN programming framework, easy to use and faster to implement IoT applications. This review will allow the proper validation of the WSN performance for IoT applications based on the Industrial or home appliances connections needs. We proposes an easy way to implement Application Programming Interface methods provided by radio transceivers in order to make it easy for the developer as long as there is no more need to be involved with lower layers of the OSI stack, and robust enough to use in an efficient way the network resources.

Development Wireless Sensor Networks, Internet of Things, APIs

Citación: ARIAS-HURTADO, Alma, RUIZ-IBARRA, Joel, ESPINOZA-ZALLAS, Edgar y VALENZUELA-MENDOZA, Filiberto. Programando redes inalámbricas de sensores aplicadas al Internet de las Cosas (IoT): Un análisis sistemático. Revista de Cómputo Aplicado. 2018, 2-7: 8-14.

* Correspondencia al Autor (Correo Electrónico: alma_isabela@hotmail.com)

† Investigador contribuyendo como primer autor.

Introducción

En este trabajo se lista y analiza el estado del arte en desarrollo de aplicaciones para Redes Inalámbricas de Sensores (WSN) orientadas al Internet de las Cosas (IoT). El principal objetivo es identificar nichos de investigación al detectar la tendencia del desarrollo de aplicaciones IoT, tanto para ciencia básica como para implementaciones comerciales. Se logra, a través de una metodología estructurada, describir y clasificar las propuestas IoT que se encontraron más relevantes. La clasificación de la información permite visualizar los aspectos clave para decidir en cuál de ellos hay mejores oportunidades de aportar al estado del arte.

Objetivos

- General: Identificar la tendencia del estado del arte en la programación de soluciones IoT, tanto para validar proyectos de investigación como para soluciones comerciales.
- Específicos: -Analizar el estado del arte en soluciones de aplicación IoT. –Clasificar las propuestas IoT. –Definir la tendencia del IoT en el diseño de protocolos y aplicaciones.

Metodología

Para obtener una revisión sistemática, como un estudio integrativo, observacional, retrospectivo, se siguieron los siguientes pasos:

- Seleccionar las referencias más significativas del estado del arte en el tema de desarrollo de soluciones IoT.
- Identificar y clasificar factores comunes de las propuestas IoT.
- Identificar oportunidades de investigación y desarrollo.
- Mostrar resultados.
- Concluir.

Se realizó la revisión sistemática del estado del arte en revistas de alto impacto. Como criterios de búsqueda se emplearon las palabras clave “IoT”, “framework”, “broker”, “IoT Gateway”, “IoT API”, “WSN IoT”, “WSN Online”, “WSN Web Service”. En la Figura 1 se muestra un diagrama de flujo de la metodología empleada.

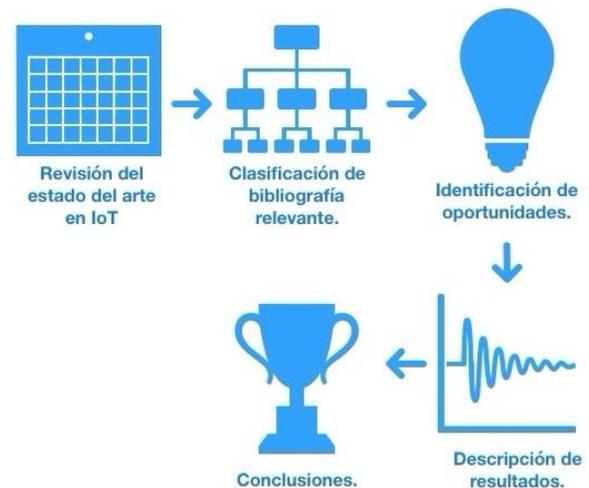


Figura 1 Metodología empleada

Fuente: Elaboración Propia

De un conjunto de alrededor de cien trabajos relacionados se seleccionaron los más significativos que se describen en este artículo. Estos trabajos ayudaron a esbozar el estado del arte en cómo el IoT logra comunicar dispositivos de diferentes lenguajes e infraestructuras a través de Application Programming Interface (APIs).

Se señalan aquellos trabajos que desarrollan marcos de trabajo para unir diferentes dispositivos en una misma nube de información bajo el concepto del IoT. Algunos de ellos están conectados y otros son fuera de línea.

La mayoría de los protocolos de comunicaciones para WSN se basan en una integración horizontal, mientras que los marcos de trabajo para IoS se basan en integraciones verticales, permitiendo interactuar a dispositivos de diferentes fabricantes y estándares.

Desarrollo

El Internet de las Cosas (IoT) se conoce como la interconexión de muebles o “appliances” que originalmente no tendrían necesidad de una conexión a internet, pero que al tenerla pueden generar grandes cantidades de información útil. Esta conexión a internet complementó consigo el concepto de comunicación Máquina a Máquina (M2M), que permite a estos muebles comunicarse entre sí para resolver problemas comunes que faciliten la vida del usuario.

Por poner un ejemplo, la lámpara de la habitación en la que se encuentra el usuario puede avisar por medio de un parpadeo que el servicio de transporte Uber ha llegado por el usuario, siendo esta una tarea diferente para la que la lámpara fue diseñada pero su conexión a internet y su comunicación con otras máquinas le permite extender sus aplicaciones.

Con las Redes Inalámbricas de Sensores (WSN) emergió la necesidad de interconectar diferentes dispositivos con un mismo protocolo. En este concepto los muebles pueden comunicarse en forma “Ad-Hoc” sin tener una infraestructura o punto de acceso viendo grandes ventajas en la coordinación y gestión de datos de la red. Sin embargo, coordinar un solo protocolo para todos los muebles es una tarea colosal.

En un ambiente cambiante como lo es la tecnología, con su evolución crece la necesidad de control y gestión de los protocolos diseñados para ella, que tarde o temprano hace que se ponga en marcha estrategias para hacer frente a esta necesidad. Se requieren estrategias de integración sobre todo cuando el desempeño de la tecnología depende de terceros o intermediarios como en el diseño de protocolos de comunicaciones por capas en el modelo OSI.

Se puede optar por diferentes integraciones de tecnología ya sea para reducir los riesgos asociados a las diferencias de diseño o bien para asegurar el mercado de negocio en el que se comercializa. El IoT trajo esta integración de la manera más amigable y funcional a las WSN. El IoT permite asumir un rol más activo en el desarrollo de tecnología, sin depender tanto de terceros siempre que tengan APIs públicas (Integración Vertical) o bien fusionarse entre diferentes protocolos “cross-layer” para reforzar una necesidad particular en una aplicación (Integración Horizontal).

La Integración Vertical beneficia la intercomunicación de diferentes fabricantes en una misma aplicación IoT, mientras que la Integración Horizontal permite reforzar aspectos particulares de la aplicación, como puede ser su seguridad o su Calidad de Servicio (QoS), haciendo de la IoT un excelente nicho de investigación, innovación y transferencias tecnológicas orientadas a negocios.

En [1] se propone una Integración Vertical de las plataformas IoT con un trabajo conocido como el proyecto Big IoT. Es un modelo de arquitectura para un ecosistema IoT que realiza cinco patrones de interoperabilidad comunes entre dispositivos IoT: Acceso multiplataformas, Acceso a dominio multiaplicación, independencia de plataforma, independencia de la escala de la plataforma y patrones de servicio de alto nivel. Donde la característica fundamental para un ecosistema IoT interoperable es el acceso multiplataformas. En la Figura 2 se muestra un bosquejo del ecosistema IoT.

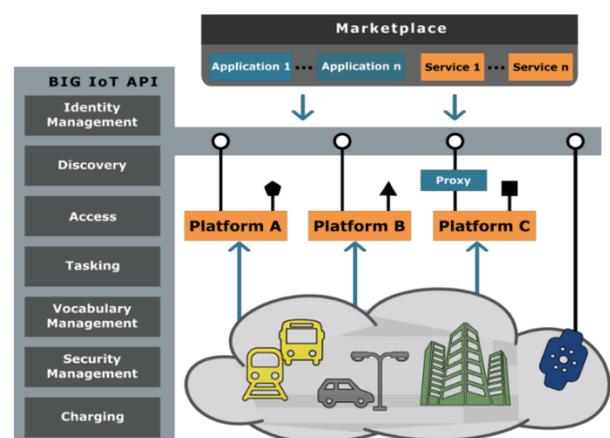


Figura 2 Ecosistema IoT, según Bröring et. al.
Fuente: *Elaboración Propia*

En [2], se presenta un desarrollo como alternativa a un módulo Ethernet para el sistema Arduino, donde lo interesante es el método para realizar la interconectividad entre un dispositivo móvil a una tarjeta arduino mediante un servidor Web, empleando PHP. Este artículo puede ser considerado una integración vertical donde se le dan capacidades de conectividad a internet a un sistema embebido, creando su propio servidor con herramientas pre-existentes. Estos mismos esfuerzos, realizados en conjunto con empresas y grupos de investigación más robustos, llevan a soluciones más redondeadas que discutiremos más adelante en este mismo trabajo.

En [3], se propone un servicio “ancla” que permita a múltiples aplicaciones IoT comunicarse entre sí y proporcionar datos al usuario mediante protocolo http en una página Web. En la Figura 3 se observa la arquitectura propuesta en [3], donde se observa un esquema basado en gerarquías donde cada proveedor está asociado a una “ancla” y los dispositivos asociados a cada proveedor pueden hacer uso del ancla que les corresponde.

Este modelo es similar al empleado por los “brokers” en el protocolo MQTT [4], basado en publicación y suscripción. Estos esquemas son eficientes para una baja huella de memoria requerida en su funcionamiento, así como bajo consumo de energía al distribuir información de un emisor a múltiples receptores de datos cortos como los generados por sensores.

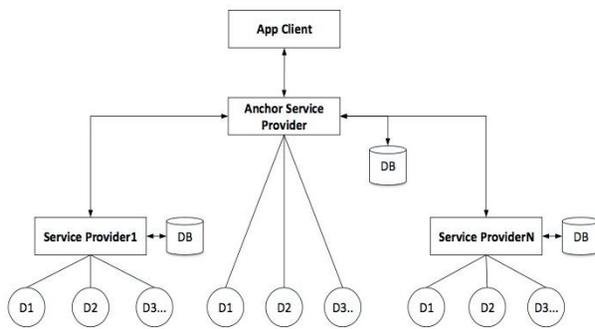


Figura 3 Arquitectura IoT basada en servicio Ancla
Fuente: Elaboración Propia

En [5] se propone un sistema IoT para cambiar el color de la iluminación pública en carreteras donde las condiciones climáticas son causantes de un porcentaje significativo de accidentes. Se plantea cambiar la temperatura (color) de la iluminación para facilitar la visibilidad en neblina o en cielo despejado, retroalimentando el sistema con microservicios como SKPlanet, AirKorea y OpenStreetMap, coordinados por una aplicación en Java Servlet.

En [6] se propone una infraestructura integrada verticalmente, tratando de mitigar las incompatibilidades de APIs entre fabricantes. Contemplan un modelo de datos basados en JSON-LD, cambiando el concepto de IoT a WoT, argumentando que es el Web de las Cosas donde se concentra la información. JSON-LD hace una propagación iterativa de datos para las interfaces WoT, generando una “super estructura” que permite operar entre sí a infraestructuras existentes.

En [7] presentan una estación meteorológica programada para RaspBerry Pi Sense Hat y un adaptador WiFi USB. Argumentan que el uso de un “Micro – Electro – Mechanical - System” (MEMS) para el procesamiento de datos en esta estación meteorológica es crucial dado que es compacto y suficientemente poderoso [8]. En [7] se emplea el servicio de ThinkSpeak como bróker o concentrador de información para detonar acciones en el entorno IoT.

ThinkSpeak te permite recolectar y analizar en MatLab la gran cantidad de datos generados, asignando una acción a un comportamiento definido de estos datos. Aun que en [7] optaron por emplear un RaspBerry Pi, ThinkSpeak funciona también sobre otras plataformas populares como Arduino, ESP8266, Aplicaciones móviles, Twitter, Twilio, entre otras. En la Figura 4 se presenta el diagrama a bloques del sistema propuesto por Govardhan.

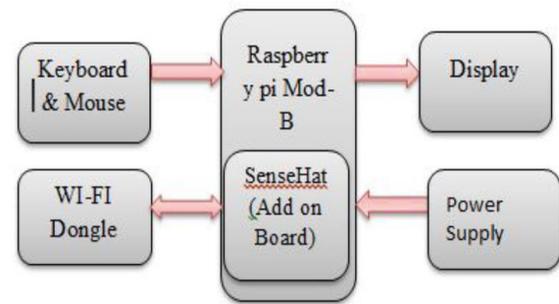


Figura 4 Diagrama a bloques del sistema propuesto en [7]
Fuente: Elaboración Propia

En [9] se presenta un servicio muy completo, de código abierto con APIs públicas para IoT. A este servicio se le llama IoTCloud, está escrito en Java con paquetes de código abierto como Apache Active MQ[10] y JBoss Netty[11]. Finalmente en [9] incluyen el análisis de una cama de pruebas geográficamente distribuida y heterogénea llamada FutureGrid, que según sus resultados, esta mesa de pruebas cumple con una buena flexibilidad para el ambiente IoT, que facilita el estudio y desarrollo de nuevas aplicaciones colaborativas, escalables, centradas en los sensores.

[12] Propone un sistema llamado Calvin. Calvin es un lenguaje y plataforma para desarrollo de aplicaciones IoT que se divide en cuatro aspectos: describir, conectar, desplegar y administrar. Calvin tiene una suite ligera de APIs que le permiten interactuar con el exterior para unir el IoT con la nube. Calvin oculta los detalles del protocolo en la capa de transporte en un intento de hacer el sistema heterogéneo, permitiendo tomar decisiones autónomas, mejorando la comunicación al evitar un esquema de comunicación directa de dispositivo a la nube y de cliente a servidor. Esta propuesta está desarrollada en software libre, en sus etapas iniciales para poder abrirse a contribuciones en su arquitectura y mejorar su rendimiento.

En [13], se aproxima a los esquemas comerciales de “broker”, donde se propone un concentrador de protocolos IoT, que tenga disponibles APIs de comunicación entre sí, orientado a sensores. Los dispositivos IoT, en esta propuesta son clasificados en ecosistemas cerrados llamados “silos”. Cada ecosistema tiene su conjunto de APIs para poder comunicarse a los dispositivos IoT del mismo. En esta propuesta se evidencia el problema de que con cada nuevo fabricante de dispositivos IoT, nace un nuevo silo que requiere poder ser interconectado con los silos pre-existentes.

En [13] se propone un estándar abierto e interoperable para resolver este problema. Entre los estándares IoT existentes, encuentran como la mejor opción el estándar de API SensorThings de el consorcio abierto geoespacial (OGC). Este estándar pre-existente considera solamente las capacidades de sensar y compartir esta información por internet. En el trabajo descrito se propone complementar a SensorThings con la capacidad de realizar tareas, en base a los datos recopilados por los sensores. En la Figura 5 se observa la secuencia de Web Service propuesta por Huang.

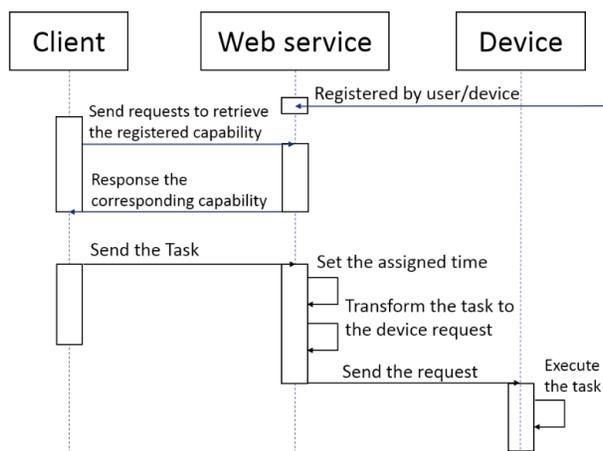


Figura 5 Diagrama de propuesta de Web Service de Huang

Fuente: *Elaboración Propia*

[14] propone un modelo llamado Epidosite, que es un sistema de programación “por demostración” que intenta resolver las limitaciones de sistemas anteriores al utilizar teléfonos inteligentes como concentradores para automatizaciones IoT.

Similar a otras publicaciones como [9, 8], en [14] se argumenta que en la programación de usuario final solamente hay compatibilidad entre dispositivos del mismo fabricante, o del mismo ecosistema que da acceso a sus APIs. Epidosite, al igual que las respectivas soluciones en [13] y [12] permiten traducir de manera sencilla para el usuario, las instrucciones disponibles en diferentes dispositivos IoT de fabricantes distintos, a través de sus APIs. Epidosite asegura tener compatibilidad entre diferentes marcas de fabricantes empleando teléfonos inteligentes como concentradores IoT.

Estos teléfonos y sus periféricos permiten controlar desde dispositivos WiFi, Bluetooth e incluso infrarrojos (Teléfonos LG G5, Galaxy S6, HTC, etc) dándole a Epidosite la extensión a dispositivos no inteligentes como televisores y aires acondicionados antiguos, sin embargo deben dirigir el haz infrarrojo hacia el dispositivo. Epidosite asegura tener interoperabilidad a través de múltiples dispositivos IoT aún de diferentes fabricantes. Finalmente Epidosite clama tener una buena usabilidad, ya que minimiza la necesidad de aprendizaje de conceptos de programación.

En [15] se aborda un tema que no se había tocado en los artículos anteriores, el consumo de energía, con la propuesta Enersave API. Este es un nicho de gran importancia donde coexistan dispositivos inalámbricos con protocolos de conectividad como en el IoT. Este artículo propone un esquema para optimizar el consumo de energía en una IoT basada en teléfonos inteligentes, similar a [14], argumentando que los principales consumidores de energía son el brillo de la pantalla, el esquema de color, la frecuencia de operación del procesador (CPU), la red 2g / 3G, mapas, localización, Bluetooth y WiFi. Se propone una API para ahorro de energía y se compara con versiones comerciales como el “DU Battery Saber” y “Battery Saber 2016” disponibles en el mercado Android. Sus comparaciones demuestran que Enersave API tiene un impacto significativo en el ahorro de energía cuando se incorpora a las aplicaciones android. Mientras que las referencias DU Battery Saver y Battery Saver 2016 ahorran de 22.2% a 40.5% de energía, respectivamente, Enersave API ahorra hasta 84.6% de la energía en batería.

En [16] se aborda la natural evolución de las WSN hacia el IoT, surgiendo la necesidad de una nube que entienda los diferentes protocolos WSN. Se maneja el concepto WSN interconectada entre diferentes fabricantes como la Web of Things (WoT). Proponen un esquema de APIs para comunicar máquinas (M2M) de diferente fabricante y protocolo. Al esquema propuesto se le llama “Mobius” y “&Cube” como Gateway entre elementos WSN sin acceso directo a internet.

[17] describe el desarrollo de un marco de trabajo para IoT basado en comunicación M2M. Se hace hincapié en que un marco de trabajo IoT debe de tener las siguientes características:

- a. Sensado: Uno de los principales retos del IoT es el unificar el mundo físico del mundo virtual.
- b. Red de redes: IoT debe unificar diferentes redes.
- c. Procesado inteligente: La inteligencia en el procesado implica liberar al humano de las grandes cantidades de información que la IoT genera y actuar conforme esa información para facilitar su trabajo.

[18] Analiza un modelo de cómputo sin servidor para aplicarlo al concepto IoT mediante el marco de trabajo Kappa, desarrollado en Calvin. Se considera combinar los recursos de la plataforma, donde se asocian las capacidades de los nodos con los requerimientos de los usuarios de la red. Estas APIs prometen una interfaz intuitiva para el desarrollo de aplicaciones.

En algunos casos es importante que la información no salga de la red local, sobretodo cuando hay aspectos de confidencialidad, por lo que este concepto es útil para el IoT aplicado a la industria (Industrial IoT-IIoT).

Finalmente en [19] se aborda el IoT desde una manera más comercial y amigable para el usuario. Se presenta un portal llamado IFTTT (<https://ifttt.com/discover>) que proviene de “If This, Then That (IFTTT)”. Esta página WEB conecta con APIs servicios de diferentes fabricantes y lenguajes de programación. Se genera una cuenta y se pueden conectar diferentes servicios con triggers o disparadores que generan una acción.

Con esta aplicación se puede desde ordenar sincronizar dropbox con instagram, hasta hacer parpadera la iluminación de la habitación cuando el vehículo de UBER ha llegado por el usuario. Estas propuestas han intentado resolver el problema de conectar a una misma aplicación diferentes dispositivos IoT y lograr que se comuniquen entre si, a pesar de haber sido programados en diferentes lenguajes y para diferentes plataformas.

Conclusiones

IoT es una corriente tecnológica muy prometedora. Se logran unificar a diferentes protocolos, fabricantes y aplicaciones con un lenguaje intuitivo y amigable para el usuario final. En este análisis sistemático se observa que es necesario ampliar la gama de APIs públicas disponibles en redes de sensores y aplicaciones móviles para mejorar el desempeño del sistema IoT y la experiencia del usuario. El IoT ha traído consigo retos en el manejo de grandes cantidades de información. La tecnología inmersa en el IoT no es necesariamente novedosa, sin embargo el concepto por sí mismo ha unificado la tecnología para generar datos a partir de sensores, procesar estos datos y ofrecer resultados en una forma ubicua, de modo que el resultado obtenido con el IoT es más productivo que la suma de las partes que lo componen.

Referencias

- [1] Bröring, Arne & Schmid, Stefan & Schindhelm, Corina-Kim & Khelil, Abdelmajid & Kabisch, Sebastian & Kramer, Denis & Phuoc, Danh & Mitic, Jelena & Anicic, Darko & Teniente, Ernest. (2017). *Enabling IoT Ecosystems through Platform Interoperability*. IEEE Software. 34. 54-61. 10.1109/MS.2017.2.
- [2] CHENOWETH, Iván R, ABRIL-GARCIA, José Humberto, MEZA-IBARRA Iván Dostoyewski. (2017). *Alternativa del módulo Shield Ethernet de Arduino usando una PC*. Revista de tecnologías de la información. Vol IV, No X, Páginas 1-7.
- [3] Hang, Lei & Kim, D.-H. (2017). *Design and Implementation of IoT Interworking of Anchor Service Provider and Mobius Platform Using RESTful API*. International Journal of Control and Automation. 10. 101-112. 10.14257/ijca.2017.10.10.09.

- [4] MQTT Community, (2018). *MQTTv3.1.1*. Extraído de <http://mqtt.org> el 25 de Julio de 2018.
- [5] Satrya, Gandeve & Reda, Haftu & Kim, Jin Woo & Daely, Philip & Shin, Soo & Chae, Seog. (2017). *IoT and Public Weather Data Based Monitoring & Control Software Development for Variable Color Temperature LED Street Lights*. International Journal on Advanced Science, Engineering and Information Technology. 7. 366. 10.18517/ijaseit.7.2.1578.
- [6] Li, Wenbin & Privat, Gilles. (2016). *Cross-Fertilizing Data through Web of Things APIs with JSON-LD*. Conferencia: European Semantic Web Conference, Workshop on "Services and Applications over Linked APIs and Data".
- [7] Govardhan, G & Hussain, Dr. S. & S A K, Jilani. (2016). *A Smart Gadget to Analyse the Weather Changes Using SenseHat Sensor and Internet of Things(IoT)*. Indian Journal of Science and Technology. 9. 10.17485/ijst/2016/v9i35/95768.
- [8] FangZ, ZhaoZ, DuL, ZhangJ, PangC, GengD. A New portable micro weather station. Proceedings of the 5th IEEE International Conference on Nano/Micro Engineered and Molecular Systems Xiamen, China, 2010, p. 379-82. ^[1]_{SEP}
- [9] Fox, Geoffrey & Kamburugamuve, Supun & D. Hartman, Ryan. (2012). *Architecture and measured characteristics of a cloud based internet of things API*. Conferencia: Collaboration Technologies and Systems (CTS), 2012 International Conference on. 6-12. 10.1109/CTS.2012.6261020.
- [10] RAO, A. 2002. *Web Services Unleashed*, garage insight <http://www.garage.com/newsletter/index.shtml>
- [11] Hapner, M., Burrige, R., Sharma, R., Fialli, J., and Stout, K. 2002. *Java Message Service*. Sun Microsystems Inc., Santa Clara, CA.
- [12] Persson, Per & Angelsmark, Ola. (2015). *Calvin – Merging Cloud and IoT*. Procedia Computer Science. 52. 10.1016/j.procs.2015.05.059.
- [13] Y. Huang, C & H. Wu, C. (2016). *Design and implement an interoperable Internet of Things application based on an extended OGC sensorthings API standard*. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLI-B4. 263-266. 10.5194/isprsarchives-XLI-B4-263-2016.
- [14] Li, Toby & Li, Yuanchun & Chen, Fanglin & Myers, Brad. (2017). *Programming IoT Devices by Demonstration Using Mobile Apps*. Conferencia: International Symposium on End User Development (IS-EUD 2017) 10.1007/978-3-319-58735-6_1.
- [15] Muharum, A.M. & Joyejob, V.T. & Hurbungs, Visham & Beeharry, Yogesh. (2017). *Enersave API: Android-Based Power-Saving Framework for Mobile Devices*. Future Computing and Informatics Journal. 2. 10.1016/j.fcij.2017.07.001.
- [16] Kim, Jaeho & Choi, S.-C & Ahn, I.-Y & Sung, N.-M & Yun, Jaeseok. (2016). *From WSN towards WoT: Open API scheme based on oneM2M platforms*. Sensors (Basel, Switzerland). 16. 10.3390/s16101645.
- [17] Bhagyashri K, Suresh V, Gita G, Amit K, Gokul T, Girishchandra, Pradeep K. (2015). *The Integrated Middleware Framework for Heterogeneous Internet of Things (IoT)*. The Integrated Middleware Framework for Heterogeneous Internet of Things (IoT). 4. 5.
- [18] Persson, Per & Angelsmark, Ola. (2017). *Kappa: serverless IoT deployment*. Conferencia: the 2nd International Workshop. DOI: 10.1145/3154847.3154853.
- [19] IFTTT Platform. (2018). *Discover IFTTT and applets*. Recuperado el 29 Julio 2018, de IFTTT Sitio web: <https://ifttt.com/discover>