

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN PROTOTIPO DE AGRICULTURA
INTELIGENTE BASADO EN IOT PARA LA EMPRESA
“ALIMENTOS ESPECIALES PROCESADOS C.A.”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

JOSÉ OSWALDO DÁVILA CHAUCANÉS

jose.davila01@epn.edu.ec

DIRECTOR: Ph.D. DIANA CECILIA YACCHIREMA VARGAS

diana.yacchirema@epn.edu.ec

QUITO, OCTUBRE 2023

CERTIFICACIÓN DEL DIRECTOR

Como director del trabajo de titulación 'Desarrollo de un prototipo de agricultura inteligente basado en IoT para la empresa "Alimentos Especiales Procesados C.A." (ALEPROCA)' desarrollado por José Oswaldo Dávila Chaucañés, estudiante de la carrera de Ingeniería en Sistemas Informáticos y de Computación, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

Ph.D. Diana Cecilia Yacchirema Vargas

DIRECTOR DEL PROYECTO

DECLARACIÓN DE AUTORÍA

Yo, José Oswaldo Dávila Chaucañés, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Dávila Chaucañés José Oswaldo

DEDICATORIA

A mis padres, Oswaldo Dávila y Rocío Chaucañés, quienes me brindaron un apoyo incondicional a lo largo de esta travesía, no solo ofreciéndome sustento económico, sino también compartiendo su conocimiento y sabiduría. A mi padre, por su incansable ayuda en mis estudios desde el colegio, y a mi madre, por sus valiosos consejos y esas conversaciones que siempre me han ayudado a seguir el camino correcto.

A mis abuelos, Oswaldo Dávila y Lady Borja, que siempre estuvieron a mi lado, apoyándome, cuidándome y compartiendo sus vivencias y consejos. Siempre fueron mis defensores, ya sea en casa o en el colegio, gracias a su inmensa bondad.

A mis tíos, Katty Dávila, Paul Dávila y Mercedes Álvarez, por su apoyo constante y palabras de aliento. Y a mi prima/hermana Estefanía Huertas, con quien he compartido más que solo un lazo familiar, siendo una presencia esencial en mi vida.

A mi esposa, Dayana Rosero, quien ha sido mi pilar en este camino que hemos buscado y construido juntos desde hace mucho tiempo. Cada desafío y alegría que hemos compartido ha contribuido a alcanzar este logro académico. A mi hija, Mía Dávila, por su amor sincero y humildad, que ha iluminado mis días.

A mis hermanos, Santiago y Alejo Dávila, siempre dispuestos a estar a mi lado, ya sea para discutir, jugar o cuidar de su sobrina. Su apoyo ha sido esencial en esta etapa de mi vida.

También quiero agradecer a mis amigos, quienes me respaldaron cuando decidí seguir esta carrera y me ofrecieron su apoyo y guía en momentos cruciales.

Por último, mi agradecimiento a la Dra. Yacchirema por su asesoramiento y dirección en este trabajo de titulación. Su conocimiento y apoyo han sido esenciales para la culminación de este proyecto.

José Dávila

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	10
1.1 Planteamiento del problema.....	10
1.2 Objetivos.....	11
1.2.1 Objetivo General.....	11
1.2.2 Objetivos Específicos	11
1.3 Caracterización de la empresa.....	11
1.4 Información de ALEPROCA.....	11
1.4.1 Misión	11
1.4.2 Visión.....	12
1.4.3 Estructura orgánica.....	12
1.5 Marco Teórico.....	14
1.5.1 Agricultura.....	14
1.5.2 Agricultura de precisión	14
1.5.3 Internet de las cosas (IoT)	15
1.5.4 Protocolos de aplicación utilizados en IoT	16
1.5.5 Arquitecturas de referencia en IoT.....	17
2. METODOLOGÍA	20
2.1 Investigación constructivista de Oyegoke	20
2.2 Diseño e implementación del prototipo de agricultura inteligente basado en IoT.....	21
2.3 Selección de la arquitectura IoT	21
2.4 Selección del protocolo IoT	22
2.5 Arquitectura del prototipo inteligente basado en IoT	23
2.5.1 Capa de percepción.....	23
2.5.2 Capa de red.....	24
2.5.3 Capa de aplicación	24
2.6 Implementación del prototipo IoT	24
2.6.1 Implementación del nodo IoT.....	24
2.6.2 Configuración del Broker MQTT	36
2.7 Desarrollo del aplicativo WEB	41

2.7.1 Base de datos.....	41
2.7.2 Aplicativo	41
2.7.3 Autenticación y Registro	42
2.7.4 Monitorización de datos.....	43
3. DESPLIEGUE Y VALIDACIÓN DEL PROTOTIPO	49
3.1 Despliegue.....	49
3.2 Validación	52
4. CONCLUSIONES Y RECOMENDACIONES	60
4.1 Conclusiones	60
4.2 Recomendaciones	62
5. REFERENCIAS BIBLIOGRÁFICAS	63

ÍNDICE DE FIGURAS

Figura 1. Estructura Orgánica ALEPROCA.....	13
Figura 2. Beneficios de IoT en los cultivos.....	16
Figura 3. Arquitecturas de referencia IoT.....	18
Figura 4. Fases del proyecto.....	20
Figura 5. Arquitectura del PAI-IoT.....	23
Figura 6. Módulo ESP8266 Wi-Fi.....	25
Figura 7. Sensor de Temperatura y Humedad DHT22.....	25
Figura 8. Pila recargable 4000 mA.....	25
Figura 9. Módulo cargador TP4056	26
Figura 10. Panel solar 6V	26
Figura 11. Diodo 1N4007	27
Figura 12. Conexión del módulo DHT22 a la placa ESP8266 WIFI	27
Figura 13. Instanciar Librerías	29
Figura 14. Definición de Variables	30
Figura 15. Conexión Wi-Fi y configuración IP estática.....	30
Figura 16. Configuración del cliente MQTT.....	31
Figura 17. Función conectar_MQTT ()	32
Figura 18. Función setup ()	33
Figura 19. Función `loop ()`	34
Figura 20. Conversión de tipo float a tipo String	34
Figura 21. Publicación de tópicos	35
Figura 22. Desconexión y forzar reposo	36
Figura 23. Comando para la configuración del servicio 'Mosquitto'	37
Figura 24. Configuración del archivo mosquitto.conf	37
Figura 25. Estado del servicio mosquitto	38
Figura 26. Instalación MQTT en el bróker.....	38
Figura 27. Script Suscriptor en Python.....	39
Figura 28. Acciones función on_connect	40
Figura 29. Generación del bash para la obtención de datos	41
Figura 30. Inserción en la base de datos los valores de temperatura y humedad.....	41
Figura 31. Pantalla de Inicio de Sesión.....	42
Figura 32. Pantalla de Registro.....	43
Figura 33. Selección del invernadero.....	44
Figura 34. Panel de gráficos de los invernaderos	45
Figura 35. Panel de datos históricos de los invernaderos	46
Figura 36. Panel de monitoreo de temperatura y humedad	47
Figura 37. Panel de monitoreo de temperatura.....	47
Figura 38. Panel de monitoreo de humedad	48
Figura 39. Ensamble inicial del prototipo	50
Figura 40. Ensamble final del prototipo.....	50
Figura 41. Instalación en el invernadero versión inferior del prototipo	51
Figura 42. Instalación en el invernadero versión superior del prototipo	51
Figura 43. Resultados pregunta 1	53
Figura 44. Resultados pregunta 2.....	54
Figura 45. Resultados pregunta 3.....	55
Figura 46. Resultados pregunta 4.....	56
Figura 47. Resultados pregunta 5.....	57
Figura 48. Resultados pregunta 6.....	58
Figura 49. Resultados pregunta 7.....	58
Figura 50. Resultados pregunta 8.....	59

RESUMEN

En el contexto agrícola de nuestro país, la mayoría de los agricultores son pequeños y las empresas carecen de herramientas tecnológicas para llevar a cabo cultivos eficientes, ya sea por falta de conocimiento o por los altos costos de implementación. El Internet de las Cosas (IoT, *Internet of Things* por sus siglas en inglés) ha revolucionado el sector agrícola al proporcionar herramientas avanzadas para la monitorización y gestión de cultivos. Además, la automatización habilitada por IoT optimiza la eficiencia de los sistemas de riego y reduce el desperdicio de recursos, contribuyendo a la sostenibilidad y productividad en la agricultura moderna.

Este trabajo propone el diseño de un prototipo basado en IoT para la monitorización de invernaderos, específicamente para la empresa “Alimentos Especiales Procesados C.A (ALEPROCA)”. El sistema emplea dispositivos IoT para medir la temperatura y humedad relativa en cada invernadero, y presenta los datos de manera gráfica a través de un portal web.

El objetivo principal es lograr un monitoreo preciso de los niveles de temperatura y humedad, con el propósito de mejorar la producción agrícola y facilitar la toma de decisiones por parte de los responsables. Para el desarrollo e implementación del prototipo se aplicó una arquitectura IoT de tres capas. A fin de promover prácticas más sustentables y aprovechar de manera eficiente los recursos naturales se implementaron paneles solares que permiten la autosostenibilidad del prototipo.

Para comprobar la efectividad de la propuesta, el prototipo fue implementado en dos invernaderos de ALEPROCA. Se recopiló el *feedback* de los encargados de estos espacios a través de encuestas, logrando así evaluar la funcionalidad, eficiencia y satisfacción que el sistema aporta al contexto agrícola de la empresa.

Los resultados preliminares revelan que el prototipo permite el monitoreo preciso de los niveles de temperatura y humedad, reduciendo así la necesidad de mano de obra y el consumo de agua, y promoviendo de esta manera prácticas más sustentables que permiten aprovechar de manera eficiente los recursos naturales y preservar el entorno ambiental.

ABSTRACT

Within the agricultural context of our country, most farmers are small-scale, and companies lack the technological tools to carry out efficient farming, whether due to a lack of knowledge or the high costs of implementation.

The Internet of Things (IoT) has revolutionized the agricultural sector by offering advanced tools for crop monitoring and management. Furthermore, the automation enabled by IoT enhances the efficiency of irrigation systems and reduces resource wastage, contributing to sustainability and productivity in modern agriculture.

This study proposes the design of an IoT-based prototype for greenhouse monitoring, specifically for the company "Alimentos Especiales Procesados C.A (ALEPROCA)." The system utilizes IoT devices to measure temperature and relative humidity in each greenhouse, presenting the data graphically through a web portal.

The primary objective is to achieve accurate monitoring of temperature and humidity levels, aiming to enhance agricultural production and facilitate decision-making by those in charge. For the prototype's development and implementation, a three-layer IoT architecture was applied. To promote more sustainable practices and efficiently utilize natural resources, solar panels were integrated, ensuring the prototype's self-sustainability.

To verify the proposal's effectiveness, the prototype was implemented in two ALEPROCA greenhouses. Feedback was gathered from the in-charge personnel via surveys, thereby evaluating the system's functionality, efficiency, and satisfaction it brings to the company's agricultural context.

Preliminary results indicate that the prototype allows for precise monitoring of temperature and humidity levels, thereby reducing labor requirements and water consumption. This promotes more sustainable practices, ensuring efficient resource utilization and environmental preservation.

1. INTRODUCCIÓN

1.1 Planteamiento del problema

La compañía ALEPROCA se encuentra situada en la parroquia de Pifo, en la parte oriental de Quito. Su principal actividad se centra en el cultivo de una variedad de productos en invernaderos, los cuales requieren condiciones ambientales cuidadosamente controladas, especialmente en términos de temperatura y humedad relativa. El propósito principal de ALEPROCA es mejorar tanto la calidad como la producción de los cultivos cosechados. Actualmente, los ingenieros agrícolas de la empresa realizan mediciones manuales y periódicas de estas condiciones en cada invernadero. Este método utilizado presenta limitaciones significativas como, por ejemplo, para medir la temperatura, solo disponen de un termómetro básico ubicado en uno de los siete invernaderos, lo que implica que las mediciones se generalizan y no proporcionan una visión precisa de las condiciones en cada área de cultivo. Por otra parte, lo que respecta a la humedad relativa, no cuentan con ningún tipo de control o medición, lo que representa una falta de información crucial para el correcto manejo de los cultivos.

Además, en la actualidad, si se requiere ajustar la temperatura o la humedad relativa, los responsables de cada invernadero deben realizar ajustes manuales, como levantar una cortina plástica que cubre el perímetro de los invernaderos o controlar el riego. Estas acciones, basadas en estimaciones y sin datos precisos, resultan en un consumo excesivo de agua para humidificar el ambiente y requieren mucho tiempo y esfuerzo para manejar la cortina plástica de manera adecuada.

Es importante destacar que el cambio de temperatura en un invernadero tiene una estrecha relación con la humedad relativa. Un incremento en la temperatura tiene como consecuencia que el aire pueda retener una mayor cantidad de humedad, lo que puede tener un impacto desfavorable en el rendimiento de los cultivos. El exceso de humedad disminuye la capacidad de las plantas para transpirar, reduce la absorción de nutrientes, dificulta el crecimiento de los cultivos, afecta la polinización y promueve la proliferación de plagas.

Con el objetivo de enfrentar estos retos y asegurar la excelencia de los cultivos orgánicos, se sugiere la introducción y aplicación de un prototipo basado en IoT, el

cual posibilite la supervisión en tiempo real de los niveles de temperatura y humedad en los invernaderos de ALEPROCA. De esta manera, se busca asegurar un control óptimo de dichos factores para garantizar la calidad de los cultivos. Esta solución busca proporcionar datos precisos y actualizados sobre las condiciones ambientales en cada área de cultivo, permitiendo un control más eficiente y reduciendo el consumo de recursos, como el agua, así como el tiempo y el esfuerzo necesarios para ajustar manualmente las condiciones.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un prototipo de agricultura inteligente basado en IoT para la empresa ALEPROCA. (PAI-IoT)

1.2.2 Objetivos Específicos

- Revisar y seleccionar la arquitectura IoT que se utilizara como base para la implementación del prototipo.
- Diseñar el prototipo IoT para la monitorización de invernaderos con base a la arquitectura seleccionada.
- Implementar y desplegar el prototipo IoT en un entorno real.
- Validar el prototipo implementado.

1.3 Caracterización de la empresa

La iniciativa de establecer ALEPROCA tuvo su origen en 1991 con el propósito de desarrollar una empresa dedicada al cultivo y producción de hortalizas, con enfoque en la implementación de tecnologías agrícolas avanzadas y prácticas sostenibles. A través de este subcapítulo, se explorarán los aspectos esenciales que definen a la empresa, sentando las bases para comprender su misión y visión, las cuales serán abordadas en el siguiente apartado. [1]

1.4 Información de ALEPROCA

1.4.1 Misión

“Somos una empresa líder e innovadora a nivel nacional, dedicada a la producción, comercialización y distribución de hortalizas y frutas sanas con valor agregado, buscando que nuestro pilar de crecimiento sea la calidad de nuestro producto y servicio, fomentando crecimiento, desarrollo y oportunidad laboral en el Ecuador” [1].

1.4.2 Visión

“Buscamos mantenernos como una empresa líder en el mercado de alimentos, ofreciendo a nuestros clientes productos sanos y de excelente calidad. Producimos nuestros alimentos bajo los más estrictos controles de calidad y en armonía con el medio ambiente, logrando el desarrollo de nuestro personal y contribuyendo a la comunidad” [1].

1.4.3 Estructura orgánica

La estructura orgánica de la empresa está constituida por 11 áreas como se muestra en la Figura 1, el prototipo desarrollado beneficiara directamente al área de sistemas y al área de campo y producción donde se encuentra el jefe de producción de invernadero quien está a cargo de coordinar y administrar todo lo relacionado a los invernaderos.

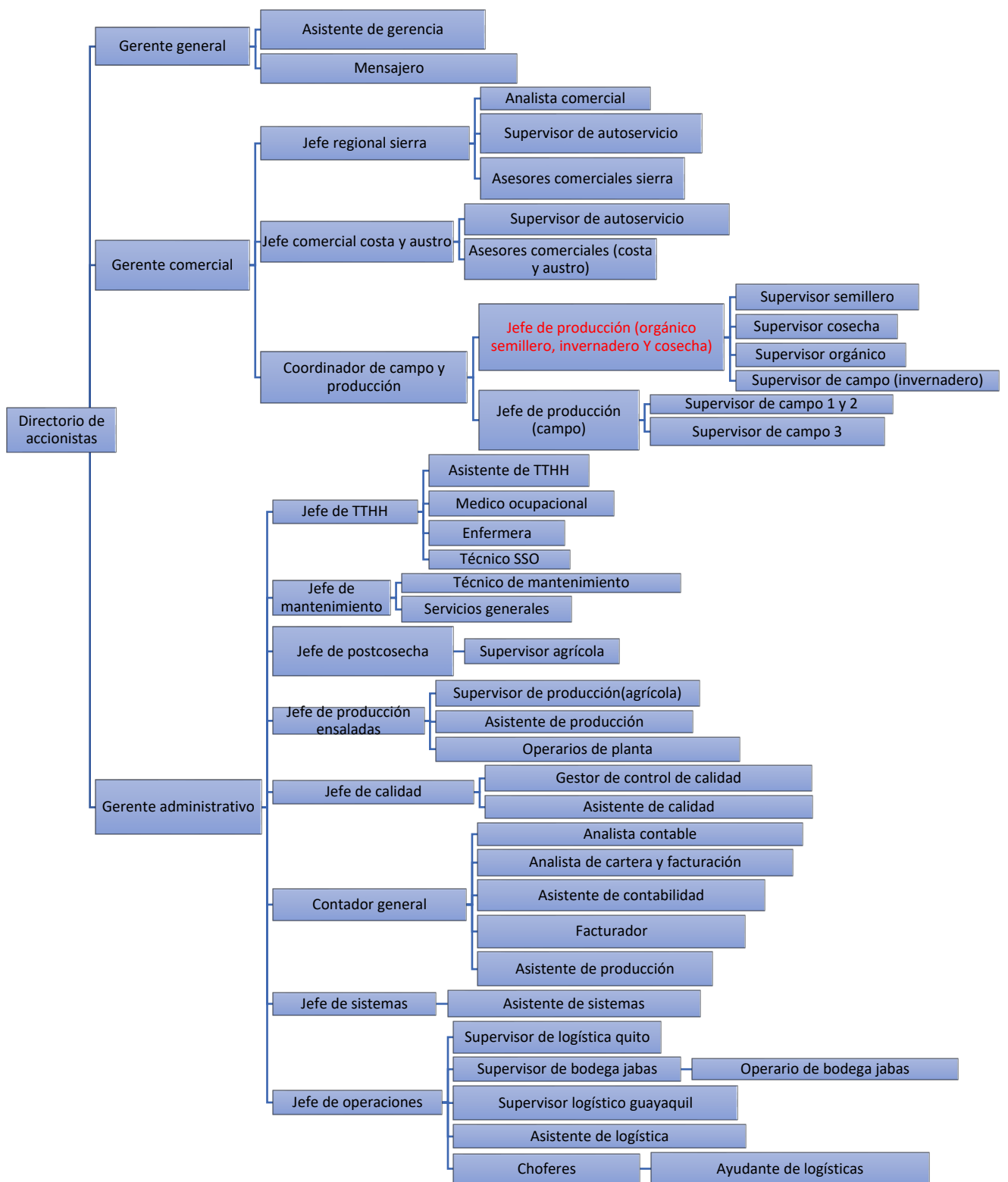


Figura 1. Estructura Orgánica ALEPROCA

1.5 Marco Teórico

1.5.1 Agricultura

De acuerdo con la definición proporcionada por la Real Academia Española, la agricultura engloba el conjunto de técnicas y saberes vinculados al cultivo y trabajo de la tierra [10].

En el ámbito económico, la agricultura se centra en las actividades destinadas al cultivo de la tierra y al tratamiento del suelo para la producción de alimentos. Los agricultores desempeñan labores como la labranza, la fertilización y la recolección de productos. Sin embargo, con el avance tecnológico, el sector agrícola se ha industrializado, lo que ha llevado a la introducción de equipos y maquinaria que han reemplazado en gran medida el trabajo manual, permitiendo incluso una escala masiva de producción [11] que busca optimizar la producción agrícola allanando de esta manera el concepto de agricultura de precisión.

1.5.2 Agricultura de precisión

La agricultura de precisión utiliza los avances tecnológicos actuales para incrementar la eficiencia y minimizar la intervención humana en los cultivos, mitigando los riesgos asociados [2].

La precisión en la agricultura se refiere a la habilidad de definir con exactitud las variables relevantes dentro del cultivo, con el fin de controlarlas y optimizar tanto el crecimiento como la calidad de las plantas. Esto implica tener en cuenta las características específicas de los suelos y los cultivos, y aprovechar una amplia gama de datos para tomar decisiones más precisas [2].

La incorporación de tecnologías de información en la agricultura genera efectos beneficiosos en la toma de decisiones y en el rendimiento de los cultivos [3].

La agricultura de precisión se divide en cuatro etapas distintas:

- Adquisición de datos: Se emplean diversos tipos de sensores para recolectar información, como aquellos que miden la humedad, temperatura, flujo y conductividad eléctrica. [3].
- Análisis de datos: Los datos recopilados se procesan para facilitar su interpretación y comprensión, a menudo mediante el mapeo de datos en gráficos [3].

- Toma de decisiones preventivas y de gestión: Con la información recopilada sobre la siembra y con la aplicación de conocimientos agrario, se pueden tomar decisiones óptimas para el manejo del cultivo, como la prevención de plagas o enfermedades, la gestión del riego y el ahorro de recursos [3].
- Monitorización del rendimiento: En esta etapa final, se realiza un seguimiento de la producción, las medidas tomadas y el rendimiento logrado, lo que permite mejorar la experiencia para futuros cultivos [3].

La implementación IoT en la agricultura de precisión ha transformado la recopilación de datos y la toma de decisiones dentro del ámbito agrícola. La conexión y comunicación en tiempo real entre dispositivos y sistemas ha mejorado la gestión de los cultivos, facilitando el intercambio de información entre agricultores de distintas escalas. Gracias al uso de tecnologías como sensores y software, la agricultura de precisión optimiza el uso de recursos, reduce la dependencia de insumos y fuerza laboral en la agricultura, y eleva la calidad de los cultivos. Los dispositivos IoT, como los sensores, capturan datos en tiempo real sobre variables clave como la humedad del suelo y la temperatura, transmitiéndolos a una plataforma centralizada para su análisis. Esto brinda información valiosa para la toma de decisiones agrícolas, permitiendo una mayor automatización y control de los procesos, mejorando el riego y la aplicación de fertilizantes, y fomentando un uso más eficiente de los recursos hídricos, así como la reducción del impacto ambiental.

1.5.3 Internet de las cosas (IoT)

El término IoT, se utiliza para describir la conexión de dispositivos comunes a través de redes, lo que les permite recopilar, transmitir y procesar datos de manera autónoma, sin requerir una interacción humana constante [4].

Dentro del ámbito agrícola, la implementación de IoT tiene como propósito automatizar los procedimientos y técnicas asociados al cultivo, con el fin de alcanzar una mayor eficiencia y efectividad en la producción agrícola. Esto implica la implementación de sensores y dispositivos inteligentes en diversos elementos del entorno agrícola, como invernaderos, sistemas de riego, monitoreo de cultivos, entre otros. Estos dispositivos recopilan datos en tiempo real sobre variables ambientales y condiciones del cultivo, que luego son utilizados para tomar decisiones informadas y optimizar los procesos agrícolas [5].

La Figura 2 exhibe diversos ejemplos de los beneficios obtenidos al aplicar IoT en la agricultura. Estos beneficios abarcan la supervisión y control preciso de variables como temperatura, humedad y calidad del suelo, la detección temprana de plagas y enfermedades, la optimización del riego y la fertilización, la reducción de costos operativos, y la mejora en la calidad y rendimiento de los cultivos. [5]

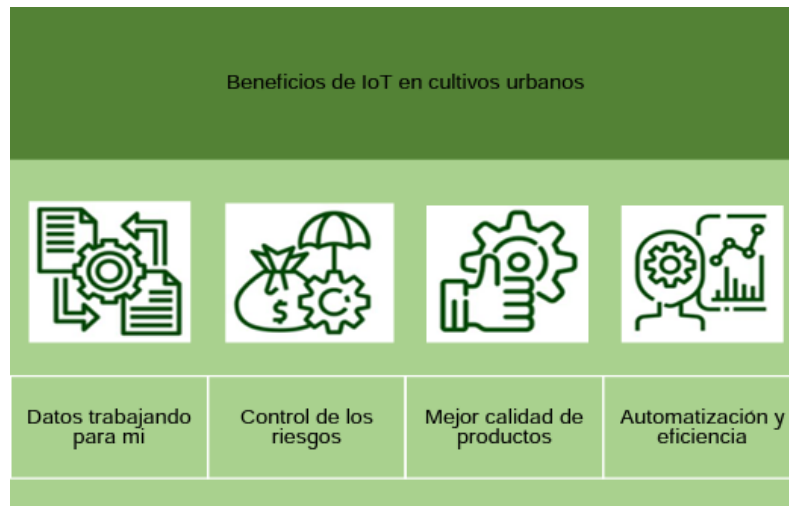


Figura 2. Beneficios de IoT en los cultivos

1.5.4 Protocolos de aplicación utilizados en IoT

En la agricultura de precisión, la elección del protocolo de comunicación adecuado desempeña un papel fundamental para garantizar una transmisión eficiente y confiable de datos entre los dispositivos IoT utilizados en los sistemas agrícolas inteligentes. Dentro del siguiente apartado se analizarán los protocolos más comunes para escoger el que mejor se adapte al proyecto del capítulo 2.

CoAP (*Constrained Application Protocol*)

CoAP es un protocolo ligero diseñado específicamente para dispositivos con recursos limitados, como dispositivos IoT. Según investigaciones previas realizadas en el campo de la agricultura de precisión, CoAP ha demostrado ser útil en entornos con restricciones de energía y ancho de banda, permitiendo la comunicación eficiente entre los dispositivos agrícolas y los sistemas de gestión. Este protocolo es especialmente adecuado para aplicaciones que requieren una transmisión de datos simple y de baja complejidad. Sin embargo, es importante tener en cuenta que CoAP

puede presentar limitaciones en términos de escalabilidad y manejo de mensajes en entornos con una mayor cantidad de dispositivos y tráfico de datos. [6]

MQTT (*Message Queue Telemetry Transport*)

MQTT es un protocolo de mensajería ligero que ha ganado popularidad en diversos campos de aplicación, incluida la agricultura de precisión. MQTT se destaca por su eficiencia, baja sobrecarga y capacidad para manejar redes con ancho de banda limitado. Además, proporciona un modelo de comunicación de publicación/suscripción que permite la transmisión de datos en tiempo real entre los dispositivos IoT utilizados en la agricultura inteligente. Este protocolo ha sido utilizado exitosamente en aplicaciones de monitoreo de cultivos y riego, entre otros. Sus características de bajo consumo de energía y soporte para la comunicación bidireccional lo convierten en una opción adecuada para implementaciones en la agricultura de precisión.

AMQP (*Advanced Message Queuing Protocol*)

El protocolo avanzado de encolado de mensajes AMQP es un estándar abierto que desempeña un papel fundamental en el contexto IoT. AMQP ofrece una solución eficiente y confiable para la comunicación entre dispositivos conectados en entornos distribuidos. Este protocolo permite el intercambio de mensajes en tiempo real, estableciendo una infraestructura de mensajería robusta y escalable. Al utilizar AMQP, los dispositivos IoT pueden enviar y recibir datos de forma asincrónica, asegurando la entrega segura de mensajes en redes heterogéneas y garantizando la interoperabilidad entre diferentes plataformas y sistemas. La flexibilidad y la capacidad de gestión de colas de mensajes de AMQP posibilitan una integración fluida de múltiples dispositivos, permitiendo la implementación de soluciones inteligentes y eficientes en el ámbito del IoT [7].

1.5.5 Arquitecturas de referencia en IoT

Existen varias propuestas de arquitecturas para la implantación de soluciones IoT basadas en estructuras de múltiples capas como se muestra en la Figura 3.

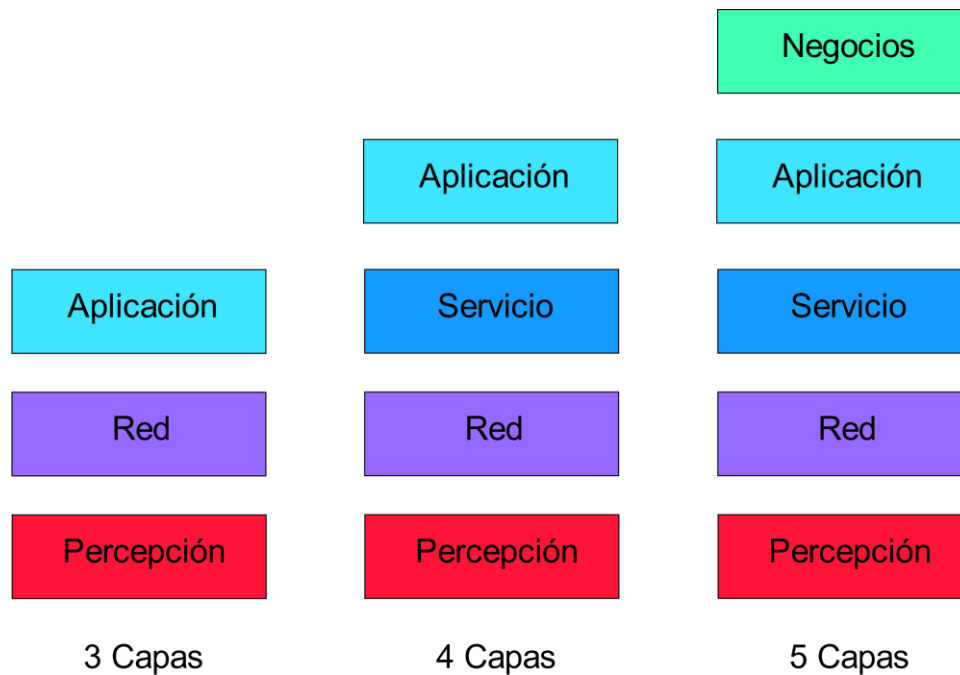


Figura 3. Arquitecturas de referencia IoT

En este apartado se describe cada una de las arquitecturas y, posteriormente, en el capítulo 2 se elegirá la arquitectura para la implementación del prototipo IoT de este proyecto.

Arquitectura de 3 capas

Capa de Percepción: Esta capa engloba todos los dispositivos IoT y tiene como objetivo principal identificar dichos dispositivos, recopilar datos ambientales a través de sensores y convertirlos en señales digitales [8].

Capa de Red: La función de esta capa es establecer la conexión con otros dispositivos físicos, de red y servidores. Entre sus características se encuentra la transmisión y procesamiento de los datos generados por los sensores. [8]

Capa de Aplicación: En esta capa se encuentran las aplicaciones y servicios del usuario, los cuales utilizan de manera efectiva la información procesada por los sensores. Además, permite el intercambio de datos con otras aplicaciones, servicios, sistemas y plataformas. [8]

Arquitectura de 4 capas

En esta arquitectura de 4 capas para dispositivos IoT, se añade una capa adicional a la arquitectura de 3 capas descrita anteriormente:

Capa de Servicio: También conocida como middleware, esta capa asume el almacenamiento de los datos provenientes de la capa de transporte. Además, se encarga del procesamiento y análisis de los datos para extraer información de valor, así como de tomar decisiones automáticas basadas en los resultados obtenidos. También administra y proporciona una variedad de servicios en las capas inferiores [8].

Arquitectura de 5 capas

En una arquitectura de 5 capas para dispositivos IoT, se incluye una capa adicional a las arquitecturas previamente mencionadas:

Capa de Negocios: En esta capa se gestiona el sistema IoT en su totalidad, proporcionando soporte para el proceso de toma de decisiones. Esto incluye aplicaciones, modelos de negocio y consideraciones relacionadas con la privacidad [8].

2. METODOLOGÍA

2.1 Investigación constructivista de Oyegoke

La investigación constructivista desempeña un papel fundamental en la definición y resolución de problemas, lo cual contribuye a mejorar el rendimiento de prototipos existentes o creados. El enfoque constructivista busca estructurar la experiencia individual de las cosas, estableciendo reglas generales para interpretar los fenómenos. Cuando un investigador aborda un problema desde esta perspectiva, lo hace con una concepción previa junto con una posible solución, reconociendo que la realidad es una construcción interpretada por el observador. [9]

El enfoque de investigación constructivista propuesto por Oyegoke consta de 6 fases, sin embargo, para este estudio se seleccionaron cuatro fases, fusionando la fase dos y la fase tres en una única fase a fin de cumplir con los objetivos del diseño del prototipo. A continuación, se describen detalladamente las tres fases seleccionadas, las cuales se presentan en la Figura 4.

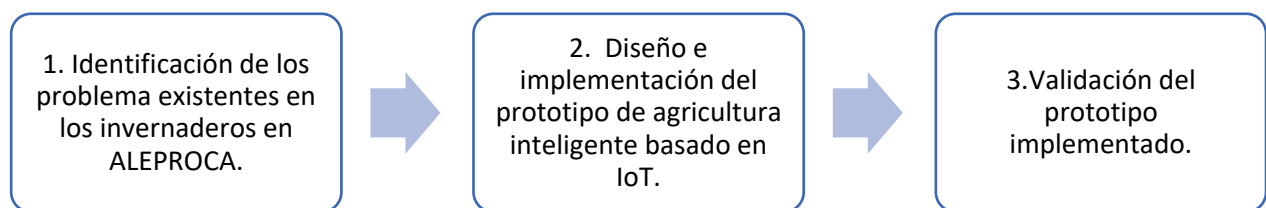


Figura 4. Fases del proyecto

- Primera fase, La primera fase consiste en la identificación de problemas y necesidades en los invernaderos de la empresa ALEPROCA. El problema para resolver en esta tesis fue analizado en la sección 1.1 del capítulo 1.
- Segunda fase, La siguiente etapa se centra en la selección de una arquitectura IoT para la implementación del prototipo. Se realizó un análisis de varias arquitecturas IoT con el objetivo de elegir la más apropiada para el desarrollo del prototipo. La selección de la arquitectura se basó en los requisitos y características específicas del proyecto, adicionalmente se realiza la selección del protocolo de comunicación que se utilizara en el prototipo.
- Tercera fase, En la tercera fase, se realiza la evaluación del funcionamiento del prototipo IoT implementado. Para esta evaluación, se utiliza la escala de

usabilidad SUS (SUS) "System Usability Scale" (SUS). Los resultados de esta evaluación se describen en el capítulo 3 de este documento de tesis.

2.2 Diseño e implementación del prototipo de agricultura inteligente basado en IoT.

En esta sección, se aborda el proceso de diseño e implementación del prototipo de agricultura inteligente basado en IoT para la empresa ALEPROCA. En primer lugar, se realizó la selección de la arquitectura para la implementación del prototipo.

2.3 Selección de la arquitectura IoT

Para el diseño del prototipo, se evaluaron tres arquitecturas IoT referenciadas previamente en el capítulo 1 sección 1.4.5, considerando diferentes parámetros, los cuales se detallan en la Tabla 1.

Parámetro	Descripción	ARQUITECTURAS		
		3 capas	4 capas	5 capas
Conectividad de dispositivos IoT	Es la capacidad de la arquitectura para permitirá la "conexión" de dispositivos IoT.	✓	✓	✓
Recopilación de datos del entorno	Es la capacidad de la arquitectura para permitir "recopilar" datos del entorno a través de los dispositivos IoT.	✓	✓	✓
Almacenamiento de datos	Es la capacidad de la arquitectura para "almacenar los datos recopilados" por los dispositivos IoT.	✓	✓	✓
Procesamiento de datos	Es la capacidad de la arquitectura para "procesar los datos recopilados" de los dispositivos IoT y "convertirlos" en información útil.	✓	✓	✓
Visualización o presentación	Es la capacidad de la arquitectura para "presentar los datos procesados" de una manera clara y comprensible para los usuarios finales.	✓	✓	✓
Facilidad de implementación	Es la "facilidad" con la que se puede implementar el prototipo tomando como referencia cada una de las capas de la arquitectura.	Bajo	Medio	Alto

Tabla 1. Análisis comparativo de las arquitecturas para la implementación de soluciones IoT

Después de un análisis basado en los parámetros presentados en la Tabla 1, se ha concluido que la “arquitectura de tres capas” es la idónea para el desarrollo de este proyecto. Esta arquitectura fue elegida debido a su capacidad para cumplir con cada uno de los parámetros evaluados, así como su simplicidad de diseño que al estar constituido por menos capas permite simplificar varios procesos lo que facilita su implementación y comprensión.

2.4 Selección del protocolo IoT

Después de analizar los protocolos IoT utilizados en la agricultura de precisión, se ha tomado la decisión de seleccionar MQTT como el protocolo para el desarrollo del prototipo, en lugar de CoAP y AMQP. Esta elección se basa en una serie de consideraciones y ventajas específicas que MQTT ofrece en comparación con los otros protocolos.

En primer lugar, MQTT es reconocido por su eficiencia en la transmisión de datos en tiempo real, lo cual resulta crucial en el ámbito agrícola para la monitorización y control en tiempo real de variables críticas, como la humedad del suelo y la temperatura. Además, MQTT proporciona un mecanismo ligero para el intercambio de información entre los dispositivos IoT utilizados en la agricultura inteligente. Esta capacidad de intercambio de datos en tiempo real y de forma eficiente es esencial para optimizar los procesos agrícolas y facilitar la toma de decisiones [10].

Además, MQTT ha demostrado ser altamente eficiente en el manejo de redes con ancho de banda limitado. En entornos agrícolas, donde los dispositivos IoT pueden estar ubicados en áreas remotas o con conectividad limitada, es crucial utilizar un protocolo que minimice el consumo de recursos de red y garantice una comunicación fluida y confiable. MQTT ofrece mecanismos de compresión de datos y una arquitectura de bajo consumo de energía, lo que lo convierte en una opción viable y apropiada para aplicaciones en agricultura de precisión [11].

En conclusión, la elección de MQTT sobre CoAP y AMQP se fundamenta en las ventajas que MQTT ofrece en cuanto a la eficiencia en la transmisión de datos en tiempo real, el manejo de redes con ancho de banda limitado y el soporte para la comunicación bidireccional. Estas características resultan fundamentales para el desarrollo de un prototipo de agricultura inteligente que se ajuste a los requisitos y

objetivos de ALEPROCA en términos de optimización de procesos agrícolas y toma de decisiones basadas en datos.

2.5 Arquitectura del prototipo inteligente basado en IoT

En base a la arquitectura seleccionada, se aborda el desarrollo del prototipo que en este estudio lo hemos denominado PAI-IoT (Prototipo de Agricultura Inteligente basado en IoT). Este prototipo permitirá la monitorización y ayudará a la gestión eficiente de los cultivos en los invernaderos de la empresa ALEPROCA.

La arquitectura del PAI-IoT se compone de tres capas funcionales, como se ilustra en la Figura 5.

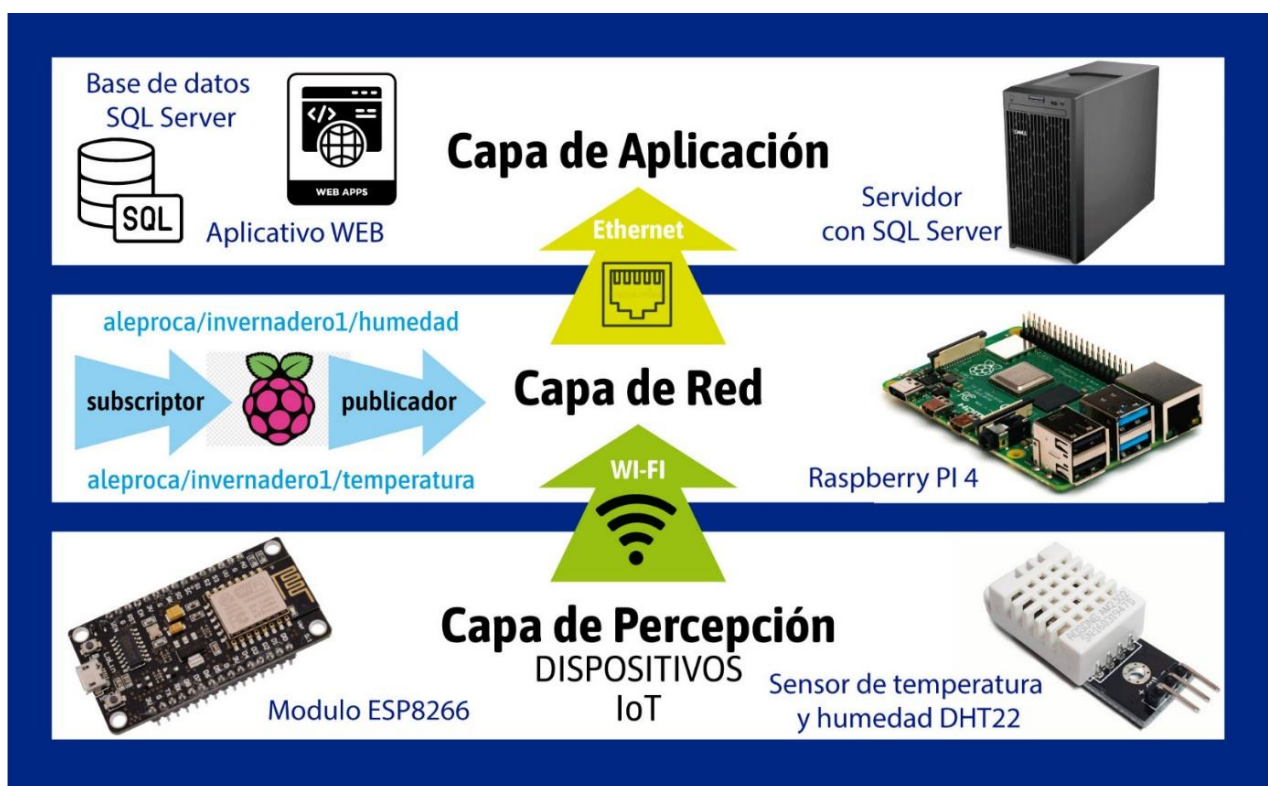


Figura 5. Arquitectura del PAI-IoT

2.5.1 Capa de percepción

La capa de percepción está representada por la implementación del nodo IoT, cuya función es monitorizar los datos de temperatura y humedad en los invernaderos utilizando un sensor ambiental. Esta capa es responsable de establecer la conexión entre los distintos dispositivos que forman parte del prototipo y enviar los datos a la capa de red para su posterior procesamiento.

2.5.2 Capa de red

La capa de red está representada por el bróker MQTT, la finalidad de esta capa es facilitar la conexión entre los distintos componentes del prototipo, asegurando así la conectividad entre el dispositivo, el bróker MQTT y la aplicación web.

2.5.3 Capa de aplicación

La capa de aplicación está representada por el aplicativo web, y desempeña un papel fundamental en la arquitectura de tres capas, esta capa se encarga del almacenamiento, procesamiento y visualización de los datos capturados por el nodo IoT.

A continuación, se explica la implementación del prototipo IoT en base a las tres capas antes definidas.

2.6 Implementación del prototipo IoT

La implementación del prototipo IoT consta de tres etapas, implementación del nodo IoT, configuración del bróker MQTT y el diseño y programación del aplicativo web.

2.6.1 Implementación del nodo IoT

La implementación del nodo IoT se realizó utilizando los siguientes componentes; el módulo ESP8266 WIFI, el sensor de temperatura y humedad DHT22, el módulo cargador TP4056, un panel solar de 6V, una pila recargable de 4000 mAh y un diodo 1N4007.

A continuación, se describe cada uno de los componentes utilizados.

Módulo ESP8266 WIFI

Este módulo es un dispositivo de conectividad inalámbrica, este posee la capacidad para conectarse a redes inalámbricas debido a su chip WiFi integrado y que es compatible con los estándares IEEE 802.11 b/g/n y se puede conectar a cualquier red doméstica o empresarial, además se caracteriza por su bajo costo, tamaño compacto y bajo consumo de energía. En la Figura 6 se muestra el módulo ESP8266 Wi-Fi utilizado.



Figura 6. Módulo ESP8266 Wi-Fi

Sensor de humedad y temperatura DHT22

El sensor DHT22 incorpora un sensor de humedad capacitivo y un sensor de temperatura y está basado un microcontrolador de 8 bits. Con un rango de detección de humedad relativa de 0 a 100 % y temperatura de -40 a 125 grados centígrados. En la figura 7 se muestra el sensor DHT22 utilizado.



Figura 7. Sensor de Temperatura y Humedad DHT22

Batería recargable

La batería recargable de 4000 mA es una fuente de energía confiable y duradera para el prototipo. Con su capacidad de almacenamiento, ofrece un suministro prolongado de energía, asegurando un funcionamiento continuo del dispositivo. En la Figura 8 se muestra la batería utilizada.



Figura 8. Pila recargable 4000 mA

Módulo cargador TP4056

El módulo cargador TP4056 es un dispositivo compacto utilizado para cargar baterías o pilas de litio. Es eficiente y seguro, con un puerto de entrada para la fuente de alimentación y un puerto de salida para la batería a cargar. Regula la corriente de carga y controla el proceso de carga de manera constante. En la Figura 9 se muestra el módulo cargador TP4056 utilizado.

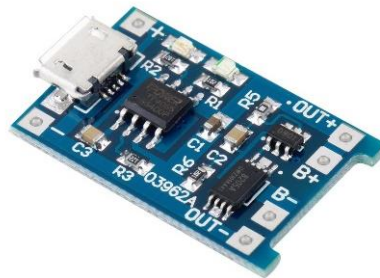


Figura 9. Módulo cargador TP4056

Panel solar de 6V 4000mAh

Es un dispositivo que convierte la luz solar en energía eléctrica. Tiene una capacidad de corriente máxima de 4000mAh y una tensión nominal de 6V. Este panel solar puede utilizarse para cargar baterías o alimentar dispositivos de baja potencia. En la Figura 10 se muestra el panel solar utilizado.



Figura 10. Panel solar 6V

Diodo 1N4007

El diodo 1N4007 es un diodo rectificador diseñado para permitir que la corriente fluya en una dirección y bloquee la corriente en la dirección opuesta. Tiene una capacidad de voltaje inverso de 1000V y una corriente máxima directa de 1A, lo que lo hace adecuado para aplicaciones de baja y mediana potencia. En la Figura 11 se muestra el diodo 1N4007 utilizado.



Figura 11. Diodo 1N4007

Configuración de la placa ESP8266

Para la construcción del nodo IoT se realizó la conexión física entre el módulo ESP8266 y el sensor DHT22, para ello, se conectó el pin de datos del sensor al pin D1 del módulo utilizando una resistencia de 4,7 KOhms para obtener los datos por medio de esta conexión. Además, se establecieron las conexiones entre el pin positivo del sensor y el pin 3V3 del módulo a fin de brindar energía desde el módulo ESP8266 al sensor DHT22. Adicionalmente se estableció la conexión entre el pin negativo del sensor y el pin GND del módulo. La Figura 12 muestra de manera ilustrativa estas conexiones.

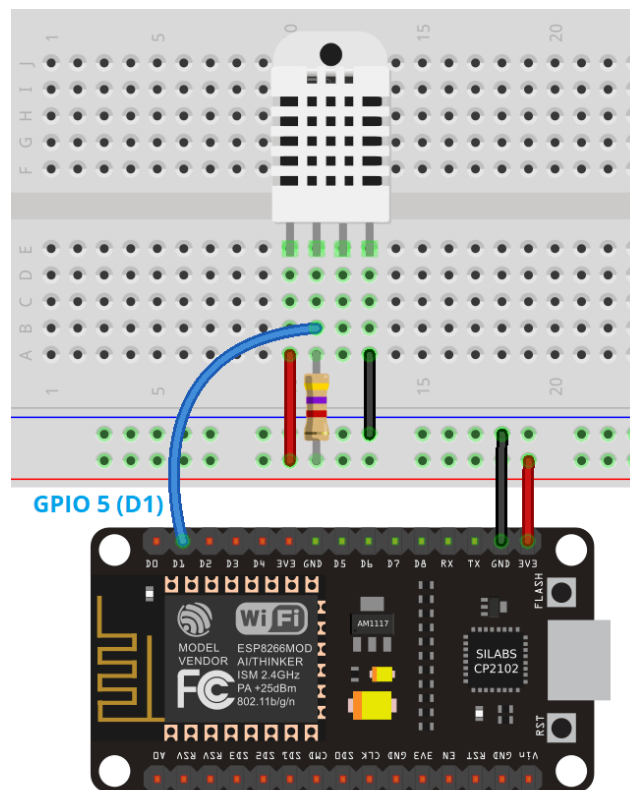


Figura 12. Conexión del módulo DHT22 a la placa ESP8266 WIFI

Debido a que el nodo IoT va a ser desplegado en un entorno exterior se utilizó energía solar, se describe a continuación los pasos utilizados para prototipar el nodo IoT para ambientes exteriores:

1. Se seleccionó un panel solar de 6V y 4000mAh como fuente principal de energía. Este panel solar captará la luz solar y la convertirá en energía eléctrica.
2. Se utilizó un módulo cargador TP4056 para transformar la energía obtenida por el panel solar en una corriente adecuada para cargar la batería.
3. Se colocó el panel solar en la parte superior externa de una caja hermética para exteriores. Esta ubicación permite que el panel solar reciba la máxima exposición a la luz solar para obtener la mayor cantidad de energía posible.
4. Se hizo un pequeño hueco en la caja hermética para permitir que los cables del panel solar se conecten al módulo cargador TP4056 en el interior de la caja. De esta manera, la energía generada por el panel solar se puede transferir al módulo cargador.
5. El módulo cargador TP4056 está conectado a una batería recargable de 4000mAh. El módulo cargador se encarga de regular la corriente de carga para evitar dañar la batería y asegurar una carga segura y eficiente.
6. La batería recargable está conectada al nodo IoT para suministrarle la energía necesaria para su funcionamiento. Esto permite que el nodo IoT se encienda y realice sus funciones utilizando la energía almacenada en la batería.
7. En la parte inferior de la caja hermética, se hizo otro pequeño hueco para permitir que un sensor pueda medir la temperatura y humedad del entorno. Este sensor se encuentra en el exterior de la caja y proporciona mediciones al nodo IoT para su posterior publicación o uso en el prototipo.

Este diseño permite proporcionar energía sostenible y mediciones ambientales al nodo IoT, además, permite que el dispositivo funcione de manera autónoma sin depender de una fuente de alimentación externa y recopile datos ambientales importantes.

Para establecer la conexión entre el módulo ESP8266 WIFI y el sensor DHT22 se utilizó el entorno de desarrollo integrado (IDE) de Arduino donde se realizaron los siguientes pasos su funcionamiento:

- **Instanciación de las bibliotecas necesarias:**

DHT.h: Permite la interacción con el sensor DHT22 para la medición de la temperatura y la humedad.

PubSubClient.h: Facilita la comunicación MQTT, que es crucial para el intercambio de datos con otros dispositivos o plataformas en la red.

ESP8266WiFi.h: Proporciona las herramientas necesarias para configurar y gestionar la conexión Wi-Fi, permitiendo al dispositivo comunicarse con la red y, por extensión, con el servidor MQTT.

Adafruit_Sensor.h: Es una biblioteca necesaria para el funcionamiento de la biblioteca DHT, ya que proporciona una interfaz común para diversos sensores.

En la figura 13, podemos observar la instanciación de las bibliotecas necesarias para el funcionamiento del dispositivo.

```
#include <DHT.h>
#include <PubSubClient.h> // Librería para el Broker MQTT
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
```

Figura 13. Instanciar Librerías

- **Definición de pines y variables:**

`FORCE_DEEPSLEEP`: Directiva de compilación condicional para habilitar el modo de suspensión profunda del ESP8266.

`DHTPIN`: El pin al que está conectado el sensor DHT22 en el ESP8266.

`DHTTYPE`: El tipo de sensor DHT utilizado (DHT22 en este caso).

Creación de una instancia de la clase `DHT` con el pin y tipo definidos anteriormente.

En la Figura 14, podemos observar la definición de pines y variables.

```
#define FORCE_DEEPSLEEP // Variable para dormir al ESP
#define DHTPIN 5 // Pin conectado al ESP8266
#define DHTTYPE DHT22 // DHT22

DHT dht(DHTPIN, DHTTYPE);
```

Figura 14. Definición de Variables

• **Configuración de la red Wi-Fi:**

Se define el nombre de la red Wi-Fi (`Invernadero`) y la contraseña (`invernadero321`).

Se configura una dirección IP estática para el ESP8266 utilizando las variables `local_IP`, `gateway`, `subnet`, `primaryDNS` y `secondaryDNS`.

En la Figura 15 podemos observar la configuración de la red WI-Fi.

```
// WiFi
const char* ssid = "Invernadero"; // Nombre de la red Wifi
const char* wifi_password = "invernadero321"; // Contraseña de la red Wifi

// Seteo IP fija ESP8266
IPAddress local_IP(192, 168, 40, 18);
IPAddress gateway(192, 168, 40, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(8, 8, 8, 8);
IPAddress secondaryDNS(8, 8, 4, 4);
```

Figura 15. Conexión Wi-Fi y configuración IP estática

• **Configuración del cliente MQTT:**

- Se definen las variables para el ID del cliente MQTT (`clientID`), la dirección IP del broker MQTT (`mqtt_server`) y los temas MQTT para publicar los datos de humedad (`humedad_topic`), temperatura (`temperatura_topic`) e invernadero (`invernadero_topic`).
- Se definen las credenciales de inicio de sesión para el broker MQTT (`mqtt_usuario` y `mqtt_password`).
- Se creó una instancia de la clase `WiFiClient` para la conexión Wi-Fi.

- Se crea una instancia de la clase `PubSubClient` con la dirección IP del broker MQTT y el puerto 1883.

En la Figura 16 se muestra la configuración del cliente MQTT.

```
// MQTT
const char* clientID = "cliente"; // MQTT cliente ID
const char* mqtt_server = "192.168.40.29"; // IP del Broker (Raspberry Pi
192.168.40.18)
const char* humedad_topic = "aleproca/invernadero1/humedad";
const char* temperatura_topic = "aleproca/invernadero1/temperatura";
const char* invernadero_topic = "aleproca/invernadero1/invernadero";
const char* mqtt_usuario = "idavila"; // MQTT Usuario
const char* mqtt_password = "idavila"; // MQTT Contraseña

// Iniciar conexión Wifi
WiFiClient wifiCliente;
// Puerto del broker 1883
PubSubClient client(mqtt_server, 1883, wifiCliente);
```

Figura 16. Configuración del cliente MQTT

Se configura la conexión Wi-Fi utilizando las credenciales proporcionadas y se espera hasta que se establezca la conexión.

En la Figura 17 se muestra la configuración función conectar_MQTT().

```

// Función personalizada para conectarse al broker MQTT a través de Wi-Fi
void conectar_MQTT(){

    if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
        Serial.println("Fallo la configuración");
    }
    Serial.print("Conectando a ");
    Serial.println(ssid);

    // Conectarse al Wi-Fi
    WiFi.begin(ssid, wifi_password);

    // Esperar hasta que se confirme la conexión antes de continuar
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // Muestra la dirección Ip del ESP8266
    Serial.println("WiFi Conectado");
    Serial.print("Dirección IP: ");
    Serial.println(WiFi.localIP());

    // Conectarse a MQTT Broker
    // client.connect devuelve un valor booleano para informarnos si la conexión
    fue exitosa.
    // Si la conexión falla, revisar el nombre de usuario y la contraseña de
    MQTT
    if (client.connect(clientID, mqtt_usuario, mqtt_password)) {
        Serial.println("¡Conectado a MQTT Broker!");
    }
    else {
        Serial.println("Error de conexión con MQTT Broker ...");
    }
}

```

Figura 17. Función conectar_MQTT ()

- **Inicialización de la función `setup ()` :**

Inicializa la comunicación serial a una velocidad de 9600 baudios e inicializa el sensor DHT como se indica en la Figura 18.


```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
}
```

Figura 18. Función setup ()

- **Inicialización de la función `loop ()`:**
 - Se establece la función `conectar_MQTT ()` para establecer la conexión con el broker MQTT.
 - Se instancia la humedad y temperatura del sensor DHT22 utilizando la librería 'Dht.H'.
 - Se convierte las variables de humedad y temperatura a cadenas de caracteres para enviarlas a través de MQTT.
 - Se publica los valores de humedad y temperatura en los respectivos temas MQTT utilizando la función `client. publish ()`.
 - Se maneja los casos en los que no se pueda enviar la información, intentando reconectarse al broker MQTT y volver a publicar los datos.
 - Se desconecta del broker MQTT utilizando la función `client. disconnect ()`.
 - Si está habilitada la directiva `FORCE_DEEPSLEEP`, el módulo entra en modo de suspensión profunda durante 20 minutos utilizando `ESP.deepSleep()`.

En la Figura 19 se muestra cómo se configuró la función loop().

```
void loop() {
  conectar_MQTT();
  Serial.setTimeout(2000);

  float h = dht.leerHumedad();
  float t = dht.leerTemperatura();
  String i = WiFi.softAPmacAddress();

  Serial.print("Humedad Relativa: ");
  Serial.print(h);
  Serial.println(" %");
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.println(" °C");
  Serial.println(i);
}
```

Figura 19. Función `loop ()`

Al iniciar la función personalizada 'conectar_MQTT' y guardar los valores de humedad y temperatura en variables de tipo float, estas variables float se convierten a String, ya que el protocolo MQTT solo puede transmitir cadenas como se muestra en la Figura 20.

```
String hs="Hum: "+String((float)h)+" % ";
String ts="Temp: "+String((float)t)+" C ";
```

Figura 20. Conversión de tipo float a tipo String

Para la publicación de los tópicos, el Broker MQTT realiza la transmisión de temperatura y humedad mediante varias configuraciones como se indica en la Figura 21.

```

// Publicación MQTT Broker (topic = Temperatura)
if (client.publish(temperatura_topic, String(t).c_str())) {
    Serial.println("Temperatura enviada!");
}
else {
    Serial.println("No se pudo enviar la temperatura. Reconnectarse a MQTT
Broker y vuelva a intentarlo");
    client.connect(clientID, mqtt_usuario, mqtt_password);
    delay(20); // Este retraso asegura que client.publish no entre en conflicto
con la llamada client.connect
    client.publish(temperatura_topic, String(t).c_str());
}
// Publicación MQTT Broker (topic = Humedad)
if (client.publish(humedad_topic, String(h).c_str())) {
    Serial.println("Humedad enviada!");
}
else {
    Serial.println("No se pudo enviar la humedad. Reconnectarse a MQTT Broker
y vuelva a intentarlo");
    client.connect(clientID, mqtt_usuario, mqtt_password);
    delay(20); // Este retraso asegura que client.publish no entre en conflicto
con la llamada client.connect
    client.publish(humedad_topic, String(h).c_str());
}

```

Figura 21. Publicación de tópicos

Finalmente, a fin de que el dispositivo ahorre lo máximo de energía, desconectamos al cliente y forzamos al dispositivo a entrar en reposo para generar autonomía en el mismo como se indica en la Figura 22.

```
client.disconnect(); // desconexión del MQTT

delay(1000*10);      // Impresión de valores cada 2 minuto

// Forzar para que entre en reposo

#ifdef FORCE_DEEPSLEEP
Serial.println("Duerme por 20 min!"); // Imprime el tiempo que va a dormir
ESP.deepSleep(20 * 60 * 1000000); // Configuración el tiempo que duerme (20
Min)
delay(100);
#endif
```

Figura 22. Desconexión y forzar reposo

2.6.2 Configuración del Broker MQTT

Se configuró el bróker en una Raspberry Pi 4, utilizando una memoria SD de 32 GB para almacenar el sistema operativo. Después de instalar el sistema operativo, se asignó una dirección IP estática para evitar conflictos de red.

Configuraciones iniciales

El bróker es un servidor que recopila los mensajes enviados por el módulo ESP8266 WiFi, los guarda y los distribuye a los suscriptores previamente inscritos a los tópicos utilizando el broker MQTT se instaló utilizando el siguiente comando por consola: "sudo apt-get install mosquitto".

Para controlar los servicios MQTT, se editó el archivo "/etc/mosquitto/mosquitto.conf" como se muestra en la Figura 23. A fin de realizar modificaciones a cuatro líneas como se indica en la Figura 24.

Con estas configuraciones previas, se parametrizo el broker MQTT para:

- Evitar que el bróker adopte la configuración predeterminada.
- Impedir que se realicen conexiones de usuarios anónimos al bróker MQTT.

- Permitir almacenar las contraseñas en otro archivo y leerlas desde allí.
- Acceder al bróker MQTT mediante el puerto 1883.

```

pi@raspberrypi: ~
libmosquittol
The following NEW packages will be installed:
  libmosquittol mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 65 not upgraded.
Need to get 124 kB of archives.
After this operation, 255 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ftp.halifax.rwth-aachen.de/raspbian/raspbian buster/main armhf libm
osquittol armhf 1.5.7-1+deb10u1 [57.6 kB]
Get:2 http://ftp.agdsn.de/pub/mirrors/raspbian/raspbian buster/main armhf mosqui
tto-clients armhf 1.5.7-1+deb10u1 [66.8 kB]
Fetched 124 kB in 2s (65.0 kB/s)
Selecting previously unselected package libmosquittol:armhf.
(Reading database ... 93898 files and directories currently installed.)
Preparing to unpack .../libmosquittol_1.5.7-1+deb10u1_armhf.deb ...
Unpacking libmosquittol:armhf (1.5.7-1+deb10u1) ...
Selecting previously unselected package mosquitto-clients.
Preparing to unpack .../mosquitto-clients_1.5.7-1+deb10u1_armhf.deb ...
Unpacking mosquitto-clients (1.5.7-1+deb10u1) ...
Setting up libmosquittol:armhf (1.5.7-1+deb10u1) ...
Setting up mosquitto-clients (1.5.7-1+deb10u1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rpil) ...
pi@raspberrypi:~ $ sudo nano /etc/mosquitto/mosquitto.conf

```

Figura 23. Comando para la configuración del servicio 'Mosquitto'

```

pi@raspberrypi: ~
GNU nano 3.2 /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /var/run/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

[ Read 13 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line

```

Figura 24. Configuración del archivo mosquitto.conf

Creación de usuario y contraseña en el broker

Para la creación de un usuario y contraseña para Mosquitto MQTT, se utilizó el comando "sudo mosquitto_passwd -c /etc/mosquitto/pwfile jdavila". Se generó el usuario "jdavila" y se ingresó la contraseña correspondiente.

Finalmente, se validó el estado del bróker MQTT mediante el comando "sudo systemctl status mosquitto" como se indica en la Figura 25.

```
pi@raspberrypi:~$ sudo systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2019-11-23 05:36:44 PST; 10s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
  Main PID: 1234 (mosquitto)
    Tasks: 1 (limit: 2200)
   Memory: 672.0K
   CGroup: /system.slice/mosquitto.service
           └─1234 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Nov 23 05:36:44 raspberrypi systemd[1]: Starting Mosquitto MQTT v3.1/v3.1.1 Broker...
Nov 23 05:36:44 raspberrypi systemd[1]: Started Mosquitto MQTT v3.1/v3.1.1 Broker.
```

Figura 25. Estado del servicio mosquitto

Configuración del suscriptor para recibir datos de MQTT Broker

La siguiente sección aborda la configuración necesaria para que el suscriptor pueda recibir los datos provenientes del MQTT Broker. Hasta este momento, se ha completado la configuración del publicador y del BROKER. Ahora, el último paso consiste en configurar el suscriptor, que en este caso es la misma Raspberry Pi 4.

Para lograr esto, utilizaremos el lenguaje de programación Python y crearemos un script que permita al Raspberry Pi escuchar los temas de MQTT enviados por el publicador. Para ello, es necesario instalar la biblioteca de "paho-mqtt" como se muestra en la Figura 26.

```
pi@raspberrypi:~$ sudo pip install paho-mqtt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting paho-mqtt
  Downloading https://files.pythonhosted.org/packages/59/11/1dd5c70f0f27a88a3a05772cd95f6087ac479fac66d9c7752ee5e16ddbcb/paho-mqtt-1.5.0.tar.gz (99kB)
    100% |#####| 102kB 797kB/s
Building wheels for collected packages: paho-mqtt
  Running setup.py bdist_wheel for paho-mqtt ... done
  Stored in directory: /root/.cache/pip/wheels/02/94/6c/8474137cb7a5a3e001d70a22c8ff919caee69435376bccce79
Successfully built paho-mqtt
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-1.5.0
pi@raspberrypi:~$
```

Figura 26. Instalación MQTT en el bróker

Con esta instalación, estaremos listos para proseguir con la programación del script de suscripción en Python como se indica en la Figura 27.

```

import paho.mqtt.client as mqtt
import pyodbc
MQTT_ADDRESS = '192.168.40.29'
MQTT_USER = 'idavila'
MQTT_PASSWORD = 'idavila'
MQTT_TOPIC = 'alenroca/+/+'
MQTT_TOPIC_Humedad = 'alenroca/invernadero1/humedad'
MQTT_TOPIC_Temperatura = 'alenroca/invernadero1/temperatura'
def on_connect(client, userdata, flags, rc):
    client.subscribe(MQTT_TOPIC_Temperatura)
def on_message(client, userdata, msg):
    print(msg.topic + ' ' + str(msg.payload))
    conn = pyodbc.connect('Driver={SQL Server};'
        'Server=186.46.255.217,14330;'
        'Database=Sensores;'
        'UID=sa;'
        'PWD=asinfo2680$;'
        'Trusted_Connection=no;'
        )
    cursor = conn.cursor()
    """ cursor.execute("SELECT * FROM datos")
    for row in cursor:
        print(row)"""
    cursor.execute("INSERT INTO temperatura (temperatura,fecha) VALUES (" + str(msg.payload).re-
place("'", "") + ", " + str(msg.payload).replace("'", "") + ");GETDATE()")
    cursor.commit()
    conn.close()
def main():
    mqtt_client = mqtt.Client()
    mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
    mqtt_client.on_connect = on_connect
    mqtt_client.on_message = on_message
    mqtt_client.connect(MQTT_ADDRESS, 1883)
    mqtt_client.loop_forever()
if __name__ == '__main__':
    main()

```

Figura 27. Script Suscriptor en Python

En el script, se realizó la importación de la biblioteca MQTT previamente instalada. Una vez importada la biblioteca, se procedió a definir todas las variables necesarias. Se verificó que la dirección IP del agente MQTT sea la IP del Raspberry Pi y que el nombre de usuario y la contraseña de MQTT estuvieran correctas.

La función "on_connect" se encarga de manejar las acciones realizadas cuando el cliente MQTT se conecta al broker. En este caso, cuando los clientes se conectan al broker, se suscriben a los temas de temperatura y humedad, y se muestra un mensaje indicando que la conexión se estableció. La variable "rc" contiene un código de éxito o error según sea el caso de que la conexión no sea exitosa como se indica en la Figura 29, lo que facilita el proceso de depuración:

- 0: Conexión exitosa.
- 1: Conexión rechazada - versión de protocolo incorrecta.
- 2: Conexión rechazada - identificador de cliente no válido.
- 3: Conexión rechazada - servidor no disponible.
- 4: Conexión rechazada - nombre de usuario o contraseña incorrectos.
- 5: Conexión rechazada - no autorizada.
- 6-255: Actualmente sin uso.

Figura 28. Acciones función on_connect

La función "on_message" se ejecuta cada vez que se publica un mensaje en el tema. En este caso, se imprimen tanto el tema como el mensaje en la terminal.

La función principal del script contiene la parte central del código en Python, donde se ejecutan todas las funciones previamente definidas. En primer lugar, se creó un objeto cliente y se estableció el nombre de usuario y la contraseña para el cliente MQTT. Luego, se indicó al cliente qué función debe invocarse al conectar al intermediario MQTT y al recibir mensajes.

Una vez que todo se configuró, se procedió a establecer la conexión con el intermediario utilizando la dirección IP y el puerto correspondiente. Después de establecer la conexión, el objeto cliente entra en un bucle que se ejecuta de forma continua.

A continuación, se ejecuta el código Python en el suscriptor con el comando que se detalla en la Figura 30:

```
@echo off
cd C:\Users\ALEPROCA\Desktop
python datos_mqtt_temperatura.py
```

Figura 29. Generación del bash para la obtención de datos

De esta manera, se observó los datos en la terminal y a su vez se realiza el registro en la base de datos SQL server como indica la Figura 31:

```
("INSERT INTO temperatura (temperatura, fecha) VALUES (" + str(msg.payload).replace("b", "").replace("'", "") + "';GETDATE()")
```

```
("INSERT INTO humedad (humedad, fecha) VALUES (" + str(msg.payload).replace("b", "").replace("'", "") + "';GETDATE()")
```

Figura 30. Inserción en la base de datos los valores de temperatura y humedad

2.7 Desarrollo del aplicativo WEB

En el desarrollo del PAI-IoT para la empresa ALEPROCA, se llevó a cabo el diseño y programación del aplicativo WEB que permite a los usuarios acceder y visualizar los datos recolectados, históricos y en tiempo real recopilados por los dispositivos IoT. Este aplicativo web se conforma de dos elementos principales, la base de datos y el propio aplicativo.

2.7.1 Base de datos

Para el presente estudio, se utilizó SQL Server como sistema de gestión de bases de datos relacionales ya que es el motor de base de datos que se utiliza en la compañía.

2.7.2 Aplicativo

El aplicativo web desarrollado forma parte de la capa de aplicación del prototipo y ofrece a los usuarios autorizados el analizar los datos recolectados.

El aplicativo web se compone de diversas secciones que presentan los datos de manera clara y organizada, brindando una interfaz intuitiva para la visualización de la información recolectada. Este aplicativo se conforma por las siguientes interfaces:

2.7.3 Autenticación y Registro

La pantalla de autenticación del aplicativo web constituye una etapa crucial para salvaguardar la privacidad y seguridad de los datos recopilados. El usuario debe ingresar sus credenciales en dicha pantalla para acceder al aplicativo. El sistema de autenticación se encarga de verificar la validez de las credenciales antes de permitir el acceso al aplicativo como se indica en la Figura 32.



Figura 31. Pantalla de Inicio de Sesión



Figura 32. Pantalla de Registro

2.7.4 Monitorización de datos

El aplicativo web cuenta con diversas pantallas que permiten al usuario visualizar de manera clara y comprensible los datos recopilados por los dispositivos IoT. Estas pantallas incluyen el control de temperatura, control de humedad y gráficos. Cada pantalla ofrece una interfaz amigable que facilita la exploración y comprensión de los datos como se muestra en la Figura 34, Figura 35, Figura 36, Figura 37 y Figura 38.

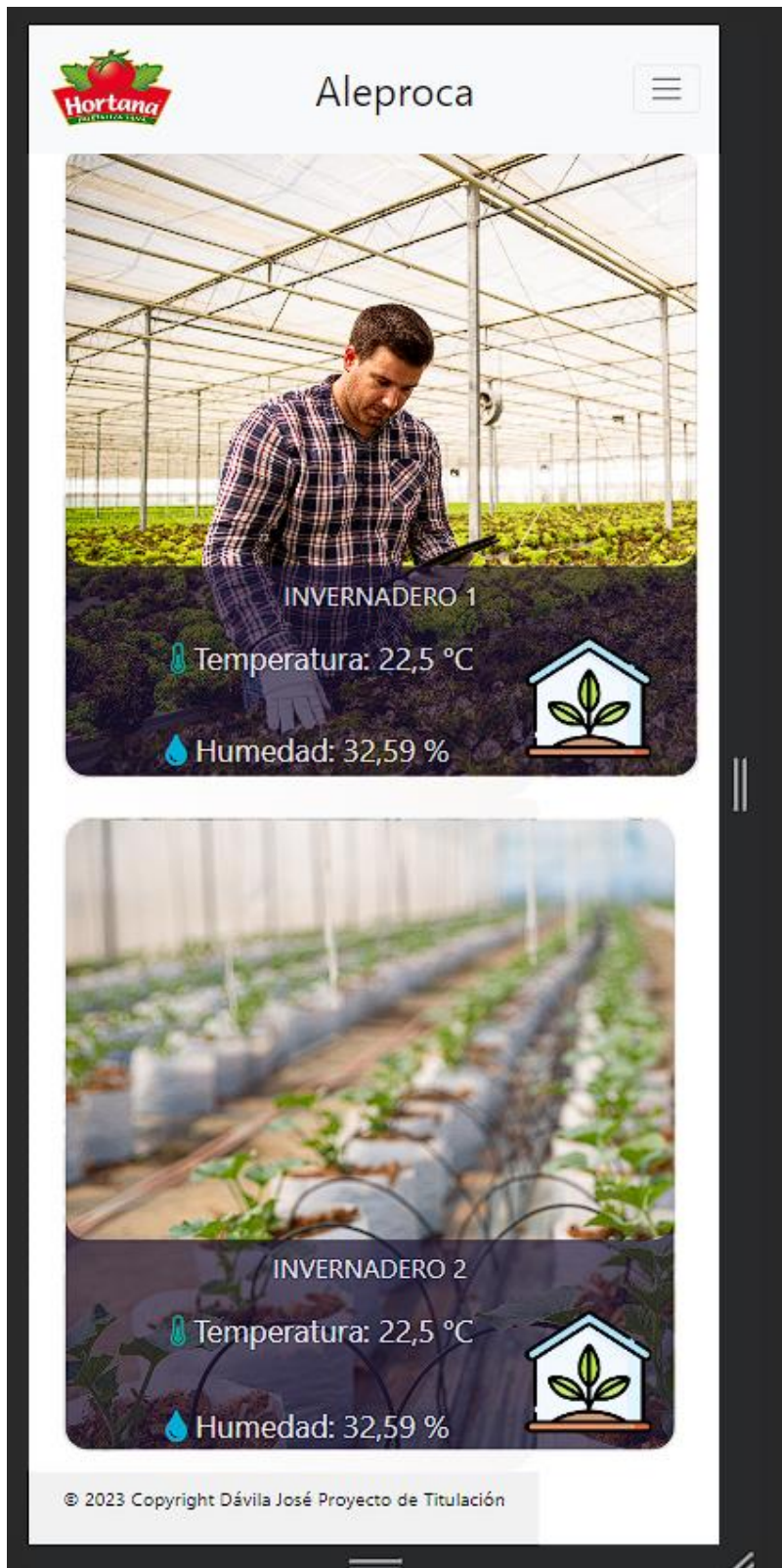


Figura 33. Selección del invernadero



septiembre de 2023

l	m	m	j	v	s	d
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

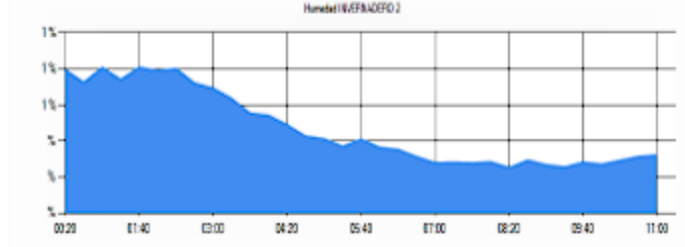
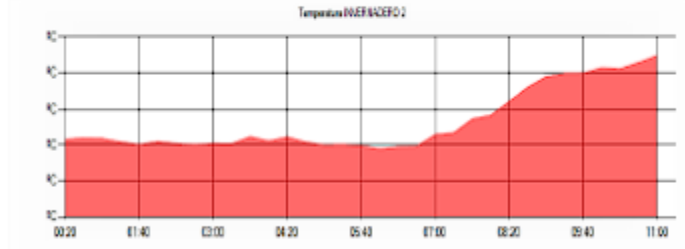
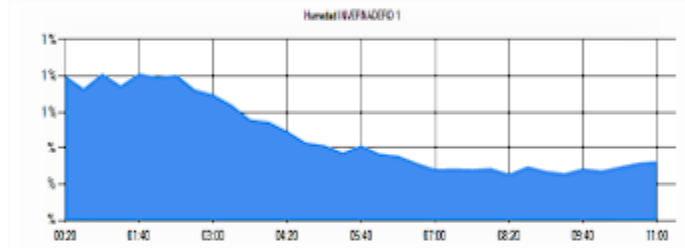
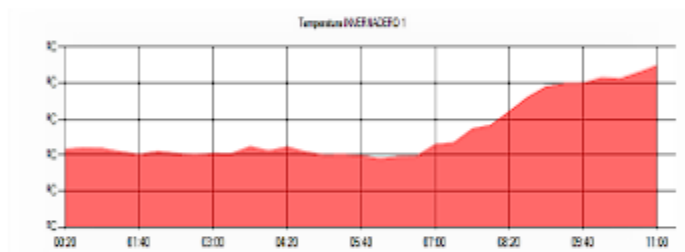


Figura 34. Panel de gráficos de los invernaderos



septiembre de 2023						
l	m	m	j	v	s	d
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Temperatura Inv 1	Hora	Día	Fecha
22,5	11:00	Jueves	09/28/23
21,53	10:40	Jueves	09/28/23
20,65	10:20	Jueves	09/28/23
20,79	10:00	Jueves	09/28/23
20,02	09:40	Jueves	09/28/23
19,97	09:20	Jueves	09/28/23
19,56	09:00	Jueves	09/28/23
18,09	08:40	Jueves	09/28/23
16,11	08:20	Jueves	09/28/23
14,18	08:00	Jueves	09/28/23
1234			

Humedad Inv 1	Hora	Día	Fecha
32,59	11:00	Jueves	09/28/23
31,81	10:40	Jueves	09/28/23
29,67	10:20	Jueves	09/28/23
27,49	10:00	Jueves	09/28/23
28,62	09:40	Jueves	09/28/23
25,97	09:20	Jueves	09/28/23
27,18	09:00	Jueves	09/28/23
29,8	08:40	Jueves	09/28/23
25,6	08:20	Jueves	09/28/23
28,8	08:00	Jueves	09/28/23
1234			

Temperatura Inv 2	Hora	Día	Fecha
22,5	11:00	Jueves	09/28/23
21,53	10:40	Jueves	09/28/23
20,65	10:20	Jueves	09/28/23
20,79	10:00	Jueves	09/28/23
20,02	09:40	Jueves	09/28/23
19,97	09:20	Jueves	09/28/23
19,56	09:00	Jueves	09/28/23
18,09	08:40	Jueves	09/28/23
16,11	08:20	Jueves	09/28/23
14,18	08:00	Jueves	09/28/23

Figura 35. Panel de datos históricos de los invernaderos

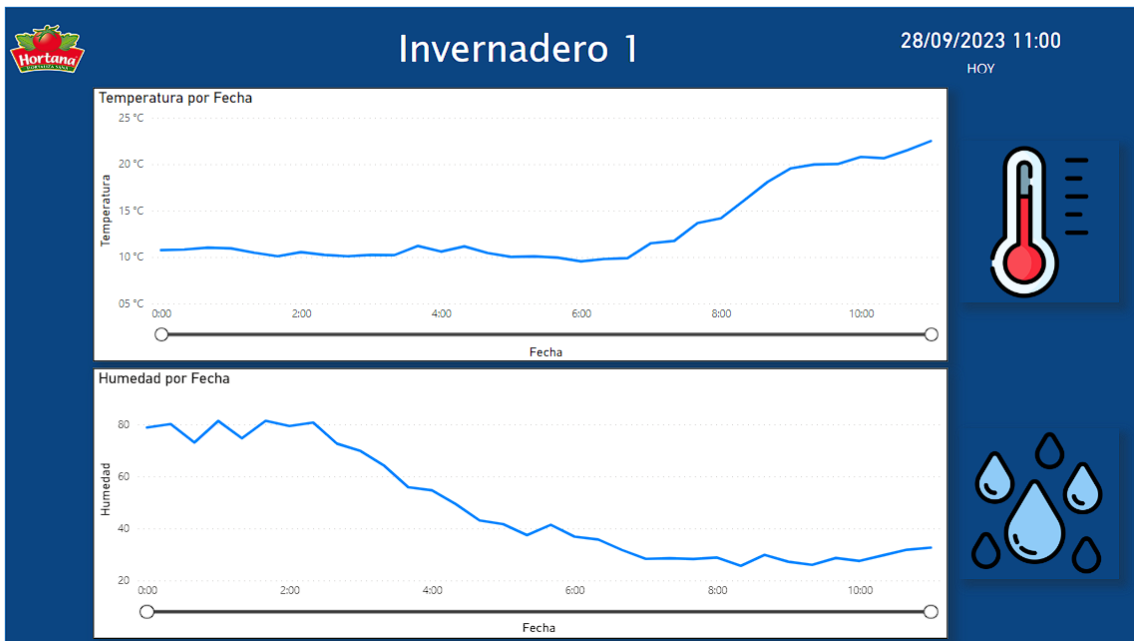


Figura 36. Panel de monitoreo de temperatura y humedad

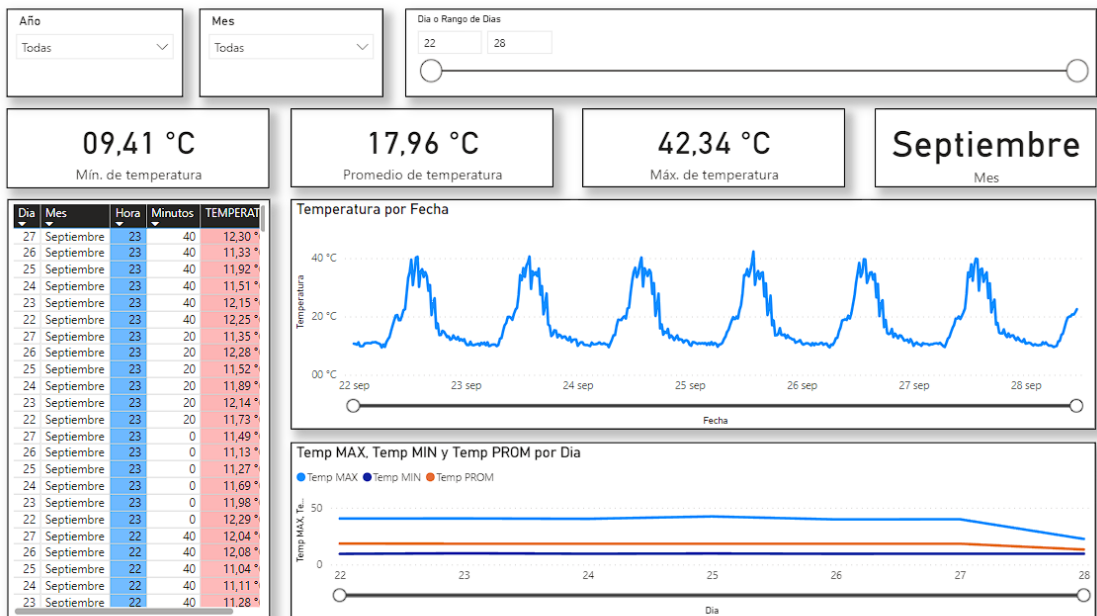


Figura 37. Panel de monitoreo de temperatura



Figura 38. Panel de monitoreo de humedad

3. DESPLIEGUE Y VALIDACIÓN DEL PROTOTIPO

3.1 Despliegue

El prototipo fue desplegado en un entorno real, específicamente en dos invernaderos distintos pertenecientes a la empresa ALEPROCA. Actualmente, los prototipos continúan operativos, recopilando datos cada 20 minutos, lo que resulta en un total de 72 mediciones diarias.

El diseño del prototipo se distingue por la incorporación de un panel solar en su parte superior. Este panel garantiza una alimentación basada en energía solar, permitiendo que el dispositivo opere de manera autónoma, sin la necesidad de fuentes de alimentación externas. En la parte inferior, se encuentra el sensor DHT22, diseñado para registrar datos ambientales relacionados con la temperatura y la humedad.

En relación con el aspecto económico, es relevante mencionar que el costo de producción de este prototipo es de aproximadamente \$30. Este monto refleja una inversión consciente en términos de innovación tecnológica, poniendo en manifiesto la relación costo-beneficio al considerar la eficiencia y precisión del dispositivo en la recopilación de datos ambientales, elementos clave para la optimización de la producción en invernaderos.

La decisión de situar el prototipo en el centro del invernadero se tomó en colaboración con el jefe agrícola, con el objetivo de evitar mediciones sesgadas en áreas determinadas, considerando que los invernaderos están equipados con cortinas. Esta posición central y elevada garantiza una exposición directa a la luz solar, optimizando la carga del panel solar y evitando interferencias con los cultivos. Además, esta localización estratégica posibilita una captura más eficaz de la temperatura y humedad ambientales, beneficiándose de la natural circulación y convección del aire caliente ascendente.

Las Figuras 39 y 40 muestran el prototipo durante sus etapas inicial y final de ensamblaje, respectivamente. Mientras que las Figuras 41 y 42 presentan el prototipo ya desplegado en el invernadero, evidenciando la practicidad y viabilidad de su implementación en un contexto real.

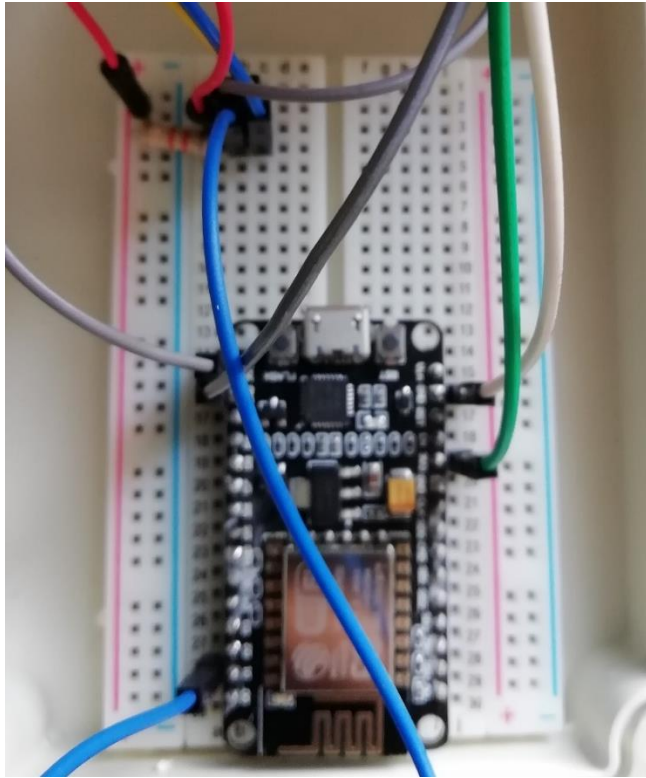


Figura 39. Ensamble inicial del prototipo

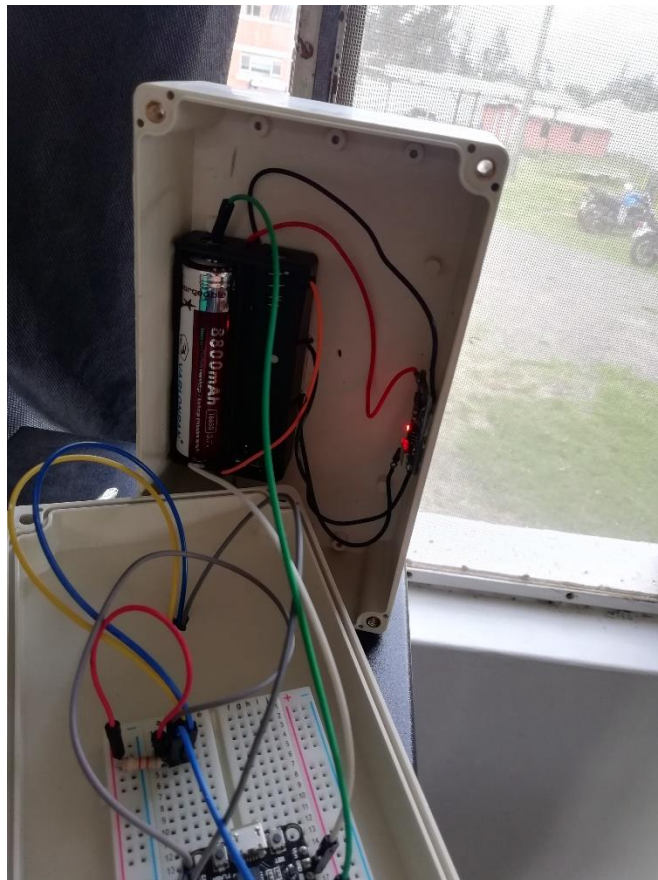


Figura 40. Ensamble final del prototipo

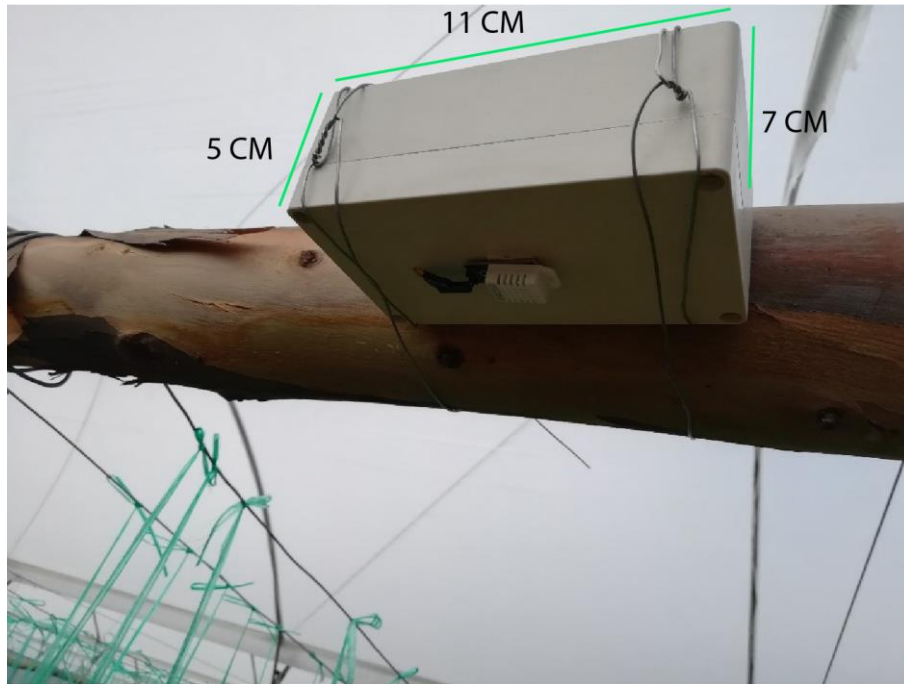


Figura 41. Instalación en el invernadero versión inferior del prototipo



Figura 42. Instalación en el invernadero versión superior del prototipo

3.2 Validación

El conjunto de datos recopilados y el proceso de validación ofrecen una base para la evaluación tanto del desempeño del nodo IoT en un entorno real como de su integración con el aplicativo web desarrollado. Para esta evaluación, se consideraron las mediciones de temperatura y humedad. Adicionalmente, con el objetivo de valorar la utilidad y facilidad de uso del aplicativo web, se desarrolló una encuesta basada en la escala "SUS". Esta encuesta fue dirigida al jefe agrícola y a nueve trabajadores de campo responsables del riego y mantenimiento de los invernaderos. Las preguntas de la encuesta se presentan a continuación con una escala Likert de 5 puntos que son:

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Pregunta	Totalmente de acuerdo	De acuerdo	Neutral	Desacuerdo	Totalmente desacuerdo
¿Considera que el prototipo es útil en la monitorización de parámetros ambientales?	8	1	0	1	0
¿La aplicación web tiene una interfaz gráfica que facilita la monitorización de parámetros ambientales?	9	1	0	0	0
¿El dispositivo proporciona datos precisos?	8	2	0	0	0
¿El dispositivo contribuye a la optimización del tiempo laboral?	8	1	1	0	0
¿El dispositivo contribuye al ahorro de agua en los cultivos?	7	3	0	0	0
¿El dispositivo tiene un impacto positivo en la mejora de la producción agrícola?	9	1	0	0	0
¿El uso del aplicativo web facilita el trabajo realizado en el invernadero?	10	0	0	0	0
¿Recomendaría el uso continuo del prototipo y su expansión a los otros invernaderos?	10	0	0	0	0

Tabla 2 Resultados Preguntas

Pregunta 1

¿Considera que el prototipo es útil en la monitorización de parámetros ambientales?

El 90% de los encuestados estuvo de acuerdo o totalmente de acuerdo con la utilidad del prototipo. Este alto porcentaje indica que el prototipo ha trabajado en una necesidad palpable dentro del invernadero. Antes de su implementación, la monitorización manual podría haber sido imprecisa y consumido una gran cantidad de tiempo. Con la solución presentada, se reduce la subjetividad y se incrementa la eficiencia. Es vital, considerar el 10% que no concordó plenamente con esta perspectiva, en particular la persona que no concuerda es de la tercera edad y no está familiarizado con esta tecnología.

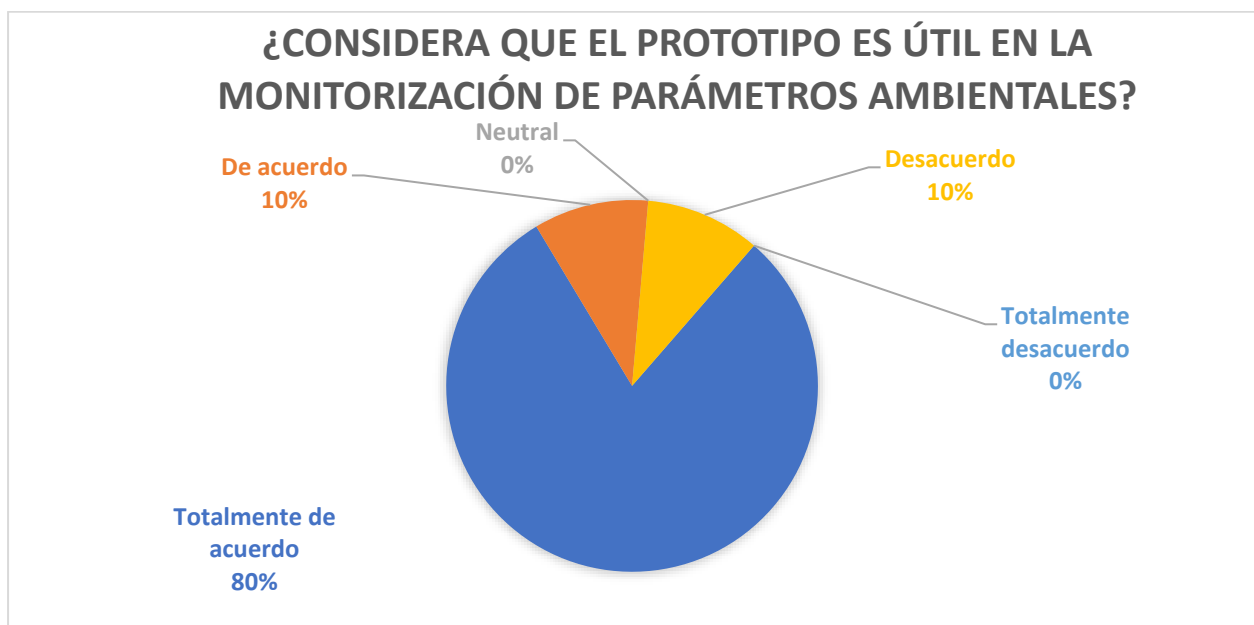


Figura 43. Resultados pregunta 1

Pregunta 2

¿La aplicación web tiene una interfaz gráfica que facilita la monitorización de parámetros ambientales?

El 90% de las respuestas afirmativas indica que la mayoría de los participantes considera que la interfaz gráfica de la aplicación web es clara y efectiva en la presentación de los datos ambientales. El diseño, que se basa en gráficos y dashboards en Power BI, es efectivo para transformar conjuntos de datos posiblemente complejos en visualizaciones comprensibles y fácilmente interpretables.

No obstante, es importante que el 10% no coincidió con esta percepción. Esto señala áreas de mejora en la interfaz, relacionadas con la navegación, la disposición de la información o la claridad en los gráficos.

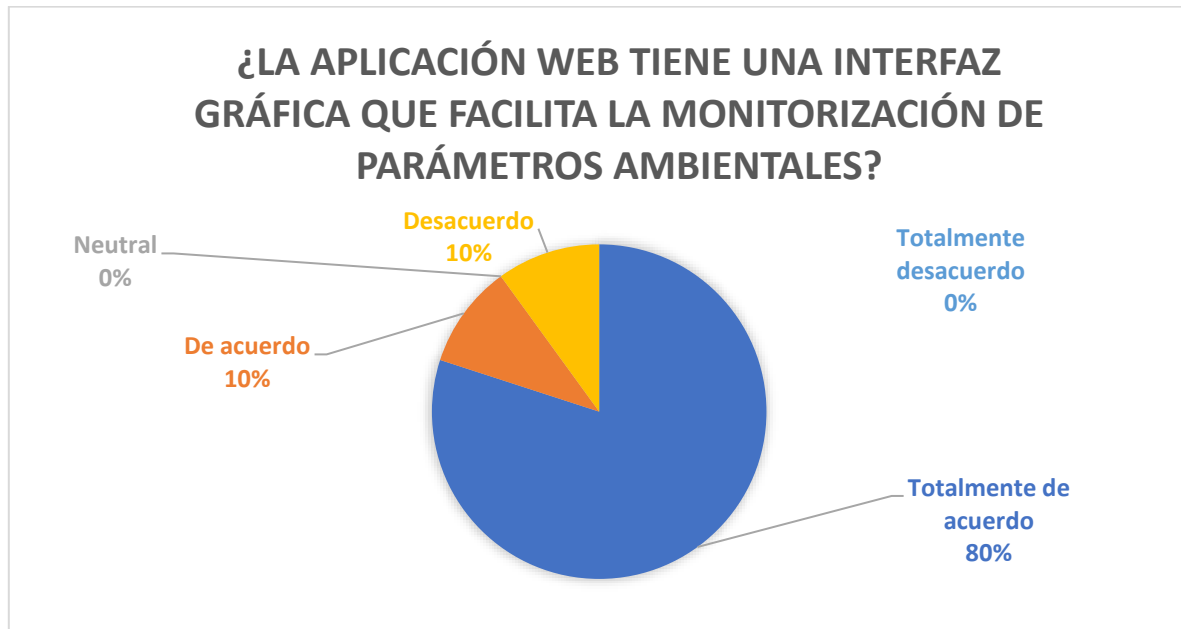


Figura 44. Resultados pregunta 2

Pregunta 3

¿El dispositivo proporciona datos precisos?

Con un 100% de respuestas entre acuerdo y totalmente de acuerdo, esta respuesta sugiere una confianza inequívoca en la precisión del dispositivo. Esta confianza no solo se basa en las respuestas de la encuesta, sino también en la experiencia práctica: al detectar que los datos sobrepasaban los umbrales de temperatura y al verificarlos en el invernadero, el jefe agrícola podía comparar esta información con un termómetro convencional. Esta validación práctica fortalece la confianza en el prototipo y subraya su relevancia en un contexto donde las condiciones óptimas son cruciales para la salud de los cultivos dentro de los invernaderos. Es fundamental resaltar que decisiones basadas en datos inexactos pueden llevar a un uso inadecuado de recursos, con repercusiones en la producción y salud del cultivo. Por lo tanto, la certeza en la precisión de la herramienta no solo garantiza un manejo

óptimo de los cultivos, sino que también fomenta una mayor confianza y dependencia en la tecnología entre los trabajadores agrícolas.

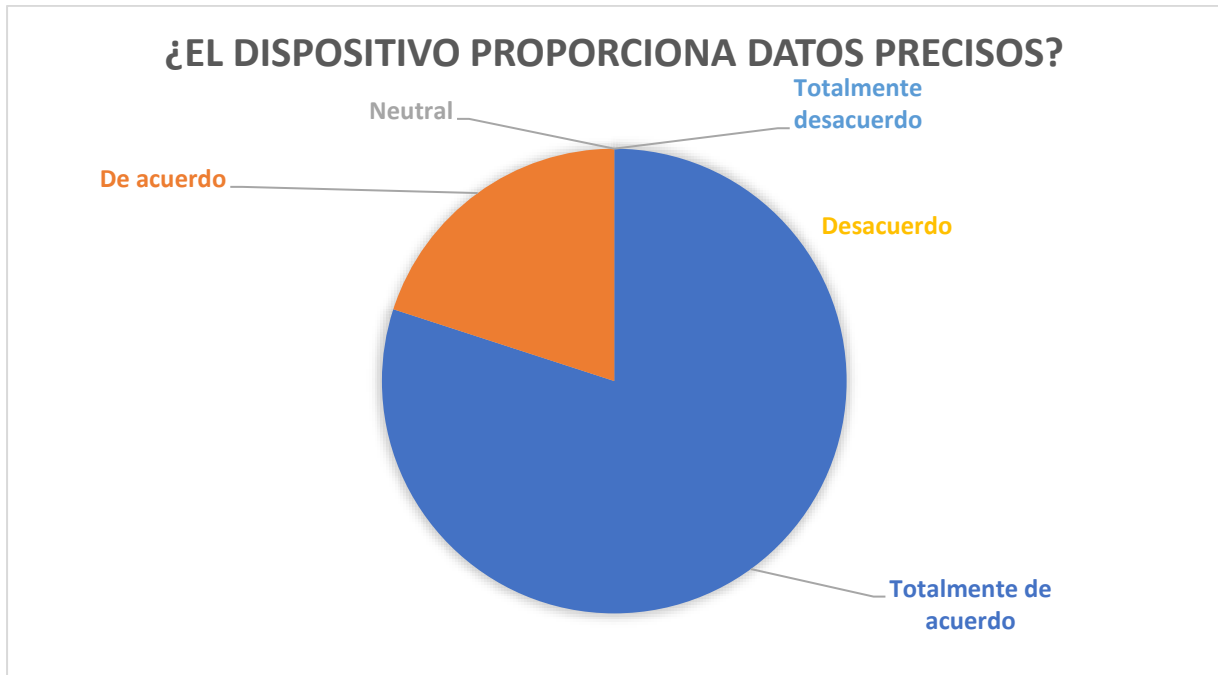


Figura 45. Resultados pregunta 3

Pregunta 4

¿El dispositivo contribuye a la optimización del tiempo laboral?

El 90% de respuestas positivas es un claro indicativo de que el prototipo ha logrado integrarse efectivamente en las operaciones diarias del invernadero. Esta adopción no solo resalta la capacidad del dispositivo para proveer información vital, sino que también subraya su potencial en agilizar procesos. En el ámbito agrícola, donde las condiciones cambiantes y las demandas constantes requieren una atención continua, el uso de tecnología que puede liberar tiempo de los trabajadores sin comprometer la calidad del monitoreo es invaluable. La respuesta no solo sugiere que el prototipo cumple con este ideal, sino que también reafirma la premisa de que una herramienta tecnológica bien diseñada debe actuar como un facilitador en las operaciones, y no como un obstáculo.

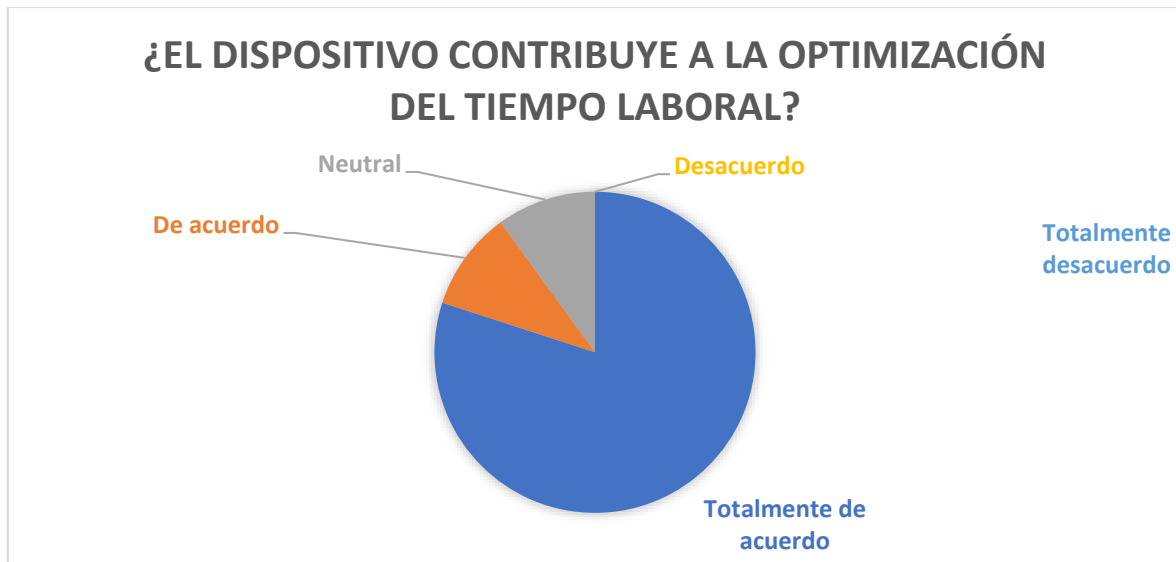


Figura 46. Resultados pregunta 4

Pregunta 5

¿El dispositivo contribuye al ahorro de agua en los cultivos?

El 70% de las respuestas afirmativas sugiere que una mayoría de los encuestados reconoce la capacidad del dispositivo para contribuir al ahorro de agua en los cultivos. Sin embargo, es vital señalar que el éxito en el ahorro de agua no depende únicamente del monitoreo, sino de la interpretación y aplicación adecuada de los datos recaudados. Un entendimiento profundo de la temperatura y la humedad puede guiar prácticas de riego más eficientes, optimizando el uso del agua. El 30 % indica que al no tener un control de la cantidad o flujo de agua que se utiliza para el riego no se puede asegurar que existe un ahorro de agua.



Figura 47. Resultados pregunta 5

Pregunta 6

¿El prototipo tiene un impacto positivo en la mejora de la producción agrícola?

El 90% de respuestas confirman el impacto positivo del prototipo en la producción agrícola, resaltando la trascendencia de un monitoreo efectivo. Estos datos en tiempo real, cuando se interpretan y aplican correctamente, pueden establecer prácticas agrícolas robustas y resilientes. Además, la capacidad de anticiparse a cambios y ajustar estrategias con base en la información proporcionada por el prototipo se traduce en una producción de mayor calidad y consistencia. Sin embargo, la presencia de un 10% de respuestas menos afirmativas nos indica mejoras en la implementación del dispositivo, sensibilización de su uso y de esta manera asegurar que todos los usuarios comprendan y aprovechen al máximo los beneficios de esta tecnología.

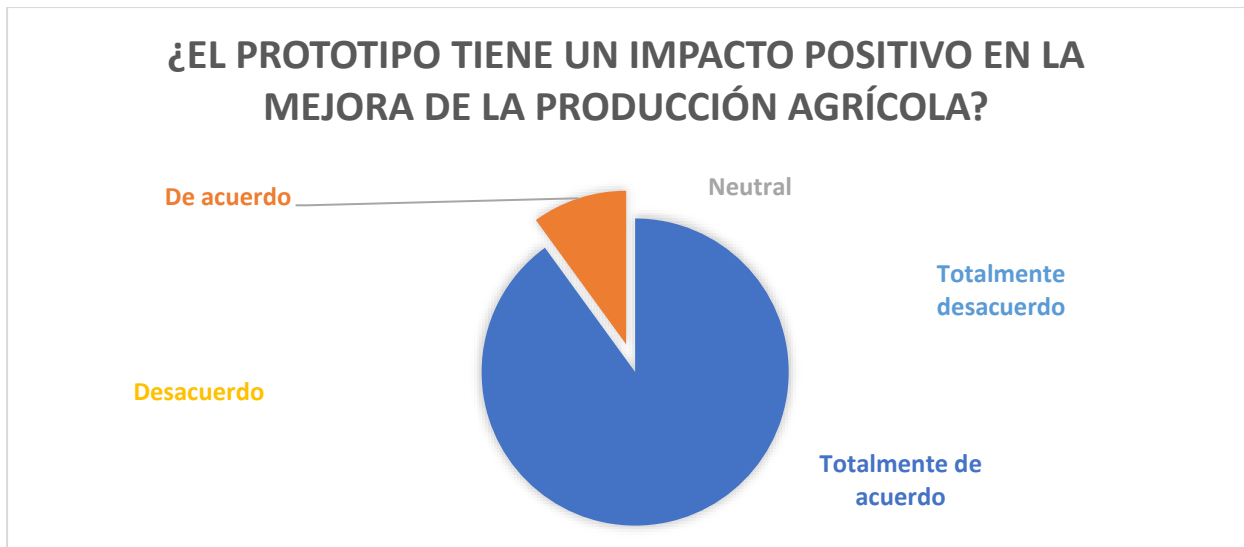


Figura 48. Resultados pregunta 6

Pregunta 7

¿El uso del aplicativo web facilita el trabajo realizado en el invernadero?

El aplicativo web ha recibido una aceptación unánime, evidenciando la valía de las herramientas digitales en el contexto agrícola moderno. El hecho de que todos los encuestados consideren que facilita el trabajo es indicativo de cómo la digitalización puede ser una aliada potente en la gestión y operación de invernaderos.



Figura 49. Resultados pregunta 7

Pregunta 8

¿Recomendaría el uso continuo del prototipo y su expansión a los otros invernaderos?

La totalidad de respuestas favorables, con un 100% de aprobación, destaca la confianza y apreciación de los trabajadores hacia el prototipo. En el contexto agrícola, se pueden encontrar resistencia hacia la adopción de nuevas tecnologías debido a tradiciones o desconfianza hacia lo desconocido, este nivel de aceptación es notablemente significativo. La respuesta sugiere que el prototipo no solo ha demostrado ser útil en la monitorización y gestión de los parámetros ambientales, sino que también ha sido fácilmente integrable en las rutinas diarias de trabajo.



Figura 50. Resultados pregunta 8

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La solución propuesta y desarrollada en esta tesis que consiste en la implementación de un prototipo denominado PAI-IoT para la empresa ALEPROCA ha demostrado ser una contribución para mejorar la producción agrícola. El despliegue de dos prototipos ha arrojado beneficios notables en términos de monitoreo de datos y optimización de la producción al generar 72 lecturas diarias por invernadero.
- El PAI-IoT ha permitido a la empresa ALEPROCA obtener datos precisos en tiempo real y ha generado una valiosa base de datos histórica en sus invernaderos para mejorar la producción a futuro. Estos datos en tiempo real han proporcionado una base sólida para la toma de decisiones, facilitando ajustes más eficientes en los procesos de cultivo y, en última instancia, traduciéndose en una mejora palpable en la producción agrícola.
- La revisión y selección de la arquitectura para la implementación de aplicaciones IoT fue un paso crítico para garantizar un diseño óptimo. Al evaluar diferentes enfoques, se identificó la arquitectura de 3 capas que permitió simplificar mucho el desarrollo del prototipo y permitió cumplir con los objetivos planteados.
- El diseño del prototipo IoT involucró la identificación de variables y sensores relevantes, la definición de requisitos de comunicación y el establecimiento de mecanismos de recopilación y procesamiento de datos. Esto garantiza la monitorización en tiempo real de los datos del invernadero, proporcionando información valiosa para la toma de decisiones por parte del jefe de producción agrícola.
- Una característica importante del diseño del prototipo radica en la auto sustentabilidad debido a que se garantiza un uso óptimo de energía maximizando la duración de la batería del prototipo IoT evitando la necesidad de reemplazar la batería o depender de estar conectado a la luz.
- La información mostrada o desplegada por el prototipo permitió al jefe agrícola y los trabajadores a cargo del invernadero en realizar ajustes precisos en los sistemas de riego, ventilación y control de plagas, lo cual se traduce en un mejor rendimiento en los cultivos.

- El despliegue del prototipo IoT en un entorno real implicó la construcción del prototipo, requirió la instalación física del prototipo en los invernaderos, la configuración de la red y la realización de pruebas para garantizar su correcto funcionamiento. Esto aseguró que el prototipo esté listo para operar en condiciones reales y capturar datos en tiempo real.³
- La validación del prototipo implementado implicó la realización de pruebas y evaluaciones para verificar su correcto funcionamiento y su capacidad para cumplir con los requisitos y objetivos establecidos. Esto incluyó evaluar su rendimiento mediante los datos recopilados, la confiabilidad de la comunicación, la usabilidad del aplicativo web y la mejora de la producción agrícola en los invernaderos monitorizados.
- Durante la validación, fue crucial involucrar a los usuarios finales, como el jefe de producción agrícola y el personal encargado de trabajar en los invernaderos. Su retroalimentación y experiencia de uso fueron fundamentales para identificar posibles mejoras y ajustes necesarios en el prototipo. Esto permitió optimizar su funcionamiento, hacerlo más intuitivo y garantizar la satisfacción de los usuarios.
- La aceptación generalizada y la recomendación unánime por parte de los encuestados sobre el uso continuo del prototipo y su expansión a otros invernaderos de la empresa respaldan el valor del PAI-IoT. Los empleados están conscientes de los beneficios que ofrece el prototipo, como la facilidad en la obtención de datos precisos y la optimización del tiempo de trabajo. Esto indica que el prototipo ha sido una solución bien recibida y es una inversión valiosa para ALEPROCA.
- Dado que el aplicativo web se centra en la monitorización de temperatura y humedad en un invernadero, un puntaje superior al 80% en la escala SUS resalta su efectividad y facilidad de uso. Una calificación de A+ sugiere que los usuarios relacionados con la gestión de invernaderos encontraron la herramienta intuitiva, especialmente útil para tomar decisiones en tiempo real. La evaluación favorable indica que el aplicativo es una herramienta valiosa para la gestión de invernaderos y siempre será fundamental continuar con su mejora y adaptación a las necesidades cambiantes del sector agrícola.

4.2 Recomendaciones

- En cada invernadero de ALEPROCA, es recomendable instalar una estación equipada con puntos de red y eléctricos. Esta infraestructura asegura que la placa ESP8266, pueda mantener una conexión ininterrumpida a la red Wi-Fi de la empresa. Así, no solo se garantiza una transmisión fluida de datos, sino también una monitorización constante de las condiciones del invernadero.
- Para proteger y organizar la transmisión de datos, se sugiere la creación de una VLAN específica en la red de la empresa. Integrando todos los prototipos a esta red segregada, se evitarían potenciales vulnerabilidades y se facilitaría el monitoreo y gestión centralizada de los datos.
- Dado el asequible costo de producción y la demostrada eficacia del prototipo, sería beneficioso expandir su implementación más allá de ALEPROCA. Tanto agricultores individuales como grandes conglomerados agrícolas podrían beneficiarse. Además, su aplicación puede ir más allá de invernaderos; sistemas como los cultivos hidropónicos, floricultura y semilleros también podrían ver mejoras significativas con su implementación.
- Con los avances continuos en inteligencia artificial, se podría potenciar el prototipo incorporando algoritmos que procesen y analicen los datos en tiempo real. Estos algoritmos podrían identificar patrones y tendencias, anticipando las necesidades de los cultivos y permitiendo intervenciones proactivas, lo que reduciría el desperdicio de recursos y mejoraría la producción.
- A medida que la ciencia avanza en el desarrollo de sensores más precisos y multifuncionales, es recomendable mantenerse al tanto y considerar su adopción. Sensores que puedan detectar aspectos más detallados, como la composición química del suelo o indicadores tempranos de enfermedades en las plantas, serían una adición invaluable.
- La tendencia global se inclina hacia la automatización agrícola. Considerando esto, sería propicio que ALEPROCA evalúe la integración de herramientas complementarias al prototipo como drones y robots. Estos dispositivos pueden realizar desde tareas básicas, como el monitoreo visual de los cultivos, hasta operaciones más complejas, como la detección y tratamiento de plagas.
- La implementación del prototipo debe realizarse teniendo en cuenta el ecosistema tecnológico de ALEPROCA. Es crucial que haya compatibilidad con sistemas operativos y bases de datos como SQL Server. Además, es esencial que las conexiones a Internet sean robustas y confiables para evitar interrupciones.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] Aleproca, «<https://aleproca.com/quienes-somos/>,» [En línea].
- [2] M. Pérez, L. Quebrajo, J. Martinez y J. Aguera, *Introducción a la agricultura de precisión en el Valle Guadalquivir*, 2015.
- [3] QAMPO, «La agricultura de precisión,» 2019. [En línea]. Available: <https://qampo.es/la-agricultura-de-precision/>.
- [4] S. E. L. C. Karen Rose, «Internet Society,» 15 Octubre 2015. [En línea]. Available: <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>. [Último acceso: Octubre 2021].
- [5] S. Shweta y G. Dhanashri , «IoT based smart irrigation monitoring and controlling system,» *nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 815-819, 2017.
- [6] H. A. a. R. M. a. E. D. S. E. Khattak, «CoAP-based healthcare sensor networks: A survey,» pp. 499-503, 14-18 enero 2014.
- [7] «AMQP: Advanced Message Queuing Protocol,» Asociación de la Industria de IoT. (2019). [En línea].
- [8] D. C. Yacchirema Vargas, «Arquitectura de interoperabilidad de dispositivos físicos para el internet de las cosas (IoT),» Valencia, España, 2019.
- [9] R. & P. A. Tiwari, «An internet of things approach for motion detection using wireless sensor networks.,» 2011. [En línea]. Available: <https://www.emerald.com/insight/content/doi/10.1108/17538371111164029/full/html..>
- [10] I. S. Jaimes Montaña y W. Y. Cendales Cabeza, «Desarrollo de una plataforma web para un sistema de control IoT sobre ARDUINO y MQTT,» 2020.
- [11] RAE, «Real Academia Española,» 2022. [En línea]. Available: <https://dle.rae.es/agricultura>.
- [12] F. Coll, «Economipedia,» 2021. [En línea]. Available: <https://economipedia.com/definiciones/agricultura.html>.

- [13] L. Elkin, «Prototipo de un sistema WEB para el monitoreo de productos agrícolas en cultivos urbanos,» Bogotá, 2020.
- [14] J. Ortiz, «Desarrollo de un prototipo de un sistema de riego automatizado para el procesamiento, monitoreo y análisis de datos utilizando lógica difusa en tiempo real e IoT para optimizar el uso de agua aplicada en el cultivo,» Cuenca, 2021.
- [15] L. Burbano, «Estudio e implementación en Matlab de un entorno de comunicación basado en protocolos del Internet de las Cosas para clientes de Teleoperación en Robótica,» Quito, 2017.
- [16] H. Vite, O. Vargas, L. Vargas y J. Vargas, «Internet de las cosas aplicado a la producción agropecuaria,» Editorial Grupo Compás, Guayaquil, 2019.
- [17] I. La Torre y D. Apaza, «Diseño e implementación de un sistema automatizado para riego tecnificado basado en el balance de humedad de suelo con tecnología Arduino en el laboratorio de control y automatización EPIME 2016,» Puno, 2017.
- [18] J. Gómez, F. Marcillo, F. Triana, V. Gallo, B. Oviedo y V. Hernández, «IoT for environmental variables in urban areas,» de *The 8th International Conference on Ambient Systems, Networks and Technologies*, 2017.
- [19] E. Jalón , L. Albarracín, L. Molina y J. Laverde, «Inserción de un sistema automatizado de riego por aspersión para parques del cantón Quevedo,» *Revista Dilemas Contemporáneos*, vol. Edición Especial, nº 59, pp. 1-12, 2019.
- [20] J. Aleotti, M. Amoretti, A. Nicoli y S. Caselli, «A Smart Precision-Agriculture Platform for Linear Irrigation Systems,» de *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Croatia, 2018.
- [21] FAO, «Agricultura Familiar en América Latina y el Caribe,» Santiago, 2014.
- [22] J. Laverde y C. Laverde, «Internet de las cosas aplicado en la agricultura ecuatoriana: Una propuesta para sistemas de riego,» *Revista Dilemas Contemporáneos: Educación, Política y Valores.*, vol. 31, nº 2, pp. 1-14, 2021.
- [23] M. Viteri y M. Tapia, «Economía ecuatoriana: de la producción agrícola,» *Revista Espacios*, vol. 39, nº 32, p. 30, 2018.

- [24] R. García, «Optimización de consultas en bases de datos relacionales,» Universidad Carlos III de Madrid, 2016.
- [25] N. Livias, «Conceptos de programación visual, fundamentos del Visual NET. elementos de MS Visual, sentencias de control, procedimientos y matrices de controles, acceso a una base de datos, asistente para aplicaciones,» Universidad Nacional de Educación, 2018.
- [26] C. Hervas, «Análisis de rendimiento de protocolos de Publicación/Subscripción en comunicación con una Red de Sensores Inalámbricos Zigbee,» Universidad Nacional de La Plata, 2018.
- [27] E. Córdova, «Sistema electrónico de monitoreo y control de captación de energía solar y electromagnética para aplicaciones IoT,» Universidad Técnica de Ambato, 2022.
- [28] C. Hervas, «Análisis de rendimiento de protocolos de Publicación/Subscripción en comunicación con una Red de Sensores Inalámbricos Zigbee,» Universidad Nacional de La Plata, 2018.
- [29] J. Bonilla, «Desarrollo de un sistema para gestión de roles de pago con envío de notificaciones en tiempo real,» Escuela Politécnica Nacional, 2019.
- [30] O. Aparicio, C. Díaz y F. Ortiz, «Análisis, diseño e implementación de una solución en comunicaciones unificadas utilizando servidor Elastix, para las pequeñas y medianas empresas (PYMES),» Corporación Universitaria de la Costa, 2012.
- [31] F. López, *Sistemas distribuidos*, México: UAM, Unidad Cuajimalpa, 2015.
- [32] N. Cobo, «Diseño de una arquitectura para el control telemático de sistemas ciber físicos basada en microservicios y con tolerancia a fallos,» ESPE, 2021.
- [33] G. Blacio, «Diseño de una red para voz sobre IP en la nube y posible implementación con HTML5,» Escuela Superior Politécnica del Litoral, 2013.
- [34] A. Rodas y A. Valencia, «Desarrollo e implementación de un prototipo para una plataforma tecnológica para la transmisión de texto y video (streaming) en tiempo real empleando tecnología websocket,» *Ingenierías USBmed*, vol. 9, nº 2, pp. 2-10, 2018.

- [35] J. Calvillo, «Contribuciones al Diseño de Arquitecturas de Sistemas Distribuidos Abiertos para la Provisión de Servicios del Cuidado de la Salud y de Soporte a la Autonomía del Ciudadano,» Universidad de Sevilla, 2012.
- [36] A. Johnson, «Improved Water Management Using IoT in Precision Agriculture.,» *IEEE Sensors Journal*, vol. 18, nº 23, pp. 9426-9434, 2018.
- [37] B. Smith, «Wireless Sensor Network for Smart Irrigation Management in Precision Agriculture,» *IEEE Transactions on Instrumentation and Measurement*, vol. 67, nº 2, pp. 387-397, 2018.
- [38] S. S. R. J. Jackson, «IoT-Enabled Smart Farming: A Systematic Review of Literature and Directions for Future Research.,» *IEEE Access*, vol. 6, pp. 6492-6503, 2018.
- [39] C. e. a. Anderson, «Smart Farming: Monitoring Beef Cattle Behavior Using IoT and Machine Learning Techniques.,» *IEEE Sensors Journal*, nº 22, pp. 10786-10794, 2019.
- [40] S. R. J. Jackson, «IoT-Enabled Smart Farming: A Systematic Review of Literature and Directions for Future Research.,» *IEEE Access*, vol. 6, pp. 6492-6503, 2018.
- [41] E. o. d. p. A. 1. p. p. OASIS, «https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=amqp,» [En línea].
- [42] M. a. R. M. a. S. V. a. B. P. Singh, «Secure MQTT for Internet of Things (IoT),» 2015. [En línea]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7280018&isnumber=7279856>.
- [43] J. Sandoval, «Análisis y diseño de un sistema de monitoreo y control de temperatura y humedad en un ambiente de invernadero utilizando tecnología IoT.,» Universidad de las Fuerzas Armadas ESPE, [En línea]. Available: <https://repositorio.espe.edu.ec/bitstream/21000/27321/1/T-ESPEL-MEI-0009.pdf>.
- [44] PickData, «MQTT vs CoAP: ¿Mejor protocolo IoT?,» PickData.net., [En línea]. Available: <https://www.pickdata.net/es/noticias/mqtt-vs-coap-mejor-protocolo-iot..>
- [45] M. O. V. P. M. R. C. G. C. C. M. B. & Cruz Vega, Las tecnologías IOT dentro de la industria conectada: Internet of things, Madrid: Fundación EOI, 2015.