

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Industria 4.0

**Sistema Edge Computing para
el manejo de cultivo de Fresa
(Fragaria) bajo Invernadero**

Trabajo Fin de Máster

Presentado por: García Pumagualle, Cristian

Director: Del Campo Jiménez, Guillermo

Resumen

Desarrollo de un sistema Edge Computing para el manejo de cultivo de Fresa (Fragaria) bajo invernadero, que permite al agricultor optimizar recursos y reducir costos por medio de la automatización del monitoreo de las variables ambientales y el control del riego. Los datos fueron recolectados por dispositivos IoT basados en el microcontrolador ESP8266 y procesados al borde de la nube por un servidor Edge basado en Android y arquitectura ARM, para posteriormente ser enviados a la base de datos persistente, esta fue configurada como fuente de datos de la herramienta de visualización Data Studio. Se configuró el sistema para interactuar con el usuario por medio de una interfaz gráfica por medio de una aplicación web. Se utilizaron diferentes habilitadores de la industria 4.0 que interactuaron entre sí como robótica, sensores, Internet de las Cosas y Big data. Para el desarrollo de la propuesta se utilizaron dos metodologías, SCRUM y POC.

Palabras clave: EDGE, IoT, ARM, NUBE, SBC, POC.

Abstract

Development of an Edge Computing system for strawberry (Fragaria) crop management under greenhouse, which allows the farmer to optimize resources and reduce costs by automating the monitoring of environmental variables and irrigation control. The data were collected by IoT devices based on the ESP8266 microcontroller and processed at the Edge of the cloud by an Edge server based on Android and ARM architecture, and then sent to the persistent database, which was configured as a data source for the Data Studio visualization tool. The system was configured to interact with the user through a graphical interface via a web application. Different Industry 4.0 enablers were used that interacted with each other such as robotics, sensors, Internet of Things and Big data. Two methodologies were used for the development of the proposal, SCRUM and POC.

Keywords: EDGE, IoT, ARM, CLOUD, SBC, POC.

Índice de contenidos

1. Introducción.....	12
1.1. Motivación	13
1.2. Planteamiento del trabajo	13
1.3. Estructura de capítulos	14
2. Contexto y estado del arte	15
2.1. Descripción general del contexto del proyecto	15
2.1.1. IoT en la agricultura	15
2.1.2. Beneficios del IoT en la agricultura	15
2.1.3. Edge Computing.....	17
2.1.4. Manejo de cultivo de fresa bajo invernadero	19
2.2. Proyectos relacionados con el tema del TFM	22
2.2.1. Productos disponibles en el mercado	22
2.2.2. Proyectos de investigación similares	24
2.3. Tecnologías relacionadas con el tema del TFM	26
2.3.1. Tecnologías habilitadoras de Edge Computing	26
2.3.2. Comunicación	26
2.3.3. Procesamiento.....	27
2.3.4. Analítica	32
2.3.5. Almacenamiento.....	32
2.3.6. Actuadores.....	33
2.3.7. Adquisición de datos	34
2.4. Conclusiones sobre el estado del arte	38
2.4.1. Comunicación	38
2.4.2. Procesamiento.....	39
2.4.3. Analítica	40
2.4.4. Almacenamiento.....	40

2.4.5.	Actuadores.....	41
2.4.6.	Adquisición de datos	41
3.	Descripción general de la contribución del TFM	43
3.1.	Objetivos	43
3.2.	Metodología del trabajo	43
3.2.1.	POC	43
3.2.1.	SCRUM.....	44
3.3.	Descripción general de las partes o componentes de la propuesta.....	46
4.	Desarrollo específico de la contribución.....	52
4.1.1.	Sprint 1: Definición	52
4.1.2.	Sprint 2: POC.....	58
4.1.3.	Sprint 3: Implementación.....	89
4.1.4.	Sprint 4: Pruebas.....	94
5.	Conclusiones y trabajos futuros.....	97
5.1.	Conclusiones.....	97
5.2.	“Líneas de trabajo futuras”.....	98
5.2.1.	Inteligencia artificial	98
5.2.2.	Evaluación del impacto de la tecnología en el rendimiento del cultivo.	98
	Referencias bibliográficas	99
Anexo A.	Código implementado.....	108
Anexo B.	Especificaciones técnicas de los dispositivos utilizados.	120

Índice de figuras

Figura 1: Obtenido de Edge and IoT-based Smart Agriculture. (Acharya et al., 2022).....	18
Figura 2: Sistema Plantect. (Plantect® (プランテクト®) 病害予測機能搭載モニタリングサービス, s. f.).	22
Figura 3: Aplicación agrícola OdinS. (OdinS, s. f.).	23
Figura 4: Libelium agricultura inteligente. (Libelium, s. f.).	24
Figura 5: Edge Computing framework (Acharya et al., 2022)	26
Figura 6: Arquitectura de la aplicación. (Elaboración propia).....	48
Figura 7: Instalaciones del invernadero de fresa (Elaboración propia).....	53
Figura 8: Panel fotovoltaico (Elaboración propia).....	53
Figura 9: Configuración del panel fotovoltaico (Elaboración propia).....	54
Figura 10: Vista satelital del centro experimental (Elaboración propia).....	54
Figura 11: Antena tp-link CPE210 y CPE510 Fuente: tp-link, s. f.....	55
Figura 12: SBC Tanix TX6 (Elaboración propia)	56
Figura 13: Bases de datos de los dispositivos (Elaboración propia).....	57
Figura 14: Son off BasicR2. (ITEAD, s. f.)	57
Figura 15: Esp8266 Cp2102 V2 Nodemcu Lua. (Naylamp, s. f.)	58
Figura 16: Escenario de red (CenturyLink, 2022).....	61
Figura 17: Conexión eléctrica antena AP (Siliceo, 2020).....	61
Figura 18: Esquema de conexión (Elaboración Propia).....	62
Figura 19: Interfaz de escritorio del servidor Edge (Elaboración Propia)	63
Figura 20: Pantalla final de la instalación (Elaboración propia)	65
Figura 21: Dashboard ngrok. (Elaboración propia)	66
Figura 22: Dashboard de monitoreo PM2 estadísticas de Node-red. (Elaboración propia) ...	66
Figura 23: Interfaz de autenticación Node-red. (Elaboración propia)	67
Figura 24: Interfaz de configuración de red Tasmota. (Tasmota, s. f.)	68
Figura 25: Administrador de red Fing. (Elaboración propia)	68

Figura 26: PCB Sonoff BASICR2. (Electrónica Joan, s. f.)	69
Figura 27: Conexión del adaptador serial con el Sonoff BASICR2. (Electrónica Joan, s. f.) .	69
Figura 28: Diagrama de conexión del módulo sensor de humedad relativa y temperatura. (Elaboración propia)	70
Figura 29: Parámetros del módulo sensor DHT21. (Elaboración propia)	71
Figura 30: Parámetros MQTT. (Elaboración propia)	71
Figura 31: Configuración de otros parámetros. (Elaboración propia)	72
Figura 32: Autenticación para ingresar a la interfaz de configuración. (Elaboración propia) .	72
Figura 33: Diagrama de conexión del módulo sensor de humedad del suelo FC-28. (Elaboración propia)	73
Figura 34: Parámetros del módulo sensor FC-28. (Elaboración propia)	73
Figura 35: Diagrama de conexión del módulo sensor de CO2 MH-Z19B. (Elaboración propia)	74
Figura 36: Parámetros del módulo sensor MH-Z19B. (Elaboración propia)	75
Figura 37: Diagrama de conexión del módulo actuador. (Elaboración propia).....	75
Figura 38: Parámetros del módulo actuador. (Elaboración propia)	76
Figura 39: Arquitectura MQTT del sistema. (Elaboración propia)	76
Figura 40: Flujo de interfaz y variables del sistema. (Elaboración propia)	77
Figura 41: Primera versión de la interfaz. (Elaboración propia)	77
Figura 42: Flujo de APIs embebidas. (Elaboración propia)	78
Figura 43: Interfaz mapa meteorológico. (Elaboración propia)	78
Figura 44: Flujo del sensor FC-28. (Elaboración propia).....	79
Figura 45: Flujo del sensor DHT21. (Elaboración propia)	79
Figura 46: Flujo del sensor MH-Z19B. (Elaboración propia)	79
Figura 47: Flujo para el control manual. (Elaboración propia)	80
Figura 48: Flujo para el control automático. (Elaboración propia)	80
Figura 49: Flujo para el monitoreo del estado de los actuadores. (Elaboración propia)	81

Figura 50: Flujo para capturar el tiempo de funcionamiento de un actuador. (Elaboración propia).....	81
Figura 51: Flujo para el monitoreo del estado de la conexión de los dispositivos. (Elaboración propia)	82
Figura 52: Bot de WhatsApp para notificaciones del sistema. (Elaboración propia)	82
Figura 53: Dashboard principal. (Elaboración propia)	83
Figura 54: Menú del sistema. (Elaboración propia)	83
Figura 55: Dashboard para el monitoreo de cultivo de fresa. (Elaboración propia)	84
Figura 56: Interfaz de control de eventos. (Elaboración propia)	84
Figura 57: Fuentes de datos Data Studio. (Elaboración propia)	85
Figura 58: Menú principal. (Elaboración propia).....	85
Figura 59: Reporte de humedad relativa. (Elaboración propia)	86
Figura 60: Modulo prototipo módulo actuador. (Elaboración propia)	86
Figura 61: Pruebas de adquisición de datos. (Elaboración propia)	87
Figura 62: Diagrama de conexión del sensor HD-38. (Elaboración propia)	88
Figura 63: Entorno de implementación del sistema. (Elaboración propia)	89
Figura 64: Instalación de la antena CPE510. (Elaboración propia)	90
Figura 65: Servidor Edge. (Elaboración propia)	90
Figura 66: Red de sensores inalámbricos. (Elaboración propia)	91
Figura 67: Modulo sensor inalámbrico de humedad del suelo. (Elaboración propia).....	92
Figura 68: Modulo sensor inalámbrico de CO2. (Elaboración propia)	92
Figura 69: Modulo sensor inalámbrico de humedad relativa y temperatura. (Elaboración propia)	93
Figura 70: Modulo actuador IoT. (Elaboración propia)	93
Figura 71: Reportes de dióxido de carbono. (Elaboración propia).....	94
Figura 72: Reporte de humedad del suelo. (Elaboración propia)	95
Figura 73: Reporte del punto de rocío. (Elaboración propia)	95

Figura 74: Reporte de temperatura. (Elaboración propia).....96

Figura 75: Reporte de humedad relativa. (Elaboración propia)96

Índice de tablas

Tabla 1: Parámetros ambientales para el cultivo de fresa	22
Tabla 2: Detalle de los valores unitarios de los componentes del sistema.....	49
Tabla 3: Presupuesto total de la inversión.....	50
Tabla 4: Retorno de la inversión	50
Tabla 5: Variables y sensores	58
Tabla 6: Hardware usado en la prueba de concepto.....	59

Índice de acrónimos

AP	Access Point
API	Application Programming Interfaces
ARM	Advanced RISC Machine
CO2	Dióxido de Carbono
CoAP	Constrained Application Protocol
GPIO	General Purpose Input/Output
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
LAN	Local Área Network,
LDR	Light Dependent Resistor
LORA	Long Range
MCU	microcontroller unit
MQTT	Message Queuing Telemetry Transport
PCB	Printed Circuit Board
POC	proof of concept
RH	Humedad relativa
SBC	Single Board Computer
SCADA	Supervisory Control and Data Acquisition
SOC	System on a Chip
TCP/IP	Transmission Control Protocol/Internet Protocol
TFM	Trabajo de fin de master
TRL	Technology Readiness Levels
TX/RX	receptores/transmisores
USB	Universal Serial Bus
UV	radiación ultravioleta
VM	Virtual Machine
WAN	wide area network
WIFI	Wireless Fidelity
WLAN	wireless local area network
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2
WSN	Wireless Sensor Networks

1. Introducción

Nuestra generación está viviendo una de las transformaciones más grandes de la historia de la humanidad, como sociedad los cambios que estamos sufriendo son comparables a los que se produjeron con el descubrimiento de la electricidad o la máquina de vapor. La cuarta revolución industrial trae consigo un conjunto de tecnologías que han tenido un impacto incalculable en cada una de nuestras actividades y la manera en que interactuamos tanto entre nosotros como con nuestro entorno.

A diferencia de otros sectores económicos que han avanzado rápidamente con la ayuda de la tecnología, en el Ecuador actualmente los procesos de producción agraria siguen siendo casi los mismos que hace 40 años, pese a que uno de los rubros más importantes de exportación son los productos agropecuarios, muy pocos avances tecnológicos se han visto en este sector, ya sea por falta de conocimiento, iniciativa o disponibilidad de la tecnología, lo cierto es que en este momento el común de los agricultores no poseen las herramientas necesarias para adaptarse a esta gran transformación.

Hablar de transformación digital en el sector agrícola es hablar de agricultura inteligente, bajo este concepto se pueden mencionar diferentes habilitadores tecnológicos de la cuarta revolución como la robótica, internet de las cosas, inteligencia artificial, tecnologías de la información y de la comunicación, etc. Pensar en una nueva forma de agricultura que utilice la tecnología y los datos nos permite facilitar la toma de decisiones en especial en cultivos bajo invernadero como la fresa, en torno al cual gira este proyecto para dar solución a problemas típicos como el bajo rendimiento, baja calidad en los productos y en el peor de los casos la pérdida de cultivos.

Según la investigación desarrollada por Grupo Editorial 3W México, (2021) en la cual menciona que el rendimiento que se obtiene en el cultivo de fresa bajo invernadero duplica a la producción a cielo abierto, esto significa que mientras el promedio de producción a cielo abierto es de 15 toneladas por hectárea al año, bajo invernadero se puede llegar a producir cerca de 40 toneladas siempre y cuando se realice un adecuado manejo de cultivo.

1.1. Motivación

Actualmente las técnicas para el manejo de cultivos de fresa dentro de invernaderos conllevan a un conjunto de actividades y procesos llevados a cabo de manera manual, como por ejemplo el monitoreo y control de humedad relativa, temperatura, riego, humedad del suelo, etc. Llevar un control diario de todas estas variables puede llegar a convertirse en una actividad tediosa para los agricultores, además el tiempo que transcurre entre la obtención de datos, el procesamiento y análisis de resultados puede ser muy prolongado y disminuir la capacidad de respuesta ante una eventualidad que ponga en riesgo el cultivo.

Las plantas de fresa no pueden sobrevivir en ambientes demasiado calientes, demasiado fríos o húmedos, incluso cuando disponen de luz, agua y CO₂. La fotosíntesis no se puede realizar bien si la temperatura es demasiado alta o baja, por otra parte, el requerimiento hídrico aumenta a medida que aumenta la temperatura, a pesar de que exista suficiente luz si la concentración de CO₂ cae por debajo de la concentración ambiental la tasa fotosintética disminuye.

Con el fin de crear un entorno ideal de crecimiento para las plantas de fresa y facilitar el manejo de cultivo, esta investigación trata el desarrollo de una solución tecnológica, en el contexto de la agricultura inteligente, el cual será capaz de facilitar el control del riego, además de monitorear de manera automática la temperatura, CO₂, Humedad Relativa, Punto de rocío y Humedad del suelo para la creación de reportes que facilitan la toma de decisiones en beneficio del cultivo, haciendo uso de tecnologías IoT y una arquitectura Edge Computing.

1.2. Planteamiento del trabajo

El presente trabajo describe el proceso para la implementación de una solución de bajo costo para Agricultura de precisión basada en Edge Computing con el cual se recolectan datos de los invernaderos del sector agrícola como la temperatura, CO₂, Humedad Relativa, Punto de rocío y Humedad del suelo por medio de sensores de suelo, higrómetro y sensor de CO₂ implementados en los cultivos, solucionando el riego innecesario y desperdicio de agua en los cultivos del sector agrícola.

Como primer punto se han escogido las tecnologías y los sensores que se van a utilizar para la implementación de la herramienta, asimismo, se ha realizado la configuración requerida para que la solución recolecte la información necesaria por medio de los diferentes sensores la cual será almacenada en una base de datos, permitiendo contar con la información para automatizar el riego del cultivo agrícola.

Según se vaya desarrollando la solución se ira detallando los problemas encontrados con la implementación y como se solucionaron.

1.3. Estructura de capítulos

A continuación, se describe de forma resumida, la estructura de los capítulos del presente proyecto.

Introducción

En este apartado se realiza una pequeña reseña de forma general sobre el problema a resolver en este Trabajo de Fin de Máster.

Contexto y estado del arte

En este apartado se realiza una búsqueda sistemática y minuciosa en las diferentes bases de datos de trabajo relacionados con la implementación de soluciones de bajo costo en la agricultura de precisión que sirvan de cimiento para el desarrollo del TFM. Asimismo, se realizará un estudio de las tecnologías de la industria 4.0 que permitan el desarrollo del proyecto.

Descripción general de la contribución del TFM

En este apartado se establecen los objetivos del proyecto y la metodología que se va a utilizar para la elaboración del trabajo.

Desarrollo específico de la contribución

En este apartado se desarrolla la estructura del trabajo según la metodología escogida para el desarrollo del proyecto, siendo este apartado el más importante del documento aquí se desarrollará la implementación de una solución de bajo costo para Agricultura de precisión basada en Edge Computing, desde la programación y configuración de los sensores, desarrollo y prueba finales de usabilidad.

Conclusiones y trabajos futuros

Finalmente, se explica las conclusiones obtenidas con la elaboración del proyecto, así también las diferentes líneas de trabajo donde se continuará con el desarrollo del proyecto.

2. Contexto y estado del arte

2.1. Descripción general del contexto del proyecto

2.1.1. IoT en la agricultura

La agricultura sostenible ha evolucionado en beneficio para los cultivos con el único objetivo de ofrecer un producto de calidad sin desperdicios al usuario final, es por ello por lo que se ha implementado la IoT en la agricultura automatizando los procesos de producción (Committee, 2018). El uso de la tecnología ayuda a los agricultores a tener una mayor producción optimizando los procesos para un cultivo más controlado y transparente (Pretty, 2007). La IoT permite realizar un análisis de datos, con el fin de identificar los requerimientos y predecir las necesidades de los cultivos y fertilizantes a aplicarse según el suelo y el medio ambiente (X. F. Wang et al., 2015). Uno de los retos más relevantes es detectar las enfermedades de las plantas y rociar pesticidas utilizando el IoT, disminuyendo las pérdidas económicas para los agricultores.

La producción agrícola en la India va disminuyendo con más frecuencia con el pasar del tiempo es por eso que los agricultores están obligados a adquirir el conocimiento adecuado para mitigar este problema (Mohanraj et al., 2016) definió en su artículo la aplicación e-agricultura la misma que está basada en el modelo de conocimiento (KM) el cual tiene una base de conocimientos con un modelo iterativo, así también, cuenta con un modelo de monitorización el mismo que está pendiente de las necesidades del entorno agrícola y de los cultivos. Con la aplicación de estos modelos inteligentes se ha minimizado los costos de inversión y la mano de obra dentro del entorno agrícola, debido que se ha automatizado la mayor parte de procesos y aumentando la producción, beneficiando principalmente a los agricultores rurales.

2.1.2. Beneficios del IoT en la agricultura

2.1.2.1. Monitoreo remoto como herramienta para la agricultura

El uso del IoT para recolectar información de manera remota con las herramientas tecnológicas para la agricultura, todo este proceso es controlado mediante herramientas de teledetección, para la correcta producción en la agricultura, el agricultor debe supervisar el campo de cultivo de vez en cuando para identificar las enfermedades o necesidad que requiere el cultivo. (Duque, 2017) en su trabajo de investigación describieron cómo un agricultor supervisa a distancia todo el proceso, haciendo uso de la IoT. (Triana Useche & Rodríguez Leguizamo, 2018) plantea un enfoque integrado para la supervisión medioambiental y de crecimiento de las plantas que ofrece una inversión asequible para la agricultura inteligente. Detecta el suelo ambiental, el nivel de agua y la humedad para el riego,

como también estima el crecimiento de las plantas por la profundidad y la altura por medio de una cámara de fotos de alta resolución mediante un dron basado en IoT.

2.1.2.2. Monitoreo del cultivo

El monitoreo de los cultivos es todo un proceso desde que se trabaja la tierra hasta que sale la cosecha, este proceso incluye también la correcta selección de semillas, humedad necesaria para el cultivo y la cantidad de pesticidas e insecticidas que se debe utilizar para la correcta cosecha. En un sistema agrícola existen varias etapas y muchas amenazas las cuales los agricultores deben controlar el suelo, la humedad, las condiciones meteorológicas, el control de pesticidas, el nivel de riesgo, etc. Esto se logra utilizando sensores integrados con microcontroladores, que permiten el servicio adecuado (Reátegui Arrivasplata, 2019).

2.1.2.3. Riego Inteligente

Los sistemas de riego inteligente adaptan el programa de riego a los tiempos de ejecución de manera automática para satisfacer las necesidades de los cultivos. Estos sistemas de riego inteligente mejoran el uso del agua en el riego, debido que solo se utiliza la cantidad requerida para el riego y es cerrado de forma automática, así no se desperdicia el líquido vital ayudando al medioambiente. Un modelo de riego inteligente en tiempo real es monitoreado a través de un teléfono móvil o a través de la IoT y de un sistema de base de datos que son generados en la nube por medio de los sensores de recolección de datos (Palacio, 2021).

2.1.2.4. Sistemas de Monitoreo y automatización de invernaderos

Un invernadero es una casa de cristal y todos sus procesos pueden controlarse bien por medio de sensores remotos y los correctos nutrientes en el suelo. El invernadero está bien automatizado por diferentes equipos científicos. Es buena iniciativa desarrollar sistemas inteligentes mediante la IoT debido que es un área limitara y el uso de la tecnología en la agricultura puede ser un buen acceso. La agricultura de precisión ha propuesto un modelo de invernadero bien controlado en que la tolerancia cero al medio ambiente es inofensivo con el aumento de la productividad de los cultivos (Felipe Osorio-Luna et al., 2019).

El sector agrícola se ha visto perjudicado por el desequilibrio de la agricultura con respecto al medio ambiente. Es así que para superar esta situación se desarrolló una combinación de IoT y un invernadero donde se evidencio una mayor producción como también se mitigo la contaminación ambiental gracias a la capacidad de controlar el riego y las necesidades de humedad y nutrientes. El sistema se diseñó de forma que se controles desde los móviles y los datos recolectados se almacenan en la nube (Felipe Osorio-Luna et al., 2019).

2.1.3. Edge Computing

El Internet de las Cosas (IoT) que es utilizado en las comunicaciones remotas inalámbricas el cual funciona mediante la interconexión de objetos por medio de sensores. Este paradigma fue desarrollado en el año 2006 de la mano de las grandes empresas tecnológicas como Google y Amazon que empezaron a ofrecer muchos servicios, promoviendo una transformación radical. Sin embargo, la computación en la nube tiene varias limitaciones, por lo tanto, surgió la computación de borde en el año 2002, donde se implementó un sistema de 12.000 servidores en más de 1000 redes consiguiendo con ello reducir la demanda de infraestructura (Dilley et al., 2002), ofreciendo un servicio más rápido a los usuarios que la utilizaba (Satyanarayanan, 2017) presentó cuatro nuevas arcas de investigación: invisibilidad, espacios inteligentes, enmascaramiento de condicionamiento desigual, y escalabilidad localizada, apoyando este concepto de computación distribuida en redes.

Seguidamente se implementó el servicio personalizado en la nube el cual se utilizaba por medio de una red local inalámbrica (LAN), donde se propuso una arquitectura de usuario móvil que era utilizada en máquinas virtuales (VM) (Satyanarayanan et al., 2009)

- Computación en la nube elástica: es un servicio web modificable para proporcionar capacidad computacional en la nube sin comprometer la seguridad.
- Computación Grid: es una computación distribuida en diferentes ordenadores dentro de la misma red que comparten recursos físicos o lógicos.
- Fog Computing: es utilizado para enviar información a distintos dispositivos conectados a la misma red y es almacenada en los mismos dispositivos y de esta forma llevar la capacidad de computación en la nube al borde de la red.
- Computación en la nube: es la que gestiona el acceso a los recursos de software desde cualquier parte del mundo, debido que es una aplicación de software que sirve a una variedad de clientes, ofreciendo servicios a través de la internet.

2.1.3.1. El rol del Edge Computing en diferentes campos de la agricultura

Las personas involucradas en esta rama de la producción pueden acceder y hacer uso de la agricultura inteligente de una manera más eficiente adoptando las metodologías del Edge Computing. El Edge Computing es una solución a los problemas de accesibilidad a los servicios de internet, se puede integrar fácilmente con otras tecnologías, se convierte en una solución muy atractiva para las múltiples facetas de la agricultura.

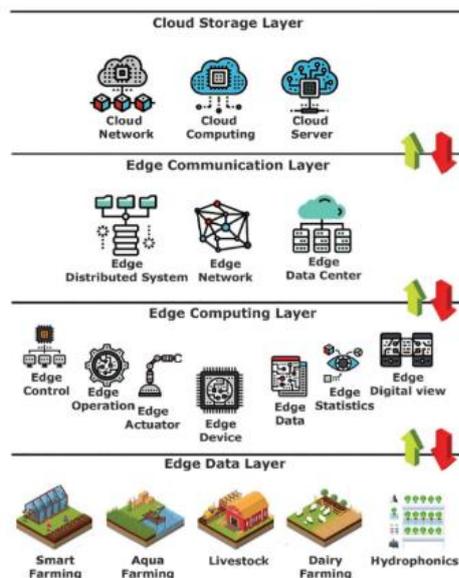


Figura 1: Obtenido de *Edge and IoT-based Smart Agriculture*. (Acharya et al., 2022)

2.1.3.2. Agricultura Inteligente

Según la investigación realizada por Conteratto et al., (2020) transmitieron las bases del análisis metagenómico en la computación de borde con respecto a agricultura, y proporcionaron múltiples opciones para los análisis microbianos del suelo, el agua y el aire. Los métodos tradicionales plantean la transferencia de datos considerables a la nube. Este tipo de combinación borde/nube proporciona una interesante metodología específica para la creación de flujos de trabajo de análisis completos, evidenciando en los resultados que hubo una disminución del 95% en la recolección de datos. Evidenciando que este método permite realizar un análisis metagenómico en lugares remotos. La transmisión de las tareas de procesamiento a la nube resulta en un aumento del coste debido a la ampliación de almacenamiento en la nube.

2.1.3.3. Hidroponía

Haciendo uso de la investigación desarrollada por (Sharma et al., 2018) donde ofrece una configuración económica general de sensores/actuadores apoyada en el IoT para ofrecer servicios de agricultura de precisión. Debido que la IoT sirve para modelar la computación de borde, es utilizada con un método multiprotocolo para el control del proceso. Los cultivos hidropónicos son realizados en un invernadero. Para el desarrollo de un modelo hidropónico de cultivo en un invernadero utilizaron un controlador habilitado para la niebla y en la nube se utilizó la plataforma ThingSpeak utilizando la capa de aplicación

2.1.4. Manejo de cultivo de fresa bajo invernadero

2.1.4.1. Descripción botánica y morfológica

La fresa (planta de la familia de las rosáceas); es una especie hortícola, se le considera herbácea, se caracteriza porque las hojas y otros órganos se forman en la parte leñosa de la corona y se le puede considerar como una planta perenne de vida corta.

2.1.4.2. Condiciones climáticas

Cada variedad de fresa se puede caracterizar según las horas de luz, esta característica dependerá de la ubicación geográfica del cultivo, podrían ser días largos o cortos y esta cantidad de luz influirá en la calidad del producto final.(Bonilla Correa, 2011)

Plantas de día corto: Variedades que responden al fotoperiodo (duración del día); requieren días cortos con una duración menor de 14 horas de luz para el desarrollo de yemas florales. En Colombia las variedades de día corto pueden presentar dos periodos de cosecha por temporada. Plantas de día neutro: Variedades que no presentan respuesta al fotoperiodo (duración del día); requieren de temperaturas en el suelo superiores a los 12°C para el desarrollo de yemas. La producción y el tamaño de los frutos es más homogéneo durante la temporada, dependiendo de la variedad; por lo general, la producción es más estable lo cual ayuda para realizar los estimados de cosecha y planeación (Núcleo Ambiental S.A.S, 2015)

2.1.4.3. Gestión de las variables climáticas de la fresa dentro de invernaderos

Basados en la información encontrada en los manuales de cultivo de fresa bajo invernadero publicados por (UNODC, s. f.) y (Núcleo Ambiental S.A.S, 2015) podemos ver que existen siete variables puntuales para un correcto manejo del cultivo, a continuación, se detalla la importancia de cada una de estas:

- **Temperatura**

La temperatura dentro del invernadero afecta la supervivencia y el crecimiento de las plantas. En invierno, es importante controlar la temperatura para que no baje demasiado desde la noche hasta la madrugada. Sin embargo, dependiendo del clima la temperatura puede bajar temprano en la mañana. En verano en cambio, la temperatura puede subir demasiado durante el día, así que se requiere verificar la temperatura y ajustar la iluminación y la ventilación. Estos parámetros pueden ser ajustados con riego por aspersion de agua, con neblina y otros (Lozano-Povis et al., 2021).

- **Humedad relativa**

Expresa la cantidad de agua contenida en el aire, es decir, el grado de humedad en el aire. Cuando la humedad relativa está fuera del rango adecuado, demasiado seco o húmedo, la fotosíntesis se reduce. Además, los períodos más prolongados de condensación a una humedad relativa del 90% o superior, se asocian con una mayor incidencia de enfermedades. Con base en la información obtenida por los sensores en el cultivo objetivo, se puede aplicar una función que permita predecir enfermedades (Jaén & Belén Estrada Valverde, 2021).

- **Concentración de CO₂**

El CO₂ es la fórmula química correspondiente al “dióxido de carbono” o “Anhídrido carbónico”. Los cultivos absorben agua y CO₂ para realizar la fotosíntesis cuando la temperatura, la humedad y la cantidad de luz solar son adecuadas. En general el ambiente por la mañana en un día soleado es propicio para la fotosíntesis porque hay mucho CO₂ en el aire y las plantas tienden a consumirlo en grandes cantidades (Fischer et al., 2022).

Dado que el invernadero está cerrado por la noche durante el invierno, la concentración de CO₂ es mayor en las primeras horas de la mañana debido a la respiración de las plantas, esto provoca que la fotosíntesis tenga mayor eficiencia. Hay que cuidar los niveles de CO₂ si se mantiene el invernadero cerrado. En verano el invernadero está ventilado, por tanto, la concentración de CO₂ en la parte interna difiere mucho de la cantidad exterior.

La concentración de CO₂ en el invernadero es una “medida de la concentración del CO₂ atmosférico” Cuando cae por debajo del nivel adecuado, se debe complementar con ventilación y aplicación de CO₂. Se debe lograr mantener el mismo nivel que en la atmósfera.

- **Radiación solar**

La fotosíntesis está activa cuando la intensidad de luz solar es alta, debido a esto los cultivos requieren una mayor cantidad de agua. Si la insolación es demasiado alta, los cultivos se marchitan. Esto elimina la humedad de la planta que se pierde a través de la transpiración. El marchitamiento se debe a la falta de agua, es una señal de que se está afectando la fotosíntesis y habrá daños posiblemente irreversibles en el cultivo (Rascón et al., 2020).

Se deben monitorear los valores de intensidad solar por medio de sensores para averiguar el punto de marchitez del cultivo. Como contra medida para la alta radiación solar se debe considerar utilizar una sombra, aumentar el riego y ventilar el lugar.

- **Punto de rocío**

Se forma una condensación cuando el aire se enfría. Cuando la diferencia entre la temperatura y la temperatura del punto de rocío es pequeña, el ambiente es propenso a la condensación. La fotosíntesis es baja en un ambiente donde es probable que ocurra condensación y se vuelve más susceptible a enfermedades.

- **Saturación**

La saturación es cuando más vapor de agua hay en el aire. Es una medida que se encuentra calculando a partir de la temperatura y la humedad. Se sugiere que se tome los datos durante el día.

Si la saturación es demasiado alta, el aire se secará y las estomas de la planta se cerrarán en cambio si la temperatura es demasiado baja, será difícil que la humedad se evapore y es menos probable que se capture CO₂, también un cambio repentino en la saturación hará que las estomas se cierren y, en cualquier caso, disminuirá la eficiencia de la fotosíntesis.

Cuando el valor de saturación está dentro del rango, la fotosíntesis se lleva a cabo activamente. Durante el día cuando la fatiga es demasiado grande, se recomienda humedecer el ambiente con aspersión o nebulización. El control de saturación abre las estomas y activa la fotosíntesis. Las estomas se abren y absorben CO₂ propiciando el proceso de fotosíntesis.

- **Temperatura acumulada**

La temperatura acumulada es la suma de la temperatura media diaria (cada día) desde la fecha del inicio del ciclo. Se sabe que la temperatura integrada requerida es aproximadamente constante para cada cultivo. Además, incluso en el mismo cultivo varía según la variedad y el entorno. Al predecir la fecha de cosecha a partir de la temperatura acumulada, se vuelve más fácil hacer un plan de trabajo estableciendo los intervalos de temperaturas en las etapas fenológicas. Se recomienda que la temperatura acumulada se tome desde la floración para predecir la fecha de cosecha. En el caso de la frutilla, el valor objetivo va de 350 – 600 °C (Lozano-Povis et al., 2021).

- **Humedad del suelo**

La humedad del suelo es determinada por una cantidad conocida de agua en una cierta cantidad de suelo o sustrato, este valor se puede calcular en base a ensayos de laboratorio como gravimetría o por medio de sensores. Se mide en porcentaje y dependiendo del tipo de suelo y el cultivo se pueden establecer rangos que alerten si el suelo está en punto de marchites permanente, capacidad de campo o saturado (Caicedo-Rosero et al., 2021).

En resumen, podemos decir que, en el caso de la fresa se requiere mantener la temperatura baja durante la noche puesto que es posible suprimir el agotamiento causado por la respiración y optimizar la fotosíntesis desde al amanecer hasta el día. También ayuda a reducir el consumo de combustible durante la noche. Si se requiere aplicar CO₂, serán unas 700 ppm cuando la instalación esté cerrada y 400 ppm cuando la instalación esté abierta. Se sugiere mantener los siguientes rangos de temperatura:

Tabla 1: Parámetros ambientales para el cultivo de fresa

Temperatura durante el día	18 – 25 °C
Temperatura durante la noche	5 – 10 °C
Concentración de CO ₂ durante el día	400 – 750 ppm

Adaptación de Eliseo Alvarado-Raya et al., (2022)

2.2. Proyectos relacionados con el tema del TFM

2.2.1. Productos disponibles en el mercado

2.2.1.1. Plantect



Figura 2: Sistema Plantect. (Plantect® (プランテクト®) | 病害予測機能搭載モニタリングサービス, s. f.).

Plantect es un servicio de seguimiento de los cultivos en invernadero. Este sistema de agricultura inteligente permite visualizar los entornos importantes del invernadero, como la temperatura, la humedad, el CO₂ y la radiación solar. La inteligencia artificial predice el riesgo de las principales enfermedades de los tomates y mini tomates, los pepinos y las fresas. Los datos que recoge el sistema el cual es informado a través de visualizaciones gráficas y avisos

a la aplicación por medio de notificaciones por correo electrónico. Los sensores adquieren datos ambientales cada 10 minutos, asimismo, se puede definir el ancho del gráfico este puede ser 1 hora, 24 horas, 7 horas y 30 días. Puede elegir el ancho del gráfico entre 1 hora, 24 horas, 7 días y 30 días. Utilízalo para gestionar el entorno actual o para mirar al pasado. También puede mostrar un gráfico de comparación con datos anteriores, lo que facilita la comprensión de los cambios anuales y estacionales, y obtener consejos sobre cómo mejorar el entorno. Los datos obtenidos pueden ser visualizados en cualquier momento y lugar desde un teléfono celular, tablet o PC, esto le permite controlar el invernadero incluso cuando se encuentre lejos de él, ya sea en la casa o en algún otro lugar (*Plantect® (プランテクト®) | 病害予測機能搭載モニタリングサービス, s. f.*).

2.2.1.2. Odins Invernadero Inteligente



Figura 3: Aplicación agrícola OdinS. (OdinS, s. f.).

OdinS permite la monitorización y telecontrol de los cultivos para automatizar el proceso de riego optimizando la cantidad de agua necesaria en tiempo real, según las condiciones ambientales y del cultivo por medio de sensores de temperatura, humedad y conductividad. Esta aplicación permite incorporar sensores agrónomos para monitorear el cultivo y controladores para automatizar el proceso de riego, Asimismo cuenta con una aplicación personalizable para visualizar los datos proporcionados por los sensores, enviar órdenes a los controlares del proceso de riego. El sistema de comunicación a Internet entre los sensores y controladores con la aplicación alojada en un servidor en la nube que proporcionará un acceso web desde cualquier lugar y dispositivo móvil (OdinS, s. f.).

2.2.1.3. Libelium agricultura inteligente

Libelium es un software de arquitectura inteligente que incluye una amplia gama de sensores agrícolas para su aplicación es diverso cultivos, ya pueden ser grandes campos de cereales, árboles frutales y hortalizas, invernaderos y flores. Es un modelo agronómico y predictivo de

enfermedades, automatización de riego, ofrece 3 años de leasing que le permitirán invertir en los aspectos más relevantes para hacer crecer su negocio.

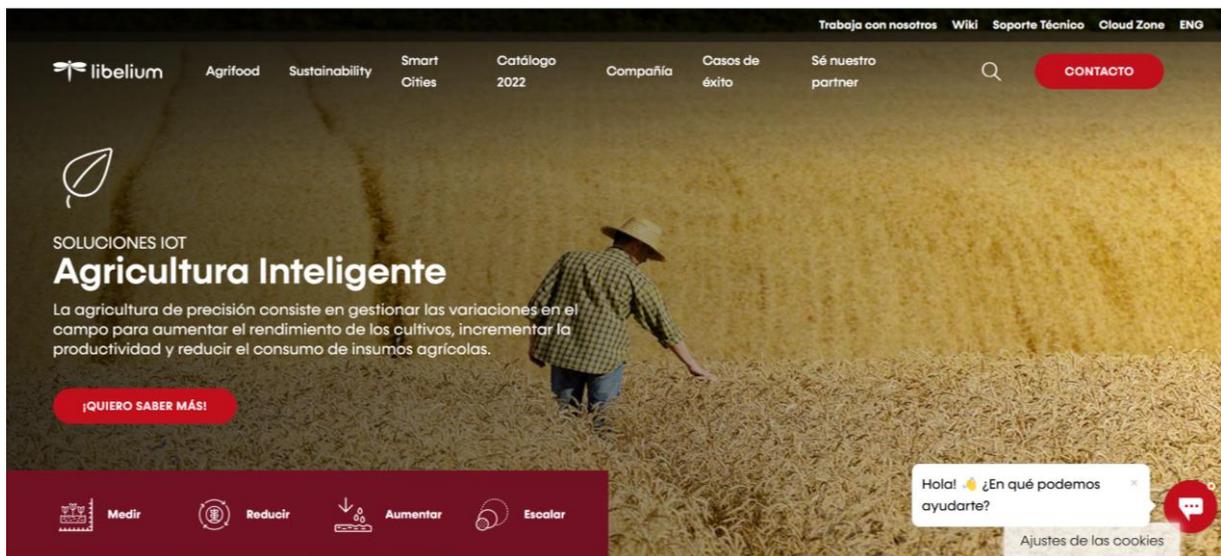


Figura 4: Libelium agricultura inteligente. (Libelium, s. f.).

Este software inteligente aumenta el rendimiento de los cultivos, la calidad del producto controla los factores de crecimiento y el rendimiento de los cultivos y la seguridad alimentaria. Asimismo, reduce la optimización de las tareas del agricultor, desplazamiento al terreno, pérdidas de cosechas por enfermedades o climatología adversa, uso de fertilizantes, pesticidas y consumibles como también el impacto de las sequías la escases y hambruna (Libelium, s. f.).

2.2.2. Proyectos de investigación similares

Según la investigación desarrollada por DH Fan y S Gao (Fan & Gao, 2018) proponen la construcción inteligente y el desarrollo de producción de un sistema para el control de riego en los cultivos de las zonas costeras de China. Con la aplicación tecnológica IoT, el proceso de monitorización de riego permite monitorizar mediante sensores en tiempo real la temperatura, humedad. Este trabajo de investigación propone una solución de gestión de enlaces de datos en combinación con la tecnología de Mobile Edge Computing, la cual permite realizar de manera precisa el hundimiento de los puntos de anclaje del negocio reduciendo el tiempo de respuesta. La aplicación práctica permite gestionar el sistema de monitorización del agua agrícola. Concluyendo que el sistema incluye todos los equipos terminales funcionales y utiliza una solución basada en la tecnología SOC. De la misma forma el sistema de monitorización utiliza la plataforma .net como entorno de desarrollo del software. Y también aprovecha los servidores en la nube para almacenar la información recolectada.

Como se señala en el proyecto de investigación realizado por (Peón-Lamparero, 2022) plantea diseñar e implementar un huerto inteligente autónomo de bajo coste conectado a una aplicación IoT, donde se estudió las diferentes tecnologías de la industria 4.0, diseñando un prototipo de huerto inteligente a bajo coste, asimismo para el desarrollo del proyecto de aplico dos metodologías POC y SCRUM. Obteniendo un producto final funcional para agricultura inteligente en el hogar.

De acuerdo con (Gaurav, 2022) hace mención que el uso de Node-Red es una plataforma de programación potente y fácil de utilizar, la misma que ayuda la implementación del IoT y la computación de borde, de igual forma se mención que el Node-RED puede interactuar con Raspberry Pi o Arduino para proporcionar comunicación en tiempo real con los sensores que se utilizaran en el desarrollo de la herramienta tecnológica para la agricultura inteligente, de tal forma que los agricultores harán un seguimiento de sus campos y cultivos. Node-RED también puede ser utilizado para la previsión meteorológica y el análisis del clima mediante plataformas de programación.

Como se menciona en el trabajo de investigación desarrollado en la ciudad de Villavicencio por Betancourth Catro, (2016) menciona que los sensores en cultivos agrícolas son fundamental para los sistemas de irrigación, puesto que permiten definir la cantidad de agua que está introduciendo en el suelo de los cultivos y evitar el desperdicio. Continuando con la investigación mención que estos sensores pueden ser colocados a varias profundidades, para identificar la cantidad de agua disponible en el suelo e identificar cuando es el momento de cerrar el suministro de agua a los cultivos. Evidenciando que son dispositivos de gran utilidad en el manejo del recurso hídrico de cultivos agrícolas. Concluyendo que un sensor de humedad tipo sonda para la medición de humedad a diferentes profundidades, basado en circuitos resonantes RLC, aplicando el principio de la variación del dieléctrico en el suelo permite recolectar los d datos humedad y son visualizados mediante tablas de datos y gráficos en un dispositivo móvil.

Nandhini et al., (2019) mencionan que la agricultura inteligente podría expandirse como un método de cultivo que trabaja en el proceso de pensamiento de una tecnología de moda para aumentar el rendimiento de la cantidad y la calidad de la mercancía agrícola. El sistema de agricultura inteligente basado en la IoT tiene como objetivo observar los cultivos de campo con el fin de identificar posibles enfermedades en las plantas por medio de los sensores y automatización del sistema de acuerdo con las necesidades. Los resultados de este método indican que los modelos de Edge Computing basados principalmente en IoT son muy económicos y extremadamente rápidos y pueden resultar y proporcionar resultados más altos cuando se ponen al lado de diferentes sistemas.

2.3. Tecnologías relacionadas con el tema del TFM

A continuación, se presenta una estructura de tecnologías habilitadoras de Edge Computing, las mismas están estrechamente relacionadas con las tecnologías de la industria 4.0, y se expondrá brevemente cada una de ellas.

2.3.1. Tecnologías habilitadoras de Edge Computing

Desde el punto de vista del nivel de preparación tecnológica (TRL, Technology readiness level), los diferentes sistemas explicados en las secciones anteriores son principalmente prototípicos y se extienden desde la validación TRL 4 en el laboratorio hasta la validación TRL 5 en un entorno aplicable (Mela et al., 2021). En resumen, el dominio de los servidores perimetrales físicos combinados con sensores locales y redes de sensores se refleja en la posición de la computación perimetral en la agricultura. A pesar de ser independiente, la computación perimetral puede ayudar al internet de las cosas IoT a resolver ciertos problemas cruciales, mejorando así el rendimiento. La integración de características similares de IoT y Edge Computing está considerablemente desarrollada, pero aún está experimentando varias mejoras incorporadas en un marco de Edge Computing, como se muestra en la Figura 5



Figura 5: Edge Computing framework (Acharya et al., 2022)

2.3.2. Comunicación

Las aplicaciones IoT hacen uso de una amplia variedad de tecnologías de comunicación del internet de las cosas para establecer su comunicación en la computación de borde, reduciendo costos y los requisitos energéticos teniendo un amplio rango de cobertura. Las tecnologías de comunicación más utilizadas dentro de este rango con las WSN, a diferencia del Wifi y Zigbee que utilizan la transferencia de datos en menor medida (Sarker et al., 2019).

2.3.2.1. Tecnologías inalámbricas

- Lora WAN es un protocolo de red que usa tecnología Lora, para redes de baja potencia y área amplia, está conformado por componentes de gateway y nodos (Bankov et al., 2017).
- SigFox es una red de comunicación celular inalámbrica de largo alcance, es utilizada para localizar los pastos para el ganado y la ubicación del ganado (Gómez et al., 2019).
- Zigbee es un desarrollo los mejores marcos IEEE 802 que permite la transferencia de datos a alta velocidad para los distintos dominios, en especial agrícolas, debido a la gran velocidad de intercambio de datos y la duración de la batería a largo plazo (Alobaidy et al., 2020).
- Wifi permite el intercambio de información de manera inalámbrica mediante la red informática y se enmarca en una red de área local sin cables (WLAN) teniendo un rango de alcance de 20-100 m y un rango de transmisión de datos de 2-54 mbs. Wifi permite conectar varios dispositivos de agricultura bajo una red local (Forero-Muñoz & Ramírez-Delgado, 2019).

2.3.2.2. Protocolos de comunicación

- MQTT es un protocolo de publicar/suscribir elaborado por SCADA y redes remotas, centrado en un mínimo encabezado y comunicaciones confiables.
- CoAP es un protocolo de aplicación restringida, provee la compatibilidad de HTTP con una mínima carga reduciendo el encabezado y requerimiento
- HTTP es un protocolo cliente/servidor sin conexión ubicado en la web, donde los clientes pueden acceder a los recursos en el servidor a través de peticiones al servidor.
- TCP/IP es un protocolo de red que permite la comunicación en redes que permite que un ordenador pueda comunicarse dentro de una red, permitiendo él envió de datos entre ordenadores.

2.3.3. Procesamiento

2.3.3.1. Dispositivos Edge

Los dispositivos Edge se han extendido a todos los aspectos de la sociedad, como hogares inteligentes y vehículos autónomos en el campo del transporte, cámaras, robots de producción inteligente en fabricación inteligente, etc. Como resultado, el número de dispositivos conectados a Internet ha aumentado significativamente (Cao et al., 2020).

- Microcontroladores

Un sistema de puerta de enlace convencional típico utiliza un microcontrolador integrado (MCU) de alto nivel como núcleo; que MCU realiza administración de red de dispositivos de capa de percepción de bajo nivel, funciones de servidor en la nube de nivel superior y servicios de computación móvil remota (Chen et al., 2018).

El ESP8266 es un microcontrolador inalámbrico cuyas características principales hacen que sea particularmente atractivo para aplicaciones IoT, entre ellas están:

- Cumplimiento del estándar IEEE 802.11 b/g/n (wifi) con antena a bordo;
 - Protocolos de seguridad WPA/WPA2;
 - Procesador de silicio M106 RISC de 32 bits con 160 MHz velocidad máxima del reloj;
 - SRAM interna de 36 kb y SPI externa de 4 Mb flash varios modos de suspensión nativos;
 - Factor de forma pequeño (24x16 mm) que lo hace integrable en muchos dispositivos;
 - Muy bajo costo, particularmente cuando se compara con otros módulos Wifi (Mesquita et al., 2018).
- Arduino nano

Arduino Nano es uno de los microcontroladores que es muy pequeño, bastante completo y se puede usar en una protoboard. Arduino Nano fue creado con la base microcontrolador ATmega 328 para Arduino Nano con la versión 3.x o ATmega168 para Arduino con la versión 2.x. en este sistema Arduino nano funciona como un procesador que procesa la señal aporte del sensor LDR (Asmi & Candra, 2020).

El microcontrolador ESP32 tiene dos núcleos, que se gestionan de forma independiente, comunicación Bluetooth y comunicación Bluetooth de baja energía (BLE), cuatro temporizadores independientes, un convertidor digital a analógico (DAC) con sensores táctiles capacitivos y un sensor de efecto Hall (Cameron, 2021).

- Microprocesadores

Los microprocesadores están en el corazón de todos los dispositivos electrónicos, incluidos los teléfonos inteligentes, tabletas, computadoras portátiles, enrutadores, servidores, automóviles y, más recientemente, objetos inteligentes que conforman el Internet de las cosas (Biggs et al., 2021).

ARM CORTEX-A9 es un microprocesador de gama alta presente en múltiples System-on-Chip programable que se utiliza en muchos campos de aplicación, incluidos aquellos con altos requisitos de confiabilidad (Aviles et al., 2021).

El Cortex-M4 se puede considerar en el extremo superior de los microcontroladores. Por lo general, la placa de desarrollo ARM Cortex-M4 ampliamente disponible también está equipada con una gran RAM y memoria flash (Kannwischer et al., 2019).

El microprocesador S3C2410 es el núcleo de todo el sistema de hardware, controla el inicio de los circuitos circundantes en tiempo real cuando se enciende, apaga, restablece, etc., con monitoreo en tiempo real del estado de funcionamiento de los módulos de circuito circundantes (T. Wang et al., 2022).

Las computadoras de placa única (SBC) de productos básicos actuales son lo suficientemente potentes como para ejecutar sistemas operativos y cargas de trabajo convencionales. Muchas de estas placas pueden estar vinculadas entre sí para crear clústeres pequeños y de bajo costo que replican algunas características de los clústeres de centros de datos grandes (Johnston et al., 2018).

Raspberry Pi 4 Modelo B es una computadora de placa única, lo que significa que no tiene las capacidades de conexión, programa y juego como la placa Arduino. Tiene que pasar por la fase de configuración e inicialización para instalar el sistema operativo en la Raspberry Pi (Jabbar et al., 2021).

Orange Pi actúa como un analizador de base de datos donde los datos se almacenan y procesan para retener resultados que pueden ser útiles para uso futuro. Orange Pi presenta como un sistema de comunicación inalámbrico liviano que puede manejar Arduino en una gran transmisión de datos y puede controlar Arduino para obtener datos (Meduri et al., 2020).

El Beaglebone se puede alimentar a través de:

- Cable USB; entonces opera a 500 MHz.
- Conector de alimentación de 5 V CC; entonces opera a 720 MHz.

La tarjeta microSD funciona como disco duro para la placa y alberga el sistema operativo. El Beaglebone funciona como un sistema independiente, pero puede combinarse con varias placas secundarias, llamadas "capas", y bibliotecas de código abierto para crear una variedad de sistemas integrados personalizados (Domínguez-Pumar, 2018).

2.3.3.2. Plataformas IoT adaptadas al Edge Computing

- Node-Red

Es una plataforma que se utiliza en el internet de las cosas para realizar el enlace de conexión entre los dispositivos electrónicos y las interfaces de programación de aplicaciones (API), por lo tanto, ofrece herramientas de programación de código abierto para trabajar en línea con las

plataformas enlazadas con la gestión de los datos en la nube. Utiliza un software de colaboración abierta donde cualquier persona tiene acceso a las líneas de programación escritas. La programación de Node-RED que se ha basado en un flujo dentro del espacio de trabajo donde se construye flujos de nodos a partir de la conexión de nodos con funciones ya antes programadas (*Node-RED*, s. f.)

- Ubidots

Es la plataforma por la cual se gestiona y almacena en la nube de los datos recolectados por los controladores electrónicos, como también para mostrar los mapas de datos en línea por medio de un SCADA de acceso remoto. Una ventaja muy importante que presenta Ubidots que es compatible con Node-RED, lo que permite la comunicación y transferencia de la información a través del nodo de salida “mqtt” de Node-RED hacia la plataforma Ubidots. (*Ubidots*, 2022)

- ThingsBoard Edge

Permite llevar el análisis y la gestión de los datos al borde, donde se crean los datos, de igual se sincroniza son problemas con la nube de ThingsBoard según las necesidades de su negocio (*ThingsBoard Edge | ThingsBoard*, s. f.).

2.3.3.3. Firmware IoT

Según la investigación realizada por Arakadakis et al., (2021) un Firmware es un soporte lógico inalterable, en el caso de los dispositivos IoT este se encarga de la lógica y los procesos de un dispositivo a bajo nivel, se integra con la parte electrónica y tiene una estrecha interacción con el hardware, en el contexto del internet de las cosas simplifican la configuración de los microcontroladores, existen varias iniciativas de código abierto que han desarrollado soluciones para un despliegue más eficiente en el campo del IoT, las mismas cuentan con una creciente comunidad que constantemente mejora sus características para adaptarse a las diferentes necesidades del constante crecimiento de este tipo de tecnologías, a continuación algunas de las más populares.

- Tasmota

Tasmota es un firmware diseñado para dispositivos basados en microcontroladores ESP8266 y ESP32, una de sus principales características es la facilidad de configuración que presta por medio de una interfaz web, permite configurar los dispositivos para usar diferentes protocolos de comunicación como MQTT, HTTP, SERIAL o KNX. Es escalable y gracias a su comunidad se encuentra en constante desarrollo. (*Tasmota*, s. f.)

- ESPEasy

ESPEasy es un proyecto de código abierto que cuenta con cerca de cien voluntarios que trabajan en su constante desarrollo, soporta dispositivos basados en los microcontroladores ESP8266, ESP8285, ESP32 y ESP32-S2. Cuenta con distintas herramientas para gestionar la configuración de dispositivos IoT, al igual que Tasmota cuenta con una interfaz de usuario que gestiona la configuración de dispositivos sin necesidad de escribir código (ESP Easy's, s. f.).

- ESPurna

Es una iniciativa de código abierto que ofrece un firmware optimizado para sensores, luces e interruptores, soporta dispositivos basados en ESP8285 Y ESP8266, utiliza el entorno Arduino Core para los microcontroladores ESP8266 y bibliotecas de terceros para la configuración de diferentes dispositivos IoT disponibles en el mercado Barlam Aspachs et al., (2010).

2.3.3.4. Sistemas operativos

Al nivel de un servidor Edge el sistema operativo facilita la interoperabilidad y escalabilidad del sistema, existe una amplia gama de sistemas operativos diseñados para dispositivos del tipo SBC basados en microprocesadores ARM, cada uno cuenta con características específicas según el campo de la aplicación y programas a ejecutarse, a continuación, se detallan los sistemas operativos más pertinentes en el contexto de este proyecto.

- Raspberry Pi OS

Raspberry Pi OS es un sistema operativo diseñado para los dispositivos Raspberry Pi, anteriormente llamado Raspbian, es un sistema operativo basado en Linux optimizado para ejecutarse específicamente en esta marca de SCB con microprocesadores basados en arquitectura ARM. (Upton & Halfacree, 2014)

- Windows 10 IoT Cores

Es una versión de Windows 10 optimizada para dispositivos SBC, con microprocesadores basados en ARM. Está diseñado para trabajar con o sin interfaz y proporciona información del dispositivo para gestionar la conexión, administración, actualización, seguridad y más. (Windows IoT, 2022)

- Android

Es el sistema operativo más utilizado en la actualidad, está diseñado para dispositivos móviles y se basa en el núcleo de Linux y otros proyectos de código abierto. El código fuente principal es conocido como AOSP que está licenciado principalmente bajo Apache. Actualmente lo podemos encontrar en teléfonos inteligentes, tabletas, automóviles, televisores, etc. Al igual

que los sistemas mencionados anteriormente este se ejecuta sobre dispositivos con microcontroladores basados en ARM. (Gironés, 2019)

2.3.4. Analítica

- Power BI

Pertenece a Microsoft de modo que permite el trabajo con múltiples datos al convertirlos en un informe, gráfico o visualización de datos. Es un programa de fácil aplicación y excelente para visualizar los datos (Excelprince, s. f.).

- Data Studio

Es un producto desarrollado por Google destinado a la visualización de datos ligeramente distinta a Power BI al ser una herramienta gratuita e independiente del sistema operativo, su interfaz es semejante a los productos de Google (Jaakkola, 2022).

- Grafana

Es una herramienta de consulta, visualización y alerta de datos de series temporales de código abierto. Al ser un modelo de fuente de datos altamente conectable soporta diferentes fuentes de datos en series temporales como: Prometheus, InfluxDB y OpenTSDB, y bases de datos SQL como MySQL y Postgres. Está destinado a la consulta de datos a través del editor de consulta, cuyos resultados se logran visualizar a través de paneles de control (Chakraborty & Kundan, 2021).

2.3.5. Almacenamiento

2.3.5.1. Local

Este tipo de almacenamiento está directamente relacionado con la capacidad que posee el hardware IoT para almacenar una cantidad limitada de información, estos dispositivos están diseñados para que el almacenamiento de información sea efímero y que el microcontrolador se enfoque en el procesamiento y comunicación, los datos que son almacenados de esta forma se conservan hasta ser enviados a otros sistemas o hasta que el dispositivo sea reiniciado. (Premkumar & Sigappi, 2022)

2.3.5.2. Edge/Cloudlet

Es uno de los elementos clave en la arquitectura Edge Computing, resulta de la convergencia de los dispositivos móviles y el cloud Computing. Explicado de una manera más simple se puede definir con un mini centro de datos que recoge información de diferentes dispositivos o sistemas para posteriormente interactuar con la nube y almacenar dicha información dentro de centros de datos más potentes. Dicha solución optimizada es empleada para reducir la

latencia de computación y transmisión, disminuir el consumo de energía dentro del procesamiento y comunicación, minimizando el ancho de banda en aplicaciones de IoT (Premkumar & Sigappi, 2022).

2.3.5.3. Cloud

El procesamiento o almacenamiento de datos requiere más recursos que los proporcionados de forma razonable por los dispositivos de almacenamiento local o Edge/cloudlet; tanto el procesamiento como almacenamiento requieren cumplirse en los servidores convencionales de la nube. Éstos contienen una capacidad de cálculo mayor, garantizando menor latencia computacional para la ejecución de tareas en los servidores de la nube (Premkumar & Sigappi, 2022).

2.3.6. Actuadores

Los actuadores recurren a distintas metodologías para el mejoramiento de procesos mecánicos repetitivos, dando paso al aumento de demandas en los servicios impulsados por actuadores dentro del área agrícola. Ellos se adaptan a cambios climáticos y exposición a distintos fertilizantes. De modo que los equipos de agricultura requieren una variedad de tipos de actuadores para las multitareas con relación al campo e invernaderos. Entre los actuadores se encuentran los termostatos, bombas de agua y equipos de oxígeno disuelto para el control de la concentración del agua (Premkumar & Sigappi, 2022).

2.3.6.1. Luz Ultravioleta

Tanto la intensidad de la luz como la radiación UV corresponden a un conjunto de parámetros que se controlan con frecuencia en los sistemas de riego IoT (García et al., 2020). La activación por luz ultravioleta es una fuente de energía de activación de la de la superficie de los materiales sensores (en particular, los óxidos metálicos) (Alali et al., 2020).

- Lámparas

Existen casos donde el crecimiento del tallo gira en la dirección de la luz solar, así existan lámparas ultravioletas (UV) situadas por encima de la planta. Esto quiere decir que en casos donde se utilizan las lámparas UV como fuente de luz principal, no significa que sea suficiente para las plantas (Fakhrurroja et al., 2019).

2.3.6.2. Bombas de agua

Una bomba de agua tiene como objetivo principal transportar el agua desde un punto hacia otro. El uso de las bombas en agricultura es esencialmente para suministrar agua desde el

depósito hasta el campo (Pathak et al., 2021). Existen diferentes tipos de bombas de agua a continuación las más comunes en aplicaciones agrícolas.

- Bombas sumergibles.
- Bombas centrífugas.
- Bombas de desplazamiento positivo.

La selección de una bomba dependerá del estudio y diseño del sistema de riego, dentro del entorno IoT una bomba de agua, así como cualquier actuador se puede comandar por medio de sistemas embebidos.

2.3.6.3. Electroválvulas

Como su nombre lo indica una electroválvula es la combinación entre una válvula convencional y un solenoide actuador que por medio un sistema mecánico utiliza la electricidad para abrir o cerrar la misma. (Gómez et al., 2006).

2.3.6.4. Ventiladores

Los ventiladores son utilizados para la circulación del aire y, por ende, para cambiar la temperatura ambiental (Marcelino et al., 2018).

2.3.6.5. Dispositivo actuador IoT

Estos dispositivos se caracterizan por transformar una señal en una acción, su principal finalidad es crear automatizaciones tanto de lazo abierto como de lazo cerrado, básicamente nos permiten comandar de manera remota cualquier dispositivo que trabaje con energía eléctrica. Se puede encontrar una gran variedad de este tipo de dispositivos en el mercado, la selección dependerá exclusivamente de ámbito de aplicación. Son PCB que cuentan con un microcontrolador que permite conexiones inalámbricas y al menos uno o varios relés. Se puede construir este tipo de dispositivos por medio de microcontroladores y módulos relevadores, es muy fácil adquirirlos casi en cualquier parte por su bajo costo y popularidad en el campo de la domótica (Khakimov et al., 2017).

2.3.7. Adquisición de datos

La adquisición de datos se refiere a un proceso de muestreo de señales que tiene como objeto medir las condiciones físicas del mundo real, para después convertir las muestras resultantes en valores numéricos digitales, los cuales pueden ser manipulados por medio de un ordenador. Los sistemas de adquisición de datos suelen convertir las formas de onda analógicas en valores digitales para su procesamiento (Kiran & Gupta, 2020).

2.3.7.1. Sensores de humedad del suelo

- Sensor resistivo FC-28

Este sensor permite la medición de humedad del suelo de manera sencilla mediante 2 electrodos resistivos. La funcionabilidad de los sensores está basada en la resistencia entre 2 electrodos insertados dentro del suelo. La resistencia que se utilice en el suelo dependerá de la humedad, por lo que para los suelos muy húmedos se utiliza una resistencia muy baja mientras que para el suelo seco la resistencia es muy alta. Para el correcto funcionamiento del sensor el electrodo se conecta a una tarjeta de acondicionamiento (YL-38) de donde se obtiene una señal digital y otra analógica (MEDICION DE LA HUMEDAD DEL SUELO, 2018). La salida digital (DO) es activada cuando la humedad del suelo es menor al nivel deseado, este nivel se regula con el potenciómetro de la tarjeta, y, la salida analógica (AO) es la salida de un divisor de tensión entre una resistencia fija y una resistencia entre electrodos, la medición de la salida analógica es 0V un suelo muy húmedo y 5V para un suelo muy seco. Para la conexión a Arduino podemos optar por utilizar la salida analógica del módulo conectada a una entrada analógica del Arduino o por utilizar la salida digital (DO) conectada a una entrada digital del Arduino. (MEDICION DE LA HUMEDAD DEL SUELO, 2018)

- Sensor de Humedad Capacitivo

Como su nombre lo indica el sensor de humedad capacitivo se basa en el principio físico de la capacitancia, a diferencia del sensor de humedad FC-28 que se basa en la resistividad, por medio de dos electrodos al ser introducidos en la tierra mide la capacitancia entre ambos, si es muy baja quiere decir que el suelo está muy húmedo y si es muy alta el suelo está muy seco, este principio de funcionamiento permite que el sensor posea una vida útil más larga. Es importante saber que los datos que podemos medir con este sensor son de tipo cualitativo. (Naylamp Mechatronics, s. f.-a).

- Sensor de humedad resistivo anticorrosivo

Este tipo de sensor al igual que el FC-28 se basa en el principio físico de la resistividad para medir la cantidad de agua presente en el suelo, posee dos electrodos anticorrosivos que permiten alargar su vida útil en comparación a su predecesor, una característica importante de este sensor es su tarjeta de acondicionamiento, específicamente el modelo LM293, esta es de uso muy común en muchos sensores analógicos (Electronilab, s. f.).

2.3.7.2. Sensores de Temperatura, humedad relativa.

- DHT11

Es un sensor digital de temperatura y humedad, fácil de implementar con cualquier tipo de microcontrolador, el mismo que utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de datos, siendo una desventaja latente la velocidad de para tomar lecturas de datos y el tiempo de espera para una nueva lectura. Este sensor trabaja en un rango de medición de temperatura de 0 a 50° con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos (Naylamp Mechatronics, s. f.).

- DHT21

Es un sensor digital de temperatura y humedad relativa de buena precisión en un empaque robusto. Integra un sensor capacitivo de humedad, un termistor y un microcontrolador encargado de realizar la conversión analógica a digital. Su empaque de plástico es más robusto comparado a los sensores DHT11 y DHT22, esto hace del DHT21 un sensor ideal para aplicaciones en exteriores. Utilizado en aplicaciones de control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura con las plataformas Arduino/Raspberry, Pi/Nodemcu, siendo muy sencillo de utilizar a nivel de software y hardware (Garnier-Fernández, 2019)

- DHT22

El rango de medición de temperatura es de -40°C a 80 °C con precisión de ± 0.5 °C y rango de humedad de 0 a 100% RH con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos (Naylamp Mechatronics, s. f.).

2.3.7.3. Sensores de CO2

Un sensor de CO2 es un instrumento que se utiliza para la medición de gas de dióxido de carbono en un ambiente determinado. Habitualmente estos aparatos registran el dióxido de carbono en partes por millón (ppm) en los espacios ocupados y nos ofrecen una muestra de la concentración de este gas en el aire que respiramos.

- Sensor MQ135

Es un sensor perteneciente a la serie de sensores MQ los cuales están diseñados para un tipo de gas en concreto, este sensor tiene un amplio rango de detección, alta sensibilidad y rápida respuesta, estable y de larga duración con un circuito sencillo. Siendo capaz de detectar el dióxido de carbono CO2, Amoniacó NH3, Nitratos y nitritos, NOX, Alcohol, Benceno y Humo (Santos, 2021).

- SEN0159

Es un sensor de tipo industrial diseñado por DFROBOTS, el cual mide la concentración de gas presente en el ambiente. La salida de voltaje decrece en la misma proporción en que la concentración de CO₂ presente en el ambiente se incrementa. Tiene una salida de voltaje análogo, donde su principal componente en la tarjeta es el sensor MG-811 de alta sensibilidad al CO₂ y baja reacción al alcohol, con alta estabilidad y reproducibilidad en la medida (Sigma Electrónica, s. f.).

- MHZ-19B

La principal característica de este módulo es su pequeño tamaño, se basa en el principio de medición infrarroja no dispersiva para detectar la cantidad de CO₂ en el aire, gracias a que no depende de la presencia de oxígeno posee una buena sensibilidad, posee auto compensación de temperatura y está diseñado para realizar mediciones muy precisas gracias a un muy buen diseño de su sensor óptico. (Zhengzhou Winsen Electronics Technologia, s. f.).

2.3.7.4. Sensor de luminosidad

- BH1750

Es un sensor de luz digital que entrega valores de medición en Lux (lumen/m²) que es una unidad de medida estándar para el nivel de iluminación. Teniendo una alta precisión y un rango entre 1 - 65535 lx, mismo que es configurable. De igual forma trabaja con 3.3 a 5 voltios sin problemas, teniendo una interfaz de comunicación I2C que permite implementarse con la mayoría de los microcontroladores (Naylamp Mechatronics, s. f.-b).

- LDR (GL55)

Es un sensor que varía su resistencia dependiendo la luz recibida, donde se puede utilizar esta variación para mediar, a través de las entradas analógicas, una estimulación de nivel de luz. Está formado por un semiconductor, típicamente sulfuro de cadmio Cds. Al incidir la luz sobre los fotones que son absorbidos, permitiendo que los electrones pasen a la banda de conducción y, por tanto, disminuyendo la resistencia del componente. Los valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante (Llamas, 2022).

- APDS-9930

Es un sensor que mide la luz ambiental. El cual puede detectar el brillo de 0,01 Lux en la oscuridad, además se puede utilizar detrás de un cristal oscuro, como la pantalla de un dispositivo móvil. Como hacen los dispositivos móviles, apaga la pantalla táctil durante las llamadas cuando el dispositivo se encuentra cerca del oído. Asimismo, es un sensor que se puede utilizar para ajustar el brillo de la luz en sus diversas aplicaciones (ArduinoQue, 2022).

2.4. Conclusiones sobre el estado del arte

Después de la revisión de las tecnologías presentadas en la sección anterior y el análisis de los proyectos planteados en el contexto de la agricultura inteligente, se puede evidenciar que el Edge Computing es una solución muy eficiente para el campo de la producción.

Basados en la investigación se puede concluir que a nivel mundial existen muy pocos estudios sobre el manejo de cultivos de fresa usando la metodología Edge Computing de hecho, en la región no existe ningún estudio similar hasta la fecha de publicación de este documento. Es importante recalcar que uno de los países más avanzados en este tema es Japón y a diferencia de Ecuador que es donde se realiza este estudio sus prácticas agrícolas son mucho más tecnificadas y en consecuencia sus cultivos son mucho más rentables.

Según la investigación realizada gracias a la agricultura inteligente es posible aumentar en un 15% el rendimiento de los cultivos, esto es el resultado del estudio de las principales variables climatológicas, por medio de la tecnología estas variables pueden ser monitoreadas y controladas oportunamente para reducir el uso de productos químicos y reducir el riesgo de enfermedades.

En base a las diferentes tecnologías habilitadoras de la infraestructura Edge Computing que se analizaron en la sección anterior a continuación, se detalla cada una de las tecnologías seleccionadas para el desarrollo de este proyecto y cuál fue el criterio de selección.

2.4.1. Comunicación

La tecnología de comunicación seleccionada fue Wifi, a pesar de que dentro del mundo IoT las tecnologías más usadas son SigFox, Zigbee o Lora WAN por su bajo consumo energía y alcance en despliegues sobre superficies extensas, en este proyecto se consideraron otros factores importantes como son el costo y la disponibilidad de estas tecnologías en el mercado local.

El estándar Wifi nos permite hacer uso de una amplia oferta de dispositivos de comunicación y dispositivos inalámbricos que reducen significativamente los costos de la solución, es importante recalcar que la selección de cada una de las tecnologías de este proyecto se basa en estos criterios pero además en este caso se consideró la facilidad del despliegue, en la actualidad esta tecnología es la más común a nivel de uso doméstico y es la que ha permitido que las tecnologías del IoT lleguen a todo el mundo, es por eso que las empresas que diseñan y fabrican estos dispositivos han facilitado mucho su configuración por medio de interfaces amigables que no requieren los conocimientos de un experto para ser desplegados en muy poco tiempo.

El protocolo de comunicación elegido es MQTT en la distribución de código abierto que ofrece Mosquitto Eclipse, se consideró que este es el más adecuado en base las características del sistema, las tecnologías de comunicación, los dispositivos inalámbricos, la interoperabilidad entre dispositivos heterogéneos y la principal característica que se tomó en cuenta para la selección dentro de este proyecto es que el broker de comunicación se puede instalar casi en cualquier plataforma de hardware y un punto muy importante es su eficiencia ya que no necesita de grandes cantidades de memoria o velocidad de procesamiento.

2.4.2. Procesamiento

Para la primera capa de procesamiento en donde se encuentran los dispositivos IoT del sistema, fue seleccionado el microcontrolador ESP8266 específicamente en la placa de desarrollo NodeMCU, esta placa está basada en Arduino, es compatible con la tecnología de comunicación seleccionada, es muy económica, fiable y comercial, actualmente es una de las placas más usadas para proyectos de internet de las cosas, al ser de código de abierto posee una gran comunidad que constantemente desarrolla esta tecnología, por su bajo costo y disponibilidad es muy fácil de reemplazar lo que asegura que el sistema pueda ser mantenido en el tiempo.

La tecnología de procesamiento seleccionada a nivel de servidor Edge fue el microprocesador basado en arquitectura ARM, encontrado específicamente en el dispositivo Tanix TX6, este es uno de los elementos claves del sistema, en características muy similar al SBC Raspberry Pi 4, pero dado que su función original no está orientada al desarrollo su precio representa una fracción de lo que cuestan la mayoría de SBC, se la conoce comúnmente como tv box y su uso está enfocado al entretenimiento, es un dispositivo que está diseñado para trabajo continuo con una vida útil de hasta 5 años, posee una carcasa que lo protege y ofrece buena disipación de calor, es muy fácil de conseguir en cualquier parte del mundo, viene con Android instalado de fábrica por lo que su configuración como servidor se puede realizar en menos tiempo a comparación de otras placas.

Al ser de bajo costo, comercial y de rápida configuración hace que este dispositivo sea muy fácil de reemplazar, de esta manera se asegura la viabilidad del proyecto en el tiempo y la escalabilidad del sistema por medio de la creación de múltiples servidores Edge según las necesidades del entorno.

Es importante recalcar que actualmente existe una variada oferta de sistemas operativos para servidores Edge, pero muy pocos son de uso gratuito y requieren de un cierto nivel de conocimientos para ser instalados, aumentando los costos y tiempo en el despliegue de la solución, al utilizar el sistema operativo Android con el que cuenta el servidor Edge

seleccionado optimizamos este proceso, ya que al estar basado en Linux podemos instalar casi cualquier aplicación de servidor además de las disponibles en las tiendas de aplicaciones.

La plataforma Edge seleccionada para el proyecto es Node-Red, en comparación al resto de opciones similares, esta es completamente gratuita, escalable y multiplataforma lo que le permite ser instalada en el servidor Edge sin ningún problema, esta plataforma no requiere de conocimientos de programación para orquestar la comunicación, procesamiento y desarrollo de interfaces de usuario por esta razón hace que el desarrollo sea más ágil y se puede llegar a una solución eficiente en muy poco tiempo, es de código abierto y posee una gran comunidad que constantemente desarrolla APIS para trabajar con otros sistemas, se puede usar tanto a nivel doméstico como a nivel industrial.

El firmware seleccionado para dar soporte lógico a los dispositivos IoT del sistema es proyecto de código abierto Tasmota, a comparación del resto de opciones es que más desarrollo ha tenido en los últimos años, esto ha permitido que por ejemplo la instalación en los dispositivos se la pueda realizar con muy pocos clics por medio de una aplicación web, reduciendo hasta en un 75% el tiempo de este proceso, posee una interfaz web muy amigable que permite una configuración rápida, este firmware nos permite instalar en lotes para posteriormente ser configurado para diferentes procesos o sensores. Gracias a su gran comunidad constantemente se mejoran las características como la seguridad y la eficiencia energética.

2.4.3. Analítica

La herramienta de analítica seleccionada fue Data Studio de Google, una vez más la principal razón de ser escogida es que a comparación del resto de opciones analizadas esta es completamente gratuita y muy fácil de usar, también ofrece la posibilidad de conectarse a diferentes fuentes de datos lo que asegura la escalabilidad del sistema.

2.4.4. Almacenamiento

A nivel de almacenamiento local los procesos son simples, el dispositivo IoT cuenta con una limitada cantidad de memoria efímera suficiente para realizar un muestreo simple y enviar la información del registro al servidor Edge.

El servidor Edge cuenta con una capacidad de almacenamiento de 16 GB expandible hasta 5 TB, por motivos de seguridad de la información y costos de hardware el servidor Edge se puede configurar para almacenar hasta un mes de datos y enviar esta información a la base de datos en la nube a diario.

Dada la facilidad que ofrece la herramienta Data Studio para usar diferentes fuentes de datos, la plataforma seleccionada como base de datos fue Google Drive, específicamente sus hojas

de cálculo, este sistema es muy intuitivo y no requiere de código para vincular la información enviada por los sensores a las herramientas de análisis, en su versión gratuita es muy completa y además de funcionar como base de datos el usuario puede hacer uso de otras aplicaciones como correo electrónico, editor de documentos, creación de páginas web, etc. Herramientas muy útiles para los primeros pasos en un proceso de transformación digital en un entorno donde los usuarios no están muy relacionados con la tecnología resulta muy fácil de usar y aprender su manejo.

2.4.5. Actuadores

Para este proyecto los actuadores serán bombas del tipo sumergible, estas ya se encuentran instaladas en el invernadero pero operan de manera manual, para operar estas bombas remotamente se ha seleccionado el dispositivo IoT Son off BASICR2, este dispositivo es uno de los más vendidos a nivel mundial para uso doméstico, pero por sus características y las del entorno de aplicación puede ser usado dentro del sistema Edge, solo es necesario realizar unas pequeñas modificaciones en el hardware para reemplazar su firmware original y al estar basado en el microcontrolador ESP8266 también es posible usar el firmware Tasmota, lo que lo convierte en un dispositivo muy versátil.

2.4.6. Adquisición de datos

En base a la investigación realizada sobre el cultivo y el tipo de sustrato, y la disponibilidad el sensor seleccionado para medir la humedad del suelo es el FC-28, el motivo principal fue su principio físico ya que actualmente dentro del cultivo experimental las mediciones se realizan con dispositivos analógicos de manera manual pero su principio de funcionamiento es el mismo, por esta razón es posible usarlo como patrón para la calibración del sensor del sistema propuesto.

El sensor de temperatura y humedad relativa seleccionado es el DHT21, este sensor cuenta con las mejores características respecto al uso que se le va a dar y las otras opciones disponibles en el mercado, tiene buena precisión y un empaque robusto características muy importantes dentro del sistema ya que este sensor está directamente expuesto a las condiciones ambientales del invernadero.

El sensor de CO2 seleccionado es el MHZ-19B, por sus especificaciones técnicas, precio y tamaño este sensor es la mejor opción, también basado en el principio de funcionamiento tiene muy buena precisión, cabe recalcar que es el único sensor de este tipo disponible en el mercado, por esta razón su funcionamiento deberá ser evaluado en las pruebas de concepto y la implementación del sistema.

En todos los puntos los principales criterios de selección han sido el costo, disponibilidad, facilidad de uso y escalabilidad, estos serán claves durante el desarrollo de todo el proyecto.

3. Descripción general de la contribución del TFM

3.1. Objetivos

Objetivo general

Implementar un sistema Edge Computing para el manejo de cultivo de Fresa (Fragaria) bajo Invernadero en los invernaderos del Centro Experimental de Riego de la Escuela Superior politécnica de Chimborazo.

Objetivos específicos

- Determinar las variables de control para el manejo de cultivos de fresa dentro de invernaderos.
- Seleccionar las tecnologías para la creación de un sistema Edge Computing.
- Construir un prototipo de sistema Edge Computing.
- Instalar el sistema en un cultivo de fresa bajo invernadero.
- Diseñar una interfaz de usuario que facilite el control y monitoreo de las variables identificadas en el manejo de cultivos de fresa bajo invernadero.

3.2. Metodología del trabajo

Para el desarrollo del presente proyecto se utilizarán dos metodologías, POC.y SCRUM.

Con la metodología POC se desarrollará un prototipo de prueba antes de empezar con el desarrollo del producto final, con ello se identifica la viabilidad del producto, los errores y problemas que se presenten en la ejecución del prototipo, los mismo que serán tomados en cuenta para el desarrollo del producto final, reduciendo los tiempos de entrega y coste de sensores que no serán necesarios para el desarrollo.

La metodología SCRUM es utilizada para desarrollar el software en trabajo colaborativo de equipo y obtener mejores resultados, identificando los objetivos a cumplir, los plazos de entrega de la aplicación y las reuniones con el equipo SCRUM dentro del avance del software.

3.2.1. POC

Las pruebas de concepto fueron desarrolladas en 5 etapas. Las cuales son descritas a continuación:

- Preparación y configuración. En esta fase de conectaran y configurarán cada uno de los dispositivos seleccionados, de acuerdo con la arquitectura planteada y la investigación realizada para configurar cada uno de los elementos físicos, en esta fase

también se configurará el software, de tal manera que el sistema sea una versión lo más cercana posible a la versión final.

- Puesta en marcha. El tiempo de construcción de la prueba de concepto del sistema será aproximadamente de dos semanas, en la primera semana se configurarán todos los dispositivos tanto a nivel de hardware como de software. La puesta en marcha de la prueba se llevará a cabo fuera del entorno de implementación. En la segunda semana se probará el funcionamiento del sistema y el comportamiento de los flujos de información planteados.
- Ejecución de pruebas. En esta etapa se verifica que los datos recolectados por los diferentes sensores sean coherentes y correctos, además se evaluará uso y manejo de la interfaz de usuario, así como la estabilidad del sistema y la respuesta ante eventualidades como un corte de energía.
- Evaluación de resultados. Con la finalización de la etapa de pruebas, se analizará los problemas y errores que surgieron durante la ejecución del prototipo con el objetivo de implementar mejoras al software, las mismas que serán analizadas y en caso de ser viables se incrementarán en el producto final.
- Conclusiones. Una vez finalizada la etapa de pruebas y corrección de errores, identificando las posibles mejoras para el proyecto las cuales serán validadas en un tiempo definido para comprobar el correcto funcionamiento del proyecto, se dará por concluida la prueba de concepto. En caso de no existir ninguna novedad y el resultado sea el esperado, se continuará con el desarrollo del proyecto final.

3.2.1. SCRUM

Para el desarrollo de la aplicación se utilizó la metodología SCRUM, donde se dividió la elaboración del software por Sprints, los cuales tienen una duración de 2 a 4 semanas según las tareas que se van a desarrollar. El equipo SCRUM que participa en esta metodología son, el Product Owner, Scrum Master y los Developers. En el desarrollo de este trabajo, el Product Owner sería la universidad UNIR, siendo la propietaria de los derechos de autoría. El Scrum Master es la persona que sigue las instrucciones del Product Owner y supervisara el trabajo que realizan los desarrolladores, siendo una figura intermediaria entre las dos partes, en este caso es el Ing. Juan Sebastián Silva Orozco, encargado del experimento de evaluación de variedad de fresas en sistema de producción semihidroponica, fenología y requerimientos hídricos. Finalmente, el equipo de Developers está conformado por una sola persona, que es el Ing. Cristian García Pumagualle, autor del proyecto.

Cada vez que se inicie un Sprint se realiza un sprint Planning meeting, donde se especifica los objetivos que se realizaran durante la ejecución de sprint. El objetivo de esta reunión

desarrollar un documento donde se especifique las tareas que están por realizarse (Backlog), por otra parte, el Scrum master y el Product Owner decidirán cuales son las tareas que tienen más prioridad, dando un tiempo de estimación para el desarrollo de cada una de ellas. De esta forma se planifica el desarrollo según la duración de cada sprint y la velocidad del desarrollador.

Deberá existir una reunión diaria (daily Scrum) entre el desarrollador y el Scrum master, donde se tratará los problemas encontrados en el desarrollo de cada tarea, en este caso esta reunión no será posible, puesto que es un TFM virtual. Así también, con la finalización de cada Sprint, se realizará una reunión entre el Scrum master y el Product Owner, para informar cómo ha ido el desarrollo del sprint y si se ha culminado con todas las tareas planificadas. Una vez finalizada esta reunión el Scrum master realiza un sprint retrospective meeting con el desarrollador para planificar el tiempo de entrega de las siguientes tareas y comentar lo que se habló en la reunión con el Product Owner. Cabe mencionar que la velocidad del equipo de desarrollo y el tiempo de ejecución de cada tarea ira mejorando cada vez más, puesto que, en la primera reunión no se tiene muy claro todas las ideas para el desarrollo de la aplicación.

Para el desarrollo de este proyecto, la duración de cada sprint será de un mes aproximadamente.

Sprint 1: Definición 01/06/2022 – 30/06/2022

En esta etapa analizamos los requerimientos del sistema, en base a la investigación geográfica con respecto al manejo del cultivo de fresa y la tecnología disponible en el mercado para construir el sistema.

Sprint 2: POC 01/07/2022 – 31/07/2022

El objetivo de este sprint es empezar con las pruebas de concepto (POC) de la aplicación para identificar la viabilidad de la aplicación, los errores y las posibles mejoras que se le puede implementar en el producto final, de igual forma se modificara las observaciones que el Scrum máster mencione en la reunión.

Sprint 3: Implementación 01/08/2022 – 14/08/2022

El objetivo del tercer sprint es empezar a desarrollar el producto final. Como también se realizará las instrucciones dadas por el Scrum master en la reunión.

Sprint 4: Pruebas 15/08/2022 – 11/09/2022

Finalmente, con el desarrollo de este sprint se tiene como objetivo realizar las respectivas pruebas de la recolección de los datos mediante los diferentes sensores colocados en el

cultivo y que la plataforma de Edge muestre información que facilite el manejo del cultivo al encargado de este.

3.3. Descripción general de las partes o componentes de la propuesta

Esta propuesta consiste en la implementación de un sistema Edge Computing para el manejo de cultivo de Fresa (Fragaria) bajo Invernadero, que permita al agricultor monitorear sus cultivos, recolectando información real mediante los diferentes sensores de humedad del suelo, temperatura, humedad relativa y CO₂, mostrando estos datos por medio de una interfaz gráfica que sea fácil de usar y que se integre con herramientas de análisis.

Gracias al desarrollo de este proyecto los agricultores podrán disminuir sus pérdidas económicas, prevenir enfermedades y aumentar el rendimiento ya que podrán monitorear sus cultivos desde cualquier parte y en cualquier momento, facilitando el manejo y control de los parámetros establecidos para el cultivo de fresa bajo invernadero.

Alcance y limitaciones

El alcance de este trabajo es desarrollar un sistema basado en arquitectura Edge Computing para el manejo de cultivo de Fresa (Fragaria) bajo Invernadero. Desde la adquisición y configuración de cada uno de los componentes del sistema hasta la instalación de estos dentro del centro experimental de riego de la Escuela Superior Politécnica de Chimborazo.

Una de las limitaciones es el presupuesto para el desarrollo de la propuesta por lo que durante la selección se ha buscado optimizar la adquisición de cada uno de los componentes, otro aspecto limitante es la disponibilidad de la tecnología, se ha procurado construir la propuesta en base a dispositivos con una alta disponibilidad.

Listado de participantes

Para el desarrollo del proyecto, se trabajó en conjunto con el Centro Nacional Experimental del Riego de la Escuela Superior Politécnica de Chimborazo, específicamente en el área experimental de “Evaluación de variedades de fresa en sistema de producción semihidropónica, fenología y requerimientos hídricos”, bajo de la línea de investigación de alternativas sustentables de diversificación de la producción agrícola.

Tecnologías implicadas

Las tecnologías implicadas en este trabajo son:

- **Robótica:** Se utilizan microprocesadores con arquitectura de procesador ARM para el servidor Edge y placas de desarrollo basadas en Arduino para gestionar la recolección de los datos mediante los sensores implementados en los cultivos.

- **Sensores IoT:** Se construirá dispositivos IoT basados en las placas de desarrollo NodeMCU y el microcontrolador ESP8266, por medio de estos se realizará la adquisición de información de los sensores y el comando remoto de los actuadores.
- **Internet of things (IoT):** Se hará uso de la plataforma IoT Node-Red adaptada a la arquitectura Edge Computing por medio de su instalación en servidor local para la conexión entre los sensores, y un ordenador/móvil donde por medio de una aplicación web el agricultor visualizará la información de sus cultivos desde cualquier parte y en cualquier momento.
- **Ciberseguridad:** se han considerado estándares y medidas de seguridad en cada uno de los componentes de la arquitectura para garantizar el funcionamiento del sistema y la integridad de los datos.
- **Impresión 3D:** esta tecnología puede ser usada en versiones futuras para mejorar el diseño y funcionalidad de las carcasas de cada uno de los dispositivos IoT.
- **Big data:** se utilizarán herramientas de visualización de la información como Data Estudio, con la visión de que a futuro se puedan analizar los datos generados por los sensores de una manera ágil y eficiente.
- **Inteligencia artificial:** gracias a la arquitectura planteada y un cierto volumen de datos temporales, en una versión futura por medio de estos sería posible entrenar modelos de aprendizaje para prevenir plagas y enfermedades en los cultivos de manera oportuna.

Arquitectura, componentes e integración de tecnologías

La capa IoT está conformada por los sensores y actuadores, así como los módulos WIFI. Por medio de los protocolos MQTT y HTTP, la capa IoT se comunica con la capa Edge, que está conformada por la plataforma Node-RED, ngrok, mosquito y AnyDesk integradas en un dispositivo Android basado en arquitectura ARM. En la capa de solución empresarial encontramos las aplicaciones de analítica, base de datos y servicios provistos por el entorno de Google Drive, al final encontramos como capa transversal a la seguridad que está basada en medidas orientadas a la ciberseguridad para proteger los datos y la integridad del sistema.



Figura 6: Arquitectura de la aplicación. (Elaboración propia)

Resultados esperados

Los resultados esperados para el presente trabajo es desarrollar un sistema Edge Computing que facilite el manejo de cultivo de Fresa (Fragaria) bajo Invernadero, por medio del monitoreo remoto y oportuno de las variables meteorológicas y el control del riego en base a parámetros y horarios configurables desde una interfaz gráfica de usuario.

Presupuesto y retorno esperado de la inversión

Al ser una propuesta de innovación tecnológica para un proyecto de investigación sin fines de lucro se fijaron valores en base a la bibliografía para así definir los elementos necesarios en el cálculo del retorno de la inversión.

Para definir el presupuesto se asociará el costo de cada uno de los componentes del sistema en base a los precios actuales del mercado con el costo de la configuración y construcción

del sistema en base a las horas de trabajo y la dificultad, otro rubro de este presupuesto será la instalación.

Tabla 2: Detalle de los valores unitarios de los componentes del sistema

ITEM	DESCRIPCIÓN	CANTIDAD	UNIDAD	VAL.UNIT	SUB.TOT
Módulo IoT Edge Computer					
1	Caja de PVC IP 68	1	uds	€ 10,00	€ 10,00
2	SBC Tx6 2GB RAM	1	uds	€ 35,00	€ 35,00
3	Prensa Estopa Plástica Para Conductor Eléctrico	3	uds	€ 0,40	€ 1,20
SUB.TOT1					€ 46,20
Módulo Sensor CO2					
1	Caja Impermeable PVC - IP68	1	uds	€ 10,00	€ 10,00
2	Esp8266 Cp2102 V2 Nodemcu Lua	1	uds	€ 10,00	€ 10,00
3	Sensor de CO2 MH-Z19B	1	uds	€ 50,00	€ 50,00
4	Fuente de alimentación	1	uds	€ 10,00	€ 10,00
5	Cables	1	glb	€ 5,00	€ 5,00
SUB.TOT2					€ 85,00
Módulo Sensor IoT Humedad del Suelo					
1	Caja Impermeable PVC - IP68	1	uds	€ 10,00	€ 10,00
2	Esp8266 Cp2102 V2 Nodemcu Lua	1	uds	€ 10,00	€ 10,00
3	Sensor de humedad de suelo Anticorrosivo - Higrómetro LM393	1	uds	€ 15,00	€ 15,00
4	Fuente de alimentación	1	uds	€ 10,00	€ 10,00
5	Cables	1	glb	€ 5,00	€ 5,00
SUB.TOT2					€ 50,00
Módulo Sensor IoT Temperatura y Humedad Relativa					
1	Caja Impermeable PVC - IP68	1	uds	€ 10,00	€ 10,00
2	Esp8266 Cp2102 V2 Nodemcu Lua	1	uds	€ 10,00	€ 10,00
3	Dht21 Sensor Temperatura Humedad AM2301	1	uds	€ 10,00	€ 10,00
4	Fuente de alimentación	1	uds	€ 10,00	€ 10,00
5	Cables	1	uds	€ 5,00	€ 5,00
SUB.TOT3					€ 45,00
Módulo actuador IoT					
1	Caja Impermeable PVC - IP68	1	uds	€ 10,00	€ 10,00
2	Son off BasicR2 esp8266 relé	1	uds	€ 24,00	€ 24,00
3	Botón pulsador verde	1	uds	€ 3,00	€ 3,00

4 Cables	1	glob	€ 5,00	€ 5,00
			SUB.TOT4	€ 42,00
Red Inalámbrica				
Radioenlace Tp-link Cpe220 De Exterior	1	uds	€ 55,00	€ 55,00
1 2.4ghz 300mbps 12dbi	1	uds	€ 55,00	€ 55,00
Radioenlace Tp-link Cpe510 De Exterior 5ghz	2	uds	€ 65,00	€ 130,00
2 300mbps 13dbi	2	uds	€ 65,00	€ 130,00
Conectores RJ45 con capuchón	5	uds	€ 0,50	€ 2,50
3	5	uds	€ 0,50	€ 2,50
4 CABLE UTP	10	M	€ 0,35	€ 3,50
			SUB.TOT4	€ 191,00
			TOTAL	€ 374,20

A continuación, un detalle del presupuesto total de la inversión:

Tabla 3: Presupuesto total de la inversión

ITEM	DESCRIPCIÓN	CANTIDAD	UNIDAD	VAL.UNIT	SUB.TOT
Costo del sistema Edge Computing para agricultura inteligente					
1	Componentes físicos del sistema	1	uds	€ 374,20	€ 374,20
2	Configuración y desarrollo del sistema	1	uds	€ 4.000,00	€ 4.000,00
3	Instalación y puesta en marcha del sistema	1	uds	€ 500,00	€ 500,00
TOTAL					€ 4.874,20

Tomando en cuenta que el valor comercial actual de la fresa es de 160 mil euros por hectárea y que el rendimiento del cultivo con manejo tradicional es de 40 toneladas por hectárea, también que según la investigación con la implementación del sistema de ege Computing el rendimiento puede aumentar de entre un 13% a 15%, se pueden definir los siguientes valores para el cálculo del retorno de la inversión

Tabla 4: Retorno de la inversión

Retorno de la inversión (ROI = (Ingreso – Inversión) / inversión)	
Detalle	Valor
Ingreso por aumento del 14 % en el rendimiento del cultivo	€ 22.400,00
Inversión total para el sistema Edge Computing	€ 4.874,20

ROI	3,6 %
------------	--------------

En base al análisis del retorno de la inversión el proyecto es viable desde el punto de vista financiero.

Planificación general

El tiempo estimado de las tareas se definió en la metodología Scrum.

- Sprint 1: Definición 01/06/2022 – 30/06/2022
- Sprint 2: POC 01/07/2022 – 31/07/2022
- Sprint 3: Implementación 01/08/2022 – 14/08/2022
- Sprint 4: Pruebas 15/08/2022 – 11/09/2022
- Fecha límite de entrega 14/09/2022

4. Desarrollo específico de la contribución

4.1.1. Sprint 1: Definición

En esta etapa analizamos los requerimientos del sistema, en base a la investigación geográfica con respecto al manejo del cultivo de fresa y la tecnología disponible en el mercado para construir el sistema.

4.1.1.1. Características del sistema

Basados en la información investigada, las instalaciones del invernadero y su cultivo, las características con las que debería contar el sistema son las siguientes.

- Por la parte del hardware deberá ser modular con el fin de facilitar la instalación y el mantenimiento.
- Cada módulo debe contar con las protecciones adecuadas para que los componentes electrónicos se encuentren protegidos de las condiciones climáticas.
- Los registros de los sensores deben ser automáticos, esto quiere decir que en el momento que se conecta el sensor a la fuente de energía empiece a enviar datos al sistema.
- Se debe considerar que el suministro eléctrico no es continuo, existe la posibilidad de que el sistema pierda alimentación y debe ser capaz de reportarlo.
- El sistema debe ser capaz de arrancar automáticamente en caso de un corte de energía.
- Los sensores deben configurarse de tal manera que sea posible calibrarlos remotamente.
- El sistema debe alertar al usuario en caso de que algún sensor falle o pierda contacto con el sistema.
- La interfaz de usuario debe ser de fácil manejo y resumir la información de todos los sensores.
- Es necesario tener acceso remoto al sistema, para el control y monitoreo.
- Deberá contar con una interfaz capaz de almacenar datos históricos y generar reportes.
- Cada uno de los componentes debe ser comercial y contar con stock de repuestos, en caso de una avería debe ser reemplazado fácilmente.

4.1.1.2. Infraestructura disponible

Antes de la implementación del sistema identificamos el entorno del invernadero y partimos de las instalaciones disponibles para diseñar el sistema y la estrategia de instalación, a continuación, se detallan las características del entorno.

- El invernadero donde se implementará el sistema cuenta con instalaciones eléctricas donde cada una de las bombas de agua se encuentran conectadas de manera directa a un tomacorriente, las mismas están sumergidas en tanques como se muestra en la figura 7 y el accionamiento es manual.



Figura 7: Instalaciones del invernadero de fresa (Elaboración propia)

- La energía eléctrica utilizada por el sistema de riego por goteo es suministrada por un panel fotovoltaico ubicado en el exterior a continuación del invernadero, como se muestra en la figura 8.



Figura 8: Panel fotovoltaico (Elaboración propia)

- El suministro eléctrico se encuentra configurado como se muestra en la figura 9, la misma detalla sus características básicas y cada uno de sus componentes.



Figura 9: Configuración del panel fotovoltaico (Elaboración propia)

- Como se muestra en la figura 10, a 91 metros del invernadero de fresa se encuentra el edificio principal del centro experimental, el mismo que cuenta con acceso a internet para sus oficinas, dato importante para la instalación de la red inalámbrica.



Figura 10: Vista satelital del centro experimental (Elaboración propia)

4.1.1.3. Selección de variables a monitorear

Basados en las recomendaciones de los manuales que hablan sobre el manejo de cultivo de fresa bajo invernadero, las variables de monitoreo seleccionadas son las siguientes.

- Dióxido de carbono
- Humedad del suelo
- Temperatura, Humedad relativa y punto del rocío

4.1.1.4. Tecnologías

- **Comunicación**

Basados en las conclusiones del apartado 2 y las características de la infraestructura, la tecnología inalámbrica a utilizar será WIFI en frecuencias de 2.4 GHz y 5 GHz, para la frecuencia de 2.4 GHz se utilizará la antena CPE210 de la marca tp-link y para la frecuencia de 5 GHz el modelo CPE510 en la misma marca.



Figura 11: Antena tp-link CPE210 y CPE510 Fuente: tp-link, s. f.

Por su ligereza y facilidad de implementar en cualquier dispositivo, el protocolo de comunicación a utilizar para intercambiar mensajes entre los dispositivos IoT del sistema será MQTT, para ejecutar este protocolo de comunicación se optó por el bróker de código abierto Eclipse Mosquitto.

Para enviar mensajes desde el dispositivo de Edge Computing hasta la base de datos en la nube el protocolo a utilizar será HTTP, mientras que para la comunicación tanto en red local como por internet el protocolo de comunicación a utilizar será TCP/IP.

- **Procesamiento**

La SBC por utilizar será una Tanix modelo TX6, dado que su uso comercial está enfocado a las aplicaciones de entretenimiento y streaming su manufactura ha sido masiva y esto ha permitido que sus costos de producción sean bajos convirtiéndola en la SBC más económica del mercado con relación a sus características, con un precio que va desde los 10 dólares por unidad. Dentro del sistema propuesto este dispositivo cumplirá el rol de computadora al borde la nube y bróker MQTT, siendo así el más importante del mismo.

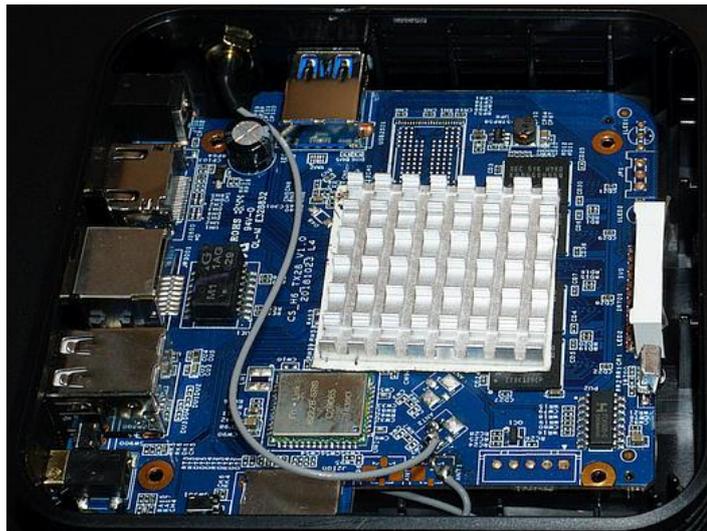


Figura 12: SBC Tanix TX6 (Elaboración propia)

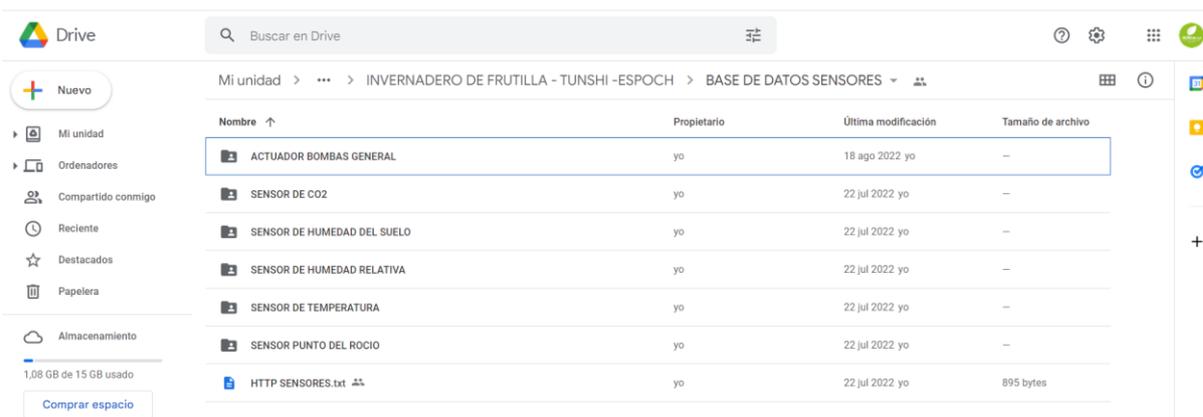
Node-Red será la plataforma de Edge Computing se encargará de la gestión y procesamiento de datos en tiempo real, a su vez cuenta con las herramientas necesarias para desarrollar la interfaz de usuario, esta framework de código abierto es muy ligero, lo que le permitirá ejecutarse sin problemas en la SCB seleccionada, además es la única plataforma que permite ser ejecutada en dispositivos basados en sistema operativo Android.

- **Analítica**

Para la analítica se utilizará Google Data Studio, es muy visual, fácil de usar y completamente gratuita, esta herramienta nos permitirá crear informes de los datos obtenidos por los sensores, hacer reportes automáticos y visualizar métricas del cultivo.

- **Almacenamiento**

Para el almacenamiento se optó por Google Drive, este servicio de almacenamiento en la nube en su versión gratuita cuenta con 15 GB de memoria, cuenta con muchas características y aplicaciones para uso personal, es muy conocida y fácil de usar. Una de estas aplicaciones es Google Sheets la misma será usada como base de datos del sistema, es importante recalcar que esta aplicación es muy fácil de conectar a Google Data Studio como fuente de datos.



Nombre	Propietario	Última modificación	Tamaño de archivo
ACTUADOR BOMBAS GENERAL	yo	18 ago 2022 yo	–
SENSOR DE CO2	yo	22 jul 2022 yo	–
SENSOR DE HUMEDAD DEL SUELO	yo	22 jul 2022 yo	–
SENSOR DE HUMEDAD RELATIVA	yo	22 jul 2022 yo	–
SENSOR DE TEMPERATURA	yo	22 jul 2022 yo	–
SENSOR PUNTO DEL ROCIO	yo	22 jul 2022 yo	–
HTTP SENSORES.txt	yo	22 jul 2022 yo	895 bytes

Figura 13: Bases de datos de los dispositivos (Elaboración propia)

Estimando que un sensor envía datos cada hora, tendríamos 24 datos al día lo que significa 8760 datos al año, según Google Drive, s. f. actualmente una hoja de cálculo tiene un límite de diez millones de celdas, cada dato ocupa dos celdas por lo que tendríamos un consumo de 17520 celdas al año. Tomando en cuenta estas cifras podemos decir que en teoría podríamos almacenar hasta 570 años de datos por sensor, bajo este ejemplo esta cantidad de espacio nunca podría ser usada, pero considerando la vida útil de los equipos y los ciclos del cultivo de la fresa, el espacio de almacenamiento con el que contamos en las hojas de cálculo de Google es lo suficientemente grande para que el sistema almacene información sin problemas.

- **Actuador**

Para este módulo seleccionamos un actuador basado en el chip Wifi ESP82266, específicamente el Son off Basic R2 de la marca ITEAD, este modelo es uno de los primeros dispositivos IoT más vendidos del mundo, es muy económico y fiable, su chip inalámbrico trabaja en la frecuencia de 2.4 GHz, funciona con 110v y posee un relé integrado capaz de soportar cargas de hasta 10A.



Figura 14: Son off BasicR2. (ITEAD, s. f.)

- **Adquisición de datos**

Como dispositivo IoT se hará uso del módulo Esp8266 Cp2102 V2 Nodemcu Lua, algo muy importante para tener en cuenta es que este trabaja en frecuencias de 2,4 GHz, el soporte lógico que se encargará de gestionar la conexión, comunicación MQTT y configuración de los sensores será el firmware de código abierto Tasmota.

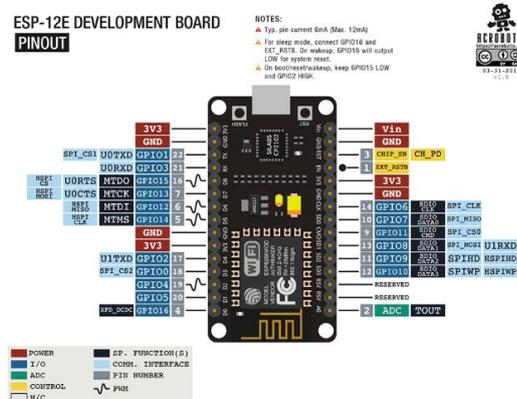


Figura 15: Esp8266 Cp2102 V2 Nodemcu Lua. (Naylamp, s. f.)

Basados en los criterios de disponibilidad, costos, fiabilidad, vida útil y tipo de variable fueron seleccionados los siguientes sensores:

Tabla 5: Variables y sensores

Variable	Nombre del sensor	Unidad de medida
Dióxido de carbono	MH-Z19B	0~5000 ppm
Humedad del suelo	Sensor de humedad de suelo FC-28	3.3V ~ 12V – 0 a 100% de Humedad
Temperatura, Humedad relativa	DHT21- AM	0 - 99.9 %RH -40 – 80 °C

4.1.2. Sprint 2: POC

4.1.2.1. Preparación y configuración

Para la prueba de concepto ensamblaremos cada uno de los componentes del sistema fuera de la estación experimental, la configuración y ensamble de cada uno de los componentes estará pensada para que trabaje de manera modular y facilite la instalación en el invernadero, A continuación, se detalla cada uno de los componentes necesarios:

Tabla 6: Hardware usado en la prueba de concepto.

Nombre	Tipo	Descripción	Cantidad
SBC Tanix TX6	Microprocesador	GPU: Mali-T720, Procesador: Allwinner H6 cuad core 1.5 GHz ARM Cortex A53 64 bits, RAM: 4 Gb, DDR3, SSD: 16 Gb, SO: Android 9.0, Antena WiFi 802,11n 2,4G/5GHz, 2.4G + 5G(AC2X2), Interfaces: Puerto DC, HDMI, RJ45, SPDIF, TF card, USB2.0, USB3.0, Funciones: DLNA, archivos ISO	1
Esp8266 Cp2102 V2 Nodemcu Lua	Microcontrolador	Voltaje de Alimentación: 5V DC, USB-serial: CP2102, SoC: ESP8266 (Espressif), CPU: Tensilica Xtensa LX3 (32 bit), Frecuencia de Reloj: 80MHz/160MHz, Instruction RAM: 32KB, Data RAM: 96KB, Memoria Flash Externa: 4MB, Pines Digitales GPIO: 17 (4 pueden configurarse como PWM a 3.3V), Pin Analógico ADC: 1 (0-1V), Puerto Serial UART: 2, Certificación FCC, Antena en PCB 802.11 b/g/n, Wi-Fi Direct (P2P), Stack de Protocolo TCP/IP integrado	3
Módulo sensor de CO2 MH-Z19B	Sensor	Precisión: 5% (50ppm), Principio: NDIR, Voltaje de alimentación: 5V DC, Interfaz de salida: UART / PWM, Nivel del interfaz: 3.3V (compatible con 5V), Corriente de consumo: <18mm, Tiempo de calentamiento: 3 min, Tiempo de respuesta: 90 < 60 s, Temperatura de trabajo: 0 ~ 50 °C	1
Sensor de humedad de suelo FC-28	Sensor	Sensibilidad ajustable ajustando el potenciómetro digital (en azul), Voltaje de operación: 3.3V ~ 12V, Modo de salida dual, salida digital y salida analógica más precisa., Agujeros de montaje para una fácil instalación, Dimensiones PCB: 30mm * 16mm, El módulo tiene un amplificador LM393, VCC (5V), GND, Interfaz de salida digital (0 y 1), Interfaz de salida analógica AO	1
Sensor de Temperatura y Humedad del	Sensor	Voltaje de Operación: 3.5V - 5.5V DC, Consumo corriente: 1mA - 1.5Ma, Rango de Temperatura: -40 hasta 80°C, Precisión Temperatura: +- 0.5°C, Resolución Temperatura: 0.1°C, Rango	1

suelo DHT21-AM2301			de Humedad Relativa: 0 a 100% RH, Precisión HR: +- 3%, Resolución Humedad: 0.1%RH, Tiempo de sensado: 2s, Interfaz digital: Single-bus (bidireccional)	
Son BASICR2	off	Actuador	Input: 100-240V AC 50/60Hz, OutPut 100-240V AC 50/60Hz, Max. Load: 10A/2200W, Wi-Fi: IEEE 802.11b/g/n 2.4GHz, Basado en ESP8266	1
Antena tp-link CPE 510		Red	Antena para exteriores de 5GHz 13dBi 2x2 MIMO direccional de doble polarizado, Potencia de transmisión ajustable de 0 a 23dBm / 200mw, Optimizaciones a nivel de sistema para la transmisión inalámbrica de más de 15 kilómetros de largo alcance.	2
Antena tp-link CPE 210		Red	Antena para exteriores de 2.4GHz 9dBi 2x2 MIMO direccional de doble polarizado, Potencia de transmisión ajustable de 0 a 27dBm / 500mw, Optimizaciones a nivel de sistema para la transmisión inalámbrica de más de 5 kilómetros de largo alcance,	1

La SBC Tanix TX6 es usada como servidor al borde y la NodeMCU como recolector de datos, el Son off BASICR2 será usado como actuador para el control de las bombas de agua para el riego. Cada uno de los sensores está conectado a una NodeMCU para medir los niveles de las diferentes variables. Estos datos serán enviados a la Tanix TX6 que actúa como servidor local y bróker MQTT.

- **Configuración de la red inalámbrica 5 GHz**

Ya que el invernadero está ubicado a 90 metros del edificio principal, usando la topología AP+CLIENTE creamos un enlace inalámbrico entre ambos y para esto se hará uso de las antenas CPE510.

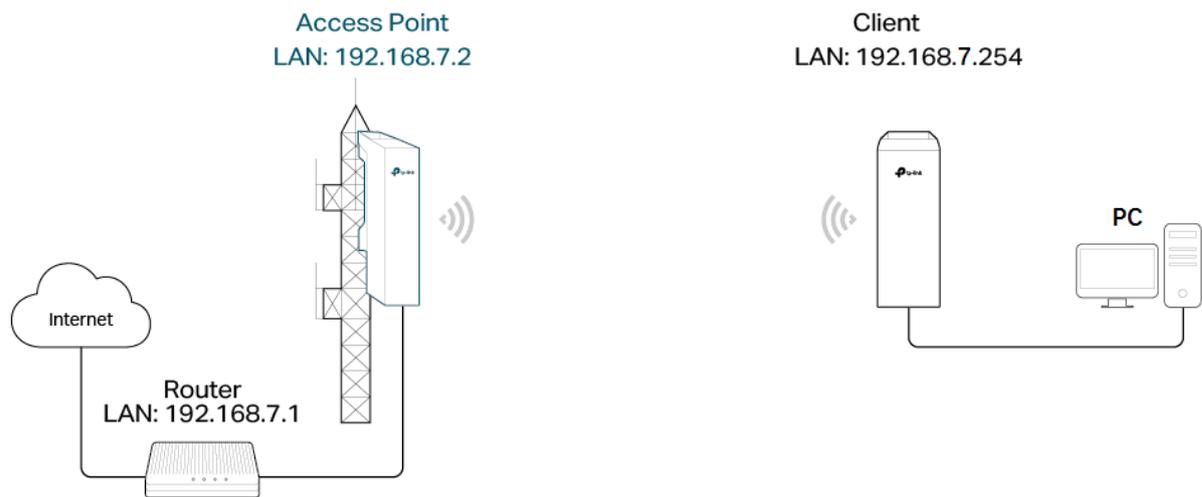


Figura 16: Escenario de red (CenturyLink, 2022)

Para la prueba de concepto esta configuración se realizará en un entorno cerrado fuera de la estación experimental y nos basaremos en la guía propuesta por Tp-link, (2020), para configurar la red según la topología de la figura 16.

El primer paso será conectar la antena CPE510 como punto de acceso a continuación, se muestra cómo se debe realizar la conexión eléctrica.

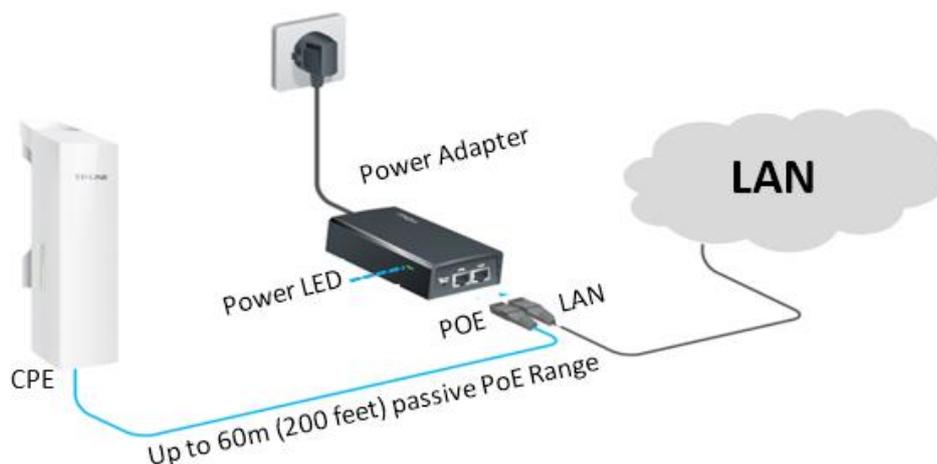


Figura 17: Conexión eléctrica antena AP (Siliceo, 2020)

De la misma manera conectaremos la antena CPE210, pero posteriormente ésta se configurará en modo cliente, quedando el puerto LAN disponible para conectar la antena CPE210, esta se creará una red inalámbrica WiFi sobre la superficie del invernadero que nos permitirá conectar el servidor Edge y cada uno de los dispositivos sensores y actuadores.

Para configurar las antenas tanto en modo cliente como AP, es necesario conectar el cable de red desde la computadora hasta el puerto LAN de la antena y configurar la tarjeta de red de la computadora con una ip estática, esto nos permitirá entrar la interfaz de configuración de la misma por medio de la ip por defecto de la antena, el resto de pasos para configurar las antenas en los diferentes modos es guiado por el sistema de administración PharOS, por medio de una interfaz amigable es posible configurar la red en pocos minutos, además se pueden encontrar muchas guías o tutoriales en internet, por esta razón no profundizaremos en este tema.

Es importante recalcar que la red inalámbrica en cada uno de sus nodos cuenta con medidas de seguridad para evitar que usuarios no autorizados realicen modificaciones.

- **Configuración del servidor Edge**

Como se menciona anteriormente en las características el dispositivo seleccionado para trabajar como servidor Edge ya cuenta con un sistema operativo lo que facilita la instalación de las aplicaciones necesarias para gestionar la configuración del servidor.

Como primer paso creamos una cuenta de en Google Drive, el proceso para crear esta cuenta es bastante simple por lo que no profundizaremos en el tema, como medida de seguridad es muy importante guardar las credenciales ya que estas también serán utilizadas para la configuración del entorno cloud y la capa de analítica del sistema.

Conectamos el Tanix Tx6 a una pantalla por medio del puerto HDMI, un teclado y un ratón, como se muestra en la siguiente figura.

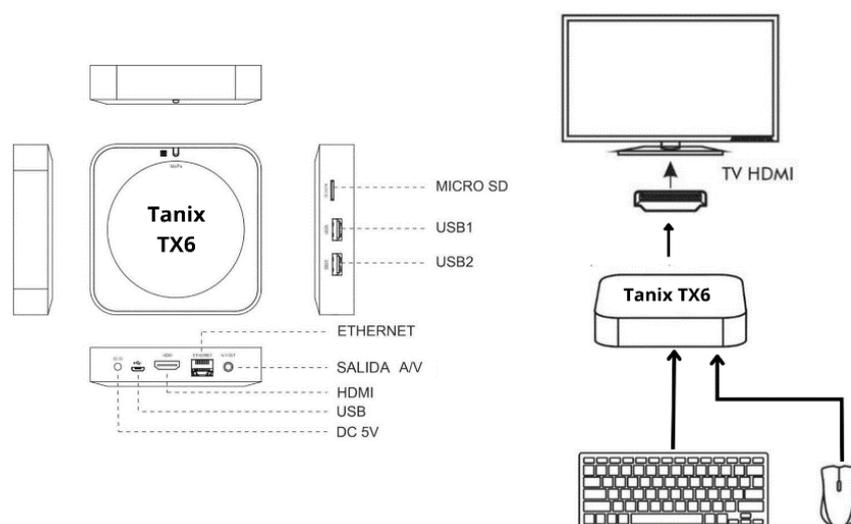


Figura 18: Esquema de conexión (Elaboración Propia)

Esta conexión facilitará el proceso de configuración y solo será usada durante las pruebas de concepto.

Una vez conectado todo, podremos ver la interfaz del dispositivo y realizamos las configuraciones necesarias para que se conecte al AP creado por la antena CPE210, después ingresamos a la tienda de aplicaciones de Android y realizamos la autenticación con las credenciales generadas anteriormente.

Desde la Play Store instalamos un lanzador con tema de Windows para facilitar la navegación con dispositivos periféricos.

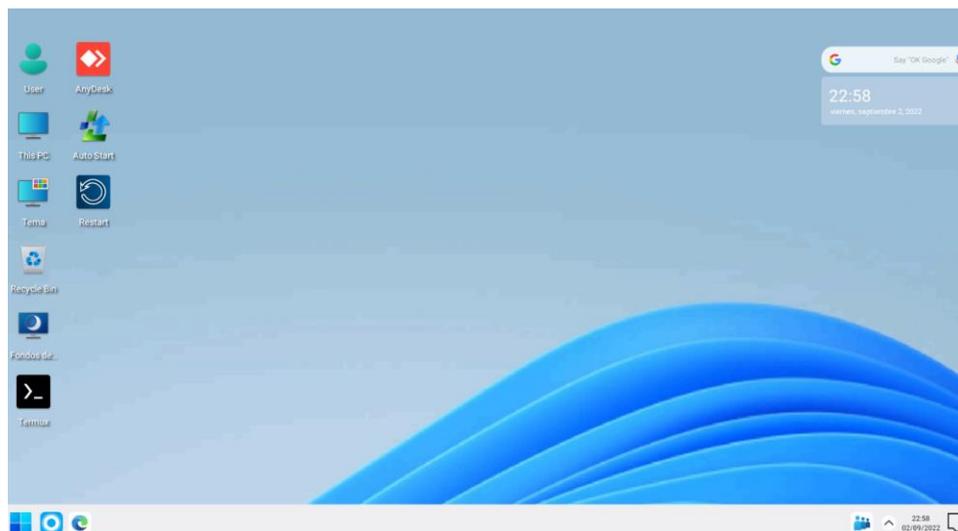


Figura 19: Interfaz de escritorio del servidor Edge (Elaboración Propia)

También desde la tienda de aplicaciones instalamos la aplicación Termux, este software es un emulador de terminal y entorno Linux, posteriormente nos permitirá instalar la plataforma Edge y el broker MQTT.

Instalamos AnyDesk, esta aplicación es un software de escritorio remoto que nos permitirá controlar el servidor Edge de manera instantánea con muy baja latencia, gracias a ella podremos trabajar sin necesidad de conectar dispositivos periféricos, además una vez instalado sistema por medio de esta aplicación tendremos acceso al servidor para realizar mantenimiento, monitoreo, actualizaciones, respaldos, solución de problemas, etc. Como medida de seguridad el acceso remoto al servidor también está protegido por credenciales generadas por el sistema, configurado de tal manera que solo los usuarios autorizados con nivel de administrador podrán ingresar.

Instalamos Auto Start como medida de seguridad, esta aplicación se encargará de arrancar automáticamente las aplicaciones Termux y AnyDesk, una herramienta muy útil en caso de

corte de energía, ya nos permitirá recuperar la conexión y el acceso remoto al servidor Edge automáticamente de inicializarlo manualmente.

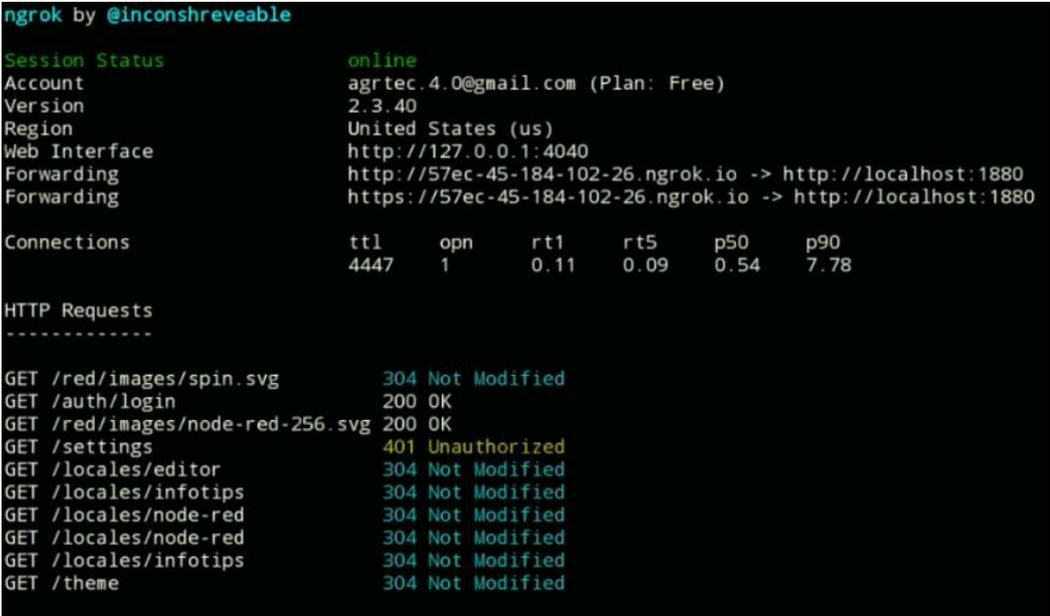
Una vez instaladas las aplicaciones mencionadas anteriormente procedemos a configurar la Tanix TX como servidor Edge, para esto entraremos a la consola Termux y ejecutaremos los siguientes comandos:

```
#Actualización de paquetes:
apt update
apt upgrade
apt list-installed
#Instalación del broker MQTT:
pkg install python coreutils nano clang mosquitto nodejs openssh termux-api
#Instalación de PM2 y Node-RED:
npm i -g --unsafe-perm node-red
npm i -g --unsafe-perm pm2
#Iniciamos las aplicaciones para verificar que la instalación fue correcta:
node-red
#Para detener: node-red with ctrl +c or volume down + c
mosquitto
#Para detener: Mosquitto with ctrl +c or volume down + c
#Iniciamos mosquitto desde PM2:
pm2 start mosquitto -- -v -c /data/data/com.termux/files/usr/etc/mosquitto/mosquitto.conf
#Modificamos las configuraciones de seguridad de Mosquitto:
nano /data/data/com.termux/files/usr/etc/mosquitto/mosquitto.conf
listener 1883
allow_anonymous true
pm2 restart mosquitto
#Iniciamos Node-red desde PM2:
pm2 start node-red --node-args="--max-old-space-size=128" -- -v
pm2 show "mosquitto"
pm2 logs "mosquitto"
pm2 save
pm2 monit
#Vemos la ip del servidor y guardamos:
Ifconfig
#Configuración de arranque automático:
```

```
AUTO RUN:  
nano ~/.bashrc  
pm2 resurrect  
pm2 monit  
ngrok http 1880  
#Reiniciar el servidor
```

Se consideró importante que como medida de seguridad se configure la autorización y autenticación en Node-red, evitando así que usuarios no autorizados accedan a la programación del sistema, este proceso se llevó a cabo siguiendo la guía de seguridad que se encuentra en la página oficial de Node-red.

Durante las pruebas se detectaron problemas en la instalación de Mosquitto, los problemas se basaban en la configuración del programa, en la secuencia de instalación mencionada anteriormente este problema se encuentra solventado. Una vez concluido el proceso de instalación del servidor Edge, se visualizará la siguiente figura:



```
ngrok by @inconshreveable  
  
Session Status      online  
Account             agrtec.4.0@gmail.com (Plan: Free)  
Version             2.3.40  
Region              United States (us)  
Web Interface       http://127.0.0.1:4040  
Forwarding          http://57ec-45-184-102-26.ngrok.io -> http://localhost:1880  
Forwarding          https://57ec-45-184-102-26.ngrok.io -> http://localhost:1880  
  
Connections         ttl      opn      rt1      rt5      p50      p90  
                   4447    1        0.11    0.09    0.54    7.78  
  
HTTP Requests  
-----  
GET /red/images/spin.svg      304 Not Modified  
GET /auth/login               200 OK  
GET /red/images/node-red-256.svg 200 OK  
GET /settings                 401 Unauthorized  
GET /locales/editor           304 Not Modified  
GET /locales/infotips         304 Not Modified  
GET /locales/node-red         304 Not Modified  
GET /locales/node-red         304 Not Modified  
GET /locales/infotips         304 Not Modified  
GET /theme                    304 Not Modified
```

Figura 20: Pantalla final de la instalación (Elaboración propia)

En la figura anterior se puede visualizar el resultado de la ejecución de ngrok, este servicio nos permite acceder de manera remota y segura a nuestro servidor Edge, por medio la URL que se muestra en esta pantalla podremos ingresar al entorno de desarrollo y a la aplicación web del sistema.

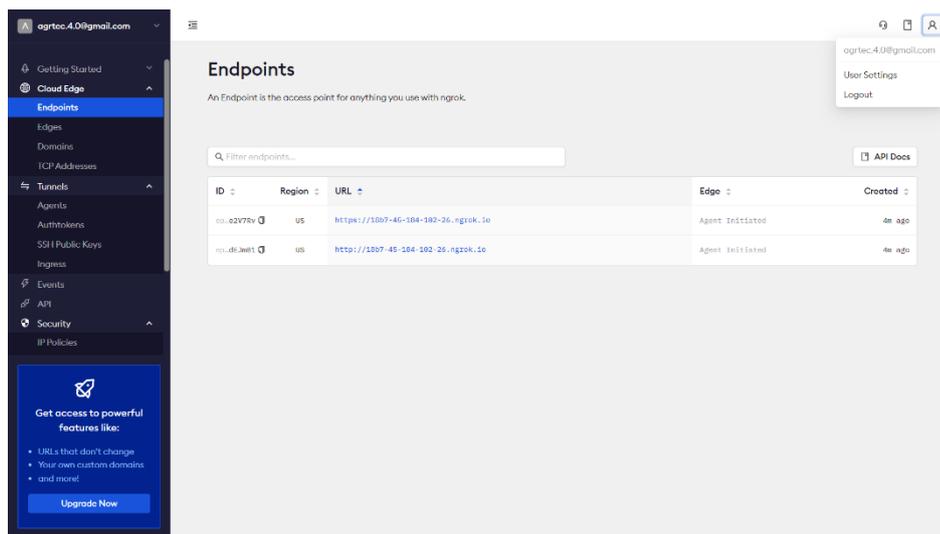


Figura 21: Dashboard ngrok. (Elaboración propia)

Una de las aplicaciones instaladas en las secuencias de comandos fue PM2 esta aplicación nos permitirá ejecutar Node-Red y el broker Mosquitto al mismo tiempo, a su vez esta aplicación nos permite monitorear remotamente el uso de recursos del servidor.

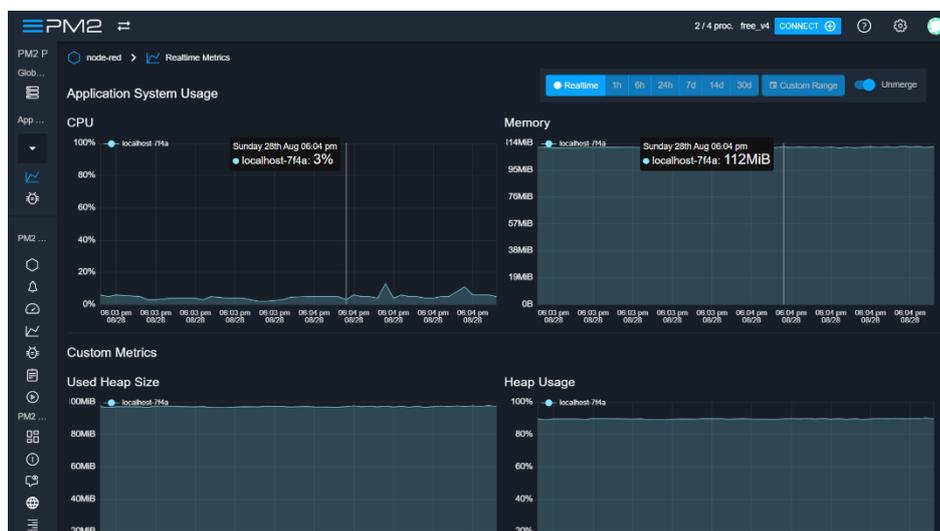


Figura 22: Dashboard de monitoreo PM2 estadísticas de Node-red. (Elaboración propia)

A continuación, en una computadora con conexión a internet ingresamos al navegador y escribimos la URL descrita anteriormente, des esta manera ya podemos ingresar a la interfaz de programación de la plataforma, como se muestra en la figura 20.

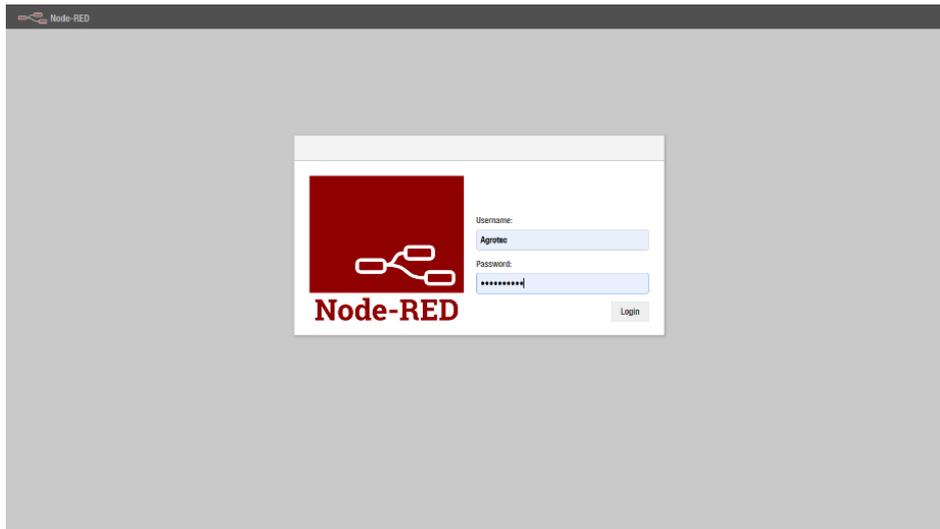


Figura 23: Interfaz de autenticación Node-red. (Elaboración propia)

El servidor Edge está listo para ser usado, pero antes de empezar a programar en el configuramos cada uno de los dispositivos IoT del sistema.

- **Instalación Tasmota en los microcontroladores NodeMCU.**

Una de las razones por la que se escogió usar un firmware de código abierto, es la optimización de tiempo en la configuración de los dispositivos ya que todo el proceso toma alrededor de 5 minutos, tomando en cuenta que todos los sensores utilizan la placa NodeMCU el proceso de instalación de Tasmota solo se detallará una vez.

Basados en la guía oficial de Tasmota, la misma que se puede encontrar en su página oficial, para instalar el firmware se deben seguir los siguientes pasos:

- 1.- Ingresar a la página oficial de instalación de Tasmota <https://tasmota.github.io/install/>
- 2.- Conectar la NodeMCU a la computadora usando un cable USB.
- 3.- De la lista desplegable seleccionar la versión de firmware llamada "Tasmota All Sensors".
- 4.- Dar clic en instalar y seleccionar el puerto al que se encuentra conectada la NodeMCU.
- 5.- En el caso de que los drivers para usar los puertos COM no estén instalados en la computadora, la interfaz de instalación dará una alerta y nos redirigirá a los enlaces para descargar los drivers.
- 6.- Cuando la instalación está completa la NodeMCU se reiniciará y creará una nueva red WIFI, nos conectamos a esta red desde una computadora o celular y automáticamente seremos redirigidos a la página de configuración de red.

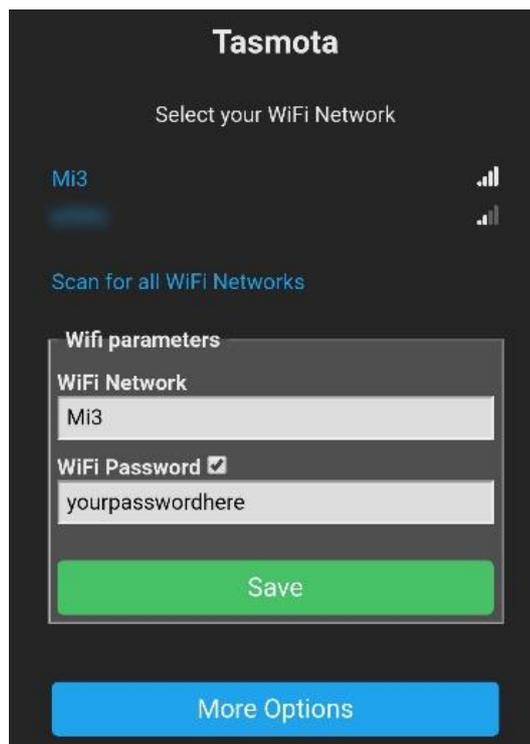


Figura 24: Interfaz de configuración de red Tasmota. (Tasmota, s. f.)

7.- Seleccionamos la red WiFi configurada anteriormente en la antena CPE210 y esperamos que la NodeMCU se conecte a la red.

8.- Automáticamente desaparecerá el punto de acceso temporal creado por la NodeMCU, y buscaremos la IP del dispositivo, para esto podemos utilizar la consola de Windows o la aplicación Fing desde dispositivos Android.

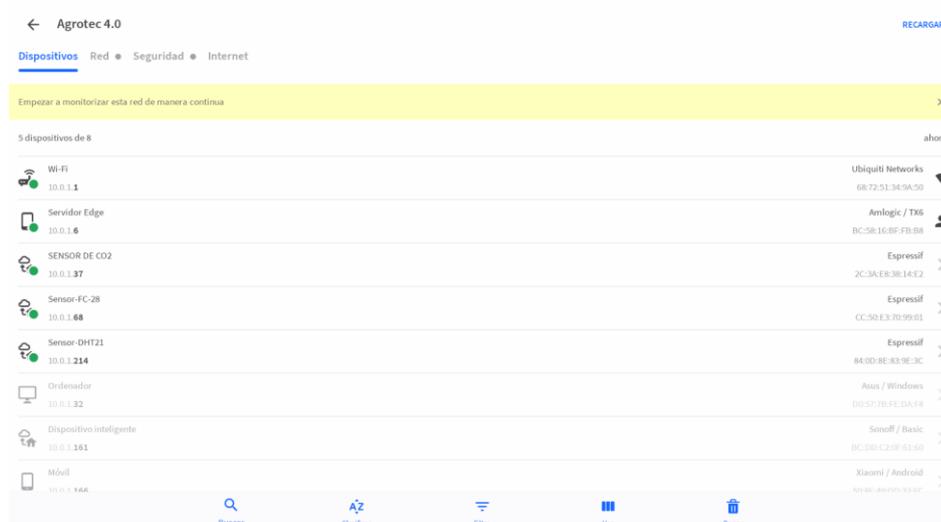


Figura 25: Administrador de red Fing. (Elaboración propia)

9.- Una vez identificada la IP del dispositivo, esta se abre desde un navegador que se encuentre dentro de la red local y tendremos acceso a la interfaz de configuración.

- **Instalación de Tasmota en el dispositivo Son off BASICR2**

Para instalar el firmware en este dispositivo es necesario realizar algunas modificaciones de hardware, para esto soldamos 5 pin headers en la PCB como se muestra en la figura 11.

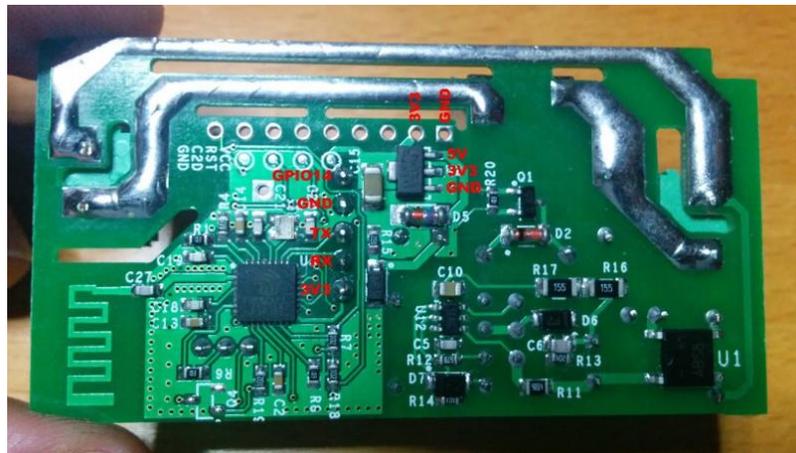


Figura 26: PCB Sonoff BASICR2. (Electrónica Joan, s. f.)

Ahora para transferir el firmware Tasmota desde la PC, conectamos el programador serial Ftdi Usb a Ttl 3.3v 5v Ft232rl con la PCB Son off BASICR2 como se muestra en la figura 24, una vez hecha la conexión se conecta el programador serial a la computadora por medio de un cable USB.

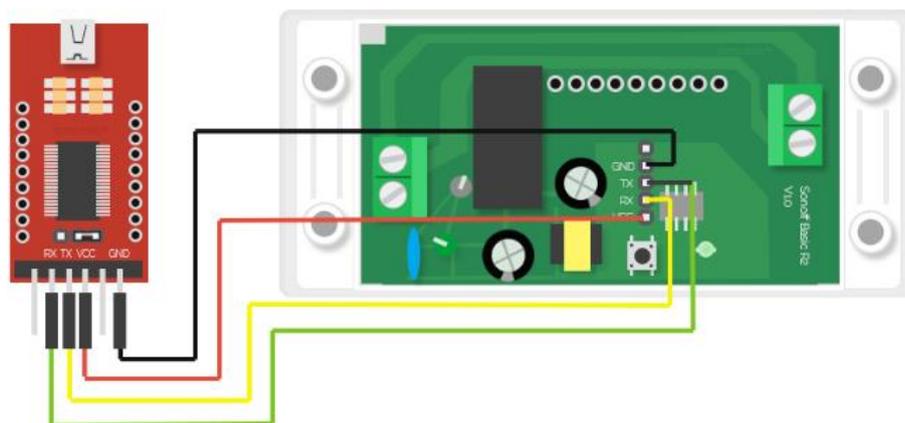


Figura 27: Conexión del adaptador serial con el Sonoff BASICR2. (Electrónica Joan, s. f.)

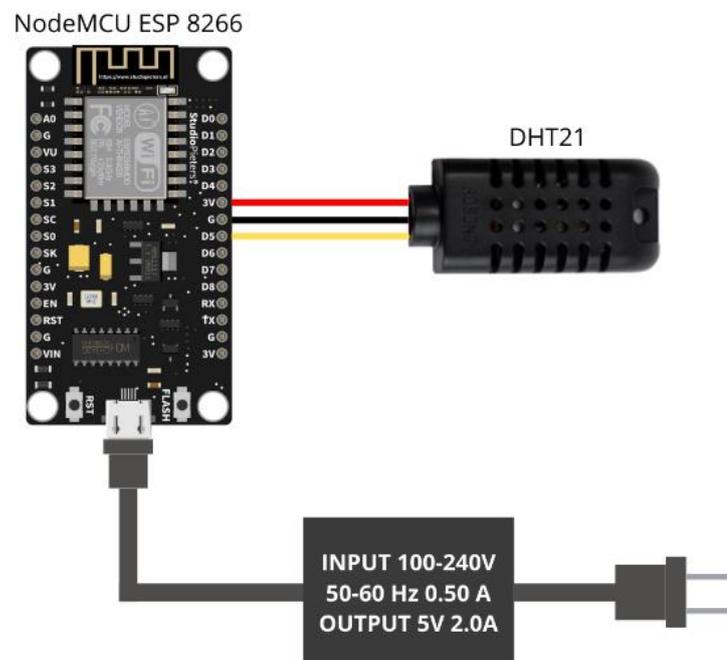
Es importante que la salida TX del adaptador se conecte con la entrada RX del Son off y la entrada RX del adaptador se conecte con la salida TX del dispositivo.

En este punto el proceso de instalación de Tasmota es el mismo que se detalló anteriormente, con la única diferencia de que al momento de escoger el firmware de la lista desplegable seleccionamos la versión “Tasmota (english)”

Una vez completados los procesos anteriores podremos configurar cada una de las NodeMCU como dispositivos IoT para que realicen la adquisición de datos desde cada uno de los sensores.

- **Módulo sensor de humedad relativa y temperatura**

Para construir el módulo de sensor de humedad relativa y temperatura, conectaremos el sensor DHT21 AM2301 con la NodeMCU y una fuente de energía como se muestra en la figura 28.



*Figura 28: Diagrama de conexión del módulo sensor de humedad relativa y temperatura.
(Elaboración propia)*

Una vez conectado a la fuente de energía el módulo se conectará automáticamente a la red previamente configurada, esto se puede verificar escaneando la red y viendo el nombre del dispositivo en la interfaz de fing, copiamos la dirección ip del dispositivo y la abrimos en un navegador para acceder a la interfaz de configuración.

En la interfaz de configuración de Tasmota, ingresamos en “Configuración”, y modificaremos la configuración del dispositivo, a continuación, se muestra los parámetros de configuración del módulo.



Figura 29: Parámetros del módulo sensor DHT21. (Elaboración propia)

En la siguiente figura se puede ver los parámetros de configuración del protocolo de comunicación:

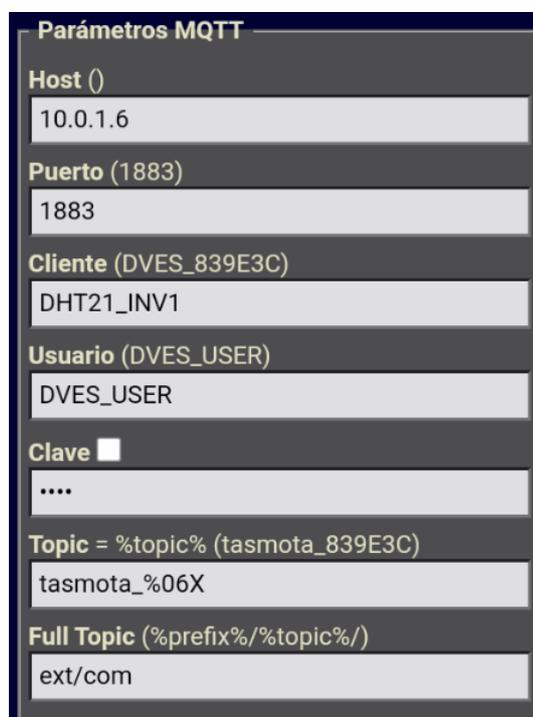


Figura 30: Parámetros MQTT. (Elaboración propia)

En la sección de otras configuraciones, es importante establecer los siguientes parámetros:



Figura 31: Configuración de otros parámetros. (Elaboración propia)

Este último es muy importante ya que en la sección de clave de administrador configuraremos una contraseña, esta contraseña será solicitada al momento de ingresar a la ip del dispositivo protegiendo así la configuración y evitando el acceso de usuarios no admitidos.

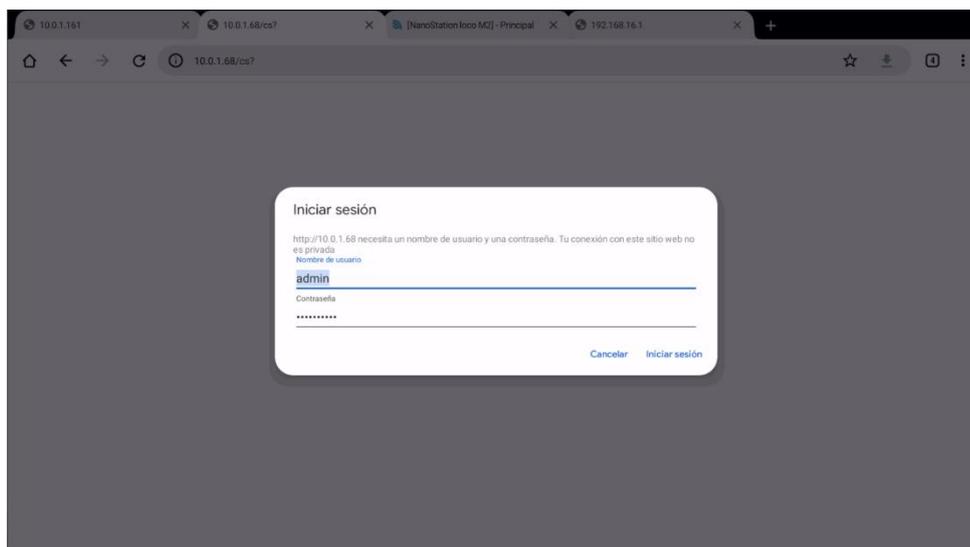


Figura 32: Autenticación para ingresar a la interfaz de configuración. (Elaboración propia)

- **Módulo sensor de humedad del suelo**

Para construir el módulo de sensor de humedad del suelo, conectaremos el sensor FC-28 con la NodeMCU y una fuente de energía como se muestra en la figura 32.

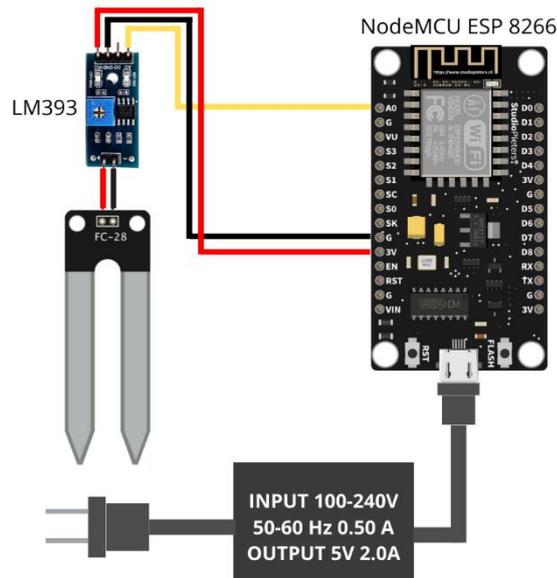


Figura 33: Diagrama de conexión del módulo sensor de humedad del suelo FC-28.
(Elaboración propia)

El proceso de configuración de este módulo es muy similar al del módulo sensor DHT21, la única diferencia serán los valores asignados al topic MQTT y la configuración de los parámetros del módulo, en este caso estos quedarían establecidos de la siguiente manera:

Parámetros del módulo	
Tipo de módulo (Sonoff Basic)	
Generic (18) ▾	
D3 GPIO0	Ninguno ▾
TX GPIO1	Ninguno ▾
D4 GPIO2	Ninguno ▾
RX GPIO3	Ninguno ▾
D2 GPIO4	Ninguno ▾
D1 GPIO5	Ninguno ▾
D6 GPIO12	Ninguno ▾
D7 GPIO13	Ninguno ▾
D5 GPIO14	Ninguno ▾
D8 GPIO15	Ninguno ▾
D0 GPIO16	Ninguno ▾
A0 GPIO17	ADC Rango ▾

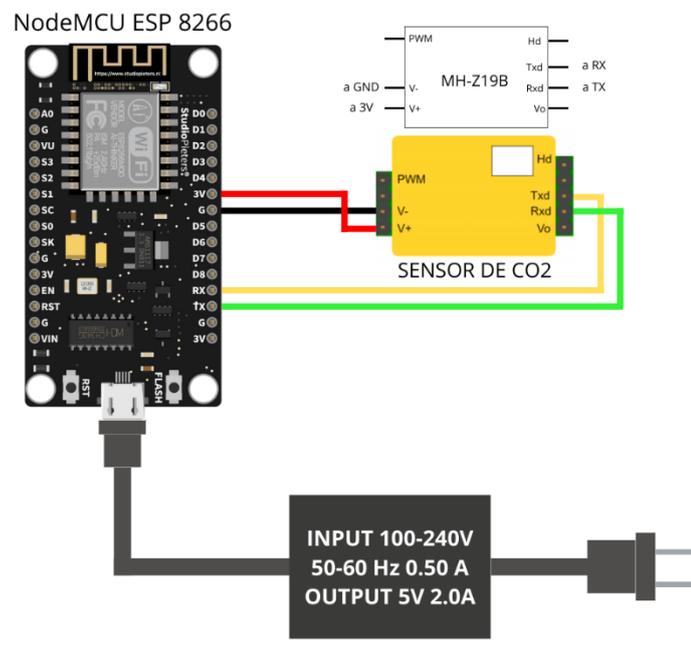
Figura 34: Parámetros del módulo sensor FC-28. (Elaboración propia)

Es importante recalcar que, en caso de este sensor el firmware no posee una librería específica y es necesario configurar la entrada A0 GPIO17 como ADC Rango, de este modo el dispositivo enviará un conjunto de valores dentro del rango de medición de la tarjeta

acondicionadora, estos datos serán de tipo cuantitativo y su variación en el tiempo depende del principio físico de medición, posteriormente estos valores serán escalados y calibrados en el servidor Edge.

- **Módulo sensor de CO₂**

Para construir el módulo de sensor de dióxido de carbono, conectaremos el sensor MH-Z19B con la NodeMCU y una fuente de energía como se muestra en la figura 34.



*Figura 35: Diagrama de conexión del módulo sensor de CO₂ MH-Z19B.
(Elaboración propia)*

El proceso de configuración de este módulo es muy similar al del módulo sensor FC-28, la única diferencia serán los valores asignados al topic MQTT y la configuración de los parámetros del módulo, en este caso el firmware dispone de una librería específica para la configuración de este sensor y quedaría establecida de la siguiente manera:

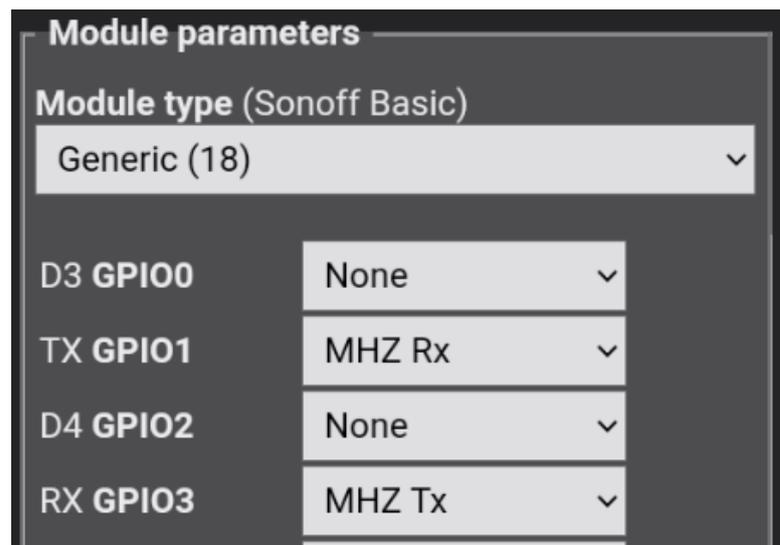


Figura 36: Parámetros del módulo sensor MH-Z19B. (Elaboración propia)

En este módulo es muy importante respetar la conexión física de los pines Rx y Tx, si estos no se encuentran bien conectados el sensor no funcionará.

- **Módulo actuador**

A diferencia de los módulos sensores, fue necesario realizar ciertas modificaciones detalladas anteriormente para instalar el firmware Tasmota, en este punto la configuración es igual que el resto de los dispositivos, pero su diagrama de conexión física si es diferente y se detalla en la figura 37.

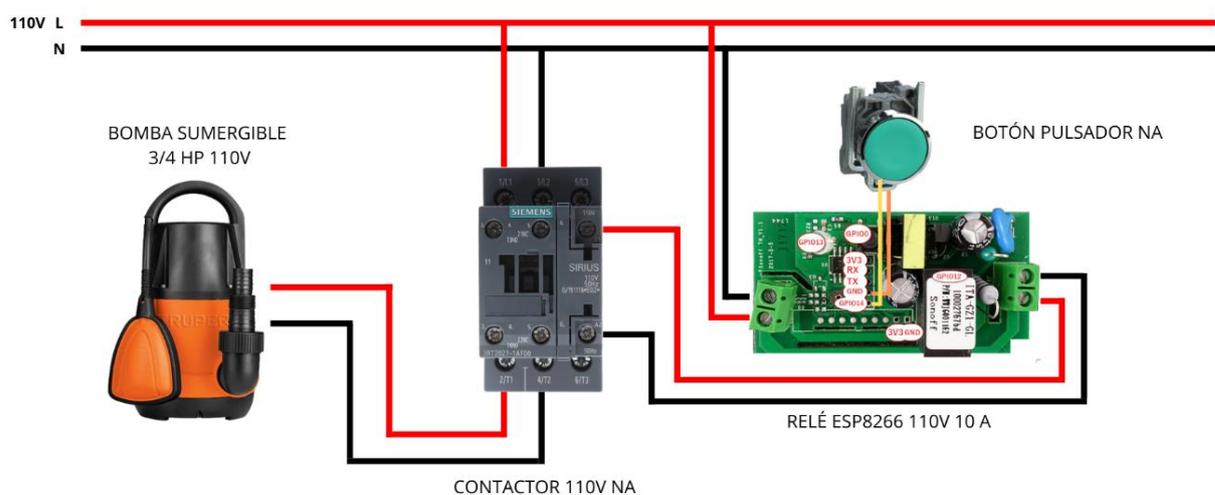


Figura 37: Diagrama de conexión del módulo actuador. (Elaboración propia)

La configuración de los parámetros del módulo quedaría establecida de la siguiente manera:

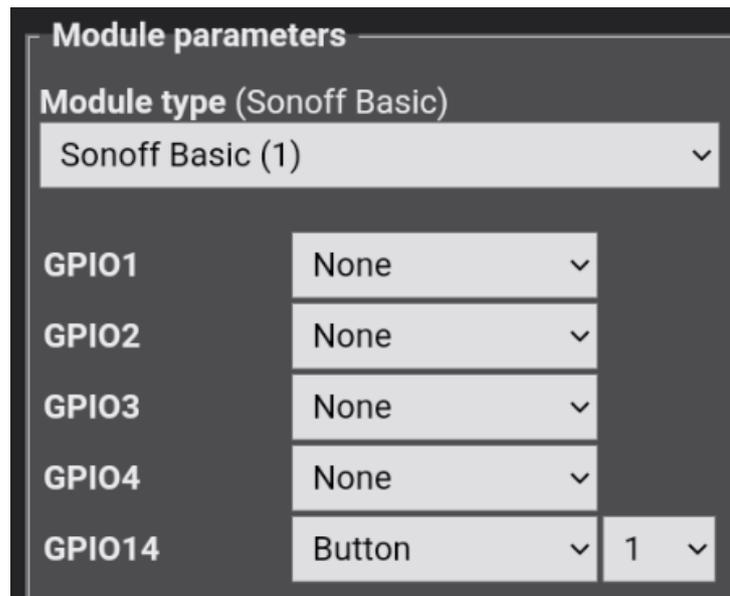


Figura 38: Parámetros del módulo actuador. (Elaboración propia)

Al usar un botón pulsador conectado al puerto GPIO14 es posible comandar las bombas tanto física como manualmente, tanto el botón físico como virtual generarán el mismo mensaje mqtt para conocer el estado del actuador.

Como se pudo observar en los procesos anteriores, el firmware Tasmota permite gestionar la configuración de cada uno de los dispositivos para que se conecten al broker MQTT instalado en el servidor Edge, de esta manera es muy fácil escalar el sistema ya que podemos agregar nuevos dispositivos y si es necesario modificar los tópicos para un despliegue más eficiente.

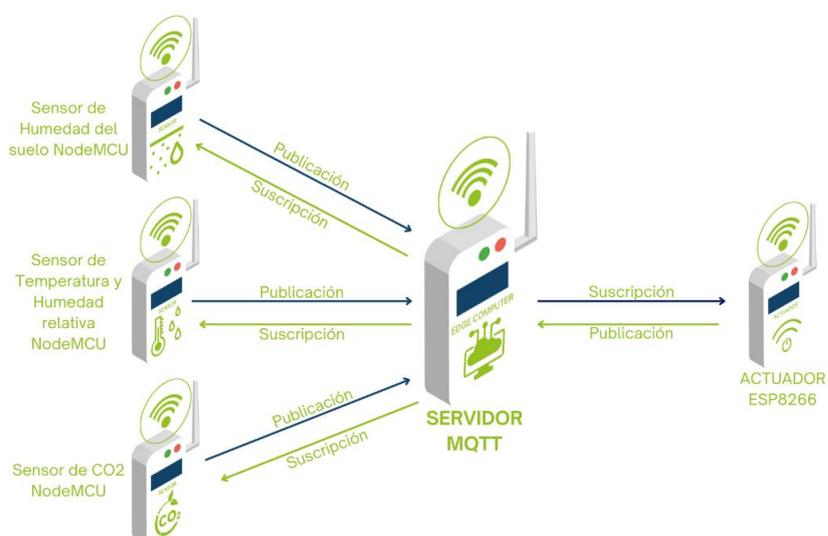


Figura 39: Arquitectura MQTT del sistema. (Elaboración propia)

- **Programación del Backend y Frontend del sistema en Node-Red.**

La programación del sistema tanto para el backend como para el frontend ha sido desarrollada en la plataforma Node-Red, esta posee una interfaz de programación gráfica basada en flujos la misma nos permite trabajar la parte lógica y la interfaz del sistema de manera simultánea por medio de nodos. En esta sección se explicará de manera breve el funcionamiento de cada uno de los flujos que conforman el sistema y la interfaz a la que están asociados.

En el siguiente flujo se puede observar los nodos que contienen la configuración de la interfaz de usuario y la definición de variables temporales por medio de flujos de tipo función, en estos nodos están definidos los colores, logos y básicamente el aspecto visual de la interfaz.

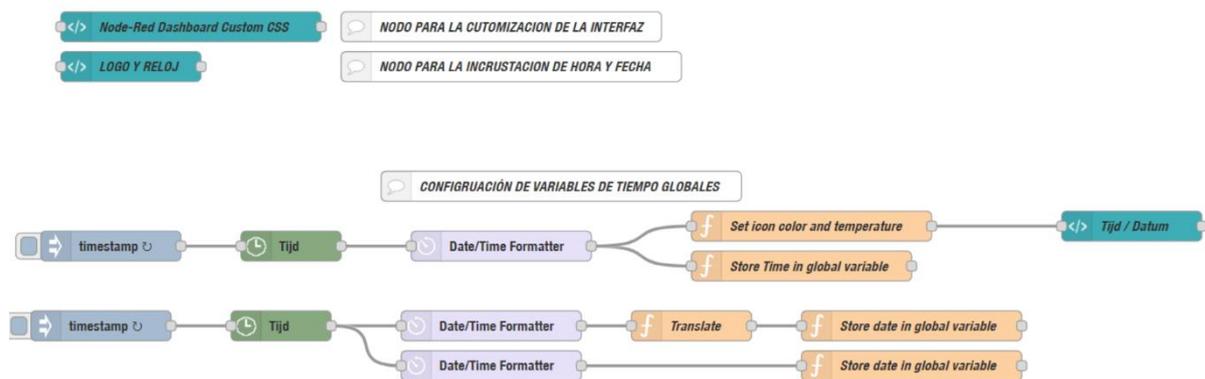


Figura 40: Flujo de interfaz y variables del sistema. (Elaboración propia)

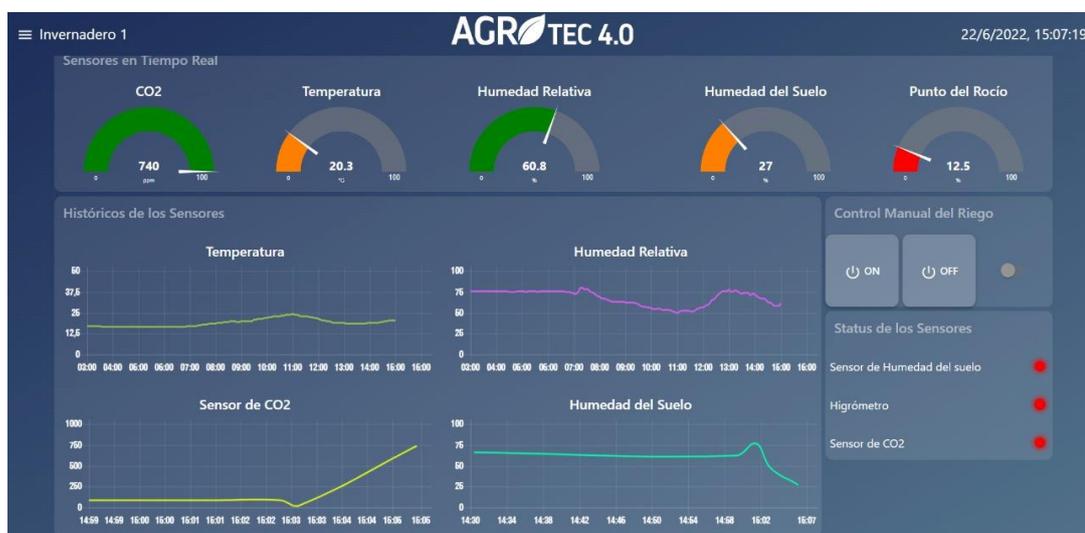


Figura 41: Primera versión de la interfaz. (Elaboración propia)

Es importante recalcar que gran parte del éxito de un sistema o aplicación es un buen diseño de interfaz, para esto se han considerado los colores, disposición de los elementos, cantidad de texto, tamaño y muchos otros aspectos con el fin de obtener un producto de fácil manejo y

atractivo a la vista. Durante el desarrollo de esta propuesta se probaron diferentes configuraciones del aspecto visual hasta llegar a la versión final que se detallará en el resto de las figuras de esta sección.

En la figura 42 y 43 respectivamente se pueden observar los nodos utilizados para obtener los datos meteorológicos de otros sistemas por medio de APIs y la interfaz gráfica generada.

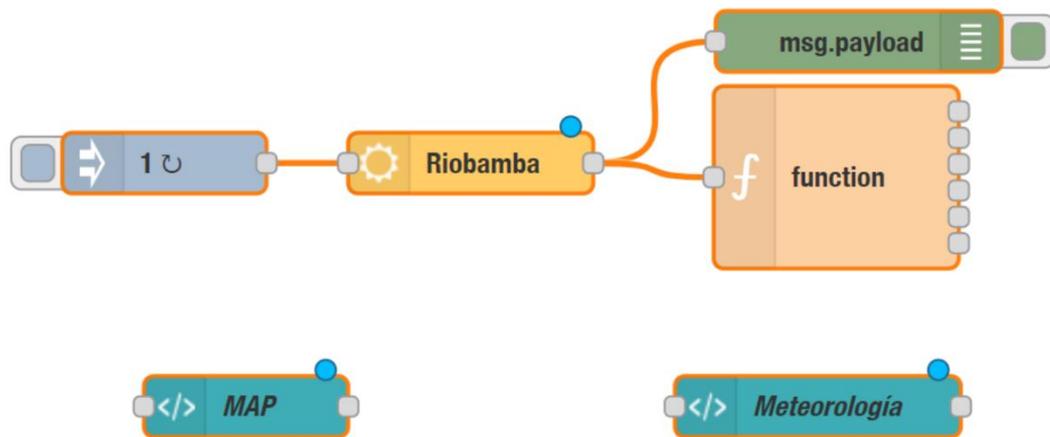


Figura 42: Flujo de APIs embebidas. (Elaboración propia)



Figura 43: Interfaz mapa meteorológico. (Elaboración propia)

El siguiente flujo describe el proceso de comunicación, transformación, procesamiento y análisis de la información generada por el sensor FC-28, por medio de este flujo el mismo es capaz de enviar notificaciones a un número de whatsapp si el valor de la variable se sale de los rangos programados, procesar los datos y enviarlos a la nube por medio de mensajes http,

calibrar el sensor por medio de funciones y mostrar mensajes de alerta en la interfaz de usuario.

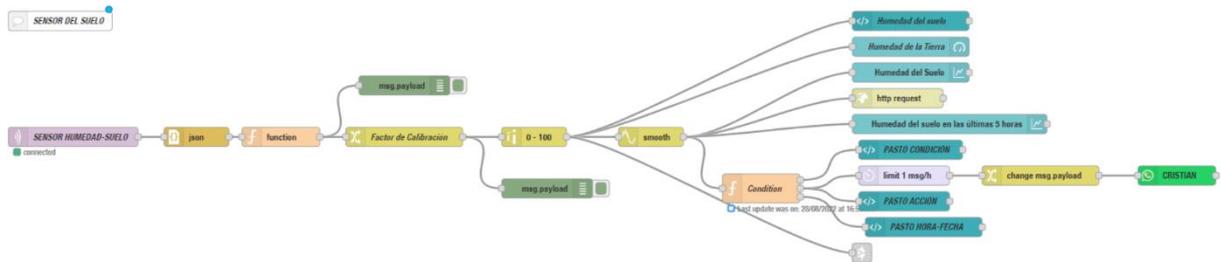


Figura 44: Flujo del sensor FC-28. (Elaboración propia)

En la figura 45 se muestra el flujo que describe el proceso realizado para obtener datos del sensor de temperatura y humedad relativa, primero se transforma el mensaje y por medio de un nodo función se divide el mismo en diferentes valores correspondientes a cada una de las tres variables que mide este sensor, posteriormente estos datos son procesados y enviados a la nube e interfaz de visualización.

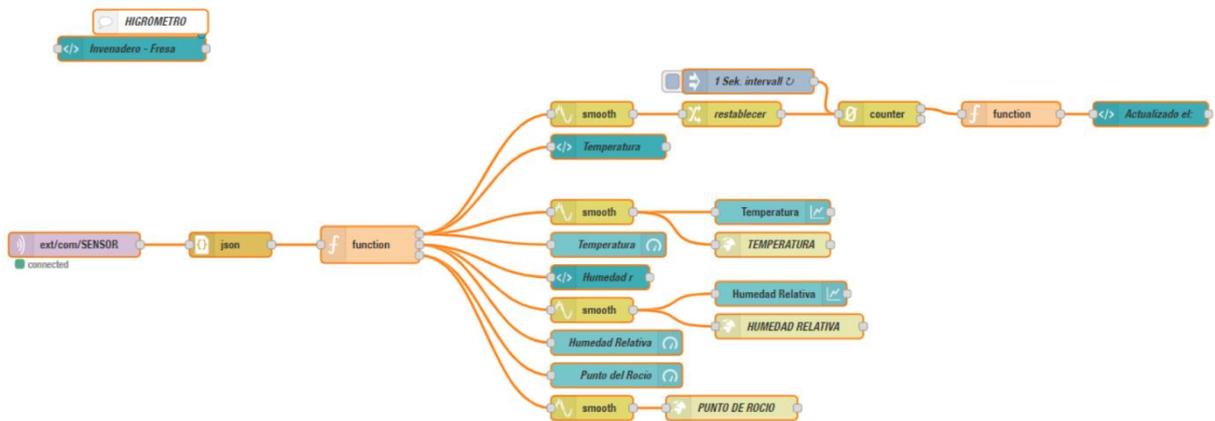


Figura 45: Flujo del sensor DHT21. (Elaboración propia)

En esencia el proceso que realiza el flujo de la figura 46 es igual al del sensor DHT21, pero con la diferencia de que este sensor solo envía un dato de tipo cuantitativo al sistema.

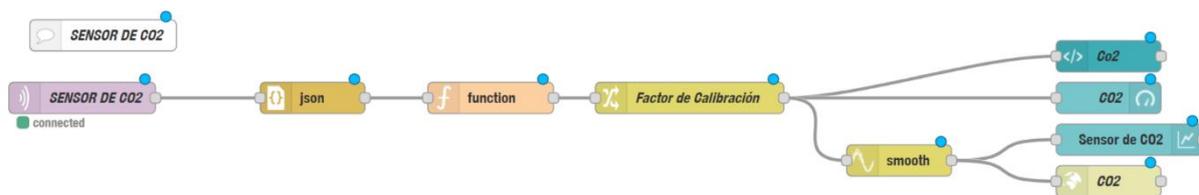


Figura 46: Flujo del sensor MH-Z19B. (Elaboración propia)

En la figura 47 se pueden observar los nodos que componen el sistema de control manual de cada uno de los actuadores y el nodo encargado de gestionar la programación de eventos.

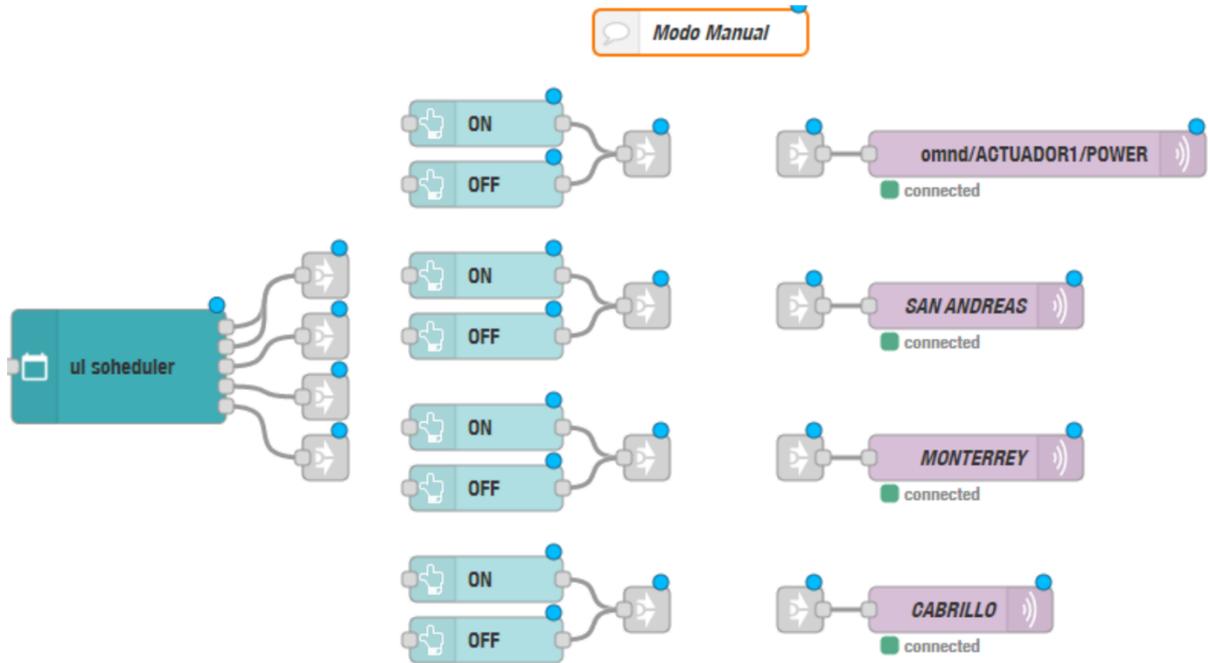


Figura 47: Flujo para el control manual. (Elaboración propia)

La figura a continuación describe el flujo mediante el cual el sistema puede automatizar el riego en base un rango configurable, esto quiere decir que el usuario puede establecer un umbral y si el sensor detecta valores inferiores a este el sistema automáticamente enviara una señal al actuador para tomar acción.

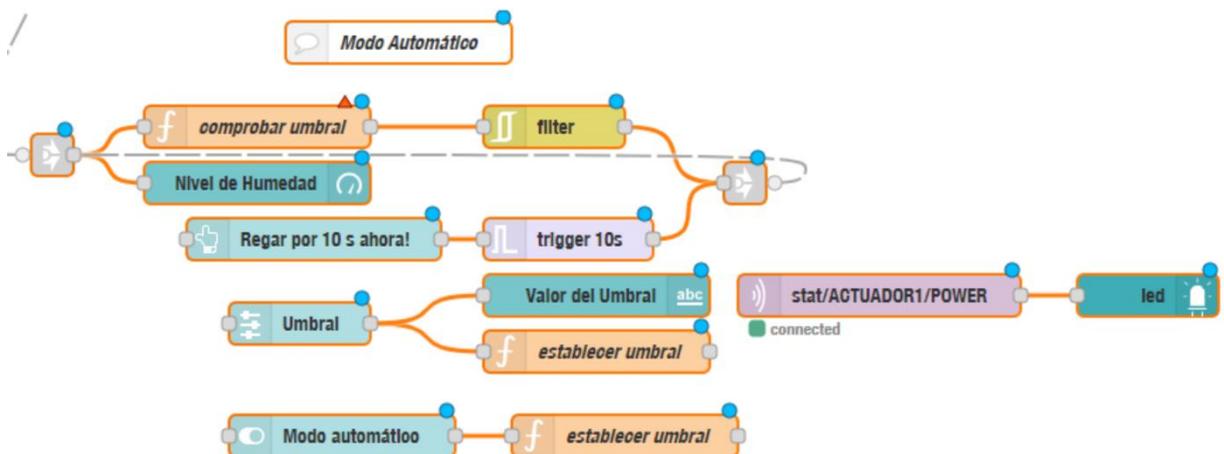


Figura 48: Flujo para el control automático. (Elaboración propia)

El conjunto de nodos de la figura 49 describe un flujo simple para verificar el estatus de los actuadores por medio de mensajes enviados desde los dispositivos IoT, únicamente con respecto a si están encendidos o apagados para ser visualizados en la interfaz del usuario.

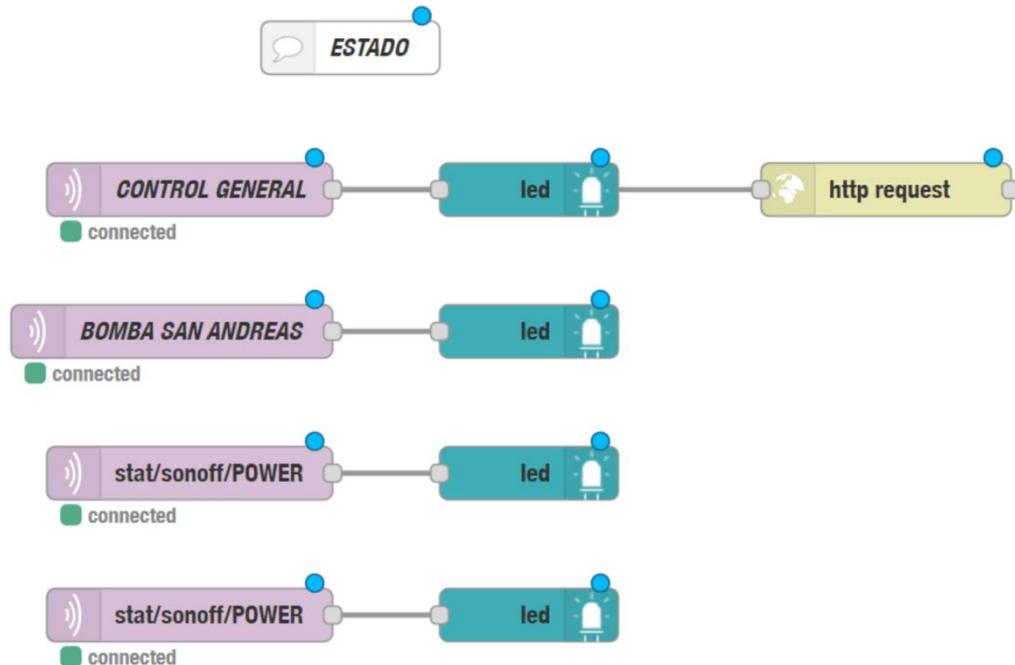


Figura 49: Flujo para el monitoreo del estado de los actuadores. (Elaboración propia)

La figura 50 describe el flujo para capturar el estado y tiempo de funcionamiento de las bombas, por medio de este flujo se puede calcular el consumo de agua en base al tiempo de funcionamiento y un caudal conocido.

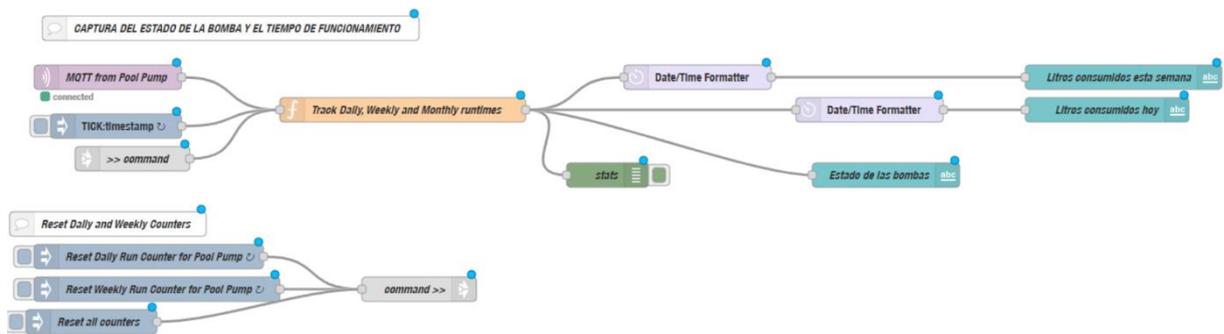


Figura 50: Flujo para capturar el tiempo de funcionamiento de un actuador. (Elaboración propia)

En el siguiente gráfico se puede ver los flujos utilizados para monitorear el estado de los sensores, este flujo comprueba la cantidad de mensajes en un determinado tiempo y si por algún motivo los datos no llegan automáticamente el sistema interpreta que hubo un problema con el sensor y notifica al usuario por medio de la interfaz y mensajes de WhatsApp.

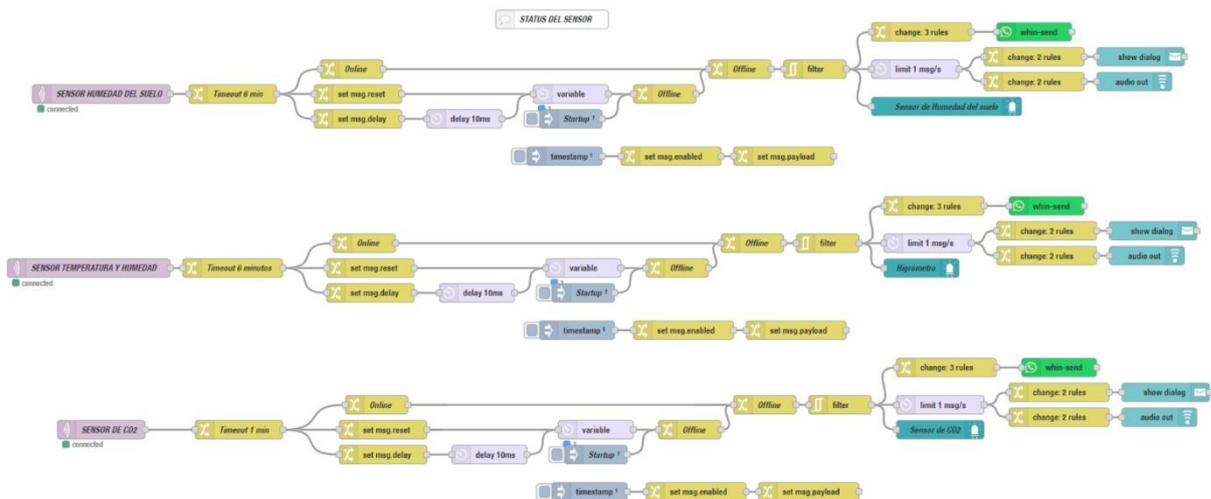


Figura 51: Flujo para el monitoreo del estado de la conexión de los dispositivos.

(Elaboración propia)

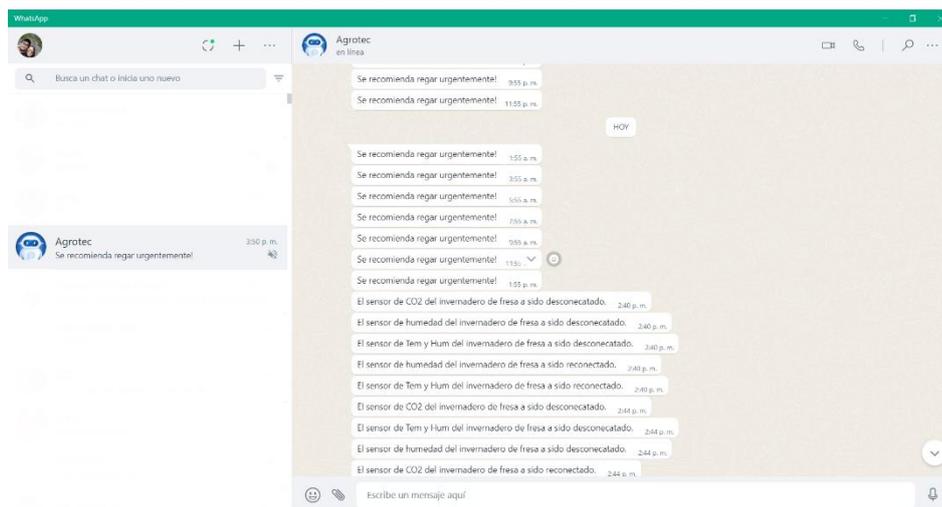


Figura 52: Bot de WhatsApp para notificaciones del sistema. (Elaboración propia)

En el siguiente grupo de figuras se puede observar las diferentes pantallas que comprenden la interfaz del usuario, las mismas fueron desarrolladas en cada uno de los flujos por medio de nodos los mismos que fueron analizados anteriormente.

En la figura 53 se puede observar la pantalla principal de la interfaz del sistema, en esta encontramos un resumen de las variables medidas en los diferentes invernaderos, además

de un resumen de la temperatura actual del lugar donde se encuentra el cultivo y un pronóstico del clima para los siguientes siete días.

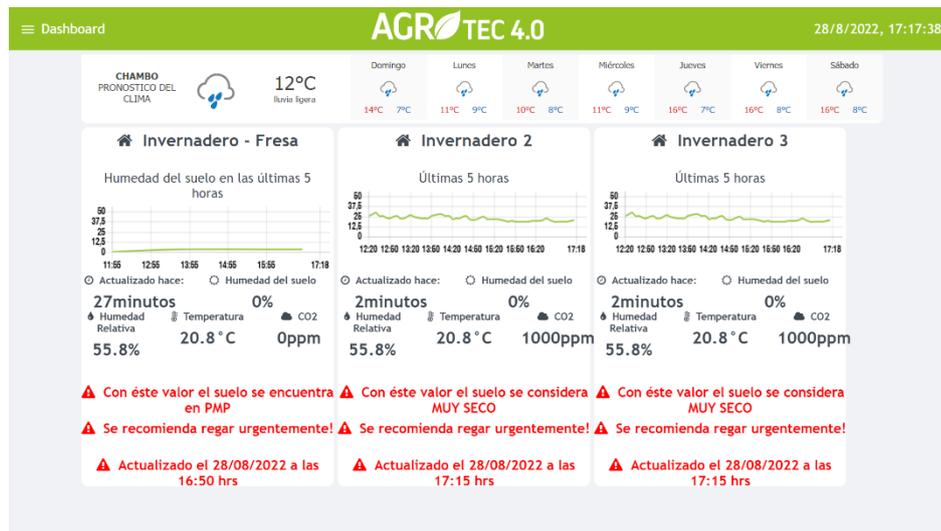


Figura 53: Dashboard principal. (Elaboración propia)

La figura 54 simplemente muestra el menú del sistema y el modo de navegación entre las diferentes funcionalidades del mismo, este menú se despliega solo cuando se da clic en el ícono a la izquierda de la palabra Dashboard

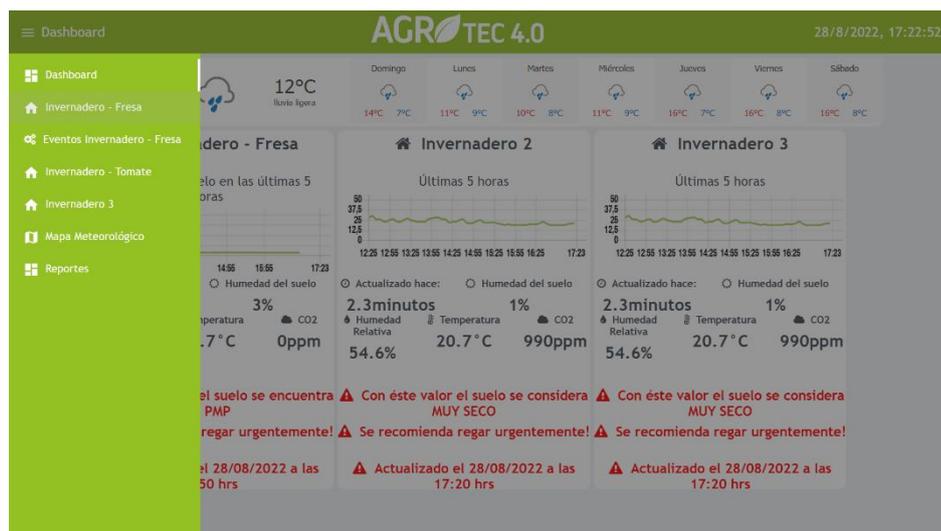


Figura 54: Menú del sistema. (Elaboración propia)

En la siguiente figura se puede observar un resumen de los parámetros que están siendo evaluados dentro de un cultivo de fresa bajo invernadero, esta pantalla esta configura de tal manera que se puede visualizar la medida del sensor en tiempo real, el estatus de los sensores y los registros históricos de las últimas 24 horas de datos.



Figura 55: Dashboard para el monitoreo de cultivo de fresa. (Elaboración propia)

La figura 56 corresponde a la interfaz de configuración de eventos del invernadero, permite visualizar la fecha y hora actual, programar eventos de encendido y apagado de los actuadores, ver el tiempo acumulado de funcionamiento de las bombas, controlar manualmente el encendido y apagado de las bombas o automatizar el riego en base a un umbral establecido como parámetro para controlar el nivel de humedad.

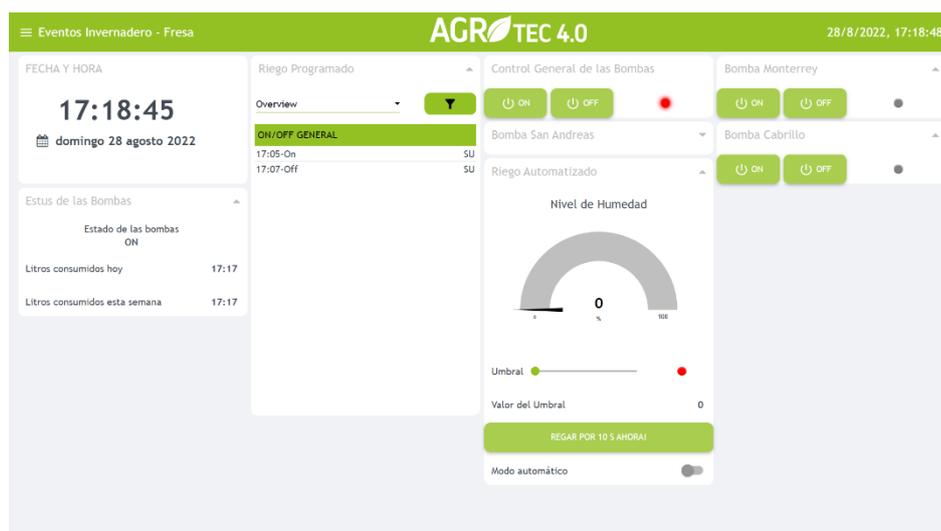


Figura 56: Interfaz de control de eventos. (Elaboración propia)

- **Diseño del dashboard en Data Studio**

Para realizar el diseño del dashboard en Data Studio el primer paso fue configurar las fuentes de datos, cada uno de los dispositivos IoT del sistema utiliza una hoja de cálculo de Google como base de datos para cada una de sus variables, a su vez estas hojas de cálculo son las

fuentes de datos para configurar Data Studio. En la figura 57 se puede observar cada una de estas fuentes agregada a la herramienta de análisis.

Nombre	Tipo de conector	Tipo	Se utiliza en el informe	Estado	Acciones	Alias
SENSOR DE HUMEDAD RELATIVA (resp...	Hojas de cálculo de Go...	Insertada	5 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds58
DB-SENSOR DE HUMEDAD (respuestas...	Hojas de cálculo de Go...	Insertada	5 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds2
Control General Bombas (respuestas) - ...	Hojas de cálculo de Go...	Insertada	3 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds150
SENSOR DE CO2 (respuestas) - Respue...	Hojas de cálculo de Go...	Insertada	5 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds89
SENSOR DE TEMPERATURA (respuesta...	Hojas de cálculo de Go...	Insertada	5 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds6
SENSOR PUNTO DEL ROCIO (respuesta...	Hojas de cálculo de Go...	Insertada	5 gráficos	En funcionamiento	EDITAR, DUPLICAR, QUITAR, HACER REUTILIZABLE	ds131

Figura 57: Fuentes de datos Data Studio. (Elaboración propia)

Data Studio es una herramienta muy fácil de usar, una vez agregadas las fuentes de datos ya se puede empezar a diseñar las pantallas necesarias para facilitar los reportes, esta interfaz está embebida dentro de la aplicación web del sistema Edge, pero también puede ser usada de manera independiente accediendo desde cualquier navegador por medio de una URL.

Cada una de las pantallas ha sido diseñada pensando en el usuario final, con el principal objetivo de facilitar el monitoreo y generación de reportes dentro de periodos configurables.



Figura 58: Menú principal. (Elaboración propia)

La siguiente figura representa a cada uno de los reportes que puede generar el sistema, uno por cada variable en el cual es posible ver sus valores máximos y mínimos dentro de un período de tiempo el cual puede ser seleccionado por el usuario, para facilitar el análisis.



Figura 59: Reporte de humedad relativa. (Elaboración propia)

4.1.2.2. Puesta en marcha

En la primera semana se construyeron las versiones finales de la prueba de concepto, de tal manera que cada uno de los dispositivos sea lo más semejante a la versión final a implementar, para esto solo fue necesario ensamblar la electrónica de estos dentro de cajas con protección IP68 y un enchufe, de esta manera solo es necesario conectarlo a un tomacorriente para que dispositivo empiece a operar.



Figura 60: Módulo prototipo módulo actuador. (Elaboración propia)

En la siguiente semana se puso a prueba el funcionamiento del sistema, y el comportamiento de los flujos de datos simulando el censado.



Figura 61: Pruebas de adquisición de datos. (Elaboración propia)

4.1.2.3. Ejecución de pruebas

Durante la etapa de pruebas se verificó la estabilidad de los sensores, la respuesta a eventualidades como la falta de energía, la capacidad del sistema de recuperarse ante un paro completo entre otros, durante este tiempo se probaron algunas características del sistema como las notificaciones ante la desconexión de los dispositivos y el funcionamiento de la configuración para acceder remotamente al servidor Edge.

4.1.2.4. Evaluación de resultados

Durante esta etapa de pruebas se descubrieron algunos problemas, el primero en presentarse fue la desconexión de los dispositivos durante la noche, todos los días sin motivo aparente a ciertas horas de la madrugada los sensores se desconectaban y dejaban de enviar información, se analizaron todas las causas posibles como falla de hardware, configuración del firmware, conexión eléctrica, hasta llegar a la solución definitiva, que fue configurar la red inalámbrica para que trabaje en el canal menos congestionado.

Otro problema que se detectó fue que el servidor Edge no arrancaba automáticamente después de un corte de energía, esto se pudo solucionar de forma simple por medio de configuración, un problema aparentemente simple que puede resultar muy problemático en el momento de un fallo dentro del entorno real de trabajo.

Se pudo probar el comportamiento de la interfaz y solucionar pequeños problemas de la interfaz con respecto a la navegación dentro de la aplicación web.

Durante período de pruebas también se pudo evidenciar que el sensor FC-28 se degrada rápidamente por la corrosión que se produce en sus electrodos, al final de la semana de pruebas este sensor era completamente inservible por lo que se empezó a buscar una mejor opción.

4.1.2.5. Conclusiones

Una solución al problema de la rápida corrosión del sensor FC-28 fue el sensor HD-38, en esencia son iguales, la única diferencia son los materiales de sus electrodos, el HD-38 cuenta con una sonda fabricada en un material resistente a la corrosión y al medio ambiente a los que se encuentra expuesto el sistema.

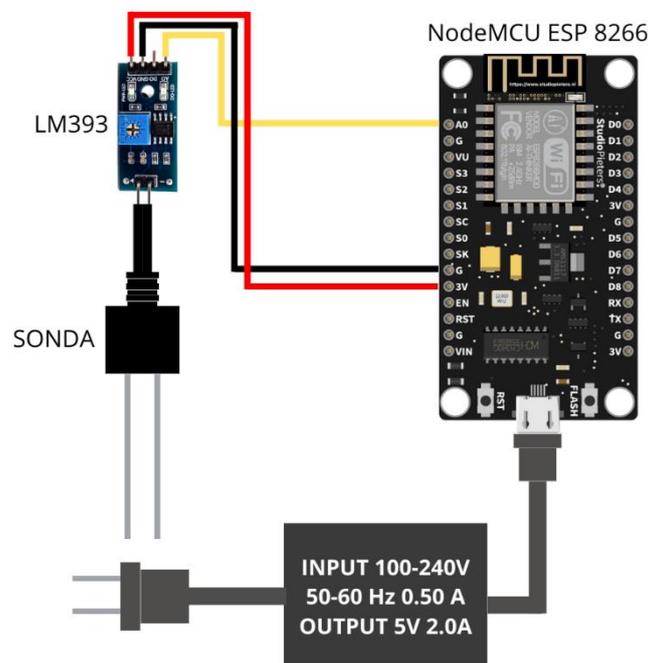


Figura 62: Diagrama de conexión del sensor HD-38. (Elaboración propia)

Después de solucionar los problemas mencionados anteriormente, en general la prueba de concepto fue exitosa, el sistema es viable desde el punto de vista técnico y funcional. Cada uno de los componentes cumplió con la función para la que fue diseñado y respondió muy bien ante la simulación de eventualidades a las que se pueda exponer el sistema.

El sistema fue diseñado y configurado de tal manera que no requiere modificaciones para empezar a trabajar en el cultivo de fresa bajo invernadero, gracias a las pruebas realizadas, para su implementación solo es necesario conectar los dispositivos a una fuente de energía y empezaran a operar automáticamente.

4.1.3. Sprint 3: Implementación

Puesto que la prueba de concepto fue un éxito, la implementación del sistema es sumamente fácil y rápida, como se mencionó anteriormente solo es necesario anclar los dispositivos, conectarlos a una toma eléctrica y empezarán a funcionar automáticamente.

4.1.3.1. Instalación de la Red inalámbrica

A continuación, se muestra un gráfico en el que se resume la instalación de las redes inalámbricas y cada uno de los componentes del sistema dentro del entorno de implementación, para la instalación de la red inalámbrica solo fue necesario anclar las antenas en las torres previamente instaladas, conectar los cables según los diagramas de conexión detallados anteriormente y esperar a que la red empiece a funcionar.



Figura 63: Entorno de implementación del sistema. (Elaboración propia)

Las antenas CPE210 y CPE510 solo requieren de una correa para ser ancladas, esta vienen incluidas en sus cajas, los cables LAN fueron los mismos que se utilizaron durante las pruebas de concepto. Se probó la calidad de la red midiendo la velocidad de conexión y latencia.

Las antenas se han colocado de tal manera que pueden cubrir una gran superficie del centro de investigación, dentro de esta misma instalación se podrían colocar más dispositivos para los diferentes cultivos experimentales.



Figura 64: Instalación de la antena CPE510. (Elaboración propia)

4.1.3.2. Instalación del servidor Edge

Para instalar el servidor Edge o Edge Computer solo fue necesario conectar la caja a un tomacorriente y anclarlo por medio de correas a la misma torre donde se encuentran ubicadas las antenas receptadora CPE510 y emisora CPE210.

Dentro de la caja Edge Computer también se encuentran las conexiones eléctricas de las antenas CPE510 y CPE210.



Figura 65: Servidor Edge. (Elaboración propia)

Una vez conectado se comprobó que el sistema está funcionando al ingresar a la interfaz web, otra forma de verificar su funcionamiento es accediendo al servidor Edge por medio del escritorio remoto.

4.1.3.3. Instalación de los sensores y el actuador

El sistema propuesto está diseñado para adaptarse a las instalaciones ya existentes, en este caso el invernadero ya contaba con tomas de luz para conectar lámparas y bombas de agua, la fuente de energía del invernadero es un panel fotovoltaico. En este punto la red inalámbrica ya se encuentra instalada y cubre de Wifi al área del invernadero por lo que solo bastara con configurar las credenciales de red para que cualquier dispositivo IoT se conecte a la red.

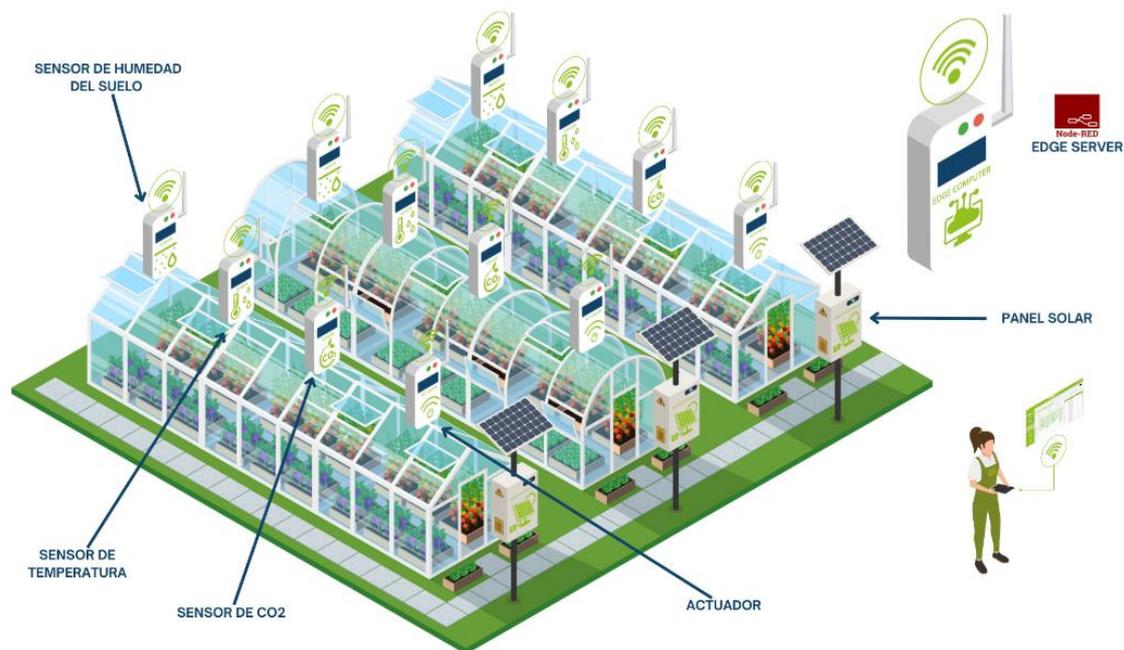


Figura 66: Red de sensores inalámbricos. (Elaboración propia)

El módulo sensor de humedad del suelo fue ubicado a un lado de la planta de fresa, al igual que los demás dispositivos solo fue necesario conectarlo y en este caso enterrar la sonda para que automáticamente empiece a operar enviando datos al sistema.



Figura 67: Módulo sensor inalámbrico de humedad del suelo. (Elaboración propia)

El módulo sensor de dióxido de carbono se ubicó en una zona donde el aire pueda circular, se conectó a la toma de energía y automáticamente empieza a enviar información al sistema.



Figura 68: Módulo sensor inalámbrico de CO2. (Elaboración propia)

El módulo sensor de temperatura y umedad relativa es el primer sensor que se instaló, al igual que el resto de sensores este fue anclado a uno de los pilares del invernadero por medio de correas y gracias a su configuración previa solo fue necesario conectarlo a la fuente de energía para que empiece a funcionar.



Figura 69: Modulo sensor inalámbrico de humedad relativa y temperatura.

(Elaboración propia)

El módulo actuador se instaló a la entrada del invernadero, de esta manera el encargado del manejo del cultivo puede activar o desactivar el riego por medio de un botón pulsador, este botón envía una señal lógica al sistema y no recibe la carga directa de la corriente de los motores de esta manera se puede alargar significativamente la vida útil de este dispositivo.



Figura 70: Modulo actuador IoT. (Elaboración propia)

4.1.4. Sprint 4: Pruebas

En este último sprint se probó el comportamiento del sistema, el mismo lleva funcionando varias semanas recopilando información de cada uno de los sensores, durante este tiempo se ha podido evidenciar que el servidor Edge ha funcionado muy bien, el único incidente de desconexión reportado por el sistema fue causado por una sobrecarga en el invertidor de voltaje del panel fotovoltaico, este fallo fue ajeno al sistema.

Con respecto al sensor de dióxido de carbono se pudo ver que después de los primeros días dejó de enviar información, esto se debió a que la humedad relativa del invernadero durante las noches alcanza el 100% provocando condensación, por el principio físico de medición este sensor no puede trabajar bien en la presencia de condensación por lo que resultó necesario analizar otro tipo de sensor, en este caso el sensor ideal sería el MH-Z16 ya que presenta una mejor respuesta ante la presencia de altos niveles de humedad y condensación.



Figura 71: Reportes de dióxido de carbono. (Elaboración propia)

Con respecto al sensor de humedad del suelo se pudo observar que a pesar de que su sonda es anticorrosiva al pasar del tiempo sus mediciones se van degradando hasta el punto de no enviar datos si no se realiza una limpieza de los electrodos, en base al análisis de la bibliografía este problema puede ser solucionado utilizando un sensor de luminosidad, con este sensor se puede medir los requerimientos hídricos del cultivo de una manera más fiable y duradera.



Figura 72: Reporte de humedad del suelo. (Elaboración propia)

El sensor que mejor ha funcionado hasta el momento es el DHT21, durante las pruebas no presento problemas y los valores medidos concuerdan con la de los instrumentos análogos con los que contaban anteriormente, este sensor arroja datos de tres variables las mismas que se pueden observar en las figuras 73,74 y 75. Al momento este dispositivo a registrado cerca de 4,500 datos, con cero fallos reportados. A continuación, se puede observar los reportes de cada una de las variables.



Figura 73: Reporte del punto de rocío. (Elaboración propia)



Figura 74: Reporte de temperatura. (Elaboración propia)



Figura 75: Reporte de humedad relativa. (Elaboración propia)

5. Conclusiones y trabajos futuros

5.1. CONCLUSIONES

El principal objetivo de este trabajo era implementar un sistema de Edge Computing para el manejo de cultivo de fresa bajo invernadero. Gracias a la investigación bibliográfica este objetivo se ve cumplido en cada uno de los sprint planteados a través de la metodología SCRUM.

Uno de los principales aportes de esta propuesta fue la configuración de un servidor Edge de muy bajo costo, en base a la investigación realizada se podría decir que es el más económico del mercado, por lo que si su funcionamiento se mantiene igual durante su vida útil resultaría una solución muy eficiente para futuros proyectos.

Por medio de la investigación se determinaron las principales variables de control para el manejo de cultivo de fresa bajo invernadero, una contribución muy significativa para el centro experimental ya que hasta este estudio no contaban con información respecto a parámetros cuantitativos para el manejo del cultivo.

A excepción del sensor MH-Z19B y el sensor FC-28 la selección de las tecnologías en base a criterios técnicos y económicos fue la más adecuada para la construcción del sistema de ege Computing, logrando así crear un prototipo funcional, económico, fiable y escalable para la agricultura inteligente en un tiempo muy corto en comparación al resto de tecnologías analizadas.

El sistema fue instalado exitosamente en un cultivo de fresa bajo invernadero real, convirtiéndose en una herramienta clave para futuras investigaciones del centro experimental, gracias a la interfaz desarrollada y la facilidad que esta brinda para llevar a cabo los procesos de control y monitoreo del cultivo en menos tiempo, el personal a cargo puede enfocarse en otras actividades más relevantes para el proyecto.

En base a los resultados conseguidos este proyecto, un sistema de Edge Computing para la agricultura inteligente es una solución muy atractiva y viable para este sector, gracias a su flexibilidad y escalabilidad puede ser aplicada no solo en cultivos de fresa sino también en otros cultivos perenes bajo invernadero como el tomate, pepinos, arándanos, etc. Buscando siempre como resultado un incremento en el rendimiento de los cultivos, estas tecnologías tienen el potencial de hacer de la agricultura una actividad mucho más rentable y menos sacrificada.

5.2. “LÍNEAS DE TRABAJO FUTURAS”

Pensando siempre en la mejora continua del sistema, las líneas de trabajo futuras que a continuación se plantean posibles mejoras para este proyecto, pudiendo así complementar ciertos aspectos necesarios del sistema que no se trataron en esta propuesta.

5.2.1. Inteligencia artificial

Basados en los datos recolectados en el tiempo por los sensores y por medio de la inteligencia artificial en una futura versión el sistema, este podría alertar si el cultivo se encuentra en riesgo de contraer alguna enfermedad o plaga, facilitando el control preventivo y reduciendo el uso de pesticidas, aspectos que se ven reflejados directamente en los costos de producción.

5.2.2. Evaluación del impacto de la tecnología en el rendimiento del cultivo.

Un proyecto muy interesante sería evaluar el impacto de la tecnología Edge en el rendimiento de diferentes cultivos, comparando los resultados de producción obtenidos por medio un manejo convencional en base a un monitoreo y control manual versus un sistema automatizado basado en las tecnologías planteadas a lo largo de este TFM.

Referencias bibliográficas

- Acharya, B., Garikapati, K., Yarlagadda, A., & Dash, S. (2022). Internet of things (IoT) and data analytics in smart agriculture: Benefits and challenges. *AI, Edge and IoT-Based Smart Agriculture*, 3-16. <https://doi.org/10.1016/B978-0-12-823694-9.00013-X>
- Alali, K. T., Liu, J., Yu, J., Moharram, D., Chen, R., Zhang, H., Liu, Q., Zhang, M., & Wang, J. (2020). HFIP-functionalized electrospun WO₃ hollow nanofibers/rGO as an efficient double layer sensing material for dimethyl methylphosphonate gas under UV-Light irradiation. *Journal of Alloys and Compounds*, 832, 154999. <https://doi.org/10.1016/j.jallcom.2020.154999>
- Alobaidy, H. A. H., Mandeep, J. S., Nordin, R., & Abdullah, N. F. (2020). A Review on ZigBee Based WSNs: Concepts, Infrastructure, Applications, and Challenges. *International Journal of Electrical and Electronic Engineering & Telecommunications*, 9(3), 190-198. <https://doi.org/10.18178/ijeetc.9.3.189-198>
- ArduinoQue. (2022). *Sensor de luminosidad arduino*. ArduinoQue. <https://arduinoque.com/arduino/sensor-de-luminosidad-arduino/>
- Asmi, J., & Candra, O. (2020). Prototype Solar Tracker Dua Sumbu Berbasis Microcontroller Arduino Nano dengan Sensor LDR. *JTEV (Jurnal Teknik Elektro Dan Vokasional)*, 6(2), 54-63. <https://doi.org/10.24036/JTEV.V6I2.108504>
- Aviles, P. M., Lindoso, A., Belloch, J. A., & Entrena, L. (2021). Evaluating reliability through soft error triggered exceptions at ARM Cortex-A9 microprocessor. *Microelectronics Reliability*, 126, 114323. <https://doi.org/10.1016/J.MICROREL.2021.114323>
- Bankov, D., Khorov, E., & Lyakhov, A. (2017). *On the Limits of LoRaWAN Channel Access*. 10-14. <https://doi.org/10.1109/ENT.2016.011>
- Betancourth Catro, L. (2016). *Sensores de Humedad del Suelo Tipo Sonda con Sistema de Monitoreo para Aplicaciones en Agricultura de Precisión*. [Universidad de los Llanos]. <https://doi.org/10.1/JQUERY.MIN.JS>
- Biggs, J., Myers, J., Kufel, J., Ozer, E., Craske, S., Sou, A., Ramsdale, C., Williamson, K., Price, R., & White, S. (2021). A natively flexible 32-bit Arm microprocessor. *Nature* 2021 595:7868, 595(7868), 532-536. <https://doi.org/10.1038/s41586-021-03625-w>
- Bonilla Correa, C. R. (2011). *Cartillas del Corredor Tecnológico Cultivando su Futuro*. Universidad Nacional de Colombia.

- Caicedo-Rosero, L. C., Méndez-Ávila, F. de J., Gutiérrez-Zeferino, E., & Flores-Cuautle, J. de J. A. (2021). Medición de humedad en suelos, revisión de métodos y características. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, 9(17), 1-8. <https://doi.org/10.29057/ICBI.V9I17.7035>
- Cameron, N. (2021). ESP32 microcontroller features. *Electronics Projects with the ESP8266 and ESP32*, 641-682. https://doi.org/10.1007/978-1-4842-6336-5_22
- Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An Overview on Edge Computing Research. *IEEE Access*, 8, 85714-85728. <https://doi.org/10.1109/ACCESS.2020.2991734>
- CenturyLink. (2022). *Cambiar la Configuración Wi-Fi de 5 GHz de tu Módem*. CenturyLink. <https://espanol.centurylink.com/home/help/internet/modems-and-routers/wireless-setup/change-wifi-channel-5ghz.html>
- Chakraborty, M., & Kundan, A. P. (2021). Grafana. *Monitoring Cloud-Native Applications*, 187-240. https://doi.org/10.1007/978-1-4842-6888-9_6
- Chen, C. H., Lin, M. Y., & Liu, C. C. (2018). Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers. *IEEE Network*, 32(1), 24-32. <https://doi.org/10.1109/MNET.2018.1700146>
- Committee, D. (2018). *Ministry of Agriculture & Farmers Welfare «Summation of Recommended Reforms and Implementation Items Covering Topics discussed in all Volumes of the DFI Report» Document prepared by the Committee on Doubling Farmers' Income*.
- Conteratto, C., Martinelli, G. do C., & Oliveira, L. de. (2020). SEGURANÇA ALIMENTAR, AGRICULTURA INTELIGENTE E SUSTENTABILIDADE: O ESTADO DA ARTE NO CAMPO CIENTÍFICO. *Journal on Innovation and Sustainability RISUS*, 11(2), 33-43. <https://doi.org/10.23925/2179-3565.2020v11i2p33-43>
- Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., & Weihl, B. (2002). Globally distributed content delivery. *IEEE Internet Computing*, 6(5), 50-58. <https://doi.org/10.1109/MIC.2002.1036038>
- Domínguez-Pumar, M. (2018). *Embedded Systems with Linux Programming the Beaglebone Board* (1.ª ed., Vol. 1). AMG Editions.
- Duque, S. I. O. (2017). Monitoreo y control de variables ambientales mediante una red inalámbrica para agricultura de precisión en invernaderos. *Revista Vector*, 12, 51-60. <https://doi.org/10.17151/VECT.2017.12.6>

- Electrónica Joan. (s. f.). *Itead Sonoff Basic R2*. Electrónica Joan. Recuperado 11 de septiembre de 2022, a partir de <https://electrojoan.com/itead-sonoff-basic-r2/>
- Electronilab. (s. f.). *Sensor de humedad de suelo Anticorrosivo – Higrómetro*. Electronilab. Recuperado 11 de septiembre de 2022, a partir de <https://electronilab.co/tienda/sensor-de-humedad-de-suelo-anticorrosivo-higrometro/>
- Eliseo Alvarado-Raya, H., Emilio Escamilla-García, P., Eugenia Estrada-Chavira, M., Pérez-Soto, F., & Elideth Moreno-López, K. (2022). La composta como reductor de gases de efecto invernadero en el sector agrícola: una revisión integral. *Tec. Econ. Agrar*, xx, 1-17. <https://doi.org/10.12706/itea.2022.012>
- Excelprince. (s. f.). *Before jump in to course contents lets understand why you should Learn Power BI*. Recuperado 5 de septiembre de 2022, a partir de <https://www.excelprince.com/pdf/7.%20Power%20BI%20Course%20Contents%20-%20ExcelPrince.pdf>
- Fakhrurroja, H., Mardhotillah, S. A., Mahendra, O., Munandar, A., Rizqyawan, M. I., & Pratama, R. P. (2019). Automatic pH and Humidity Control System for Hydroponics Using Fuzzy Logic. *2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 156-161. <https://doi.org/10.1109/IC3INA48034.2019.8949590>
- Fan, D. H., & Gao, S. (2018). The application of mobile Edge Computing in agricultural water monitoring system. *IOP Conference Series: Earth and Environmental Science*, 191(1). <https://doi.org/10.1088/1755-1315/191/1/012015>
- Felipe Osorio-Luna, D., Molina-Ruiz, P. A., & Artículo, I. del. (2019). Automation and control of greenhouse implemented technologies: a review. *Visión Electrónica, ISSN-e 2248-4728, ISSN 1909-9746, Vol. 2, N°. Extra 2, 2019 (Ejemplar Dedicado a: Special Edition), Págs. 381-394, 2(2)*, 381-394. <https://doi.org/10.14483/issn.2248-4728>
- Fischer, G., Melgarejo, L. M., & Balaguera-López, H. E. (2022). Revisión del impacto de concentraciones elevadas de CO₂ sobre frutales en la era del cambio climático. *Ciencia y Tecnología Agropecuaria*, 23(2), 2475. https://doi.org/10.21930/rcta.vol23_num2_art:2475
- Forero-Muñoz, A. M., & Ramírez-Delgado, M. A. (2019). Revisión sistemática del uso de los diferentes protocolos de interconexión para internet de las cosas (IoT). *IET Computing and Control Engineering*, 17(6), 18-23. <https://doi.org/10.1049/CCE:20060601>

- García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors*, 20(4), 1042. <https://doi.org/10.3390/s20041042>
- Garnier-Fernández, MSc. D. (2019). *Sistema sensor inalámbrico para el estudio y control de la calidad del aire en interiores*.
- Gaurav, K. (2022). Using Node-RED for IoT and Edge Computing. *OpenSource*, 1-7. <https://www.opensourceforu.com/2022/02/using-node-red-for-iot-and-Edge-Computing/>
- Gómez, A. H., Wang, J., Hu, G., & Pereira, A. G. (2006). Electronic nose technique potential monitoring mandarin maturity. *Sensors and Actuators B: Chemical*, 113(1), 347-353. <https://doi.org/10.1016/j.snb.2005.03.090>
- Gomez, C., Veras, J. C., Vidal, R., Casals, L., & Paradells, J. (2019). A Sigfox Energy Consumption Model. *Sensors 2019, Vol. 19, Page 681, 19(3)*, 681. <https://doi.org/10.3390/S19030681>
- Google Drive. (s. f.). *Archivos que se pueden guardar en Google Drive*. . Google Drive. Recuperado 18 de septiembre de 2022, a partir de <https://support.google.com/drive/answer/37603?hl=es>
- Grupo Editorial 3W México (WWW México, S. de C. (2021). Fresa en invernadero una opción de empresa familiar . *Revista Industrial Del Campo*. <http://www.2000agro.com.mx/agroindustria/fresa-en-invernadero-una-opcion-de-empresa-familiar/>
- ITEAD. (s. f.). *SONOFF BASICR2- WiFi Wireless Smart Switch* . Itead.Cc. Recuperado 11 de septiembre de 2022, a partir de <https://itead.cc/product/sonoff-basicr2/>
- Jaakkola, J. (2022). Dashboard-Työkalujen Soveltuvuus Yrityskäyttöön. *Tampereen Yliopisto Tietojenkäsittelytieteiden Tutkinto-Ohjelma* .
- Jabbar, W. A., Wei, C. W., Azmi, N. A. A. M., & Haironnazli, N. A. (2021). An IoT Raspberry Pi-based parking management system for smart campus. *Internet of Things*, 14, 100387. <https://doi.org/10.1016/J.IOT.2021.100387>
- Jaén, U. de, & Belén Estrada Valverde, A. (2021). *Agricultura biosalina y perspectivas de futuro*. <http://tauja.ujaen.es/jspui/handle/10953.1/14541>
- Johnston, S. J., Basford, P. J., Perkins, C. S., Herry, H., Tso, F. P., Pezaros, D., Mullins, R. D., Yoneki, E., Cox, S. J., & Singer, J. (2018). Commodity single board computer clusters

- and their applications. *Future Generation Computer Systems*, 89, 201-212. <https://doi.org/10.1016/J.FUTURE.2018.06.048>
- Kannwischer, M. J., Rijneveld, J., Schwabe, P., & Sto'elen, K. (2019). pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. *Radbound Repository*. <http://hdl.handle.net/2066/210214>
- Khakimov, A., Muthanna, A., Kirichek, R., Koucheryavy, A., & Muthanna, M. S. A. (2017). Investigation of methods for remote control IoT-devices based on cloud platforms and different interaction protocols. *Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, EIconRus 2017*, 160-163. <https://doi.org/10.1109/EICONRUS.2017.7910518>
- Kiran, S., & Gupta, G. (2020). A Pragmatic Analysis On Wireless Sensors And Actuator Technologies For Security In Internet Of Things kiran kits A Pragmatic Analysis On Wireless Sensors And Actuator Technologies For Security In Internet Of Things. *International Journal of Future Generation Communication and Networking*, 13(4), 5109-5116. https://d1wqtxts1xzle7.cloudfront.net/67721485/IJFGCN-with-cover-page-v2.pdf?Expires=1662483872&Signature=Or7nlwoO0fVXo3QvCRhPAjiX6tnhyXwsfWynWzNXwNCZwhAFMvLG~4a8aA8~8mMGCN5vFh~cbRr-P2Yc5DGctKZoB4hS5kDDtdBO5-ssHRIRDuvcllk4yzXPXtWJCcMm1Giyw1-NSTIMqJ7xc47adVIXy3QFZ9xW5ngsjTYVeLsAJe2FNGdi2Ldp7ryL3x0UmYFFoUHOwu zMFjUtPJANrw4cZiAsINyUpoHydPBmc1oOKWYj6th2m3eqb996GUxAT2EY1YGP6nTyWr6AKZC6bUsVII51qfKZL~Ceaf-PLtULI8zRM-OVvV1gFJxT0HsAHUC8QeL-uq8o4a7QH7XWJA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
- Libelium. (s. f.). *Soluciones IoT para una agricultura inteligente*. Libelium. Recuperado 4 de septiembre de 2022, a partir de <https://www.libelium.com/es/soluciones-iot/agricultura-inteligente/>
- Llamas, L. (2022, septiembre 7). *Medir nivel de luz con Arduino y fotoresistencia LDR (GL55)*. Ingeniería, Informática y Diseño. <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoresistencia-ldr/>
- Lozano-Povis, A., Alvarez-Montalván, C. E., Moggiano, N., Lozano-Povis, A., Alvarez-Montalván, C. E., & Moggiano, N. (2021). El cambio climático en los andes y su impacto en la agricultura: una revisión sistemática. *Scientia Agropecuaria*, 12(1), 101-108. <https://doi.org/10.17268/SCI.AGROPECU.2021.012>

- Marcelino, R., Casagrande, L. C., Cunha, R., Crotti, Y., & Gruber, V. (2018). *Internet of Things Applied to Precision Agriculture* (pp. 499-509). https://doi.org/10.1007/978-3-319-64352-6_46
- Meduri, S. P., Diwan, N., & Meduri, A. (2020). A sophisticated irrigation system with passive and analogue sensors over web using orange Pi. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, 395-399. <https://doi.org/10.1109/ICSSIT48917.2020.9214192>
- Mela, J. L., Cedeño, G. D., & Cedeño Herrera, E. (2021). Edge Computing: Aplicaciones y desafíos actuales. *Visión Antataura*, 5(1), 75-91. <https://doi.org/10.1109/FICLOUD.2014.83>
- Mesquita, J., Guimaraes, D., Pereira, C., Santos, F., & Almeida, L. (2018). Assessing the ESP8266 WiFi module for the Internet of Things. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2018-September*, 784-791. <https://doi.org/10.1109/ETFA.2018.8502562>
- Mohanraj, I., Ashokumar, K., & Naren, J. (2016). Field Monitoring and Automation Using IOT in Agriculture Domain. *Procedia Computer Science*, C(93), 931-939. <https://doi.org/10.1016/J.PROCS.2016.07.275>
- Nandhini, S., Bhrathi, S., Dheeraj Goud, D., & Pranay Krishna, K. (2019). Smart Agriculture IOT with Cloud Computing, Fog Computing and Edge Computing. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9, 2249-8958. <https://doi.org/10.35940/ijeat.B2600.129219>
- Naylamp, M. (s. f.). *NodeMCU v2 ESP8266 WiFi*. Naylamp, Mechatronics. Recuperado 11 de septiembre de 2022, a partir de https://naylampmechatronics.com/img/cms/ESP8266/esp8266_devkit_horizontal-01.png
- Naylamp Mechatronics. (s. f.-a). *Sensor de Humedad de Suelo Capacitivo*. Naylamp Mechatronics. Recuperado 11 de septiembre de 2022, a partir de <https://naylampmechatronics.com/sensores-temperatura-y-humedad/538-sensor-de-humedad-del-suelo-capacitivo.html#:~:text=El%20Sensor%20de%20humedad%20de,la%20vida%20%C3%BAtil%20del%20sensor.>
- Naylamp Mechatronics. (s. f.-b). *Tutorial módulo sensor de luz BH1750*. NaylampMechatronics. Recuperado 6 de septiembre de 2022, a partir de https://naylampmechatronics.com/blog/44_tutorial-modulo-sensor-de-luz-bh1750.html

- Naylamp Mechatronics. (s. f.-c). *Tutorial sensor de temperatura y humedad DHT11 y DHT22*. Naylampmechatronics. Recuperado 6 de septiembre de 2022, a partir de https://naylampmechatronics.com/blog/40_tutorial-sensor-de-temperatura-y-humedad-dht11-y-dht22.html
- Node-RED. (s. f.). Node-Red. Recuperado 27 de abril de 2022, a partir de <https://nodered.org/>
- Núcleo Ambiental S.A.S. (2015). MANUAL DE FRESA. En *Camara de Comercio de Bogotá*. https://www.academia.edu/34991683/MANUAL_DE_FRESA
- OdinS. (s. f.). *Invernadero inteligente*. OdinS. Recuperado 4 de septiembre de 2022, a partir de <https://www.odins.es/soluciones/invernadero-inteligente/>
- Palacio, I. C. (2021). Oportunidades para la transformación digital de la cadena de suministro del sector bananero basado en software con inteligencia artificial. *Revista Politécnica*, 17(33), 47-63.
- Pathak, S., Shivam, Mr., Yadav, T., & Gupta, A. (2021). Remote monitoring of Agriculture sector using IOT. *Int. J. of Aquatic Science*, 12(2), 1373-1379. http://www.journal-aquaticscience.com/article_132809.html
- Peón-Lamparero, C. (2022). IoT aplicado al ámbito de la agricultura inteligente en el hogar (Master's thesis). *Reunir*, 4-421. <https://reunir.unir.net/handle/123456789/12803>
- Plantect® (プランテクト®) | 病害予測機能搭載モニタリングサービス. (s. f.). Recuperado 4 de septiembre de 2022, a partir de <https://cropscience.bayer.jp/ja/home/plantect/>
- Premkumar, S., & Sigappi, A. N. (2022). Functional framework for Edge-based agricultural system. *AI, Edge and IoT-Based Smart Agriculture*, 71-100. <https://doi.org/10.1016/B978-0-12-823694-9.00029-3>
- Pretty, J. (2007). Agricultural sustainability: concepts, principles and evidence. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1491), 447-465. <https://doi.org/10.1098/RSTB.2007.2163>
- Rascón, J., Rascón, J., Huaman, J. C., Ángeles, W. G., Oliva, M., Marin, Y. R., Leiva-Tafur, D., & Gamarra-Torres, O. (2020). Tecnologías para el tratamiento de aguas con radiación solar para el desarrollo sustentable: Una Revisión. *Revista de Investigación de Agroproducción Sustentable*, 4(3), 81-95. <https://doi.org/10.25127/aps.20203.671>
- Reátegui Arrivasplata, C. R. (2019). Desarrollo de un sistema de monitoreo y control micro climático en apoyo al cultivo de arándanos en invernadero en la ciudad de Caraz,

- Departamento de Áncash [Universidad Tecnológica del Perú]. En *Universidad Tecnológica del Perú*. <https://hdl.handle.net/20.500.12867/2878>
- Santos, E. (2021, mayo 3). *Medidor de CO2 con Arduino*. Medidor de CO2. <https://medidordeco2.info/medidor-de-co2-arduino/>
- Sarker, V. K., Queralta, J. P., Gia, T. N., Tenhunen, H., & Westerlund, T. (2019). A survey on LoRa for IoT: Integrating Edge Computing. *2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019*, 295-300. <https://doi.org/10.1109/FMEC.2019.8795313>
- Satyanarayanan, M. (2017). The emergence of Edge Computing. *Computer*, *50*(1), 30-39. <https://doi.org/10.1109/MC.2017.9>
- Satyanarayanan, M., Bahl, P., Cáceres, R., & Davies, N. (2009). The case for VM-based cloudlets in mobile Computing. *IEEE Pervasive Computing*, *8*(4), 14-23. <https://doi.org/10.1109/MPRV.2009.82>
- Sharma, N., Acharya, S., Kumar, K., Singh, N., & Chaurasia, O. P. (2018). Hydroponics as an advanced technique for vegetable production: An overview. *Journal of Soil and Water Conservation*, *17*(4), 371. <https://doi.org/10.5958/2455-7145.2018.00056.5>
- Sigma Electrónica. (s. f.). *SEN0159* . Sigma Electrónica. Recuperado 6 de septiembre de 2022, a partir de <https://www.sigmaelectronica.net/producto/sen0159/>
- Siliceo. (2020). ▷ *Router repetidor de WiFi, cómo añadir un router Open-WRT a una antena WiFi con cable USB*. -. Siliceo. <https://www.siliceo.es/router-repetidor-de-wifi-como-anadir-un-router-open-wrt-a-una-antena-wifi-con-cable-usb/>
- Tasmota. (s. f.). *Tasmota Documentation* . Tasmota. Recuperado 11 de septiembre de 2022, a partir de <https://tasmota.github.io/docs/>
- ThingsBoard Edge | ThingsBoard*. (s. f.). Recuperado 4 de septiembre de 2022, a partir de <https://thingsboard.io/products/thingsboard-Edge/>
- tp-link. (s. f.). *CPE510 | CPE de Exterior 5GHz 300Mbps 13dBi*. TP-Link Argentina. Recuperado 11 de septiembre de 2022, a partir de <https://www.tp-link.com/ar/business-networking/outdoor-radio/cpe510/>
- Tp-link. (2020). *Como Configurar la topologia CPE AP + CPE Cliente con éxito* . <https://www.tp-link.com/ar/support/faq/2575/>

- Triana Useche, J. C., & Rodríguez Leguizamo, R. E. (2018). *Prototipo de solución Iot con tecnología Lora en monitoreo de cultivos agrícolas* - hdl:11349/13388 [Universidad Distrital Francisco José de Caldas]. <http://hdl.handle.net/11349/13388>
- Ubidots. (2022). Ubidots. <https://www.ubidots.com/platform/>
- UNODC. (s. f.). *Manual para el Productor El cultivo de las Hortalizas*. Recuperado 4 de septiembre de 2022, a partir de d2b5e9c3-fb6f-4975-b43c-ef6dd3906974/enhanced-reader.html?openApp&pdf=https%3A%2F%2Fwww.unodc.org%2Fdocuments%2Fbolivia%2FDIM_Manual_de_cultivo_de_hortalizas.pdf
- Wang, T., Wang, L., Yan, P., Popli, R., Rani, P., & Kumar, R. (2022). Application of embedded Linux in the design of Internet of Things gateway. *Journal of Intelligent Systems*, 31(1), 1014-1023. <https://doi.org/10.1515/JISYS-2021-0208>
- Wang, X. F., Wang, Z., Zhang, S. W., & Shi, Y. (2015). *Monitoring and Discrimination of Plant Disease and Insect Pests based on agricultural IOT*. 112-115. <https://doi.org/10.2991/ICITMI-15.2015.21>
- Zhengzhou Winsen Electronics Technologia. (s. f.). Intelligent Infrared CO2 Module (Model: MH-Z19B) . *Winson*. Recuperado 11 de septiembre de 2022, a partir de www.winsen-sensor.com

Anexo A. Código implementado

```
//CONFIGURACIÓN NODE-RED
// Root myDeserializedClass =
JsonConvert.DeserializeObject<List<Root>>(myJsonResponse);
public class AngularTheme
{
    public string primary { get; set; }
    public string accents { get; set; }
    public string warn { get; set; }
    public string background { get; set; }
    public string palette { get; set; }
}

public class BaseColor
{
    public string @default { get; set; }
    public string value { get; set; }
    public bool edited { get; set; }
}

public class BaseFont
{
    public string value { get; set; }
}

public class BirthMsg
{
}

public class CloseMsg
{
}

public class ColorForValue
{
    public string color { get; set; }
    public string value { get; set; }
    public string valueType { get; set; }
}

public class CustomTheme
{
    public string name { get; set; }
    public string @default { get; set; }
    public string baseColor { get; set; }
    public string baseFont { get; set; }
    public bool reset { get; set; }
}

public class DarkTheme
```

```
{
    public string @default { get; set; }
    public string baseColor { get; set; }
    public string baseFont { get; set; }
    public bool edited { get; set; }
    public bool reset { get; set; }
}

public class EventOption
{
    public string label { get; set; }
    public string @event { get; set; }
}

public class GroupBackgroundColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class GroupBorderColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class GroupTextColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class In
{
    public int x { get; set; }
    public int y { get; set; }
    public List<Wire> wires { get; set; }
}

public class LightTheme
{
    public string @default { get; set; }
    public string baseColor { get; set; }
    public string baseFont { get; set; }
    public bool edited { get; set; }
    public bool reset { get; set; }
}

public class Out
{
    public int x { get; set; }
    public int y { get; set; }
}
```

```
        public List<Wire> wires { get; set; }
    }

    public class PageBackgroundColor
    {
        public string value { get; set; }
        public bool edited { get; set; }
    }

    public class PageSidebarBackgroundColor
    {
        public string value { get; set; }
        public bool edited { get; set; }
    }

    public class PageTitlebarBackgroundColor
    {
        public string value { get; set; }
        public bool edited { get; set; }
    }

    public class Prop
    {
        public string p { get; set; }
        public string vt { get; set; }
    }

    public class Root
    {
        public string id { get; set; }
        public string type { get; set; }
        public string label { get; set; }
        public bool disabled { get; set; }
        public string info { get; set; }
        public List<object> env { get; set; }
        public string name { get; set; }
        public string category { get; set; }
        public List<In> @in { get; set; }
        public List<Out> @out { get; set; }
        public Theme theme { get; set; }
        public Site site { get; set; }
        public string group { get; set; }
        public int? order { get; set; }
        public object width { get; set; }
        public object height { get; set; }
        public string link { get; set; }
        public string icon { get; set; }
        public string target { get; set; }
        public bool? hidden { get; set; }
        public string tab { get; set; }
        public bool? disp { get; set; }
        public bool? collapse { get; set; }
    }
}
```

```
public string className { get; set; }
public bool? d { get; set; }
public string broker { get; set; }
public string port { get; set; }
public string clientid { get; set; }
public bool? autoConnect { get; set; }
public bool? usetls { get; set; }
public bool? compatmode { get; set; }
public string protocolVersion { get; set; }
public string keepalive { get; set; }
public bool? cleansession { get; set; }
public string birthTopic { get; set; }
public string birthQos { get; set; }
public string birthRetain { get; set; }
public string birthPayload { get; set; }
public BirthMsg birthMsg { get; set; }
public string closeTopic { get; set; }
public string closeQos { get; set; }
public string closePayload { get; set; }
public CloseMsg closeMsg { get; set; }
public string willTopic { get; set; }
public string willQos { get; set; }
public string willPayload { get; set; }
public WillMsg willMsg { get; set; }
public string sessionExpiry { get; set; }
public string phone { get; set; }
public string token { get; set; }
public string cert { get; set; }
public string key { get; set; }
public string ca { get; set; }
public string certname { get; set; }
public string keyname { get; set; }
public string caname { get; set; }
public string servername { get; set; }
public bool? verifyservercert { get; set; }
public string z { get; set; }
public string filename { get; set; }
public string format { get; set; }
public bool? chunk { get; set; }
public bool? sendError { get; set; }
public double? x { get; set; }
public double? y { get; set; }
public List<List<string>> wires { get; set; }
public string property { get; set; }
public string propertyType { get; set; }
public List<Rule> rules { get; set; }
public string checkall { get; set; }
public bool? repair { get; set; }
public object outputs { get; set; }
public bool? appendNewline { get; set; }
public bool? createDir { get; set; }
public string overwriteFile { get; set; }
```

```
public string action { get; set; }
public bool? pretty { get; set; }
public string repeat { get; set; }
public string crontab { get; set; }
public bool? once { get; set; }
public object onceDelay { get; set; }
public string topic { get; set; }
public string payload { get; set; }
public string payloadType { get; set; }
public string from { get; set; }
public string to { get; set; }
public bool? reg { get; set; }
public string func { get; set; }
public int? noerr { get; set; }
public string pauseType { get; set; }
public string timeout { get; set; }
public string timeoutUnits { get; set; }
public string rate { get; set; }
public string nbRateUnits { get; set; }
public string rateUnits { get; set; }
public string randomFirst { get; set; }
public string randomLast { get; set; }
public string randomUnits { get; set; }
public bool? drop { get; set; }
public string gap { get; set; }
public string start { get; set; }
public string inout { get; set; }
public bool? storeOutMessages { get; set; }
public bool? fwdInMessages { get; set; }
public bool? resendOnRefresh { get; set; }
public string templateScope { get; set; }
public bool? mydate { get; set; }
public bool? myymd { get; set; }
public bool? myyear { get; set; }
public bool? mymonth { get; set; }
public bool? mymonthn { get; set; }
public bool? mydom { get; set; }
public bool? mydoy { get; set; }
public bool? myday { get; set; }
public bool? myhourpm { get; set; }
public bool? myhour { get; set; }
public bool? mytime { get; set; }
public bool? mytimes { get; set; }
public bool? myminute { get; set; }
public bool? myminutes { get; set; }
public bool? mysecond { get; set; }
public bool? mymillis { get; set; }
public bool? myepoch { get; set; }
public bool? myrawdate { get; set; }
public bool? mypm { get; set; }
public string initialize { get; set; }
public string finalize { get; set; }
```

```
public List<object> libs { get; set; }
public string input { get; set; }
public string inputType { get; set; }
public string inTz { get; set; }
public string adjAmount { get; set; }
public string adjType { get; set; }
public string adjDir { get; set; }
public string locale { get; set; }
public string output { get; set; }
public string outputType { get; set; }
public string outTz { get; set; }
public List<string> outputLabels { get; set; }
public bool? active { get; set; }
public bool? tosidebar { get; set; }
public bool? console { get; set; }
public bool? tostatus { get; set; }
public string complete { get; set; }
public string targetType { get; set; }
public string wtype { get; set; }
public string lon { get; set; }
public string lat { get; set; }
public string city { get; set; }
public string country { get; set; }
public string language { get; set; }
public string gtype { get; set; }
public string title { get; set; }
public int? min { get; set; }
public string max { get; set; }
public List<string> colors { get; set; }
public string seg1 { get; set; }
public string seg2 { get; set; }
public string qos { get; set; }
public string datatype { get; set; }
public bool? nl { get; set; }
public bool? rap { get; set; }
public int? inputs { get; set; }
public string method { get; set; }
public string ret { get; set; }
public string paytoqs { get; set; }
public string url { get; set; }
public string tls { get; set; }
public bool? persist { get; set; }
public string proxy { get; set; }
public string authType { get; set; }
public bool? senderr { get; set; }
public bool? allowrate { get; set; }
public string statusVal { get; set; }
public string statusType { get; set; }
public string position { get; set; }
public string displayTime { get; set; }
public string highlight { get; set; }
public bool? sendall { get; set; }
```

```
public string ok { get; set; }
public string cancel { get; set; }
public bool? raw { get; set; }
public string voice { get; set; }
public bool? always { get; set; }
public string labelPlacement { get; set; }
public string labelAlignment { get; set; }
public List<ColorForValue> colorForValue { get; set; }
public bool? allowColorForValueInMessage { get; set; }
public string shape { get; set; }
public bool? showGlow { get; set; }
public string chartType { get; set; }
public string legend { get; set; }
public string xformat { get; set; }
public string interpolate { get; set; }
public string nodata { get; set; }
public bool? dot { get; set; }
public string ymin { get; set; }
public string ymax { get; set; }
public string removeOlder { get; set; }
public string removeOlderPoints { get; set; }
public string removeOlderUnit { get; set; }
public int? cutout { get; set; }
public bool? useOneColor { get; set; }
public bool? useUTC { get; set; }
public bool? useDifferentColor { get; set; }
public List<Prop> props { get; set; }
public string init { get; set; }
public string step { get; set; }
public string lower { get; set; }
public string upper { get; set; }
public string mode { get; set; }
public string auth { get; set; }
public bool? septopics { get; set; }
public string topi { get; set; }
public string count { get; set; }
public object round { get; set; }
public string mult { get; set; }
public bool? reduce { get; set; }
public string minin { get; set; }
public string maxin { get; set; }
public string minout { get; set; }
public string maxout { get; set; }
public string startDay { get; set; }
public int? refresh { get; set; }
public List<string> devices { get; set; }
public bool? singleOff { get; set; }
public bool? onlySendChange { get; set; }
public bool? customPayload { get; set; }
public bool? eventMode { get; set; }
public List<EventOption> eventOptions { get; set; }
public bool? sendTopic { get; set; }
```

```
public string customContextStore { get; set; }
public List<string> links { get; set; }
public string retain { get; set; }
public string respTopic { get; set; }
public string contentType { get; set; }
public string userProps { get; set; }
public string correl { get; set; }
public string expiry { get; set; }
public bool? passthru { get; set; }
public string tooltip { get; set; }
public string color { get; set; }
public string bgcolor { get; set; }
public string topicType { get; set; }
public string outs { get; set; }
public string layout { get; set; }
public string op1 { get; set; }
public string op2 { get; set; }
public string op1type { get; set; }
public string op2type { get; set; }
public string duration { get; set; }
public bool? extend { get; set; }
public bool? overrideDelay { get; set; }
public string units { get; set; }
public string reset { get; set; }
public string bytopic { get; set; }
public bool? l { get; set; }
public string decouple { get; set; }
public string style { get; set; }
public string onvalue { get; set; }
public string onvalueType { get; set; }
public string onicon { get; set; }
public string oncolor { get; set; }
public string offvalue { get; set; }
public string offvalueType { get; set; }
public string officon { get; set; }
public string offcolor { get; set; }
public bool? animate { get; set; }
}

public class Rule
{
    public string t { get; set; }
    public string p { get; set; }
    public string pt { get; set; }
    public string to { get; set; }
    public string tot { get; set; }
    public string from { get; set; }
    public string fromt { get; set; }
}

public class Site
{
```

```
    public string name { get; set; }
    public string hideToolbar { get; set; }
    public string allowSwipe { get; set; }
    public string lockMenu { get; set; }
    public string allowTempTheme { get; set; }
    public string dateFormat { get; set; }
    public Sizes sizes { get; set; }
}

public class Sizes
{
    public int sx { get; set; }
    public int sy { get; set; }
    public int gx { get; set; }
    public int gy { get; set; }
    public int cx { get; set; }
    public int cy { get; set; }
    public int px { get; set; }
    public int py { get; set; }
}

public class Theme
{
    public string name { get; set; }
    public LightTheme lightTheme { get; set; }
    public DarkTheme darkTheme { get; set; }
    public CustomTheme customTheme { get; set; }
    public ThemeState themeState { get; set; }
    public AngularTheme angularTheme { get; set; }
}

public class ThemeState
{
    [JsonProperty("base-color")]
    public BaseColor BaseColor { get; set; }

    [JsonProperty("page-titlebar-backgroundColor")]
    public PageTitlebarBackgroundColor PageTitlebarBackgroundColor {
get; set; }

    [JsonProperty("page-backgroundColor")]
    public PageBackgroundColor PageBackgroundColor { get; set; }

    [JsonProperty("page-sidebar-backgroundColor")]
    public PageSidebarBackgroundColor PageSidebarBackgroundColor {
get; set; }

    [JsonProperty("group-textColor")]
    public GroupTextColor GroupTextColor { get; set; }

    [JsonProperty("group-borderColor")]
    public GroupBorderColor GroupBorderColor { get; set; }
}
```

```
[JsonProperty("group-backgroundColor")]
public GroupBackgroundColor GroupBackgroundColor { get; set; }

[JsonProperty("widget-textColor")]
public WidgetTextColor WidgetTextColor { get; set; }

[JsonProperty("widget-backgroundColor")]
public WidgetBackgroundColor WidgetBackgroundColor { get; set; }

[JsonProperty("widget-borderColor")]
public WidgetBorderColor WidgetBorderColor { get; set; }

[JsonProperty("base-font")]
public BaseFont BaseFont { get; set; }
}

public class WidgetBackgroundColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class WidgetBorderColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class WidgetTextColor
{
    public string value { get; set; }
    public bool edited { get; set; }
}

public class WillMsg
{
}

public class Wire
{
    public string id { get; set; }
    public int port { get; set; }
}

//CONFIGURACIÓN CSS WEB UI
<style>
    root,
    a,
    button,
    p,
    div {
```

```
    -webkit-user-select: none !important;
}

body {
    background: rgba(240, 242, 245, 1);

    -webkit-touch-callout: none !important;
}

body.nr-dashboard-theme md-toolbar {
background-color: rgba(147, 193, 35, 1) !important;
color: #FFFFFF;
}

body.nr-dashboard-theme md-content md-card {
    background-color: transparent !important;
    color: #000;
}

ui-card-panel {
    background-color: rgba(255, 255, 255, 1) !important;
    border-radius: 10px !important;
}

#Homey_Dashboard_Navigation {
    background-color: transparent !important;
    border-radius: 10px !important;
}

.md-card.md-default-theme,
md-card {
    border-radius: 10px;
}

.nr-dashboard-switch.ng-scope {
    background-color: rgba(255, 255, 255, .2) !important;
}

.nr-dashboard-switch.ng-scope:hover {
    background-color: rgba(255, 255, 255, .5) !important;
}

.nr-dashboard-theme .nr-dashboard-button .md-button {
    background-color: rgba(147, 193, 35, .8);
}

.md-button {
    border-radius: 10px;
}

.nr-dashboard-cardpanel layout-column {
    background-color: rgba(255, 255, 255, 0.2) !important;
}
```

```
        background-color: transparent !important;
        border-radius: 10px !important;
    }

    .nr-dashboard-theme ui-card-panel {
        border: none;
    }

    .nr-dashboard-template {
        overflow-y: visible;
    }

    .nr-dashboard-theme md-content md-card {
        background-color: transparent !important;
    }

    .nr-dashboard-theme ui-card-panel p.nr-dashboard-cardtitle {
        color: rgba(125, 122, 125, 0.5);
    }
</style>

<script>
    document.ontouchmove = function (e) {
        e.preventDefault();
    }
</script>
```

Anexo B. Especificaciones técnicas de los dispositivos utilizados.

- ▶ Sensor DHT21 – AM2:

https://drive.google.com/file/d/0BzaKjvCRihgbYVd5NU1zcGY2Qk0/view?resourcekey=0-EIYz8e4djgUHe3KM_iYJcw

- ▶ Sensor FC-28:

<https://www.ti.com/lit/ds/symlink/lm393-n.pdf>

- ▶ Sensor MHZ-19B:

https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf

- ▶ SBC Tanix TX6:

<https://www.tvbox-manufacturer.com/product/tanix-tx6-tv-box/>

- ▶ NodeMCU LUA:

https://www.elecrow.com/download/ESP8266_Specifications_English.pdf

- ▶ Son off BASICR2:

<https://sonoff.tech/product/diy-smart-switch/basicr2/>

- ▶ Sensor de Humedad del suelo anticorrosivo HD-38

<https://electronilab.co/tienda/sensor-de-humedad-de-suelo-anticorrosivo-higrometro/>

- ▶ Antena CPE 510:

<https://www.tridimas.gr/pdf/cpe-510.pdf>

- ▶ Antena CPE 210:

https://www.linkdataguard.com/datasheets/Wireless/CPE210_V1_Datasheet.pdf