

# Sistema de monitoreo remoto de refrigeradores utilizando dispositivos móviles e IOT

**Abstract**—En el presente trabajo se describen las actividades realizadas para el desarrollo de proyecto de esta conformado por un periodo de 14 semanas, para el Dr. Marco Aurelio Nuño Maganda en la Universidad Politécnica de Victoria, localizada en Ciudad Victoria Tamaulipas. Las actividades descritas en este documento se desarrollaron con la finalidad de construir un sistema para la monitorización de la temperatura y la humedad, consiste en realizar una serie de investigación, diseño y desarrollo para crear un sistema de monitorización el cual consta del uso de internet de las cosas, así como una aplicación móvil, la cual pueda operar en base a el uso del protocolo MQTT, así mismo ofrecer una la capacidad de graficar la monitorización. En el presente documento se describe cada una de las actividades realizadas, describiendo la investigación realizada como el desarrollo llevado acabo y pruebas realizadas en diferentes fases del proyecto.

**Index Terms**—MQTT, Internet de las cosas, Microcontrolador, DHT22, Grafificación móvil.

## I. INTRODUCTION

La necesidad de tener más información a nuestro alcance con la ayuda de los avances tecnológicos es posible controlar y monitorear equipos electrónicos cotidianos o no cotidianos que requieran ser intervenidos por un usuario, actualmente existen varias alternativas de domótica para realizar dicha actividad que reduzcan la intervención del usuario incluso reducirla casi por completo, actualmente con la ayuda del Internet de las cosas (Internet of things por sus siglas en inglés IoT) se puede conseguir realizar un gran número de actividades de interés sin la necesidad de tener demasiada infraestructura.

Actualmente la tecnología ha aumentado la productividad y con ello la eficacia de llevar a cabo los diferentes procesos de calidad de los distintos sectores con el fin de minimizar errores y mermas en dado caso, de esta manera una de las variables más sobresalientes a la hora de buscar una optimización en los diferentes sectores como lo son salud, fábricas, empresas del sector gastronómico e incluso en el sector de agricultura depende de estándares constantes de calidad los cuales radican de llevar a cabo un monitoreo de temperatura y humedad. Por lo tanto, se busca tener una alternativa a una aplicación nativa en sistema operativo Android la cual permita tener un censado obtenido de un microcontrolador el cual integre un censor junto con un módulo de Internet Inalámbrico (Wireless Fidelity por sus siglas en inglés Wi-Fi), su usabilidad dependerá del contexto del sector al que se aplique y la económica del sector, de igual forma la necesidad de tener el control, monitoreo de la humedad y temperatura, aplicándolo en una variedad de medios y herramientas como pudiera ser, refrigeradores, cuartos fríos, refrigeración general de áreas etcétera. Mientras por otro lado en el sector de salud se puede aprovechar esta tecnología para mantener los medicamentos en un ambiente

apto en cuestión de humedad o temperatura para su conservación, evitando mermas en medicamentos y manteniendo utensilios en un estado apto. Finalmente en la agricultura su uso más importante se implementaría para tener un estándar alto del control de la humedad en la siembra, además de la temperatura del ambiente en la que se encuentra la cosecha.

Diseñar y desarrollar un sistema de monitoreo de temperatura y humedad utilizando sensores y microcontroladores, para ser más preciso un Arduino. El cual, con ayuda del chip esp8266 Wi-Fi, sea capaz de enviar los datos a un broker provisto por un tercero de forma gratuita para que posteriormente los datos sean enviados a dispositivos Android utilizando el protocolo Message Queuing Telemetry Transport (MQTT) en el cual serán modelados en gráficas amigables para el usuario.

## II. METODOLOGÍA

### A. Internet de las cosas (IOT)

Internet de las cosas (o también IoT por su nombre en inglés) es un concepto que existe desde 1999 propuesto por Kevin Ashton. Con la comercialización de Internet en 1995 hasta el día de hoy, las conexiones no han dejado de incrementar, se globalizó. Siendo que al principio solo se podía acceder a Internet desde un computador, 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 nuestros dispositivos no computarizados o que simplemente que no fueron diseñados para que se puedan conectar a otros dispositivos (Electrodomésticos, Objetos cotidianos) ahora pueden ser utilizados de forma remota, obteniendo su información y enviándola a otro en donde éste puede mandar una orden programada o simplemente notificar del estado del equipo, a esto se le conoce como IoT. [1][2]

### B. Interfaz de programación de aplicaciones (API)

En la actualidad a la hora de desarrollar una aplicación sin especificar el contexto esta puede implementar funciones de terceros por medios del uso de API's, esto con el fin de obtener como resultado en el contexto que sé este diseñando la aplicación deseada una variedad de herramientas que facilitaran la integración de la programación que requiera la aplicación deseada. De esta manera es importante destacar que una API es una interfaz informática que define las integraciones entre varios software, es decir que identifica los tipos de funciones que puede realizar, ya sea de forma saliente o de entrada de datos, permitiendo a los desarrolladores extraerlos de diferentes formas y propósitos dependiendo de la funcionalidad que requieran. [3]

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 Maquina a Maquina (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.[4]

### C. 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 Maquina a Maquina (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.[4]

1) *Servidor MQTT (broker)*: 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, Google Cloud, thingspeak, Mosquitto.

### D. DHT11

DHT11 es la integración de un sensor capacitivo el cual se encarga de medir la humedad, también integra un termistor para medir el valor de temperatura, estos valores son enviados como señal de salida digital en el pin de salida. Las características del sensor son:

- Alimentación de 3 Vdc a 5Vdc.
- Rango de temperatura de 0° a 50 °C.
- Precisión de temperatura de 2 °C.
- Rango de humedad de 20% a 90%.
- Precisión de humedad del 4%.
- Tiempo de respuesta de un segundo.

### E. DHT22

DHT22 al igual que el sensor DHT11 es un sensor de temperatura y humedad, la diferencia son sus valores ya que el primero cuenta con mejores precisiones de sus lecturas. Este cuenta con las siguientes características:

- Alimentación de 3 Vdc a 5Vdc.
- Rango de temperatura de -40° a 125 °C.
- Precisión de temperatura de 0.5 °C.
- Rango de humedad de 0% a 100%.
- Precisión de humedad del 2%.
- Tiempo de respuesta de dos segundos.

### F. Thingspeak

ThingSpeack es una aplicación y API de IoT la cual nos permite almacenar y obtener datos de cosas mediante protocolos de internet como el HTTP y MQTT a través de peticiones web. ThingSpeack fue lanzado en 2010 como un servicio de apoyo en aplicaciones de IoT. ThingSpeack tiene soporte del software MATLAB permitiendo a los usuarios analizar y visualizar datos.

## III. SISTEMA PROPUESTO

### A. Investigación con respecto al esquema de NodeMCU y sensor DHT22

El principal objetivo del proyecto presente es el desarrollo de un sensor de lectura de temperatura y humedad, por lo que uno de los puntos clave de investigación es respecto al sensor DHT22.

De esta manera se indago el sensor DHT22, el cual es un sensor que requiere de una alimentación de 3.3 Volts. a 6V además de una salida de tierra o masa , cabe mencionar que el sensor tiene un precio bastante accesible además de tener en cuenta como consideración primordial que es muy eficiente para su funcionamiento respecto a la lectura de datos y con un rango mayor de precisión que su antecesor. 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 GND, 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.

Cabe destacar que el propósito de su investigación parte desde el objetivo principal del proyecto a presentar, ya que su análisis y la evidencia de su funcionamiento muestra la perspectiva a la cual se busca lograr físicamente del diseño y desarrollo del sensor.

Posteriormente se realizó una investigación para el desarrollo del código programado en el lenguaje C++ el cual permite generar una lectura directa en el pin que está conectado entre el sensor DHT22 y al NodeMCU.

En este mismo sentido el código se desarrolló mediante el uso de la librería DHT con el fin de realizar la programación de la lectura de los datos del sensor, una vez iniciada dicha programación con el pin en el cual se conectó al NodeMCU nos permite acceder a diferentes métodos los cuales nos proporcionan la lectura de datos dependiendo del método elegido, de igual manera pueden ser almacenados en variables para su uso en un futuro.

Una vez realizado la tarea anteriormente mencionada respecto a la conexión entre los pines del sensor y los del NodeMCU, por consiguiente para realizar de manera adecuada la lectura de los datos es importante destacar que el primer paso es verificar que el sensor se encuentre establecido, de lo contrario de no ser así este enviará una asignación vacía a una de las variables previamente elegidas. No obstante el presente proyecto se encuentra en el proceso

actual de lo anteriormente mencionado con la verificación de las especificaciones descritas.

#### *B. Implementación de servidor*

Para el desarrollo del presente proyecto es importante destacar que existen diferentes alternativas de servidor bróker entre ellas están las siguientes; Google Cloud, Microsoft Azure IoT Suite, IBM Watson IoT Platform, AWS IoT Platform y Thingspeak, cada una de estas plataformas cuentan con características a su favor y en contra ya que en algunos casos es muy compleja, por mencionar algunas como por ejemplo, no tiene soporte, es limitada entre otras, de esta manera se eligió la plataforma Thingspeak la cual es una plataforma práctica de usar, eficiente y cuenta con menos características que pueda afectar significativamente al proyecto, sin embargo hablando en características a su favor es que tiene una carga de datos limitada para API la cual permite tener la posibilidad de extenderse adquiriendo algún paquete de pago del sitio.

#### *C. Diseño y desarrollo de aplicación móvil y creación gráficos*

Generalmente cuando se tiene un sistema, el cual consta de la monitorización en algún contexto deseado, este debe ser monitoreado con un tiempo de respuesta de menor tiempo, es decir que la necesidad de tener una actividad basada en el concepto de “Tiempo Real”, de esta manera las actividades debe estar lo más cerca a lo que está sucediendo en la vida real.

En este mismo sentido el tener actividades las cuales estén hechas con la intención de tener la máxima precisión pudiera ser una tarea compleja, puesto que se pudiera pensar en tratar de utilizar un proceso el cual se encargue de estar actualizando datos, sin embargo dicho proceso está limitado al tiempo el cual se le ha establecido, ya que por este mismo motivo lo que se utilizaría, terminaría siendo demasiado recurso y solo se consideraría en ciertos momentos en donde la aplicación actualizaría casi al mismo tiempo en el que él se enviaron los datos del sensor. Por otro lado se tiene una alternativa más, con la cual ayuda del protocolo MQTT se puede mantener una conexión mediante el método de máquina a máquina el cual permitirá que con ayuda del protocolo TCP/IP la seguridad se realizará mediante la autenticación de toda la comunicación, de esta manera durante la fase de comunicación un cliente puede realizar operaciones de publicación, suscripción y cancelar suscripción. En relación a lo anteriormente mencionado gracias a la forma establecida de comunicación se puede saber cuándo el sensor ha enviado un nuevo valor sin necesidad de tener que utilizar varios procesos los cuales dependan de una previa programación; Imaginemos que se tiene dos sensores los cuales son publicadores, estos envían información al servidor bróker el cual este encargado de enviar a todos los suscriptores, y por consiguiente estos suscriptores reciben el mensaje retransmitido sin necesidad de que el suscriptor realizará la petición.

En el presente proyecto el desarrollo de dicha aplicación se

requiere del uso de TCP/IP como un proceso único de una sola ejecución y de MQTT como un proceso continuo, como se ha explicado con anterioridad se requiere una conexión en tiempo real por lo que se utilizará un proceso para solicitar un histórico al momento de ejecutar la aplicación, seguido de otro este estará en la espera de nuevos datos. Dichos datos serán modelados en gráficas para ofrecer una mejor apreciación al usuario.

Mientras tanto el primer proceso se realizará utilizando la librería Retrofit, para que en el segundo proceso se lleve a cabo la implementación de una conexión con ayuda de la librería de eclipse mqttv3.

Dicho lo anterior Retrofit se utilizará, puesto como se mencionó anteriormente es una librería para hacer llamadas en red y obtener un resultado, el cual está hecho en base a la labor manual de hacer consultas en red HttpClient para posteriormente procesar la respuesta JSON obtenida y poder construir objetos y utilizarlos en dicho proyecto. De esta manera la instalación de retrofit es muy simple de realizar, ya que consta de tener un proyecto en Android Studios, en el cual se debe incluir en la implementación de retrofit en el sistema de automatización de construcción, dicho esto actualmente es Gradle; de esta forma retrofit requiere tener una interfaz Java la cual usará para convertir su API HTTP en dicha interfaz.

La conexión se establecerá con ayuda de MqttClient ya que es una librería diseñada por Eclipse con funciones útiles para realizar dicha conexión deseada.

Como ya se ha explicado en apartados anteriores es necesario tener implementada la librería, para comenzar a utilizar dicha librería es necesario crear una variable de la clase MqttClient la cual nos pide tres parámetros en su constructor para la creación, los cuales son; el contexto en el cual se está ejecutando, como segundo es la dirección del servidor broker y por último nos pide ingresar un cliente el cual puede ser cualquier cadena de texto, sin embargo para fines dinámicos este se genera gracias al método “generateClientId” proporcionados por la misma librería.

Dicho lo anterior ahora todo está listo para intentar una comunicación o conexión, si la conexión se realiza con éxito se puede llevar a cabo la preparación para solicitar una suscripción al canal, de esta manera esto se puede realizar con ayuda de la función subscribe, la cual nos pide dos parámetros; el primero es un tema, es decir que el tema se refiere a una estructura de texto la cual el servidor espera para determinar si se puede suscribir o no y en el segundo es un canal en el cual se está enviando los datos desde el sensor.

#### *D. Investigación y desarrollo de para la graficación en dispositivos móviles*

La modelación de gráficas tiene con finalidad ofrecerle al usuario una interacción simple con los datos que se obtuvieron, en este proyecto se modeló dos gráficas para la temperatura y otras dos para la humedad, un tipo de gráfica es la representación del porcentaje del último valor obtenido, la segunda gráfica es una gráfica de líneas la cual representa

al historial de los valores que dicha grafica debe representar. Se utilizaron las librerías MPAndroidChart y CircleProgress para la creación de las gráficas, dichas librerías deben estar implementadas en el proyecto.

MPAndroidChart permite mostrar de forma fácil y personalizable grandes conjuntos de datos en forma de gráficas.

CircleProgress permite la creación de graficas dibujadas en circulo para representar un porcentaje coloreando parte representativa en dicho dibujo circular.

Lo primero que debemos realizar es crear los recursos gráficos en la vista xml. El conjunto de datos obtenido por retrofit y MQTT será graficado, para poder graficar es necesario pasar los datos a una instancia de DataSet por lo que la librería nos ofrece la clase LineDataSet, dicha clase requiere de una lista de datos de entra, dichos datos es un compuesto del valor X, Y, ya que solo queremos representar los datos de temperatura solo será necesario crear un dataset.

Primero se debe contar con una vista para poder hacer uso del elemento que conforman esa vista en la cual debe de ir la gráfica la cual será necesaria referenciar en la clase en la cual será utilizada.

Posteriormente el conjunto de datos obtenido por retrofit y MQTT será graficado, para poder graficar es necesario pasar los datos a una instancia de DataSet por lo que la librería nos ofrece la clase LineDataSet, dicha clase requiere de una lista de datos de entra, de esta manera los datos es un compuesto del valor X, Y, ya que solo se quiere representar los datos de temperatura, por lo que solo será necesario crear un dataset, mismo daaset será asignado a la gráfica a la cual representará los valores.

Para la graficas circulares las cuales se usaran para representar de forma coloreada el porcentaje entre un rango mínimo y máximo usados para el control de la temperatura o la humedad se realiza una simple operación matemática la cual nos dará como resultado el porcentaje el cual se le enviara 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.

#### IV. RESULTADOS

Dado que se desarrolló una aplicación móvil para dispositivos Android, requiere de las siguientes especificaciones de dispositivo para funcionar.

Hardware	Software
Conexión a internet	Versión de Android API 21 a 30

TABLE I  
ESPECIFICACIONES DEL HARDWARE Y SOFTWARE

#### A. Pruebas y resultados

Con base a los redactado anteriormente obtenemos como resultado la conexión correcta entre el sensor y el NodeCMU,

en la imagen 1 se muestra en operación.

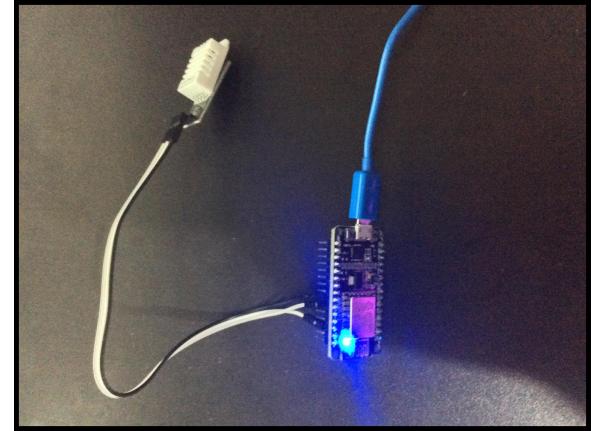


Fig. 1. Sensor DHT22 conectado a NodeCMU

Una vez que ya obtenemos nuestro correcto funcionamiento de nuestro hardware a utilizar y se realizó la programación correspondiente obtenemos los resultados esperados los cuales se muestran en la siguiente imagen, en la cual podemos observar salidas de pantalla las cuales se establecen con respecto se hacen las validaciones correspondientes para su funcionamiento hasta llegar a la lectura y el envío de los datos del ambiente obtenidos por el sensor.

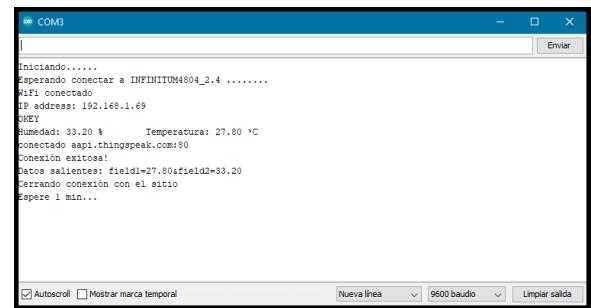


Fig. 2. Resultado al obtener una lectura del sensor con el NodeCMU y el envío al servidor

Como resultado final de este apartado se puede utilizar graficas predefinidas para el muestreo de los datos en el sitio Thingspeak.



Fig. 3. Graficas de temperatura y humedad generadas a partir de los datos enviados por el sensor en el servidor

Una de las tareas principales de la aplicación desarrollada es establecerle la dirección del servidor y datos de acceso, dado

lo anterior y redactado en el subtema referenciado, tenemos como resultado una pantalla de inicio la cual se encarga de la configuración del de la aplicación.



Fig. 4. Pantalla de configuración

Como resultado el funcionamiento de las gráficas descritas en su correspondiente subsección, como se puede observar en la imagen 6 se cuenta con una gráfica la cual corresponde a la última medición y otra grafica de líneas la cual grafica el histórico de los datos, por último se puede obtener el control de la cantidad de datos a graficar con respecto a un máximo de datos basado en la cantidad de datos históricos se cuentan en el dispositivos.



Fig. 5. Pantalla de temperatura

Como resultado dado si el usuario estuvo interesado en recibir 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 el cual por lo general es en la parte superior del sistema operativo Android.

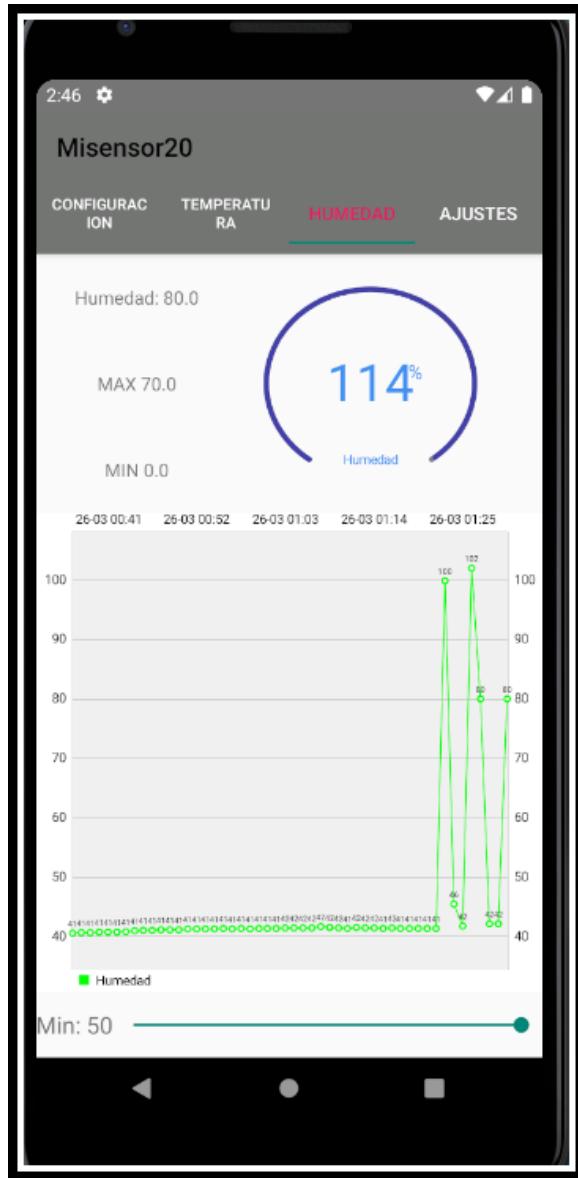


Fig. 6. Pantalla de humedad

Se puede obtener las siguientes mediciones:

Temperatura	Humedad
Habitación climatizada	
28.4	42.1
28.4	44.1
28.5	43.1
28.6	42.9
28.6	43.9
Maceta de jardín	
30.6	76.1
31.6	53
31.5	54.3
31.6	53.4
31.7	54.1

TABLE II

TABLA DE RESULTADOS DEL MODELO DE PRUEBA

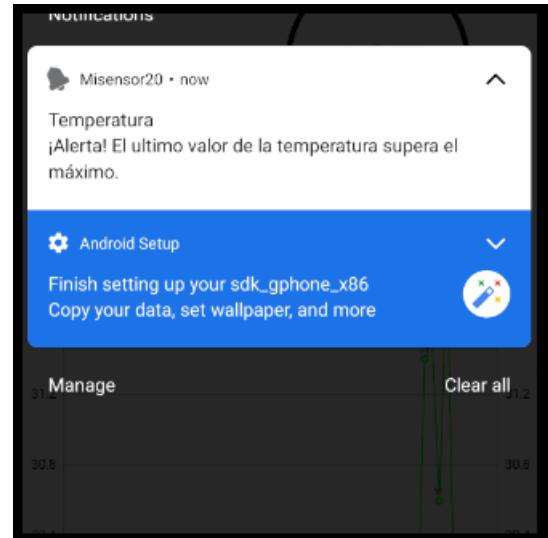


Fig. 7. Alerta temperatura

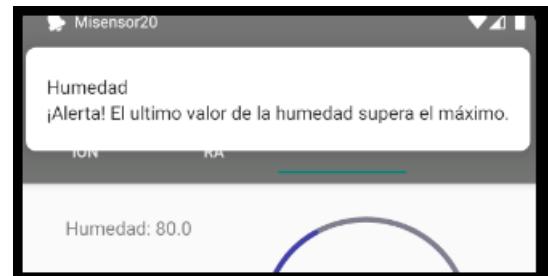


Fig. 8. Alerta humedad

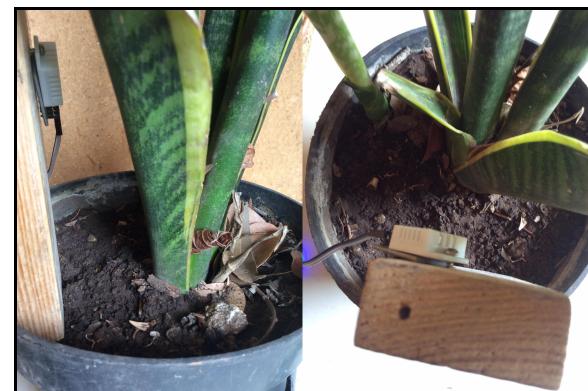


Fig. 9. Monitorización de planta

## V. CONCLUSIONES Y TRABAJO FUTURO

En conclusión, el sistema para la medición de la temperatura y humedad funciona muy bien. Este sistema puede adaptarse y podría crear una red de nodos de sensores.

En cuanto al funcionamiento de la aplicación, resultó cumplir su objetivo de visualizar los datos en gráficas, alertar al usuario.

Una de las complicaciones fue al momento de realizar una suscripción sin embargo se logró solucionar. El proyecto

cumple con cada una de las secciones y funcionalidades que se acordaron, sin embargo, puede existir fallas en la validación en la interfaz gráfica.

Aún hay cosas por mejorar en la monitorización de variables físicas utilizando IoT, sin embargo, entran nuevas oportunidades de estudio en este proyecto.

#### *A. Trabajo futuro*

- 1) **Aumentar la cantidad de sensores:** Como primera mejora se puede realizar principalmente una modificación para almacenar múltiples lecturas de diferentes sensores en un mismo servidor, dado que por el momento solo se realizó este proyecto con un sensor, este no siempre es la mejor opción ya que en el mercado existe una gama de sensores para diferentes situaciones ambientales por lo que intento diseñar de forma que en un futuro se pueda modificar ligeramente para funcionar con más sensores.
- 2) **Interfaz Grafica :** Considerando en la experiencia de usuario, esta se puede mejorar con mejoras en el diseño en la aplicación, además de obtener una aplicación mas robusta y confiable.
- 3) **Optimización de conectividad y obtención de datos :** Actualmente el sistema funciona de forma correcta, pero existen una serie de aspectos en los cuales se pudiera mejorar obteniendo una mejor conexión y obtención de datos.

#### REFERENCES

- [1] A. Uribe Castro *et al.*, “Análisis del nivel de seguridad presente en los dispositivos que componen el internet de las cosas,” *Vale Del Cauca*, 2019.
- [2] F. Xia, L. T. Yang, L. Wang, and A. Vinel, “Internet of things,” *International journal of communication systems*, vol. 25, no. 9, p. 1101, 2012.
- [3] “Cumplimiento del protocolo api en software orientado a objetos,” Tech. Rep.
- [4] D. Soni and A. Makwana, “A survey on mqtt: a protocol of internet of things (iot),” in *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*, vol. 20, 2017.