

# Sistema de monitoreo remoto de temperatura y humedad utilizando dispositivos móviles e IOT

José Ramon Arratia-Zapata, Marco Aurelio Nuño-Maganda,  
Yahir Hernández-Mier, Said Polanco-Martagón

Universidad Politécnica de Victoria,  
Laboratorio de Sistemas Inteligentes,  
México

{mnuom, yhernandezm, spolanco}@upv.edu.mx

**Resumen.** En el presente trabajo se presenta un sistema para el monitoreo de temperatura y humedad, basado en un microcontrolador con un stack WiFi, que envía la información de sensores a un MQTT. Mediante una aplicación móvil desarrollada nativamente para Android, se realiza la lectura de la información alojada en la nube mediante un protocolo MQTT y se despliegan gráficas de monitoreo en tiempo-real, así como el historial de mediciones. Se probó el esquema propuesto en un ambiente de vivero simulado y los resultados muestran que la aplicación nativa desarrollada permite la lectura y visualización de los datos sin problemas. Esta aplicación podría ser de utilidad en todos los procesos que requieran el monitoreo continuo de temperatura y humedad, como en procesos de almacenamiento de productos agrícolas, comida y productos farmacéuticos.

**Palabras Clave:** Aplicación móvil, MQTT, internet de las cosas, microcontrolador, DHT22.

## Remote Temperature and Humidity Monitoring System Using Mobile Devices and IOT

**Abstract.** In the present work, a system for monitoring temperature and humidity is presented, based on a microcontroller with a WiFi stack, which sends information from sensors to an MQTT. Through a mobile application developed natively for Android, the information stored in the cloud is read using an MQTT protocol and real-time monitoring graphs are displayed, as well as the measurement history. The proposed scheme was tested in a simulated nursery environment and the results show that the developed native application allows the reading and visualization of the data without problems. This application could be useful in all processes that require continuous monitoring of temperature and humidity, such as storage processes for agricultural products, food and pharmaceuticals.

**Keywords:** Mobile application, MQTT, internet of things, microcontroller, DHT22.

## 1. Introducción

Los avances tecnológicos en el campo del internet de las cosas (IoT, por sus siglas en inglés) hacen posible controlar y monitorear en cualquier parte del mundo, ambientes y equipos. Actualmente, existen varias alternativas de domótica para realizar dicha actividad que permiten reducir la intervención del usuario incluso reducirla casi por completo. Estos avances se han acelerado en la última década y sus costos se han reducido tanto, que hoy en día es posible acceder a esta tecnología con muy poco dinero.

El IoT es un concepto que existe desde 1999 y fue propuesto por Kevin Ashton. Desde la expansión masiva del Internet en 1995, hasta el día de hoy, las conexiones no han dejado de incrementarse y globalizarse.

Al principio solo se podía acceder a Internet desde una computadora, pero hoy en día, gracias al aumento de ancho de banda de la red y la accesibilidad de nuevas tecnologías de computación más baratas y eficientes, es posible que dispositivos no computarizados o que simplemente que no fueron diseñados para que se puedan conectar a otros dispositivos, como los electrodomésticos, ahora pueden ser utilizados y monitoreados en forma remota. A esto se le conoce como IoT [1, 2].

El monitoreo remoto de temperatura y humedad es importante en diversas áreas, como la agrícola [3, 4], la farmacéutica [5, 6] o la alimentaria [7, 8]. Con la tecnología actual es posible realizar este monitoreo con diversos dispositivos y diversas tecnologías de recolección y visualización de información.

Un trabajo relacionado con el sistema propuesto en este artículo es el de [9], donde los autores proponen un sistema basado en IoT para el monitoreo de temperatura usando un sensor no invasivo. Proponen también el desarrollo de una aplicación móvil usando un protocolo de envío de datos y una base de datos no especificada.

Otro trabajo relacionado se puede encontrar en [10], donde los autores desarrollan una aplicación móvil basada en hardware y software abiertos para el monitoreo de la cadena de frío en el transporte y almacenaje de comida. En este trabajo, al igual que en el propuesto, se utiliza la plataforma en la nube de ThingSpeak para el almacenaje y visualización de la información enviada por los sensores.

En este trabajo se propone una alternativa al uso de plataformas para la visualización de información adquirida en sistemas de IoT en la forma de una aplicación nativa desarrollada en el sistema operativo Android, la cual permite visualizar la información adquirida por sensores de temperatura y humedad, en conjunto con un microcontrolador con un stack WiFi integrado, que envían información a una plataforma de almacenaje y gestión de datos en la nube.

Esta aplicación forma parte de un sistema completo de IoT para el monitoreo de la humedad y temperatura, el cual puede ser aplicado a refrigeradores, cuartos fríos, viveros, y cualquier área donde sea necesario vigilar estos dos parámetros.

Por ejemplo, en el sector de salud se puede aprovechar esta tecnología para mantener los medicamentos con una humedad y temperatura adecuadas para su conservación, evitando mermas en medicamentos. En la agricultura, se podría usar para controlar de forma continua la humedad y la temperatura en silos de almacenamiento.

El sistema propuesto está basado en un microcontrolador Arduino, el cual, con ayuda de un chip ESP8266, envía los datos adquiridos por un sensor de temperatura y humedad a un *broker*, disponible gratuitamente en internet, en donde pueden ser leídos

por dispositivos Android utilizando el protocolo Message Queuing Telemetry Transport (MQTT), y presentados en forma de gráficas a los usuarios.

## **2. Materiales y métodos**

### **2.1. Interfaz de programación de aplicaciones (API)**

En la actualidad, cuando se desarrollan aplicaciones, sin importar el contexto, se hace uso de APIs (Interfaces de Programación de Aplicaciones, por sus siglas en inglés), las cuales facilitan el proceso de diseño y desarrollo de la aplicación móvil. Una API es un software que permite que las aplicaciones se comuniquen entre sí.

Es un conjunto de funciones que permiten a las aplicaciones acceder a datos e interactuar con componentes de software externos, sistemas operativos o microservicios. Una API pasa una solicitud de un usuario a un sistema y le devuelve la respuesta del sistema. Esto permite a los desarrolladores extraer datos de diferentes formas y para diversos propósitos, dependiendo de la funcionalidad que se requiera [11].

### **2.2. Protocolo MQTT**

Es un protocolo de transporte de mensajes entre el cliente y un servidor el cual está basado en publicaciones y suscripciones, es creado por IBM con el propósito de recolectar información, es un protocolo utilizado para el intercambio de mensajes entre dispositivos que trabajan con IoT, permite la conectividad Máquina a Máquina (Machine to Machine por sus siglas en inglés M2M) e IoT. Este protocolo de mensajería ligera la cual es esquematizado en base a un servidor broker con la funcionalidad de publicación – suscripción que se emplea sobre el protocolo TCP/IP [12, 13].

### **2.3. Servidor MQTT (Bróker)**

Un servidor MQTT se encarga de administrar el protocolo, el flujo de los mensajes de cada nodo de un sensor, además del estado del servicio. En el mercado existen distintas formas de software y proveedores que implementan el servidor MQTT, como por ejemplo Google Cloud, Thingspeak o Mosquitto. En el sistema propuesto se usó ThingSpeak como el servidor MQTT, debido a que la interfaz de lectura y escritura de datos a la plataforma es amigable. ThingSpeak tiene soporte de la empresa Mathworks. ThingSpeak es un aplicación y API de IoT que permite almacenar y obtener datos de IoT mediante protocolos de internet como el HTTP y MQTT, a través de peticiones web.

### **2.4. Sensores de temperatura/humedad empleados**

En este trabajo se emplearon dos diferentes sensores de temperatura/humedad. El primero fue el DHT-11, el cual integra un sensor capacitivo que mide la humedad, y un termistor para medir la temperatura. El segundo fue el sensor DHT-22, cuyo funcionamiento es similar al DHT-11, pero tiene mayor resolución y puede ser

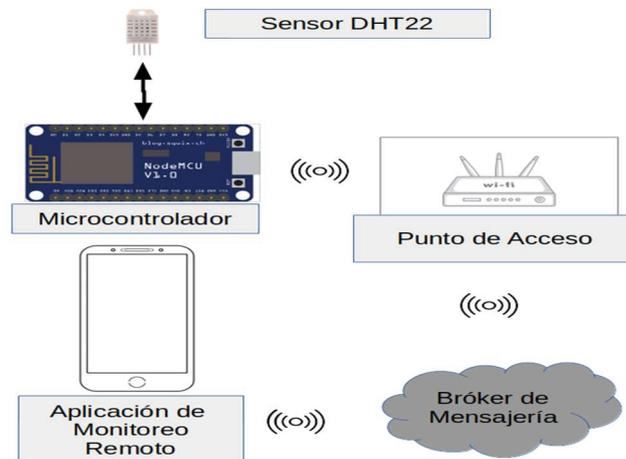


Fig. 1. Diagrama de bloques de los componentes del sistema.

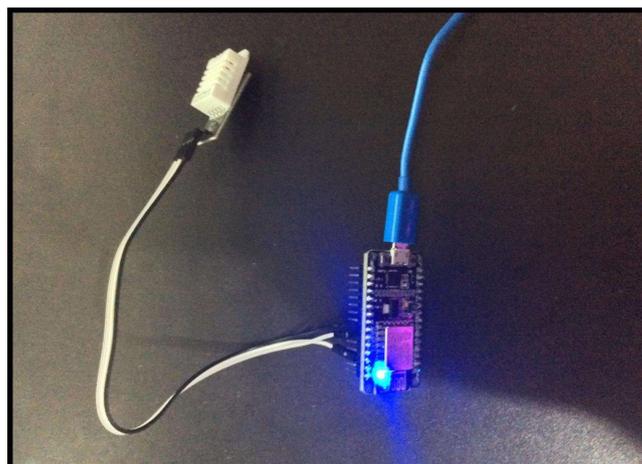


Fig. 2. Sensor DHT22 conectado al NodeCMU.

alimentado a 3.3V. El DHT-11 tiene una exactitud de  $\pm 2^{\circ}\text{C}$ , mientras que el DHT-22 cuenta con una exactitud de  $\pm 0.5^{\circ}\text{C}$ .

El sistema propuesto involucra los siguientes componentes de hardware:

- Una fuente de alimentación,
- Un sensor DHT22,
- Una tarjeta microcontroladora NodeMCU,
- Un punto de acceso a internet. Este es un requerimiento obligatorio del proyecto, ya que los datos son enviados vía WiFi.
- Un teléfono inteligente.

**Tabla 1.** Resultados de sensado de temperatura y humedad.

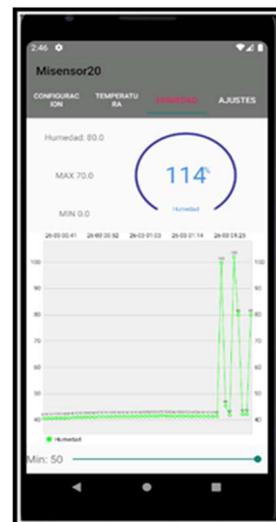
	Temperatura	Humedad
<b>Habitación Climatizada</b>	28.4	42.1
	28.4	44.1
	28.5	43.1
	28.6	42.9
	28.6	43.9
<b>Maceta de Jardín</b>	30.6	76.1
	31.6	53.0
	31.5	54.3
	31.6	53.4
	31.7	54.1



(a) Pantalla de configuración



(b) Pantalla de temperatura



(c) Pantalla de humedad

**Fig. 3.** Pantallas de la aplicación de monitoreo.

Para el desarrollo de la interfaz nativa en Android se usaron las siguientes herramientas de software:

- Android Studio.
- Arduino IDE.
- Biblioteca DHT.
- Biblioteca MPAndroidChart. Esta biblioteca permite mostrar de forma fácil y personalizable grandes conjuntos de datos en forma de gráficas [14].

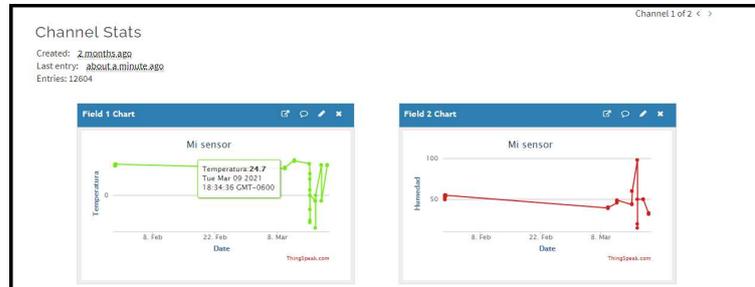


Fig. 4. Gráficas de temperatura y humedad a partir de los datos enviados por el sensor hacia el servidor.



Fig. 5. Ejemplos de alertas recibidas mediante la aplicación desarrollada.



Fig. 6. Monitorización de la humedad de una planta.

- Biblioteca CircleProgress. Esta biblioteca permite la creación de gráficas de pastel para representar un porcentaje [15].
- Retrofit. Es un cliente REST para aplicaciones Android.
- Broker de mensajería. Para el desarrollo del proyecto se evaluaron los siguientes brokers: Google Cloud, Microsoft Azure IoT Suite, IBM Watson IoT Platform, AWS IoT Platform y Thingspeak. Después de evaluar estos brokers, se optó por utilizar Thingspeak, debido a que la interfaz de lectura y escritura de datos a la plataforma en la nube es de fácil uso y el límite de datos en la versión de prueba es suficiente para las pruebas realizadas.

### **3. Sistema propuesto**

El sistema de monitoreo propuesto obtiene las lecturas de los sensores en tiempo real, y que estos datos sean visibles por un usuario final mediante una aplicación móvil. Para lograr este objetivo, un microcontrolador lee continuamente datos de los sensores y los envía continuamente a través de internet. El broker de mensajería concentra los datos de los sensores, y continuamente envía los datos hacia la aplicación móvil suscrita a los canales disponibles en el broker.

El usuario configura la aplicación móvil para establecer los umbrales de las lecturas que para el dominio particular representan valores fuera de rango. Una vez configurada la alerta, al registrarse una lectura fuera del rango, una notificación es desplegada en todos los dispositivos donde la aplicación haya sido instalada, independientemente si la aplicación está abierta o no. La aplicación permite al usuario desplegar gráficas del comportamiento de la temperatura y humedad a lo largo del tiempo. En la figura 1, se muestran los componentes del sistema previamente descritos.

#### **3.1. Sensores de temperatura/humedad empleados**

El sensor DHT22, es un sensor que requiere de una alimentación de 3.3 a 6 Volts, además de una salida de tierra. Este sensor tiene un precio bastante accesible además de ser muy eficiente con respecto a la lectura de datos y con un rango mayor de precisión que su antecesor (el DHT11).

Por otro lado, el NodeMCU está integrado por varios pines de salida de 3.3 Volts lo suficiente para mantener un buen funcionamiento en relación con el sensor y varios pines de tierra GDN, de esta forma el microcontrolador puede realizar la lectura de temperatura y humedad, sin embargo, para llevar a cabo dicha tarea es necesario conectar el sensor a uno de los pines de entrada y salida del NodeMCU. La conexión correcta entre el sensor y el NodeCMU se muestra en la figura 2.

#### **3.2. Diseño y desarrollo de aplicación móvil de monitoreo remoto**

La aplicación propuesta fue diseñada para incluir múltiples pestañas para una mejor organización. Esta aplicación emplea el protocolo MQTT (Message Queue Telemetry Transport), el cual es un protocolo diseñado específicamente para las aplicaciones IoT como la descrita en este artículo.

El protocolo MQTT funciona mediante publicaciones y suscripciones a un tema (modelo), mediante un Broker que gestiona las publicaciones y suscripciones. En la primera pestaña (mostrada en la figura 3(a)), el usuario configura los datos de la conexión al servidor para la adquisición de los datos. En esta pestaña es posible configurar la cantidad de datos que serán visualizados.

En las figuras 3(b) y 3(c), se muestran diferentes pestañas de la aplicación que incluyen la última medición y una gráfica de líneas mostrando el histórico de los datos de temperatura y humedad. Se muestran dos gráficas diferentes en cada una de las pestañas mostradas.

- La gráfica circular es utilizada para representar de forma coloreada el porcentaje entre un rango mínimo y máximo usados para el control de la temperatura o la

humedad. Para obtener esta gráfica se realiza una simple operación matemática la cual nos dará como resultado el porcentaje el cual se le enviará a un objeto de la clase ArcProgress, al igual que la gráfica de líneas debe pasar por todos los procesos desde la creación en la vista de la aplicación hasta ser referenciado, la única diferencia es que este solo requiere un valor entero el cual será el porcentaje a colorear y se podrá visualizar en su centro.

- La gráfica de líneas es desplegada usando el conjunto de datos obtenido por retrofit y la librería MQTT, y vaciando los datos hacia un objeto Dataset.

#### **4. Resultados**

Con respecto al Bróker de mensajería, se configuró la conexión del microcontrolador con el servidor, de tal forma que fue posible la visualización en el servidor de las lecturas reportadas por el sensor.

Se procedió a verificar que los datos sean visibles en el servidor y en la aplicación desarrollada. Para esto, se utilizaron las gráficas predefinidas para el muestreo de los datos en el sitio Thingspeak para el proyecto creado para la aplicación, como las mostradas en la figura 4.

La aplicación desarrollada fue probada en varios dispositivos con diferentes versiones del sistema operativo Android. La versión más antigua en la que el sistema fue probado es la versión 5.0, también se probó en las versiones 6.0, 7.0, 8.0 y la versión más reciente disponible al momento de probar el sistema (Android 11.0).

Para efectos de envío de notificaciones, el usuario puede configurar el lapso de tiempo y la frecuencia para el envío de una o varias alertas estas se manifiestan en forma de mensaje y una alerta de sonido y vibración del dispositivo, los mensajes se mostrarán en el apartado de notificaciones del dispositivo.

En la figura 5 se muestran dos ejemplos de estas notificaciones en dispositivos diferentes, donde para la temperatura se muestra la figura 5(a) y para la humedad se muestra la figura 5(b).

Finalmente, se hicieron mediciones en dos diferentes escenarios: una habitación climatizada y una maceta de jardín. Las mediciones obtenidas se muestran en la tabla 1. Estas pruebas permitieron simular las condiciones necesarias para un par de escenarios más realistas: monitorear de manera remota las condiciones de humedad y temperatura de un invernadero, y monitorear la temperatura de refrigeradores con aplicación en agricultura y salud.

En la figura 6, se muestra el sensor montado en una maceta, de donde se obtuvieron las lecturas reportadas en la tabla.

#### **5. Conclusiones y trabajo futuro**

Con base a las pruebas realizadas, fue posible corroborar que el sistema realiza las funciones para las cuales fue diseñado: la medición de temperatura y de humedad. Para validar el sistema propuesto, se verificaron dos posibles escenarios: el monitoreo remoto de refrigeradores y la detección de los niveles de humedad en un huerto remoto.

La aplicación móvil es capaz de mostrar información que alerta al usuario acerca de anomalías en las lecturas, incluyendo la configuración de la aplicación para desplegar al usuario notificaciones informando de estas anomalías.

Además, la aplicación despliega el histórico de las mediciones obtenidas en forma gráfica, lo cual permite al encargado del monitoreo tener un panorama más amplio de las variaciones en las mediciones.

Algunas mejoras que pueden realizarse consisten en el aumento del número de sensores, mejorar la interfaz gráfica y optimizar la conectividad y la obtención de datos de los sensores.

## Referencias

1. Uribe-Castro, A.: Análisis del nivel de seguridad presente en los dispositivos que componen el internet de las cosas (2020)
2. Xia, F., Yang, L. T., Wang, L., Vinel, A.: Internet of things. *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1101–1102 (2012)
3. Karthikeyan, P. R., Chandrasekaran, G., Kumar, N. S., Sengottaiyan, E., Mani, P., Kalavathi, D. T., Gowrishankar, V.: Iot based moisture control and temperature monitoring in smart farming. *Journal of Physics: Conference Series*, no. 1, pp. 1–7 (2021)
4. Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., Hajjaji, S. E.: An intelligent IoT-based food quality monitoring approach using low-cost sensors. *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32 (2021)
5. Widjaya, D.: Enhancing vaccine refrigerator temperature reporting system using IoT technology. *Suranaree Journal of Science and Technology*, vol. 25, no. 3, pp. 225–234 (2018)
6. Roduit, B., Luyet, C. A., Hartmann, M., Folly, P., Sarbach, A., Dejeaifve, A., Dobson, R., Schroeter, N., Vorlet, O., Dabros, M., Baltensperger, R.: Continuous monitoring of shelf lives of materials by application of data loggers with implemented kinetic parameters. *Molecules*, vol. 24, no. 12 (2019)
7. Popa, A., Hnatiuc, M., Paun, M., Geman, O., Hemanth, D. J., Dorcea, D., Son, L. H., Ghita, S.: An intelligent IoT-based food quality monitoring approach using low-cost sensors. *Symmetry*, vol. 11, no. 1, pp. 18 (2019)
8. Konur, S., Lan, Y., Thakker, D., Morkyani, G., Polovina, N., Sharp, J.: Towards design and implementation of industry 4.0 for food manufacturing. *Neural Computing and Applications* (2021)
9. Ramalho, J. F.C. B., Carlos, L. D., André, P. S., Ferreira, R. A. S.: Moptical sensing for the internet of things: A smartphone-controlled platform for temperature monitoring. *Advanced Photonics Research*, vol. 2, no. 6 (2021)
10. Ramírez-Faz, J., Fernández-Ahumada, L. M., Fernández-Ahumada, E., López-Luque, R.: Monitoring of temperature in retail refrigerated cabinets applying IoT over open-source hardware and software. *Sensors*, vol. 20, no. 3 (2020)
11. Bierhoff, K.: Api protocol compliance in object-oriented software. PhD thesis, Carnegie Mellon University, USA AAI3370353 (2009)
12. Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., Al-Hatmi, R.: Internet of things: Survey and open issues of MQTT protocol. In: *International Conference on Engineering MIS (ICEMIS)*, pp. 1–6 (2017)
13. Soni, D., Makwana, A.: A survey on MQTT: A protocol of internet of things (IoT). In: *International Conference On Telecommunication, Power Analysis And Computing Techniques ICTPACT'17*, vol. 20 (2017)
14. Jahoda, P.: MPAndroidChart. <https://github.com/PhilJay/MPAndroidChart> (2019)

*José Ramon Arratia-Zapata, Marco Aurelio Nuño-Maganda, Yahir Hernández-Mier, et al.*

15. LianHaiLiang, M.: Circleprogress. <https://github.com/MichaelLianHaiLiang/CircleProgress> (2018)