

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL DE INVERNADERO POR MEDIO DE COMANDOS DE VOZ CON ALEXA**

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

**STALIN JEFFERSON MALES MALDONADO**

**DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.**

**DMQ, marzo 2022**

## **CERTIFICACIONES**

Yo, Stalin Jefferson Males Maldonado 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.

---

**Stalin Jefferson Males Maldonado**

**stalin.males@epn.edu.ec**

**kofflml1999@outlook.es**

Certifico que el presente trabajo de integración curricular fue desarrollado por Stalin Jefferson Males Maldonado, bajo mi supervisión.

---

**Leandro Antonio Pazmiño Ortiz**

**DIRECTOR**

**leandro.pazmino@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.

Stalin Males

## **DEDICATORIA**

Este desarrollo de este documento y proyecto está dedicado a mi familia, principalmente a mis padres y hermanos que supieron apoyarme y guiarme día a día tanto durante mi carrera estudiantil y el objetivo propuesto para este proyecto. Finalmente, a mis compañeros y amigos de la universidad que me supieron compartir sus conocimientos todos los días en las aulas, además del apoyo por algún día lograr este objetivo grande que es culminar con la carrera universitaria.

## **AGRADECIMIENTO**

Un agradecimiento enorme a los profesores que en todo el proceso universitario supieron brindarme sus enseñanzas todos los días en las aulas de la universidad e incluso por medio de plataforma virtuales. Finalmente, un agradecimiento a mi tutor que clase tras clase supo orientarme y apoyarme para culminar satisfactoriamente el desarrollo de este proyecto.

## ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VI
ABSTRACT .....	VII
1 INTRODUCCIÓN .....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos .....	2
1.3 Alcance .....	2
1.4 Marco teórico .....	2
2 METODOLOGÍA .....	6
2.1 Descripción de la mitología usada .....	6
3 RESULTADOS.....	8
3.1 Identificación de los requerimientos necesarios para el diseño del prototipo... 8	
3.2 Selección del hardware y <i>software</i> del prototipo.....	11
3.3 Diseño del sistema del prototipo .....	15
3.4 Implementación del prototipo .....	41
3.5 Pruebas de funcionamiento del prototipo.....	45
4 CONCLUSIONES .....	50
5 RECOMENDACIONES.....	52
6 REFERENCIAS BIBLIOGRÁFICAS.....	53
7 ANEXOS .....	55
ANEXO I.....	56

## RESUMEN

En la primera sección se tiene una breve introducción del documento, además se plantea el objetivo general como los objetivos específicos que se buscan lograr con la implementación del proyecto. Para lo cual, se muestra conceptos fundamentales que se utilizaron para el desarrollo del proyecto como el sensor de humedad y temperatura, los microcontroladores, módulo *Wi-Fi* y GSM, entre otros que ayudan al funcionamiento correcto del control de invernadero.

En la segunda sección se tiene la descripción de la metodología del proyecto, en donde se detalla brevemente el procedimiento que se realizó para cumplir con los requerimientos de cada uno de los objetivos específicos que conforman el desarrollo del proyecto de inicio a fin.

En la tercera sección denominada resultados y discusión, se describe el procedimiento de manera detallada de los requerimientos que fueron necesarios para implementar el proyecto, tanto en la parte de *software* como en hardware, es decir que se describe los circuitos que fueron requeridos y el código de programación que fue desarrollado tanto para la parte del circuito como para la comunicación con Alexa. Finalmente, se muestra los resultados de las pruebas de funcionamiento

En la cuarta sección se tiene las conclusiones que fueron descritas en base a la experiencia adquirida durante el desarrollo del proyecto y recomendaciones que ayudaran a mejorar el funcionamiento del proyecto para implementarlo en un escenario real.

Finalmente se tiene la documentación correspondiente a los anexos, que comprenden el código de programación y el video de funcionamiento que tienen como objetivo facilitar la comprensión y uso del proyecto.

**PALABRAS CLAVE:** Arduino, sensores, *Blynk*, invernadero, Alexa.

## ABSTRACT

*The first chapter is a brief introduction of the document, in addition to the general objective and the specific objectives to be achieved with the implementation of the project. For which, fundamental concepts that were used for the development of the project are shown, such as the humidity and temperature sensor, microcontrollers, Wi-Fi module and GSM module, among others that help the correct operation of the greenhouse control.*

*The second chapter describes the methodology of the project, which briefly details the procedure that was carried out to meet the requirements of each of the specific objectives that make up the development of the project from start to finish.*

*In the third chapter called results and discussion, the procedure is described in detail of the requirements that were necessary to implement the project, both in software and hardware, i.e., the circuits that were necessary and the programming code that was developed for both the circuit part and for communication with Alexa are described. Finally, the results of the functional tests are shown.*

*In the fourth chapter we have the conclusions that were described based on the experience obtained during the development of the project and the recommendations that will help to improve the operation of the project to implement it in a real scenario.*

*Finally, we have the documentation corresponding to the annexes, which include the programming code and the operating video that aim to facilitate the understanding and use of the Project.*

**KEYWORDS:** *Arduino, sensors, Blynk, greenhouse, Alexa.*



# 1 INTRODUCCIÓN

La producción agrícola bajo invernadero es una de las técnicas más utilizadas y modernas en la actualidad, debido a que la implementación de un invernadero proporciona la obstrucción del medio ambiente externo y el cultivo agrícola del invernadero [1]. Esta barrera cumple la función de limitar un microclima, el cual permite proteger al cultivo de factores externos perjudiciales como la lluvia, el viento, los animales, plagas, enfermedades, etc. [2]. Además, permite al agricultor controlar la temperatura y humedad, factores climáticos de suma importancia para el crecimiento del cultivo [2].

La idea principal de este proyecto se da por la necesidad en la producción agrícola de monitorear las variaciones climatológicas en invernaderos, con el propósito de lograr tener un control total en el invernadero. Para lo cual es de gran importancia contar con sistemas que permitan mejorar la calidad de producción, optimizar los procesos y manejar un sistema para controlar remotamente los dispositivos electrónicos mediante la tecnología *Wi-Fi*. Además, con el propósito de brindar un servicio más eficiente se incluye la capacidad de poder controlar estos dispositivos electrónicos, y poder saber los valores en tiempo real tanto de la temperatura como de la humedad, por medio de un asistente virtual, utilizando específicamente para este proyecto el denominado Alexa.

El proyecto implementa un prototipo que permite controlar el sistema de riego por goteo para el control de humedad y el sistema de ventilación para el control de temperatura, sin embargo, estos dos factores también provocan variaciones notables entre sí. Por lo cual dependiendo del producto que se va a cultivar en el invernadero los valores de humedad y temperatura pueden variar por lo que este proyecto se permite la posibilidad de configurar los parámetros de temperatura y humedad tope para que se active de manera automática los dispositivos electrónicos mediante la lectura del sensor de humedad y temperatura.

Día tras día, el entorno de vida del ser humano está en constante variación lo que produce la necesidad de controlar el entorno y adaptarlo a la necesidad del ser humano [1]. A esto si se añade el gran problema actual del cambio climático que ha impactado de forma negativa a la productividad y calidad de la producción agrícola [2], explican por qué en la actualidad el control de un invernadero con ayuda de dispositivos electrónicos es de suma importancia. Un invernadero automatizado representa ahorro de dinero y de tiempo ya que varias tareas que realizan los agricultores se están realizando de forma automática.

## 1.1 Objetivo general

Implementar un prototipo de control de invernadero por medio de comandos de voz con Alexa.

## 1.2 Objetivos específicos

1. Identificar los requerimientos para el diseño del prototipo.
2. Seleccionar el hardware y *software* que conforman el prototipo.
3. Diseñar el sistema del prototipo.
4. Implementar el prototipo de control de invernadero.
5. Realizar pruebas de funcionamiento del prototipo.

## 1.3 Alcance

Por medio del presente proyecto se busca implementar un prototipo para realizar el control de la temperatura y humedad de un invernadero, de tal manera que se ejecuten acciones automáticas como:

- Activación del sistema de ventilación o del sistema de riego por goteo para mantener la temperatura y humedad en base a parámetros configurables.
- El sistema de ventilación y de riego para controlar la temperatura y humedad serán activados por medio de comandos de voz por medio del asistente virtual Alexa.
- Los parámetros de temperatura y humedad serán monitoreados y enviados a través de la red celular por medio de mensajes de texto, conocido como SMS, a un número telefónico determinado.
- Los parámetros de temperatura y humedad serán monitoreados y configurados a través de la aplicación móvil.

## 1.4 Marco teórico

### Concepto de invernadero

Un invernadero es una estructura construida a base de vidrio o plástico, en donde en el interior se cultivan plantas a factores climáticos distintos a la del exterior, formando un microclima para que la planta alcance su completo desarrollo incluso cuando se encuentra

fuera de temporada. También cumple la función de proteger al cultivo de factores externos como plagas, viento, insectos y enfermedades [3].

### **Variables relacionadas en el funcionamiento de un invernadero**

El desarrollo óptimo del cultivo en un invernadero depende de diversos factores, tales como, temperatura, humedad relativa, concentración de gases, radiación solar, humedad del suelo. Para que las plantas desarrollen sus funcionalidades es necesario mantener estos factores climáticos controlados en función de los valores óptimos para su crecimiento, ya que si no se controla estos factores las plantas incluso pueden llegar a morir [3].

### **Arduino Uno**

Arduino Uno es una placa que fue hecho en base al microcontrolador ATmega328P, que contiene 14 que cumplen la función de entrada y salida digital, 6 entradas analógicas, un conector USB, un conector Jack para alimentación y un botón que permite el reinicio de la placa. Contiene toda la electrónica necesaria para que la placa funcione, por lo cual únicamente se lo debe conectar a la energía por el puerto USB que contiene [4]. Arduino está basado en hardware y *software* libre, por lo cual es una plataforma de código abierto y mediante el microcontrolador que tiene incorporado se puede grabar instrucciones a través del lenguaje de programación que maneja Arduino IDE. Los programas que son creados en la placa Arduino permiten interactuar con los circuitos o dispositivos electrónicos que se encuentren conectados a la placa [5].

### **Modulo NodeMCU V3 ESP8266**

El módulo NodeMCU es una placa *Wi-Fi* que se encuentra montada en conjunto con el chip ESP8266 para que este pueda ser conectado a la red internet en combinación con las funciones de un microcontrolador [6]. En similitud con la placa Arduino, el módulo NodeMCU funciona conectándolo a través del puerto USB y puede ser programado mediante la plataforma LUA o Arduino IDE.

El chip ESP8266 contiene múltiples funcionalidades como procesamiento y almacenamiento debido a que permite la capacidad de integrarse con sensores y dispositivos electrónicos específicos a través de sus pines GPIO que sirven como pines digitales y un pin analógico [6]. Entre las especificaciones principales se tiene, que trabaja en las tecnologías *Wi-Fi 802.11 b/g/n*, contiene 9 pines GPIO con I2C y SPI, contiene un pin analógico y un botón de *reset* [6].

## **Sensor de temperatura y humedad relativa DHT22**

El sensor DHT22 es un sensor digital que permite medir la temperatura y humedad relativa en tiempo real, gracias a que este integra un termistor para medir el aire circundante en la localidad ubicada. Permite obtener los datos de temperatura a través de una señal digital saliente del pin de datos. Este sensor opera de 3 (V) a 5 (V), mide la temperatura en un rango de -40 (°C) a 80 (°C) con una precisión de lectura del  $\pm 0.5$  (°C), por último, se tiene que la lectura en tiempo real lo realiza cada 2 (s) [7].

## **Sensor de humedad de suelo FC-28**

El sensor de humedad de suelo FC-28 permite medir la humedad de cualquier tipo de suelo mediante 2 electrodos resistivos insertados en el suelo a ser medido. El valor de resistencia entre los electrodos dependerá de la cantidad de humedad del suelo, por lo cual el sensor toma como resistencia muy baja un suelo considerablemente húmedo y como valor de resistencia muy alto un suelo seco [8].

## ***Blynk***

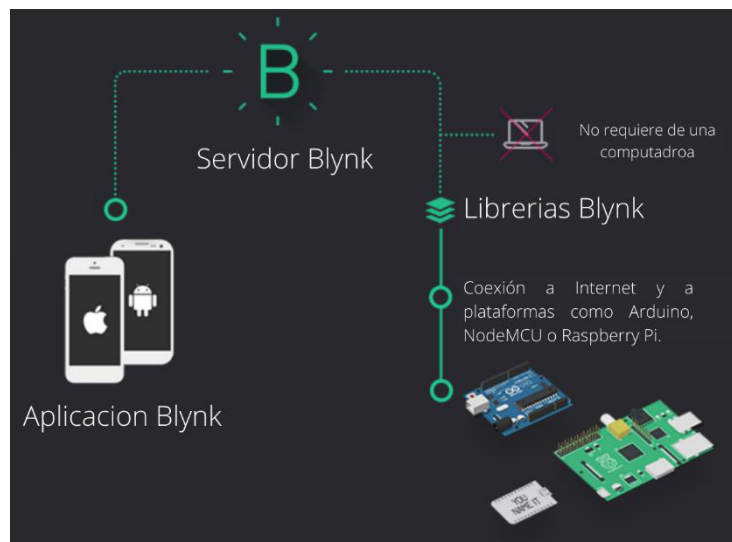
*Blynk* es una plataforma que fue ideado para aplicaciones referentes con el internet de las cosas (IoT). La plataforma permite controlar hardware de manera remota, mostrar resultados de censado en tiempo real, almacenar datos, enviar notificaciones, entre otras aplicaciones prácticas más. *Blynk* es muy utilizado para proyectos de IoT debido a que permite incorporar funcionalidades en conjunto con Arduino, Raspberry Pi, módulos *Wi-Fi* u otras tarjetas o placas similares [9].

En el teléfono móvil, la plataforma *Blynk*, permite crear una aplicación de con una interfaz de control fácil de manejar, en donde se selecciona el dispositivo con el que se va a interactuar, en la Figura 1.1 se muestra la representación del funcionamiento de la plataforma *Blynk*. Esta interacción es posible debido a tres componentes principales que maneja *Blynk*, que se detallan a continuación [9]:

- Aplicación *Blynk*: este componente permite crear aplicaciones móviles mediante una interfaz simple utilizando los distintos *widgets* que proporciona la aplicación.
- Servidor *Blynk*: este componente es el responsable de la comunicación entre la aplicación móvil desarrollada en el dispositivo móvil inteligente y el hardware correspondiente. Se puede utilizar la nube de *Blynk*, incluso montar localmente un servidor *Blynk* privado. El servidor *Blynk* es de código abierto por lo cual es

completamente compatible con dispositivos que interactúan con un módulo NodeMCU.

- Bibliotecas *Blynk*: este componente siempre se mantiene actualizado para cualquier plataforma en cuanto a hardware, permitiendo la comunicación entre el servidor y la aplicación *Blynk*.



**Figura 1.1** Funcionamiento de la plataforma *Blynk* [9]

### ***Voiceflow***

Es una plataforma que concede a los usuarios la capacidad de crear sus propias aplicaciones de voz, denominadas *skills*, para el asistente virtual Alexa y Google *Assistant*. *Voiceflow* está diseñado con el propósito de no ser muy técnico y lograr que la programación sea muy fácil de manejar. Por ejemplo, los dos bloques más básicos de programación se denominan “hablar” y “elección” y pueden ser organizados simplemente arrastrándolos y soltando en el espacio de trabajo, es decir, que la programación en *Voiceflow* es más visual que técnico [10].

### ***Shield SIM900 GSM/GPRS***

El *shield* SIM900 GSM/GPRS permite conectar de forma sencilla proyectos desarrollados en placas como Arduino a la red celular. El módulo SIM900 permite enviar mensajes de texto (SMS), llamadas de voz e incluso la capacidad de conectarse a la red Internet mediante la tecnología GPRS. El *shield* permite trabajar en bandas de frecuencia GSM de 850, 900, 1800, y 1900 (MHz).

El módulo SIM900 es programado mediante comandos AT a través del serial UART, para lo cual se debe conectar el *shield* con el microcontrolador mediante los pines seriales para la comunicación serial [11].

### **Módulo relé 2 canales a 5 (V)**

Este módulo permite controlar componentes electrónicos de alto voltaje o corriente que no pueden ser manejados específicamente con placas como Arduino, NodeMCU, entre otras. Por lo cual es conveniente implementar este tipo de relés que permiten controlar cargas de alto voltaje o amperaje mediante señales pequeñas. La salida del módulo relé comúnmente es activado al recibir 0 (V) y desactiva la salida al recibir 5 (V) [12].

El módulo relé permite trabajar con cargas de luminarias, electroválvulas, bombillas de luz, calentadores de agua, ventiladores, motores y una gran variedad de actuadores más [12].

## **2 METODOLOGÍA**

### **2.1 Descripción de la metodología usada**

#### **Objetivo 1: Identificar los requerimientos para el diseño del sistema**

Se analizó el escenario donde el prototipo será implementado, de esta manera se establecieron las características que el prototipo debe tener y en base a esto se identificarán los dispositivos a nivel de hardware que se requieran para satisfacer de manera adecuada las necesidades del sistema. Se determinó que el invernadero será implementado en la región Sierra del Ecuador, el cual se caracteriza por un clima tropical muy húmedo, semihúmedo, húmedo, cálido y seco dependiendo de la zona siendo la humedad y la temperatura los factores de control más importantes para la implementación del proyecto [1]. En cuanto al desarrollo de *software* se analizó un *software* libre que permite crear el código fuente y el código para la comunicación con Alexa que satisfaga las necesidad y requerimientos del prototipo.

El prototipo debe enviar mensajes de texto cuando un factor de clima se encuentre sobre el nivel permitido, por lo cual se optó la comunicación GSM para cumplir este propósito. Finalmente se investigó los dispositivos electrónicos necesarios para el funcionamiento del prototipo tales como sensores, *shields*, fuentes de voltaje, microcontroladores, entre otros con la finalidad de adquirir los componentes electrónicos adecuados que se adapten de la mejor manera al proyecto.

## **Objetivo 2: Seleccionar el hardware y *software* requerido para el prototipo**

Una vez establecidos los requerimientos del hardware requerido para el prototipo se realizará un análisis para determinar los dispositivos que satisfagan estos requerimientos de la manera más adecuada y en función de eso se hará la selección de estos. Para lo cual se podrán considerar a parte de las características técnicas, factores como: costo, facilidad de uso, tamaño y calidad.

Para la parte de medición de temperatura y humedad se investigó el sensor más adecuado que se adapte al propósito del proyecto, permitiendo la medición en tiempo real y con los menores errores posibles. Por otro lado, para el envío de mensajes de texto SMS se determinó la opción de un módulo de telefonía móvil GSM/GPRS, el cual envíe a través de la red celular un mensaje a un número determinado.

Finalmente, para la comunicación con Alexa se investigó los dispositivos en los cuales se integran los servicios de Amazon que permitan cumplir con el objetivo principal del proyecto.

## **Objetivo 3: Diseñar el sistema del prototipo**

Para el diseño del sistema se realizó una investigación para determinar que plataformas se emplearan para el reconocimiento de voz, así como para el desarrollo de la aplicación. Adicionalmente se hizo el desarrollo del código mediante bloques para reconocer los diferentes comandos de voz mediante Alexa para que proporcione información y permitir la activación o desactivación de los dispositivos electrónicos, además de los códigos de los programas que deberán ser programados en las placas de desarrollo o microcontroladores para el sistema automatizado y el envío de mensajes de texto.

Finalmente, se creó la aplicación móvil que sirve como intermediario entre el sistema de control de invernadero y el asistente virtual Alexa, y que estos se puedan comunicar sin problemas.

## **Objetivo 4: Implementar el prototipo**

Una vez seleccionado el hardware y desarrollado los códigos de programación y bloques para el prototipo se realizó la implementación del sistema para lo cual se unió las dos partes previamente desarrolladas y se montó sobre una maqueta que fue construido para evidenciar el funcionamiento del prototipo del sistema.

## **Objetivo 5: Realizar pruebas de funcionamiento del prototipo**

Con el prototipo ya implementado se realizó pruebas de funcionamiento que validen el correcto desempeño del prototipo, además estas pruebas sirvieron para corregir errores que se presentaron.

Se verifico que la comunicación entre el sistema en el invernadero y el asistente virtual Alexa funcione correctamente, es decir, que reconozca los comandos de voz que fueron programados para el prototipo. Finalmente se comprobó que los mensajes de texto se envíen a través de la red de telefonía móvil con el propósito de garantizar que el prototipo cumpla con todos los requerimientos establecidos.

## **3 RESULTADOS**

En esta sección del documento se explica la forma en la que se identificaron y seleccionaron los requerimientos del prototipo, el diseño del circuito para el sistema automatizado refiriéndose tanto al *software* como al *hardware*. Además, se describe de manera detallada el código de programación de las placas en Arduino IDE, el código de programación en bloques que se utilizó para el asistente virtual Alexa, de la misma forma como se desarrolló la aplicación móvil para que sea posible la comunicación entre el microcontrolador y Alexa. Finalmente se muestran las pruebas de funcionamiento del prototipo finalizado.

### **3.1 Identificación y selección de los requerimientos necesarios para el diseño del prototipo**

#### **Selección de los factores climáticos para el control de invernadero**

Para cumplir esto fue necesario investigar detalladamente sobre factores climáticos y características que debe cumplir un invernadero controlado, siendo más importante conocer los factores climáticos que más afectan al cultivo en la región sierra del Ecuador, ya que con estos datos se seleccionó y diseño el sensor más adecuado para el control del invernadero.

Cabe recalcar que la región Sierra, como ya se mencionó, se caracteriza por tener un clima tropical, siendo muy húmedo en las zonas cercanas a la región litoral y amazonia, húmedo y semihúmedo en las zonas interandinas, por último, cálido y seco por los valles interandinos [1]. Tomando esto en cuenta se establecieron como factores climáticos importantes para el control de invernadero la humedad y la temperatura. Se destaca que en el prototipo se puede modificar la temperatura y humedad tope para que se activen los



dispositivos electrónicos por lo que se puede cultivar cualquier tipo de fruto o grano que se cultive en la región Sierra del Ecuador.

### **Selección del método de control de humedad y temperatura en el invernadero**

La temperatura óptima en un invernadero depende del cultivo y etapa de desarrollo en el que se encuentre el mismo. Normalmente la temperatura máxima permitida por las plantas en un invernadero es de 30 grados centígrados, mientras que la temperatura mínima requerida es de 10 grados centígrados [13].

Para controlar la temperatura de un invernadero se lo puede realizar mediante sistemas de calefacción o sistemas de ventilación [13]. Los sistemas de calefacción pueden darse mediante dos formas, que son: por agua caliente y por generadores de aire caliente [14]. Sin embargo, para implementar este tipo de sistema para el control de temperatura se requiere de altos costos por la maquinaria que conlleva la instalación de este tipo de sistemas en un invernadero.

Por otro lado, los sistemas de ventilación en un invernadero son de gran ayuda sin importar el tiempo o variación de clima que haga dentro del mismo. La ventilación es de suma importancia debido a que este permite expulsar el aire caliente que se encuentra circulando dentro del invernadero y sustituir ese aire caliente por otra masa de aire frío proveniente del exterior del invernadero [15]. Logrando la disminución en gran parte de la sobrecarga de calor y disminuyendo la temperatura del invernadero, además que se modifica la humedad y la concentración de gases [15].

En la Tabla 3.1 se muestra una comparativa entre ambos sistemas en donde se puede observar que la implementación de un sistema de calefacción para un invernadero trae consigo algunas desventajas si se la compara con un sistema de ventilación, por lo cual para el desarrollo del proyecto se optó por implementar el sistema de ventilación. Además de añadir que un sistema de calefacción sería más factible implementar en zonas donde la temperatura sea muy baja.

**Tabla 3.1** Comparativa entre sistema de calefacción y sistema de ventilación [13]

Sistema de calefacción	Sistema de ventilación
El costo de implementación es elevado	Se requiere un presupuesto bajo para su implementación

Ciertos sistemas de calefacción requieren de gas para funcionar	No se requiere ningún elemento extra para su funcionamiento
Requiere de 5 o más mantenimientos al año	El mantenimiento se lo puede realizar dos o tres veces al año
Requiere de un consumo de energía eléctrica elevada	El consumo de energía eléctrica es óptimo

Si se tiene un sistema de ventilación adecuado en el invernadero el control de la humedad no es aspecto por el cual se deba tener preocupación, debido a que este tipo de sistemas previenen la condensación de la humedad en el invernadero y en los cultivos [14]. Sin embargo, si se tiene un ambiente seco la mejor solución es tener un sistema de riego automatizado.

Los sistemas de riego en invernaderos ayudan a enfocar el agua en la raíz de cada planta. Por lo que siempre resulta beneficiosos para el cultivo disponer de sistema automáticos de control de riego. Existen algunos métodos o técnicas de riego que se emplean en invernaderos, sin embargo, los más utilizados son el sistema de riego por aspersión, sistema de riego por goteo y sistema de riego por inundación [16]. En la Tabla 3.2 se muestra una comparativa entre estos sistemas de riego, se puede observar que el método de riego que brinda más beneficios es el sistema de riego por goteo. Por lo que se optó este tipo de riego para implementar en el prototipo.

**Tabla 3.2** Comparativa entre sistema de riego por goteo, sistema de riego por aspersión y sistema de riego por inundación [2]

Sistema de riego por goteo	Sistema de riego por aspersión	Sistema de riego por inundación
Mejor adaptación a terrenos de cualquier tipo de pendiente	Funciona correctamente únicamente en terrenos planos o semi planos	Funciona correctamente únicamente en terrenos planos o semi planos
Distribuye el agua de manera lenta por lo que las plantas la aprovechan de mejor manera	Dependiendo de la presión de aspersión, la distribución del agua es desigual para las plantas	Distribuye el agua de manera masiva por lo que reduce la transpiración de las plantas
Consumo de agua eficiente	Consumo de agua notable	Consume demasiada agua
Se puede controlar de manera adecuada el flujo de agua	Se puede controlar de manera adecuada el flujo de agua	Dificultad para controlar el flujo de agua

## **Sistema de envío de notificaciones**

El proyecto requiere del envío de una alerta a modo de mensaje de texto SMS, que se realice a través de la red celular a un número predeterminado con la finalidad de notificar cuando exista un aumento perjudicial de la temperatura y/o humedad relativa en el invernadero. Por lo cual, se debe tomar en consideración el modelo de módulo GSM que se va a incluir en el prototipo, además de los elementos y características que permitan realizar esta acción.

## **Sistema de comunicación entre el circuito y Alexa**

El prototipo requiere de la comunicación del microcontrolador con el asistente virtual Alexa. Para lo cual se requiere de un *software* intermediario que permita la comunicación entre ambos.

## **Estructura del proyecto**

Una vez terminado los requerimientos de hardware y *software* del circuito, el prototipo requiere de una maqueta en donde se incorporen los componentes del circuito finalizado con la finalidad de facilitar la visualización del funcionamiento del prototipo y poder verificar que se cumple con los objetivos del proyecto.

## **3.2 Selección del hardware y *software* que conformaran el sistema del prototipo.**

### **Selección de las placas de desarrollo para el sistema**

Se tomo en consideración placas de desarrollo basadas en *software* libre y cumplan con las peticiones del prototipo. Entre estas características necesarias para el desarrollo del proyecto se tiene entradas y salidas digitales para las conexiones entre los diferentes sensores, *shields* y componentes electrónicos. Además, que permita establecer una comunicación serial (Rx y Tx) para la conexión con el módulo GSM. Se investigaron a los microcontroladores Arduino y al minicomputador Raspberry Pi, sin embargo, como otro de los requerimientos importantes del prototipo es la conexión *Wi-Fi* y permita incluir librerías de *Blynk* para que sea posible la comunicación con la aplicación móvil, por ende, la comunicación con Alexa se optó por placas que contengan el chip ESP8266 o el chip ESP32.

En la Tabla 3.3 se muestra la comparativa entre estos dos chips que permiten la comunicación *Wi-Fi*, tomando en cuenta los requerimientos del sistema se optó por el chip

ESP8266 debido a que se adapta de mejor manera al prototipo principalmente por el costo y por la facilidad de uso. Por lo cual, se seleccionó a la placa de desarrollo NodeMCU con el chip ESP8266.

**Tabla 3.3** Comparativa entre el chip ESP8266 y ESP32 [17]

ESP8266	ESP32
Precio económico en el mercado	Precio mayor al del chip ESP32
Se puede programar con el IDE de Arduino	Se puede programar con el IDE de Arduino
Mayor compatibilidad con la mayoría de las librerías	Ciertas librerías podrían dar fallos al momento de realizar pruebas
Es posible encontrar mayores recursos de apoyo, por ejemplo, en foros	Dificultad para encontrar recursos de apoyo

### Selección del módulo GSM para el envío de notificaciones

Para el envío de notificaciones mediante mensajes de texto SMS se optó por el módulo SIM900, el cual es un módulo GSM/GPRS que incluye las características necesarias para los requerimientos del prototipo, principalmente debido a que este es de bajo costo y amplio soporte. En la Tabla 3.4 se muestra una comparativa entre dos *shields* GSM/GPRS, uno con el módulo SIM900 y otro con el módulo SIM800, en donde se muestra similares características, sin embargo, debido a que el *shield* SIM800 es más actual este es más caro actualmente en el mercado. Por lo tanto, debido al propósito del prototipo el *shield* que se adapta más a los requerimientos del prototipo y a un costo accesible es el *shield* SIM900 GSM/GPRS para cumplir con el objetivo de enviar mensajes de texto (SMS).

**Tabla 3.4** Comparativa entre el *Shield* SIM900 GSM/GPRS y *Shield* SIM800 GSM/GPRS [18]

<i>Shield</i> SIM900 GSM/GPRS	<i>Shield</i> SIM800 GSM/GPRS
Costo económico en el mercado	Más costoso en comparación al <i>Shield</i> SIM900 GSM/GPRS
Trabaja en las bandas GSM de 850, 900, 1800 y 1900 (MHz)	Trabaja en las bandas GSM de 850, 900, 1800 y 1900 (MHz)
Fuente de alimentación de 5 a 12 (V)	Fuente de alimentación de 5 a 12 (V)
Fácil desarrollo del código de programación	Dificultad al desarrollar el código de programación debido a que incorpora funcionalidad extra como <i>Bluetooth</i>

Al realizar las pruebas del *shield* SIM900 en conjunto con el módulo NodeMCU ESP 8266, se tuvieron problemas en la comunicación serial entre ambos dispositivos debido a que los puertos seriales (Rx y Tx) del módulo NodeMCU trabajan con 3.3 (V) y los puertos seriales

(Rx y Tx) del *shield* SIM900 trabajan a 5.5 (V) por lo que existían inconvenientes a la hora de comunicarse [17]. Sin embargo, se optó por incluir un microcontrolador adicional que sus puertos seriales (Rx y Tx) trabajen a 5 (V) sin la necesidad de añadir un circuito adicional, además de que permitirá añadir futuras funcionalidades ya que quedan entradas y salidas del microcontrolador adicionales libres.

### Selección del microcontrolador

Uno de los microcontroladores es Arduino, el cual es usado en la mayoría de proyectos similares debido a que cumple con todos los requerimientos para la comunicación. Sin embargo, la plataforma Raspberry Pi también se puede adaptar a estos requerimientos. Tomando en cuenta la comparativa que se muestra en la Tabla 3.5 entre estos dos dispositivos se optó por el microcontrolador Arduino debido a que la plataforma Raspberry Pi trabaja en sus entradas y salidas digitales a 3.3 (V) por lo que se tendría el mismo inconveniente que con el módulo NodeMCU ESP8266, además la programación del dispositivo se debería realizar en otro lenguaje de programación (*Python*) [19].

**Tabla 3.5** Comparativa entre Arduino Y Raspberry Pi [19]

Arduino	Raspberry Pi
Se utiliza el Arduino IDE para su programación	Es recomendable utilizar <i>Python</i> , sin embargo, se puede utilizar C y C++
Funciona a 5 voltios	Funciona a 3.3 voltios
Contiene 20 pines de entrada y salida digital	Contiene 24 pines de entrada y salida digital a 3.3 voltios
Precio económico	Precio mayor al de Arduino

### Selección del sensor de temperatura

Para medir la temperatura del prototipo se optó por incluir un sensor el cual mida en tiempo real dicha magnitud. Tomando esto en cuenta se investigó que los sensores DHT11 y su predecesor DHT22 cumplen con los requerimientos necesarios. Ambos son sensores digitales fáciles de manejar con cualquier microcontrolador. En la Tabla 3.6 se muestra una comparativa entre ambos sensores, tomando en cuentas las características de ambos se optó por manejar el sensor DHT22 debido a que este maneja mayor precisión en la lectura de la temperatura en tiempo real que el sensor DHT11.

**Tabla 3.6** Comparativa entre el sensor de temperatura DHT11 y DHT22 [20]

Sensor DHT11	Sensor DHT22
Precio económico en el mercado	Precio mayor en comparación con el sensor DHT11

Lee la temperatura en un rango de 0 a 50 (°C), con precisión en la lectura de 2 (°C)	Lee la temperatura en un rango de -40 a 125 (°C), con precisión en la lectura de 0.5 (°C)
Tiempo de respuesta de 1 (s)	Tiempo de respuesta de 2 (s)

### Selección del sensor de humedad de suelo

Para medir la humedad del suelo en el invernadero se optó por incluir un sensor el cual mida dicho factor climático en tiempo real y en cualquier parte del suelo del cultivo. Para la cual se investigaron varios sensores de humedad de suelo que cumplieran con este propósito. Durante la investigación se encontró dos sensores, que son el sensor de humedad de suelo FC-28 y el sensor de humedad de suelo capacitivo. En la Tabla 3.7 se muestra una tabla comparativa entre ambos sensores, tomando en cuenta las similitudes de ambos sensores se optó por el sensor de humedad de suelo FC-28 debido a que se encuentra en un precio económico, y posee un pin analógico y un pin digital en comparación con el sensor capacitivo [21].

**Tabla 3.7** Comparativa entre el sensor de humedad de suelo FC-28 y el sensor de humedad de suelo capacitivo [21]

Sensor de humedad de suelo FC-28	Sensor de humedad de suelo capacitivo
Precio económico en el mercado	Precio elevado en comparación al sensor FC-28
Opera a una corriente de 35 (mA)	Opera a una corriente de 5 (mA)
Voltaje de señal en la salida analógica de 0 a VCC (V)	Voltaje de señal en la salida analógica de 0 a 5 (V)
Voltaje de señal en la salida digital de 3.3/5 (V)	No posee salida digital
Con el electrodo sumergido el ciclo de vida útil del sensor es de 3 a 6 meses	El ciclo de vida útil del sensor es de 3 años mínimo

### Selección del actuador para el sistema de ventilación y sistema de riego por goteo

Para el sistema de ventilación se optó por utilizar un ventilador de PC ya que cumple con el objetivo del prototipo para el control de temperatura y puede ser accionado mediante un relé. El ventilador de PC cumple la función de extraer el aire caliente del interior, en este

caso del invernadero, y expulsarlo al exterior logrando disminuir la temperatura del invernadero.

Para el sistema de riego por goteo se optó por utilizar una electroválvula, que consiste en una válvula eléctrica que permite el flujo de agua mediante funciones de apertura o cierre total de la válvula. Por lo cual, cumple con los requisitos necesarios para ser conectado al sistema de riego por goteo.

### **Selección del *software* de programación para la *skill* de Alexa**

Para que el asistente virtual Alexa controle el invernadero mediante comandos determinados se tomó en cuenta el desarrollador de *skills* de Amazon denominado *Amazon Developer Console* y la plataforma *Voiceflow*, sin embargo, tomando en cuenta que la programación en *Voiceflow* se realiza mediante bloques, por lo cual sería más fácil desarrollar el programa, y tiene total compatibilidad con la plataforma *Blynk* se seleccionó trabajar con el *software Voiceflow*. Añadir que en *Voiceflow* se puede probar en tiempo real el programa que se va desarrollando por lo que se puede corregir fallas antes de terminar el programa, y una vez finalizado el programa únicamente se sube la *skill* a la cuenta de Amazon.

## **3.3 Diseño del sistema del prototipo**

### **Esquema general del proyecto**

El esquema general del proyecto plasma de manera general el funcionamiento del proyecto y la forma en la que se relacionan entre si los distintos elementos y sus funciones. El prototipo fue desarrollado con dos esquemas de control, en los cuales uno está encargado de controlar e interactuar con el sensor de temperatura y sensor de humedad del suelo, los dispositivos electrónicos y el asistente virtual y otro encargado de la comunicación con la red celular a través del módulo GSM.

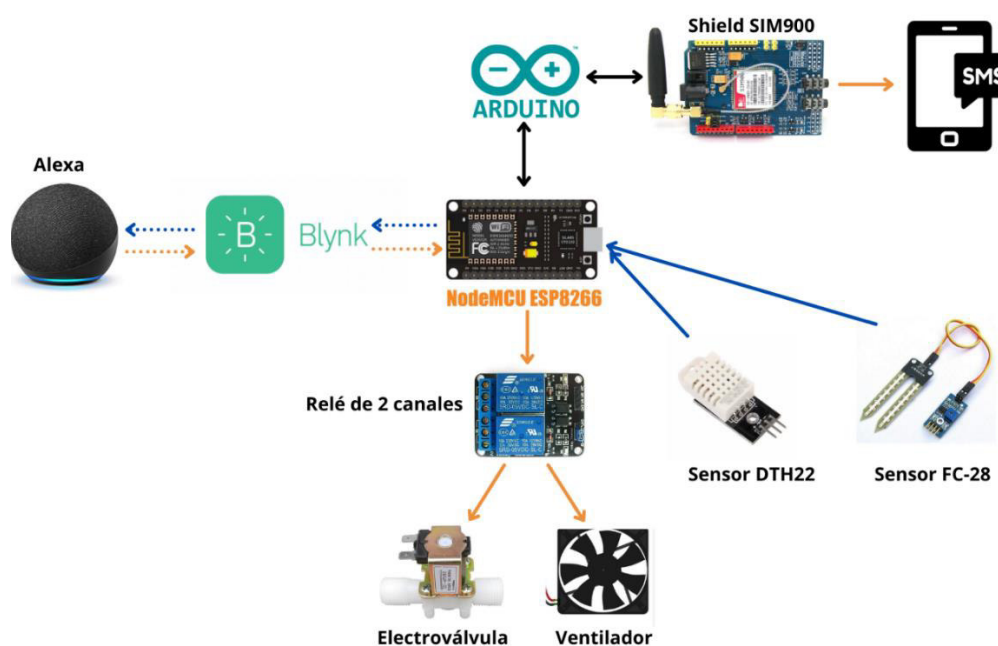
En la Figura 3.1 la línea azul representa la comunicación por parte del módulo NodeMCU ESP8266 para recepción de datos del sensor DHT22 y el sensor de humedad FC-28, en cambio las líneas naranjas representan la transmisión de datos del módulo NodeMCU hacia el módulo relé para la activación o desactivación de los actuadores. Por otro lado, las líneas naranjas y azules entrecortadas representan la comunicación de transmisión y recepción entre el módulo NodeMCU ESP8266, la plataforma *Blynk* y el asistente virtual Alexa. Como se puede observar en el esquema general de funcionamiento sin la plataforma *Blynk* no sería posible la comunicación entre el módulo *Wi-Fi* y Alexa. Finalmente, la

comunicación serial entre el módulo NodeMCU y Arduino, y entre Arduino y el *shield* GSM/GPRS SIM900 están representados por una flecha en color negro.

Tomando en cuenta que el prototipo trata sobre el control de un invernadero, el proceso empieza en cualquiera de las siguientes situaciones: cuando el usuario inicia la comunicación con Alexa, cuando el valor de la temperatura y/o humedad es mayor, en caso de la temperatura, y menor, en caso de la humedad, al valor permitido o cuando se realiza una acción en la aplicación móvil. En cualquiera de los casos mencionados, el módulo NodeMCU siempre está en comunicación con el sensor DHT22, el sensor de humedad FC-28 y la plataforma *Blynk* para que se lean los valores de la temperatura y humedad en tiempo real.

La activación o desactivación de los actuadores sucede de forma automática cuando el valor de la temperatura y/o humedad sea mayor o menor, según corresponda, del valor permitido. Sin embargo, también se lo puede realizar en cualquier momento mediante los comandos establecidos con el asistente virtual o por medio de la aplicación móvil.

Finalmente, el envío de la notificación mediante mensaje de texto SMS se da automáticamente cuando el valor de la temperatura y/o humedad es mayor o menor, según corresponda, al del valor permitido. En la Figura 3.1, se presenta el esquema del funcionamiento general, donde se muestra la forma de interacción de los distintos elementos.



**Figura 3.1** Esquema general del funcionamiento del prototipo



## Método de comunicación entre el sistema y Alexa

Para establecer la comunicación entre el módulo NodeMCU ESP8266 y Alexa se empleó la plataforma *Blynk*. En la Figura 3.2 se muestra como es la forma de interacción entre el módulo NodeMCU, el asistente virtual Alexa y la plataforma *Blynk*. La comunicación se logra a través de pines virtuales que se crean en la aplicación móvil de donde se obtienen los datos y posteriormente es obtenido por el asistente virtual mediante un enlace API [22].



Figura 3.2 Comunicación entre el módulo NodeMCU y Alexa

## Desarrollo de la aplicación móvil con la plataforma *Blynk*

Para crear la aplicación móvil se utilizó un dispositivo Android en donde se instaló la plataforma *Blynk* mediante la aplicación *Play Store*. Una vez obtenido la plataforma *Blynk* en la ventana de inicio se tiene la opción de crear un nuevo proyecto, seguido de eso se escribe el nombre del proyecto y se selecciona el dispositivo mediante al cual se va a conectar mediante *Wi-Fi* que es el módulo NodeMCU y se selecciona el tipo de conexión *Wi-Fi* tal y como se muestra en la Figura 3.3 . Finalmente se crea el proyecto.

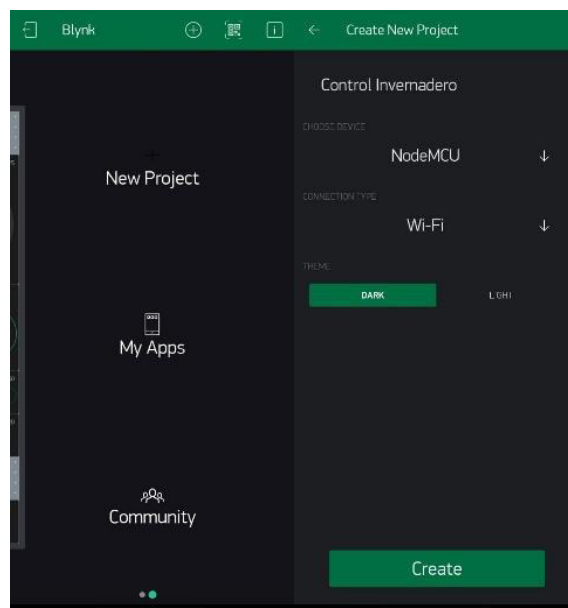
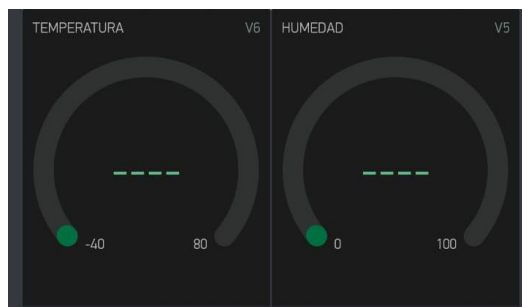


Figura 3.3 Creación del nuevo proyecto

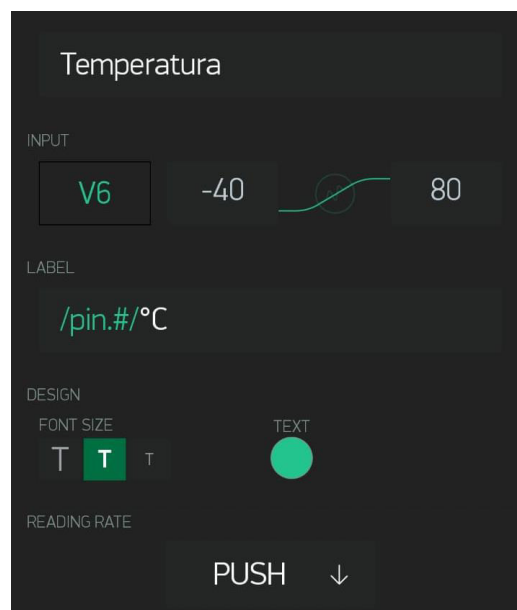
### Creación de los indicadores del valor de temperatura y humedad

Una vez creado el proyecto se procedió a insertar los *widgets* necesarios del *widget box*, de la plataforma *Blynk*, que ayudara al funcionamiento correcto de la aplicación móvil. Para observar en tiempo real la temperatura y humedad del invernadero se seleccionó el *widget gauge* y se colocó en el espacio de trabajo del proyecto. Se colocó un *gauge* para el valor de la humedad y otro *gauge* para el valor de la temperatura como se muestra en la Figura 3.4.



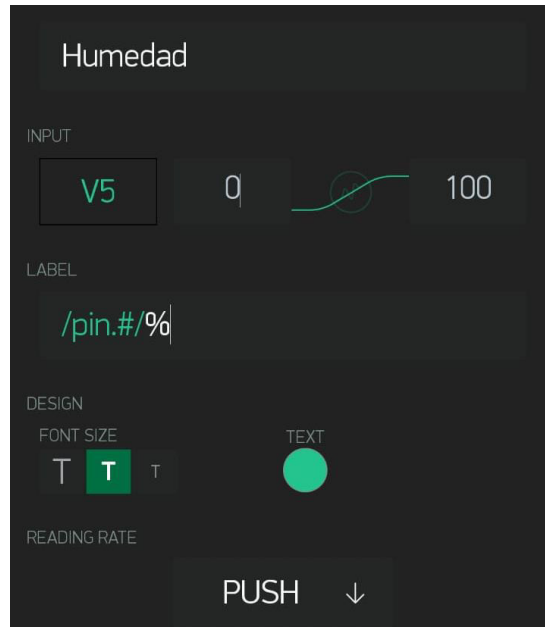
**Figura 3.4** *Gauges* colocados para observar el valor de temperatura y humedad

En la Figura 3.5 y Figura 3.6 se muestra respectivamente la configuración realizada en el indicador de temperatura y en el indicador de humedad. En el caso del indicador de temperatura se colocó el pin virtual 6, un rango de valores de -40 a 80 debido a que en ese rango mide la temperatura el sensor DHT22, se configuró la etiqueta para que se muestre el valor de temperatura con un decimal seguido de la unidad de medición de temperatura (°C). Finalmente, se coloca la tasa de lectura en modo *push*.



**Figura 3.5** Configuración del indicador de temperatura

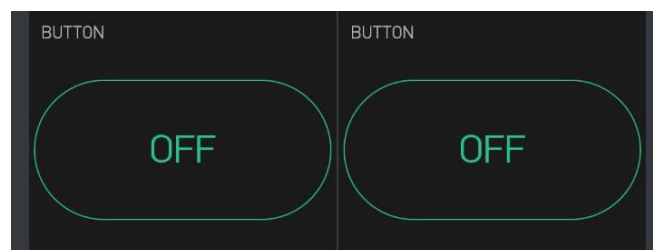
La configuración es similar para el caso del indicador de humedad en el cual se colocó el pin virtual 5, un rango de valores de 0 a 100 debido a que en ese rango mide la humedad el sensor FC-28, se configuró la etiqueta para que se muestre el valor de humedad con un decimal seguido de la unidad de medición de humedad (%). Finalmente, se coloca la tasa de lectura en modo *push*.



**Figura 3.6** Configuración del indicador de humedad

### Creación de los botones de encendido o apagado del sistema de ventilación y sistema de riego

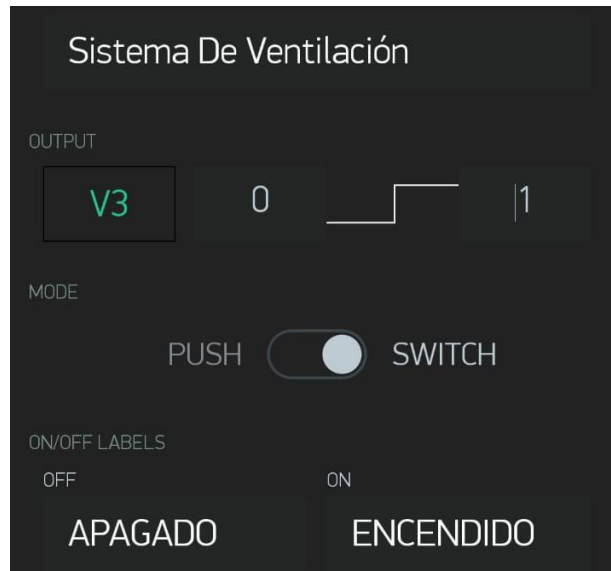
Para poder activar o desactivar tanto el sistema de ventilación como el sistema de riego por goteo se seleccionó el *widget button* como se muestra en la Figura 3.7.



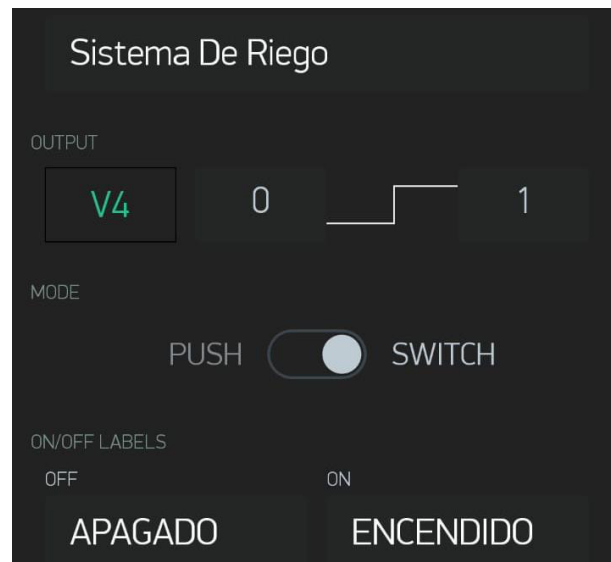
**Figura 3.7** Botones colocados para activar o desactivar el sistema de ventilación y de riego

Una vez ingresado los botones se procedió a realizar las configuraciones mostradas en la Figura 3.8 y Figura 3.9 respectivamente para activar o desactivar el sistema de ventilación y sistema de riego. Para el sistema de ventilación se seleccionó el pin virtual 3 y para el

sistema de riego se seleccionó el pin virtual 4. Finalmente, en ambos casos se configuro el botón en tipo *switch* con el fin de que la activación o desactivación de cada sistema sea de forma inmediata. De igual forma se colocaron las etiquetas como encendido o apagado para que sea más fácil la comprensión y manejo de los botones en la aplicación móvil.



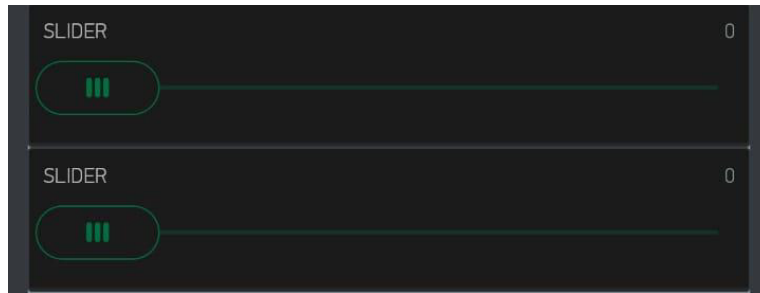
**Figura 3.8** Configuración del botón de encendido o apagado del sistema de ventilación



**Figura 3.9** Configuración del botón de encendido o apagado del sistema de riego

### **Configuración de los *sliders* para determinar el valor de temperatura y humedad crítica del invernadero**

Para poder establecer el valor de temperatura y humedad crítico en el invernadero mediante la aplicación se seleccionó el *widget slider* como se muestra en la Figura 3.10.

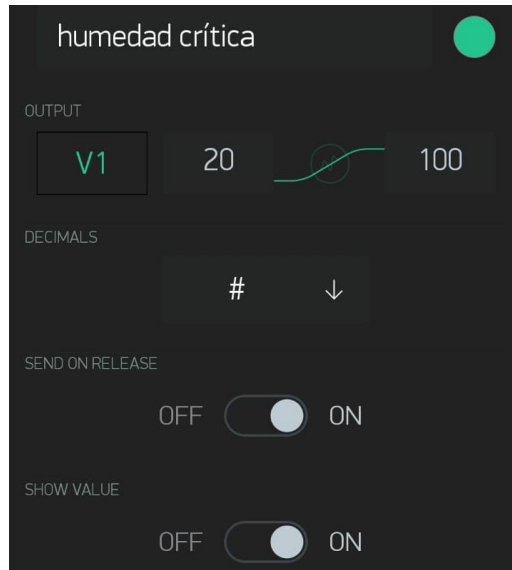


**Figura 3.10** Sliders colocados para determinar el valor critico de temperatura y humedad en el invernadero

Una vez ingresado los *sliders* se procedió a realizar las configuraciones mostradas en la Figura 3.11 y Figura 3.12 respectivamente para establecer el valor critico de temperatura y humedad en el invernadero. Para el valor critico de temperatura se seleccionó el pin virtual 2 con un rango de valores de 0 a 60 y para el valor critico de humedad se seleccionó el pin virtual 1 con un rango de valores de 20 a 100. Finalmente, en ambos casos se configuro que el valor a establecer sea entero sin decimales y las demás opciones se las deajo por defecto.



**Figura 3.11** Configuración del *slider* para establecer el valor de temperatura critica

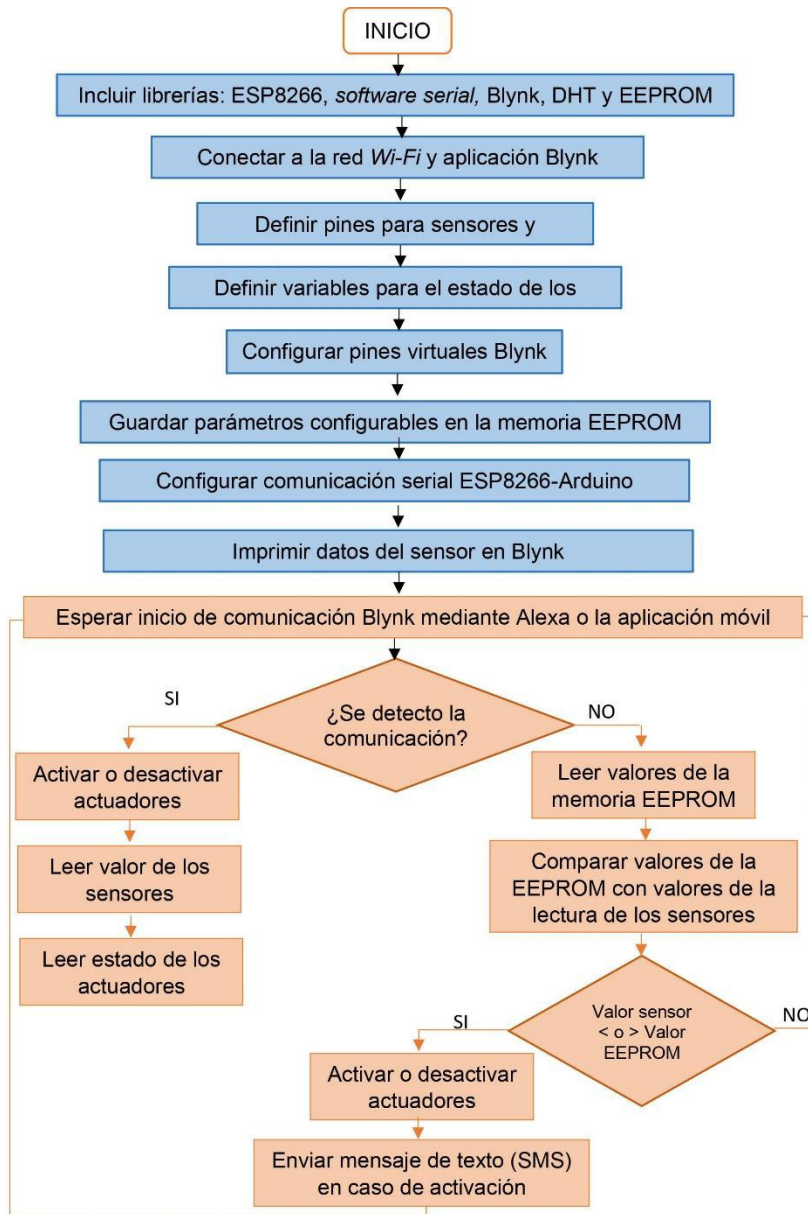


**Figura 3.12** Configuración del *slider* para establecer el valor de humedad critica

### **Esquema general del código de programación**

Con el propósito de tener una idea general clara del proceso de desarrollo del código de programación se realizó un diagrama de flujo general. En la Figura 3.13 se muestra el esquema general que conlleva el código de programación. En el diagrama de flujo se detalla de manera general como se programó el prototipo, se muestra la inclusión de librerías, la conexión a la red *Wi-Fi*, la inicialización de cada elemento electrónico o actuador, la configuración de pines virtuales y físicos, la configuración de entradas y salidas digitales y analógica, la configuración para el control automatizado y finalmente la configuración para la activación o desactivación de los actuadores.

Como se muestra en el diagrama de flujo de la Figura 3.13 el principal proceso del prototipo es la aplicación *Blynk* en comunicación con el módulo NodeMCU, ya que si se inicia los comandos de voz con el asistente virtual la plataforma *Blynk* es la encargada de realizar las acciones que se dicten, y de igual forma si ocurre un evento perjudicial en la humedad y/o temperatura se activa o desactiva los actuadores y esto debe ser plasmado en la aplicación móvil creada en *Blynk*.



**Figura 3.13** Diagrama de flujo general

### **Programación por bloques en la plataforma *Voiceflow* de la *skill* para el asistente virtual**

En la Figura 3.14 se muestra el programa desarrollado mediante bloques en la plataforma *Voiceflow*. Para acceder a la plataforma únicamente se necesita crear una cuenta mediante un correo electrónico y la plataforma permite crear un proyecto gratuito. Se utilizaron distintos bloques detallados a continuación:

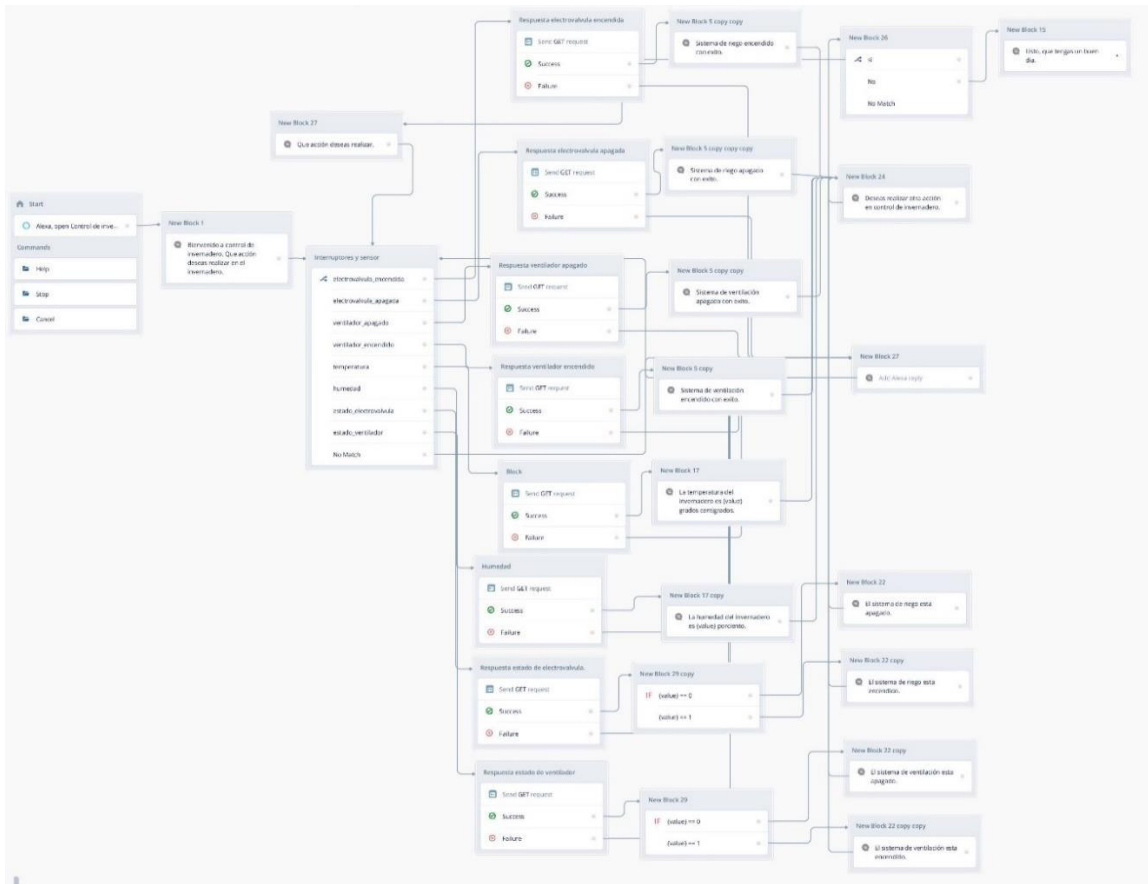


Figura 3.14 Programa desarrollado en Voiceflow

### Bloque start

En este bloque se configura el comando de voz mediante el cual se va a iniciar la *skill* a desarrollarse. En la Figura 3.15 se muestra el bloque configurado, para iniciar la *skill* desarrollada para el prototipo se utilizó el comando de voz “Alexa, abrir Control de invernadero”.

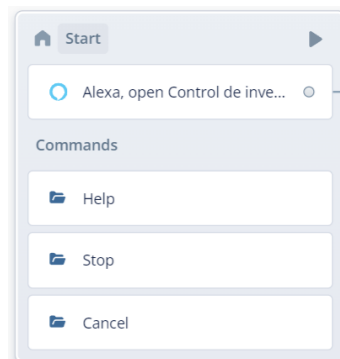
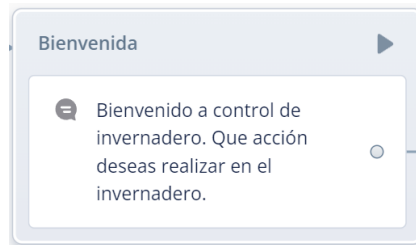


Figura 3.15 Bloque start



### Bloque *speak*

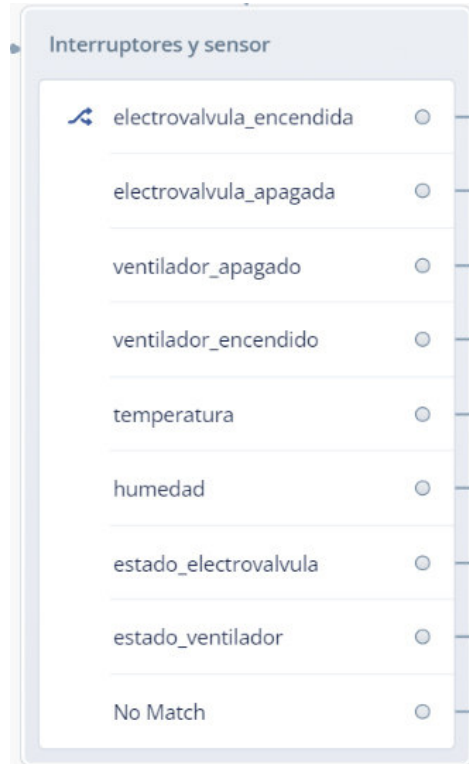
En este bloque se configura las líneas de texto que va a decir el asistente virtual. Usando este bloque se da la bienvenida a la *skill* para el control del invernadero. En la Figura 3.16 se muestra las líneas de texto configuradas.



**Figura 3.16** Bloque *speak*

### Bloque *choice*

Este bloque permite introducir varios comandos de voz distintos, cada comando de voz permite realizar una acción específica. Se usó este bloque para introducir las opciones que Alexa puede realizar en el invernadero. En la Figura 3.17 se muestra las configuraciones realizadas en el bloque.



**Figura 3.17** Bloque *choice*

Los comandos de voz especificados para las distintas acciones son:

- Encender sistema riego: este comando de voz enciende el sistema de riego.
- Apagar sistema de riego: este comando de voz apaga el sistema de riego.
- Encender sistema de ventilación: este comando de voz enciende el sistema de ventilación.
- Apagar sistema de ventilación: este comando de voz apaga el sistema de ventilación.
- Dime la temperatura del invernadero: este comando de voz devuelve el valor de la temperatura del invernadero en tiempo real.
- Dime la humedad del invernadero: este comando de voz devuelve el valor de la humedad del invernadero en tiempo real.
- Dime el estado del sistema de riego: este comando de voz permite conocer si el sistema de riego esta activado o desactivado en ese momento.
- Dime el estado del sistema de ventilación: este comando de voz permite conocer si el sistema de ventilación esta activado o desactivado en ese momento.

## Bloque API

Este bloque permite la comunicación e interacción entre la aplicación móvil creada en la plataforma Blynk y la *skill* desarrollada para el asistente virtual. Se uso este bloque para realizar las funciones mencionadas en el anterior bloque *choice*. Para lo cual, en este bloque es necesario utilizar el servidor gratuito en la nube que nos proporciona Blynk que se obtuvo realizando una petición de ping hacia la página web blynk-cloud.com y da como resultado la dirección IP de Norteamérica como se muestra en Figura 3.18.

```
C:\Users\user>ping blynk-cloud.com

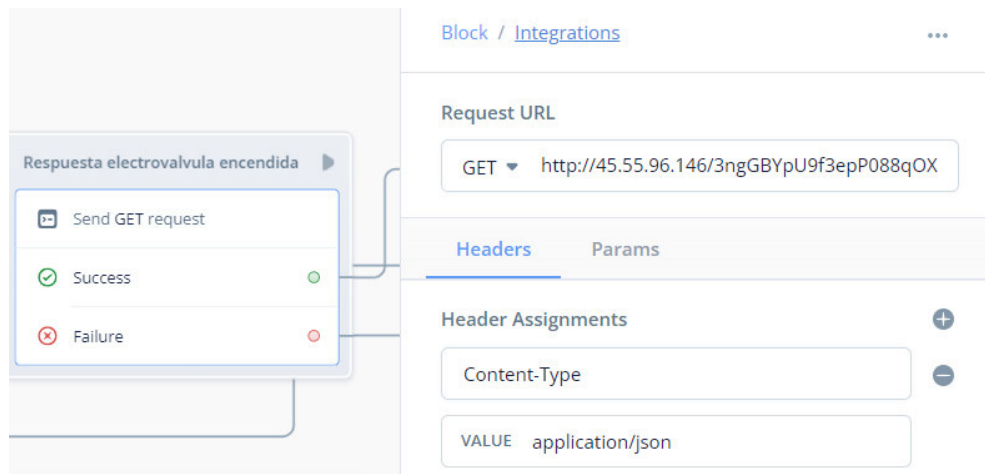
Haciendo ping a blynk-cloud.com [45.55.96.146] con 32 bytes de datos:
Respuesta desde 45.55.96.146: bytes=32 tiempo=91ms TTL=44
Respuesta desde 45.55.96.146: bytes=32 tiempo=98ms TTL=44
Respuesta desde 45.55.96.146: bytes=32 tiempo=91ms TTL=44
Respuesta desde 45.55.96.146: bytes=32 tiempo=91ms TTL=44

Estadísticas de ping para 45.55.96.146:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 91ms, Máximo = 98ms, Media = 92ms
```

**Figura 3.18** Obtención de la dirección IP del servidor *cloud* de Blynk

La dirección IP correspondiente al servidor *cloud* de Blynk es utiliza debido para configurar el bloque API es necesario de dicho servidor en donde se aloja las distintas configuraciones

realizadas en la aplicación móvil. En el bloque API se utiliza la dirección “http://45.55.96.146/3ngGBYpU9f3epP088qOXUI8nYqkmqo\_c/parámetros”, el primer parámetro corresponde a la dirección IP ser servidor *cloud* de Blynk seguido del token de acceso que permite la comunicación API con la aplicación móvil y, por último, los parámetros que realizaran la acción en la aplicación móvil. En la Figura 3.19 se muestra el bloque API y un ejemplo de configuración.



**Figura 3.19** Bloque API

Por ejemplo, los parámetros configurados para encender el sistema de riego son: `http://45.55.96.146/3ngGBYpU9f3epP088qOXUI8nYqkmqo_c/update/V4?value=1`, en donde:

- *update*, especifica que se va realizar la acción en la aplicación móvil,
- *V4*, especifica el pin virtual correspondiente al actuador del sistema de riego
- *value=1*, especifica que se va a colocar un 1 lógico en el pin virtual para que se encienda el sistema de riego.

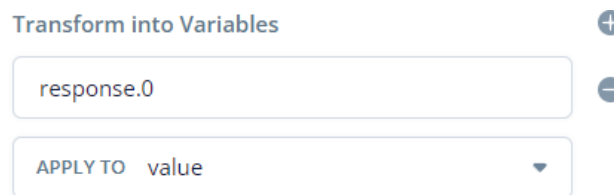
La misma configuración es utilizada para encender el sistema de ventilación, cambiando el pin virtual a *V3* que corresponde al ventilador. De igual forma en ambos casos para apagar el sistema se utiliza los mismos parámetros con la particularidad que en el parámetro *value* se coloca un 0 lógico para que se apague el sistema.

Para obtener el valor de temperatura o humedad, además del estado del sistema de ventilación o sistema de riego se utiliza la dirección `http://45.55.96.146/3ngGBYpU9f3epP088qOXUI8nYqkmqo_c/get/V6`, en donde los parámetros configurados son:

- Get, especifica que va obtener un valor de la aplicación móvil.
- V6, especifica el pin virtual del cual va obtener el valor, V6 para obtener el valor de temperatura, V5 para obtener el valor de humedad, V3 para obtener el estado del sistema de ventilación y V4 para obtener el estado del sistema de riego.

En el caso de encendido o apagado del sistema de ventilación o el sistema de riego seguido del bloque API se colocó un bloque *speak* para mencionar que se activó o desactivo el sistema correspondiente exitosamente.

En el caso en donde se obtiene el valor de temperatura o humedad, en el bloque API se realiza una configuración extra como se muestra en la Figura 3.20 que cumple la función de que el valor obtenido del pin virtual sea guardado en la variable *value*. Seguido del bloque API se ingresó un bloque *speak* que cumple la función de decir el valor de temperatura o humedad usando la variable *value* respectivamente. Por ejemplo, para que el asistente virtual mencione la temperatura del invernadero se muestra el mensaje respectivo en la Figura 3.21.



**Figura 3.20** Configuración para guardar el valor obtenido en la variable *value*

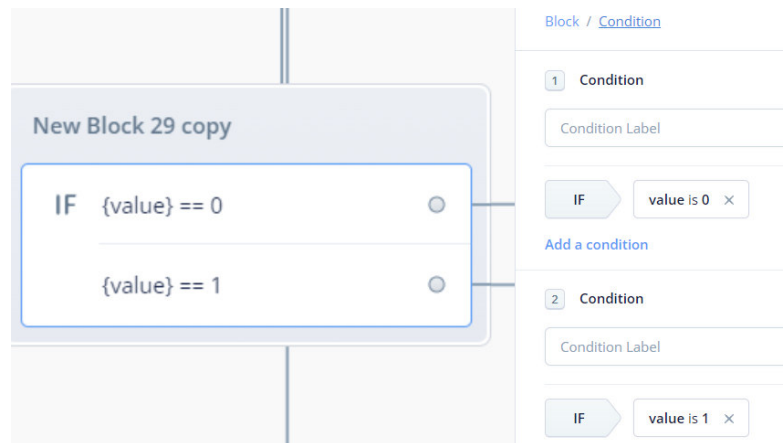


**Figura 3.21** Bloque *speak* utilizado para mencionar el valor de temperatura del invernadero

### Bloque de condición

Finalmente, en el caso donde se obtiene el estado sea del sistema de ventilación o del sistema de riego seguido del bloque API se utiliza un bloque de condición, el cual permite realizar una acción únicamente si se cumple una regla configurada en el bloque como se muestra en la Figura 3.22. Para esto también en el bloque API se guarda el valor obtenido

del estado del sistema correspondiente en la variable *value* y mediante el bloque de condición se genera una regla que si el valor de *value* es igual a 0 el sistema se encuentra apagado, y si el valor de *value* es igual a 1 el sistema se encuentra encendido. El asistente virtual mencionará el estado del sistema correspondiente gracias a que se utiliza un bloque *speak* en donde se especifica el mensaje correspondiente, como se muestra en la Figura 3.23.



**Figura 3.22** Bloque de condición



**Figura 3.23** Configuración de los bloques para que Alexa mencione el estado del sistema de riego

Para finalizar con la programación por bloques de la *skill* para Alexa, una vez finalizado con cualquiera de las acciones especificadas en el bloque *choice*, se utilizó un bloque *speak* seguido de un bloque de condición para que el asistente virtual mencione si se desea realizar otra acción en el invernadero y mediante el bloque de condición se colocó una regla en el caso de que el usuario diga “si” se regrese al bloque *choice* para que se realice otra acción y otra regla en el caso de que el usuario diga “no” se finalice la *skill* con un mensaje mediante un bloque *speak*.

## Programación del módulo NodeMCU ESP8266 en Arduino IDE

Analizando de manera detallada el diagrama de flujo general se contempló en cuenta varios asuntos importantes para el desarrollo del código y su correcto funcionamiento que se detallan a continuación:

### Inclusión de librerías en el IDE de Arduino

Tomando en cuenta los elementos electrónicos seleccionados para el funcionamiento del prototipo, se incluyeron las librerías que se muestran en la Figura 3.24.

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
```

**Figura 3.24** Librerías

Se incluyeron un total de 5 librerías que cumplen el propósito de asegurar el correcto funcionamiento de cada elemento electrónico. A continuación, se detalla la función que cumple cada librería:

- Librería ESP8266

Permite manejar el módulo NodeMCU ESP8266. Se usa para realizar la conexión a la red *Wi-Fi* y subir el código desarrollado en el IDE de Arduino al módulo NodeMCU ESP8266 específico.

- Librería *BlynkSimpleEsp8266*

Esta librería contiene todas las funciones o herramientas necesarias para poder programar en el IDE de Arduino. Se usa para lograr la interacción de la aplicación *Blynk* con el módulo NodeMCU.

- Librería DHT

Permite trabajar con la familia de sensores DHT. Se usa para programación del sensor DHT22.

- Librería EEPROM

Permite trabajar con la memoria no volátil del módulo NodeMCU. Se usa para escribir y leer datos que se configura según la sintaxis de programación.

- Librería *SoftwareSerial*

Esta librería proporciona la capacidad de la comunicación serial del módulo NodeMCU. Se usa para la interacción entre el módulo NodeMCU y Arduino, además de la comunicación entre Arduino y el *shield* SIM900 GPS/GPRS.

### **Creación de variables para los sensores y parámetros configurables de temperatura y humedad**

Para el correcto funcionamiento del prototipo fue necesario utilizar variables de diferentes naturalezas, específicamente se utilizó dos tipos de variables las cuales son: variable de tipo *int* y variable de tipo *char*.

En la Figura 3.25 se muestra las variables de tipo *int* que fueron utilizadas para el ingreso de la temperatura y humedad crítica, además de la lectura de la humedad en tiempo real.

```
int hi;  
int ti;  
int h=0;
```

**Figura 3.25** Variables para parámetros de automatización

Por otro lado, en la Figura 3.26 se muestra las variables de tipo *char* que se utilizaron para el envío de mensajes de texto (SMS). Estas variables cumplen la función de que, si existe un valor de temperatura y humedad perjudicial, se envíe la letra correspondiente del módulo NodeMCU al Arduino, y posteriormente el Arduino sepa en qué momento realizar el envío del mensaje de texto (SMS).

```
char letra='#';  
char letral='?';
```

**Figura 3.26** Variables para el envío de mensajes de texto (SMS)

El valor de la temperatura y humedad crítica puede ser configurada mediante la aplicación móvil desarrollada en *Blynk*, para lo cual mediante la función *BLYNK\_WRITE* se define el pin virtual correspondiente al parámetro configurable y se lee el valor que se está generando en la aplicación móvil como se muestra en la Figura 3.27. Posteriormente se guarda dicho valor en la memoria EEPROM para poder utilizarla en otra función.

```

BLYNK_WRITE(V1)
{
  int hi = param.asInt();
  EEPROM.put(0, hi);
}
BLYNK_WRITE(V2)
{
  int ti = param.asInt();
  EEPROM.put(4, ti);
}

```

**Figura 3.27** Configuración de pines virtuales para configurar el valor de la temperatura y humedad critica

Los actuadores y sensor de temperatura se definen según el pin digital del módulo NodeMCU al cual estén conectados. Para lo cual se utiliza la función *define* se coloca el nombre respectivo del actuador o sensor seguido del pin al cual está conectado. Finalmente, para que el sensor de temperatura funcione correctamente se especifica el sensor DHT que se usa, que en mi caso es el sensor DHT22, como se muestra en la Figura 3.28.

```

#define DHTPIN D2
#define switchventilador D5
#define switchelectrovalvula D7
#define DHTTYPE DHT22

```

**Figura 3.28** Declaración de pines del módulo NodeMCU y definición del tipo de sensor DHT

Mediante la aplicación se puede activar o desactivar los actuadores, para lo cual mediante la función *BLYNK\_WRITE* se define el pin virtual al que corresponde el actuador, se lee el valor que se genera en la aplicación (1 o 0 lógico) y finalmente se escribe dicho valor mediante la función *digitalWrite* en el actuador correspondiente como se muestra en la Figura 3.29.



```

BLYNK_WRITE(V3)
{
  int pinValue = param.asInt();
  digitalWrite(switchventilador, pinValue);
  delay(200);
}

BLYNK_WRITE(V4)
{
  int pinValue = param.asInt();
  digitalWrite(switchelectrovalvula, pinValue);
  delay(200);
}

```

**Figura 3.29** Configuración de pines virtuales para activación o desactivación de los actuadores

### Configuración de la función *void setup*

En esta función se inicializa los diferentes elementos y módulos que compone el programa, además se define los actuadores como salida, como se muestra en la Figura 3.30.

Como primer punto, se inicializa la comunicación serial del módulo NodeMCU con Arduino, seguido de la memoria EEPROM y el sensor DHT22. En este punto también se inicia la comunicación con la plataforma *Blynk* para que se identifique los parámetros de autenticación y se inicie la comunicación con la aplicación móvil. Finalmente se definen los actuadores y se inicializan en bajo para que estes apagados al iniciar el prototipo.

```

Serial.begin(115200);
ARD.begin(115200);
delay(20000);
EEPROM.begin(512);
Blynk.begin(auth, ssid, pass);
dht.begin();

pinMode(switchventilador, OUTPUT);
pinMode(switchelectrovalvula, OUTPUT);
digitalWrite(switchventilador, LOW);
digitalWrite(switchelectrovalvula, LOW);

```

**Figura 3.30** Configuración del *Setup*

### Configuración de la función *sensores*

Como primer punto se lee los valores que fueron guardados en la memoria EEPROM de la temperatura y humedad critica configuradas en la aplicación móvil, para la cual se utiliza la configuración que se muestra en la Figura 3.31.

```
int hi = EEPROM.get(0, hi);
int ti = EEPROM.get(4, ti);
```

**Figura 3.31** Lectura de los valores guardados en la EEPROM

En la Figura 3.32 se muestra la configuración para el funcionamiento correcto del sensor de temperatura, así como de los parámetros para el control automatizado y el envío del mensaje (SMS). Primero se lee la lectura de temperatura del sensor y se imprime en el pin virtual correspondiente a la aplicación móvil, mediante el comando *Blynk.virtualWrite* con el propósito que se muestre la temperatura en tiempo real. Posteriormente se compara el valor tomado de la EEPROM correspondiente a la temperatura crítica con la lectura de temperatura del sensor y se los compara, en caso de que el valor sea menor o igual automáticamente se escribe en el actuador correspondiente y pin virtual correspondiente un bajo que significa desactivado, y caso contrario se escribe un alto que significa activado.

Finalmente, si el valor de la lectura de temperatura del sensor es mayor al valor crítico se envía una alerta en forma de notificación, con el mensaje “Temperatura del invernadero perjudicial”, mediante el comando *Blynk.notify* y se imprime en la simulación serial con el Arduino la variable letra correspondiente al símbolo “#”.

```
sensors_event_t event;
dht.temperature().getEvent(&event);
if (isnan(event.temperature)) {
  Serial.println("Failed to read temperature!");
}
else {
  Serial.print(F("Temperature: "));
  Serial.println(event.temperature);
  Serial.println(F("°C"));
  Blynk.virtualWrite(V6, event.temperature);

  if (event.temperature <= ti) {
    digitalWrite(switchventilador, LOW);
    Blynk.virtualWrite(V3, LOW);
  }
  else{
    if (event.temperature > ti){
      digitalWrite(switchventilador, HIGH);
      Blynk.virtualWrite(V3, HIGH);
    }
  }
}

if (event.temperature > ti){
  Blynk.notify("Temperatura del invernadero perjudicial");
  ARD.print(letra);
}
```

**Figura 3.32** Configuración del funcionamiento del sensor de temperatura

En la Figura 3.33 se muestra la configuración para el funcionamiento correcto del sensor de humedad similar al funcionamiento del sensor de temperatura, así como de los parámetros para el control automatizado y el envío del mensaje (SMS). Primero se lee la lectura de humedad del sensor FC-28, se mapea ese valor siendo el valor 0 igual a 100 % de humedad y 1024 igual a 0% de humedad utilizando el comando *map* y se imprime en el pin virtual correspondiente a la aplicación móvil, mediante el comando *Blynk.virtualWrite* con el propósito que se muestre la humedad en tiempo real.

Posteriormente se compara el valor tomado de la EEPROM correspondiente a la humedad crítica con la lectura de humedad del sensor y se los compara, en caso de que el valor sea menor o igual automáticamente se escribe en el actuador correspondiente y pin virtual correspondiente un alto que significa activado, y caso contrario se escribe un alto que significa desactivado.

Finalmente, si el valor de la lectura de humedad del sensor es menor al valor crítico se envía una alerta en forma de notificación, con el mensaje “Humedad del invernadero perjudicial”, mediante el comando *Blynk.notify* y se imprime en la simulación serial con el Arduino la variable letra1 correspondiente al símbolo “?”.

```
int humedad = map(analogRead(humedad), 0, 1023, 100, 0);
Serial.print("Humedad: ");
Blynk.virtualWrite(V5, humedad);
Serial.print(humedad);
Serial.println("%");
delay(100);
if (humedad <= hi) {
  digitalWrite(switchelectrovalvula,LOW);
  Blynk.virtualWrite(V4,LOW);
}
else{
  if (humedad > hi){
    digitalWrite(switchelectrovalvula,HIGH);
    Blynk.virtualWrite(V4,HIGH);
  }
}
if (humedad < hi){
  Blynk.notify("Humedad del invernadero perjudicial");
  ARD.print(letra1);
}
```

**Figura 3.33** Configuración del funcionamiento del sensor de humedad

## **Programación de Arduino para el funcionamiento del *shield* SIM900 GSM/GPRS**

Para el funcionamiento correcto de prototipo la placa Arduino cumple la función de comunicarse con el *shield* SIM900 GSM/GPRS. Como se detalla en el anterior punto el módulo NodeMCU se comunica con el Arduino cuando el valor de temperatura y/o humedad es perjudicial en el invernadero, enviando un símbolo mediante comunicación serial entre ambos. Tomando esto en cuenta, como primer punto se tiene definir los pines seriales para la comunicación con el *shield* utilizando el comando *softwareserial* como se muestra en la Figura 3.34, también se incluye la librería *softwareserial*.

```
SoftwareSerial SIM900(7, 8)
```

**Figura 3.34** Definición de pines seriales del Arduino

### **Declaración de variables**

Se definió una única variable de tipo *char* denominada “recibir” como se muestra en la Figura 3.35, esta variable cumplirá la función de leer en la comunicación serial si existe un valor o no.

```
char recibir;
```

**Figura 3.35** Declaración de variables del Arduino

### **Configuración de la función *void setup***

En esta función se inicia la comunicación serial para la comunicación con el módulo NodeMCU y con el *shield* SIM900 como se muestra en la Figura 3.36.

```
Serial.begin(115200);  
delay(500);  
SIM900.begin(115200);  
delay(500);
```

**Figura 3.36** Configuración de la función *void setup*

### **Configuración de la función *void loop***

Esta función fue utilizada para la lectura de la comunicación serial, en esta parte se utiliza la variable “recibir” la cual es igual a la lectura en ese instante del puerto serial. Una vez obtenido la lectura del puerto serial se inicia el condicional *if* para comparar el valor obtenido con el símbolo “#” y el símbolo “?”. En caso de que sea igual al primer símbolo se llama a la función para el envío del mensaje de temperatura perjudicial y si es igual al segundo símbolo se llama a la función para el envío del mensaje de humedad perjudicial. El código

de programación se muestra en la Figura 3.37. Finalmente, se utiliza el comando *delay* para que se envíe el mensaje respectivo cada 10 (s).

```
if(Serial.available()){
  recibir=(char)Serial.read();
  if(recibir == '#'){
    envioMensaje();
  }else{
    if(recibir == '?'){
      envioMensaje1();
    }
  }
}
delay(10000);
```

**Figura 3.37** Configuración de la función *void loop*

### **Configuración de la función para el envío de mensaje en caso de temperatura y/o humedad perjudicial del invernadero**

La configuración para el envío del mensaje se realiza mediante una serie de impresiones de comandos AT en la comunicación serial para configurar el *shield* SIM900. Como primer punto se imprime el comando AT “AT+CMGS=1/r” el cual cumple la función de configurar el *shield* en modo texto, en el segundo punto se tiene el comando AT “AT+CMGS=“0999592971” en el cual se especifica el número de celular al cual se va enviar el mensaje, seguidamente se escribe e imprime la línea de texto que se va a enviar. Finalmente se indica el final del mensaje mediante el código ASCII 26. En la Figura 3.38 y Figura 3.39 se muestra el código de programación para el envío del mensaje de temperatura perjudicial y/o humedad perjudicial en el invernadero respectivamente.

```
SIM900.print("AT+CMGF=1\r");
delay(200);
SIM900.println("AT + CMGS = \"0999592971\"");
delay(200);
SIM900.println("Temperatura perjudicial en el invernadero");
delay(200);
SIM900.println((char)26);
delay(200);
SIM900.println();
```

**Figura 3.38** Configuración de la función para el envío de mensaje en caso de temperatura perjudicial del invernadero

```

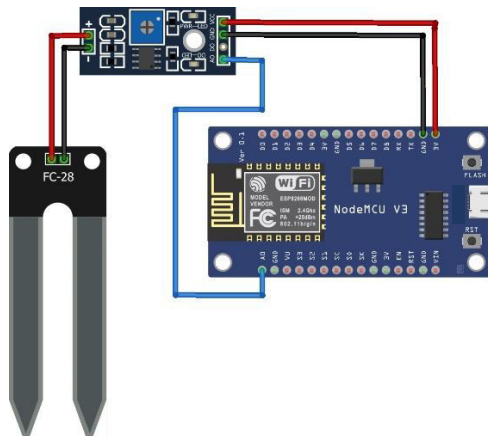
SIM900.print("AT+CMGF=1\r");
delay(200);
SIM900.println("AT + CMGS = \"0999592971\"");
delay(200);
SIM900.println("Humedad perjudicial en el invernadero");
delay(200);
SIM900.println((char)26);
delay(200);
SIM900.println();

```

**Figura 3.39** Configuración de la función para el envío de mensaje en caso de humedad perjudicial del invernadero

### Conexión del módulo NodeMCU ESP8266 con el sensor de humedad de suelo FC-28

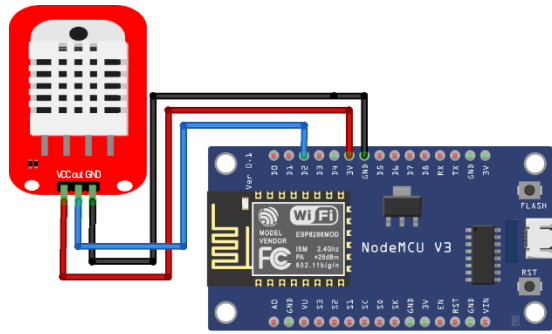
El sensor de humedad de suelo FC-28 contiene 2 pines en el lado izquierdo de la placa que fueron conectados a los electrodos con la polaridad respectiva. En el lado derecho de la placa consta de 4 pines, el pin de alimentación fue conectado a un pin de 3.3 (V) del módulo NodeMCU, el pin GND fue conectado a un pin GND del módulo NodeMCU y el pin analógico fue conectado al pin analógico A0 del módulo NodeMCU. El pin digital no se conectó debido a que ya se está realizando la medición de la humedad mediante el pin analógico. Las conexiones realizadas se muestran en la Figura 3.40.



**Figura 3.40** Conexión del módulo NodeMCU con el sensor FC-28

### Conexión del módulo NodeMCU ESP8266 con el sensor DHT22

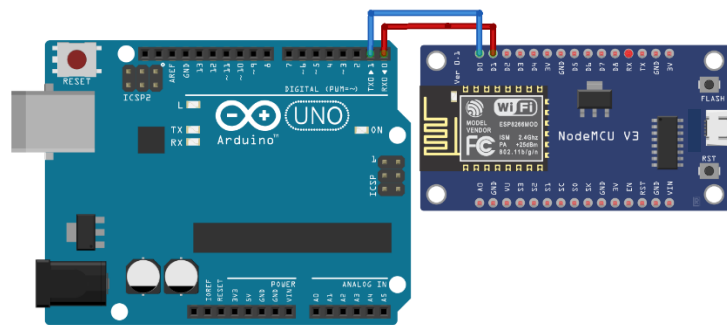
El sensor digital DHT22 tiene 3 pines, el pin de alimentación fue conectado a un pin de 3.3 (V) del módulo NodeMCU, el pin GND fue conectado a un pin GND del módulo NodeMCU y por último el pin de datos digital se conectó al pin digital D2 del módulo NodeMCU. Las conexiones realizadas se muestran en la Figura 3.41.



**Figura 3.41** Conexión del módulo NodeMCU con el sensor DHT22

### Conexión del módulo NodeMCU ESP8266 a Arduino

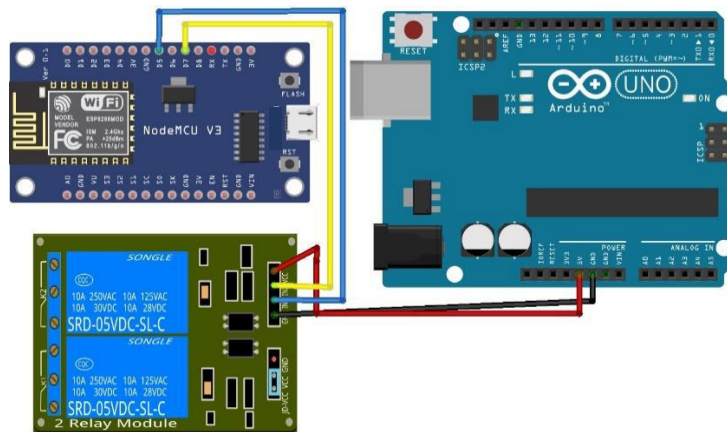
El módulo NodeMCU con Arduino se comunicó con un enlace serial, para lo cual se conectó el pin digital D0 y D1, para simular la comunicación serial (Rx y Tx respectivamente) de módulo NodeMCU, a los pines seriales (Rx y Tx) del Arduino asegurándose que se conecte transmisión con recepción y recepción con transmisión, como se muestra en la Figura 3.42.



**Figura 3.42** Conexión del módulo NodeMCU con Arduino

### Conexión del módulo NodeMCU ESP8266 con el módulo relé 2 canales

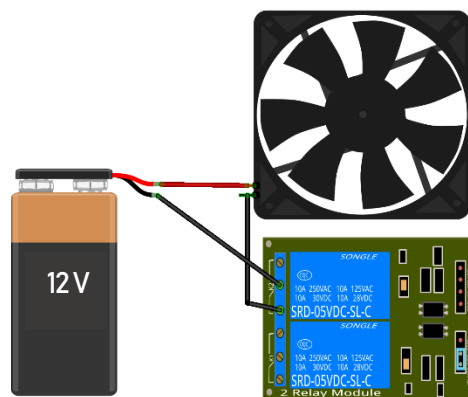
El módulo relé 5 voltios de 2 canales contiene cuatro pines, el pin de alimentación fue conectado a un pin de 5 (V) del Arduino, el pin GND fue conectado a un pin GND de Arduino, el pin de entrada del primer canal del relé fue conectado al pin digital D5 del módulo NodeMCU que fue designado para el sistema de ventilación, y finalmente el pin de entrada del segundo canal del relé fue conectado al pin digital D7 del módulo NodeMCU que fue designado para el sistema de riego. Las conexiones realizadas se muestran en la Figura 3.43.



**Figura 3.43** Conexión del módulo NodeMCU con el módulo relé 2 canales

### **Conexión del ventilador de PC y electroválvula con el módulo relé 2 canales**

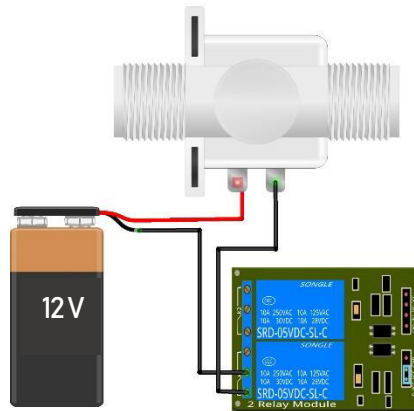
En lado de las borneras del módulo relé, se conectó en las borneras del relé 1 el ventilador de PC, el cual es parte del sistema de ventilación, que contiene 2 pines, el pin de 12 (V) se conectó directamente al positivo de la fuente de alimentación y el pin GND se conectó a la bornera denominada normalmente cerrada. Por otro lado, el negativo de la fuente de alimentación fue conectada en la bornera denominada común, como se muestra en la Figura 3.44.



**Figura 3.44** Conexión del ventilador al relé 2 canales

En las borneras del relé 2 se conectó la electroválvula, el cual maneja el sistema de riego por goteo, que contiene 2 pines, el pin de 12 (V) se conectó directamente al positivo de la fuente de alimentación y el pin GND se conectó a la bornera denominada normalmente cerrada. Por otro lado, el negativo de la fuente de alimentación fue conectada en la bornera denominada común como se muestra en la Figura 3.45.

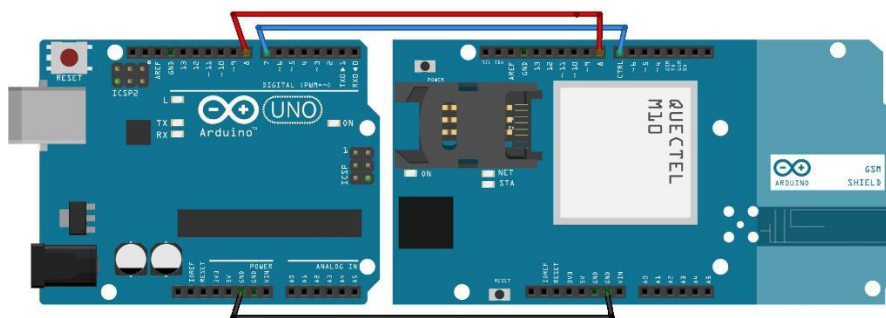




**Figura 3.45** Conexión de la electroválvula con el relé 2 canales

### **Conexión de Arduino con el *shield* SIM 900 GSM/GPRS**

Arduino con el *shield* SIM900 se comunican mediante un enlace serial, para lo cual se conectó el pin digital 7 y 8, para simular la comunicación serial (Rx y Tx respectivamente), del Arduino a los pines seriales (Rx y Tx) del *shield* SIM900 asegurándose que se conecte transmisión con recepción y recepción con transmisión. Además de conectar un pin GND del Arduino a un pin GND del *shield*, como se muestra en la Figura 3.46.



**Figura 3.46** Conexión de Arduino con el *shield* SIM900.

## **3.4 Implementación del prototipo**

### **Construcción de la maqueta de invernadero para el prototipo**

Para lograr verificar el funcionamiento del circuito y el código de programación desarrollado se realizó una maqueta de un invernadero. Se tomó como base del invernadero una tabla de madera de 35 x 55 (cm). Para las paredes del invernadero en sí, se tomaron postes hechos de madera de 2 (cm) de grosor y 40 (cm) de altura. En la Figura 3.47 se muestra una foto de los materiales utilizados para la construcción de la maqueta.



**Figura 3.47** Materiales usados para la construcción de las paredes y base de la maqueta.

Con todos los materiales adquiridos, se procedió a juntar todos los postes para formar las paredes del invernadero y colocarlas en la base del mismo. En la Figura 3.48 se muestra como las paredes colocadas en la base del invernadero. Se colocó las paredes a 5 (cm) de cada lado de la base para que termine de manera uniforme la maqueta.



**Figura 3.48** Colocación de las paredes en la base del invernadero

Para el techo del invernadero se utilizó los mismos postes utilizados para las paredes del invernadero, sin embargo, se cortaron a la mitad para formar un techo triangular como se muestra en la Figura 3.49.



**Figura 3.49** Techo triangular para la maqueta de invernadero.

### **Instalación del sistema de riego en el invernadero**

Una vez terminado con la estructura del invernadero, se colocaron plantas artificiales en tierra para poder evidenciar el funcionamiento correcto del sistema de riego por goteo. En la Figura 3.50 se muestra la forma en que fueron colocadas las plantas por surcos para poder visualizar de mejor manera el funcionamiento del prototipo. Una vez colocado las plantas se instaló el sistema de riego por goteo que básicamente consiste en colocar postes, que sostenga la manguera, cada 5 (cm) y colocar los goteros distanciados por la misma dimensión como se observa en la Figura 3.50. Finalmente, la manguera va conectado a la electroválvula.



**Figura 3.50** Sistema de riego por goteo implementado

### **Instalación del sistema de ventilación en el invernadero**

Como primer punto se colocaron canaletas en cada poste de las paredes del invernadero para que se pueda ordenar el cableado dentro de las mismas y no exista deterioro de los cables por factores del exterior. Como se muestra en la Figura 3.51 el ventilador de PC se lo coloco en la parte frontal del invernadero con el fin de evidenciar el funcionamiento correcto del mismo. El ventilador fue colocado encima de las canaletas para poder pasar los cables de este por las canaletas. Finalmente, se conecta los cables al relé correspondiente.



**Figura 3.51** Sistema de ventilación implementado

### **Instalación del sensor de humedad FC-28**

El sensor de humedad FC-28 fue colocado de tal forma que pueda ser movilizado sus electrodos con el fin de medir la humedad en tiempo real en cualquier parte del suelo cuando se requiera hacerlo. Además, los cables de conexión del sensor de humedad hacia los electrodos fueron cubiertos con cinta *taipe* de color negro para protegerlo de factores externos. En la Figura 3.52 se muestra el sensor implementado.



**Figura 3.52** Sensor de humedad FC-28 implementado

### Instalación del sensor de temperatura DHT22

El sensor DHT22 se instaló en el techo del invernadero con el fin de que realice la lectura de temperatura del invernadero correctamente, es decir, que pueda medir en tiempo real la temperatura interna del invernadero, además que los cables de conexión del sensor están protegidos por las canaletas colocadas en los postes de las paredes como se muestra en la Figura 3.53.



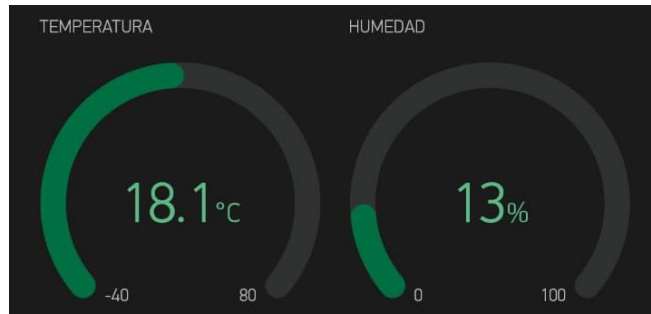
**Figura 3.53** Instalación del sensor de temperatura DHT22la

### 3.5 Pruebas de funcionamiento del prototipo

Con el propósito de comprobar el funcionamiento del prototipo completado se llevó a cabo las siguientes pruebas:

#### **Lectura en tiempo real del valor de temperatura y humedad en el invernadero mediante la aplicación y Alexa**

Al conectar el prototipo el módulo NodeMCU se conecta automáticamente a la red *Wi-Fi* e inmediatamente se muestra en los indicadores de la aplicación creada la temperatura y humedad en tiempo real del invernadero como se muestra en la Figura 3.54.



**Figura 3.54** Lectura en tiempo real de la temperatura y humedad en el invernadero

De igual forma el asistente virtual Alexa dio los siguientes valores según el comando de voz:

- Dime la temperatura del invernadero.

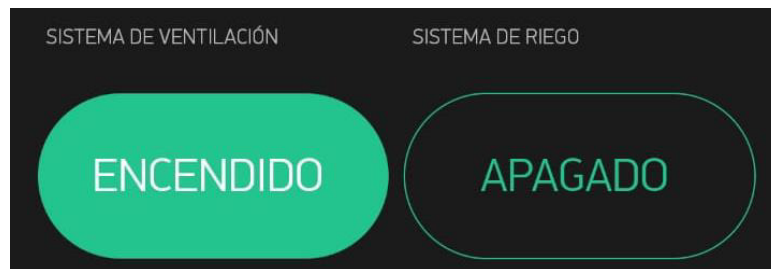
En este caso Alexa responde: La temperatura del invernadero es 18.1 (°C).

- Dime la humedad del invernadero.

En este caso Alexa responde: La humedad del invernadero es 13 (%).

### **Encendido del sistema de ventilación mediante la aplicación y Alexa**

En la Figura 3.55 se muestra que el sistema de ventilación este encendido debido a que el botón que controla dicho sistema se encuentra activado (encendido).



**Figura 3.55** Encendido del sistema de ventilación mediante la aplicación

Una vez encendido el sistema de ventilación mediante la aplicación se observa que automáticamente se enciende el sistema de ventilación como se muestra en la Figura 3.56.



**Figura 3.56** Sistema de ventilación encendido

De igual forma se comprobó que los comandos de voz para activar el sistema de ventilación mediante Alexa funcionan correctamente. La prueba realizada dio resultados positivos, se dijo a Alexa:

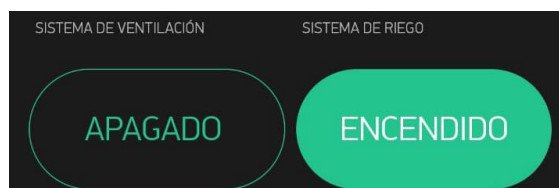
- Encender sistema de ventilación.

En este caso Alexa responde: Sistema de ventilación encendido con éxito.

Se realizó las mismas pruebas para desactivar el sistema de ventilación mediante la aplicación y Alexa dando los resultados correctos por lo que se comprobó que las funciones de la aplicación y Alexa para controlar el sistema de ventilación funcionan acorde a lo esperado.

### **Encendido del sistema de riego por goteo mediante la aplicación y Alexa**

En la Figura 3.57 se muestra que el sistema de ventilación este encendido debido a que el botón que controla dicho sistema se encuentra activado (encendido).



**Figura 3.57** Encendido del sistema de riego mediante la aplicación

De igual forma se comprobó que los comandos de voz para activar el sistema de riego mediante Alexa funcionan correctamente. La prueba realizada dio resultados positivos, se dijo a Alexa:

- Encender sistema de riego.

En este caso Alexa responde: Sistema de riego encendido con éxito.

Se realizó las mismas pruebas para desactivar el sistema de riego mediante la aplicación y Alexa dando los resultados correctos por lo que se comprobó que las funciones de la aplicación y Alexa para controlar el sistema de riego funcionan acorde a lo esperado.

### **Envío de mensaje de texto SMS al número predeterminado y comprobación del funcionamiento de los *sliders***

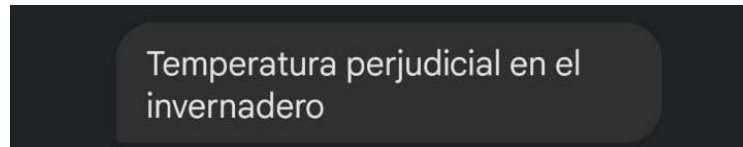
Una vez verificado el funcionamiento de la aplicación y los comandos de voz para Alexa, se procedió a comprobar que el mensaje sea enviado cuando ocurre un evento perjudicial en el invernadero. Además, que se coloque de forma correcta el valor de temperatura o humedad crítica en función del *slider* correspondiente.

Para comprobar el envío del SMS se colocó en el *slider* de la temperatura crítica un valor inferior al que se mostraba en ese momento en el indicador de temperatura, como se muestra en la Figura 3.58. En la Figura 3.59 se muestra la llegada automática del mensaje de texto al usuario predeterminado en donde se especifica que la temperatura del invernadero es perjudicial debido a que el valor del indicador de temperatura era mayor al de la temperatura crítica permitida.



**Figura 3.58** Funcionamiento correcto del *slider* de temperatura crítica



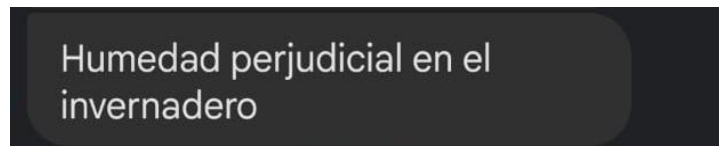


**Figura 3.59** SMS al usuario cuando existe un evento perjudicial de la temperatura en el invernadero

Para comprobar el envío del SMS se colocó en el *slider* de la humedad crítica un valor superior al que se mostraba en ese momento en el indicador de humedad como se muestra en la Figura 3.60. En la Figura 3.61 se muestra la llegada automática del mensaje de texto al usuario predeterminado en donde se especifica que la humedad del invernadero es perjudicial debido a que el valor del indicador de humedad era menor al de la humedad crítica permitida.



**Figura 3.60** Funcionamiento correcto del *slider* de humedad crítica



**Figura 3.61** SMS al usuario cuando existe un evento perjudicial de la humedad en el invernadero

### **Demostración del funcionamiento del prototipo**

En el ANEXO III se adjunta los códigos QR que dirigen al enlace de los videos demostrativos del funcionamiento y de mantenimiento del prototipo. En el video de funcionamiento se muestra una pequeña introducción de los elementos utilizados para la

implementación del prototipo, además, se explica el funcionamiento de la aplicación móvil creada en la plataforma Blynk, el funcionamiento automático tanto del sistema de ventilación como del sistema de riego en función del valor de temperatura y/o humedad crítica colocado en el *slider* respectivo y finalmente el funcionamiento del asistente virtual Alexa para controlar el invernadero.

En el video de mantenimiento, se muestran las acciones respectivas que se debe realizar a ciertos componentes del prototipo durante su uso o cada cierto tiempo para que funcione correctamente el prototipo en general.

## 4 CONCLUSIONES

- Por medio del desarrollo del presente proyecto, se comprobó que la implementación de sistemas de control en invernaderos, tiene una importancia crítica. En este caso se agregaron funcionales extra a la arquitectura tradicional de los sistemas de control, como el desarrollo de una aplicación móvil y el uso de un asistente virtual (Alexa) para permitir que el sistema sea controlado por medio de un teléfono móvil con el sistema operativo Android, o simplemente por medio de comandos de voz. Como principal ventaja del prototipo implementado se destaca que este permite reducir la posibilidad de perder el cultivo en el invernadero debido a factores climáticos como la humedad y la temperatura, razón por la cual es viable la implementación del proyecto, debido a que, se tiene como objetivo mejorar la calidad y servicios de control en invernaderos.
- Durante la elección del hardware y software para la implementación del prototipo, tomando en cuenta los requerimientos del mismo. La mejor selección fue el módulo NodeMCU ESP8266 gracias a su facilidad de programación en el IDE de Arduino, además de su costo, compatibilidad y capacidad de incluir sensores y módulos externos. Principalmente fue seleccionado ya que permite establecer una conexión *Wi-Fi* siendo el principal requerimiento para la comunicación con el asistente virtual.
- Las funcionalidades del presente proyecto se basan en automatizar el control de temperatura y humedad por parte de los agricultores en un invernadero. Esto se consiguió por medio del uso de sensores para detectar factores climáticos como la temperatura y humedad, además de una electroválvula para la apertura o cierre del flujo de agua en función de la humedad medida, un ventilador cuya activación se hará acorde a la temperatura medida, así como el empleo de la plataforma *Blynk* para establecer la comunicación con el asistente virtual (Alexa), gracias a la *skill* desarrollada en *Voiceflow*. Adicionalmente, se incluyeron funcionalidades extra como el envío de una notificación mediante SMS a un número predeterminado, así como la creación de una aplicación para

teléfonos inteligentes basados en Android, en la cual por medio del uso de *sliders* se permite configurar los parámetros críticos de humedad y temperatura, y realizar el control de los actuadores.

- Debido a que el *shield* SIM900 GSM/GPRS trabaja a 5 (V), y el módulo NodeMCU con 3.3 (V) fue necesario emplear un módulo Arduino que permita establecer la comunicación entre estos dos dispositivos y así agregar la funcionalidad de envío de notificaciones críticas del sistema por medio de mensajes de texto. El Arduino es el que se encarga de la comunicación con el *shield* GSM/GPRS, debido a que trabaja con los mismos niveles de voltaje que el *shield*, además se lo empleo debido a su facilidad de uso, compatibilidad y costo. Como resultado con la inclusión del Arduino no se necesitó de ningún componente electrónico extra para adecuar la comunicación acorde a los niveles de voltaje requeridos.
- Se concluye que el desarrollo del código de programación para que el módulo NodeMCU trabaje en conjunto con la aplicación *Blynk* no es complejo gracias a las librerías de *Blynk* disponibles, debido a esto, únicamente se debe incluirla y usar las diferentes instrucciones por medio de comandos fáciles de leer y comprender como: *Blynk.virtualWrite*, *Blynk.notify*, además existen funciones como *Blynk\_Write* que permiten realizar el control de los actuadores. Por lo tanto, la programación de los dispositivos microprocesados (o tarjetas de desarrollo) empleados en el presente proyecto como Arduino y NodeMCU no represento un alto nivel de dificultad.
- El proceso de programación de la *skill* para los comandos de voz del asistente virtual fue simple, debido a que la forma de programación mediante bloques en la plataforma *Voiceflow* es fácil de comprender y desarrollar, puesto que, durante el desarrollo del programa la plataforma permitía simular la *skill* y esto permitió ir depurando el código para corregir los errores en tiempo real.
- La agricultura bajo invernadero tiene el propósito de controlar de forma adecuada el clima dentro del mismo, por lo tanto, el aspecto importante es la forma en la que se controla estos factores. Gracias a las investigaciones realizadas durante el desarrollo de este proyecto se concluyó que el sistema de riego por goteo es una de las técnicas más aplicadas actualmente en la agricultura debido a su bajo costo de implementación, bajo consumo de agua y eficiencia para controlar la humedad en los cultivos, además que su instalación es sencilla y se puede controlar mediante una electroválvula como se realizó en el proyecto. Por otro lado, para controlar la temperatura en el invernadero la implementación del ventilador fue la mejor elección principalmente por su costo y

cumplimiento de los requerimientos del prototipo en comparación con otros sistemas para controlar la temperatura como un sistema de calefacción. La implementación de estos dos sistemas de control fue debido principalmente al uso eficiente de los recursos de los elementos electrónicos que se tenía.

## 5 RECOMENDACIONES

- Durante la conexión de las fuentes de alimentación para los distintos elementos electrónicos, como son el Arduino, el módulo NodeMCU y el *shield* GSM/GPRS se tuvieron problemas con el *shield* debido a que al intentar alimentarlo con el Arduino se provocaban reinicios no deseados del *shield*, esto debido a que no se proporcionaba la corriente necesaria para el funcionamiento correcto del dispositivo, por esta razón se recomienda alimentar el *shield* GSM/GPRS con una fuente externa.
- Se recomienda conectar el módulo NodeMCU ESP8266 al *router* o *Access Point Wi-Fi* principal del usuario, y que este trabaje en la banda 2.4 (GHz), esto debido a que durante las pruebas de funcionamiento se presentaron fallos cuando se conectó el módulo NodeMCU a un repetidor *Wi-Fi*, lo que fue principalmente provocadas por un nivel de señal bajo. Este mismo inconveniente se presentó con el asistente virtual (Alexa), ya que cuando se lo conectó al repetidor *Wi-Fi* no se iniciaba la *skill* al mencionar el comando de voz de inicio.
- Se recomienda, para futuras actualizaciones, cambiar el sensor de humedad FC-28 por un sensor de humedad capacitivo debido a que el sensor FC-28 tiene un tiempo corto de vida útil y esto podría ocasionar fallos en la lectura de la humedad.
- Colocar el asistente virtual Alexa y el módulo NodeMCU lo más cercano posible al router inalámbrico con la finalidad de que no exista retardos en la transmisión y el nivel de la potencia recibida sea mejor para tener una conexión estable y con una buena velocidad.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] R. Herrero, *INVERNADERO INTELIGENTE*, Ciudad de México , 2014.
- [2] N. Salazar, *DISEÑO DE UN SISTEMA INTELIGENTE PARA EL CONTROL DE INVERNADEROS*, Cuernavaca , 2020.
- [3] P. Castro, *COMPORTAMIENTO ENERGETICO DE INVERNADEROS AGRÍCOALAS EN EL ESTADO DE CHIHUAHUA, MEDIANTE SIMULACIÓN DINÁMICA EN TRNSYS*, Chihuahua, Chich: cimav, posgrado, 2011.
- [4] M. EElectronics, «ARDUINO.cl,» [En línea]. Available: <https://arduino.cl/arduino-uno/>. [Último acceso: 2 Enero 2022].
- [5] X. Mexico, *Qué es Arduino, cómo funciona y qué puedes hacer con uno*, 2020.
- [6] Sandorobotics, «SDR,» 2019. [En línea]. Available: <https://sandorobotics.com/producto/hr0126/>. [Último acceso: 01 Enero 2022].
- [7] «Naylamp Mechatronics,» 2021. [En línea]. Available: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>. [Último acceso: 01 Enero 2022].
- [8] N. Mechatronics, *SENSOR DE HUMEDAD DE SUELO FC-28*, Trujillo, Perú, 2021.
- [9] H. Miño, *DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CERRADURA CON REGISTRO DE ACCESO, SUPERVISIÓN REMOTA Y VIDEOVIGILANCIA*, Quito, 2021.
- [10] S. Pérez, «TechCrunch+,» 23 Abril 2019. [En línea]. Available: <https://techcrunch.com/2019/04/23/voiceflow-which-allows-anyone-to-make-voice-apps-without-coding-raises-3-5-million/>. [Último acceso: 5 Enero 2022].
- [11] N. Mechatronics, *SHIELD SIM900 GSM/GPRS*, Trujillo, Perú, 2021.
- [12] N. Mechatronics, *MÓDULO RELAY 2CH 5VDC*, Trujillo, Perú, 2021.
- [13] N. A. S.A, *Clima de un Invernadero. ¿Cómo conseguir la Temperatura Ideal?*, Madrid, 2016.

- [14] I. Invernaderos, «IMA Invernaderos,» [En línea]. Available: <https://www.invernaderosima.com/es/sistemas-de-calefact%C3%B3n.html>. [Último acceso: 26 Diciembre 2021].
- [15] «Revista Mundo HVAC&R,» Grupo Editorial Puntual Media, 2018. [En línea]. Available: <https://www.mundohvacr.com.mx/2014/05/sistemas-de-climatizacion-en-invernaderos/>. [Último acceso: 27 Diciembre 2021].
- [16] «Sembralia,» Cefetra Digital Services, 12 Agosto 2020. [En línea]. Available: <https://sembralia.com/sistema-de-riego-en-invernaderos/>. [Último acceso: 27 Diciembre 2021].
- [17] J. Fonseca, *DISEÑO E IMPLEMENTACIÓN DE CONTROL DOMÓTICO (CON SISTEMAS EMBEBIDOS) PARA CONECTARSE CON APLICACIONES ADAPTIVAS BASADOS EN IOT*, Quito, 2020.
- [18] R. Oña, *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO REMOTO VÍAGSM PARA EL HOTEL SAN MIGUEL*, Quito, 2015.
- [19] E. herrera, *Estudio, diseño e implementación de un prototipo de Entrenador de módulo Raspberry Pi*, 2014.
- [20] C. Monta, *“SISTEMA DE CONTROL DIFUSO Y MONITOREO PARA GRANJAS AVICOLAS UTILIZANDO SUELO REDIANDO CON ENERGÍA SOLAR*, 2020, Ambato.
- [21] «Naylamp Mechatronics,» 2021. [En línea]. Available: <https://naylampmechatronics.com/37-sensores-temperatura-y-humedad>. [Último acceso: 5 Enero 2022].
- [22] Techiesms, «Techiesms,» 2020. [En línea]. Available: <https://www.youtube.com/watch?v=YMDR4tQQdhE>. [Último acceso: 29 Diciembre 2021].

## 7 ANEXOS

## **ANEXO I**

### **Reporte de similitud de Turnitin**



## ANEXO II

**Código QR del enlace al código fuente del prototipo**



### ANEXO III

**Código QR con el link del video de funcionamiento del prototipo**



**Código QR con el link del video de mantenimiento del prototipo**

