UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:

INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:

INGENIEROS ELECTRÓNICOS

TEMA:

DESARROLLO DE UN DISPOSITIVO IOT PARA EL INVERNADERO DE LA ASOCIACIÓN DE MUJERES PRODUCTORAS AGROECOLÓGICAS DE CANGAHUA

AUTORES:

WILMER ERNESTO NECPAS LECHON

JAIME FERNANDO QUISHPE TUTILLO

TUTOR:

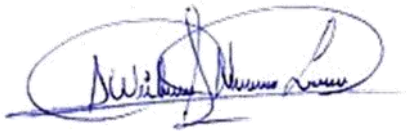
LUIS GERMÁN OÑATE CADENA

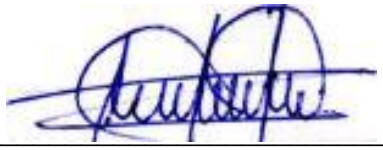
Quito, agosto del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Wilmer Ernesto Necpas Lechon y Jaime Fernando Quishpe Tutillo con documento de identificación N.º 100480919−8 y N.º 175030280−2 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **DESARROLLO DE UN DISPOSITIVO IOT PARA EL INVERNADERO DE LA ASOCIACIÓN DE MUJERES PRODUCTORAS AGROECOLÓGICAS DE CANGAHUA**, mismo que ha sido desarrollado para optar por el título de Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.





Wilmer Ernesto Necpas Lechon

C.I. 100480919−8

Jaime Fernando Quishpe Tutillo

C.I. 175030280−2

Quito, agosto del 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DESARROLLO DE UN DISPOSITIVO IOT PARA EL INVERNADERO DE LA ASOCIACIÓN DE MUJERES PRODUCTORAS AGROECOLÓGICAS DE CANGAHUA, realizado por Wilmer Ernesto Necpas Lechon y Jaime Fernando Quishpe Tutillo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, agosto del 2021

_____

Luis Germán Oñate Cadena

C.I. 1712157401

DEDICATORIA

Este trabajo de titulación lo dedico de todo corazón a mi madre Maria Esthela Lechon por su gran sacrificio y esfuerzo, por darme la oportunidad de culminar mis estudios y por confiar en mis capacidades. A mi hermana Diana Ximena Necpas y su esposo Juan Diego Albacura quienes me motivaron para no decaer en los momentos más difíciles y poder cumplir mi meta profesional por lo cual les considero un pilar muy importante en toda mi vida.

Wilmer Ernesto Necpas Lechon

Dedico este trabajo de titulación a mis padres Isabel Tutillo y Enrique Quishpe ya que su apoyo y sacrificio ha sido fundamental para la consecución de este logro, a mis hermanos Fabian y Segundo y mi hermana Mélida quienes siempre me apoyaron de sobremanera para no desistir en las adversidades y a Sandy que con sus palabras de amor y aliento me motiva día a día a continuar en el camino para conseguir las metas propuestas.

Jaime Fernando Quishpe Tutillo

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de vivir todas estas experiencias y ser mi fortaleza en todo momento.

Muy agradecido con mi madre Maria Esthela Lechon quien se ha esforzado en todo momento para darme la oportunidad de realizar mis estudios, muy agradecido por sus consejos y lecciones que me formaron como persona.

Al ingeniero Luis Germán Oñate quien confió en nosotros y nos acogió en su tutoría, le agradezco por brindarnos su apoyo constante y compartir todo su valioso conocimiento.

Wilmer Ernesto Necpas Lechon

Agradezco a mis padres por el gran sacrificio que han hecho a lo largo de mi vida, por brindarme sus consejos y por darme la oportunidad de vivir una de las experiencias más hermosas como lo es la vida universitaria.

Gracias a los docentes de la Universidad Politécnica Salesiana quienes impartieron su conocimiento a lo largo de mi carrera universitaria. A mis amigos Alberto, Esteban, Fernando, Henry y Wilmer con quienes vivimos experiencias inolvidables y se volvieron mi familia.

Al Ing. Luis Germán Oñate un agradecimiento especial por compartir sus conocimientos y apoyarnos en la realización de este trabajo de titulación.

Jaime Fernando Quishpe Tutillo

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR i
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR ii
DEDICATORIA iii
AGRADECIMIENTO iv
ÍNDICE GENERAL v
ÍNDICE DE FIGURAS vii
ÍNDICE DE TABLAS ix
RESUMEN x
ABSTRACT xi
INTRODUCCIÓN xii
CAPÍTULO 1 1
ANTECEDENTES 1
1.1 Planteamiento del problema 1
1.2 Justificación 1
1.3 Objetivos 2
1.3.1 Objetivo general 2
1.3.2 Objetivos específicos: 2
1.4 Metodología 2
CAPÍTULO 2 4
FUNDAMENTACIÓN TEÓRICA 4
2.1 IoT 4
2.2 Adafruit IO 5
2.3 IFTTT 7
2.4 Aplicación Web móvil 8
2.5 Node MCU ESP32 10

2.6 Sensor de Humedad de suelo FC-28 12
2.7 Sensor de Temperatura DHT11 13
CAPÍTULO 3 14
DISEÑO E IMPLEMENTACIÓN 14
3.1 Invernadero 14
3.2 Diagrama de bloques de funcionamiento general del dispositivo IoT 17
3.3 Diagramas de flujo del dispositivo IoT 18
3.4 Diagramas esquemáticos 25
3.5 Cálculo de la humedad del suelo 27
3.6 Uso del servicio de Adafruit IO 28
3.7 Aplicación móvil 29
3.8 Uso de la plataforma IFTTT 34
CAPÍTULO 4 36
ANÁLISIS DE RESULTADOS 36
4.1 Pruebas de latencia 36
4.2 Retardos 39
4.3 Pérdida de paquetes 43
4.4 Pruebas de tiempo de respuesta 48
4.5 Monitoreo de la humedad relativa del suelo y temperatura 49
4.6 Encuesta 50
CAPÍTULO 5 51
CONCLUSIONES 51
RECOMENDACIONES 52
BIBLIOGRAFÍA 53
ANEXOS 56

ÍNDICE DE FIGURAS

Figura 2.1 Sectores en los que se puede aplicar el Internet de las Cosas..... 4
Figura 2.2 Logotipo de Adafruit IO 5
Figura 2.3 Bloques que se pueden añadir en el dashboard de Adafruit IO 6
Figura 2.4. IFTTT nos permite interconectar con varios de los servicios en la nube. . 7
Figura 2.5. Funcionamiento de la plataforma IFTTT 8
Figura 2.6. Formas de visualización de un contenido web en un dispositivo Android. 9
Figura 2.7. Módulo ESP32. 10
Figura 2.8. Sensor de humedad de suelo FC-28. 12
Figura 2.9. Módulo acondicionador de señal para el higrómetro FC-28. 13
Figura 2.10. Sensor de temperatura DHT11. 13
Figura 3.1. Plano del Invernadero realizado en AutoCAD 15
Figura 3.2. Invernadero AMPAC 15
Figura 3.3. Diagrama del proceso general de control del invernadero. 17
Figura 3.4. Diagrama de flujo de la adquisición y envío de datos al servidor. 19
Figura 3.5. Diagrama de flujo para la acción de control mediante la aplicación móvil. 20
Figura 3.6. Diagrama de flujo del control automático de la humedad. 21
Figura 3.7. Diagrama de flujo del control automático de la temperatura. 22
Figura 3.8. Diagrama de flujo de navegación de la aplicación móvil. 23
Figura 3.9. Diagrama de flujo de los mensajes de alerta IFTTT de humedad. 24
Figura 3.10. Diagrama de flujo de los mensajes de alerta IFTTT de temperatura. 25
Figura 3.11. Diagrama esquemático, ruteo de PCB. 26
Figura 3.12. Vista 3D de la placa PCB 26
Figura 3.13. Diagrama de conexión de elementos 27
Figura 3.14. Histograma de la humedad relativa de la cama 4. 28
Figura 3.15. Tabla de datos generada por Adafruit IO 29
Figura 3.16. Pantalla de bienvenida de la aplicación móvil 30
Figura 3.17. Interfaz para ingresar credenciales 31
Figura 3.18. Panel de Visualización de sensores 32
Figura 3.19. Pantalla 3 de la aplicación móvil 33
Figura 3.20. Creación del activador en IFTTT 34
Figura 3.21. Establecer el chat y el mensaje que será enviado a Telegram 35

Figura 3.22. Mensajes de advertencia 35
Figura 4.1. Prueba de latencia realizada a las 10:00 am 37
Figura 4.2. Prueba de latencia realizada a las 12:00 pm. 37
Figura 4.3. Prueba de latencia realizada a las 14:00 pm. 38
Figura 4.4. Prueba de latencia realizada a las 17:00 pm. 38
Figura 4.5. Prueba de latencia realizada a las 19:00 pm. 38
Figura 4.6. Código en Arduino para la prueba de pérdida de paquetes 44
Figura 4.7. Caracteres recibidos en el servidor web a las 10:18 am 44
Figura 4.8. Caracteres recibidos en el servidor web a las 12:57 am 44
Figura 4.9. Caracteres recibidos en el servidor web a las 14:27 pm 45
Figura 4.10. Caracteres recibidos en el servidor web a las 17:19 pm 45
Figura 4.11. Caracteres recibidos en el servidor web a las 19:38 pm 46
Figura 4.12. Código para recibir el dato en el dispositivo IoT 46
Figura 4.13. Resultados del envió de datos hacia el dispositivo IoT a las 10:23 am . 46
Figura 4.14. Resultados del envió de datos hacia el dispositivo IoT a las 13:23 pm. 47
Figura 4.15. Resultados del envió de datos hacia el dispositivo IoT a las 14:45 pm. 47
Figura 4.16. Resultados del envió de datos hacia el dispositivo IoT a las 17:55 pm. 48
Figura 4.17. Resultados del envió de datos hacia el dispositivo IoT a las 19:42 pm. 48

ÍNDICE DE TABLAS

Tabla 2.1 Características para la elección del módulo. 11
Tabla 3.1. Elementos del circuito eléctrico y sus potencias 16
Tabla 3.2. Temperatura y Humedad para cultivo de hortalizas 17
Tabla 4.1. Prueba de retardos realizada a las 10:00 am 39
Tabla 4.2. Prueba de retardos realizada a las 12:00 am 40
Tabla 4.3. Prueba de retardos realizada a las 14:00 pm 41
Tabla 4.4. Prueba de retardos realizada a las 17:00 pm 42
Tabla 4.5. Prueba de retardos realizada a las 19:00 pm 43
Tabla 4.6. Prueba de tiempo de respuesta 49
Tabla 4.7 Valores recolectados de los sensores de humedad de suelo y temperatura ambiente. 50

RESUMEN

El desconocimiento de factores como la humedad relativa del suelo y la temperatura ambiente ha sido una problemática al momento de cultivar en el invernadero AMPAC, ya que estos son factores que inciden directamente en su producción, por lo cual se propuso el desarrollo de un dispositivo IoT para el mejoramiento de la producción de brócoli, cebolla, pepino, pimiento y tomate, este dispositivo posee un módulo ESP32 y sensores que permiten la adquisición y visualización de datos como; temperatura ambiente y humedad relativa del suelo tiempo real, estos datos son enviados al servidor de Adafruit IO en la nube y se almacenan en su base de datos, también el dispositivo permite tener el control de electroválvulas, focos y un ventilador, además con el uso de Android Studio se desarrolló una aplicación que posibilita a la persona a cargo el monitoreo de los parámetros de forma remota, de esta manera se pueden reducir los traslados del personal encargado al invernadero AMPAC, finalmente con la vinculación del servicio IFTTT se envían mensajes de alerta cuando los parámetros de humedad y temperatura hayan salido de su rango, de manera que el encargado del invernadero AMPAC pueda usar esta información y tomar las acciones necesarias en bienestar del cultivo.

ABSTRACT

The lack of knowledge of factors such as relative soil humidity and ambient temperature has been a problem when growing in the AMPAC greenhouse, since these are factors that directly affect their production, so it was proposed the development of an IoT device for the improvement of the production of broccoli, onion, cucumber, bell pepper and tomato, this device has an ESP32 module and sensors that allow the acquisition and visualization of data such as; These data are sent to the Adafruit IO server in the cloud and stored in its database. The device also allows the control of solenoid valves, spotlights and a fan, and with the use of Android Studio an application was developed that enables the person in charge to monitor the parameters remotely, Finally, with the linking of the IFTTT service, alert messages are sent when the humidity and temperature parameters have gone out of range, so that the person in charge of the AMPAC greenhouse can use this information and take the necessary actions for the welfare of the crop.

INTRODUCCIÓN

En la parroquia de Cangahua los campesinos han optado por cambiar la agricultura por la actividad agropecuaria, es decir cultivar en invernaderos pequeños para luego comercializar sus productos en ferias agroecológicas de tal manera que consiguen un ingreso extra para sus familias, pero el modo de cultivo se ha mantenido de acuerdo a sus conocimientos, sembrando y cosechando productos de manera tradicional, esto ha generado que algunas veces existan pérdidas en los productos cultivados. (Beltrán, 2013)

El invernadero AMPAC ubicado en la parroquia de Cangahua no cuenta con un sistema de adquisición que permita conocer los parámetros como humedad relativa del suelo y temperatura ambiente dentro del invernadero, esto conlleva a que sea complicado mejorar la producción de brócoli, cebolla, pepino, pimiento y tomate dentro del invernadero.

En este proyecto técnico se desarrolló un dispositivo IoT para el invernadero AMPAC, capaz de realizar el control de riego y un monitoreo de la humedad y la temperatura para un ambiente ideal para mejorar la producción brócoli, cebolla, pepino, pimiento y tomate, con el aporte de tecnologías como el NodeMCU ESP32 que es el módulo encargado de enviar información al servidor de Adafruit IO mediante comunicación WIFI, y a su vez interactuar con los diferentes actuadores, todo esto mediante una aplicación web desarrollada en Android Studio que permite al usuario realizar este control y monitoreo del invernadero de forma remota, también este dispositivo IoT cuenta un sistema de alertas para valores fuera del rango permitido de humedad y temperatura, mediante Telegram que es uno de los servicios con los que cuenta la plataforma IFTTT y así mantener informado al usuario mediante mensajes ante posibles eventualidades.

A continuación se describen los contenidos de los capítulos y anexos:

En el Capítulo 1: Planteamiento del problema, Justificación, Objetivo general, Objetivos específicos y Metodología.

En el Capítulo 2: La fundamentación teórica referente a conceptos necesarios para la comprensión del proyecto como IoT, Adafruit io, IFTTT, Android Studio, y varios componentes necesarios para el desarrollo del dispositivo IoT.

En el Capítulo 3: Se detalla el diseño e implementación a nivel de hardware y software correspondientes al dispositivo IoT.

En el Capítulo 4: Se detalla el análisis de las pruebas realizadas para verificar la latencia, retardos y pérdidas de paquetes.

En el Capítulo 5: Se muestran las conclusiones y recomendaciones.

En los Anexos: Se presenta el código de programación del módulo ESP32 y del desarrollo de la aplicación web, además de la instalación del dispositivo IoT en el invernadero AMPAC y los resultados de la encuesta realizada a los miembros de la Asociación AMPAC

CAPÍTULO 1
ANTECEDENTES

1.1 Planteamiento del problema

Según PDOT del GAD parroquial de Cangahua afirma que la parroquia cuenta con temperaturas máximas de 20ºC y mínimas de 0ºC y una humedad relativa mayor al 80%, siendo estas condiciones inadecuadas para el cultivo de algunas especies que se ven afectadas por estos factores (GAD Cangahua, 2020), por tal motivo se ha implementado un sistema de cultivo en invernaderos gracias a la ayuda de algunas entidades públicas y privadas, y aunque de esta manera se ha logrado mejorar algunas producciones, esto no puede solventar la problemática por completo.

El invernadero AMPAC no cuenta con un sistema de adquisición de datos para verificar que los parámetros de cultivo sean los idóneos y cultivar en él trae consigo algunos inconvenientes como; el poco control de la temperatura interna y la humedad del suelo que son factores que afectan a la producción directamente, por lo que, si no se cuenta con personal capacitado y equipo adecuado para el control, probablemente exista perdida de un porcentaje de productos.

Por lo anterior mencionado, es necesario el uso de la tecnología para implementar un dispositivo IoT que permita mejorar la producción del invernadero AMPAC, cuyas funciones serán; adquisición de datos como temperatura ambiente, humedad relativa del suelo de cada una de las camas, controlar el sistema de riego del invernadero y controlar actuadores, dichos datos podrán ser revisados por el personal a cargo del invernadero por medio de una aplicación móvil.

1.2 Justificación

El desarrollo de un dispositivo IoT para el invernadero AMPAC como proyecto de ámbito tecnológico beneficiará al grupo de personas que pertenecen a la Asociación de Mujeres Productoras Agroecológicas de Cangahua con una mejora de calidad en sus productos tales como su apariencia, textura, esto aportara directamente en el aumento de sus ventas, además con la aplicación móvil se puede controlar las distintas acciones como la cantidad de riego, lectura de temperatura y humedad relativa del

suelo desde cualquier parte con acceso a internet esto reducirá perdidas por deficiencia en el cuidado que es uno de los principales problemas que AMPAC reporta diariamente, adicional optimizar los tiempos que las mujeres ocupan diariamente en el cuidado de sus productos puesto que podrán obtener datos en tiempo real desde su dispositivo celular o computadora.

Con este proyecto se busca fortalecer la agricultura en el sector de CANGAHUA por medio del aporte de la tecnología para mejorar su productividad.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un Dispositivo IoT en el invernadero mixto AMPAC para el mejoramiento de la producción de brócoli, cebolla, pepino, pimiento y tomate.

1.3.2 Objetivos específicos:

- Analizar los requerimientos del invernadero AMPAC para la determinación de los sensores y su conectividad.
- Implementar el dispositivo IoT para el invernadero AMPAC cumpliendo con los parámetros que mejore los cultivos de brócoli, cebolla, pepino, pimiento y tomate.
- Desarrollar una aplicación en un dispositivo móvil para el conocimiento de los parámetros del invernadero AMPAC.
- Realizar pruebas para la verificación de latencia, retardos y pérdidas de paquetes para la comprobación del funcionamiento del dispositivo IoT.

1.4 Metodología

- **Metodología analítica:** Mediante este método se podrá analizar las variables físicas que nos permitirán buscar los sensores adecuados para el proyecto.

- **Metodología deductiva:** En base a la información recolectada se creará el dispositivo IoT y la red de comunicaciones.

- **Metodología experimental:** Por medio de las pruebas realizadas verificaremos el funcionamiento del dispositivo IoT y de la red de comunicaciones.

CAPÍTULO 2
FUNDAMENTACIÓN TEÓRICA

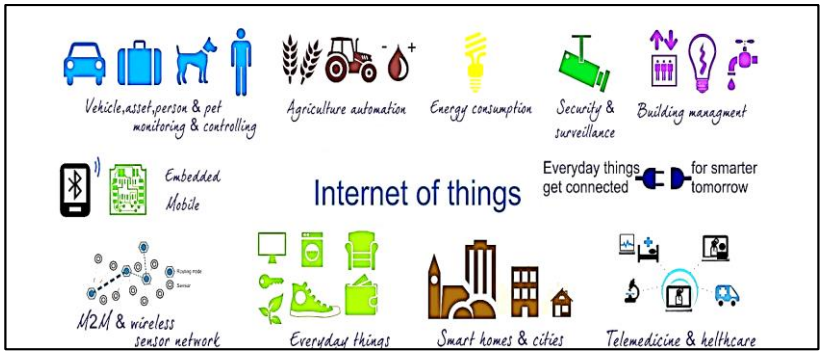
En el Capítulo 2 se analizan las fuentes primarias en los cuales se fundamenta la realización de la investigación, que ayudarán al lector a tener una idea clara acerca del tema tratado. Además, se presentan conceptos y bases teóricas sobre los cuales se establecerá la solución del problema.

2.1 IoT

El "internet de las cosas" está cada vez más presente en la vida cotidiana, y manifiesta la evolución del Internet ya que por medio de ello se puede reunir, analizar y distribuir datos que posteriormente se convertirán en información. Por tal motivo se convirtió en una herramienta importante debido a que se define una interconexión entre las personas y/u objetos que tengan capacidad de intercambiar datos. (Barrera, 2018).

En la Figura 2.1 se puede observar algunos de los sectores en los cuales podemos encontrar el IoT, aplicaciones que van desde controlar los sistemas eléctricos de un apartamento hasta controlar un proceso complejo en un sector industrial.

Figura 2.1 Sectores en los que se puede aplicar el Internet de las Cosas



El Internet de las Cosas y la sostenibilidad medioambiental.

Fuente: (Mancilla & De Armentia,2015). Recuperado de: https://revistaingenieria.deusto.es/el-internet-de-las-cosas-y-la-sostenibilidad-medioambiental/

En su mayoría la manera de interconexión entre objetos y personas se la realiza por medio de dispositivos móviles, muchos de los objetos que encontramos presentes en nuestra vida social ,familiar y empresarial se pueden conectar al IoT, con el objetivo de mejorar nuestra calidad de vida.(Mendieta, Herrera, & Peña, 2019)

2.2 Adafruit IO

La plataforma Adafruit IO ha sido diseñada con el propósito de transmitir, registrar e interactuar con los datos que se obtienen de sensores a través de la conexión de un controlador a la nube (Rubell, 2021). Adafruit IO cuenta con bibliotecas cliente como API REST y MQTT. (Cooper, 2016)

En la Figura 2.2 se muestra el logotipo de la plataforma Adafruit IO la cual desempeñará la función de servidor para realizar la publicación y suscripción a feeds mediante el protocolo MQTT.

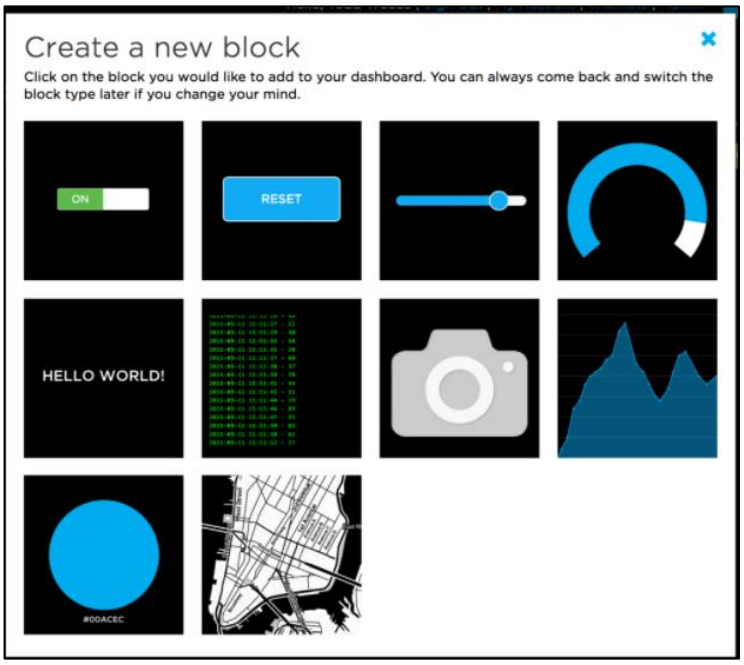
Figura 2.2 Logotipo de Adafruit IO

Adafruit IO.

Fuente: (Cooper, 2017). Recuperado de: https://learn.adafruit.com/adafruit-io

Adafruit permite crear un dashboard (Panel de control) en el cual se puede visualizar datos y controlar proyectos por medio de widgets o también llamados bloques los cuales pueden ser interruptores, botones, sliders, texto, gráficas en líneas de tiempo, luces indicadoras, entre otros bloques como se observa en la Figura 2.3.(Treece, 2019)

Figura 2.3 Bloques que se pueden añadir en el dashboard de Adafruit IO



Adafruit IO Basics: Dashboards.

Fuente: (Treece,2018). Recuperado de: https://learn.adafruit.com/adafruit-io-basics-dashboards

Estos bloques van a ser asociados a feeds que son el núcleo de Adafruit, en él se almacenarán datos los cuales serán enviados desde el controlador, así como también datos que serán enviados desde la plataforma al controlador, por ello se debe crear un feed para cada dato.(Treece, 2018)

La plataforma posibilita ver los datos en tiempo real, los cuales serán almacenados en una base de datos provista por la plataforma durante 30 días, la base de datos puede ser descargada en formato CSV, en el archivo podemos encontrar los valores o estados del feed, id_feed y la fecha de cambio del valor o estado.

Además, Adafruit IO permite crear disparadores que permitirán programar el envió de un dato al feed a una determinada hora del día.(Rubell, 2020)

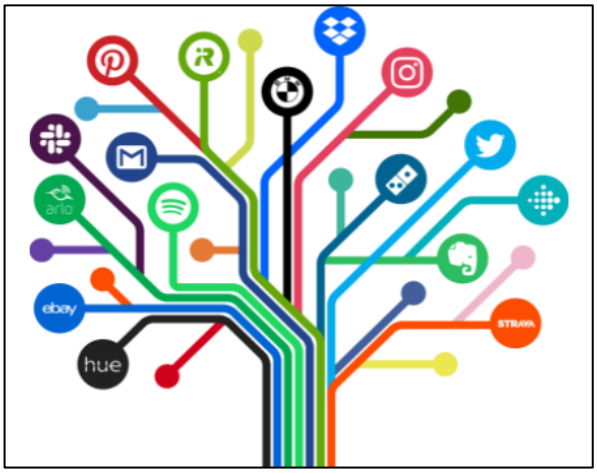
2.3 IFTTT

IFTTT es la abreviatura de “If This Then That” o su equivalente “Si es eso entonces hacer esto”, es una plataforma que permite conectar servicios en la nube.

Existen dos tipos de conexiones; la conexión con servicios no autenticados que no requieren que el usuario posea una cuenta, un ejemplo de este tipo de servicios es “Weather Underground” y el otro tipo de conexión se da con servicios autenticados, los cuales tienen como requisito principal que el usuario posea una cuenta y una contraseña existentes, primero deberá ingresar al servicio al cual se va a conectar y otorgar los permisos solicitados por IFTTT. (IFTTT, 2021)

En la Figura 2.4 se representan algunos de los servicios con los cuales permite interactuar la plataforma IFTTT con el fin de programar eventos por medio de la nube.

Figura 2.4. IFTTT nos permite interconectar con varios de los servicios en la nube.

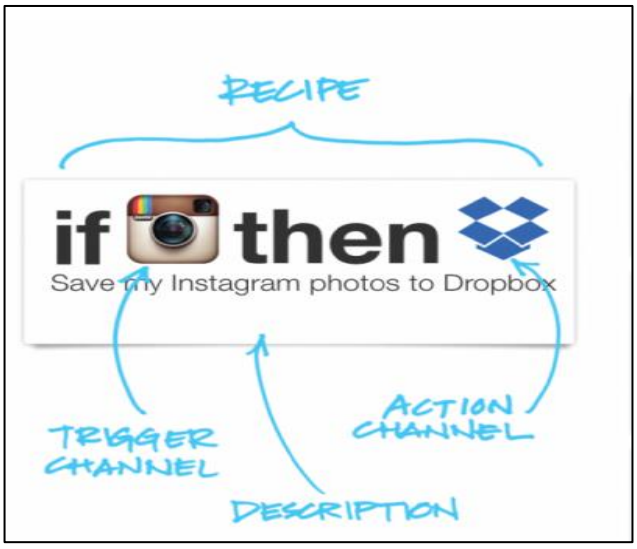


Empiece a conectar su mundo.

Fuente: (IFTTT, 2021). Recuperado de: https://ifttt.com/home

La Figura 2.5, representa el funcionamiento de IFTTT, el cual solicita que se cree un "Trigger" o disparador que debe basarse en algún evento que suceda en uno de los servicios conectados a IFTTT, a partir de este disparador se define una acción a realizar en otro servicio, de tal manera que se conectaran los dos servicios.(Mi, Zhang, Qian, & Wang, 2017)

Figura 2.5. Funcionamiento de la plataforma IFTTT



«If This Then That»: Haz que Internet trabaje para ti.

Fuente: (Escudero, 2017). Recuperado de: https://www.donostik.com/ifttt-if-this-then-that/

En definitiva, la plataforma IFTTT está dirigida a todos los usuarios de internet que tenga como propósito automatizar acciones a través del internet, aprovechar al máximo todas las tareas que podemos realizar en la plataforma y hacer que "el internet trabaje por nosotros". (Escudero, 2021)

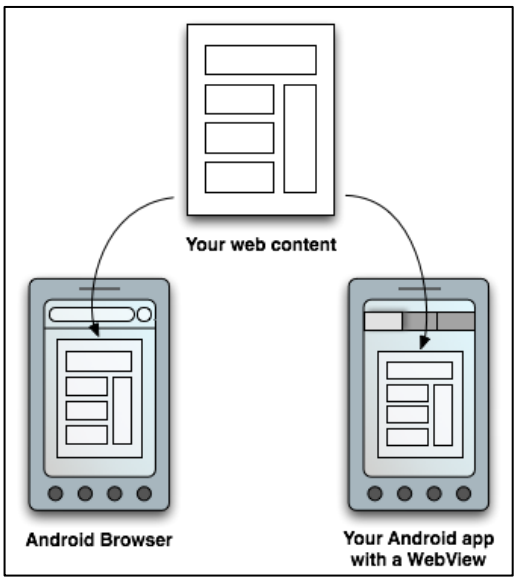
2.4 Aplicación Web móvil

Las tecnologías móviles que hoy en día conocemos fueron desarrolladas en sus inicios para poder usarlas en computadoras de escritorio, pero con el pasar de los años la tecnología ha avanzado por tal motivo han aparecido dispositivos más sofisticados,

reducidos de tamaño, con conexión inalámbrica y debido a ello las tecnologías web también presentaron una evolución notable. (Garita, 2013)

En la Figura 2.6, se muestran las dos formas de visualización de una página web en un dispositivo Android, las cuales son: un navegador web o una aplicación de Android con una WebView.

Figura 2.6. Formas de visualización de un contenido web en un dispositivo Android.



Contenido basado en la web.

Fuente: (Android Developers,2020). Recuperado de: https://developer.android.com/guide/webapps?hl=es-419

De este constante desarrollo tecnológico varias de las formas convencionales de publicidad, ventas y servicios han dado un gran salto al mundo de las aplicaciones móviles, sin embargo, se conoce que varias de las aplicaciones móviles nativas llevan un proceso que demanda más trabajo en su creación, por lo cual, ya se está implementando varias modificaciones para solventar dicho trabajo como es el caso de las aplicaciones híbridas, que contienen a un componente de navegador en una aplicación. (Mole, 2020)

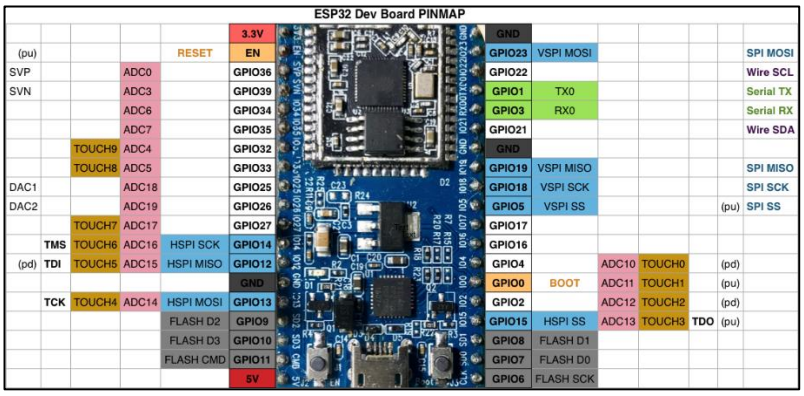
En la actualidad varias de las aplicaciones en parte se encuentran usando lo denominado WebView de esta manera se corrobora que la WebView es parte esencial de una aplicación móvil. La Webview tiene en sí, una serie de interfaces de programación de aplicaciones o conocido como API, permitiendo a la aplicación interactuar con servidores web.(Imamura et al., 2021)

2.5 Node MCU ESP32

El ESP32 es un módulo WIFI y Bluetooth, que está dirigido para una variedad de aplicaciones, estas pueden ser; elaborar una red de sensores, codificación de voz, entre otras.(Systems Espressif, 2021). El módulo puede ser programado por medio de Arduino Integrated Development Environment (IDE). (Rai & Rehman, 2019)

Como se puede observar en la Figura 2.7, el módulo cuenta con pines denominados GPIO que se los puede configurar como entrada y salida, además de pines ADC o TOUT los cuales sirven como receptores de señales analógicas.(Owner, 2017)

Figura 2.7. Módulo ESP32.



Instalando el ESP32.

Fuente: (Prometec, 2018). Recuperado de: https://www.prometec.net/instalando-esp32/

El módulo ESP32 es una herramienta que ayuda potencialmente a la realización de proyectos de hogar inteligente, automatización, aplicaciones de audio, aplicaciones

IoT basadas en la nube, entre otras más (Babiuch, Foltynek, & Smutny, 2019). La placa opera bajo protocolos de comunicación como: TCP/IP, Wi-Fi, Protocolo 802.11 b/g/n/e/i, Bluetooth, Protocolos: V4.2 BR/EDR and BLE. (Maier, Sharp, & Vagapov, 2017)

Para realizar la elección del módulo específico para la implementación del dispositivo IoT se toman en cuenta las características principales como se observa en la Tabla 2.1. De acuerdo a esta tabla se eligio el modulo ESP32 ya que cuenta con un procesador de bajo consumo, su frecuencia de operación es mayor a los otros dispositivos, tiene más capacidad de memoria SRAM y memoria Flash, cuenta con un mayor numero de entradas analogicas y digitales lo cual es muy importante para la realización del proyecto, además posee 16 pines PWM, cabe recalcar que posee tecnología WIFI y Bluetooth, finalmente su bajo costo fue un factor importante para optar por el módulo.

Tabla 2.1 Características para la elección del módulo.

CARACTERÍSTICAS	**ARDUINO UNO**	**ESP8266**	**ESP32**
Procesador	ATMEGA 328	Tensilica LX106	Tensilica Xtensa X36
Velocidad	16 MHz	80 Mhz (hasta 160 Mhz)	160 Mhz (hasta 240 Mhz)
SRAM	2KB	160 KB	512 KB
Memoria flash	32KB	4MB	16MB
ADC	6	1	18
DAC	14	NO	2
PWM	6	8	16
Costo	29$	7.90$	12.50$

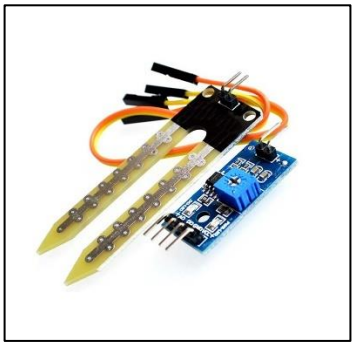
Elaborado por: Wilmer Necpas, Jaime Quishpe.

2.6 Sensor de Humedad de suelo FC-28

Los sensores son una parte importante del IoT, son dispositivos que pueden ser instalados en cualquier lugar, poseen la gran ventaja de poder adquirir datos para poder anticiparse a problemas mediante la información recopilada. (Salazar & Silvestre, 2017)

En la Figura 2.8, se observa el higrómetro FC-28 que sirve para medir la humedad del suelo, son utilizados en sistemas de riego para conocer el momento exacto en que se activen las válvulas de agua.

Figura 2.8. Sensor de humedad de suelo FC-28.

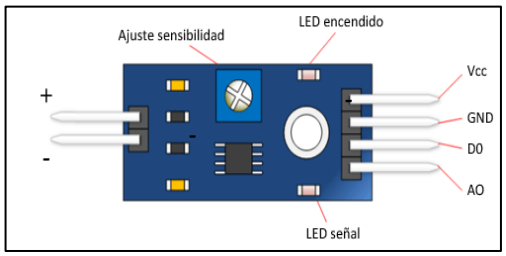


Sensor FC-28.

Fuente: (ElectroStore, s.f). Recuperado de: https://grupoelectrostore.com/shop/sensores/temperatura/modulo-sensor-de-humedad-y-temperatura-de-suelo-fc-32-higrometro/

El funcionamiento del sensor consiste en medir la resistividad entre los dos electrodos, a menor resistividad el suelo es húmedo y a mayor resistividad significa que el suelo es seco, estos electrodos están acoplados a un circuito de acondicionamiento de señal de 4 pines Figura 2.9, dos de alimentación, una salida digital y una salida analógica que en este proyecto nos dará lecturas de 0 a 5v, el cual posteriormente será acondicionado de 0 a 100% con la ayuda del controlador. (Llamas, 2021)

Figura 2.9. Módulo acondicionador de señal para el higrómetro FC-28.



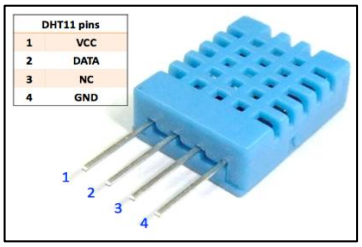
Medir la humedad del suelo con Arduino e higrómetro fc-28.

Fuente: (Llamas, 2016). Recuperado de: https://www.luisllamas.es/arduino-humedad-suelo-fc-28/

2.7 Sensor de Temperatura DHT11

La Figura 2.10, presenta un sensor DHT11 el cual está compuesto de un sensor de humedad capacitivo y un termistor NTC además de un circuito de acoplamiento de señal, el cual nos entregan el dato de temperatura (Zhou, Zhou, Kong, & Cai, 2012). Consta de 4 pines; el pin 1 de alimentación 3,5 a 5 Vdc, el pin 2 es la salida del dato, el pin 3 no se conecta y el pin 4 es la conexión a GND.(Swetha, Sn, Nevetha, Sarathy, & Deepa, 2019)

Figura 2.10. Sensor de temperatura DHT11.



Sensor DHT11.

Fuente: (Martínez, 2015). Recuperado de: https://simplesoftmx.blogspot.com/2015/04/sensor-dht11-con-pic-16f628a-y-pic-ccs.html

CAPÍTULO 3
DISEÑO E IMPLEMENTACIÓN

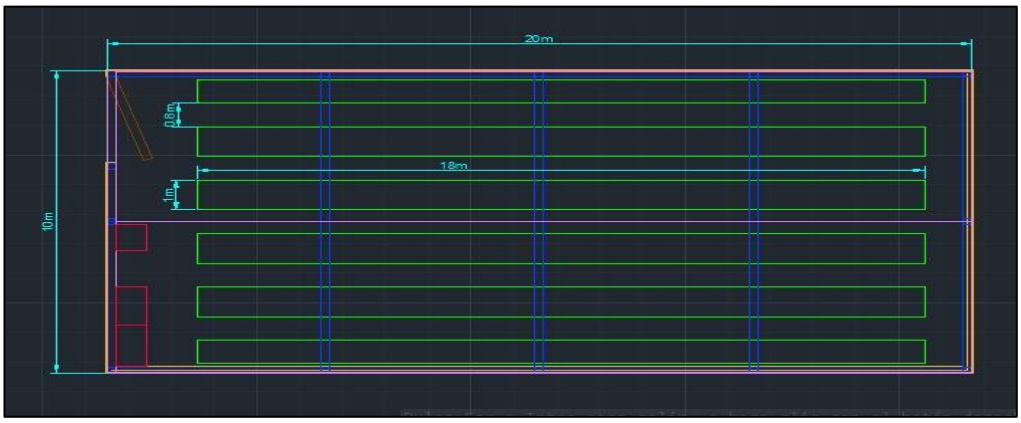
En el presente capítulo, se realiza un análisis de los requerimientos técnicos del invernadero del grupo AMPAC, también se realiza la descripción del software y hardware desarrollado para el aporte tecnológico para el invernadero, se muestra el diagrama de bloques del funcionamiento general del dispositivo IoT, también diagramas de flujo de los subprocesos como; adquisición de datos, envío de datos, comunicación, controles automáticos, navegación en la aplicación móvil, así como el diagrama del servicio de mensajería de alerta basado en IFTTT.

3.1 Invernadero

En la Figura 3.1, se pueden observar las dimensiones del invernadero AMPAC, las cuales son; 20 metros de largo, 10 metros de ancho y 3 metros de alto, en su interior consta de 6 camas (espacios delimitados para sembrar), cada una tiene 1 metro de ancho y 18 metros de largo, y la separación que existe entre camas es de 0,80 metros, la altura de las camas es en promedio 0,50 metros.

El invernadero AMPAC está ubicado en la parroquia de Cangahua, es un invernadero tipo túnel, el material que lo cubre es plástico como lo muestra la Figura 3.2, y no cuenta con ningún tipo de adquisición de datos como humedad relativa del suelo y temperatura ambiental que son factores que influyen directamente en la produccion, tambien no posee un sistema de riego que pueda ayudar a mejorar la productividad, además de lo mencionado anteriormente el invernadero no está en constante vigilancia ya que el personal encargado también realizan otras actividades como ganadería o comercio de hortalizas.

Figura 3.1. Plano del Invernadero realizado en AutoCAD



Dimensiones del invernadero AMPAC.

Elaborado por: Wilmer Necpas, Jaime Quishpe

Figura 3.2. Invernadero AMPAC

Foto del Invernadero AMPAC ubicado en la parroquia de Cangahua.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

En el invernadero AMPAC se instaló una caja de distribución eléctrica, para realizar la conexión de luminarias y ventiladores, por lo cual se realizó el cálculo para dimensionar las características del interruptor magnético que se utilizará. Primero se calcula la potencia total del circuito, observar Tabla 3.1.

Tabla 3.1. Elementos del circuito eléctrico y sus potencias

ELEMENTO	**CANTIDAD**	**POTENCIA UNITARIA [W]**	**POTENCIA TOTAL [W]**
Foco	2	300	600
Ventilador	1	75	75
		TOTAL	675

Elaborado por: Wilmer Necpas, Jaime Quishpe.

Luego calculamos la corriente total del circuito con la ecuación Ec. (3.1).

$$\boldsymbol{I_c = \frac{P_T}{V}\ [A]} \qquad \text{Ec. (3.1)}$$

Siendo:

$\boldsymbol{I_c}$ **:** Corriente total del circuito [A]

$\boldsymbol{P_T}$ **:** Potencia total del circuito [W]

$\boldsymbol{V}$ **:** Voltaje del circuito [V]

Reemplazando los valores en la Ec. (3.1):

$$\boldsymbol{I_c = \frac{675}{120} = 5,62\ [A]}$$

Se recomienda utilizar un interruptor electromagnético con el equivalente superior a la corriente calculada.

El invernadero necesita contar con una temperatura ambiente que esté en el rango de 15 a 30 grados y una humedad relativa del suelo entre 50 y 80 %, para poder mejorar la producción de brócoli, cebolla, pepino, pimiento y tomate, en base a la tabla 3.2.

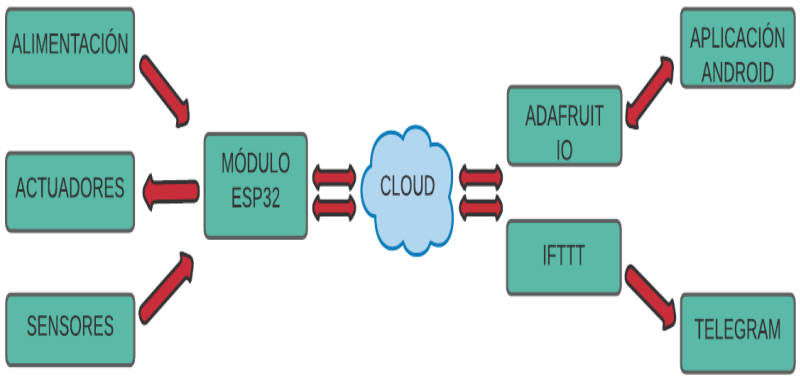
Tabla 3.2. Temperatura y Humedad para cultivo de hortalizas

Hortaliza	**Temperatura ambiente**			**Humedad relativa de suelo**	
	Temperatura mínima	**Temperatura óptima**	**Temperatura máxima**	**Humedad mínima**	**Humedad máxima**
Brócoli	5ºC	16ºC – 18ºC	30ºC	60%	75%
Cebolla	15ºC	20ºC – 25ºC	30ºC	70%	80%
Pepino	17ºC	20ºC – 25ºC	30ºC	60%	85%
Pimiento	16ºC	20ºC – 24ºC	28ºC	50%	80%
Tomate	16ºC	20ºC – 25ºC	30ºC	60%	70%

Elaborado por: Wilmer Necpas, Jaime Quishpe

3.2 Diagrama de bloques de funcionamiento general del dispositivo IoT

Figura 3.3. Diagrama del proceso general de control del invernadero.



El diagrama muestra los componentes generales del proceso de control del invernadero.

Elaborado por Wilmer Necpas y Jaime Quishpe.

En la Figura 3.3 se muestra el diagrama de bloques del proceso general del aplicativo en la cual existe una alimentación, envío de datos y recepción de señales por parte del módulo ESP32 al servidor en la nube la cual permite tener un acceso para su visualización e interacción por medio de una aplicación híbrida con el uso de las plataformas y servicios como ADAFRUIT IO, IFTTT, TELEGRAM.

El bloque de alimentación refiere al suministro de energía para el módulo ESP32 de 5V, y adicional a la alimentación de los sensores y los actuadores a 5V.

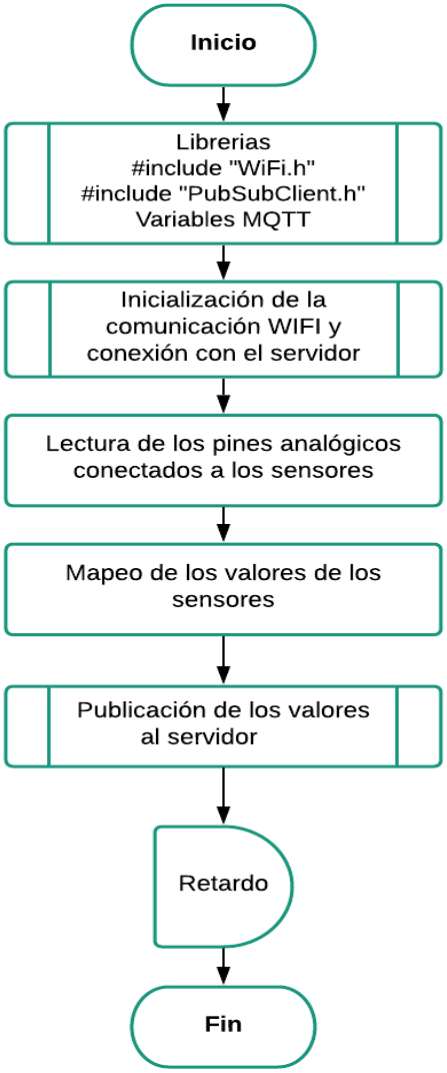
El módulo ESP32 es el encargado de enviar datos y receptar señales por medio de su comunicación WIFI, es capaz de comunicarse a la red inalámbrica y la nube, esto permite la visualización de datos y la manipulación de los actuadores por medio de la aplicación y del servidor Adafruit IO.

Adicional la plataforma IFTTT permite obtener el servicio de alerta para niveles críticos que está anclado al servidor en la nube, el cual será un mensaje al TELEGRAM del usuario en el que se especifica los niveles de alerta y su acción recomendada por defecto.

3.3 Diagramas de flujo del dispositivo IoT

En la etapa de control por algoritmo se desarrolla un diagrama de flujo en el cual se muestra la inicialización del módulo ESP32, conjunto de librerías necesarias para la comunicación y validación de conexión con la plataforma Adafruit IO por medio de una clave generada en dicha plataforma, la declaración de variables que se usarán para los distintos pines del módulo, así como su respectivo mapeo para obtener un valor referente al tipo acción y publicarlo al servidor en la nube, como se observa en la Figura 3.4.

Figura 3.4. Diagrama de flujo de la adquisición y envío de datos al servidor.

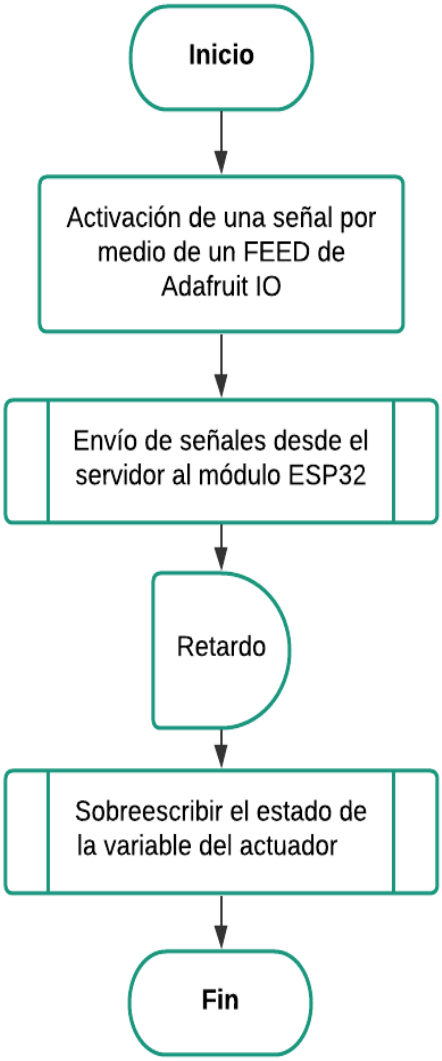


El diagrama de flujo muestra los procesos en la adquisición de datos por medio de los sensores y envío de los mismos al servidor en la nube.

Elaborado por Wilmer Necpas y Jaime Quishpe.

Como parte de acción de control, el diagrama de flujo que describe la acción desde la aplicación móvil montada en la plataforma de Adafruit IO se puede observar en la Figura 3.5, la aplicación consta con un grupo de FEEDs de control que permiten enviar una señal al módulo ESP32 para controlar actuadores.

Figura 3.5. Diagrama de flujo para la acción de control mediante la aplicación móvil.

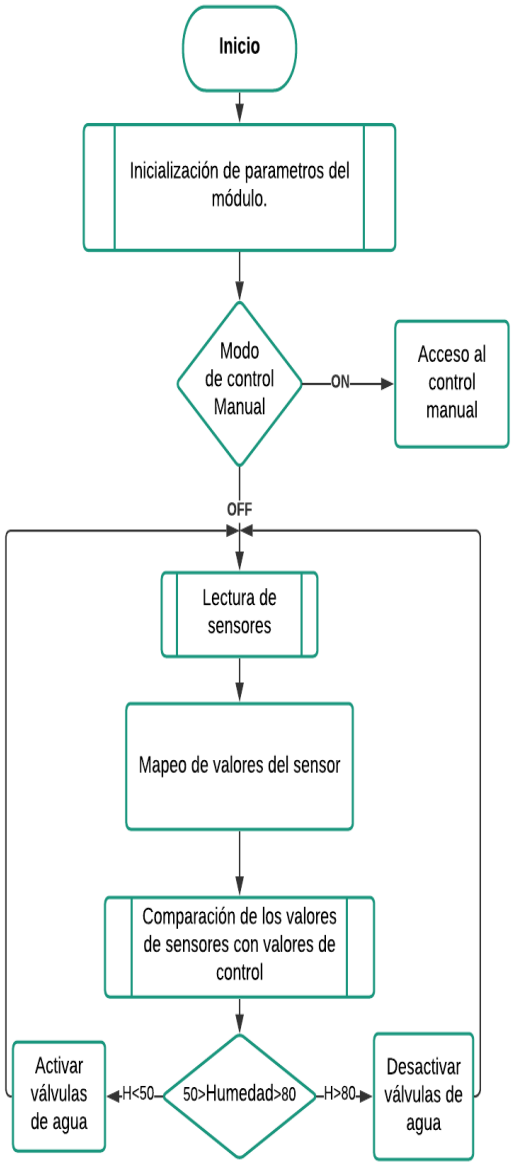


El diagrama de flujo muestra los procesos para el envío de una señal de control desde la aplicación móvil al módulo ESP32.

Elaborado por Wilmer Necpas y Jaime Quishpe.

En la Figura 3.6 se muestra el proceso de control automático de humedad en el cual se adquieren los valores del sensor de humedad y realiza una comparación si la variable humedad es menor que 50 se activan las válvulas de agua y si la humedad es mayor que 80 se desactivan las válvulas de agua.

Figura 3.6. Diagrama de flujo del control automático de la humedad.



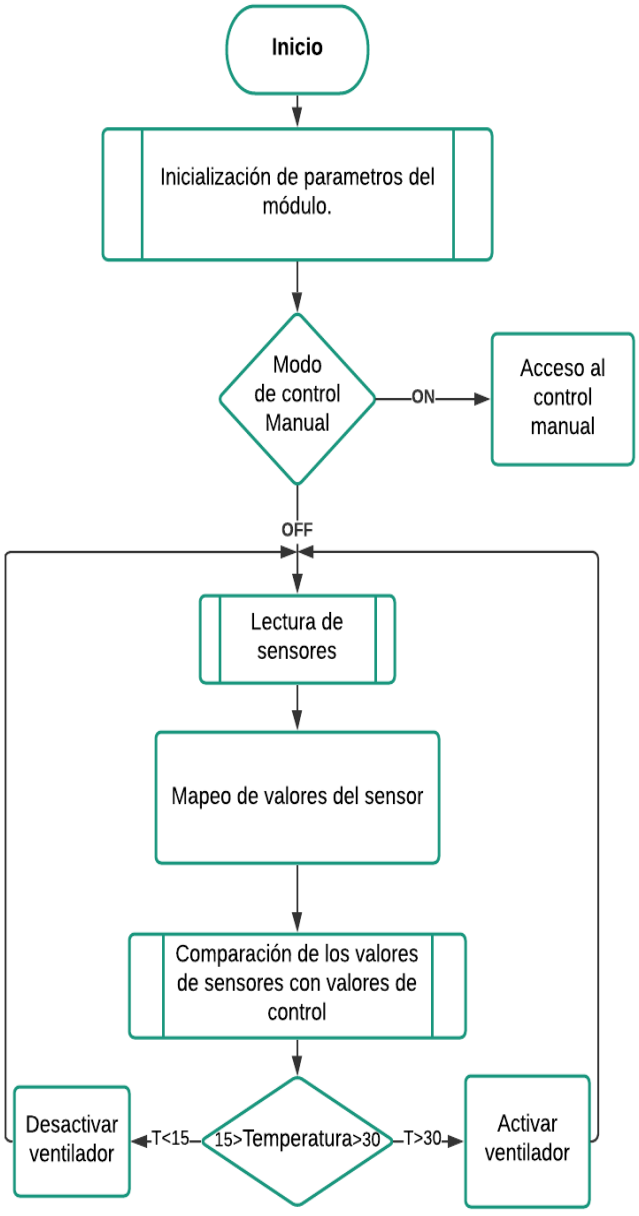
El diagrama de flujo muestra el proceso de control automático para la humedad.

Elaborado por Wilmer Necpas y Jaime Quishpe.

En la Figura 3.7 se muestra el proceso de control automático de temperatura en el cual adquiere los valores del sensor de temperatura y realiza una comparación si la variable

temperatura es menor que 15 se desactiva el ventilador y de ser el caso que la temperatura es mayor que 30 se activa el ventilador.

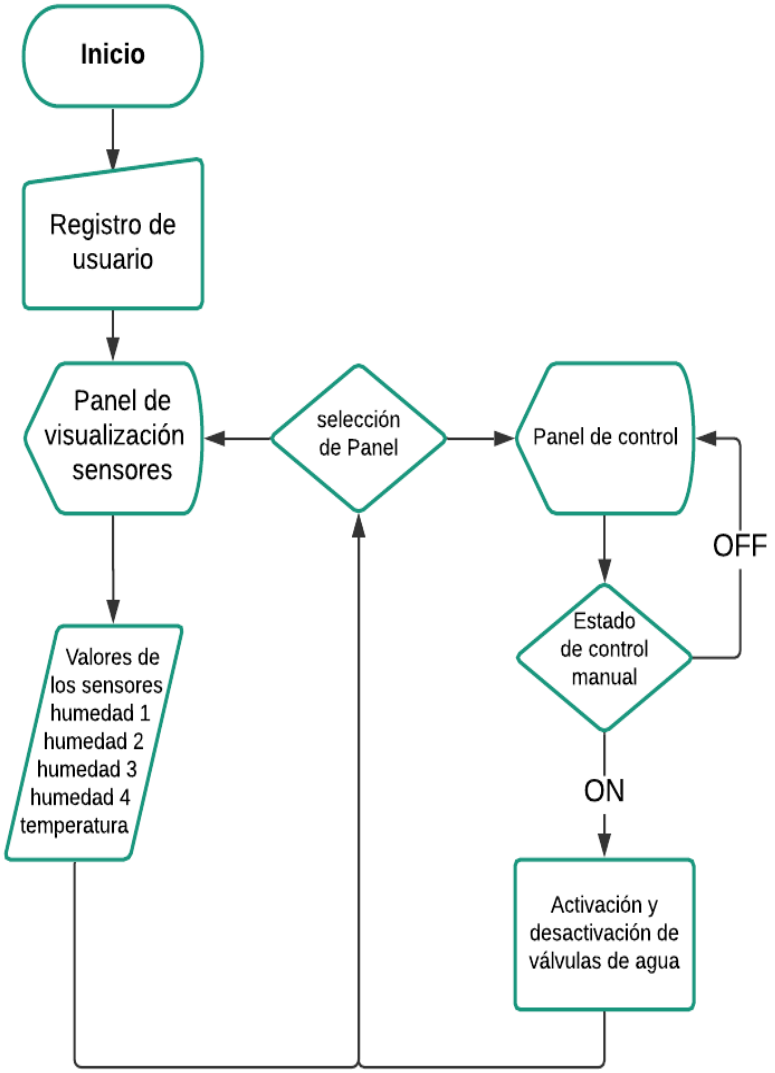
Figura 3.7. Diagrama de flujo del control automático de la temperatura.



El diagrama de flujo muestra el proceso de control automático para la temperatura.

Elaborado por Wilmer Necpas y Jaime Quishpe.

Figura 3.8. Diagrama de flujo de navegación de la aplicación móvil.



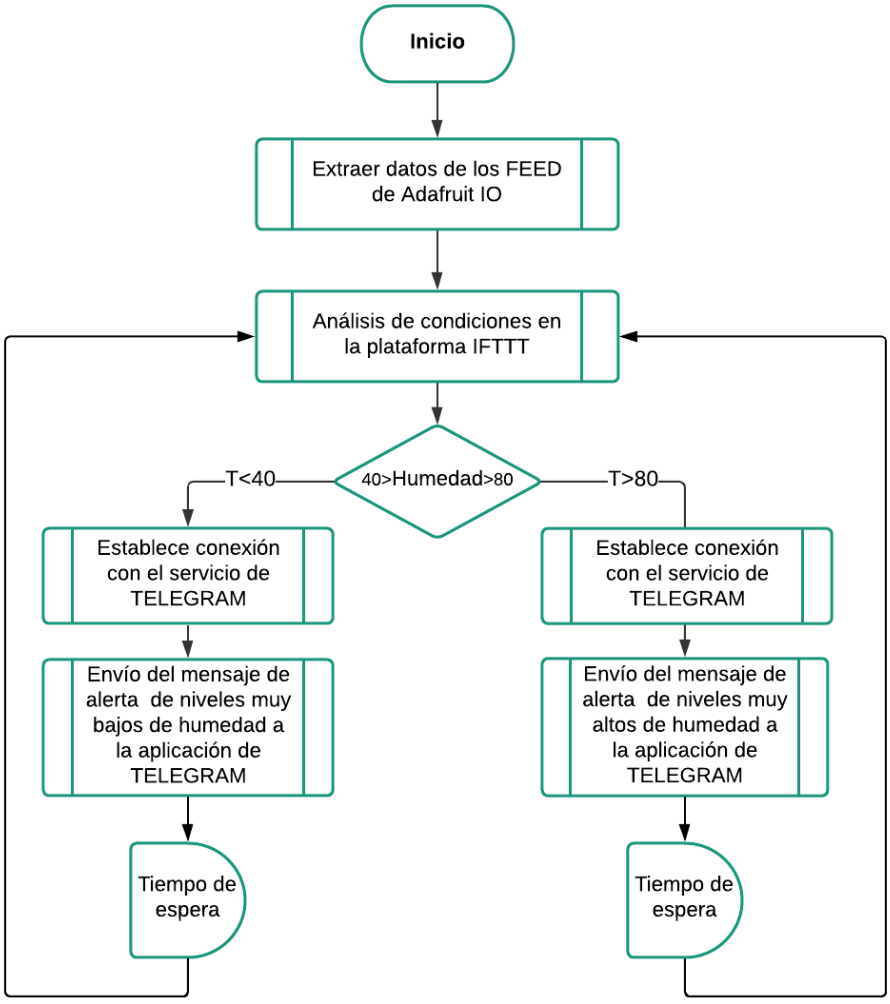
El diagrama de flujo de la funcionalidad de la aplicación móvil muestra dos plantillas principales para la visualización y control de los diferentes componentes.

Elaborado por Wilmer Necpas y Jaime Quishpe.

En la Figura 3.8 se muestra el proceso de navegación en la aplicación móvil la cual consta de un verificación de usuario que en este caso sería AMPAC el cual permite el acceso a las dos plantillas predeterminadas, en primera instancia luego de acceder con el usuario se establece en la plantilla de visualización en la cual se muestra los niveles de humedad y los niveles de temperatura del invernadero, adicional permite desplazarse a la siguiente plantilla en la que podremos activar el modo de control

manual, el cual permite tener control de la activacion y desactivacion de actuadores, esto esta establecido para casos de preferencias y emergencias.

Figura 3.9. Diagrama de flujo de los mensajes de alerta IFTTT de humedad.



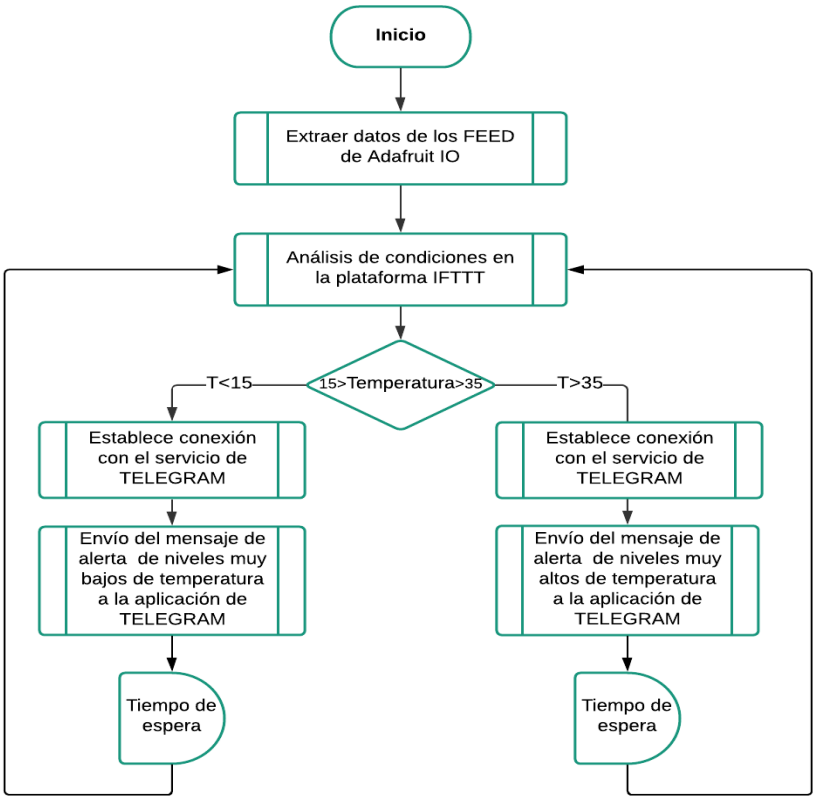
El diagrama de flujo de los mensajes de alerta para valores críticos de humedad enviados con el uso de la plataforma IFTTT a uno de sus servicios que es TELEGRAM.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

El diagrama de flujo de los envíos de alertas por medio de la plataforma IFTTT a una aplicación de mensajería como lo es TELEGRAM se muestra en la Figura 3.9, en la cual tiene inicio en la toma de valores de humedad por el servidor Adafruit IO y establecida la interacción entre Adafruit IO y IFTTT, esta plataforma de IFTTT realiza un comparación de valores establecidos para la variable designada que en este caso

será la de humedad y los valores son cruzan el rango establecido que son niveles muy críticos este enviará un mensaje con uno de sus servicios añadidos en este caso será TELEGRAM, luego de realizar la alerta al usuario este podrá realizar los cambios necesarios en el control manual. El proceso de alertas para el nivel de temperatura es muy similar simplemente que es establecida como variable el nivel de temperatura y evaluada en su rango de niveles críticos como se muestra en la Figura 3.10.

Figura 3.10. Diagrama de flujo de los mensajes de alerta IFTTT de temperatura.



El diagrama de flujo de los mensajes de alerta para valores críticos de temperatura enviados con el uso de la plataforma IFTTT a uno de sus servicios que es TELEGRAM.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

3.4 Diagramas esquemáticos

Se elaboró una placa de circuito impreso la cual nos ayuda a conectar el módulo ESP32 con los sensores y actuadores, para ello se utilizaron cabeceras hembras para acoplar

el módulo ESP32 a la placa y borneras para los sensores y actuadores como se observa en la Figura 3.11 y Figura 3.12, la dimensión de la placa PCB es 11x8cm.

Figura 3.11. Diagrama esquemático, ruteo de PCB.

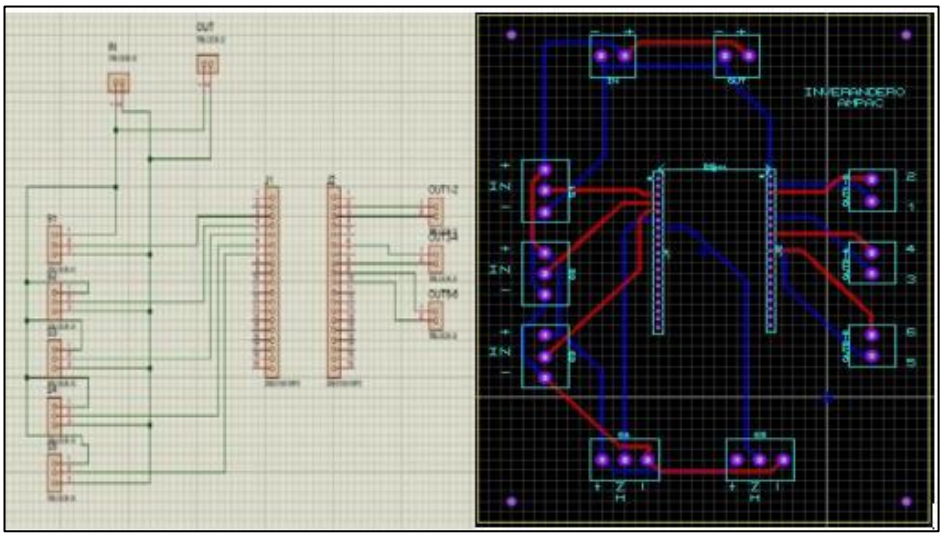
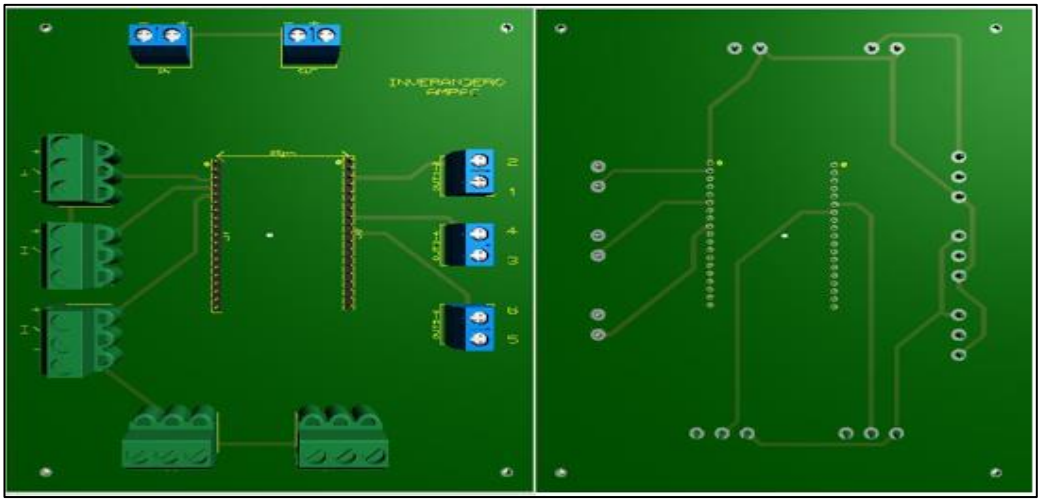


Diagrama esquemático para la realización de la PCB, ruteo y etiquetado de la PCB realizado en el software PROTEUS.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

Figura 3.12. Vista 3D de la placa PCB



Vista 3D de la placa PCB realizada en el Software PROTEUS.

Elaborado por: Wilmer Necpas, Jaime Quishpe

En la Figura 3.13 se muestra los elementos utilizados en el proyecto, en la sección 1 se tiene el módulo ESP32, en la sección 2 se muestran los sensores utilizados, en la

sección 3 se muestra la alimentación y el Switch que controla las luminarias y finalmente la sección 4 se presentan los módulos Relés que activan los actuadores.

Figura 3.13. Diagrama de conexión de elementos

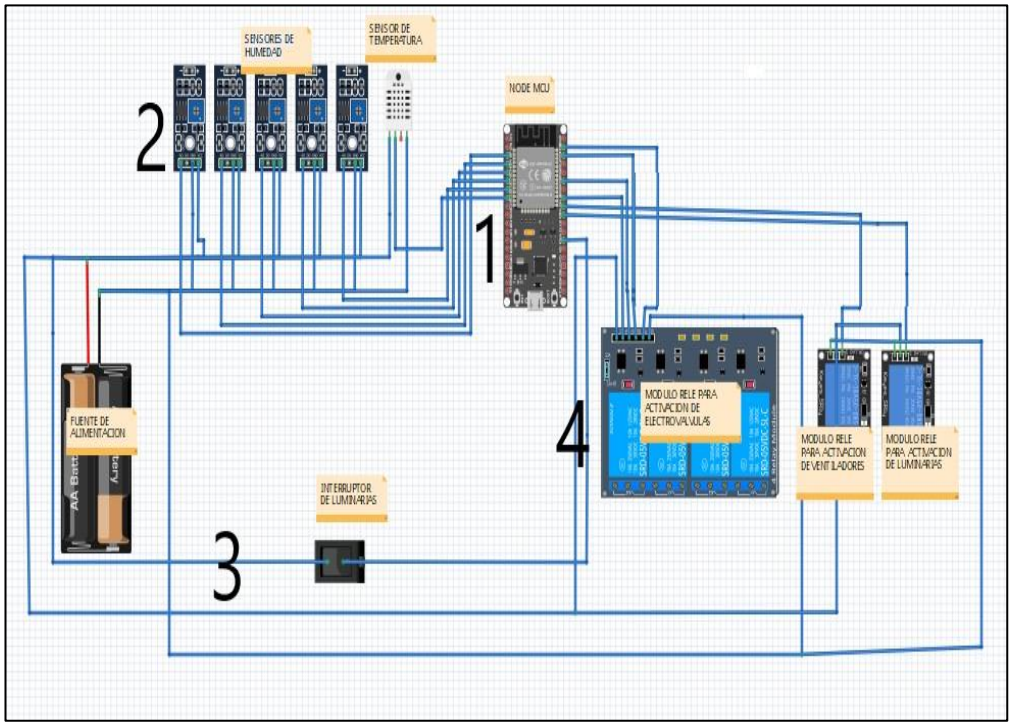


Diagrama de conexión de sensores y actuadores realizado en el software Fritzing.

Elaborado por: Wilmer Necpas, Jaime Quishpe

3.5 Cálculo de la humedad del suelo

Para obtener el valor real de la humedad del suelo por medio del sensor FC-28 se ocupará la ecuación Ec. (3.3), la cual realiza el escalonamiento de la señal para mostrar el valor de la humedad en un rango de 0 a 100%.

$$Escalonamiento = \frac{Valor\ analogico * 100}{4096} \qquad \text{Ec. (3.2)}$$

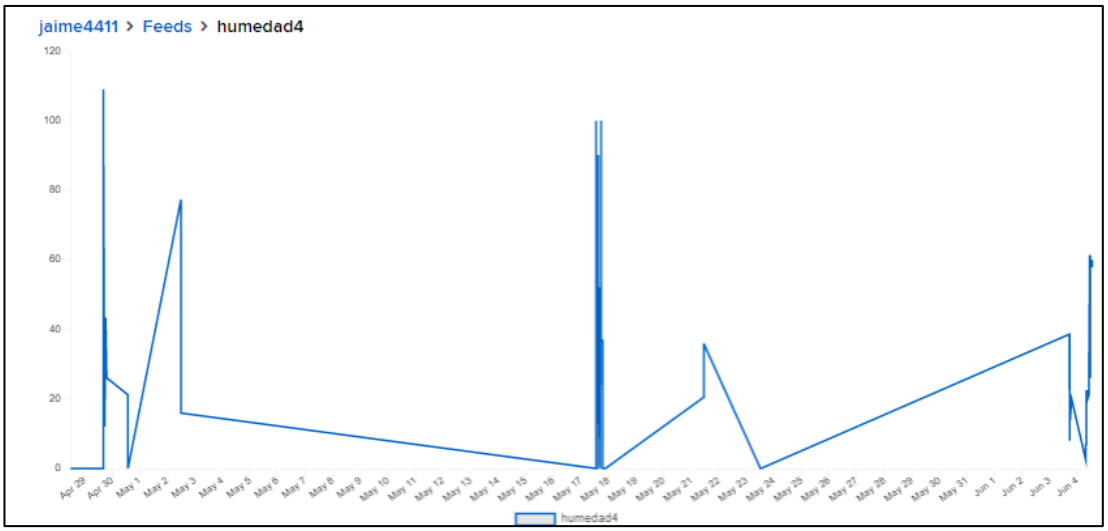
El valor analógico que ingresa al módulo ESP32 va a estar en un rango de 0 a 4096 ya que las entradas analógicas del módulo son de 12 bits, por lo que en el escalonamiento se divide para el valor de 4096 que será el valor máximo que podrá entregar el sensor.

Para la señal de la temperatura ambiente se ocupará el sensor DHT11 el cual es un sensor que nos brinda la facilidad de obtener el valor real de la temperatura sin necesidad de realizar el escalonamiento en el código de programación.

3.6 Uso del servicio de Adafruit IO

La plataforma nos permite visualizar los valores de los sensores, por medio de la creación de dashboard que contienen los bloques de visualización, cada bloque de visualización está enlazado con un FEED único el cual es encargado de recibir el dato desde el módulo ESP32 y almacenarlo, también la plataforma cuenta con histogramas que permite ver la variación de los datos en función del tiempo como se observa en la Figura 3.14.

Figura 3.14. Histograma de la humedad relativa de la cama 4.



Gráfica de la variación de los datos de humedad relativa de la cama 4 en función del tiempo.

Elaborado por: Wilmer Necpas, Jaime Quishpe

Además de los histogramas, se puede observar el cambio de los datos en tiempo real por medio de una tabla generada por la plataforma en la cual se observa la fecha y hora de la publicación del dato en la plataforma, el valor del dato y la locación. La locación

es un dato opcional que puede ser activado por el usuario si fuera necesario, como se observa en la Figura 3.15.

Figura 3.15. Tabla de datos generada por Adafruit IO

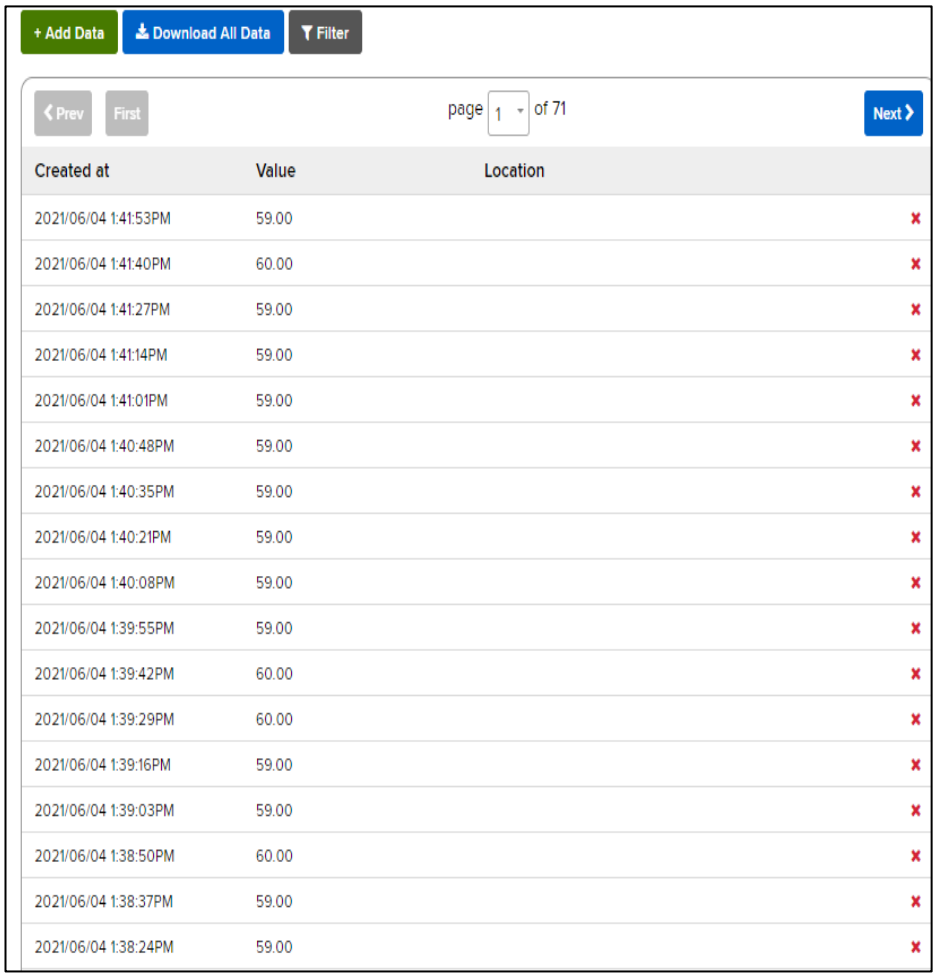


Tabla de la variación de los datos de la humedad relativa de la cama 4 generada por Adafruit IO.
Elaborado por: Wilmer Necpas, Jaime Quishpe.

3.7 Aplicación móvil

La aplicación está desarrollada para que la interacción con el usuario sea sencilla e intuitiva, consta de 3 pantallas o ventanas de visualización que serán presentadas en esta sección, la aplicación móvil se desarrolló en el software Android Studio, que es el entorno de desarrollo integrado (IDE) para la elaboración de aplicaciones para el sistema operativo Android, se hizo uso del complemento WebView el cual en la segunda y tercera ventana permitirá el acceso a la página oficial del servidor, sin la

necesidad de salirse de la aplicación, se mostrará la visualización de sensores y la selección de modo de uso del dispositivo IoT.

Figura 3.16. Pantalla de bienvenida de la aplicación móvil

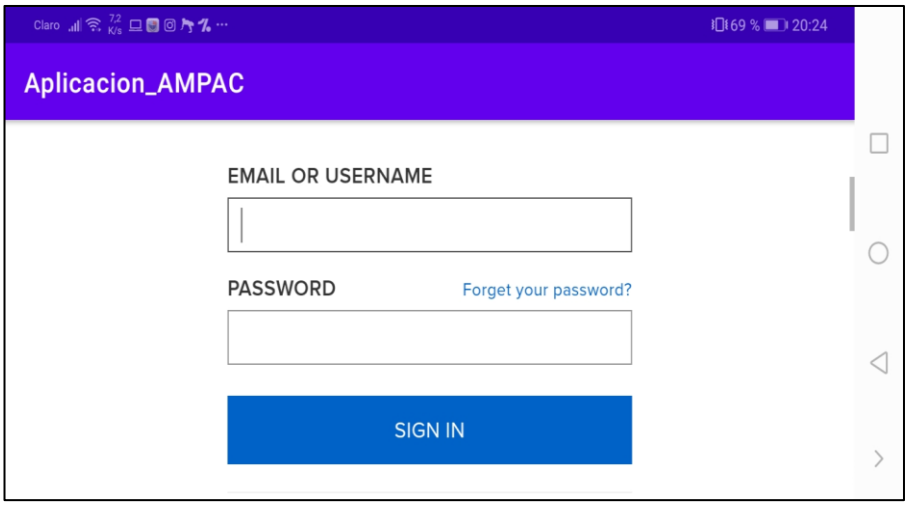
Primera pantalla de la aplicación móvil desarrollada en Android Studio.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

En la Figura 3.16 se muestra la pantalla de inicio de la aplicación presenta cuatro elementos en su interfaz, el primer elemento es el botón “Ingresar” el cual nos permite avanzar a la siguiente pantalla, el segundo elemento es el botón “Salir” que finaliza la aplicación sin necesidad de utilizar los botones por defecto del móvil, el tercer elemento muestra el logotipo de la organización AMPAC la cual es beneficiaria del

desarrollo del dispositivo IoT, y por último el logo de la Universidad Politécnica Salesiana.

Figura 3.17. Interfaz para ingresar credenciales



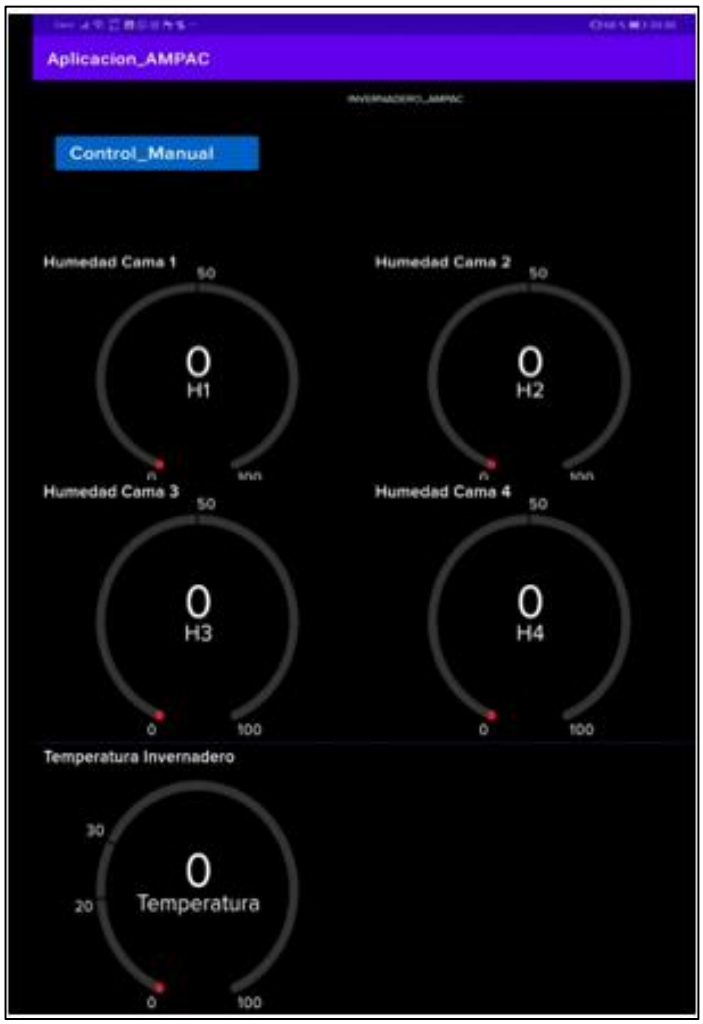
Petición de credenciales

Elaborado por: Wilmer Necpas, Jaime Quishpe

En la Figura 3.17 se observa la segunda pantalla que fue desarrollada con el complemento WebView, la cual nos permite visualizar la página web de Adafruit IO en nuestra aplicación, en esta pantalla se solicita el correo electrónico o el usuario con el que la persona se registró en la plataforma Adafruit IO y también solicita la contraseña.

Una vez se ingresen las credenciales correctas la aplicación redirige al panel de visualización de sensores, si el usuario utiliza una cuenta que no es la misma con la que desarrolló los paneles en Adafruit IO, se dirigirá sin problema al panel de visualización de sensores y podrá observar los valores de los sensores.

Figura 3.18. Panel de Visualización de sensores

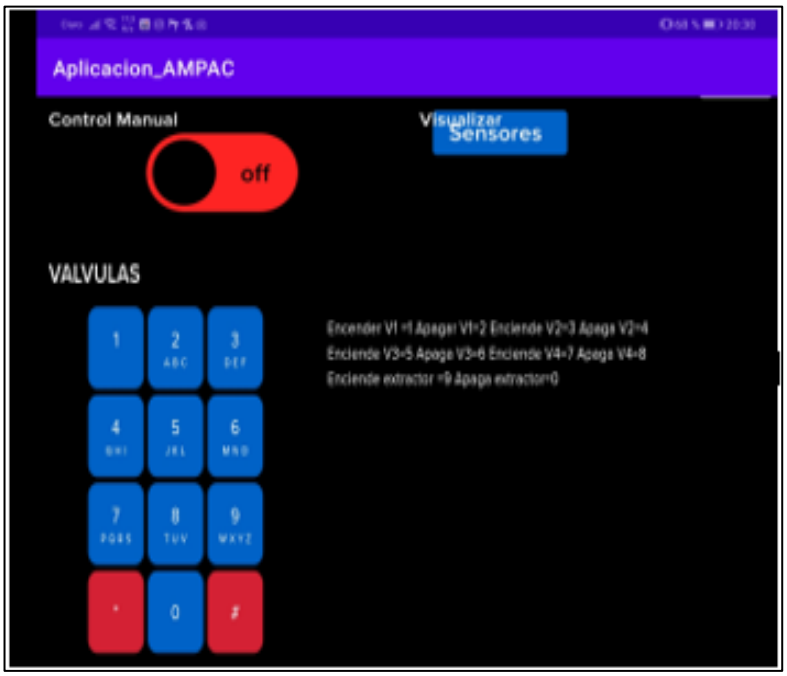


Pantalla 2 panel de visualización de sensores de humedad y temperatura.

Elaborado por: Wilmer Necpas, Jaime Quishpe.

En la Figura 3.18, se observa la pantalla de visualización de sensores, consta de cinco bloques de visualización denominados “GAUGE”, los cuales asimilan un tacómetro de 0 a 100, existen límites en los cuales cambiaran de color dependiendo la sección donde se encuentre el valor de los sensores, cada bloque cuenta con etiquetas para identificar cada uno de los sensores, además en esta pantalla se encuentre el botón “Control_Manual” el cual al ser presionado nos dirige a la última pantalla.

Figura 3.19. Pantalla 3 de la aplicación móvil



Pantalla 3, panel de selección de control manual o control automático.

Elaborado por: Wilmer Necpas, Jaime quishpe

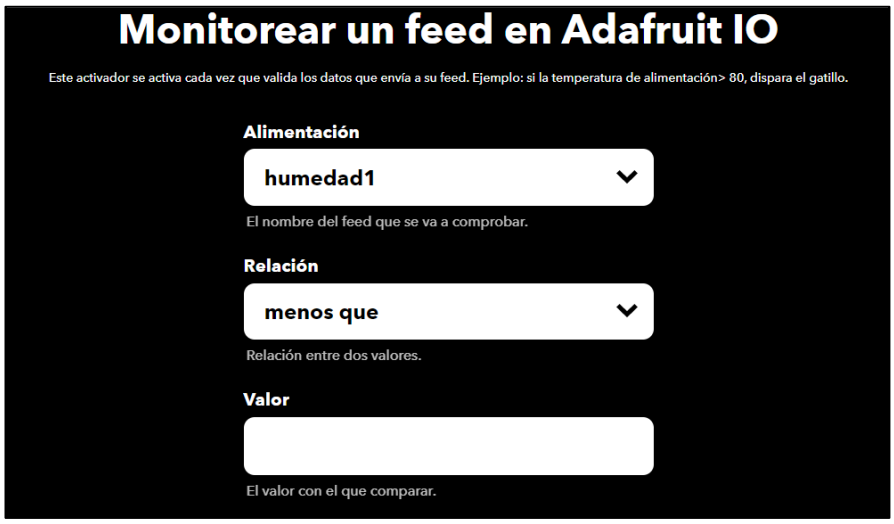
En la Figura 3.19, se muestra el panel de control manual o control automático, el cual consta de un switch el cual elije el modo de funcionamiento, si el switch está en posición “off” el dispositivo está en control automático y si el switch está en posición “on” el dispositivo está en modo de control manual, es decir que se pueden controlar el estado de los actuadores por medio del teclado numérico, esta función permitirá realizar mantenimientos al momento de que existiera fallo en alguno de los sensores, además cuenta con el botón para poder volver al panel de visualización.

Si el usuario ingresa con una cuenta que no es la misma con la que creó el panel en el servidor Adafruit IO por seguridad no se le permitirá cambiar los valores de los FEED con los cuales se controlan los actuadores.

3.8 Uso de la plataforma IFTTT

La plataforma de IFTTT nos permite conectar al servidor de Adafruit IO con el servicio de Telegram, el primer paso es crear un activador para que monitoree los valores del feed seleccionado, este se activa cada vez que se valide la condición, para establecer la condición existen 3 parámetros; primero se elige el feed del cual se quiere obtener el dato, segundo se establece la relación, las relaciones que se pueden establecer son: menor que, menor igual que, mayor que, mayor igual que, igual que y diferente de, por último se establece el valor fijo para comparar, como se puede observar en la Figura 3.20.

Figura 3.20. Creación del activador en IFTTT



Parámetros para definir el activador.

Elaborador por: Wilmer Necpas, Jaime Quishpe.

Una vez definido el activador, se realiza la conexión con el servicio de Telegram. IFTTT envía un mensaje a Telegram pidiendo permiso para que pueda enviar y recibir mensajes, fotos, audios, archivos.

Una vez concedido los permisos, ya se puede definir la acción que se realizará al momento de validar el activador, se define el chat al cual se va a enviar el mensaje y el mensaje que va a enviar IFTTT a Telegram como se observa en la Figura 3.21.

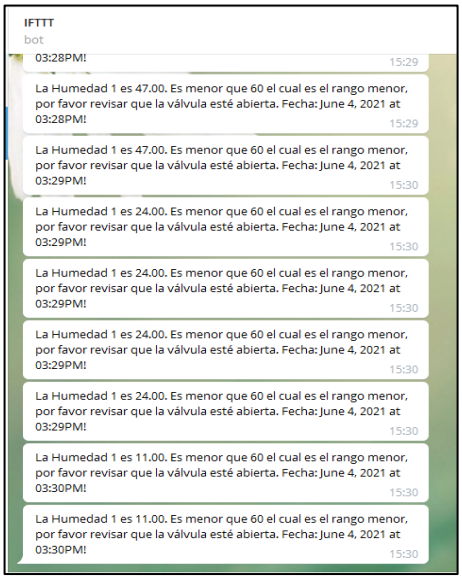
Figura 3.21. Establecer el chat y el mensaje que será enviado a Telegram

Mensaje que será enviado al servicio de Telegram.

Elaborado por: Wilmer Necpas, Jaime Quishpe

En la Figura 3.22, se pueden observar los mensajes de advertencia que se envían desde la plataforma IFTTT hacia el servicio de Telegram, dicho mensaje está compuesto por el valor que lee y el valor del activador y una sugerencia para el usuario, este mensaje se repetirá para cada una de las condiciones establecidas.

Figura 3.22. Mensajes de advertencia



Mensajes de advertencia recibidos por Telegram.

Elaborado por: Wilmer Necpas, Jaime Quishpe

CAPÍTULO 4
ANÁLISIS DE RESULTADOS

En este capítulo se detalla los resultados de las pruebas que miden la latencia, retardos y pérdidas de paquetes para corroborar el correcto funcionamiento del dispositivo IoT.

Las pruebas de rendimiento en distintos momentos de tiempo son unas de las herramientas más efectivas para controlar y solucionar problemas en la red, por tal motivo las pruebas de latencia, retardos y pérdida de paquetes fueron realizadas en horas del día en las cuales existen más uso de internet. (CCNA, 2018)

4.1 Pruebas de latencia

La latencia es la cantidad de tiempo que tarda un paquete en transmitirse a través de la red y de esto dependerá el tiempo que tarda en enviar y recibir los datos el dispositivo IoT desde el servidor web. (Garcia,2016)

Para medir el tiempo que tarda en comunicarse al servidor web mediante la red se utilizó el siguiente comando: ping ***dirección-ip***.

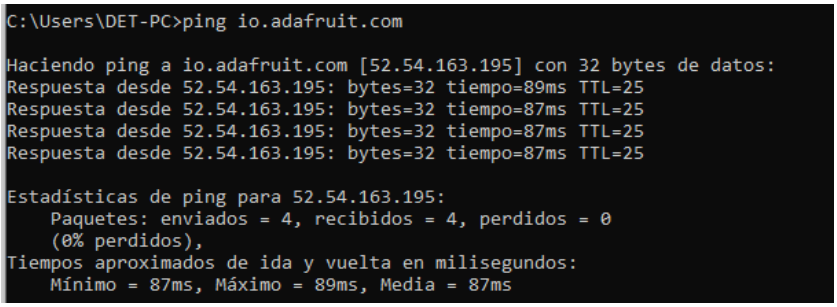
En la Figura 4.1 se observa el resultado de la primera prueba realizada a las 10:00 am, en donde se enviaron cuatro solicitudes de las cuales todas fueron recibidas, además los resultados de latencia prueba fueron: Tmin= 86ms, Tmax= 90ms, Tmedia= 88ms.

Figura 4.1. Prueba de latencia realizada a las 10:00 am

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la prueba realizada a las 12:00 pm, los resultados obtenidos son similares a la prueba realizada anteriormente, los resultados de latencia fueron: Tmin= 87ms, Tmax= 89ms, Tmedia= 87ms como se observa en la Figura 4.2.

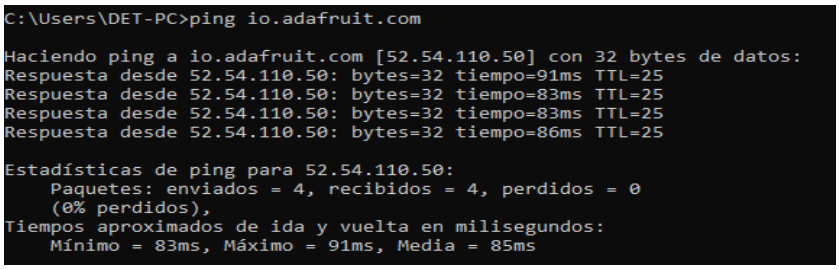
Figura 4.2. Prueba de latencia realizada a las 12:00 pm.



Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.3, se observan los resultados de la prueba realizada a las 14:00 pm, en donde los tiempos fueron los siguientes: Tmin= 83ms, Tmax= 91ms, Tmedia= 85 ms, cabe recalcar que todos los paquetes fueron recibidos.

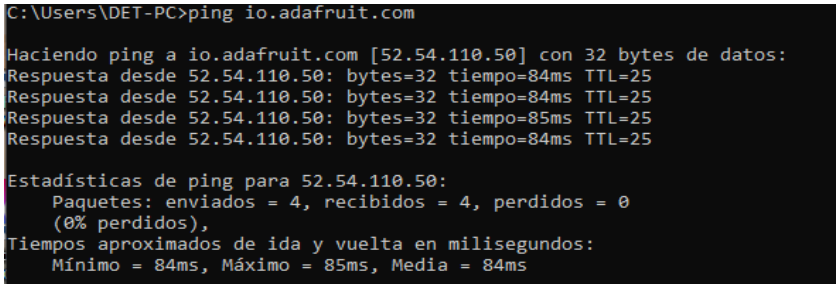
Figura 4.3. Prueba de latencia realizada a las 14:00 pm.



Elaborador por: Wilmer Necpas, Jaime Quishpe

La latencia en la prueba realizada a las 17:00 pm fueron las siguientes: Tmin= 84 ms, Tmax= 85 y Tmedia= 84 ms como se puede observar en la Figura 4.4.

Figura 4.4. Prueba de latencia realizada a las 17:00 pm.



Elaborador por: Wilmer Necpas, Jaime Quishpe

La última prueba fue realizada a las 19:00 pm, en la cual los tiempos de latencia fueron: Tmin= 83ms, Tmax= 84ms, Tmedia= 83ms como se observa en la Figura 4.5.

Figura 4.5. Prueba de latencia realizada a las 19:00 pm.

Elaborador por: Wilmer Necpas, Jaime Quishpe

4.2 Retardos

La información se transmite a través de una red entre dos puntos, esta pasa a través de distintos nodos, los retardos son los tiempos que demora la información para pasar entre nodos. (Baquero,2016)

Para medir estos retardos se utilizó el software WinMTR, el cual nos entrega los tiempos que se demoran los datos en pasar por cada uno de los nodos para llegar a su destino final.

Tabla 4.1. Prueba de retardos realizada a las 10:00 am

HOST	MÍNIMO	MEDIO	MÁXIMO
192.168.2.254	1	9	863
192.168.1.1	1	2	25
192.168.0.1	1	2	28
172.16.16.1	2	3	36
host-181-188-216-65.nedetel.net	2	3	9
172.16.154.25	2	12	848
172.18.15.5	3	12	848
172.18.8.8	3	13	848
172.18.2.16	4	13	863
cloudflare-uio.nap.ec	3	4	39
104.20.39.240	4	13	847

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Tabla 4.1, se observan los resultados que entregó el software WinMTR en la primera prueba realizada a las 10:00 am, en la primera columna se encuentra la dirección ip del host, en la segunda, tercera y cuarta columna nos muestra el mejor

tiempo, el tiempo promedio y el peor tiempo de comunicación. Al finalizar la prueba encontramos el tiempo promedio que demora en enviarse la información desde el nodo inicial al nodo final el cual es de 89ms.

Tabla 4.2. Prueba de retardos realizada a las 12:00 am

HOST	MÍNIMO	MEDIO	MÁXIMO
192.168.2.254	1	3	57
192.168.1.1	1	2	5
192.168.0.1	1	4	151
172.16.16.1	2	6	152
host-181-188-216-65.nedetel.net	3	7	155
172.16.154.25	3	6	158
172.18.15.5	3	13	22
172.18.8.8	4	12	112
172.18.2.16	4	9	126
cloudflare-uio.nap.ec	3	12	147
104.20.39.240	3	15	156

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Tabla 4.2, se pueden observar los retardos que existen desde el nodo inicial hasta el nodo final en la prueba realizada a las 12:00 am, el tiempo promedio de la suma de los retardos es de 89ms, siendo este un tiempo aceptable para la comunicación.

Tabla 4.3. Prueba de retardos realizada a las 14:00 pm

HOST	MÍNIMO	MEDIO	MÁXIMO
192.168.2.254	1	4	126
192.168.1.1	1	3	125
192.168.0.1	1	3	126
172.16.16.1	2	4	80
host-181-188-216-65.nedetel.net	2	4	81
172.16.154.25	2	9	140
172.18.15.5	2	14	81
172.18.8.8	3	10	82
172.18.2.16	4	10	82
cloudflare-uio.nap.ec	3	9	38
104.20.39.240	3	13	82

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Tabla 4.3 se observan los retardos de la prueba realizada a las 14:00 pm, si sumamos los tiempos promedio de retardos obtenemos 83ms que será el tiempo en que se demora en enviar la información.

Tabla 4.4. Prueba de retardos realizada a las 17:00 pm

HOST	MÍNIMO	MEDIO	MÁXIMO
192.168.2.254	1	4	126
192.168.1.1	1	9	125
192.168.0.1	1	2	126
172.16.16.1	2	8	80
host-181-188-216-65.nedetel.net	2	7	81
172.16.154.25	2	5	140
172.18.15.5	2	5	81
172.18.8.8	3	9	82
172.18.2.16	4	9	82
cloudflare-uio.nap.ec	3	10	38
104.20.39.240	3	14	82

Elaborador por: Wilmer Necpas, Jaime Quishpe

La prueba realizada a las 17:00 pm, nos entrega los resultados que podemos observar en la Tabla 4.4, la suma de los retardos promedios en esta prueba es de 80ms.

Tabla 4.5. Prueba de retardos realizada a las 19:00 pm

HOST	MÍNIMO	MEDIO	MÁXIMO
192.168.2.254	1	3	126
192.168.1.1	1	8	125
192.168.0.1	1	4	126
172.16.16.1	2	9	80
host-181-188-216-65.nedetel.net	2	6	81
172.16.154.25	2	6	140
172.18.15.5	2	9	81
172.18.8.8	3	6	82
172.18.2.16	4	11	82
cloudflare-uio.nap.ec	3	9	38
104.20.39.240	3	16	82

Elaborador por: Wilmer Necpas, Jaime Quishpe

La última prueba fue realizada a las 19:00 pm, los resultados pueden ser observados en la tabla 4.5, la suma de los tiempos de retardo promedio en esta prueba es de 81ms.

4.3 Pérdida de paquetes

Cuando la información no llega completa o simplemente no llega al nodo final se denomina pérdida de paquetes, una de las razones por las cuales pueden existir pérdida de paquetes es la desconexión de la red, también colisiones entre datos, entre otras razones. (Baquero,2016)

Para esta prueba se implementó el código que se puede observar en la Figura 4.6, que envía un dato String “A” a un feed del servidor web 1000 veces, de esta manera se

revisará si el dato llega las 1000 veces que se enviaron, además de revisar que llegue el mismo carácter que se envió.

Figura 4.6. Código en Arduino para la prueba de pérdida de paquetes

```
env = ("AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA");
mqtt_client.publish(String("jaime4411/f/prueba").c_str(),String(env).c_str());
```

Código Arduino.

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.7, se puede observar el feed de Adafruit en donde se recibieron todos los caracteres enviados en la prueba realizada a las 10:18 am.

Figura 4.7. Caracteres recibidos en el servidor web a las 10:18 am

Información recibida en el servidor web.

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.8, se puede observar que se recibieron todos los caracteres enviados a las 12:24 am.

Figura 4.8. Caracteres recibidos en el servidor web a las 12:57 am

Información recibida en el servidor web.

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la prueba realizada a las 14: 21 pm se recibieron todos los caracteres enviados desde el dispositivo al servidor web como se observa en la Figura 4.9.

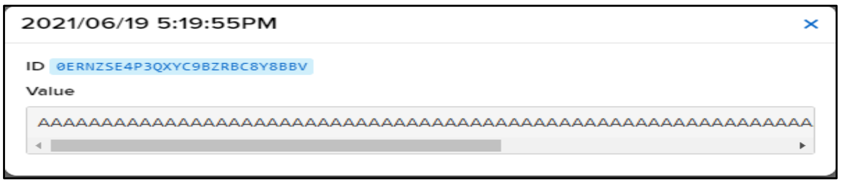
Figura 4.9. Caracteres recibidos en el servidor web a las 14:27 pm

Información recibida en el servidor web.

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.10, se observa todos los caracteres enviados a las 17:19 pm desde el dispositivo IoT hacia el servidor web.

Figura 4.10. Caracteres recibidos en el servidor web a las 17:19 pm

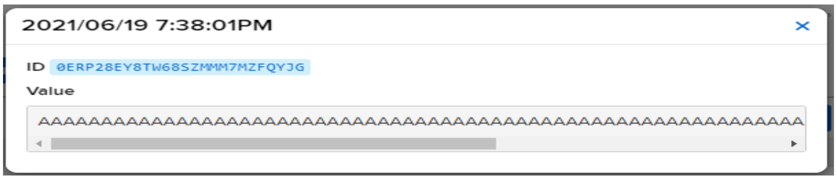


Información recibida en el servidor web.

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la última prueba realizada a las 19:38 pm también se recibieron todos los caracteres enviados desde el dispositivo IoT como se observa en la Figura 4.11.

Figura 4.11. Caracteres recibidos en el servidor web a las 19:38 pm



Información recibida en el servidor web.

Elaborador por: Wilmer Necpas, Jaime Quishpe

También se realizó la prueba de pérdida de paquetes enviando datos desde el servidor web y recibiendolos en el dispositivo IoT, para ello se ocupo el codigo que se muestra en la Figura 4.12, que nos permite suscribirnos a un feed del servidor e imprimirlo en el puerto serial.

Figura 4.12. Código para recibir el dato en el dispositivo IoT

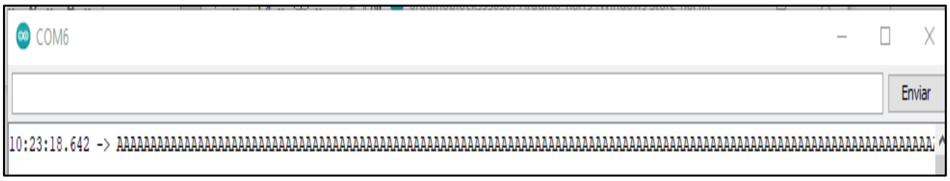
```
mqtt_client.subscribe(String("aio_OcIw34wYPIj40xmI5hzP5pYCqt7o").c_str());
 Serial.println(String(s_letra));
```

Código Arduino.

Elaborador por: Wilmer Necpas, Jaime Quishpe

Como resultado se recibieron todos los caracteres que se enviaron desde el servidor web al dispositivo IoT como se observa en la Figura 4.13, esta prueba confirmará que no existen pérdidas de paquetes.

Figura 4.13. Resultados del envió de datos hacia el dispositivo IoT a las 10:23 am

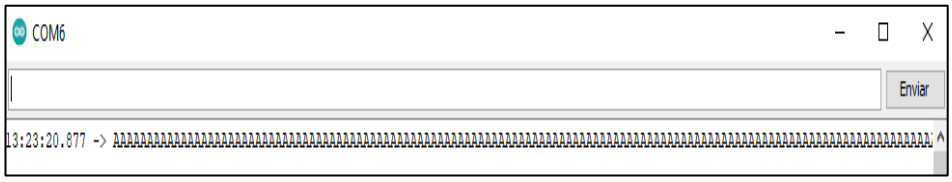


Capturas del puerto serie del IDE de Arduino .

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.14 se observa el resultado de la prueba realizada a las 13:23 pm se recibieron todos los caracteres que se enviaron desde el servidor web al dispositivo IoT.

Figura 4.14. Resultados del envió de datos hacia el dispositivo IoT a las 13:23 pm

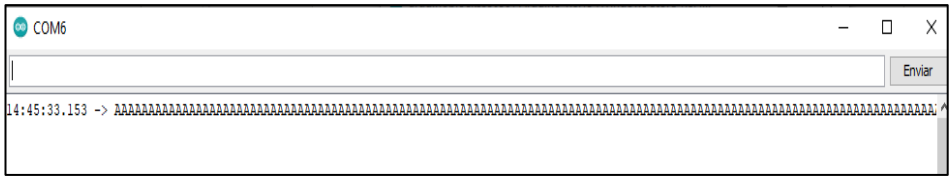


Capturas del puerto serie del IDE de Arduino .

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.15 se observa el resultado de la prueba realizada a las 14:45 pm se recibieron todos los caracteres que se enviaron desde el servidor web al dispositivo IoT, en esta prueba tampoco hubo pérdida de información.

Figura 4.15. Resultados del envió de datos hacia el dispositivo IoT a las 14:45 pm

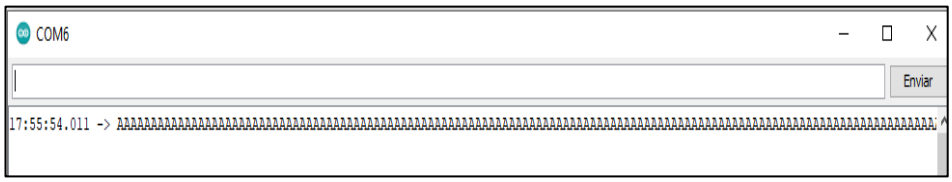


Capturas del puerto serie del IDE de Arduino .

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.16 se observa el resultado de la prueba realizada a las 17:55 pm se recibieron todos los caracteres que se enviaron desde el servidor web al dispositivo IoT.

Figura 4.16. Resultados del envió de datos hacia el dispositivo IoT a las 17:55 pm

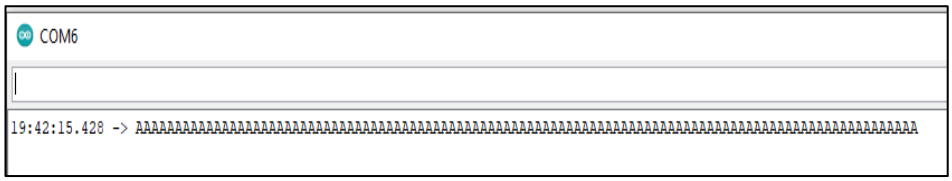


Capturas del puerto serie del IDE de Arduino .

Elaborador por: Wilmer Necpas, Jaime Quishpe

En la Figura 4.17 se observa el resultado de la última prueba realizada a las 19:42 pm donde también se recibieron todos los caracteres que se enviaron desde el servidor web al dispositivo IoT.

Figura 4.17. Resultados del envió de datos hacia el dispositivo IoT a las 19:42 pm



Capturas del puerto serie del IDE de Arduino .

Elaborador por: Wilmer Necpas, Jaime Quishpe

4.4 Pruebas de tiempo de respuesta

En la tabla 4.6 se presentan los resultados de la prueba de tiempos de respuesta que consiste en enviar un dato desde el servidor web al dispositivo IoT y medir la diferencia entre el tiempo de envío y recepción del dato.

Tabla 4.6. Prueba de tiempo de respuesta

Número de prueba	Hora de envío	Hora de recepción	Diferencia (segundos)
1	10:22:15.874	10:22:15.899	0.25s
2	12:15:21.059	12:15:21.569	0.51s
3	14:35:45.136	14:35:46.181	1.045s
4	17:40:13.523	17:40:14.423	0.90s
5	19:27:28.635	19:27:29.395	0,76s
		Promedio	0,7s

Elaborador por: Wilmer Necpas, Jaime Quishpe

En esta prueba se envió el dato para activar la electroválvula de la cama número uno y se tomaron la hora de envío y recepción del dato para luego calcular la diferencia de los dos tiempos y así hallar el tiempo de respuesta.

4.5 Monitoreo de la humedad relativa del suelo y temperatura

Finalmente se tomaron medidas de los sensores para corroborrar el funcionamiento del dispositivo IoT y el rango de humedad del suelo y temperatura ambiente.

Como se observa en la Tabla 4.7, las medidas tomadas de humedad relativa de suelo no tienen una variación notable entre días ya que el suelo que posee el invernadero AMPAC es humífero esto conlleva a que retenga el agua y sea muy difícil que su humedad cambie muy rápido en el tiempo, la temperatura más alta registrada por el dispositivo IoT fue de 28,7 ºC.

Tabla 4.7 Valores recolectados de los sensores de humedad de suelo y temperatura ambiente.

Fecha de monitoreo	Humedad cama 1	Humedad cama 2	Humedad cama 3	Humedad cama 4	Temperatura ambiente
15/06/2021	67%	66%	65%	73%	27%
16/06/2021	74%	69%	62%	75%	30,25%
17/06/2021	74%	68%	64%	74%	28,72%
18/06/2021	68%	66%	58%	72%	24,80%
19/06/2021	73%	68%	63%	76%	27,60%
20/06/2021	72%	67%	60%	75%	24,80%
21/06/2021	71%	70%	61%	74%	25%
22/06/2021	74%	72%	59%	75%	28,40%
23/06/2021	75%	69%	58%	73%	22,47%
24/06/2021	73%	70%	58%	72%	18%

Elaborador por: Wilmer Necpas, Jaime Quishpe

4.6 Encuesta

Para conocer la aceptación de la aplicación IoT desarrollada en beneficio de la organización AMPAC, se realizó una encuesta a 15 personas del grupo AMPAC, la encuesta cuenta con 5 preguntas como se muestra en el Anexo 4 en la cual se mide la calificación de la implementación con un 72% en la opción excelente, si la implementación ayudará a reducir pérdidas de producción con un 60% favorable, la facilidad en el manejo de la aplicación con 64 % favorable, el impacto que tiene el dispositivo en la Asociación con un 100% en positivo y finalmente si facilitará las labores al personal encargado del invernadero con un 73%.

CAPÍTULO 5
CONCLUSIONES

Se analizaron los requerimientos del invernadero AMPAC y se determinó el uso de sensores FC-28 para la medición de la humedad relativa del suelo en cada una de las camas, además de un sensor DHT11 que se encarga de medir la temperatura ambiente del invernadero, los sensores se conectan al NodeMCU ESP32 el cual es encargado de enviar la información mediante conexión WIFI al servidor web de Adafruit IO.

Se implementó un dispositivo IoT el cual nos permite realizar la adquisición de datos que influyen directamente en la producción como: la humedad relativa del suelo que tiene que permanecer un rango de 50% a 80% y la temperatura ambiente del invernadero que debe mantenerse en un intervalo de 15 ºC a 30ºC, estos datos se pueden monitorear de forma remota de manera que la persona acargo pueda tomar decisiones en base a la informacion brindad por el dispositvo IoT.

Se desarrolló una aplicación con la ayuda del Software Android Studio, la cual cuenta con tres pantallas, la primera es la pantalla de acceso, la segunda es una pantalla donde podemos observar los datos de temperatura ambiente y humedad relativa del suelo de las camas del invernadero y la última pantalla que nos permite tener un control manual de los actuadores del invernadero como; electroválvulas, focos y ventilador.

Se realizaron las pruebas de latencia, retardos y pérdidas de paquetes en distintas horas del día, el tiempo máximo de latencia obtenido fue de 88ms en la prueba realizada a las 10:00 am, también se comprobó que no existe pérdida de información al comunicarse desde el dispositivo IoT al servidor web y viceversa, además se obtuvo un tiempo de respuesta promedio de 0,7 segundos a la activación de actuadores, concluyendo así que el funcionamiento del dispositivo IoT es óptimo.

RECOMENDACIONES

Para posibles mejoras en el dispositivo IoT se puede incorporar sensores que permitan la adquisición de datos adicionales a los utilizados en este proyecto técnico, estos datos pueden ser: pH del suelo, cantidad de dióxido de carbono (CO2), que son variables que también influyen en el correcto desarrollo del cultivo, además se puede incluir sensores de caudal para el monitoreo del consumo de agua.

Es recomendable incluir un módulo arduino o una raspberry al NodeMCU ESP32 para aumentar las entradas analogicas, ya que si se está utilizando la comunicación WIFI el módulo ESP32 solo permite el uso de seis entradas analogicas y esto seria una problemática al momento de replicar el dispositivo para invernaderos de mayor dimensión y con mayor demanda de sensores.

BIBLIOGRAFÍA

Babiuch, M., Foltynek, P., & Smutny, P. (2019). Using the ESP32 microcontroller for data processing. *Proceedings of the 2019 20th International Carpathian Control Conference, ICCC 2019*.: https://doi.org/10.1109/CarpathianCC.2019.8765944´

Baquero, P. (2016). Pérdidas de datos · fundamentos-de-redes. Retrieved June 17, 2021, Recuperado el 17 de Junio del 2021, de Gitbooks.io: https://pedrobq.gitbooks.io/fundamentos-de-redes/content/perdidas_de_datos.html

Baquero, P. (2016). Retardos: su origen y medidas para su corrección · fundamentos-de-redes. Recuperado el 17 de Junio del 2021, de Gitbooks.io: https://pedrobq.gitbooks.io/fundamentos-de-redes/content/retardos_su_origen_y_medidas_para_su_correccion/

Barrera, G. (2018). Estilo arquitectónico para aplicaciones IoT. *Universidad Del Centro de Estudios Macroeconómicos de Argentina (UCEMA)*, 664. Obtenido de: http://hdl.handle.net/10419/203805

Beltrán, J. (2013). *Producción agropecuaria y desarrollo local en los cantones Cayambe y Pedro Moncayo* (1ra. Edici). Quito-Euador: Editorial Universitaria ABYA-YALA.

CNEL EP. (13 de Febrero del 2019). CNEL EP expone tarifa residencial y tips de consumo eléctrico - CNEL EP. Recuperado el 2 de Junio, de CNEL EP: https://www.cnelep.gob.ec/2019/02/cnel-ep-expone-tarifa-residencial-y-tips-de-consumo-electrico/

Comandos para medir el Rendimiento de Red - CCNA desde Cero. (10 de Febrero del 2018). Recuperado el 19 de Junio del 2021, de CCNA desde Cero: https://ccnadesdecero.es/comandos-para-medir-rendimiento-red/

Cooper, J. (22 de Enero del 2015). Adafruit IO. Recuperado el 27 de Mayo del 2021, de Adafruit Learning System: https://learn.adafruit.com/adafruit-io

EINSTRONIC (2017). *Introduction to NodeMCU ESP32 DevKIT*. Obtenido de: http://www.hit-karlsruhe.de/hit-info/info-ws18/CM-IoT_Fuell-30/data/NodeMCU-32S-Catalogue_Entwicklerboard_PINS.pdf

Escudero, M. (10 de Marzo del 2017). « If This Then That »: Haz que Internet trabaje para ti. Recuperado el 28 de Mayo del 2021, de donosTik: https://www.donostik.com/ifttt-if-this-then-that/

García, A. (18 de Agosto del 2016). Qué es la latencia, y cómo podemos mejorarla. Recuperado el 17 de Junio del 2021, de Testdevelocidad.es: https://www.testdevelocidad.es/2016/08/18/la-latencia-podemos-mejorarla/

Garita, R. (2013). Tecnología Móvil: desarrollo de sistemas y aplicaciones para las Unidades de Información. *E-Ciencias de La Información*, *3*(2), 1–14. Obtenido de: https://revistas.ucr.ac.cr/index.php/eciencias

IFTTT. (2021). Bienvenido a la guía IFTTT. Recuperado el 27 de Mayo del 2021, de

IFTTT : https://ifttt.com/explore/welcome_to_ifttt

Imamura, Y., Orito, R., Uekawa, H., Chaikaew, K., Leelaprute, P., Sato, M., & Yamauchi, T. (2021). Web access monitoring mechanism via Android WebView for threat analysis. *International Journal of Information Security*. https://doi.org/10.1007/s10207-020-00534-3

Llamas, L. (19 de Enero del 2016). Medir la humedad del suelo con Arduino e Higrómetro. Recuperado el 27 de Mayo del 2021, de: https://www.luisllamas.es/arduino-humedad-suelo-fc-28/

Maier, A., Sharp, A., & Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. *2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference*, 143–148. https://doi.org/10.1109/ITECHA.2017.8101926

Mendieta, T. P., Herrera, J., & Peña, A. J. (2019). La Capacidad del IOT de Transformar el Futuro. *Revista Avenir*, *1*(1), 15–18. Obtenido de: https://fundacionavenir.net/revista/index.php/avenir/article/view/79

Mi, X., Zhang, Y., Qian, F., & Wang, X. (2017). An empirical characterization of IFTTT: Ecosystem, usage, and performance. https://doi.org/10.1145/3131365.3131369

Mole, P. V. (2020). *Progressive Web Apps: A Novel Way for Cross-Platform Development*. Obtenido de: https://www.researchgate.net/publication/344170769.

Rai, P., & Rehman, M. (2019). ESP32 based smart surveillance system. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies, ICoMET 2019*, 2019–2021. https://doi.org/10.1109/ICOMET.2019.8673463

Rubell, B. (29 de Septiembre del 2020). Adafruit IO Basics: Schedule Triggers. Recuperado el 27 de Mayo del 2021, de Adafruit Learning System: https://learn.adafruit.com/adafruit-io-basics-scheduled-triggers

Rubell, B. (13 de Junio del 2018). Welcome to Adafruit IO. Recuperado el 18 de Mayo del 2021, de Adafruit Learning System: https://learn.adafruit.com/welcome-to-adafruit-io

Salazar, J., & Silvestre, S. (2017). Internet de las cosas (IoT) - Cisco. Obtenido de: https://core.ac.uk/download/pdf/81581111.pdf

Swetha, S., Sn, I., Nevetha, R., Sarathy, S., & Deepa, R. (2019). *Automatic Room Temperature and Monitoring System Using Arduino*. 4647–4651. https://doi.org/10.15680/IJIRSET.2019.0804128

Systems Espressif. (2021). ESP32 Series Datasheet. Obtenido de: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Treece, T. (20 de Abril del 2015). Adafruit IO Basics : Feeds. Recuperado el 23 de Mayo del 2021, de Adafruit Learning System: https://learn.adafruit.com/adafruit-io-basics-feeds

Treece, T. (21 de abril del 2015). Adafruit IO Basics : Dashboards. Recuperado el 20 de Mayo del 2021, de Adafruit Learning System: https://learn.adafruit.com/adafruit-io-basics-dashboards

Zhou, Y., Zhou, Q., Kong, Q., & Cai, W. (2012). Wireless temperature & humidity monitor and control system. *2012 2nd International Conference on Consumer Electronics, Communications and Networks, CECNet 2012 - Proceedings*, 2246–2250. https://doi.org/10.1109/CECNet.2012.6201725

ANEXOS

Anexo 1. Código de Programación del módulo ESP32

Anexo 1.1 Código para incluir librerías e inicializar variables.

```
#include "WiFi.h"
#include "PubSubClient.h"
#include "DHT.h"
#define DHTPIN 33
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
double h1;
double h2;
double h3;
double h4;
double h5;
double grados;
String s_auto="off";
String s_Accion;
```

Anexo 1.2 Código para definir credenciales para la conexión WIFI y la conexión con el servidor

```
const char mqtt_wifi_ssid[]="Invernadero";
const char mqtt_wifi_pass[]="123456789";
const char mqtt_broker[]="io.adafruit.com";
const int mqtt_port=1883;
const char mqtt_user[]="jaime4411";
const char mqtt_pass[]="aio_OcIw34wYPIj40xmI5hzP5pYCqt7o";
const char mqtt_clientid[]="jaime4411";
```

Anexo 1.3 Código para la conexión WIFI y conexión con el servidor

```
  WiFi.begin(mqtt_wifi_ssid,mqtt_wifi_pass);
  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.println("Connecting to WiFi...");
  }
  mqtt_client.setServer(mqtt_broker, mqtt_port);

}

void mqtt_loop(){
  if (!mqtt_client.connected()) {
    mqtt_client.connect(mqtt_clientid,mqtt_user,mqtt_pass);
    mqtt_subscribe();
  }
  if (mqtt_client.connected()) {
    mqtt_client.loop();
  }
}
```

Anexo 1.4 Código para definir el modo de uso de los pines del módulo e inicializar el sensor DHT 11, la comunicación serial

```
  Serial.begin(9600);
  dht.begin();

pinMode(23, OUTPUT);
pinMode(22, OUTPUT);
pinMode(21, OUTPUT);
pinMode(19, OUTPUT);
pinMode(18, OUTPUT);
pinMode(5, OUTPUT);

pinMode(36, INPUT);
pinMode(39, INPUT);
pinMode(34, INPUT);
pinMode(35, INPUT);
pinMode(33, INPUT);
pinMode(32, INPUT);
pinMode(4, INPUT);
```

Anexo 1.5 Código para la lectura del sensor de temperatura

```
void Sensor_Temp_hum() {
  float hh = dht.readHumidity(); //Leemos la Humedad
       grados = dht.readTemperature(); //Leemos la temperatura en grados Celsius
      float ff = dht.readTemperature(true); //Leemos la temperatura en grados Fahrenheit
      //--------Enviamos las lecturas por el puerto serial------------
}
```

Anexo 1.6 Código para la lectura de los sensores de humedad

```
void humedades() {
  h1 = map(analogRead(36),0,4096,100,0); // CAMA 1
  Serial.println(h1);
  h2 = map(analogRead(39),0,4096,100,0);// CAMA 2
  Serial.println(h2);
  h3 = map(analogRead(34),0,4096,100,0);// CAMA 3
  Serial.println(h3);
  h4 = map(analogRead(35),0,4096,100,0);// CAMA 4
  Serial.println(h4);
  delay (3000);
}
```

Anexo 1.7 Código para el control automático de actuadores

```
void Control() {
  if ((h1 < 50)) {
    digitalWrite(23, HIGH);
  }
  if ((h1 > 50)) {
   digitalWrite(23, LOW);
  }
  if ((h2 < 50)) {
    digitalWrite(22, HIGH);
  }
  if ((h2 > 50)) {
    digitalWrite(22, LOW);
  }
  if ((h3 < 50)) {
    digitalWrite(21, HIGH);
  }
  if ((h3 > 50)) {
    digitalWrite(21, LOW);
  }
  if ((h4 < 50)) {
    digitalWrite(19, HIGH);
  }
  if ((h4 > 50)) {
    digitalWrite(19, LOW);
  }
  if ((grados > 40)) {
    digitalWrite(18, HIGH);
  }
  if ((grados < 40)) {
    digitalWrite(18, LOW);
  }
}
```

Anexo 1.8 Código para la suscripción a los feeds de Adafruit IO

```
void mqtt_subscribe(){
  mqtt_client.subscribe(String("jaime4411/f/control").c_str());
  mqtt_client.subscribe(String("jaime4411/f/automatico").c_str());
}
```

Anexo 1.9 Código para la publicación en los feeds de Adafruit IO

```
mqtt_client.publish(String("jaime4411/f/humedad1").c_str(),String(h1).c_str());
mqtt_client.publish(String("jaime4411/f/humedad2").c_str(),String(h2).c_str());
mqtt_client.publish(String("jaime4411/f/humedad3").c_str(),String(h3).c_str());
mqtt_client.publish(String("jaime4411/f/humedad4").c_str(),String(h4).c_str());
mqtt_client.publish(String("jaime4411/f/humeda5").c_str(),String(h5).c_str());
mqtt_client.publish(String("jaime4411/f/Temperatura").c_str(),String(grados).c_str());
```

Anexo 1.10 Código para el control manual de actuadores

```
    if (String(s_auto).equals(String("on"))) {
      if (String(s_Accion).equals(String("1"))) {
        digitalWrite(23, HIGH);
delay(1000);
      }
      if (String(s_Accion).equals(String("2"))) {
        digitalWrite(23, LOW);
delay(1000);
      }
      if (String(s_Accion).equals(String("3"))) {
        digitalWrite(22, HIGH);
delay(1000);
      }
      if (String(s_Accion).equals(String("4"))) {
        digitalWrite(22, LOW);
delay(1000);
      }
      if (String(s_Accion).equals(String("5"))) {
        digitalWrite(21, HIGH);
delay(1000);
      }
      if (String(s_Accion).equals(String("6"))) {
        digitalWrite(21, LOW);
delay(1000);
      }
      if (String(s_Accion).equals(String("7"))) {
        digitalWrite(19, HIGH);
delay(1000);
      }
      if (String(s_Accion).equals(String("8"))) {
        digitalWrite(19, LOW);
delay(1000);

        }
        if (String(s_Accion).equals(String("0"))) {
          digitalWrite(18, LOW);
  delay(1000);
        }
        if (String(s_Accion).equals(String(" *"))) {
          digitalWrite(5, HIGH);
  delay(1000);
        }
        if (String(s_Accion).equals(String("#"))) {
          digitalWrite(5, LOW);
  delay(1000);
        }
```

ANEXO 2. Código de programación de la aplicación en Android Studio

Anexo 2.1 Código de programación de la pantalla 1

```
package com.example.aplicacion_ampac;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView tvBoton1;
    TextView tvBoton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvBoton1 = (TextView) findViewById(R.id.tvBoton1);
        tvBoton = (TextView) findViewById(R.id.tvBoton);

        tvBoton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            Intent i = new Intent(MainActivity.this, DosActivity.class);
           startActivity(i);
            }
        });

        tvBoton1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
}
}
```

Anexo 2.2 Código de programación con el complemento WebView

```
package com.example.aplicacion_ampac;

import androidx.appcompat.app.AppCompatActivity;

import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;

public class DosActivity extends AppCompatActivity {
    final String url= "https://accounts.adafruit.com/users/sign_in";
    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
        setContentView(R.layout.activity_dos);
        webView = (WebView) findViewById(R.id.inter);
        WebSettings setting= webView.getSettings();
        webView.getSettings().setJavaScriptEnabled(true);
        webView.getSettings().setLoadWithOverviewMode(true);
        webView.getSettings().setUseWideViewPort(true);

        webView.getSettings().setSupportZoom(true);
        webView.getSettings().setBuiltInZoomControls(true);
        webView.getSettings().setDisplayZoomControls(false);

        webView.setScrollBarStyle(WebView.SCROLLBARS_OUTSIDE_OVERLAY);
        webView.setScrollbarFadingEnabled(false);
        setting.setDomStorageEnabled(true);
        setting.setUseWideViewPort(true);
        webView.getSettings().supportZoom();
        setting.setJavaScriptEnabled(true);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl(url);
        webView.setBackgroundColor(0x77000000);
        setDesktopMode(webView, true);

    }
```

```
    public void setDesktopMode(WebView webView,boolean enabled) {
        String newUserAgent = webView.getSettings().getUserAgentString();
        if (enabled) {
            try {
                String ua = webView.getSettings().getUserAgentString();
                String androidOSString = webView.getSettings().getUserAgentString().substring(ua.indexOf("("), ua.indexOf(")") + 1);
                newUserAgent = webView.getSettings().getUserAgentString().replace(androidOSString, "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.4) Gecko/20100101 Firefox/4.0");
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            newUserAgent = null;
        }
        webView.getSettings().setUserAgentString(newUserAgent);
        webView.getSettings().setUseWideViewPort(enabled);
        webView.getSettings().setLoadWithOverviewMode(enabled);
        webView.reload();
    }
    @Override
    public void onBackPressed() {
        if(webView.canGoBack()){
            webView.goBack();
        }else {
            super.onBackPressed();
        }
    }
}
```

Anexo 2.3 Código para incluir botones e imágenes en la pantalla 1

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E4D8E0"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvBoton"
        android:layout_width="188dp"
        android:layout_height="58dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:background="#673AB7"
        android:cursorVisible="true"
        android:text="INGRESAR"
        android:textSize="34sp"
        android:textStyle="bold"
        android:translationX="100sp"
        android:translationY="40sp"
        app:autoSizeTextType="uniform" />

    <TextView
        android:id="@+id/tvBoton1"
        android:layout_width="130dp"
        android:layout_height="62dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:background="#673AB7"
        android:bufferType="normal"
        android:clickable="true"
        android:cursorVisible="true"
        android:text="SALIR"
        android:textSize="34sp"
        android:textStyle="bold"
        android:translationX="130sp"
        android:translationY="80sp"
        app:autoSizeTextType="uniform" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:translationY="100sp"
        app:srcCompat="@mipmap/logo1" />

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:translationY="130sp"
        app:srcCompat="@mipmap/ups" />

</LinearLayout>
```

Anexo 2.4 Código para incluir el complemento WebView en la pantalla 2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".DosActivity">

    <WebView
        android:id="@+id/inter"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp" />
</LinearLayout>
```

ANEXO 3. Instalación del dispositivo IoT

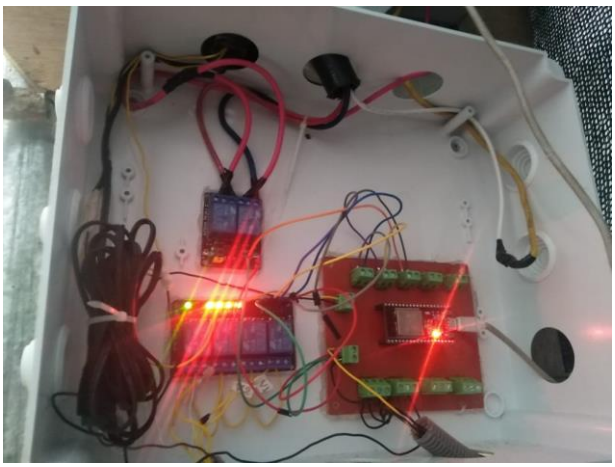
Anexo 3.1 Instalación eléctrica en el invernadero AMPAC

Anexo 3.2 Instalación del sistema de riego en el invernadero AMPAC

Anexo 3.3 Cableado para electroválvulas y sensores

Anexo 3.4 Instalación de focos y ventilador

Anexo 3.5 Conexión de sensores y actuadores al dispositivo IoT

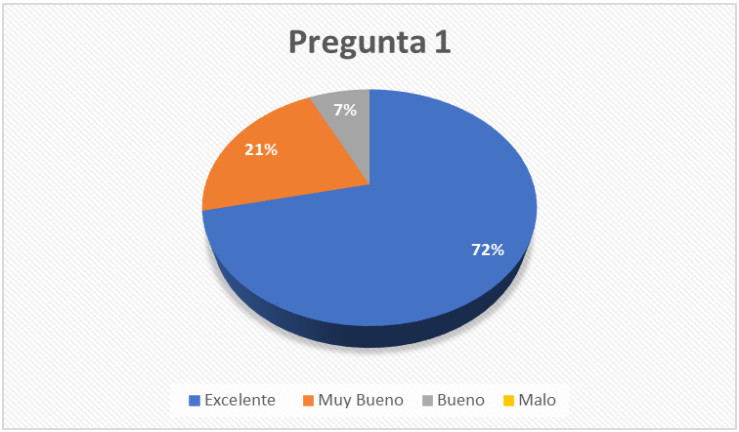


Anexo 3.5 Pruebas de encendido de focos y ventilador

ANEXO 4. Resultados de la encuesta

Anexo 4.1. Pregunta 1

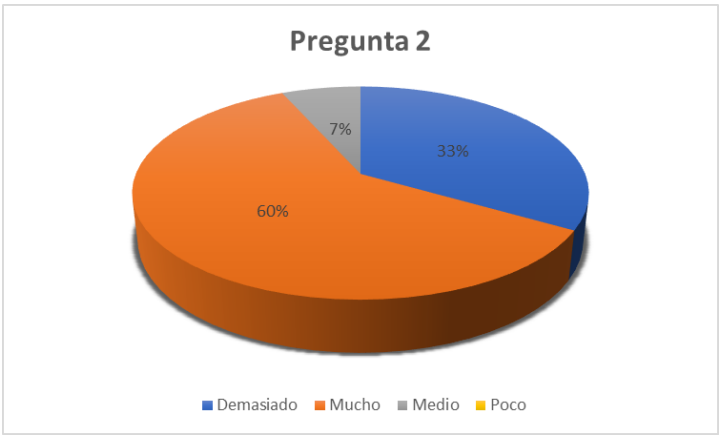
¿Como calificaría la implementación del dispositivo IoT para el monitoreo del invernadero AMPAC?



Excelente	72%
Muy Bueno	21%
Bueno	7%
Malo	0%

Anexo 4.2. Pregunta 2

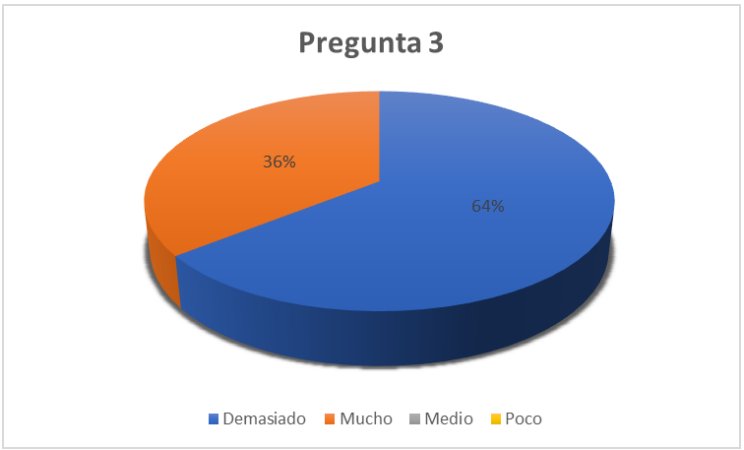
¿Cree usted que el dispositivo IoT ayudara a reducir las pérdidas de producción en el Invernadero AMPAC?



Demasiado	60%
Mucho	33%
Medio	7%
Poco	0%

Anexo 4.3. Pregunta 3

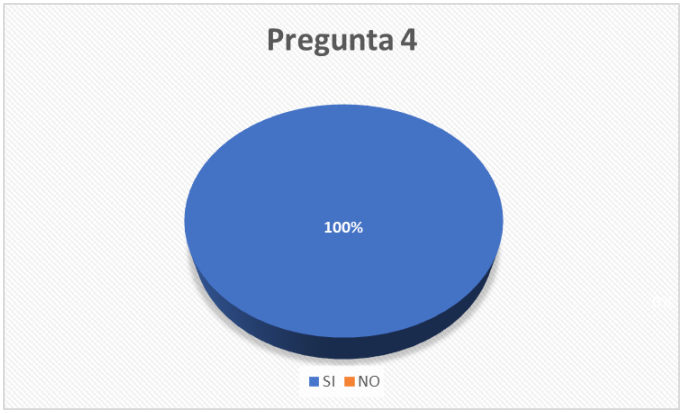
¿Para usted la aplicación que permite la visualización de la humedad de suelo y la temperatura ambiente es fácil de manejar?



Demasiado	64%
Mucho	36%
Medio	0%
Poco	0%

Anexo 4.4. Pregunta 4

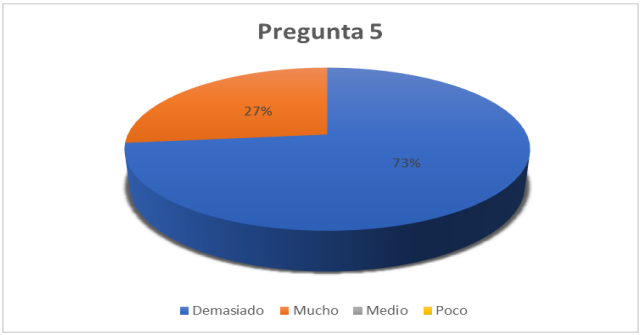
¿Considera usted que este aporte tecnológico tendrá un impacto positivo en el invernadero?



SI	100%
NO	0%

Anexo 4.5. Pregunta 5

¿Piensa usted que el dispositivo IoT facilitará las labores de la persona a cargo del cuidado del invernadero AMPAC?



Demasiado	73%
Mucho	27%
Medio	0%
Poco	0%