

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**IMPLEMENTACIÓN DE PROTOTIPOS DE SISTEMAS DE ALERTA  
BASADOS EN COMUNICACIÓN INALÁMBRICA**

**IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA  
POR DETECCIÓN DE CAMBIOS DE TEMPERATURA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**RICARDO FERNANDO SIMBAÑA CASCUERAN**

**DIRECTOR: FANNY PAULINA FLORES ESTÉVEZ**

**DMQ, agosto 2022**

## **CERTIFICACIONES**

Yo, RICARDO FERNANDO SIMBAÑA CASCUERAN declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



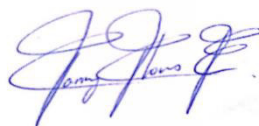
---

**RICARDO FERNANDO SIMBAÑA CASCUERAN**

**ricardo.simbana@epn.edu.ec**

**ricardos.fernandoc@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por RICARDO FERNANDO SIMBAÑA CASCUERAN, bajo mi supervisión.



---

**FANNY PAULINA FLORES ESTÉVEZ**

**fanny.flores@epn.edu.ec**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



---

**RICARDO FERNANDO SIMBAÑA CASCUERAN**

**ricardo.simbana@epn.edu.ec**

**ricardos.fernandoc@gmail.com**

## **DEDICATORIA**

A mi familia y en especial a mis padres Hernando y Blanca quienes, con su amor, esfuerzo, paciencia, apoyo y sacrificio me han permitido continuar con mi formación y alcanzar una meta más en mi vida.

Ricardo.

## **AGRADECIMIENTO**

Este proyecto no hubiese sido posible sin la ayuda de muchas personas. A mi directora de tesis, Ing. Fanny Flores quien realizó numerosas revisiones y me guió durante todo el proceso, su conocimiento y experiencia me han permitido desarrollar este trabajo.

A la Escuela Politécnica Nacional por otorgarme la oportunidad de formarme en sus aulas, brindándome el conocimiento necesario para completar este proyecto. Finalmente, gracias a mis amigos, Sebastián, Edison, Santiago, Stalin quienes me acompañaron en esta maravillosa etapa de mi vida, siempre podrán contar con mi apoyo y amistad.

Ricardo.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN.....	VII
<i>ABSTRACT</i> .....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
Ambientes controlados de temperatura .....	3
Redes inalámbricas de área local .....	4
Módulo Wi-Fi ESP8266.....	4
Sensor DHT 11 .....	5
Arduino .....	6
ThingSpeak .....	6
Blynk.....	7
2 METODOLOGÍA.....	7
3 RESULTADOS .....	9
3.1 Establecimiento de requerimientos del prototipo.....	9
3.2 Definición de componentes de hardware y software.....	10
3.3 Diseño del prototipo .....	16
Proceso de comunicación .....	16
Rangos de funcionamiento .....	17
Circuito Electrónico.....	17

Diseño 3D del prototipo .....	19
Lugar de implementación.....	20
Diagrama de flujo.....	20
Alimentación .....	22
3.4 Implementación del prototipo .....	22
Elaboración de la placa electrónica.....	22
Alimentación del prototipo.....	25
Elaboración del módulo protector .....	25
Configuración de plataformas IoT .....	26
Programación del NodeMCU-ESP8266 .....	33
Implementación del prototipo en el invernadero.....	36
3.5 Pruebas de funcionamiento .....	37
4 CONCLUSIONES.....	41
5 RECOMENDACIONES.....	42
6 REFERENCIAS BIBLIOGRÁFICAS.....	43
7 ANEXOS.....	46
ANEXO I: Certificado de Originalidad .....	i
ANEXO II: Enlaces .....	ii
ANEXO III: Códigos Fuente .....	iii

## RESUMEN

Este proyecto de fin de grado presenta la implementación de un sistema de alerta por detección de cambios de temperatura, el cual capta las variaciones de temperatura y humedad de un orquideario. Se notifica al usuario vía correo electrónico y aplicativo móvil al detectar temperaturas inadecuadas.

El desarrollo del proyecto consiste en una serie de procesos claves en el diseño, fabricación e implementación del sistema.

Inicialmente, se realiza un análisis e investigación de componentes basados en los requerimientos del sistema.

La selección de una metodología de trabajo permitirá el desarrollo de los objetivos del proyecto de una manera secuencial.

Para llevar a cabo el desarrollo del proyecto se considera emplear un sensor de temperatura y un microcontrolador con un módulo Wi-Fi integrado, compatibles con las plataformas IoT para visualización de datos; además, se desarrollará la placa del circuito electrónico y un módulo de protección de componentes.

Se comprobará el funcionamiento del sistema, y se obtendrá las conclusiones y recomendaciones para fortalecer y mejorar el desarrollo del proyecto. Finalmente, todas las fuentes de investigación y evidencia del desarrollo del proyecto son presentadas en las secciones bibliografía y anexos.

**PALABRAS CLAVE:** sensor de temperatura, orquideario, módulo Wi-Fi, plataformas IoT, correo electrónico.



## **ABSTRACT**

*This dissertation work presents the implementation of an alert system for detecting changes in temperature, which captures the variations in temperature and humidity of an orchid garden. The user is notified by email and mobile application when inadequate temperatures are detected.*

*The development of the project consists of a series of key processes in the design, building and implementation of the system.*

*Initially, an analysis and investigation of components based on the system requirements are carried out.*

*The selection of a work methodology will allow the development of the project objectives in a sequential manner.*

*To implement the project, it is considered to use a temperature sensor and a microcontroller with an integrated Wi-Fi module, compatible with IoT platforms for data visualization. In addition, the electronic circuit board and a component protection module will be developed.*

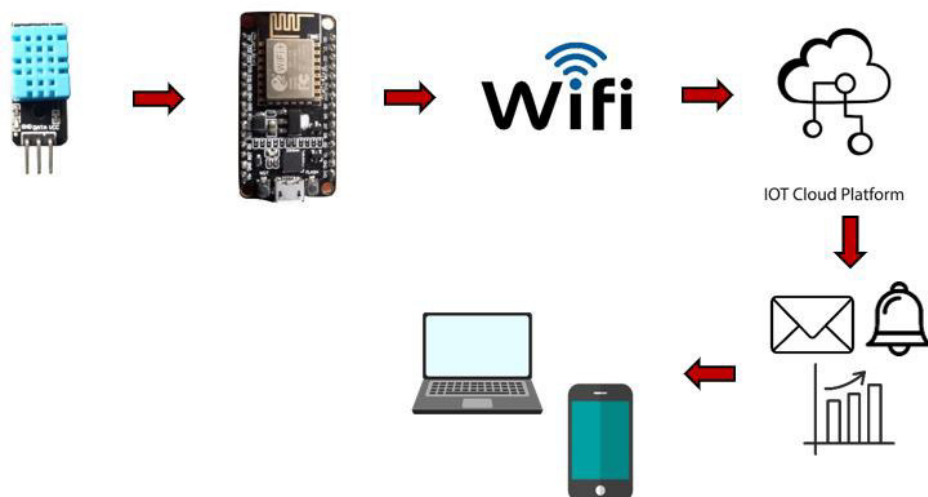
*The operation of the system will be verified, and the conclusions and recommendations will be obtained to strengthen and improve the development of the project. Finally, all the sources of research and evidence of the development of the project are presented in the bibliography and annexes sections.*

**KEYWORDS:** *temperature sensor, orchid garden, Wi-Fi module, IoT platforms, email.*

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Se ha desarrollado un sistema de alerta por detección de cambios de temperatura. El sistema permite detectar cambios de temperatura y humedad relativa en el interior de un orquideario, esto permitirá obtener resultados positivos de desarrollo, tanto en su crecimiento como en su floración. De existir una temperatura inadecuada dentro del orquideario, el personal encargado recibirá una notificación de alerta con posibles soluciones. El principio de funcionamiento del diseño es presentado en la Figura 1.1; el sensor DHT11 capta la información del sitio y la envía por un pin digital hacia el NodeMCU previamente programado. El microcontrolador cuenta con un módulo Wi-Fi ESP8266 para la transmisión de datos hacia plataformas *Internet Of Things* (IoT). La información es procesada, visualizada y notificada a través de estas plataformas.

La notificación de alerta es activada al percibir valores inadecuados de temperatura. El personal es notificado en la aplicación móvil y correo electrónico. Establecer el enlace de comunicación inalámbrica requiere del nombre de la red inalámbrica disponible o *Set Service Identifier* (SSID) seguido de la contraseña de la red Wi-Fi. Se ingresan las credenciales y se configuran parámetros de funcionamiento de las plataformas IoT en el Entorno de Desarrollo Integrado o por sus siglas en inglés *Integrated Development Environment* (IDE) de Arduino; en este caso, se usaron las plataformas Blynk y ThingSpeak. Todo el sistema es alimentado mediante un cargador de 9 (V) conectado directamente al *jack* micro-USB del microcontrolador.



**Figura 1.1** Diagrama de funcionamiento del sistema [1]-[3]

## 1.1 Objetivo general

Implementar un prototipo de sistema de alerta por cambios de temperatura.

## 1.2 Objetivos específicos

- Establecer los requerimientos del prototipo
- Definir los componentes de hardware y software
- Diseñar el prototipo
- Implementar el prototipo
- Realizar pruebas de funcionamiento

## 1.3 Alcance

El punto de partida del presente proyecto radica en el análisis del entorno y requerimientos de funcionamiento del sistema en el escenario de trabajo.

Es necesario considerar requerimientos de alimentación, tipo de comunicación, lugar de instalación, y protección del sistema. Fabricar el dispositivo requiere una investigación profunda para garantizar que los elementos que lo componen sean compatibles a nivel de hardware y de software. El sistema cuenta con los siguientes elementos principales de funcionamiento: módulo Wi-Fi, sensor de temperatura y humedad relativa, microcontrolador maestro y plataformas IoT; dispositivos que se encuentran disponibles en el mercado y en plataformas de desarrollo. Además, se toma en cuenta precios y capacidad de los elementos para su adquisición.

El sistema cuenta con un diseño 3D del prototipo, el cual es realizado en plataformas de modelado 3D que permitan recrear espacios, elementos y materiales para tener una vista general del prototipo a fabricar. Además, se incluyen los cálculos matemáticos ligados al funcionamiento del dispositivo, seguido de un diagrama de flujo en el cual se describe paso a paso el lenguaje de alto nivel empleado en la programación de los elementos que permitirán que el hardware y software trabajen como un sistema.

La fabricación del sistema considera la elaboración de una *Printed Circuit Board* (PCB), así como el establecimiento de materiales, medidas y distribución de los elementos contenidos en el módulo protector. Se usó el sensor de temperatura y humedad, que permite la captación de información de un sitio. Al realizar pruebas y detección de temperaturas inadecuadas, se verifica el envío de una notificación de alerta en la

aplicación móvil y correo electrónico del encargado. Finalmente, se realizaron grabaciones demostrativas de funcionamiento y manejo del sistema.

## **1.4 Marco Teórico**

### **Ambientes controlados de temperatura**

El monitoreo de ambientes controlados para obtener entornos optimizados está adquiriendo un gran valor de mercado. La recopilación de información y un correcto análisis permitirán que dispositivos electrónicos previamente diseñados y configurados puedan realizar acciones con el fin de resguardar, optimizar y fortalecer la integridad de un espacio determinado. IoT es un paradigma, con el cual se propone implementar cierta cantidad de elementos electrónicos conectados a la red Internet, ya sea de manera alámbrica o inalámbrica para que estos puedan interactuar entre sí y así proveer información de gran valor [4].

La temperatura es la cantidad de energía que posee un cuerpo o sustancia. Según el Sistema Internacional de unidades, es posible cuantificar la temperatura de un objeto en función de unidades tales como: grados Fahrenheit, grados Rankine, grados Kelvin y grados Celsius. La unidad empleada de manera oficial en el territorio ecuatoriano es la de grados Celsius o también conocida como grados centígrados ( $^{\circ}\text{C}$ ) [5].

Ecuador posee un clima tropical; afectado por las latitudes geográficas del país, se tiene variaciones en cada región. Las temperaturas en su mayoría generan que el clima sea templado, dando la sensación de una primavera constante durante todo el año. Determinar el comportamiento del clima puede ser complicado debido a que este varía frecuentemente entre lluvia y sol, el invierno y verano no son marcados en el territorio ecuatoriano. En la Sierra, el clima varía en función de su altitud, la cual estaría entre 500 (m) y 6 000 (m); alcanzar grandes altitudes, implica una caída de temperatura, la cual provoca una caída de 5 ( $^{\circ}\text{C}$ ) cada 200 (m). La época de sequía va desde junio hasta septiembre, en la cual se tienen las más altas temperaturas del año [6].

La capital del Ecuador, Quito, se encuentra en el norte de dicho país ubicado en el altiplano andino con una altura de 2 850 (m) sobre el nivel del mar, posicionándola como la segunda ciudad con mayor altitud en toda Latinoamérica y solo superada por La Paz en Bolivia. Los rayos solares son intensos debido a su altitud y posicionamiento geográfico. Quito es una ciudad que está rodeada por montañas y volcanes, al oeste de la ciudad se encuentra el volcán Guagua Pichincha y alrededor muchos otros como el volcán Cotopaxi, Antisana y Cayambe, los cuales influyen en el clima quiteño [7].

Un invernadero proporciona un entorno cerrado en el cual es posible controlar, proteger y estudiar especies de plantas. Estos tipos de entornos son diseñados en función de los requerimientos de los especímenes [8].

En el caso de un orquideario, se debe tener en cuenta principalmente dos fases: fase de crecimiento y fase de floración. Durante la fase de crecimiento, es importante garantizar una humedad entre 70% y 80%. Con respecto a la temperatura del medio, varía en función del día y la noche y en función de la especie de orquídea que se esté cultivando. Típicamente, los valores de temperatura de una orquídea *Odontoglossum*, una de las más populares, es de 18 - 20 (°C) en la noche y 20 - 22 (°C) durante el día. Para la fase de floración se requiere de valores óptimos de humedad entre 70% - 80% con temperaturas de 14 - 16 (°C) durante el día y 18 - 20 (°C) en la noche para la *Odontoglossum* [9].

Finalmente, para definir un rango en el cual las orquídeas puedan desarrollarse sin inconvenientes, es necesario garantizar un promedio de temperatura y humedad en el orquideario el cual contendría varias especies de orquídeas en diferentes fases. La temperatura promedio que favorece al crecimiento y floración es de 15 (°C) a 25 (°C) con una humedad del 78% [9].

### **Redes inalámbricas de área local**

Las redes inalámbricas permiten que el despliegue de una red sea sencillo, rápido y versátil, no requieren cableado. Las redes domiciliarias *Wireless Local Area Networks* (WLAN) permiten la interconexión de dispositivos electrónicos al engancharse a la red *Wireless Fidelity* (Wi-Fi). Este tipo de tecnología de acceso inalámbrico es una de las más populares. Forma parte de los estándares de red de área local *Wi-Fi Alliance*, detallada en el estándar IEEE802.11 [10].

### **Módulo Wi-Fi ESP8266**

El microcontrolador ESP8266 es un *chip* que cuenta con la capacidad de establecer un enlace inalámbrico del tipo Wi-Fi en el estándar IEEE 802.11. Este módulo es altamente empleado en el campo de desarrollo IoT debido a la facilidad de manejar protocolos como *Transmission Control Protocol / Internet Protocol* (TCP/IP). El módulo ESP8266 presentado en la Figura 1.2 fue desarrollado por la empresa *Espressif Systems* localizada en Shanghái y altamente comercializado a nivel mundial, presentándolo al mundo como una solución revolucionaria en el despliegue de redes Wi-Fi. El *chip* que maneja este módulo está basado en la tecnología *Surface Mount Technology* (SMT); esto quiere decir que existe una gran variedad de microcontroladores que disponen de

este *chip*. El ESP8266 es resistente, ideal para ser sometido a condiciones climáticas externas o internas sin que su funcionamiento se vea afectado [11][12].



**Figura 1.2** Módulo Wi-Fi ESP8266 [13]

### **Sensor DHT 11**

El sensor DHT11 es un dispositivo pequeño y de bajo consumo energético. Es altamente empleado en entornos en los que se requiere la obtención de muestras, tanto de temperatura como de humedad. El sensor posee la capacidad de obtener registros de temperatura y humedad de manera rápida y efectiva con niveles altos de respuesta. El dispositivo requiere de un elemento maestro de mayor capacidad para interpretar la información y posteriormente ser almacenada de manera parcial en una base de datos durante el funcionamiento del dispositivo. El usuario puede recuperar los datos y emplearlos en un análisis de condiciones para posteriormente actuar o no [14]. El sensor DHT11 presentado en la Figura 1.3 maneja temperaturas que van desde los 0-50 (°C) con un criterio de precisión de 2%, requiriendo un voltaje de 3-5,5 (V) para su funcionamiento [15].



**Figura 1.3** Sensor de temperatura y humedad relativa [16]

El principio de funcionamiento de estos sensores de temperatura está basado en la transformación de la energía térmica en energía eléctrica; debido a esto, se requiere de transductores termoeléctricos para que se pueda relacionar las variaciones térmicas con las eléctricas. Estos transductores reconocen a los conductores y semiconductores para realizar sus funciones. Este tipo de sensores poseen en su estructura resistencias eléctricas, termistores, medidores ópticos y de radiación [14]. Un termistor es un

elemento fabricado en materiales semiconductores sensibles a las variaciones de temperatura, generando alteraciones en el valor de la resistencia [14].

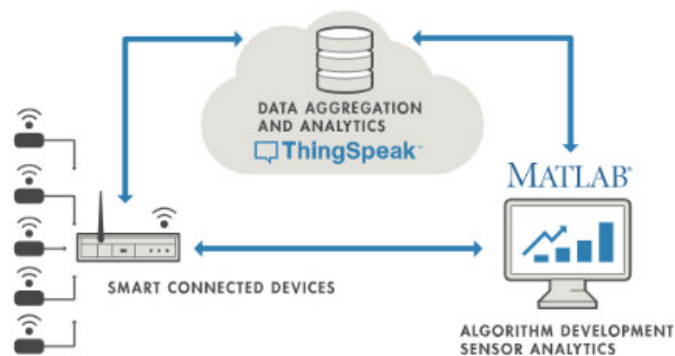
## Arduino

Una placa Arduino es un elemento electrónico que emplea un microcontrolador montado sobre un PCB, superficie de soporte no conductora sobre la cual existen rutas laminadas de conducción de energía para su funcionamiento. Este elemento es de libre desarrollo, es así como varias compañías desarrollan sus propios microcontroladores basados en los principios de funcionamiento de Arduino. Posee su propio software de desarrollo, compatible con una gran cantidad de elementos electrónicos comerciales [17].

El desarrollo del software requiere de una interfaz IDE, con la cual será posible representar ideas, soluciones y modelos para que estos posteriormente sean programados y representados en líneas de código para ser compilados, depurados y ejecutados. Este entorno de programación IDE se caracteriza por ser amigable al contar con soporte técnico, extensiones y complementos, facilidades de importar y exportar proyectos, gestión de versiones, *frameworks* populares y multiplataforma. Es posible reconocer distintos tipos de sintaxis empleados en diversos lenguajes de programación, los cuales emplearían librerías específicas para reducir el tiempo de programación, facilitando el desarrollo de proyectos [17].

## ThingSpeak

*ThingSpeak* es una plataforma con la cual es posible analizar la información de dispositivos IoT, empleando una base de datos la cual estaría entregando la información a Matlab para su interpretación; en la Figura 1.4 se presenta un esquema de funcionamiento. La información es visualizada, almacenada y estudiada en tiempo real, empleando los servicios de la nube. Es compatible con Arduino, Rasberry Pi, entre otros [18].



**Figura 1.4** Principio de funcionamiento ThingSpeak [18]

## Blynk

Blynk está compuesto por un conjunto de elementos de software, con los cuales es posible implementar y administrar de manera remota dispositivos electrónicos enlazados a la red. Es posible establecer conexión con los dispositivos de hardware y enlazarlos a Blynk *cloud* para posteriormente configurar el entorno de monitoreo y control. Blynk es compatible con los sistemas operativos iOS y Android; la simplicidad de funcionamiento se ve presentada en la Figura 1.5. No es necesario poseer un conocimiento avanzado en el desarrollo de aplicaciones móviles, esto se debe a que Blynk ya tiene creados los aplicativos para que los usuarios solo los implementen en sus diseños IoT [19].

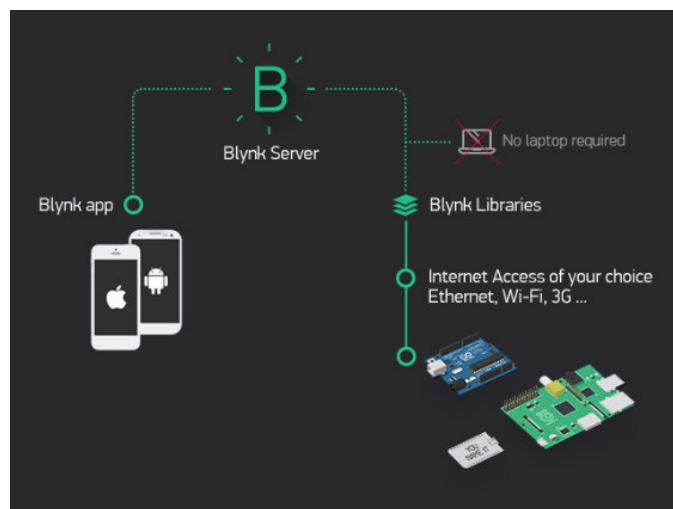


Figura 1.5 Principio de funcionamiento Blynk [20]

## 2 METODOLOGÍA

El desarrollo del proyecto consiste en cinco objetivos claves, consiguiendo la implementación del sistema de alerta por detección de cambios de temperatura, que permite optimizar el crecimiento y floración al notificar al encargado de temperaturas inadecuadas dentro del orquideario. La metodología empleada se encuentra descrita en la Figura 2.1.



## IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR DETECCIÓN DE CAMBIOS DE TEMPERATURA



**Figura 2.1** Metodología empleada

El orquideario que se consideró no posee un sistema de medición de temperatura o humedad; se ha tenido problemas con la determinación de la temperatura durante el día o noche. Monitorear el orquideario constantemente requiere de personal y tiempo, el o los encargados deben ingresar al orquideario y verificar la temperatura. Se capta la temperatura en ese instante, pero no es una forma precisa de recopilación de datos, ya que el personal estaría monitoreando el orquideario solo en ciertos intervalos y horas del día.

Al contar con Wi-Fi y un cableado eléctrico en el interior del orquideario, se optó por una instalación del tipo fija. La señal Wi-Fi captada en el orquideario es ideal para la transmisión de datos. El encargado podrá supervisar el estado del orquideario de manera remota siempre y cuando tenga acceso a Internet. La alerta es enviada vía correo electrónico de manera automática, al detectar una temperatura inadecuada. Es complementada con una notificación de alerta en la aplicación móvil IoT, en la cual se detalla posibles soluciones del incremento y decremento de temperatura.

Los dispositivos fueron seleccionados en función de su rendimiento y coste; para la medición de temperatura se investigó los sensores DHT11, DHT22 y el LM35. Por otra parte, se realizó un análisis de los posibles microcontroladores entre ellos: NodeMCU ESP8266, ESP8266-01, ESP 32. También, se indagó plataformas IoT para el procesamiento y visualización de los datos, luego de realizar una investigación se obtuvo las siguientes alternativas: Arduino cloud, Blynk, ThingSpeak, Ubidots y AskSensor.

Se realizó el modelado, simulado y diseño PCB del circuito electrónico empleando la herramienta Proteus. Se ingresó librerías de los componentes electrónicos empleados para la creación de pistas y posicionamiento de elementos. La construcción del circuito fue sobre una baquelita de cobre, la cual posee el diseño del circuito transmitido térmicamente al emplear el método del planchado. Se cortó, taladró y soldó los elementos sobre la placa fabricada. Asimismo, para la creación del módulo de protección se empleó SketchUp para el dimensionamiento y AutoCad para la realización exacta de las piezas a cortar. El módulo de protección fue fabricado en acrílico de 3 (mm) de espesor. Los rangos de funcionamiento del sensor fueron extraídos de estudios estadísticos captados en el interior de orquidearios, en los cuales señalan temperaturas en promedio favorecedoras.

Se realizó un sin número de pruebas de funcionamiento; en principio, a través de las plataformas de visualización IoT y monitor serial del IDE de Arduino para comprobar latencias y valores captados por el sensor DHT11. Adicional, se verificó el funcionamiento de la notificación de alerta en el correo electrónico y aplicativo móvil.

### **3 RESULTADOS**

Al obtener el prototipo del sistema de alerta por detección de cambios de temperatura, es posible notificar al encargado sobre temperaturas inadecuadas, al poseer acceso a Internet para visualizar el correo electrónico y notificación del aplicativo móvil. Se pone a disposición información relevante, que le permita al encargado normalizar la temperatura del orquideario. La detección del cambio de temperatura es captada por el sensor DHT11, y procesada por el microcontrolador NodeMCU previamente programado, para luego ser transmitida por el módulo Wi-Fi ESP8266 hacia las plataformas IoT Blynk y ThingSpeak. La información es mostrada en las plataformas y aplicativos móviles que posean acceso a Internet. El desarrollo del dispositivo está comprendido de cinco fases; inicialmente, se indaga y establece requerimientos del sistema. Luego, se adquiere el hardware y software más convenientes; seguido, se procede con el diseño del prototipo e instalación del sistema. Finalmente, se realizan pruebas de funcionamiento.

#### **3.1 Establecimiento de requerimientos del prototipo**

El modelo a implementar usaría un módulo Wi-Fi debido a que el sitio permanecería estático con disponibilidad de una red común de Wi-Fi, la cual emplearía las bandas del espectro no licenciado de 2.4 (GHz) [21]. Se podrá monitorear la temperatura y humedad

relativa que existen dentro de un invernadero de orquídeas, reduciendo los tiempos de crecimiento y floración de los especímenes.

El dispositivo contará con una fuente de alimentación principal cableada, la cual estaría conectada al tendido eléctrico común en un domicilio. El sensor, microcontrolador y módulo de comunicación deberán ser de tamaño reducido ya que se pretende que el dispositivo pase desapercibido en el sitio de despliegue. Todos los elementos serán almacenados en una caja de protección de pequeñas dimensiones. Se podrá salvaguardar la integridad de los elementos electrónicos y extender su tiempo de vida útil.

Para el despliegue, notificación y análisis de información es necesaria la implementación de una interfaz de usuario móvil en la cual se mostrarán los datos del entorno en tiempo real siempre y cuando se tenga acceso a una red de Internet. La interfaz del aplicativo sería amigable e intuitiva hacia los usuarios. Además, será posible visualizar la información desde un computador o *smartphone*. Visualizar la información con gráficos representativos simples de temperatura permitirán que el análisis sea sencillo y directo para los usuarios. Para ello, será necesario almacenar la información en una base de datos en la nube, permitiendo que los dispositivos se encuentren en constante actualización. De esta manera, la información sería transmitida desde los dispositivos hacia la aplicación en tiempo real.

### **3.2 Definición de componentes de hardware y software**

Existe una gran variedad de módulos Wi-Fi empleados en el área IoT, los cuales cumplen con los requerimientos del proyecto. El análisis físico de los módulos, indican que poseen características similares entre sí, el rendimiento energético, tamaño del módulo, protocolos de seguridad, voltaje de operación y manejo de estándares Wi-Fi. La selección del módulo definitivo requirió un análisis comparativo de requerimientos esenciales del proyecto; debido a esto, se tomaron en cuenta las características principales de tres módulos Wi-Fi: NodeMCU ESP8266, ESP32 y ESP8266-01, presentados en la Tabla 3.1.

NodeMCU ESP8266 es un microcontrolador que cuenta con una cantidad de pines intermedio, el número de pines es inferior al que posee el módulo ESP32, visto en la Figura 3.2. Al comparar sus dimensiones, resulta que el módulo NodeMCU ESP8266 visto en la Figura 3.1, es más pequeño. Al contar con una cantidad intermedia de pines, el proyecto puede ser escalable [22]. El IoT busca la obtención de información de varios dispositivos conectados a la red para optimizar, monitorear y actuar sobre un entorno.

NodeMCU tiene incorporado un microcontrolador con un módulo Wi-Fi ESP8266 todo en uno, esto permitirá reducir el número de elementos y gestionarlos de una manera centralizada.

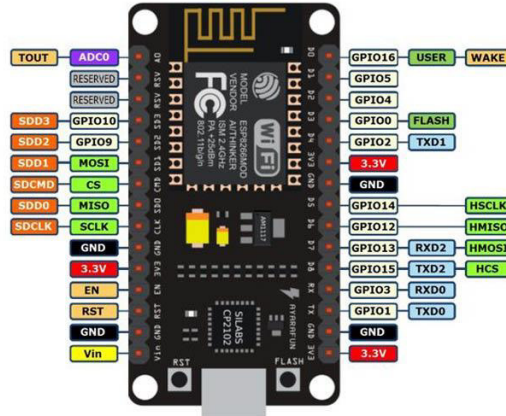


Figura 3.1 Módulo Wi-Fi NodeMCU ESP8266 [22]

Por otro lado, se tiene el módulo ESP32, el más popular en el desarrollo del IoT, presentado en la Figura 3.2. Cuenta con un número abundante de pines; puede transmitir una gran cantidad de información sin que el rendimiento del *chip* se vea afectado. Posee en su infraestructura módulos *bluetooth* y Wi-Fi, complementando una transmisión inalámbrica de área personal y local. Sin embargo, el costo y tamaño de este microcontrolador es superior al compararlo con otros módulos. Además, se considera que, para la implementación del sistema de detección de temperatura, el módulo en mención estaría sobredimensionado.

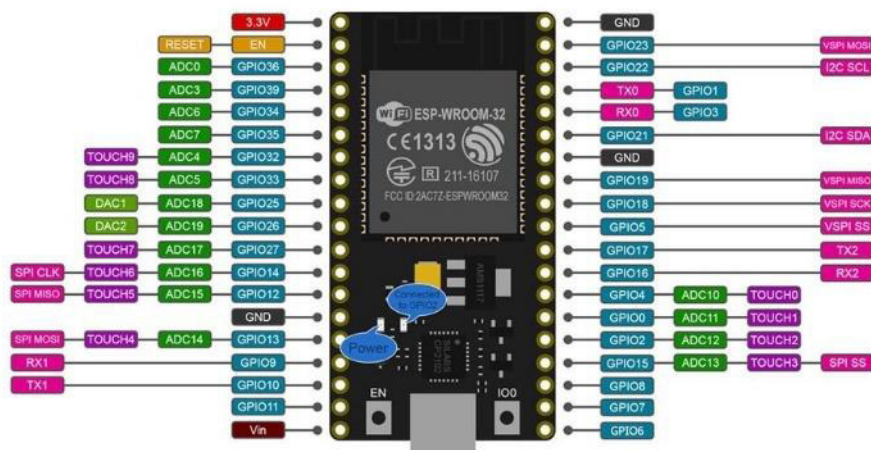
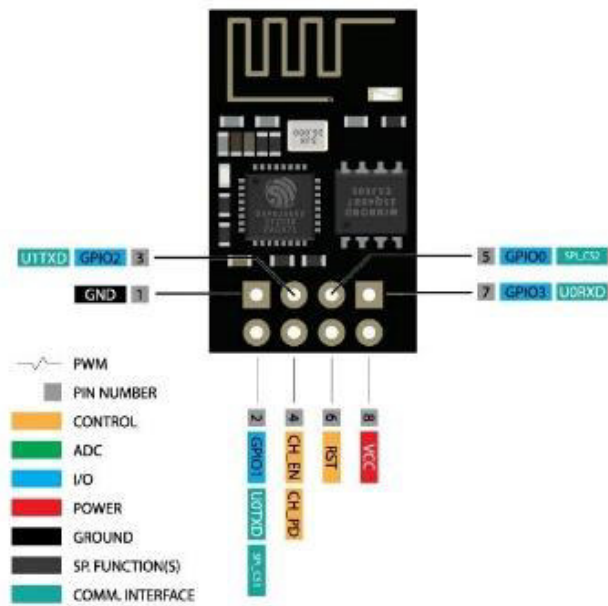


Figura 3.2 Módulo Wi-Fi / *bluetooth* ESP32 [23]

El módulo ESP8266-01 presentado en la Figura 3.3 es uno de los dispositivos que presenta una gran compatibilidad con el proyecto. Las dimensiones del módulo y el rendimiento energético son dos de los factores que posicionan a este módulo sobre otros ESP. El ancho de banda empleado es suficiente para la transmisión de datos de temperatura y humedad; sin embargo, la limitada cantidad de pines GPIO restringe la expansión del sistema de detección de temperatura [24]. De emplear este módulo, se requeriría de un microcontrolador adicional, el cual estaría conectado en serie con el módulo ESP8266-01, incrementado el costo y número de elementos electrónicos en la placa del sistema.



**Figura 3.3** Módulo Wi-Fi ESP8266-01 [24]

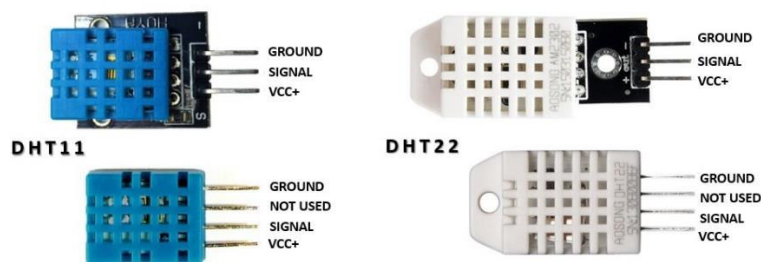
El módulo seleccionado que permite establecer una conexión inalámbrica Wi-Fi en el proyecto es el NodeMCU ESP8266, ideal para el prototipo que busca ser portable, de pequeñas dimensiones, de alto rendimiento energético, y compatible con las plataformas que permitirán visualizar la información de manera objetiva. En la Tabla 3.1 se puede ver claramente los parámetros a considerar y la razón por la cual se optó por el módulo mencionando [22]–[24].

**Tabla 3.1** Comparación de módulos Wi-Fi [22], [24], [25]

Especificación	NodeMCU ESP8266	ESP32	ESP8266-01
<b>Microcontrolador</b>	Tensilica 32-bit RISC CPU Xtensa LX106	Tensilica Xtensa LX6	Tensilica L106 32- bit
<b>Voltaje de operación</b>	3,3 -5 (V)	3,3 – 5 (V)	3,3 (V)
<b>Pines Digitales I/O (DIO)</b>	16	39	2
<b>Pines Analógicos I/O (ADC)</b>	1	-	-
<b>Puerto USB</b>	Micro USB	Micro USB	Mini USB
<b>Estándares Wi-Fi</b>	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
<b>Seguridad Wi-Fi</b>	WPA/WPA2	WPA/WPA2	WPA/WPA2
<b>Rango de temperatura</b>	-40 – 125 (°C)	-40 – 125 (°C)	-40 – 125(°C)
<b>Dimensiones</b>	49x 6 (mm)	55 x 28 (mm)	14,3 x 24.8 (mm)
<b>Frecuencia de operación</b>	80 (MHz)	160 (MHz)	80 (MHz)

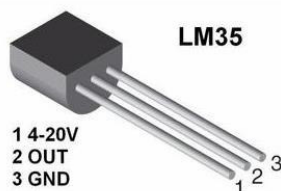
La selección del sensor de temperatura se basa en el consumo de energía, tamaño y precisión de toma de medidas. Se analizaron los datos propuestos en la Tabla 3.2, obtenidos de los *datasheets* correspondientes [26], [27].

Se tomó en cuenta los sensores de temperatura y humedad DHT11 y DHT12, vistos en la Figura 3.4; ambos poseen características similares entre sí. Por un lado, se tiene al DHT11, el cual maneja rangos de temperatura que están sobre los 0 (°C) y no sobrepasan los 50 (°C). Resulta que los requerimientos de temperatura del sistema de detección de dicha magnitud en el interior de un sitio no apuntan a la detección de temperaturas extremas. Debido a este hecho, el sensor DHT12 es descartado por su sobredimensionamiento y coste adicional, a pesar de que la precisión de este sensor es superior al DHT11 [26].



**Figura 3.4** Sensores DHT11 y DHT12 [26]

Finalmente, se tiene al LM35, visto en la Figura 3.5; un sensor exclusivo de temperatura que puede operar con rangos altos de voltaje. Su toma de medidas es de alta precisión; sin embargo, está limitada únicamente a obtener datos de temperatura comprendida entre un gran rango [27].



**Figura 3.5** Sensor LM35 [27]

Ya que este dispositivo no cuenta con la posibilidad de detectar la humedad del medio, factor clave en la toma de medidas de un entorno controlado, se opta por emplear un sensor DHT11 en el modelo a implementar.

**Tabla 3.2** Comparación de sensores de temperatura [26], [27]

Especificación	DHT11	DHT22	LM35
<b>Voltaje de alimentación</b>	3,3 – 5 (V)	3,3 – 5 (V)	4 – 30 (V)
<b>Precisión de lectura de Humedad</b>	(+/-) 5%	(+/-) 2-5%	-
<b>Lectura de temperatura</b>	(+/-) 2 (°C)	(+/-) 0,5 (°C)	(+/-) 0,5 – 25 (°C)
<b>Capacidad de medir humedad</b>	20 - 80%	0 - 100%	-
<b>Capacidad de medir temperatura</b>	0 – 50 (°C)	-40 – 125 (°C)	-55 – 150 (°C)
<b>Dimensiones</b>	15,5x12x5,5(mm)	15,1x25x7,7(mm)	28,2x13,1x5,5(mm)

El software con el cual se podrá monitorear, controlar y visualizar la información complementa la funcionalidad del modelo. Se consideró las siguientes plataformas de desarrollo: *Arduino cloud*, ThingSpeak, Blynk, Ubidots y AskSensors.

*Arduino cloud* es una plataforma digital empleada en la configuración, programación y control remoto de dispositivos usados en proyectos IoT mediante una App disponible para teléfonos celulares iOS y Android. Su plataforma permite la implementación de *dashboards* de una manera sencilla y amigable, la cual es compatible con múltiples placas y módulos. Cuenta con una gran cantidad de librerías y *sketches*, permitiendo comprobar e implementar proyectos de manera sencilla [28].

Por otro lado, se tiene a Blynk, una plataforma especializada en dispositivos IoT que cuenta con todas las características vistas en Arduino *cloud*, pero con una interfaz más llamativa, profesional y funcional. Además, la programación es realizada mediante el ingreso de librerías específicas de funcionamiento ligadas al hardware. Debido a esto, la programación del sistema de detección de temperatura con alertas y visualización de datos en tiempo real es altamente configurable. Incluso, es posible implementar arreglos sin la necesidad de reescribir el código por completo [19].

Se tiene otras plataformas de desarrollo IoT tales como Ubidots y AskSensor, las cuales cumplen con lo mencionado anteriormente, pero con limitaciones extras las cuales hacen que sean descartadas debido a la dificultad que existe con la compatibilidad. Las interfaces son poco llamativas pero funcionales. Estas plataformas se encuentran en pleno desarrollo y esto dificulta la obtención de documentación oficial de proyectos IoT [29], [30].

Un punto importante a ser considerado es que la selección del software a implementar en el proyecto es gratuita. En la Tabla 3.3 se puede apreciar la comparación de los planes gratuitos ofertados por estas empresas de desarrollo IoT.

**Tabla 3.3** Comparación de softwares IoT [18], [19], [28]–[30]

<b>Características</b>	<b>Arduino Cloud</b>	<b>Blynk</b>	<b>ThingSpeak</b>	<b>Ubidots</b>	<b>AskSensor</b>
<b>Retención de datos</b>	1 día	1 semana	Importados	-	3 meses
<b>Número de dispositivos</b>	2	2	Ilimitado	3	2
<b>Duración del plan gratuito</b>	Permanente	Permanente	1 año	Permanente	3 meses
<b>Dashboards</b>	1	1	1	1	No dispone
<b>Indicadores y widgets</b>	Básicos	Básicos	Gráficas Matlab	+30 Básicos	No dispone
<b>Alertas</b>	<i>Webhooks</i>	Email, SMS, Web y App	<i>Web</i>	Email, telegram, SM, llamada de voz y <i>Webhooks</i>	Email

El proyecto será implementado sobre las plataformas de desarrollo, visualización, control y monitoreo Blynk y ThingSpeak debido a que estos presentan un mejor desempeño con respecto de otros softwares de gestión IoT. La documentación y



compatibilidad de los elementos de hardware seleccionados permitirán establecer una transmisión de datos sobre la red Wi-Fi.

ThingSpeak será un servicio complemento debido a que esta plataforma emplearía Matlab para la representación de datos, lo cual permitiría una mejor manipulación de información si se desea un estudio más detallado del entorno [18].

### 3.3 Diseño del prototipo

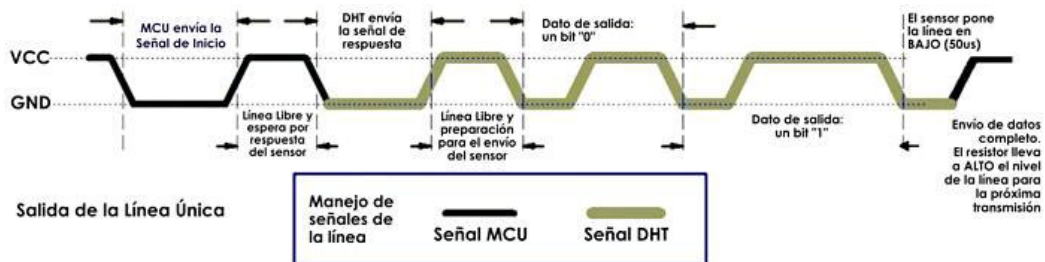
#### Proceso de comunicación

El sensor DHT11 envía la información captada por un pin digital DATA, facilitando la programación del sensor. El sensor de temperatura maneja una trama de 40 bits de información, en la cual estarían contenidos los datos de temperatura y humedad. Estos bits de información están distribuidos en 3 grupos; los dos primeros grupos de bits son para la medición de temperatura y humedad, mientras que el tercer grupo corresponde a bits de paridad. Estos dos primeros grupos manejan 16 bits cada uno, 8 bits para la parte entera y 8 bits para la parte decimal, tanto para temperatura como para humedad. Los 8 bits sobrantes son empleados en la detección de errores. En la Figura 3.6 se puede apreciar el formato de secuencia de bits que se está manejando.



**Figura 3.6** Formato del proceso de comunicación de 5 Bytes [31]

El microcontrolador NodeMCU ESP8266 envía una señal de activación hacia el sensor DHT11, cambiando su estado de bajo consumo a un estado de ejecución, como se puede apreciar en la Figura 3.7. DHT11 envía la información de temperatura y humedad mediante un mensaje respuesta de 40 bits de longitud. Este proceso de comunicación dura aproximadamente 4(ms).



**Figura 3.7** Proceso de comunicación entre el MCU y DHT11 [31]

El periodo de muestreo del sensor DHT11 dependerá de las características eléctricas que se estén manejando, como el voltaje y corriente de alimentación. El correcto funcionamiento del sensor dependerá de un periodo de muestreo de al menos 1 segundo. Si el periodo es menor, la sincronización de la señal digital se torna compleja y puede generar problemas en la transmisión de datos.

### Rangos de funcionamiento

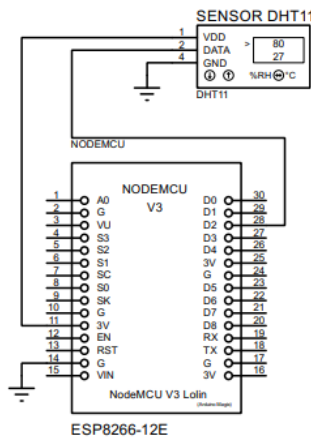
Los valores definidos en el funcionamiento del sistema de detección de temperatura están basados en datos estadísticos obtenidos del comportamiento de las orquídeas dentro del invernadero. Al analizar y comprender las condiciones de temperatura que tienen estas especies de orquídeas, se determinó un rango de temperatura promedio que va desde los 15 - 25 (°C) con una humedad relativa promedio de 78% [27]. En la Tabla 3.4 se detalla los rangos de funcionamiento que maneja el sistema.

**Tabla 3.4** Rangos de funcionamiento

Nivel de Temperatura	Rangos de funcionamiento
Extremo frío	$\leq 15(^{\circ}\text{C})$
Ambiente medio	$16(^{\circ}\text{C}) \leq T \leq 24(^{\circ}\text{C})$
Extremo caliente	$\geq 25(^{\circ}\text{C})$

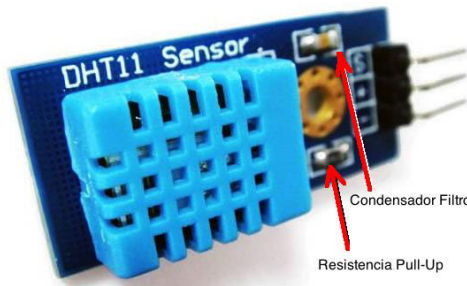
### Circuito Electrónico

El diseño del circuito fue generado mediante el software Proteus; este programa permite realizar la simulación, modelado 3D y diseño PCB de elementos electrónicos que se encuentren dentro de la librería local. El circuito del sistema de detección de temperatura se encuentra en la Figura 3.8.



**Figura 3.8** Circuito del sistema de temperatura

Como se puede ver en el diagrama del circuito, un sensor de temperatura DHT11 posee tres pines al estar montado sobre el módulo KY-015: Vcc, DATA y GND. El módulo sobre el cual está montado el sensor contiene elementos SMD, como se puede ver en la Figura 3.9, una resistencia de 1 (k $\Omega$ ), un capacitor de 0,01 ( $\mu$ F) y un led indicador de poder.



**Figura 3.9** Módulo DHT11 KY-015 [32]

El uso de este módulo facilita la implementación y diseño del circuito, permitiendo una compatibilidad directa con el microcontrolador NodeMCU ESP8266. Los componentes internos del módulo KY-015 pueden ser reemplazados por resistencias obtenidas mediante la ley de Ohm, fórmula descrita en la Ecuación 3.1; la resistencia es directamente proporcional al voltaje e inversamente proporcional a la corriente del componente electrónico.

$$R_{DHT} = \frac{V_{CC}}{I_{DHT}}$$

**Ecuación 3.1** Ley de Ohm [33]

Donde:

- $V_{CC}$  : 3,3 o 5 (V)
- $I_{DHT}$  : 0,5 o 1,5 (mA)
- $R_{DHT}$  : ( $\Omega$ ) resistencia *Pull-Up*

Es así, que se tiene:

$$R_{DHT} = 6,6 \text{ (k}\Omega\text{)} \text{ o } R_{DHT} = 3,3 \text{ (k}\Omega\text{)}$$

Los valores obtenidos son meramente teóricos, las resistencias comerciales están estandarizadas. El valor real de las resistencias cuyo voltaje de alimentación es de 3,3 (V) y 5 (V) son típicamente de 10 (k $\Omega$ ) y 3,3 (k $\Omega$ ) respectivamente, como se observa en la Figura 3.10.

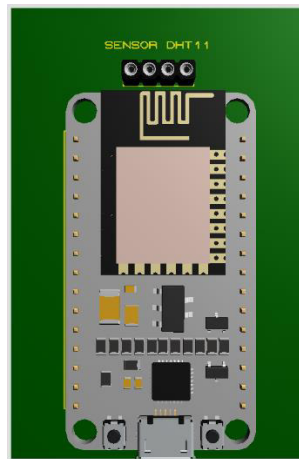
VALORES COMERCIALES EN $\Omega$						
x1	x10	x100	x(k)	x(10k)	x(100k)	x(M)
1 $\Omega$	10 $\Omega$	100 $\Omega$	1 K $\Omega$	10 K $\Omega$	100 K $\Omega$	1M $\Omega$
1,2 $\Omega$	12 $\Omega$	120 $\Omega$	1K2 $\Omega$	12 K $\Omega$	120 K $\Omega$	1M2 $\Omega$
1,5 $\Omega$	15 $\Omega$	150 $\Omega$	1K5 $\Omega$	15 K $\Omega$	150 K $\Omega$	1M5 $\Omega$
1,8 $\Omega$	18 $\Omega$	180 $\Omega$	1K8 $\Omega$	18 K $\Omega$	180 K $\Omega$	1M8 $\Omega$
2,2 $\Omega$	22 $\Omega$	220 $\Omega$	2K2 $\Omega$	22 K $\Omega$	220 K $\Omega$	2M2 $\Omega$
2,7 $\Omega$	27 $\Omega$	270 $\Omega$	2K7 $\Omega$	27 K $\Omega$	270 K $\Omega$	2M7 $\Omega$
3,3 $\Omega$	33 $\Omega$	330 $\Omega$	3K3 $\Omega$	33 K $\Omega$	330 K $\Omega$	3M3 $\Omega$
3,9 $\Omega$	39 $\Omega$	390 $\Omega$	3K9 $\Omega$	39 K $\Omega$	390 K $\Omega$	3M9 $\Omega$
4,7 $\Omega$	47 $\Omega$	470 $\Omega$	4K7 $\Omega$	47 K $\Omega$	470 K $\Omega$	4M7 $\Omega$
5,1 $\Omega$	51 $\Omega$	510 $\Omega$	5K1 $\Omega$	51 K $\Omega$	510 K $\Omega$	5M1 $\Omega$
5,6 $\Omega$	56 $\Omega$	560 $\Omega$	5K6 $\Omega$	56 K $\Omega$	560 K $\Omega$	5M6 $\Omega$
6,8 $\Omega$	68 $\Omega$	680 $\Omega$	6K8 $\Omega$	68 K $\Omega$	680 K $\Omega$	6M8 $\Omega$
8,2 $\Omega$	82 $\Omega$	820 $\Omega$	8K2 $\Omega$	82 K $\Omega$	820 K $\Omega$	8M2 $\Omega$

**Figura 3.10** Resistencias comerciales estándar [34]

Se debe recordar que el módulo KY-015 permite omitir la introducción de una resistencia adicional entre el pin digital y alimentación.

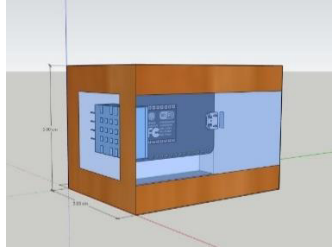
### Diseño 3D del prototipo

El diseño 3D fue obtenido como una de las funcionalidades que tiene Proteus. Se debe tener dentro de las librerías locales de Proteus los diseños de los elementos electrónicos con medidas reales que simularían el área requerida para el diseño de la PCB. En la Figura 3.11 se muestra la distribución de los elementos electrónicos sobre su respectiva placa.



**Figura 3.11** Modelo 3D de los elementos electrónicos

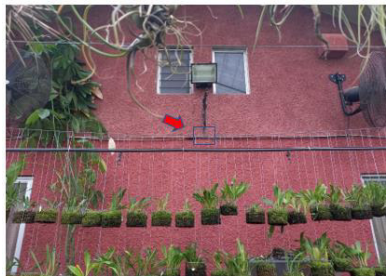
Por otro lado, el diseño del módulo protector del circuito fue desarrollado en el software SketchUp *online*; altamente empleado en el ámbito profesional e industrial, siendo uno de los pilares en el diseño y modelado 3D de infraestructuras arquitectónicas, tecnológicas y de ingeniería civil [35]. Como resultado del diseño, se tiene la Figura 3.12.



**Figura 3.12** Diseño 3D del módulo de protección

### **Lugar de implementación**

El lugar en donde el sistema de temperatura será instalado es un invernadero de 12 x 8 x 6 (m) de largo, ancho y alto, respectivamente. El área cubierta por el orquideario está dentro del alcance del sistema de detección de temperatura. Al emplear el sensor DHT11 es posible realizar la transmisión hasta en un radio de 20 (m). En la Figura 3.13 se muestra el orquideario y el sitio específico en el cual se implementará el sistema.



**Figura 3.13** Lugar de implementación

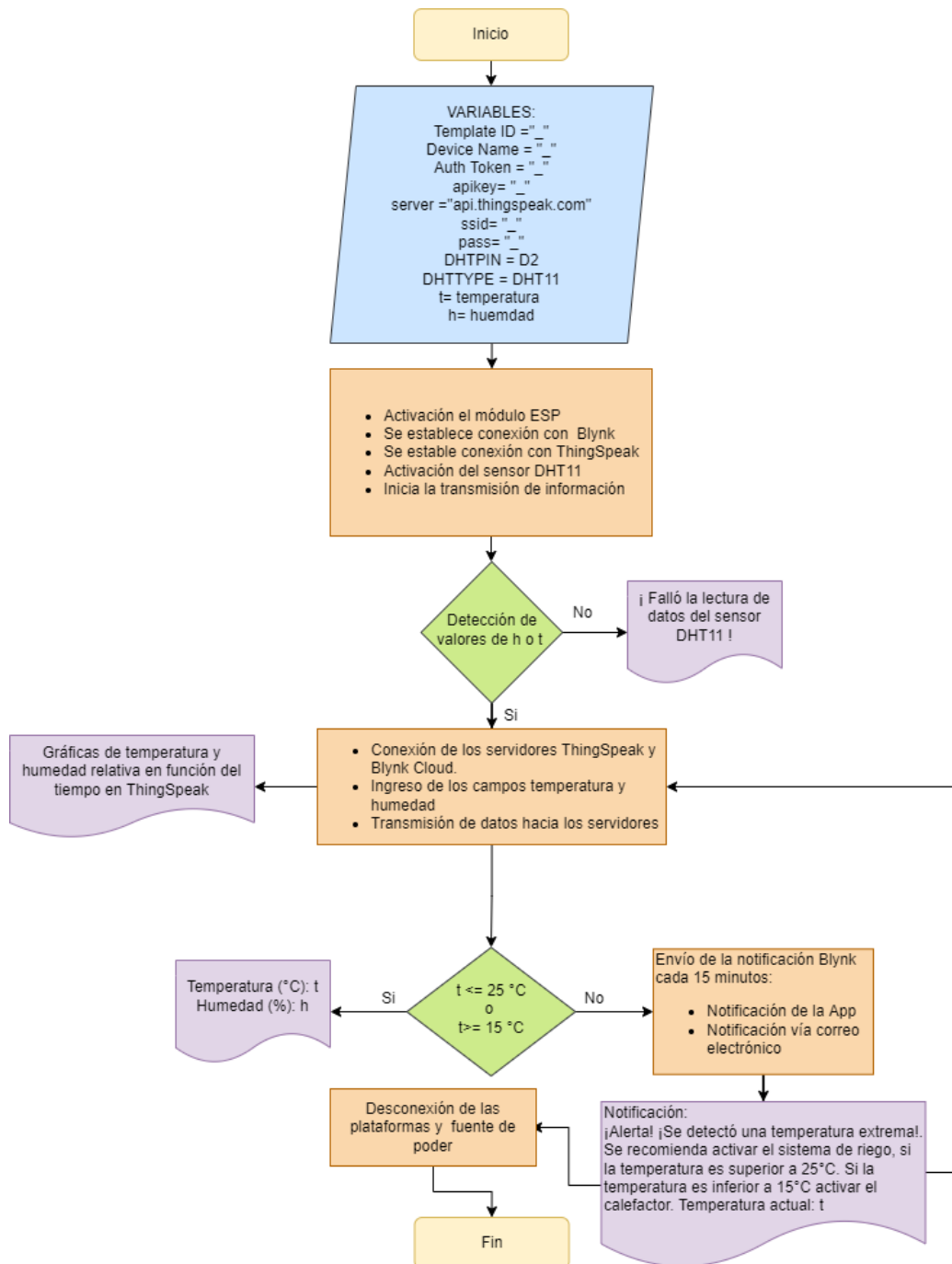
### **Diagrama de flujo**

El funcionamiento del código de programación del sistema de temperatura presentado en el diagrama de flujo de la Figura 3.14 cumple con principios básicos de programación. El encabezado contiene los requerimientos generales de funcionamiento, se define las variables de funcionamiento y se ingresa las librerías blynk, ESP8266, DHT y ThingSpeak. Al terminar la declaración de variables, se habilita el monitor serial y se activa el reloj de transmisión. El sensor DHT11 y NodeMCU ESP8266 son activados y establecen una conexión del tipo Wi-Fi con las credenciales SSID y PASS ingresadas con anterioridad. Al establecer la conexión, se mostrará en el monitor serial de manera automática el enlace, esta es una de las funcionalidades que tienen las librerías blynk.

Al establecerse la comunicación del dispositivo, se validará el funcionamiento del sensor DHT11 mediante el uso de un operador *If*. Si el sensor no entrega la información de temperatura o humedad, se visualizará en pantalla un mensaje “Falló la lectura del sensor DHT11”. Si está recibiendo los datos de temperatura y humedad, estos ingresan en un operador *If*, el cual permitirá establecer los parámetros de notificación de alerta.

La temperatura que activa la alerta deberá ser inferior a 15 (°C) o superior a 25 (°C). La notificación será enviada vía correo electrónico y vista en las notificaciones Blynk.

Al validarse el funcionamiento del sensor, ThingSpeak transmite la información sobre los campos temperatura y humedad, los cuales son procesados en la nube Matlab. Se genera las gráficas históricas de temperatura y humedad relativa en función del tiempo. Los datos de temperatura y humedad siguen transmitiéndose hasta que se retire la fuente de alimentación del sistema.



**Figura 3.14** Diagrama de flujo del sistema de detección de temperatura

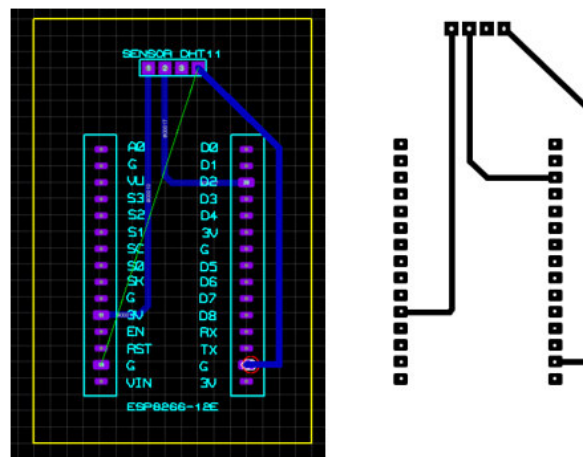
## Alimentación

La alimentación del circuito será mediante el puerto Micro-USB que posee el microcontrolador NodeMCU. Se empleará un cargador que entregue un voltaje de 9 (V). El microcontrolador posee un mecanismo de regulación de voltaje, permitiéndole manejar voltajes de 7 - 12 (V). NodeMCU entregará un voltaje de 3,3 (V) al sensor, activando el LED indicador del DHT11 e iniciando la transmisión de datos.

## 3.4 Implementación del prototipo

### Elaboración de la placa electrónica

El diseño del circuito PCB a implementar sobre una baquelita de 8 x 6 (cm) de largo y ancho fue desarrollado en el software Proteus. Las librerías del sensor DHT11 y NodeMCU permiten recrear las dimensiones exactas de los dispositivos electrónicos con sus respectivas conexiones de funcionamiento. En la Figura 3.15 se puede ver el diseño PCB del circuito a ser impreso sobre una baquelita de cobre.



**Figura 3.15** Diseño del circuito PCB en Proteus

Una vez terminado el circuito, este será impreso sobre una lámina de transferencia térmica; para ello se empleó una impresora láser de alta precisión, con el fin de calcar el circuito diseñado sobre la baquelita. La baquelita es cortada y limpiada para proceder con la transferencia térmica, empleando la técnica del planchado. La plancha es configurada a su máxima potencia y es colocada sobre la lámina de transferencia por unos 30 a 60 segundos. Finalmente, se deja enfriando la placa y se retira la lámina con mucho cuidado, obteniendo como resultado la Figura 3.16.



**Figura 3.16** Transferencia térmica del diseño empleando el método del planchado

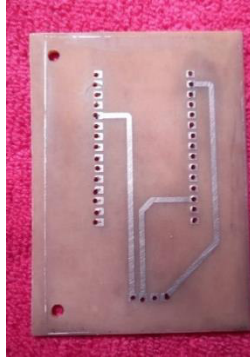
Al tener el diseño correctamente calcado, se procede a realizar las perforaciones correspondientes para que los elementos electrónicos puedan ser montados sobre la placa. Para realizar las perforaciones fue necesario poseer una broca para metal de 1(mm) y taladro convencional montado sobre una estructura de apoyo, como se puede apreciar en la Figura 3.17.



**Figura 3.17** Perforación de los agujeros del diseño

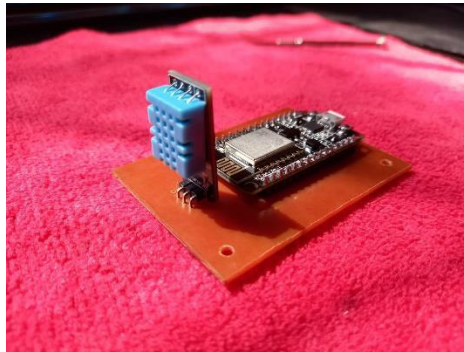
Luego, se procede a introducir la baquelita en un recipiente plástico en el cual se agregaría agua y cloruro férrico. A la mezcla le tomaría 30 minutos en remover el cobre de las zonas libres de tinta, obteniendo la placa vista en la Figura 3.18.





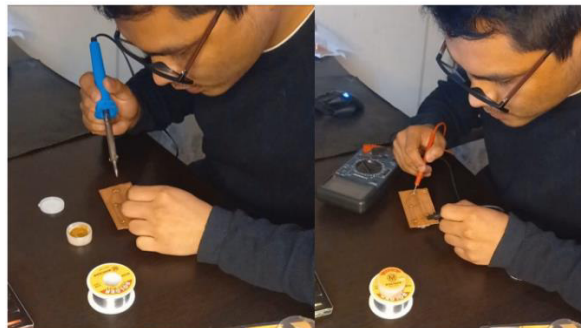
**Figura 3.18** Quemado de la placa PCB

Al terminar el quemado y las perforaciones de la PCB, se procede a introducir los elementos electrónicos sobre la placa, tal y como se ve en la Figura 3.19.



**Figura 3.19** Montado de los elementos sobre la placa

Al encajar los elementos correctamente sobre la placa, se procede a soldar los pines con la ayuda de un caudín y estaño. Se verificó la continuidad de las pistas, y la calidad de los puntos de suelda, como se ve en la Figura 3.20.



**Figura 3.20** Soldado de elementos y continuidad entre pines

### Alimentación del prototipo

El sistema es alimentado mediante un cargador celular, visto en la Figura 3.21, el cual estaría entregando de 5 - 9 (V), dentro del rango de operación que maneja el NodeMCU. Este cargador estaría directamente conectado al puerto micro-USB del NodeMCU. Finalmente, el microcontrolador entregaría 3,3 (V) por uno de los pines para el funcionamiento del sensor DHT11.



**Figura 3.21** Cargador micro-USB

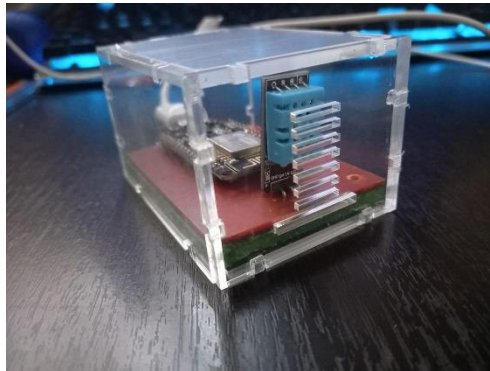
### Elaboración del módulo protector

El medio al cual estaría expuesto el circuito posee índices altos de humedad. Debido a esto, el módulo protector fue fabricado sobre una plancha de acrílico y cartón. La placa desarrollada de 5 x 7 (cm) sería encerrada dentro de una caja protectora de acrílico, la cual tendría una base de soporte de cartón. Las piezas de la caja fueron diseñadas en AutoCAD con especificaciones exactas y exportado como un archivo en formato xdf para posteriormente ser cortadas con una cortadora láser de alta precisión. Al finalizar el corte de las 6 piezas, se obtuvo como resultado la Figura 3.22.



**Figura 3.22** Piezas del módulo protector

Se ingresa la placa dentro de la caja de acrílico para posteriormente pegar las piezas con silicona, obteniendo así lo presentado en la Figura 3.23.



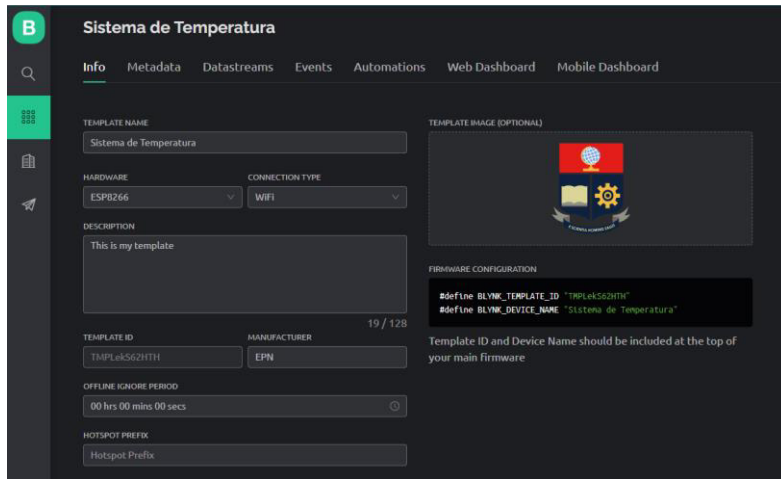
**Figura 3.23** Circuito encerrado en el módulo protector

### Configuración de plataformas IoT

En esta sección se aborda la creación del proyecto e íconos de visualización de las plataformas IoT Blynk y ThingSpeak.

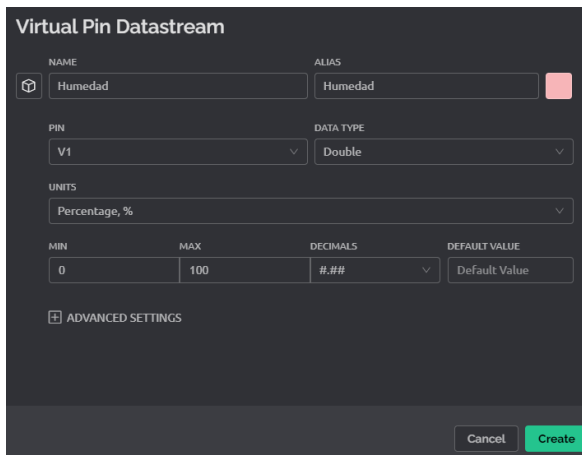
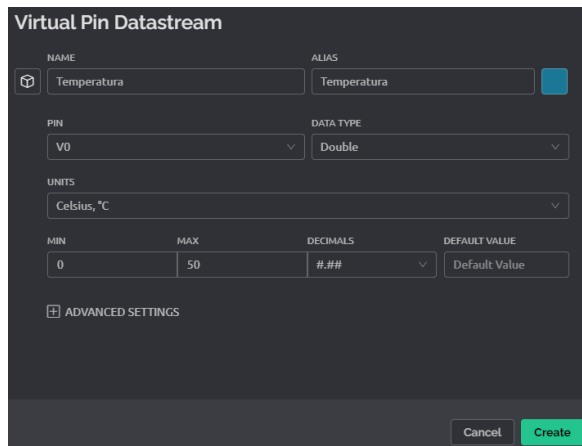
Se requiere generar una cuenta Blynk; al ingresar, se desplegará un menú de desarrollo permitiendo la creación un *template*. Este *template* es la plantilla del proyecto, contiene la información del Hardware y el tipo de conexión, como se puede ver en la Figura 3.24 y Figura 3.25.

**Figura 3.24** Creación del *template*



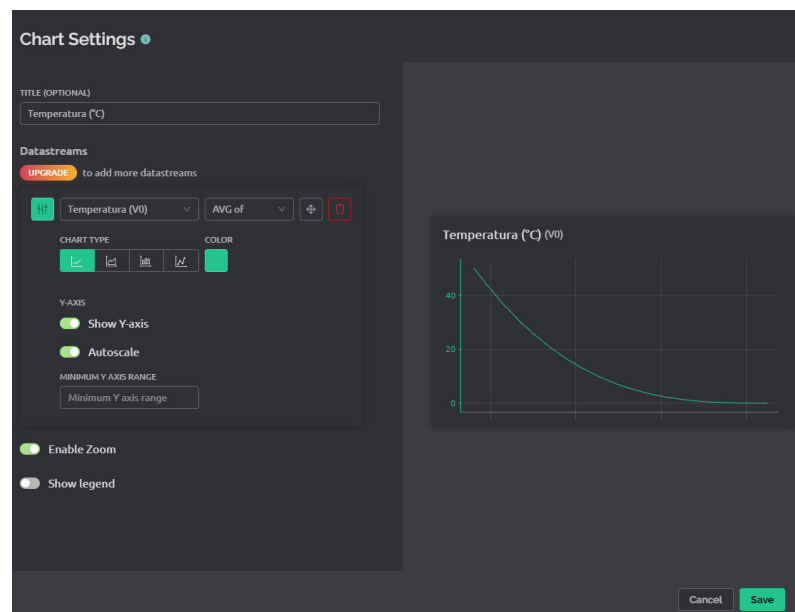
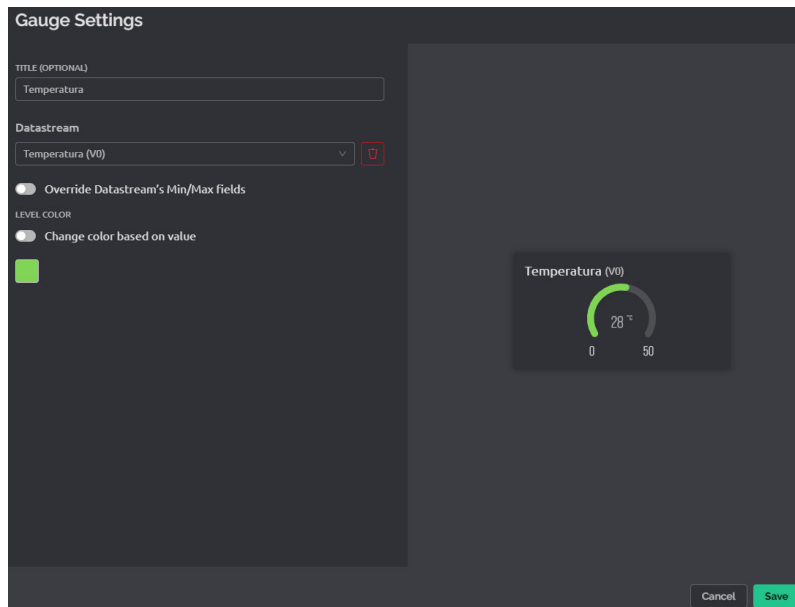
**Figura 3.25** Información del *template*

Luego, se ingresa en *datastreams* y se generan los pines virtuales V0 y V1 asociados a la temperatura y humedad relativa, respectivamente. Estos pines son de gran importancia en la configuración de los elementos de visualización.



**Figura 3.26** Configuración de pines digitales de temperatura y humedad

Después, se ingresa en el apartado *web dashboard*. Se selecciona alguno de los *widgets* proporcionados por Blynk; se optó por *widgets* de medición y cuadros gráficos históricos para la visualización de los datos de temperatura y humedad, como se puede ver en la Figura 3.27 y Figura 3.28. Cada uno de los *widgets* es asociado con su correspondiente pin virtual, obteniendo así la Figura 3.29.



**Figura 3.27** Configuración de indicadores de temperatura en Blynk

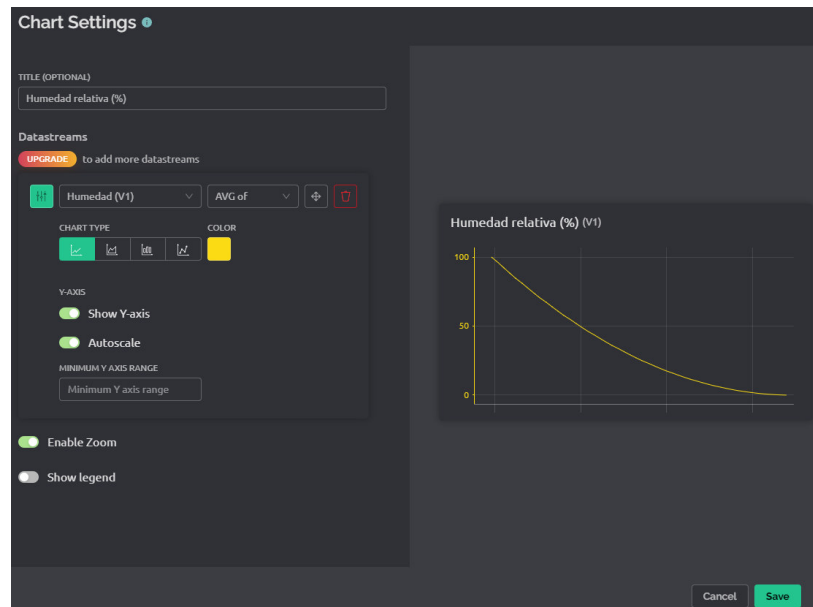
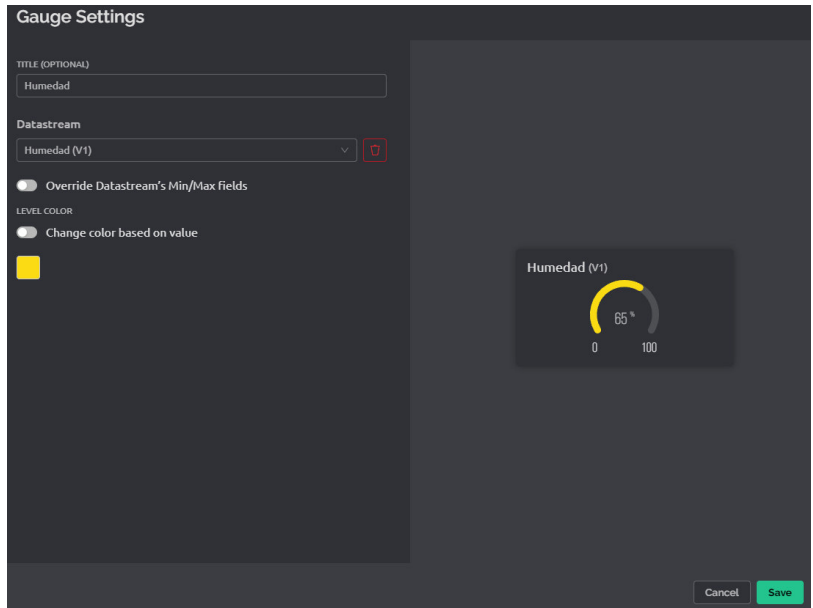


Figura 3.28 Configuración de indicadores de humedad en Blynk

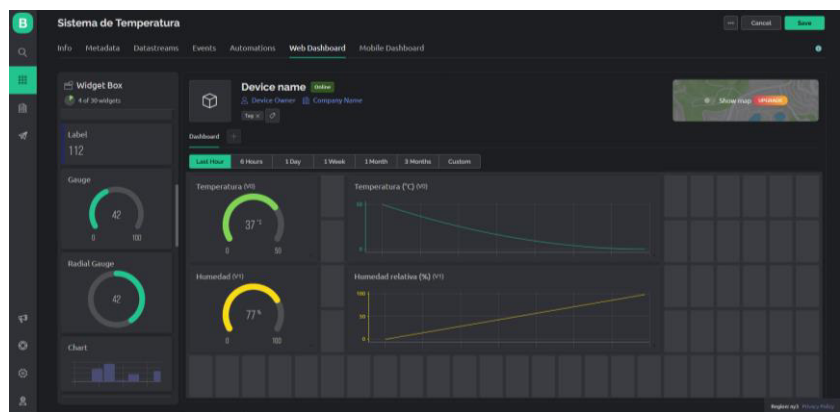
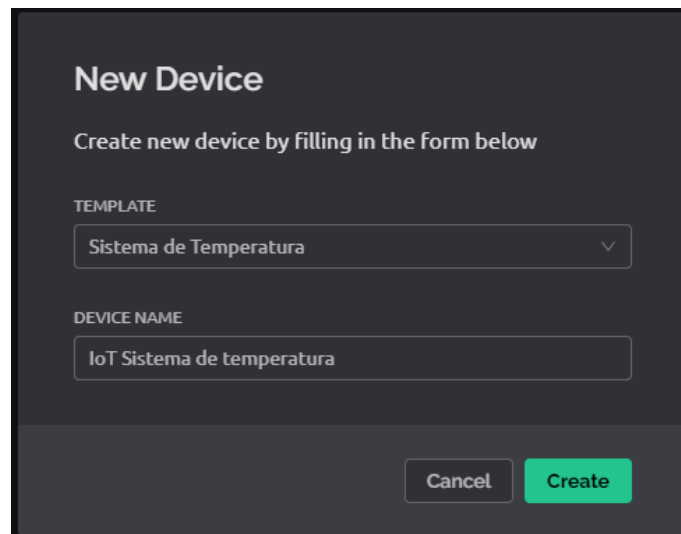


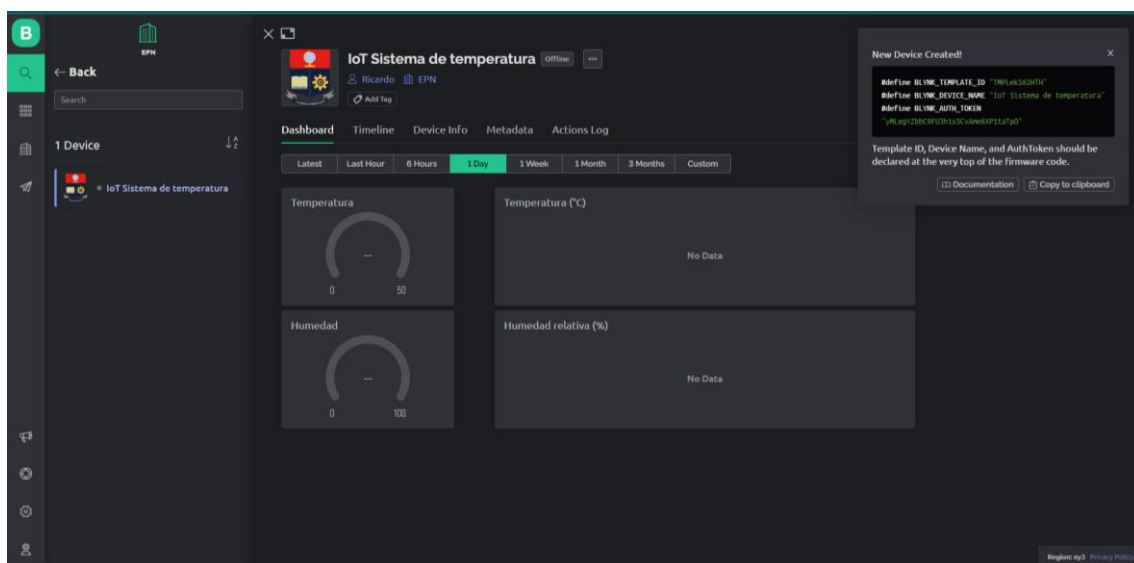
Figura 3.29 Distribución de los elementos en el web dashboard

Al terminar el *template*, se ingresa en “Mis dispositivos” y se crea uno nuevo, empleando el *template* ya generado anteriormente, como se ve en la Figura 3.30.



**Figura 3.30** Creación de un nuevo dispositivo empleando el *template* generado

Se generó el *dashboard* de visualización; la información es transmitida hacia la nube de Blynk y visualizada en la plataforma *web*, como se puede observar en la Figura 3.31.



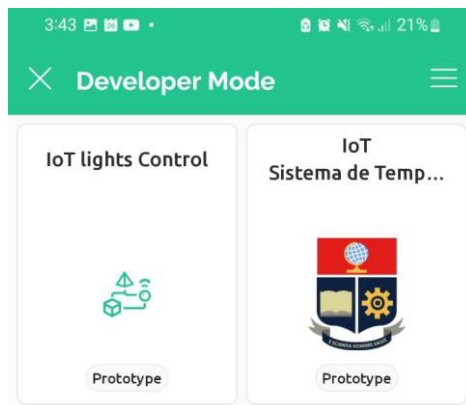
**Figura 3.31** *Dashboard* de visualización de datos generado con éxito

Al generar el dispositivo, Blynk proporciona un conjunto de instrucciones de programación, el código visto en la Figura 3.32 es incorporado directamente en el IDE de Arduino el cual contiene el código de programación del proyecto.

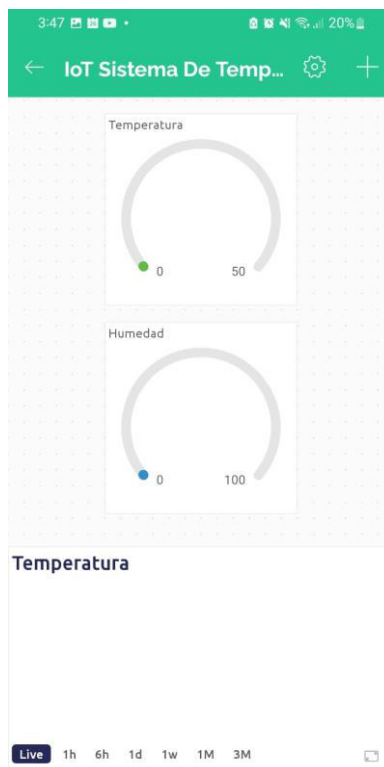
```
#define BLYNK_TEMPLATE_ID "TMPLekS62HTH"
#define BLYNK_DEVICE_NAME "Sistema de Temperatura"
#define BLYNK_AUTH_TOKEN "yMLegYZbbC9FU3h1s5CvAme6XP1taTp0"
```

**Figura 3.32** Códigos identificadores del proyecto y dispositivo

La aplicación móvil Blynk IoT se encuentra disponible en la PlayStore y AppStore; se instala e ingresa las credenciales de la cuenta creada previamente. Al estar dentro de la aplicación, se cambia al modo desarrollador, como se ve en la Figura 3.33. Luego, se ingresa en el *template* y se agrega los *widgets*, tal y como se muestra en la Figura 3.34.



**Figura 3.33** Selección del *template* creado



**Figura 3.34** Configuración y disposición de elementos en aplicativo móvil



Para terminar, se tiene el aplicativo ThingSpeak, el cual requiere de la creación de una cuenta registrada en Matlab; luego se crea y configura el canal, como se ve en la Figura 3.35. El canal contendría los campos de temperatura y humedad del sistema.

Channel Settings

Percentage complete 30%

Channel ID 1751458

Name Sistema de Temperatura

Description

Field 1 Temperatura

Field 2 Humedad

**Figura 3.35** Configuración del canal ThingSpeak

Al crear el canal, se genera un identificador ThingSpeak, visto en la Figura 3.36, el cual sería ingresado en el código de programación del sistema.

Write API Key

Key WXIOYDMVCDSLI781

Generate New Write API Key

**Figura 3.36** Identificador ThingSpeak

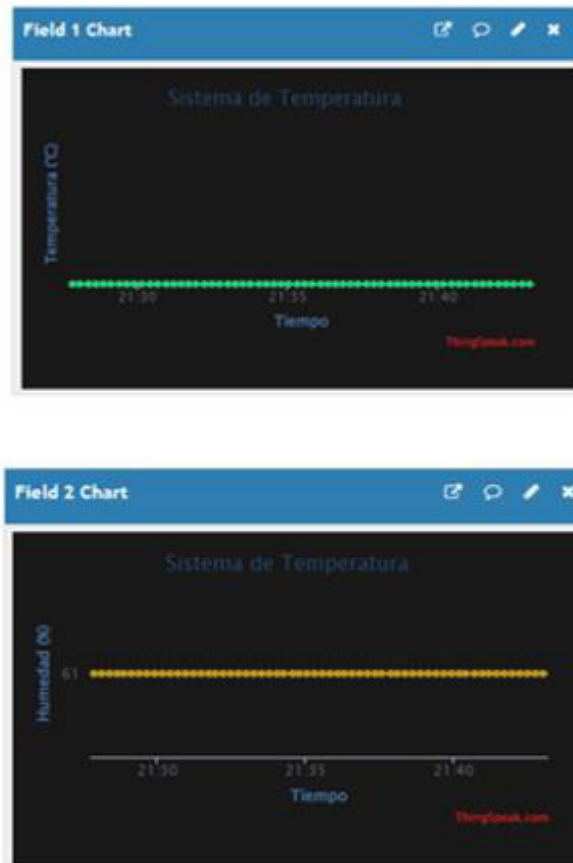
Se genera una visualización pública de los campos, la cual estaría ligada a una cuenta Matlab, obteniendo así la Figura 3.37.

## Channel Stats

Created: [about a month ago](#)

Last entry: [less than a minute ago](#)

Entries: 25959



**Figura 3.37** Configuración y creación de gráficos representativos en ThingSpeak

### **Programación del NodeMCU-ESP8266**

El desarrollo y compilación del programa fue sobre el IDE de Arduino; inicialmente, se establecieron parámetros básicos de funcionamiento mediante el uso de librerías y definición de variables presentadas en la Figura 3.38.

```

#define BLYNK_TEMPLATE_ID "TMPLekS62HTH"
#define BLYNK_DEVICE_NAME "IoT Sistema de Temperatura"
#define BLYNK_AUTH_TOKEN "yMLegYZbbC9FU3hls5CvAme6XP1taTpO"

char auth[] = BLYNK_AUTH_TOKEN;
String apiKey = "WXIOYDMVCDSLI781";
const char* server = "api.thingspeak.com";

#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>

#include <ESP8266WiFi.h>
#include <DHT.h>

// Credenciales Wi-Fi
char ssid[] = "NETLIFE-JAH";
char pass[] = "0913423141";

BlynkTimer timer;
WiFiClient client;

#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

```

**Figura 3.38** Ingreso de librerías y variables

En la Figura 3.39 se puede apreciar el *void sendSensor* empleado en la configuración y validación de parámetros de funcionamiento. La notificación de alerta Blynk del sensor es configurada en este punto; además de otras funcionalidades, como la vista de los datos en el monitor serial.

```

void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
    Serial.println("¡Falló la lectura del sensor DHT11!");
    return;
  }

  if (t >= 25 || t <= 15 )
  {
    Blynk.logEvent("temperaturaext", String("¡Alerta!;Se c
  }

  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);

  // Impresión de temperatura
  Serial.print("Temperatura(°C): ");
  Serial.print(t);
  // Impresión de humedad
  Serial.print("\t Humedad(%) : ");
  Serial.println(h);
}

```

**Figura 3.39** Configuración del Sensor DHT11

Las instrucciones descritas en las líneas de código vistas en la Figura 3.40 muestran los principios de funcionamiento de envío, autenticación y sincronización del sistema.

```

void setup()
{
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  WiFi.begin(ssid, pass);

  dht.begin();
  timer.setInterval(100L, sendSensor);
}

```

**Figura 3.40** Parámetros de funcionamiento del sistema

Finalmente, se tiene la función *void loop* o también denominada bucle principal. En esta función se establecen los parámetros repetitivos claves de funcionamiento del sistema. Se configura la transmisión de datos hacia el cliente ThingSpeak y Blynk, tal y como se ve en la Figura 3.41.

```

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (client.connect(server,80))
  {
    String postStr = apiKey;
    postStr += "sfield1=";
    postStr += String(t);
    postStr += "sfield2=";
    postStr += String(h);
    postStr += "\r\n\r\n";

    //Configuración del cliente ThingSpeak
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    client.print("Content-Type: application/x-www-form-urle");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
  }

  client.stop();
  Blynk.run();
  timer.run();
  delay(1000);
}

```

**Figura 3.41** Transmisión de datos hacia plataformas ThingSpeak y Blynk

### Implementación del prototipo en el invernadero

El dispositivo es instalado en el invernadero de orquídeas de 9 (m<sup>2</sup>). El módulo que contiene al sistema de temperatura y humedad está sujeto contra la pared mediante un colgador tipo cierra. La calidad de la red inalámbrica del domicilio es aceptable y no presenta ninguna falencia de conexión. El dispositivo queda instalado de manera fija y estable al tendido eléctrico del invernadero como se ve en la Figura 3.42.



**Figura 3.42** Instalación del sistema dentro del invernadero

En la Tabla 3.5 se puede ver un listado de los elementos y precios empleados en el desarrollo del proyecto.

**Tabla 3.5** Resumen de costos

Elemento	Precio
NodeMCU-ESP8266	\$8,80
Sensor DHT11	\$4,50
Cargador y cable micro USB	\$6,50
Broca de 1 (mm)	\$1,25
Estaño 1 (m)	\$1,20
Cautín	\$7,00
Crema para soldar	\$2,00
Baquelita de cobre	\$3,00
Cortes láser	\$1,00
Grabado EPN	\$1,00
Acrílico	\$3,00
Colgador tipo serrucho	\$0,35
Total	\$39,60

### 3.5 Pruebas de funcionamiento

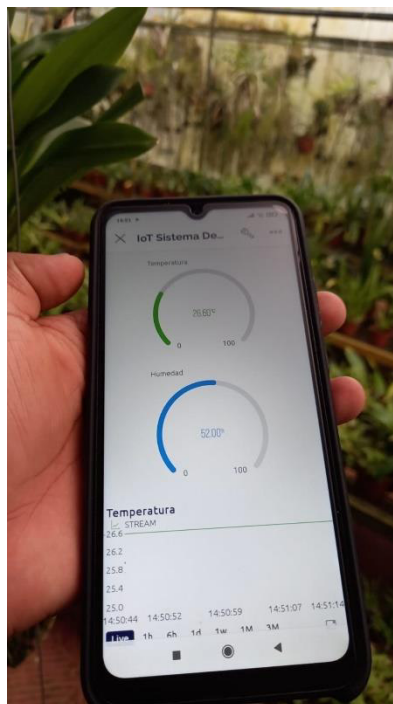
Al terminar con el diseño, elaboración e instalación del sistema, se procede a realizar las pruebas de funcionamiento dentro del orquideario.

El sistema es alimentado mediante el uso de un cargador directamente conectado al tendido eléctrico en un extremo y el otro extremo hacia el puerto *jack* micro-USB de la placa NodeMCU. El dispositivo se encuentra activo al ver el encendido del LED indicador de poder del sensor DHT11, como se puede ver en la Figura 3.43



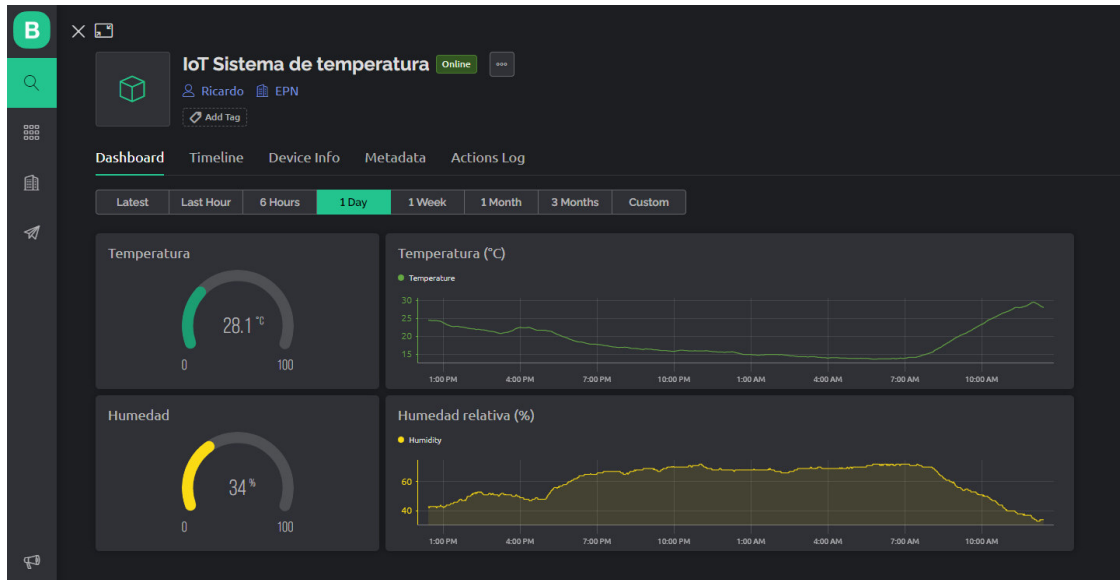
**Figura 3.43** Sistema activo

El sistema comienza a transmitir los datos captados por el sensor de temperatura y humedad empleando la red Wi-Fi del domicilio. La información es visualizada en la aplicación Blynk instalada previamente en el teléfono celular, tal y como se puede ver en la Figura 3.44.



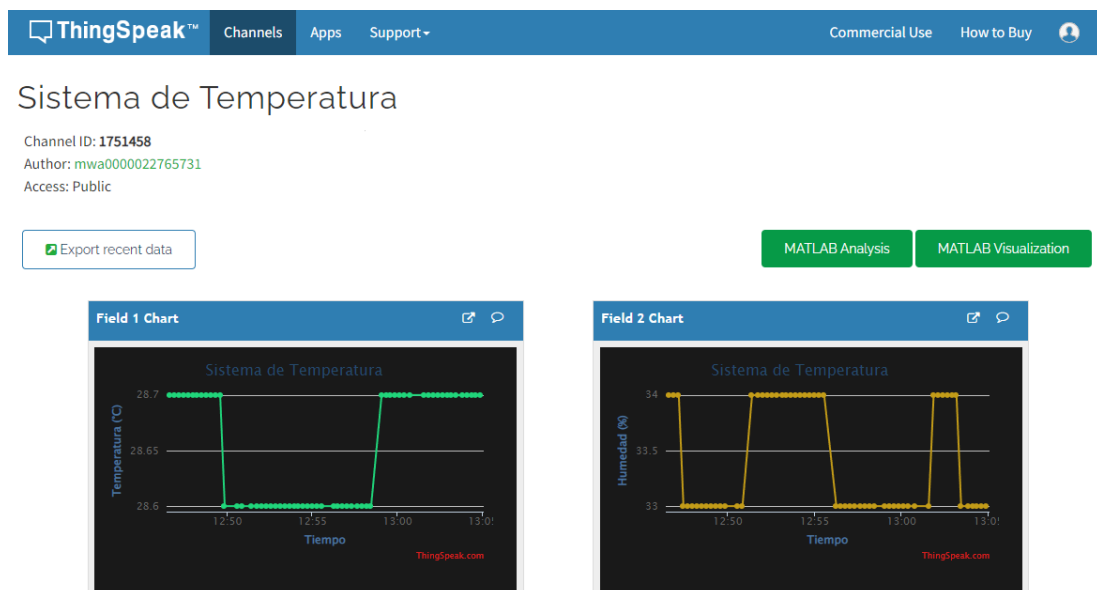
**Figura 3.44** Visualización de los datos en la aplicación móvil Blynk

Es posible visualizar la información de manera remota tanto en la aplicación móvil como interfaz *web*. En la Figura 3.45 se muestra la interfaz gráfica *web* de las gráficas históricas de temperatura y humedad relativa en función del tiempo e íconos indicadores del estado actual del orquideario.



**Figura 3.45** Visualización de los datos en la interfaz gráfica *web* Blynk

Como servicio complemento se tiene a la plataforma ThingSpeak, en la cual es posible visualizar los campos temperatura y humedad en una gráfica generada por Matlab. Al activar la visualización pública de los datos, cualquier persona que tenga el enlace podrá ver o incluso exportar los datos captados en el interior del orquideario. En la Figura 3.46 se muestra la plataforma ThingSpeak.



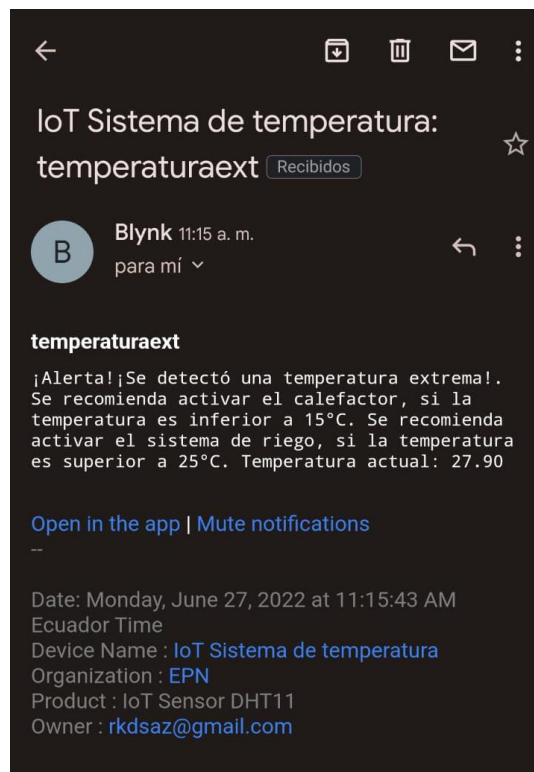
**Figura 3.46** Visualización de los datos en la plataforma ThingSpeak



Se realizó mediciones de temperatura y humedad durante 3 días en el interior del orquideario. Se registraron temperaturas inadecuadas, esto generó el envío de una notificación de alerta Blynk. Como se puede ver en la Figura 3.47 y Figura 3.48, es posible visualizar el texto de la notificación, en la aplicación móvil y correo electrónico.



**Figura 3.47** Notificación de alerta de temperatura en la aplicación Blynk



**Figura 3.48** Notificación de alerta vía correo electrónico

En la Tabla 3.6 se muestran los resultados obtenidos al realizar las pruebas de funcionamiento del sistema dentro del orquideario.

**Tabla 3.6** Evaluación del sistema

Parámetro evaluado	Funcionamiento correcto	Funcionamiento incorrecto
Encendido del sistema	x	
Visualización de datos en la aplicación móvil	x	
Visualización de datos en la <i>web</i>	x	
Emisión y recepción de la notificación de alerta de temperatura	x	

Finalmente, se tiene en el Anexo II.I el código QR del video demostrativo de funcionamiento.

## 4 CONCLUSIONES

- El desarrollo e implementación del sistema de detección de cambios de temperatura empleando el NodeMCU ESP8266 como elemento maestro y el DHT11 sensor captador de temperatura y humedad relativa, permite monitorear en tiempo real las condiciones internas de temperatura y alertar al personal encargado mediante una notificación. Dicha notificación podrá ser visualizada en la *web* o dispositivo celular con acceso a Internet.
- El principio de funcionamiento del proyecto es la comunicación inalámbrica de área local, debido a que la infraestructura de un invernadero es fija y con disponibilidad de una red Wi-Fi residencial. El módulo ESP8266 facilita la transmisión de datos sobre las bandas no licenciadas ISM empleando el estándar 802.11. Este tipo de comunicación permite el envío de información sobre Internet para posteriormente ser almacenada en una nube de gestión IoT.
- La selección del microcontrolador NodeMCU con un módulo ESP8266 integrado, en comparación con ESP32 y ESP-01, cumple con requerimientos técnicos de hardware y software. De este modo, se reducen elementos electrónicos innecesarios para la compatibilidad del sensor DHT11, los cuales solo incrementarían el costo y tamaño del dispositivo de detección de cambios de temperatura y humedad.

- El diseño de la PCB y esquema del circuito fue desarrollado en la herramienta Proteus, luego de instalar las librerías de los elementos electrónicos es posible visualizar la disposición de los elementos en 3D. El software SketchUp web permitió generar el primer bosquejo del módulo protector de componentes, permitiendo la simulación de espacios y materiales a emplear en el módulo.
- La transferencia térmica del circuito impreso fue desarrollada con normalidad al emplear el método del planchado. El tamaño de los agujeros necesarios para los componentes requiere de una broca de 1 (mm). Así mismo, para la fabricación del módulo protector se empleó acrílico de alta resistencia; fue necesario diseñar una caja en AutoCAD. Las piezas fueron cortadas en una máquina láser de alta precisión y armada con silicona caliente.
- Se verificó el funcionamiento del sistema por una semana en el interior del orquideario. Se comparó el número de notificaciones con los gráficos históricos de temperatura y se comprobó la exactitud del sistema.

## **5 RECOMENDACIONES**

- Se recomienda complementar el diseño, con la implementación de sistemas actuadores de entorno que permitirán modificar la temperatura y humedad del orquideario al instalar ventiladores, calefactores y sistema de riego, los cuales serían activados de manera remota mediante la plataforma de control.
- Se sugiere implementar un listado de procedimientos a realizar al detectar un cambio anormal de temperatura. Este listado sería programado en función de las necesidades de cada especie.
- Se debería incorporar una batería recargable LiPo de 11.1(V) al sistema, lo cual mejorará la versatilidad del diseño, el dispositivo ya no requeriría un tomacorriente cercano.
- Se recomienda verificar el funcionamiento del dispositivo durante una semana; de esta manera, se podrá modificar la frecuencia con la cual se estaría enviando la notificación de alerta.
- Se recomienda generar un evento personalizado en el aplicativo Blynk para facilitar la programación de la notificación de alerta, ya que este método permitiría comparar la información, al implementar una sola función Blynk en el código de programación.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] "PNG y SVG de grafico de barras con fondo transparente para descargar." <https://es.vexels.com/svg-png/grafico-de-barras/> (accessed Jun. 28, 2022).
- [2] "Pinterest." <https://co.pinterest.com/pin/805370345855939428/> (accessed Jun. 28, 2022).
- [3] "Cómo solucionar los problemas de conectar a la red WiFi de casa." <https://www.redeszone.net/tutoriales/redes-wifi/problemas-conectar-wifi-casa-soluciones/> (accessed Jun. 28, 2022).
- [4] "ARDUINO - PDF Descargar libre." <https://docplayer.es/58891497-Arduino.html> (accessed May 18, 2022).
- [5] C. Revisión, "NORMA TÉCNICA ECUATORIANA NTE INEN 1:2013".
- [6] "Clima de Ecuador." <http://www.ecuador-viaje.com/clima.html> (accessed May 18, 2022).
- [7] "Quito clima: temperatura, precipitaciones, cuando ir." <https://www.climieviaggi.it/clima/ecuador/quito> (accessed May 18, 2022).
- [8] "Clima de un Invernadero. ¿Cómo conseguir la Temperatura Ideal?" <https://www.novagric.com/es/blog/articulos/clima-invernadero-como-conseguir-temperatura-ideal> (accessed May 28, 2022).
- [9] O. de la Junta Parroquial Río Verde En La Provincia De Tungurahua and M. Quitiaquez Jenniffer Fernanda, ""SISTEMA ELECTRÓNICO DE CONTROL Y MONITOREO REMOTO PARA EL".
- [10] J. Salazar, "REDES INALÁMBRICAS", Accessed: May 18, 2022. [Online]. Available: <http://www.techpedia.eu>
- [11] "Guía para configurar un ESP-01, el módulo WiFi basado en ESP8266." <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/> (accessed May 18, 2022).
- [12] "idUS - Puesta a punto de un sistema inalámbrico de bajo coste para análisis vibratorios: MPU6050 y módulo ESP8266." <https://idus.us.es/handle/11441/126239> (accessed May 18, 2022).

- [13] "Módulo WiFi Serial WSP8266 ESP8266 p/PCB." <https://hetprostore.com/modulo-wifi-serial-wsp8266-esp8266-p-pcb/> (accessed Jun. 08, 2022).
- [14] L. Corona, G. Abarca, and J. Mares, "Sensores y actuadores: aplicaciones con Arduino (2a. ed.)," p. 321, 2019, Accessed: May 18, 2022. [Online]. Available: <https://elibro-net.ezproxy.utem.cl/es/ereader/utem/121284>
- [15] D. Srivastava, A. Kesarwani, and S. Dubey, "Measurement of Temperature and Humidity by using Arduino Tool and DHT11," *International Research Journal of Engineering and Technology*, vol. 876, 2008, Accessed: May 18, 2022. [Online]. Available: [www.irjet.net](http://www.irjet.net)
- [16] "Sensor de temperatura y humedad relativa DHT11." <https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html> (accessed May 18, 2022).
- [17] A. Moreno and S. Córcoles, "Arduino: curso práctico :edición 2018," p. 452, 2018, Accessed: May 18, 2022. [Online]. Available: <https://elibro-net.ezproxy.utem.cl/es/ereader/utem/106517>
- [18] "IoT Analytics - ThingSpeak Internet of Things." <https://thingspeak.com/> (accessed May 18, 2022).
- [19] "Developers | Blynk IoT platform." <https://blynk.io/en/developers> (accessed May 18, 2022).
- [20] "Blynk | Aprendiendo Arduino." <https://aprendiendoarduino.wordpress.com/category/blynk/> (accessed May 18, 2022).
- [21] O. Eduardo. Gutiérrez, "Comunicaciones móviles y redes inalámbricas," 2020.
- [22] "NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet." <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet> (accessed May 23, 2022).
- [23] "ESP32 DevKitC Pinout, Overview, Features & Datasheet." <https://components101.com/microcontrollers/esp32-devkitc> (accessed May 23, 2022).
- [24] "ESP-01 pdf, ESP-01 Description, ESP-01 Datasheet, ESP-01 view ::: ALLDATASHEET :::." <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179098/ETC2/ESP-01.html> (accessed May 23, 2022).

- [25] "ESP32 Series Datasheet Including," 2018, Accessed: May 23, 2022. [Online]. Available: [www.espressif.com/en/subscribe](http://www.espressif.com/en/subscribe).
- [26] "DHT11 Sensor Datasheet pdf - Temperature Sensor. Equivalent, Catalog." <https://datasheetspdf.com/pdf/792210/ABCPROYECTOS/DHT11/1> (accessed May 23, 2022).
- [27] "LM35 pdf, LM35 Description, LM35 Datasheet, LM35 view :: ALLDATASHEET ::" <https://pdf1.alldatasheet.com/datasheet-pdf/view/1188696/TGS/LM35.html> (accessed May 23, 2022).
- [28] "Arduino Cloud IoT | Arduino Documentation | Arduino Documentation." <https://docs.arduino.cc/cloud/iot-cloud> (accessed May 24, 2022).
- [29] "IoT platform | Internet of Things | Ubidots." <https://ubidots.com/> (accessed Jun. 08, 2022).
- [30] "AskSensors IoT platform | The easiest application to manage your IoT devices and data in the cloud." <https://asksensors.com/index.html> (accessed Jun. 08, 2022).
- [31] "Hoja de datos del sensor de temperatura y humedad DHT11 | Robots Didácticos." <https://robots-argentina.com.ar/hoja-de-datos-del-sensor-de-temperatura-y-humedad-dth11.htm> (accessed Jun. 07, 2022).
- [32] "Arduino y solo Arduino - Todo lo que necesitas lo encontrarás aquí.: Sensor de temperatura y humedad - Keyes KY015 - Parte 1." <https://soloarduino.blogspot.com/2013/12/sensor-de-temperatura-y-humedad-keyes.html> (accessed Jun. 07, 2022).
- [33] G. Arenas Sicard, "Electricidad y magnetismo," 2008.
- [34] "Resistencia de LED, Como calcularla en función a la corriente - HeTPro." <https://hetpro-store.com/TUTORIALES/resistencia-de-led/> (accessed Jun. 07, 2022).
- [35] "3D Design Software | 3D Modeling on the Web | SketchUp." <https://www.sketchup.com/> (accessed Jun. 08, 2022).

## 7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Códigos Fuente

## **ANEXO I: Certificado de Originalidad**

### **CERTIFICADO DE ORIGINALIDAD**

Quito, D.M. 20 de agosto de 2022

De mi consideración:

Yo, FANNY PAULINA FLORES ESTÉVEZ, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA DE ALERTA POR DETECCIÓN DE CAMBIOS DE TEMPERATURA elaborado por el estudiante RICARDO FERNANDO SIMBAÑA CASCUERAN de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

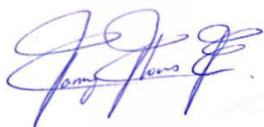
El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

[Turnitin TIC2022A Ricardo Simbaña.pdf](#)

Atentamente,



FANNY PAULINA FLORES ESTÉVEZ

Docente

Escuela de Formación de Tecnólogos



## ANEXO II: Enlaces



**Anexo II.I** Código QR de la implementación y pruebas de funcionamiento

## ANEXO III: Códigos Fuente

```
#define BLYNK_TEMPLATE_ID "TMPLekS62HTH"

#define BLYNK_DEVICE_NAME "IoT Sistema de Temperatura"

#define BLYNK_AUTH_TOKEN "yMLegYZbbC9FU3h1s5CvAme6XP1taTpO"

char auth[] = BLYNK_AUTH_TOKEN;

String apiKey = "WXIOYDMVCDSLI781";

const char* server = "api.thingspeak.com";

#define BLYNK_PRINT Serial

#include <BlynkSimpleEsp8266.h>

#include <ESP8266WiFi.h>

#include <DHT.h>

// Credenciales Wi-Fi

char ssid[] = "NETLIFE-JAH";

char pass[] = "0913423141";

BlynkTimer timer;

WiFiClient client;

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void sendSensor()
```

```

{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
    Serial.println("¡Fallo en la lectura del sensor DHT11!");
    return;
  }

  // Notificación de alerta Blynk
  if (t <= 15 || t >= 25)
  {
    Blynk.logEvent("temperaturaext", String("¡Alerta! Se detectó una temperatura
    extrema!. Se recomienda activar el calefactor, si la temperatura es inferior a 15°C. Se
    recomienda activar el sistema de riego, si la temperatura es superior a 25°C.
    Temperatura actual: ") + t);
  }

  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);

  Serial.print("Temperatura : ");
  Serial.print(t);
  Serial.print("°C\t");
  Serial.print("Humedad : ");
  Serial.println(h);
  Serial.print("%");

```

```

}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  WiFi.begin(ssid, pass);
  dht.begin();
  timer.setInterval(100L, sendSensor);
}

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (client.connect(server,80))
  {
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(t);
    postStr += "&field2=";
    postStr += String(h);
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
  }
}

```

```
client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");  
  
client.print("Content-Type: application/x-www-form-urlencoded\n");  
  
client.print("Content-Length: ");  
  
client.print(postStr.length());  
  
client.print("\n\n");  
  
client.print(postStr);  
  
}  
  
  
client.stop();  
  
Blynk.run();  
  
timer.run();  
  
delay(1000);  
  
}
```