



Universidad
LOYOLA

Escuela Técnica Superior de Ingeniería
Grado en Ingeniería Mecatrónica y Robótica

Trabajo Fin de Grado

**Sistema de monitorización y control
remoto de las condiciones de cultivo
para un invernadero con tecnología IoT
y energía solar**

Iván Vilariño Martel

Tutor: James Brian Romaine

Sevilla, Julio de 2023

*A mi familia, amigos y seres queridos por su
paciencia y apoyo constante; A mis
profesores y mentores por su invaluable
guía. Este logro refleja mi perseverancia y
estoy orgulloso de ello.*

Ad maiorem Dei gloriam.

IHS.

Resumen

Este trabajo de investigación tiene como objetivo desarrollar un invernadero inteligente que incorpore tecnologías innovadoras para mejorar la eficiencia y la sostenibilidad de la producción agrícola. La solución propuesta utiliza tecnologías IoT (Internet de las Cosas) y sistemas de automatización, permitiendo a los agricultores monitorizar la humedad y la temperatura de los cultivos, controlar el riego de forma remota a través de una plataforma online y reducir el consumo de agua/energía evitando el derroche en zonas donde no son necesarios.

Además, este invernadero inteligente pretende abordar los problemas de complejidad asociados a la instalación de nodos/sensores de comunicación con servidores mediante la implementación de un servidor altamente modular y fácil de usar. Se utilizarán módulos solares para generar energía renovable con el fin de reducir la dependencia de la red y minimizar la huella de carbono.

La solución propuesta ofrece numerosos beneficios al abordar retos como el cambio climático, que afecta a los rendimientos agrícolas en todo el mundo. Se ha demostrado que el uso de tecnologías IoT en la agricultura aumenta el rendimiento y reduce el uso de recursos, ofreciendo soluciones sostenibles para las generaciones futuras.

Palabras clave: Invernadero inteligente, Internet de las cosas (IoT), Monitorización de cultivos, Control de riego, Energía solar, Sostenibilidad

Abstract

This research project aims to develop a smart greenhouse that incorporates innovative technologies to improve the efficiency and sustainability of agricultural production. The proposed solution uses IoT (Internet of Things) technologies and automation systems, allowing farmers to monitor crop humidity and temperature, control irrigation remotely through an online platform and reduce water/energy consumption by avoiding waste in areas where they are not needed.

In addition, this smart greenhouse aims to address the complexity issues associated with the installation of server communication nodes/sensors by implementing a highly modular and easy-to-use server. Solar modules will be used to generate renewable energy to reduce grid dependency and minimize the carbon footprint.

The proposed solution offers numerous benefits in addressing challenges such as climate change, which affects agricultural yields worldwide. The use of IoT technologies in agriculture has been proven to increase yields and reduce resource use, offering sustainable solutions for future generations.

Keywords: Smart greenhouse, Internet of Things (IoT), Crop monitoring, Irrigation control, Solar energy, Sustainability.

Índice

.....	1
RESUMEN	3
ABSTRACT	4
ÍNDICE	5
1. INTRODUCCIÓN	6
1.1 OBJETIVOS.....	7
1.2 METODOLOGÍA.....	7
1.3 DEFINICIÓN DE INVERNADERO INTELIGENTE.....	8
1.4 IMPORTANCIA DE LOS INVERNADEROS INTELIGENTES.....	9
1.5 HISTORIA DE LOS INVERNADEROS INTELIGENTES	10
2. ESTADO DEL ARTE	12
2.1 OBJETIVOS DEL ESTADO DEL ARTE	12
2.2 TECNOLOGÍAS IoT Y APLICACIONES	12
2.3 ESCENARIOS DE IMPLEMENTACIÓN IoT EN INVERNADEROS	13
2.3.1 AGRICULTURA TRADICIONAL	14
2.3.2 AGRICULTURA HIDROPÓNICA	14
2.3.3 AGRICULTURA VERTICAL.....	15
2.4 VENTAJAS Y DESVENTAJAS DE LOS INVERNADEROS INTELIGENTES.....	16
2.4.1 VENTAJAS.....	16
2.4.2 DESVENTAJAS.....	17
2.5 CONCLUSIÓN Y FUTURAS PERSPECTIVAS	17
3. MI TRABAJO	19
3.1 INTRODUCCIÓN.....	19
3.2 COMPONENTES	19
3.2.1 HARDWARE	21
3.2.2 SOFTWARE	33
3.2.3 ACCESORIOS	45
3.3 IMPLEMENTACIÓN.....	47
OE.1. FRAMEWORK IoT	47
OE.2. SISTEMA DE COMUNICACIÓN ASÍNCRONA.....	53
OE.3. COMUNICACIÓN POR RADIOFRECUENCIA	56
OE.4. FUENTES DE DATOS	57
OE.5. TOMA DE DECISIONES	59
OE.6. SISTEMA DE CONTROL POR VOZ.....	62
OE.7. USO DE ENERGÍA SOLAR	66
4. RESULTADOS Y CONCLUSIONES	70
4.1 EVIDENCIAS	70
4.2 PROBLEMAS ENCONTRADOS	74
4.3 TRABAJO FUTURO	75
ANEXO I – CONFIGURACIÓN AVANZADA HA	82
ANEXO II – SCHEMATICS Y DIAGRAMAS	87
ANEXO III - CÓDIGOS	89

1. Introducción

En los últimos años, la agricultura ha experimentado grandes cambios, debido sobre todo a la integración de tecnologías innovadoras. Estas tecnologías han mejorado notablemente las opciones de cultivo, lo que se ha traducido en una mayor productividad de estos. Como resultado, los invernaderos inteligentes han ganado gran popularidad como solución eficaz para cultivar en condiciones cuidadosamente reguladas.

Estos invernaderos se basan en el uso de tecnologías avanzadas como sensores, sistemas de control automático y software de análisis de datos. Estas tecnologías permiten a los agricultores controlar variables importantes como la temperatura, la humedad y la luz, creando el entorno perfecto para el crecimiento óptimo de las plantas.

En este contexto, se utilizan sensores para medir las condiciones ambientales del invernadero y proporcionar información en tiempo real sobre factores como la temperatura y la humedad.

La introducción de estas tecnologías ha permitido a los agricultores maximizar el rendimiento de los cultivos y minimizar el uso de recursos naturales como el agua y los fertilizantes. La capacidad de supervisar y controlar las condiciones ambientales en tiempo real también ha permitido a los agricultores adaptar sus métodos de cultivo a las necesidades específicas de sus cosechas, lo que se traduce en mejores rendimientos y mayores volúmenes.

A pesar de las ventajas que ofrecen los invernaderos inteligentes para optimizar la producción agrícola, su implantación puede estar plagada de dificultades técnicas. Uno de los problemas más comunes es la complejidad de la instalación de nodos y sensores en el invernadero.

La instalación de sensores en un invernadero puede ser un proceso complejo que requiere amplios conocimientos técnicos. Los sensores deben instalarse en distintas zonas del invernadero para controlar la temperatura, la humedad y otros parámetros ambientales. Además, los sensores deben conectarse a una red de nodos que recogen, procesan y envían datos a un servidor central.

Esta complejidad de instalar sensores y nodos en los invernaderos puede suponer un gran problema. La falta de conocimientos técnicos puede hacer que la instalación de sensores y nodos sea un proceso costoso y lento.

La complejidad de la transmisión de datos también puede acarrear problemas de fiabilidad en los resultados de las mediciones y en la toma de decisiones basada en ellos. Si los datos no se transmiten o procesan correctamente, se pueden tomar decisiones incorrectas que repercutan negativamente en la producción agrícola. Se necesita una solución que simplifique la instalación de sensores y nodos para mejorar la fiabilidad de las mediciones y la transmisión de datos.

1.1 Objetivos

El objetivo de este trabajo es desarrollar un invernadero inteligente que incorpore tecnologías innovadoras para mejorar la eficiencia y la sostenibilidad en la producción agrícola. En concreto, se monitorizarán la humedad y la temperatura de las plantas y se controlará el riego a distancia a través de una plataforma online. También se propone el uso de paneles solares para reducir el consumo energético del invernadero y hacerlo más sostenible.

Además, se abordará el problema de la complejidad relacionada con la instalación de los nodos y sensores del invernadero y su comunicación con el servidor y visualización de datos.

Para lograr estos objetivos, se utilizarán tecnologías y sistemas de automatización IoT (Internet de las Cosas) y se implementará un servidor altamente modular y fácil de usar, en dónde se desplegará una aplicación para la visualización de datos y control de actuadores. Esto permitirá a los agricultores supervisar las condiciones ambientales del invernadero en tiempo real, ajustar automáticamente los sistemas de riego para garantizar que las plantas reciban la cantidad adecuada de agua y escalar fácilmente sus datos según sea necesario. Además, el uso de tecnologías IoT reducirá el consumo de agua y energía al evitar el derroche de recursos en zonas donde no son necesarios.

Además, el invernadero inteligente propuesto tratará de reducir la huella de carbono mediante el uso de paneles solares. Esta tecnología permitirá a los invernaderos generar su propia energía renovable, reduciendo la dependencia de la red eléctrica y su impacto en el medio ambiente.

1.2 Metodología

Para alcanzar estos objetivos, se ha desarrollado un método que consta de varios pasos. En primer lugar, se realizará un estudio del estado del arte de las tecnologías más avanzadas en el campo de los invernaderos inteligentes y se identificarán las tecnologías y soluciones más adecuadas para llevar a cabo el proyecto. A continuación, se desarrollará un sistema de monitorización y control basado en un microcontrolador para

leer los sensores de temperatura y humedad y controlar el riego a través de la bomba de agua.

Además, se desarrollará una plataforma en línea que permitirá a los usuarios acceder al sistema de control del invernadero desde cualquier parte del mundo, lo que facilita el seguimiento y la gestión de los cultivos. Por último, se instalarán paneles solares funcionales, junto con baterías que pueden alimentar el sistema de forma autónoma y sostenible.

Este trabajo contribuirá al desarrollo de soluciones agrícolas inteligentes que mejoran la eficiencia y la sostenibilidad de los procesos agrícolas. También demostrará las ventajas de utilizar energías renovables en el sector agrícola y promoverá la adopción de tecnologías más sostenibles en el sector.

1.3 Definición de Invernadero Inteligente

Los invernaderos inteligentes son una tecnología emergente que combina las ventajas de los invernaderos y los sensores IoT para mejorar el rendimiento de los cultivos y reducir el impacto de la agricultura en el medio ambiente [3].

El término "Internet de las cosas" (IoT) hace referencia a un sistema masivo conectado a numerosos sensores, controladores integrados, plataformas de toma de decisiones, Internet y un servidor en la nube. Los sensores recogen datos y los envían automáticamente al servidor en la nube. Los servidores en la nube almacenan los datos y permiten el acceso remoto a los mismos. Por tanto, los sistemas IoT se utilizan principalmente para supervisar y tomar medidas inteligentes sin intervención humana, como refrigeración, calefacción, iluminación, riego, encendido y apagado de motores y actuadores mediante el análisis de los datos de los sensores, lo que convierte al invernadero en inteligente.

Los invernaderos inteligentes también pueden ayudar a los agricultores a obtener información sobre la mejor temporada o momento para cosechar, la calidad del suelo, la cantidad de nutrientes necesarios para el crecimiento saludable de las plantas, la calidad del agua y muchos otros factores importantes. En otras palabras: Los invernaderos inteligentes pueden hacer que la agricultura sea fiable y rentable, maximizando el rendimiento de los cultivos con una labranza mínima. Esto puede ayudar a los agricultores a gestionar sus explotaciones y utilizar los recursos de forma eficiente. En el futuro, el cultivo en un invernadero inteligente (o al menos parte del proceso) podría estar totalmente automatizado y controlado a distancia [2].

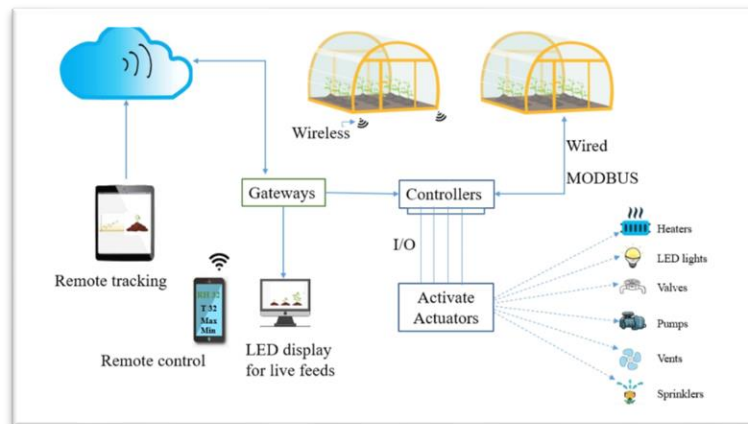


Figura 1.1 Diagrama de la arquitectura de un Invernadero Inteligente

Este trabajo pretende explorar el diseño y la implementación de un invernadero inteligente para monitorizar la humedad y la temperatura de las plantas y controlar el riego desde cualquier parte del mundo, así como implementar módulos funcionales con energía solar.

1.4 Importancia de los Invernaderos Inteligentes

El uso de invernaderos inteligentes es cada vez más importante ante el cambio climático mundial y la creciente demanda de alimentos. Según las Naciones Unidas, se prevé que la población mundial alcance los 9 700 millones de habitantes en 2050, lo que requerirá un aumento del 60% en la producción mundial de alimentos [1]. El desarrollo de tecnologías de invernaderos ha sido posible gracias al Internet de las Cosas (IoT), que permite la recopilación y el análisis de datos en tiempo real para posibilitar un control preciso de factores ambientales como la temperatura, la humedad y la humedad del suelo.

Al integrar tecnologías como sensores IoT, sistemas hidropónicos, iluminación LED, energía solar y aprendizaje automático, los invernaderos son capaces de crear un entorno optimizado y controlado para el crecimiento de las plantas. Esto permite un uso más eficiente de los recursos, así como una reducción del uso de fertilizantes y pesticidas químicos. Además, se ha demostrado que la tecnología en los invernaderos reduce significativamente el consumo de agua y energía, así como las emisiones de gases de efecto invernadero.

La importancia de los invernaderos inteligentes radica en su capacidad para abordar muchos de los retos de sostenibilidad a los que se enfrenta la agricultura moderna, como la escasez de agua, el consumo de energía y la degradación del medio ambiente. Tales, tienen el potencial de desempeñar un papel crucial en el futuro de la agricultura y la producción de alimentos, especialmente en las zonas urbanas, donde el espacio es limitado y el acceso a productos frescos puede ser difícil [2].

1.5 Historia de los invernaderos inteligentes

Desde 1991 [4], el cultivo en invernaderos ha aumentado considerablemente en todo el mundo. Esto se debe principalmente a la llegada de la tecnología del plástico, que abarata los costes de construcción. Sin embargo, la construcción de granjas en invernadero no sólo depende del material (plástico, vidrio). También depende en gran medida de las instalaciones [9]. El prerrequisito general para la construcción de un invernadero es la selección de un terreno llano que garantice suficiente luz solar (durante todo el año) [9]. El principio fundamental y la historia de los invernaderos se han utilizado para cultivar plantas u hortalizas que requieren calor para crecer en climas fríos. Esto se debe al efecto invernadero y al efecto cortavientos, que favorecen el crecimiento de las plantas. Sin embargo, con el desarrollo de nuevas tecnologías y sistemas inteligentes, los invernaderos se han vuelto más versátiles y ahora pueden construirse en cualquier lugar.

En general, las condiciones climáticas pueden dividirse en tres categorías principales: 1) clima invernal frío y templado 2) clima tropical y subtropical 3) clima semiinvernal árido o semiárido [9]. En función de las condiciones climáticas y de las inversiones de los agricultores, la forma, el material de construcción, la cubierta y la ventilación de los edificios pueden variar.

En la tabla 1.1 se resumen los tipos de explotaciones de invernaderos, sus materiales de construcción y los cultivos que se realizan en ellos. En cuanto a las tecnologías de cultivo, los invernaderos pueden dividirse principalmente en tres tipos [12], como se muestra en la Tabla 1.2.

Climatic conditions	Type of greenhouses	Construction materials	Crops grown
Cold or temperate	Cold frame [18] or solar greenhouse [19]	Solid plastic or glass [20]	Spinach, lettuce, carrots, scallion [21]
Tropical or sub-tropical	Shade house [22]	Metal and polyethylene fabric [23]	Cucumber, tomato, strawberries, capsicum, orchids [23]
Arid or semi-arid	Screen houses [16]	Steel and plastic film [24]	Musk melon, watermelon, hot pepper, cucumber, tomato [24]

Tabla 1.1 [2]

Greenhouse technologies	Construction material	Automation level	Crops grown	Cost(\$/ m ²)	Suitable climate
Low Technology Greenhouse	Wood	Poor	Vegetables and flowers	25-30	Cold or temperate
Medium Technology Greenhouse	Metals	Average	Vegetables, flowers and decorative plants	30-100	Tropical or sub-tropical
High Technology Greenhouse	Galvanized Iron	High	Vegetables, flowers and decorative plants	100-200	Any climate

Tabla 1.2 [2]

Los tipos de invernadero individuales se describen brevemente a continuación.

1) Invernadero de baja tecnología (LTG):

Los invernaderos de baja tecnología consisten en una estructura de madera simple, similar a una casa, con un techo cubierto de plástico. Suelen construirse en zonas frías o templadas. Este tipo de establecimiento puede, o no, tener un sistema de calefacción inferior, no tiene ventilación y generalmente tiene un control deficiente de las condiciones climáticas internas.

Estos invernaderos se utilizan normalmente para la producción de hortalizas y flores, y los costes de instalación son inferiores a 25-30 dólares por metro cuadrado de superficie total cultivada [5], [12].

2) Invernaderos de tecnología media (MTG):

Los invernaderos de tecnología media se construyen con estructuras metálicas. Los paneles de plástico y vidrio se utilizan como techo. En contraste con los sistemas LTG, los sistemas MTG son más eficientes en el control de las condiciones climáticas, lo que hace que la operación sea independiente de las condiciones externas. Esto facilita la construcción en climas tropicales o subtropicales. Las tecnologías de cultivo modernas, como la hidroponía, la aeroponía y los cultivos de película de nutrientes, utilizan MTG. Los MTG suelen ser granjas semiautomáticas y se pueden usar para cultivar flores de alto valor como rosas, plantas ornamentales en macetas y vegetales fuera de temporada. Los costos de instalación rondan los \$30-100 por metro cuadrado [5], [12].

3) Invernaderos de alta tecnología (HTG):

Los Invernaderos de Alta Tecnología (HTG) se construyen generalmente con estructuras de hierro galvanizado y cubierta de vidrio.

Las granjas HTG cuentan con todas las tecnologías avanzadas de control climático, lo que las hace adecuadas para todas las condiciones climáticas [13]. Pueden controlar la calefacción mediante la detección de las condiciones climáticas internas y proporcionar refrigeración y ventilación forzadas. Además, se puede controlar la humedad, el nivel de luz y el nivel de CO₂ en el invernadero. Los HTG se utilizan para cultivos verticales y pueden minimizar los costos de mano de obra agrícola. Son muy utilizados en regiones de clima frío para la producción de diversos tipos de hortalizas y plantas ornamentales. Los costos de instalación de los HTG oscilan entre \$100 y \$200 por metro cuadrado [5], [12].

2. Estado del arte

2.1 Objetivos del Estado del Arte

El estado del arte pretende ofrecer una visión general del desarrollo histórico de los invernaderos inteligentes, las tecnologías utilizadas en ellos, su diseño e implementación, sus aplicaciones, sus ventajas e inconvenientes, su futuro, así como casos prácticos y ejemplos de aplicación. Este estado del arte pretende ofrecer una visión global del estado actual de los invernaderos inteligentes y de sus posibles repercusiones en la agricultura y el medio ambiente.

2.2 Tecnologías IoT y aplicaciones

Los avances tecnológicos recientes han llevado al desarrollo de varios protocolos de IoT como ZigBee, REST, MQTT, LPWAN, LoRaWAN, Servicio de distribución de datos (DDS) [35][36], Protocolo de presencia y mensajería extensible (XMPP), Zwave e IPv6. [37][38]. Cada protocolo tiene aplicaciones específicas para cada contexto. Por ejemplo, protocolos de comunicación altamente eficientes como el protocolo MQTT (Message Queuing Telemetry Transport) [39][40][41] han reemplazado gradualmente al HTTP (Protocolo de transferencia de hipertexto) [42]. MQTT puede ejecutarse con menos ancho de banda, lo que genera menos sobrecarga. Del mismo modo, DDS ofrece beneficios prácticos, ya que permite mensajería fluida y de baja latencia, que es crucial para la comunicación entre agentes en redes inteligentes [43]; Esto explica por qué se ha utilizado en varios contextos, incluidas las microrredes y las industrias inteligentes, la defensa, las finanzas y la automoción [44][45][43]. Los defensores de DDS argumentan que se pueden personalizar fácilmente para admitir la interoperabilidad con RIOT-OS, FreeRTOS y pilas ZigBee independientes [46].

Otras aplicaciones notables de los nuevos sistemas IoT incluyen el uso de LoRa y NB-IoT para mejorar la conectividad del dispositivo y la QoS, la latencia, la confiabilidad y el alcance [47]; la fusión de sensores de calidad del aire y LPWAN para el monitoreo de CoVID-19 [48]; Protocolos de comunicación IoT y RFID, sensores inteligentes para comunicación máquina a máquina.

Sin embargo, es necesario abordar las barreras existentes para la introducción de nuevas tecnologías en el sector agrícola; incluyendo la relación desproporcionada entre velocidad, distancia y potencia (véase figura 2.1).

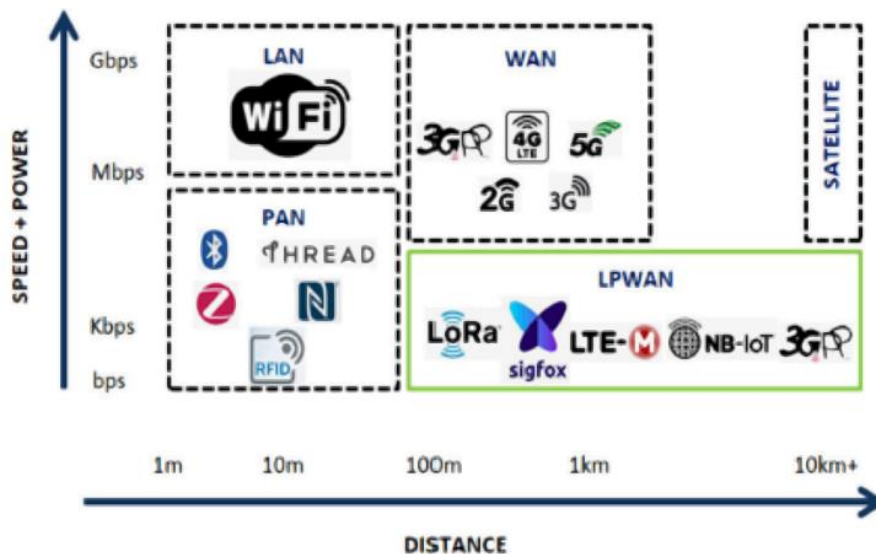


Figura 2.1 Impacto de la velocidad, potencia, distancia y elección de WAN y LAN [48].

2.3 Escenarios de implementación IoT en invernaderos

Hoy en día, los dispositivos IoT se utilizan ampliamente en los sistemas de control y diagnóstico de la industria agrícola [14], [15]. Las tecnologías IoT incluyen sensores, actuadores, servicios de datos basados en computación en la nube, drones, sistemas de navegación y análisis que permiten que la arquitectura tome decisiones inteligentes para aumentar el rendimiento de los cultivos [16].

Los dispositivos IoT pueden proporcionar información sobre variables ambientales como la humedad, la temperatura y las condiciones climáticas, así como variables de campo como el suelo y la biomasa vegetal [17].

Se puede utilizar para predecir y controlar la calidad de los cultivos para los consumidores [18]. Además, IoT se puede utilizar para recopilar datos y almacenarlos en dispositivos de computación en la nube para crear alertas y enviar mensajes cortos (SMS) a los agricultores. Los datos almacenados en la nube también se pueden usar para desarrollar modelos predictivos que pueden predecir las variables que afectan la cosecha [18].

Esta sección resume el estado actual de las arquitecturas IoT utilizadas en invernaderos, específicamente las tres técnicas típicas de cultivo en invernadero, es decir, tradicional, hidropónico y vertical.

2.3.1 Agricultura Tradicional

En 2012, Li et al. [19] desarrollaron un sistema de monitor ambiental basado en IoT para invernaderos mediante la integración de redes inalámbricas, redes móviles e Internet con el objetivo de realizar un control remoto en tiempo real de las plantas. La arquitectura consta de módulos de alta frecuencia CC2420 (ZigBee), sensores de temperatura y humedad, un microcontrolador de un solo chip (STC9051 SCM) con módulo de transmisión inalámbrica. Esta arquitectura podría ayudar a alertar a los agricultores a través de mensajes cortos sobre el estado de los cultivos y también podría proporcionar información en tiempo real sobre los cultivos [19].

En 2014, Kaewmard y Saiyod [20] desarrollaron un sistema de medición portátil que incluye sensores para medir la humedad, la temperatura y la humedad del suelo con el objetivo de controlar el flujo de agua/sistema de riego de vegetales a través de Smartphones y red de sensores inalámbricos. Se desarrolló e instaló un sensor de flujo de agua subterránea en la raíz de las plantas vegetales para monitorizar la humedad del suelo.

Minh et al. [21] desarrollaron un prototipo ligero mediante la combinación de datos de sensores y módulos de comunicación para automatizar el entorno de un invernadero y lo llamó "Smart IoT Kit". El estudio se realizó sobre el crecimiento del maíz y los hongos ostra, y el prototipo podría ayudar a los agricultores a monitorizar los cultivos en cualquier momento y en cualquier lugar.

2.3.2 Agricultura Hidropónica

La palabra "hidropónico" se refiere a la condición en la que las plantas crecen en agua en lugar de suelo. La hidroponía es particularmente adecuada para las operaciones de invernadero. Una de las hortalizas hidropónicas del invernadero es la lechuga [26]. Hoy en día, el brócoli y los tomates también se cultivan hidropónicamente [27]. Sin embargo, cultivar estas plantas es muy difícil ya que son muy sensibles a los cambios en el pH del agua y los niveles de nutrientes. Por ello, se han estudiado arquitecturas IoT para automatizar el control de estas plantas en invernaderos [27].

Sarasvathi et al. [27] desarrollaron un mecanismo basado en IoT para controlar el pH del agua a través de una aplicación de smartphone que podría ayudar a monitorizar la condición actual de las plantas. Utilizaron una placa Raspberry Pi 3 conectada a los sensores de humedad, temperatura, pH y presión. Desarrollaron una aplicación Android para smartphones utilizando el lenguaje de programación Python, que puede ayudar al agricultor (usuario) a conocer el estado actual de las plantas. sobre el estado actual de los cultivos.

Ferentinos y Albright [28] desarrollaron un sistema de control GUI utilizando la red de predicción bayesiana. El sistema activa el control basado en la retroalimentación de los

sensores que incluyen datos de crecimiento de plantas, nivel de agua, pH, intensidad de luz y sensores de humedad ambiental.

Mehra et al. [29] propusieron un prototipo para el cultivo de la planta de tomate que integra sensores con una Raspberry Pi 3 y la placas Arduino con el objetivo de monitorizar parámetros ambientales en tiempo real. El prototipo también utiliza redes neuronales artificiales (ARN) para procesar múltiples parámetros (como la intensidad de la luz, la humedad, la temperatura, etc.) y guardar los datos en la nube en tiempo real. El modelo desarrollado mostró un desempeño satisfactorio (88% de precisión) y puede mejorarse considerando otros parámetros importantes.

Paulchamy et al. [30] han desarrollado una "caja hidropónica inteligente" habilitada para IoT que puede encender y apagar los actuadores automáticamente al detectar cambios en los datos del sensor. Esta arquitectura es capaz de reducir los niveles de CO₂ y controlar el flujo de agua.

Pitakphongmetha et al. [31] propusieron usar una red de sensores inalámbricos (WSN) para transmitir datos de sensores a la nube. Conectado a través de una aplicación de Android llamada "Blynk", monitorizaron con éxito el crecimiento de las plantas en hidroponía.

El beneficio principal de la hidroponía de invernadero basada en IoT es que elimina la necesidad de pesticidas innecesarios para el suelo.

El requerimiento de agua es menor que en invernaderos tradicionales. Además, el rendimiento de las plantas no se ve afectado por cambios ambientales externos [32]. El principal desafío de las granjas hidropónicas es que requieren una extensa supervisión manual para rociar el agua y proporcionar los nutrientes necesarios [29]. Además, la escalabilidad de las granjas hidropónicas es un gran problema, ya que las arquitecturas hidropónicas basadas en IoT requieren una gran inversión inicial para su configuración. La red de comunicación que se suele utilizar en las granjas hidropónicas no es adecuada para la comunicación a larga distancia [32].

2.3.3 Agricultura Vertical

La agricultura vertical es un cultivo vertical compuesto mayoritariamente integrado en invernaderos (ambiente controlado).

La agricultura vertical es ecológica y no requiere insecticidas.

Sin embargo, este tipo de operación requiere mucho mantenimiento para el cultivo de plantas. La agricultura vertical utiliza luces LED para equilibrar la luz solar. Como en otras operaciones de invernadero, es necesario controlar la temperatura, la humedad y la humedad del suelo [33].

Bhowmick et al. [33] propuso un prototipo IoT para la agricultura vertical. La arquitectura IoT propuesta incluye sensores para recopilar datos sobre temperatura, intensidad de luz, humedad y humedad del suelo, un módulo microcontrolador inalámbrico para transmisión de datos y una aplicación basada en web para observar los datos de los sensores. Los datos de los sensores se analizan a través de un módulo de radio (Intel Edison) con una placa Arduino conectada al sistema operativo Linux, Eclipse e Intel XDK. El servidor web utilizado para conectar los datos a la nube es ThingSpeak y para los dispositivos móviles se utilizó la aplicación de Android Virtuino para el envío de SMS.

Bin Ismail y Thamrin [34] desarrollaron un sistema de monitorización de cultivos basado en IoT para las hierbas *Typhonium flagelliforme*. La arquitectura incluye un sistema de encendido/apagado automático de la bomba de agua, sensores inalámbricos (un sensor de nivel de agua y tres sensores de humedad del suelo) y luces LED para la fotosíntesis. El sistema está interconectado a través de Ethernet, lo que facilita la comunicación.

El cultivo vertical en invernaderos inteligentes tiene muchas ventajas, una de ellas es que la productividad de los cultivos permanece igual incluso si ocurre un desastre natural (tifón, tormenta, huracán) [34]. Los cultivos verticales inteligentes pueden tener un sistema de riego automático y reducir los costos de mano de obra. Sin embargo, requieren una iluminación y brillo adecuados, lo que aumenta el consumo de energía. Esto también aumenta el capital de inversión. La agricultura vertical es relativamente nueva en la investigación.

2.4 Ventajas y desventajas de los invernaderos inteligentes

2.4.1 Ventajas

La implementación de invernaderos inteligentes puede traer múltiples beneficios potenciales. Algunas de estas ventajas pueden ser:

- *Uso más eficiente del agua:* Al poder monitorizar y controlar el riego de las plantas de forma más precisa y automática, se puede reducir el consumo de agua en el invernadero, lo que no solo reduce los costes operativos, sino que contribuye a una gestión más responsable de los recursos hídricos.

- *Mejora de la calidad y el rendimiento de los cultivos:* monitorizar y controlar la humedad y la temperatura en un invernadero puede ayudar a crear condiciones óptimas para el crecimiento y desarrollo de las plantas, lo que a su vez puede mejorar el rendimiento de los cultivos, produciendo una cosecha de calidad.

- *Ahorro de energía:* El uso de paneles solares reduce la dependencia de las fuentes de energía convencionales, lo que puede ayudar a reducir las emisiones y ahorrar energía.

- *Control remoto*: la capacidad de controlar el riego y el seguimiento del invernadero desde cualquier parte del mundo puede ser beneficiosa para los agricultores que no pueden estar en el invernadero todo el tiempo, ya que les permite realizar ajustes en tiempo real y garantizar que se respeten las condiciones óptimas.

- *Innovación tecnológica*: El despliegue de invernaderos inteligentes y el uso de energía solar pueden verse como formas innovadoras y sostenibles de ayudar a los agricultores a mejorar su imagen y marcar la diferencia en el mercado.

2.4.2 Desventajas

Aunque los invernaderos inteligentes pueden tener ventajas, también pueden tener algunas desventajas a considerar.

Una desventaja potencial es el costo inicial de implementar el sistema. La tecnología utilizada para monitorizar y controlar los invernaderos, así como la combinación de paneles solares y otros dispositivos de energía renovable, puede resultar costosa al principio. Sin embargo, el uso eficiente de la energía y el control preciso de las condiciones de cultivo pueden reducir los costos de funcionamiento y aumentar las ganancias del invernadero a largo plazo.

Otro inconveniente potencial es la complejidad del sistema. La implementación de un sistema de invernadero dotado de tecnología puede requerir un conocimiento especializado de tecnología y agricultura, lo que puede dificultar que algunos agricultores adopten el sistema. Además, la necesidad de mantenimiento y reparación de equipos y sensores puede requerir capacitación y tiempo adicionales para los trabajadores del invernadero.

Finalmente, es importante considerar la fiabilidad del sistema en caso de falla técnica o falla de energía. Si el sistema de control automático falla, existe el riesgo de daños y pérdidas de cultivos. Es importante contar con un sistema de respaldo o un plan de contingencia para mitigar estos riesgos.

Aunque implementar un invernadero con tecnología IoT puede tener desventajas, las ventajas superan los inconvenientes con creces. Un análisis cuidadoso óptimo-económico y un enfoque planificado para la implementación y el mantenimiento del sistema pueden mejorar significativamente la eficiencia, la rentabilidad y la sostenibilidad del cultivo en invernadero.

2.5 Conclusión y futuras perspectivas

Como se discutió en esta sección, mientras que el cultivo inteligente en invernaderos ofrece una variedad de beneficios actuales y potenciales, la tecnología también ha

evolucionado enormemente en los últimos años. Los principales desafíos para la agricultura inteligente con IoT se resumen a continuación:

- Solución confiable y rentable para monitorizar el crecimiento de la planta, incluidos todos los rasgos de la planta (altura, mortalidad de hojas, fruta/hueso/tamaño de hoja, etc.) (proteína, azúcar, etc.).
- Sensor confiable y rentable para monitorizar el clima local alrededor de un solo cultivo o un cultivo grande.
- Sensor económico y confiable para monitorizar las condiciones del suelo (composición química).
- Protocolo unificado de transferencia de datos. Se deben considerar la interoperabilidad y los estándares comunes para la anotación de datos, la visualización y la toma de decisiones. Esto requiere coordinación entre las asociaciones de la industria.
- Transmisión y almacenamiento seguro de datos. La integración de la tecnología blockchain en invernaderos inteligentes impulsados por IoT ofrece una posible solución para esto.

3. Mi trabajo

3.1 Introducción

El invernadero inteligente que se presenta en este trabajo consta de varios sensores, actuadores y controladores que trabajan juntos para mantener las condiciones ideales de temperatura y humedad para el crecimiento de las plantas. Los sensores miden la temperatura y la humedad tanto del aire como del suelo, mientras que los actuadores controlan el riego y la ventilación del invernadero. El sistema de control se encarga de procesar los datos recibidos de los sensores y, dependiendo de las condiciones ambientales actuales, activa los actuadores para mantener las condiciones óptimas de crecimiento.

Para habilitar la monitorización y control remoto del invernadero, se utilizan una conexión a Internet y una plataforma de software, lo que permite a los usuarios acceder a los datos de los sensores y controlar los actuadores desde cualquier parte del mundo. Esto significa que los usuarios pueden monitorizar y ajustar las condiciones ambientales de sus cultivos en tiempo real, aumentando la eficiencia de la producción y reduciendo el desperdicio de agua y energía.

Además de la tecnología IoT, el invernadero también utiliza módulos funcionales alimentados por energía solar para reducir la dependencia de las fuentes de energía convencionales. Estos módulos convierten la energía solar en electricidad para alimentar sensores y otros dispositivos electrónicos. De esta forma, se reduce el consumo de energía y se reduce la huella de carbono asociada a la producción de cultivos.

Todo esto se implementa en una estructura de invernadero con una maceta, una planta y un sistema de riego por goteo. Esta información se detalla en la siguiente sección.

3.2 Componentes

Para implementar un invernadero inteligente de manera efectiva, es esencial la combinación correcta de componentes y accesorios de hardware y software que aseguren tanto su funcionalidad como su estética. En cuanto a los componentes de hardware, se requiere la integración de sensores, actuadores, sistemas de riego, controles y sistemas de energías renovables. Estos componentes juegan un papel crucial en la recopilación de datos sobre las condiciones ambientales y la automatización de procesos en el invernadero.

Los sensores se utilizan para medir varios parámetros ambientales como la temperatura, la humedad y la humedad del suelo. Estos dispositivos, ya sean analógicos o digitales,

están conectados a microcontroladores o placas de desarrollo que se encargan de leer y procesar los datos obtenidos. Los sensores son esenciales para monitorizar y regular automáticamente las condiciones ambientales en el invernadero.

Por otro lado, los actuadores son dispositivos que permiten el control de diversos procesos en el invernadero, como la ventilación, la iluminación y el riego. Estos componentes, que pueden incluir motores, bombas y válvulas de solenoide, están controlados por señales eléctricas. El uso de actuadores es necesario para automatizar y controlar eficientemente los procesos en el invernadero, lo que se traduce en un importante ahorro de tiempo y energía.

Asimismo, los sistemas de riego son fundamentales para el abastecimiento de agua controlado y eficiente de las plantas. Estos sistemas manuales o automatizados incluyen bombas, tuberías, mangueras y rociadores. Contar con sistemas de riego adecuados es vital para mantener un entorno óptimo para el crecimiento de las plantas, evitando así la pérdida innecesaria de agua y otros recursos.

Se requieren controladores apropiados para administrar de manera efectiva los datos recopilados por los sensores y enviar señales precisas a los actuadores. Estos dispositivos, como microcontroladores, placas de desarrollo o controladores lógicos programables (PLC), son los encargados de recibir, procesar y transmitir los datos necesarios para controlar los procesos en el invernadero. En este caso específico, solo se utilizaron microcontroladores. El uso de controladores es fundamental para automatizar procesos y construir un sistema inteligente.

Por otro lado, la incorporación de sistemas de energías renovables como los paneles solares puede ser beneficiosa para alimentar los componentes del invernadero y reducir la dependencia de la red eléctrica tradicional. Estos sistemas contribuyen a una gestión más eficiente y sostenible de los recursos energéticos utilizados en el funcionamiento del invernadero.

Los componentes de software son esenciales para procesar y analizar los datos recopilados por los sensores y actuadores y para controlar de manera eficiente los sistemas. Estos componentes incluyen el sistema operativo, la programación, las bases de datos, las interfaces de usuario y las comunicaciones utilizadas.

El sistema operativo representa el núcleo del software utilizado y puede proporcionar una interfaz para controlar los diversos componentes de hardware y ejecutar los programas necesarios para monitorizar y controlar el invernadero.

La base de datos es esencial para almacenar los datos recopilados por los sensores y otros dispositivos en el invernadero. Una base de datos adecuada facilita el análisis de estos datos y ayuda a tomar decisiones informadas en el manejo del invernadero.

También se requieren interfaces de usuario que permitan la interacción con el invernadero. Estas interfaces pueden adoptar la forma de aplicaciones móviles o aplicaciones web y permiten visualizar y controlar los distintos procesos del invernadero desde cualquier parte del mundo.

Finalmente, la comunicación juega un papel esencial en la transmisión de los datos recopilados y la recepción de señales para controlar los actuadores en el invernadero. Esta comunicación puede realizarse a través de varios protocolos, como Wi-Fi, Bluetooth, Zigbee, LoRa, entre otros.

3.2.1 Hardware

A continuación, se detallarán todos los componentes de hardware utilizados en este trabajo.

i. **Raspberry Pi 3 Model B+**

Una Raspberry Pi es una pequeña computadora de placa única (SBC) desarrollada por la Fundación Raspberry Pi. Es una de las opciones más populares en el mercado de SBC debido a su bajo costo, tamaño compacto y facilidad de uso.



Figura 3.1 Raspberry Pi

Raspberry Pi se basa en un sistema en chip (SoC) que incluye un procesador, RAM y controladores integrados de entrada/salida (E/S). La mayoría de los modelos de Raspberry Pi tienen un procesador ARM de 32 o 64 bits que ofrece un buen rendimiento y un bajo consumo de energía.

El modelo Raspberry Pi 3 B+ es una versión mejorada de su antecesor, el modelo 3 B. Cuenta con un procesador de cuatro núcleos de 64 bits a 1,4 GHz, 1 GB de RAM, Bluetooth 4.2 y WiFi 802.11ac. También tiene un puerto Gigabit Ethernet, cuatro puertos USB 2.0, un puerto HDMI y una ranura para tarjetas microSD.

El dispositivo también tiene conectores GPIO (General Purpose Input/Output) que permite la conexión de varios componentes electrónicos como sensores, LED y motores. Esto lo convierte en una excelente opción para proyectos de electrónica y robótica.

Además, puede ejecutar una variedad de sistemas operativos, incluido Raspbian, una versión personalizada de Debian Linux optimizada para Raspberry Pi. También hay versiones de otros sistemas operativos populares como Ubuntu, Windows 10 IoT Core y Android.

La Raspberry Pi 3 B+ es ideal para proyectos que requieren un pc pequeño y económico con funciones de conectividad inalámbrica y puertos USB para periféricos. Además, su bajo consumo de energía lo hace ideal para proyectos portátiles o de energía solar.

Algunos ejemplos de proyectos que se pueden realizar con una Raspberry Pi 3 B+ son:

- Servidor web o una nube personal para almacenar y compartir archivos.
- Centro multimedia para reproducir películas, programas de televisión y música en casa.
- Sistema de seguridad.
- Robot o un vehículo autónomo con sensores y actuadores.
- Sistema de automatización para controlar luces, puertas y otros dispositivos.

[49]

En este trabajo se instaló Home Assistant como sistema operativo en la Raspberry Pi con el objetivo de monitorizar y controlar el invernadero. Home Assistant y sus funciones se explican en detalle en la sección <Software>.

ii. **Arduino nano IOT**

Arduino Nano IoT es una placa de desarrollo de hardware diseñada para crear proyectos IoT (Internet de las cosas) utilizando la plataforma Arduino. La placa Arduino Nano IoT combina el microcontrolador Atmel SAMD21G18A de 32 bits con el módulo WiFi ESP32 de Espressif, lo que permite que la placa se conecte a Internet e interactúe con otros dispositivos a través de la red.

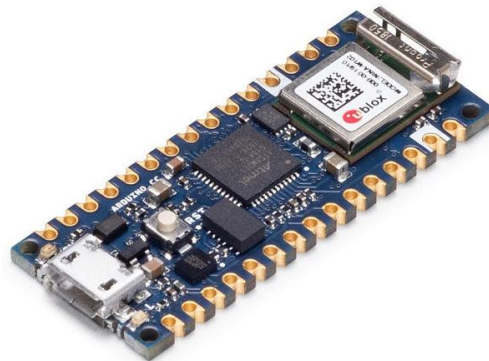


Figura 3.2 Arduino IoT

La placa Arduino Nano IoT es compatible con la mayoría de *shields* de Arduino, lo que significa que se pueden agregar módulos o sensores adicionales para ampliar las capacidades de la placa. Además, la placa nano IoT tiene una amplia cantidad de entradas y salidas digitales y analógicas, lo que la hace extremadamente versátil para una amplia gama de aplicaciones.

La placa Arduino Nano IoT generalmente se programa utilizando el entorno de desarrollo integrado (IDE) de Arduino, una plataforma de software de código abierto para programar placas Arduino. El IDE de Arduino es fácil de usar y tiene una gran comunidad de usuarios que ofrece soporte y tutoriales en línea. Sin embargo, en este trabajo se programó todo en Visual Studio Code usando C++, con el objetivo de tener todos los códigos bien organizados y localizados.

La placa Arduino Nano IoT es ideal para proyectos IoT que requieren conectividad Wi-Fi, ya que el módulo ESP32 integrado permite que la placa se conecte a redes Wi-Fi y se comunique con otros dispositivos en la red. Los ejemplos de proyectos que se pueden realizar con la placa Arduino Nano IoT incluyen sistemas de monitorización remoto, control de dispositivos domésticos inteligentes y seguimiento de objetos o personas.

En este trabajo se utilizó Arduino Nano IoT para integrar los siguientes sensores y actuadores:

- Sensor de temperatura y humedad del aire DHT11
- Sensor de temperatura y humedad del suelo I2C
- Relé
- Bomba de agua

Cada uno de ellos se detalla en los siguientes apartados.

iii. Sensores

Un sensor es un dispositivo diseñado para medir o detectar propiedades físicas o químicas del medio ambiente y convertir esa información en una señal eléctrica. Los sensores se utilizan en una gran variedad de aplicaciones, desde medir la temperatura y la humedad en una casa hasta detectar gases tóxicos en un entorno industrial.

Por lo general, un sensor mide una variable física como temperatura, presión, luz, sonido, humedad y otras, y luego convierte esa medida en una señal eléctrica que puede ser procesada por un sistema electrónico u ordenador. La mayoría de los sensores constan de dos partes: el elemento sensor y el circuito de acondicionamiento de señales.

El elemento sensor es la parte del sensor que capta la magnitud física a medir. Puede ser un termómetro, un fotodiodo, un micrófono o cualquier otro dispositivo que responda a la variable física que se está midiendo. El circuito de acondicionamiento de señal es la parte del sensor que convierte la señal eléctrica producida por el elemento sensor en una señal que puede ser procesada por un sistema electrónico.

En los últimos años, los sensores han sido fundamentales en el desarrollo de Internet de las cosas (IoT), ya que permiten que los dispositivos se comuniquen entre sí y recopilen información sobre el entorno que los rodea.

Los sensores utilizados en este trabajo se dividen en dos zonas: zona aire y zona tierra.

Zona de aire:

En la zona de aire hay un sensor que mide la humedad y la temperatura del aire. Este sensor es el DHT11.

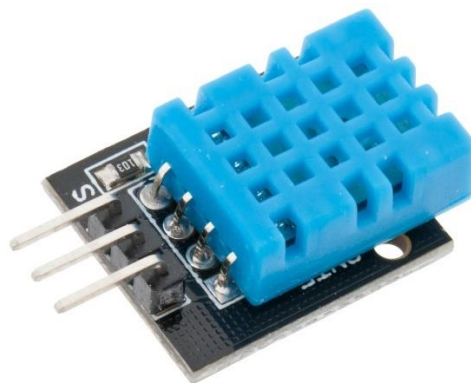


Figura 3.3 Sensor DHT11

El DHT11 es un sensor de temperatura y humedad que se comunica con un microcontrolador o dispositivo electrónico a través de un solo pin de datos. Es un sensor económico y fácil de usar.

El sensor tiene un rango de medición de temperatura de 0 °C a 50 °C y un rango de humedad de 20 % a 90 %. Utiliza un termistor para medir la temperatura y un sensor capacitivo para medir la humedad relativa ambiental.

El DHT11 también cuenta con una resistencia pull-up interna de 5k ohm, lo que significa que no es necesario agregar una resistencia externa a través de la conexión del pin de datos del sensor al microcontrolador o dispositivo electrónico.

Para utilizar el DHT11 en este proyecto, es necesario conectarlo al microcontrolador o dispositivo electrónico Arduino Nano IoT y programarlo para leer los datos de temperatura y humedad generados en el pin de datos del sensor.

Zona de tierra:

En la zona de tierra hay un sensor que mide la humedad y la temperatura del suelo, esta última no se procesa por sobrecarga del procesador. Este sensor fue desarrollado por CATNIP ELECTRONICS.



Figura 3.4 Sensor de humedad y temperatura del suelo

Este sensor utiliza tecnología capacitiva para medir la humedad del suelo y cuenta con una interfaz I2C para comunicarse con un microcontrolador o dispositivo electrónico, en este caso el Arduino Nano IoT.

El sensor de humedad del suelo capacitivo de CATNIP ELECTRONICS tiene una carcasa de plástico resistente al agua y una longitud de cable de aproximadamente 20 cm. El rango de medición de la humedad del suelo es de 0 a 100 % y tiene una precisión de ± 3 % a una temperatura de 25 °C. El sensor también tiene una compensación de temperatura incorporada para mejorar la precisión de las mediciones a diferentes temperaturas ambientales.

La interfaz I2C del sensor permite la comunicación con un microcontrolador o un dispositivo electrónico a través de dos líneas de datos: SDA (Serial Data) y SCL (Serial Clock). Esto permite una conexión rápida y fácil.

iv. Actuadores

Los actuadores son componentes que convierten una señal de control en una acción física. En otras palabras, son dispositivos que producen movimiento o energía en respuesta a una señal eléctrica, electrónica, hidráulica o neumática.

Los actuadores se utilizan en una gran variedad de aplicaciones que incluyen ingeniería mecánica, robótica, automatización industrial y sistemas de control. Ejemplos comunes de actuadores incluyen motores eléctricos, válvulas hidráulicas y neumáticas, solenoides, cilindros hidráulicos y neumáticos, entre otros.

Los actuadores se pueden usar para controlar la apertura y cierre de ventanas, regular el flujo de agua en el sistema de riego o incluso encender y apagar luces y ventiladores.

El tipo de actuador utilizado depende de la aplicación específica y los requisitos de control. A la hora de elegir el actuador adecuado para el proyecto, es importante considerar las características del actuador, tales como: B. su potencia, velocidad, precisión, capacidad de respuesta y durabilidad.

En este trabajo se utilizaron dos actuadores, una pequeña bomba de agua (5V) y un módulo relé.

Una bomba de agua (5V) es un dispositivo eléctrico que se utiliza para mover agua de un lugar a otro creando una presión positiva que fuerza el líquido a través de un sistema de tuberías. Estas bombas son pequeñas y funcionan con un suministro de 5 V, lo que las hace ideales para proyectos que requieren un tamaño compacto y un bajo consumo de energía.



Figura 3.5 Bomba de agua 5V

Estas bombas funcionan en su mayoría con corriente continua y están controladas por un circuito electrónico que ajusta la velocidad y la intensidad del flujo de agua. Existen diferentes tipos de bombas de agua, como bombas de diafragma, etc., cada una con características y aplicaciones específicas.

En este trabajo se ha utilizado la bomba de agua para controlar el suministro de agua en un sistema de riego en un invernadero.

Es importante señalar que estas bombas tienen una capacidad limitada y no son adecuadas para aplicaciones que requieran grandes volúmenes de agua o alta presión.

Para elegir la bomba adecuada para el proyecto, se tuvo en cuenta el caudal de agua requerido, la altura máxima requerida para bombear el agua, el diámetro de la tubería y otros factores relevantes.

El relé utilizado es un relé de 5V: es un dispositivo electromecánico utilizado para controlar el flujo de corriente eléctrica en un circuito. Consiste en un interruptor que se activa mediante una señal eléctrica de baja potencia, como un voltaje de 5 V, lo que permite que una corriente eléctrica de mayor potencia fluya a través del circuito.

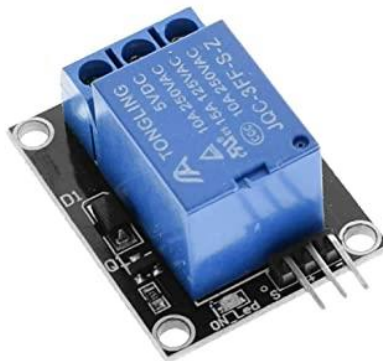


Figura 3.6 Relé 5V

Los relés de 5V se usan comúnmente en proyectos electrónicos que requieren la activación o desactivación de dispositivos de alta potencia como motores, luces o electrodomésticos. Por ejemplo, se podría usar un relé de 5 V para encender y apagar una luz a través de un microcontrolador.

Los relés de 5V son controlados por un circuito electrónico que activa una bobina en el relé, creando un campo magnético que mueve un interruptor, uniendo los contactos del relé. Hay diferentes tipos de relés, relés de acción rápida, relés de retardo de tiempo y relés de estado sólido, cada uno con funciones y aplicaciones específicas.

En este trabajo se utilizó el relé para controlar el encendido y apagado de un ventilador instalado en el invernadero.

v. **ESP 32**

El ESP32 es un microcontrolador de bajo consumo diseñado para aplicaciones de Internet de las cosas (IoT). Es un chip muy versátil y potente, que cuenta con un procesador de doble núcleo de 32 bits, mucha memoria flash y RAM, y una amplia gama de periféricos que incluyen Wi-Fi, Bluetooth y Ethernet.

El ESP32 puede conectarse a Internet y otros dispositivos a través de WiFi y Bluetooth, lo que lo hace ideal para aplicaciones IoT donde se requiere conectividad inalámbrica. Además, se puede programar usando una variedad de lenguajes de programación, incluidos C ++, Python y Lua.

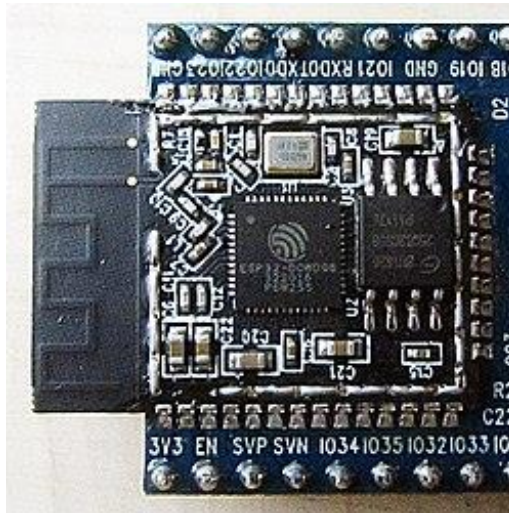


Figura 3.7 Placa de desarrollo ESP 32

Algunas de las aplicaciones comunes para el ESP32 incluyen domótica, automatización industrial, monitorización y control de dispositivos, y recopilación y análisis de datos en tiempo real. Su bajo consumo de energía también lo hace ideal para aplicaciones portátiles y alimentadas por batería.

vi. **Cámara TAPO C100**

La cámara Tapo C100 es una cámara de seguridad interior con resolución 1080p y tecnología de visión nocturna. Es capaz de capturar imágenes de alta calidad y resolución incluso en condiciones de poca luz y enviarlas a una aplicación móvil para verlas en tiempo real.

La cámara se conecta a la red Wi-Fi local y se puede controlar a través de una aplicación móvil dedicada. A través de la aplicación, el usuario puede ver la imagen en tiempo real, ajustar la configuración de la cámara y recibir notificaciones al detectar movimiento o sonido.



Figura 3.8 Cámara Tapo C100

En el contexto de este TFG, la cámara Tapo C100 se utiliza para obtener una imagen en tiempo real del interior del invernadero, permitiendo no depender de la aplicación comercial. Esto puede ser útil para monitorizar el estado de las plantas, detectar problemas o cambios en las condiciones ambientales, y tomar medidas oportunas para mantener el invernadero en condiciones óptimas.

vii. Batería 18650

Las baterías 18650 son baterías recargables de iones de litio que tienen forma cilíndrica y miden 18 mm de diámetro y 65 mm de longitud. Son ampliamente utilizadas en aplicaciones electrónicas, incluidos dispositivos portátiles como linternas, pc portátiles, drones y electrodomésticos.



Figura 3.9 Batería 18650

En el contexto de este trabajo, se utilizarán baterías 18650 para alimentar dispositivos electrónicos que requieran una fuente de alimentación portátil y de larga duración. Su alta capacidad de energía y capacidad de recarga los hacen ideales para aplicaciones que requieren una fuente de energía confiable a largo plazo.

En términos de integración de paneles solares, las baterías 18650 son compatibles con los sistemas de carga solar y se pueden usar junto con pequeños paneles solares para generar energía renovable. Esto es particularmente útil para aplicaciones portátiles o remotas donde no se dispone de una fuente de alimentación estable.

Los factores considerados al integrar baterías 18650 con paneles solares incluyen la capacidad de la batería, la eficiencia del panel solar y la calidad del controlador de carga solar. Al abordar estos factores, se puede crear un sistema de energía renovable portátil y de larga duración para alimentar una variedad de dispositivos electrónicos.

viii. Paneles solares

Los paneles solares son dispositivos que utilizan la energía solar para generar electricidad. Están formados por células solares que convierten la luz solar en electricidad a través de un proceso llamado efecto fotovoltaico.

Es importante tener en cuenta que la capacidad de un panel solar para generar electricidad depende de la cantidad de luz solar que reciba. Por lo tanto, el rendimiento del panel solar es mejor en áreas con luz solar directa y menos eficiente en áreas sombreadas o nubladas. Además, el rendimiento del panel solar también puede verse afectado por la temperatura ambiente y otros factores ambientales.

El funcionamiento de un módulo solar se basa en el efecto fotovoltaico, que es la capacidad que tienen ciertos materiales como el silicio para generar electricidad cuando se exponen a la luz solar.

Los paneles solares están hechos de celdas solares, pequeñas unidades que contienen materiales semiconductores, que tienen la capacidad de convertir la energía de la luz solar en energía eléctrica. Cuando los fotones de la luz solar golpean las células solares, liberan electrones que viajan a través del material semiconductor y generan electricidad.

Los paneles solares están diseñados para capturar la mayor cantidad de energía solar posible. Generalmente tienen una superficie plana y están posicionados de manera óptima para recibir la mayor cantidad posible de luz solar directa.



Figura 3.10 Panel Solar

En el contexto de este TFG, se han utilizado paneles solares pequeños de 7V y 250mA, que son ideales para aplicaciones portátiles. Se han utilizado para cargar baterías y alimentar dispositivos electrónicos.

ix. Módulo de carga

Se ha diseñado una placa de circuito impreso (PCB) que integra un sistema de carga y alimentación para el microcontrolador ESP32 (véase [Anexo II](#)). El propósito de la placa es cargar una batería 18650 usando dos paneles solares de 7V 250mA cada uno y usar la energía almacenada en la batería para alimentar el ESP32 durante el día y el microcontrolador durante la noche.

La placa consta de varios elementos, entre ellos un módulo de carga tp4056, que se encarga de cargar la batería, evitando la sobrecarga y protegiendo la batería de posibles daños. También tiene dos diodos Schottky que actúan como barreras y evitan que la corriente regrese a la batería, protegiendo el circuito y la batería. Además, tiene un boost-converter de 5 V que aumenta el voltaje de la batería de 3,7 V a 5 V para alimentar el ESP32.

La PCB diseñada para el proyecto fue creada en EasyEDA, una herramienta en línea que permite la creación de esquemas y diseños de circuitos electrónicos. Una vez diseñada la PCB, se procedió a la fabricación del circuito impreso mediante servicios de fabricación de PCB online, lo que nos permitió conseguir un resultado profesional y de gran calidad. Toda la información relacionada con el diseño de PCBs se encuentra en el [Anexo II](#).

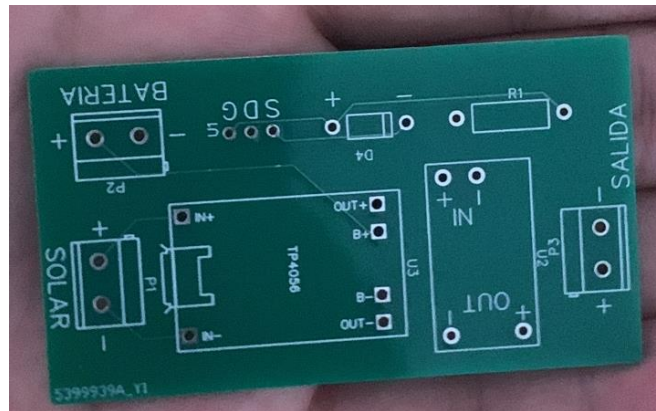


Figura 3.11 PCB diseñada para el control de carga

Posteriormente se adquirieron los componentes necesarios y se soldaron a la placa de circuito.

Después de montar los componentes en la PCB, se realizó una verificación de continuidad y correcta conexión de los componentes utilizando herramientas como un multímetro. Finalmente, se realizó una prueba funcional de todo el circuito para asegurar que la carga de la batería y el funcionamiento del ESP32 se realizan de manera correcta y eficiente.

x. **Alexa**

El Alexa Echo Dot es un altavoz inteligente creado por Amazon que incorpora Alexa, un asistente virtual activado por voz. Los usuarios pueden utilizar comandos de voz para interactuar con diversos servicios, gestionar electrodomésticos inteligentes, reproducir música, responder a consultas, proporcionar información y llevar a cabo diversas actividades.

El Echo Dot empieza a escuchar órdenes cuando un usuario dice la "palabra despertador" (normalmente "Alexa") para encenderlo. A continuación, actúa de acuerdo con esas órdenes.

Para analizar y comprender las instrucciones de voz, el Echo Dot utiliza el Alexa Voice Service (AVS), un servicio basado en la nube. El gadget envía la información vocal del usuario a la nube, donde se evalúa y transforma en instrucciones que se pueden seguir. El Echo Dot recibe las órdenes procesadas y las devuelve tras realizar las acciones deseadas o proporcionar los datos necesarios.

El asistente virtual, conocido como Alexa, es capaz de realizar una amplia gama de tareas, como reproducir música de distintos servicios de streaming, programar temporizadores y alarmas, ofrecer previsiones meteorológicas, responder a preguntas de cultura general, controlar dispositivos domésticos inteligentes

compatibles (como luces, termostatos, etc.), realizar pedidos en línea e incluso integrarse con “skills” de terceros para ampliar su funcionalidad.

A través de las actualizaciones de Amazon, las capacidades del Echo Dot se mejoran continuamente. Con el fin de mejorar la experiencia del usuario y mantenerse al día con las cambiantes tecnologías y servicios, se añaden con frecuencia nuevas funciones y capacidades.

En este trabajo, se ha implementado Alexa junto con una “Skill” desarrollada para realizar acciones sobre el invernadero mediante comandos por voz. Estas acciones incluyen: iniciar el riego, detenerlo, ventilar, conocer la temperatura del interior, etc.



Figura 3.12 Alexa Echo Dot (3 Gen.)

3.2.2 Software

Es considerado Software todo aquello que se debe programar, sin ser un dispositivo físico, y es necesario para el correcto funcionamiento del sistema.

i. Home Assistant

Home Assistant es un software de código abierto para automatizar y controlar dispositivos inteligentes. Este software es muy popular en el campo de la domótica y se puede utilizar para controlar diversos dispositivos como luces, interruptores, termostatos, cámaras de vigilancia, entre otros.



Figura 3.13 Logo Home Assistant

Home Assistant se ejecuta en un servidor local y se puede acceder a él a través de una interfaz web. Este software está diseñado para ser modular y altamente personalizable,

lo que significa que puede integrarse con diferentes dispositivos y servicios para crear una solución personalizada que se adapte a las necesidades de cada usuario.

Las características más notables de Home Assistant incluyen:

- *Soporte para una amplia gama de dispositivos:* Home Assistant es compatible con una amplia gama de dispositivos inteligentes, desde los más populares hasta los menos conocidos. Esto permite a los usuarios integrar todos sus dispositivos en una sola plataforma para un control centralizado.
- *Automatización:* Con Home Assistant se puede crear automatizaciones complejas basadas en eventos, condiciones y acciones. Esto permite que los dispositivos se comuniquen entre sí y realicen acciones de forma autónoma, lo que puede mejorar la eficiencia energética y reducir los costos operativos.
- *Integración con servicios web:* Home Assistant se integra con varios servicios web, como servicios meteorológicos, servicios de seguimiento de paquetes, servicios de noticias y otros. Esto permite a los usuarios obtener información útil y tomar decisiones basadas en ellos.
- *Personalización:* Home Assistant es altamente personalizable y se puede adaptar a las necesidades de cada usuario. Esto se consigue gracias a la arquitectura modular, que permite la integración de diferentes componentes y la creación de nuevas funcionalidades.
- *Soporte Multiplataforma:* Home Assistant es compatible con varias plataformas, como Linux, Windows, macOS, Docker, Raspberry Pi, entre otras, lo que facilita su instalación y uso en diferentes sistemas. En este trabajo se utilizó una Raspberry Pi 3 como se menciona en la sección de hardware.

Home Assistant se puede instalar en una gran variedad de dispositivos de hardware, incluidos los pc de escritorio, portátiles, Raspberry Pi, dispositivos NAS y otros dispositivos compatibles con Linux. La elección del hardware depende de las necesidades específicas del usuario, como el tamaño, la complejidad del sistema, la cantidad de dispositivos a integrar y el presupuesto.

Uno de los dispositivos más populares para instalar Home Assistant es la Raspberry Pi, que son dispositivos económicos que ocupan poco espacio y son ideales para proyectos de automatización pequeños. También se pueden utilizar dispositivos más potentes como Intel NUC, una mini PC, permitiendo manejar sistemas de automatización más complejos y exigentes.

Además, Home Assistant también se puede instalar en contenedores Docker como se describe a continuación.

En cuanto a las diferentes instalaciones posibles de Home Assistant, existen cuatro opciones principales: Supervised, Contenedor, Core y OS. Cada uno de ellos tiene sus propias ventajas y desventajas y es importante entenderlas para elegir la que mejor se adapte al proyecto.

- Supervised: esta opción es adecuada para usuarios que desean tener más control sobre la instalación de Home Assistant y desean conservar la flexibilidad de un sistema operativo completo. La instalación supervisada de Home Assistant se realiza en un sistema operativo basado en Linux como Ubuntu o Debian y permite la instalación de complementos y aplicaciones adicionales a través de la tienda de complementos de Home Assistant. Sin embargo, se requieren algunos conocimientos técnicos para esta instalación.

- Contenedor: esta opción es adecuada para usuarios que desean una instalación más rápida y sencilla de Home Assistant. En este caso, Home Assistant se ejecuta en un contenedor Docker. Introducir Home Assistant en un contenedor Docker permite una mayor portabilidad y flexibilidad.

- Core: esta opción es adecuada para usuarios que desean una instalación más simple y fácil de Home Assistant. Home Assistant Core es la versión "básica" de Home Assistant, sin complementos ni aplicaciones adicionales. Más rápido y fácil de instalar que la instalación supervisada o en contenedores.

- Sistema operativo: esta opción es adecuada para usuarios que desean una instalación completa y personalizada de Home Assistant. La instalación del sistema operativo Home Assistant es similar a la instalación de un sistema operativo completo y se realiza desde una imagen ISO.

	<i>OS</i>	<i>Container</i>	<i>Core</i>	<i>Supervised</i>
<i>Automations</i>	✓	✓	✓	✓
<i>Dashboards</i>	✓	✓	✓	✓
<i>Integrations</i>	✓	✓	✓	✓
<i>Blueprints</i>	✓	✓	✓	✓
<i>Uses container</i>	✓	✓	✗	✓
<i>Supervisor</i>	✓	✗	✗	✓
<i>Add-ons</i>	✓	✗	✗	✓
<i>Backups</i>	✓	✓	✓	✓
<i>Managed OS</i>	✓	✗	✗	✗

Tabla 3.1 Comparativa de métodos de instalación de Home Assistant

Se observa en la tabla anterior las características que ofrecen los diferentes métodos de instalación de Home Assistant. A continuación, se detallará y explicará las características más relevantes.

En términos de automatizaciones, Home Assistant ofrece la capacidad de automatizar acciones y tareas en función de varios eventos o condiciones. Estas automatizaciones se definen en el archivo de configuración de Home Assistant y se pueden activar o desactivar según se desee.

Una automatización se define por tres elementos principales: el desencadenante, la condición y la acción. El desencadenante es el evento que activa la automatización, como un sensor que cambia de estado. La condición es un requisito previo que debe cumplirse para que se ejecute la automatización, por ejemplo, que sea de noche o que un sensor determinado tenga un valor determinado. La acción es la tarea realizada en respuesta al desencadenante y la condición, por ejemplo, encender una luz o enviar una notificación.

Hay diferentes tipos de desencadenantes que se pueden usar para definir automatizaciones en Home Assistant. Las condiciones pueden ser tan simples o complejas como se desee, abarcando múltiples condiciones en una sola automatización.

Home Assistant también ofrece una serie de herramientas de automatización integradas, como la capacidad de configurar temporizadores. Además, los usuarios pueden crear y compartir sus propias automatizaciones a través de la comunidad de Home Assistant mediante el uso de *Blueprints*, los cuales se explicarán más adelante.

Crear automatizaciones en Home Assistant puede ser un poco complejo, pero la documentación oficial es muy completa con muchos ejemplos y guías. Una vez que se crean las automatizaciones, Home Assistant ofrece una vista integral de todas las automatizaciones activas en la interfaz, lo que permite a los usuarios monitorizar y ajustar fácilmente sus automatizaciones según sea necesario.

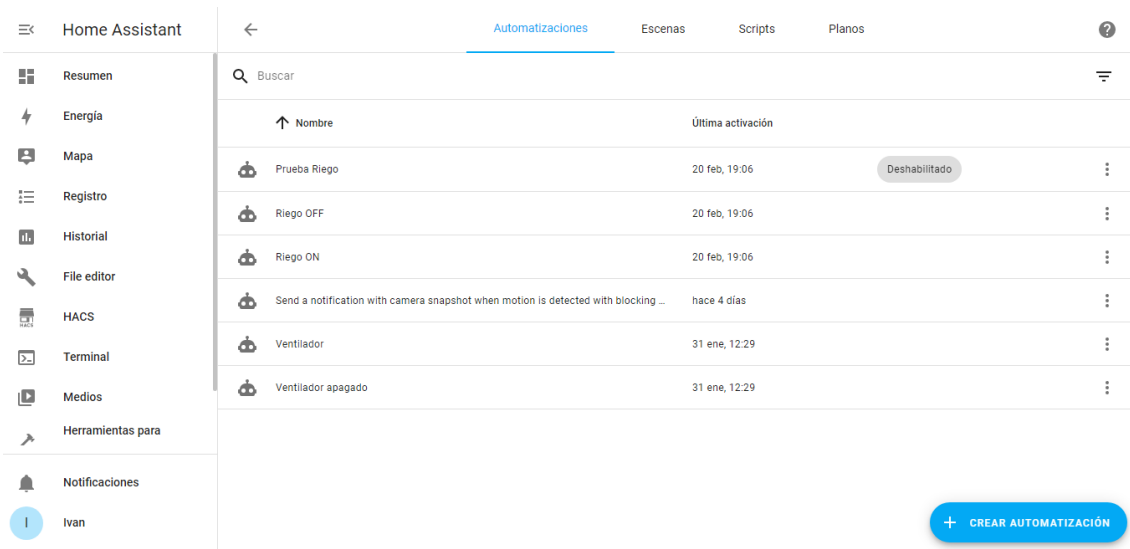


Figura 3.14 Visualización de automatizaciones

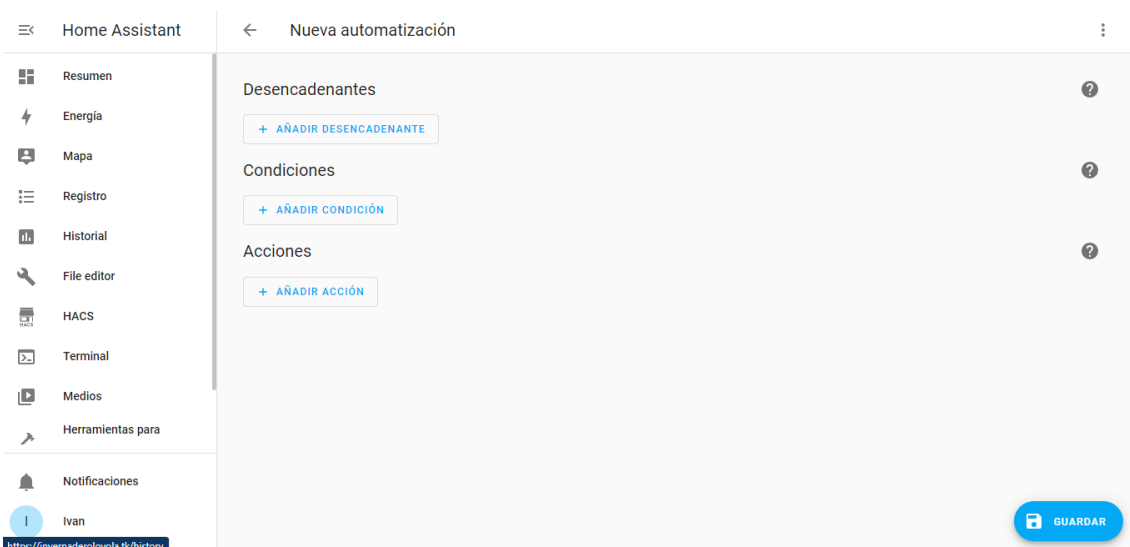


Figura 3.15 Creación de una automatización

Los “Blueprints” están disponibles junto con las automatizaciones. En Home Assistant, los planos o “Blueprints” son automatizaciones o configuraciones prefabricadas que pueden compartirse, importarse y modificarse rápidamente. Sin una gran experiencia en programación, se pueden crear fácilmente automatizaciones e integraciones sofisticadas utilizando Blueprints.

El usuario puede importar un blueprint existente de la comunidad del Home Assistant o empezar desde cero y elegir uno para utilizarlo al crear una nueva automatización o integración. El usuario puede cambiar cualquier parámetro o configuración de un Blueprint determinado para adaptarlo a sus necesidades.

Los Blueprint en Home Assistant suelen estar escritos en YAML. El código especifica los campos de entrada del plano, las opciones y las salidas o acciones relacionadas que se llevarán a cabo cuando se active el plano.

Una de las ventajas de utilizar Blueprints es que facilitan a los usuarios compartir sus integraciones y automatizaciones con otros miembros de la comunidad de Home Assistant a través de los foros.

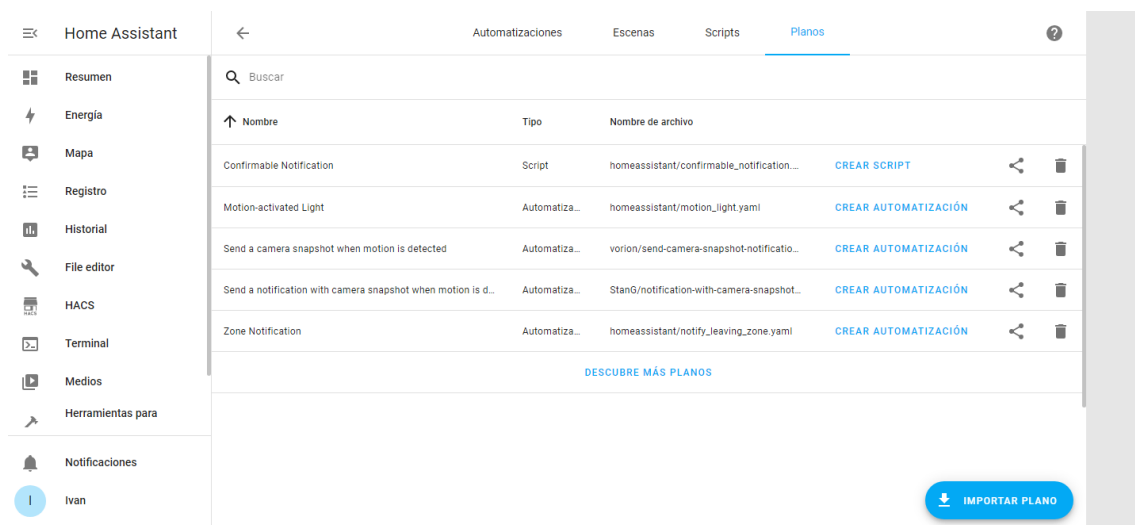


Figura 3.16 Blueprints

Los “dashboards” son un tipo de interfaz gráfica de usuario que puede personalizarse y que permite a los usuarios acceder fácilmente a los datos y dispositivos de Home Assistant. Mediante el editor de interfaz de usuario “Lovelace”, los usuarios pueden añadir y personalizar diferentes tarjetas, paneles y vistas para adaptarlos a sus necesidades.

Los temas, que permiten a los usuarios modificar los colores, fuentes y otros componentes visuales del cuadro de mandos, también pueden utilizarse para personalizar dashboard en Home Assistant, permitiendo a los usuarios adaptarlos a sus gustos individuales o mejorar el atractivo visual de su aplicación.

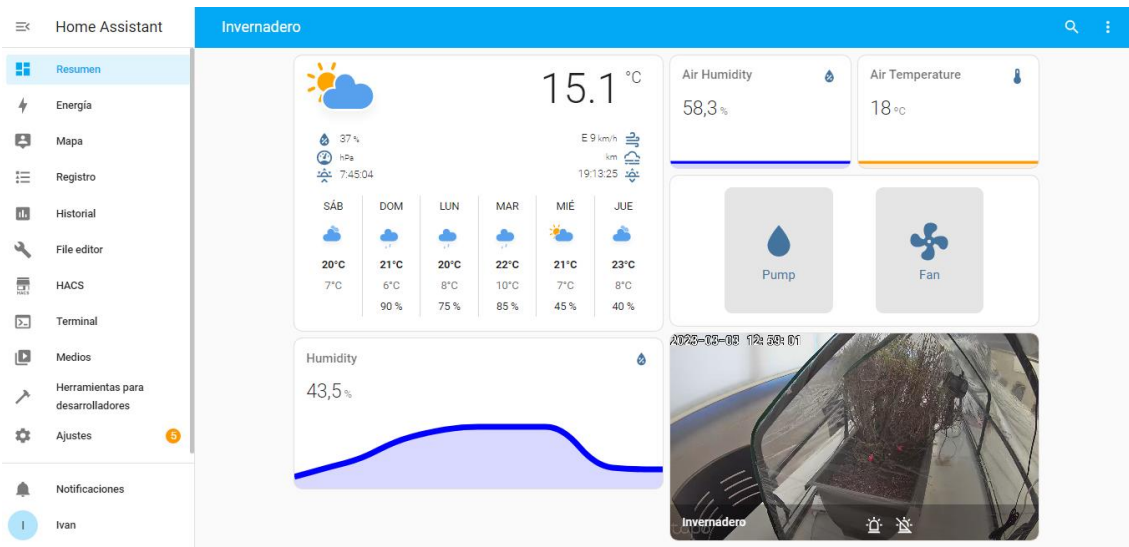


Figura 3.16 Dashboards

En cuanto a las integraciones de Home Assistant, son las formas en que diversos servicios y dispositivos se vinculan a la plataforma.

Hay más de 1600 integraciones disponibles para Home Assistant, desde plataformas multimedia y aplicaciones de terceros hasta dispositivos inteligentes como cámaras y termostatos. Google Assistant, Amazon Alexa, Philips Hue, Sonos y Nest son algunos ejemplos de integraciones muy utilizadas.

Al navegar por la configuración en la interfaz de usuario, Home Assistant presenta una sección para las integraciones. Todas las integraciones del sistema se muestran allí, donde se pueden añadir más según sea necesario.

Una vez configurada una integración, la interfaz de usuario de Home Assistant puede utilizarse para controlar y supervisar los dispositivos conectados. Dependiendo de la integración, se puede acceder a una amplia gama de funcionalidades, desde el simple control de dispositivos hasta sofisticadas estadísticas y presentaciones gráficas.

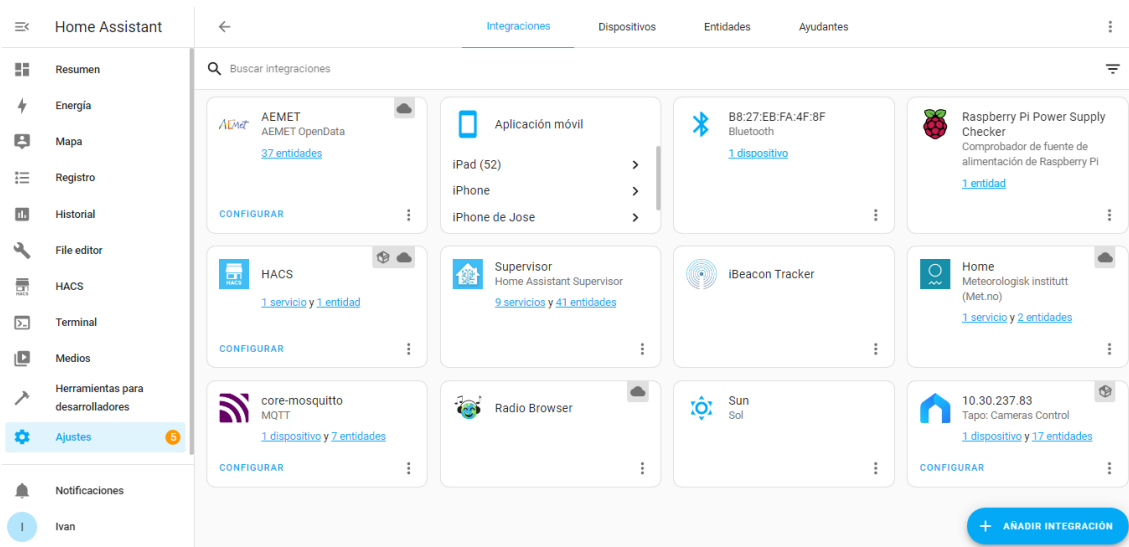


Figura 3.17 Integraciones

Los complementos para Home Assistant son programas preestablecidos que pueden instalarse y utilizarse directamente desde la interfaz de usuario. Estos complementos proporcionan al sistema más capacidades sin necesidad de que los usuarios tengan que configurar manualmente aplicaciones difíciles.

El siguiente apartado a considerar en Home Assistant es la tienda de complementos, que incluye una lista de “Add-on” preconstruidos que pueden instalarse con un solo clic, puede encontrarse en la sección de complementos.

Los siguientes son algunos de los complementos más populares que se ofrecen en la tienda de complementos Home Assistant:

- Mosquitto Broker: Este complemento permite a Home Assistant acceder a un broker MQTT para integrar dispositivos y sensores inteligentes.
- Node-RED: es una aplicación de programación basada en flujos que permite a los usuarios diseñar programaciones visuales para automatizaciones en Home Assistant.
- ESPHome: este programa permite a los usuarios diseñar rápidamente un firmware único para gadgets basados en ESP8266/ESP32, que posteriormente pueden conectarse a Home Assistant.
- AppDaemon: es un potente motor de automatización que permite a los usuarios crear automatizaciones sofisticadas utilizando Python.

A través de la interfaz en línea de Home Assistant, se pueden añadir y configurar complementos. Una vez instalados, los complementos pueden visualizarse y controlarse a través de la página de la tienda de complementos.

Cabe recalcar que un complemento en sí es un contenedor Docker, solo que esta versión de Home Assistant (Supervised), permite su gestión mediante un panel propio.

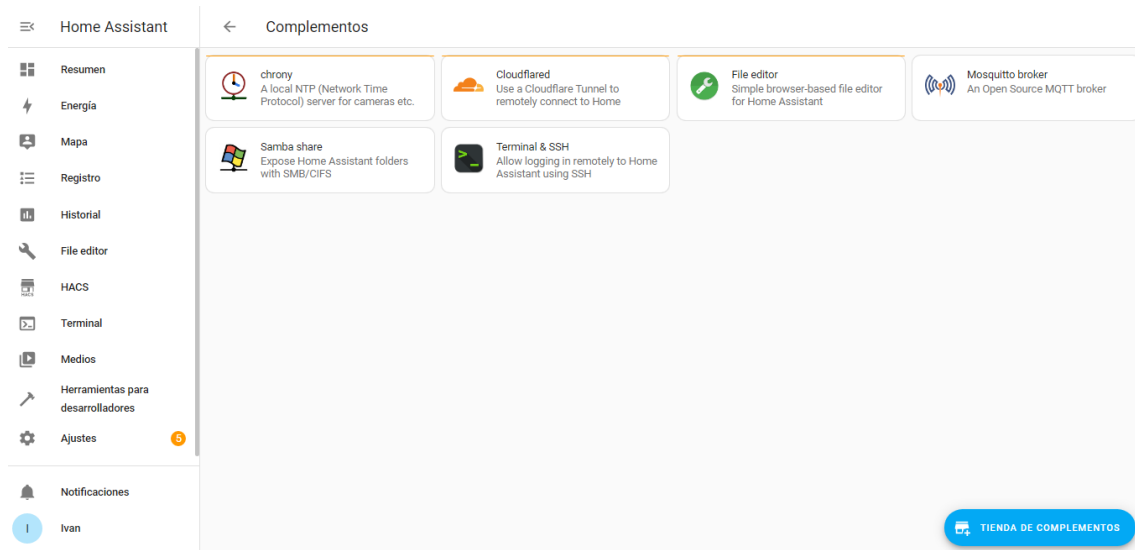


Figura 3.18 Complementos (Add-ons)

Después de cubrir todos los elementos que Home Assistant ofrece como base, llega HACS, una nueva función no incluida en la instalación de Home Assistant.

El complemento de Home Assistant llamado HACS (Home Assistant Community Store) ofrece una interfaz de usuario para instalar y actualizar rápidamente complementos e integraciones de terceros.

HACS proporciona una amplia selección de extras y componentes que no se incluyen en la instalación estándar de Home Assistant. Esto incluye temas personalizados, paneles de administración y sistemas de automatización complejos.

Instalar HACS en Home Assistant es muy sencillo. Una vez instalado el complemento, aparecerá un nuevo botón "HACS" en la barra lateral de Home Assistant. Al hacer clic en este botón, se abrirá la interfaz de usuario de HACS, que permitirá a los usuarios buscar y descargar complementos y componentes.

Cualquier usuario de Home Assistant que busque un método más sencillo y eficaz para gestionar complementos y componentes de terceros en su instalación encontrará en HACS una solución útil.

En este trabajo se ha utilizado HACS para instalar algunos complementos visuales y para integrar la cámara Tapo.

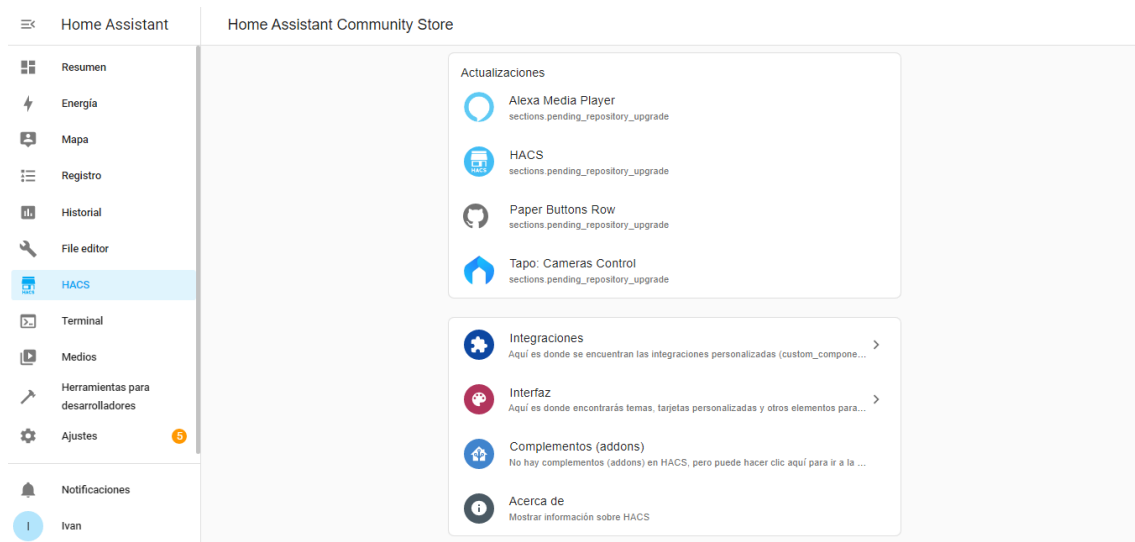


Figura 3.19 HACS

ii. LoRa

Para la transmisión de datos a larga distancia existe un protocolo de comunicación inalámbrica de bajo consumo denominado LoRa (Long Range). Su uso en canales de frecuencia sin licencia permite un despliegue económico. Las aplicaciones que requieren comunicaciones a larga distancia y un uso eficaz de batería son especialmente adecuadas para la tecnología LoRa.

“Chirp Spread Spectrum” (CSS) es el método fundamental de transmisión de datos utilizado por LoRa. Con la ayuda de esta técnica de modulación, los dispositivos LoRa pueden mantener un bajo consumo de energía a la vez que consiguen una conexión de largo alcance. LoRa puede lograr una comunicación estable incluso en situaciones con importantes interferencias y barreras dispersando la señal a través de una amplia banda de frecuencias.

La capacidad de largo alcance de LoRa es una de sus principales ventajas. Incluso en las ciudades, los dispositivos LoRa pueden alcanzar rangos de comunicación de varios kilómetros, dependiendo del entorno y de la implementación exacta. LoRa es, por tanto, excelente para la automatización industrial, las ciudades inteligentes, la agricultura de precisión y la vigilancia medioambiental.

El bajo consumo de LoRa es otro aspecto importante. Los dispositivos LoRa son apropiados para situaciones en las que la sustitución o recarga periódica de las baterías resulta prohibitiva, ya que están diseñados para funcionar con baterías durante largos

periodos de tiempo. Las redes de sensores a gran escala pueden desplegarse utilizando la tecnología LoRa, y pueden funcionar por sí solas durante meses o incluso años debido a sus bajos requisitos de energía.

Es importante recordar que LoRa tiene sus límites. En comparación con tecnologías como Wi-Fi o las redes móviles, compromete la velocidad de transmisión de datos y, sin embargo, proporciona una comunicación de largo alcance. En lugar de una comunicación de gran ancho de banda en tiempo real, LoRa está diseñada para aplicaciones de bajo ancho de banda que necesitan una entrega de datos intermitente.

En este trabajo, se ha utilizado LoRa para comunicar dos “Nodos” (se explicarán en el siguiente apartado) con el fin de transmitir información desde el exterior del invernadero.

iii. **WiFi**

En una red de área local (LAN), los dispositivos pueden conectarse a Internet y compartir datos de forma inalámbrica a través de WiFi, siglas de Wireless Fidelity (fidelidad inalámbrica). Utiliza radiofrecuencias para enviar y recibir datos y se basa en la familia de protocolos IEEE 802.11.

Sin necesidad de conexiones físicas, dispositivos como teléfonos inteligentes, ordenadores portátiles, tabletas y dispositivos del Internet de las Cosas (IoT) pueden conectarse a Internet e interactuar entre sí gracias a la tecnología WiFi. Ofrece transporte de datos a alta velocidad y se utiliza ampliamente en residencias, lugares de trabajo, zonas públicas y otras empresas.

Un punto de acceso inalámbrico (router) y los dispositivos inalámbricos son las dos partes principales de las redes WiFi. El punto de acceso sirve como nodo central que promueve la conexión de dispositivos y ofrece conectividad a Internet. Los paquetes de datos se envían entre los dispositivos y la red asociada a través de ondas de radio.

Las redes WiFi utilizan diferentes bandas de frecuencia, como 2,4 GHz y 5 GHz. Aunque la banda de 2,4 GHz tiene un mayor alcance, también puede ser más propensa a las interferencias de otros dispositivos que utilicen la misma banda de frecuencia. Aunque tiene un alcance menor, la banda de 5 GHz ofrece velocidades de datos más rápidas.

Para garantizar una transferencia de datos segura, el WiFi admite varios métodos de encriptación y seguridad, como WEP, WPA y WPA2. Estos protocolos permiten mantener la privacidad de los usuarios de la red y evitar accesos ilícitos.

Las altas velocidades de transmisión de datos de WiFi, que permiten un acceso rápido a Internet y una transmisión multimedia impecable, son una de las principales ventajas de

esta tecnología. varios dispositivos pueden unirse e interactuar simultáneamente dentro de la red gracias a su capacidad para varias conexiones.

En este trabajo, se ha utilizado WiFi para interconectar todos los dispositivos que miden o actúan sobre el invernadero. WiFi destaca por las altas velocidades de transmisión, por ello, se ha implementado una cámara de vídeo, para visualizar el estado del invernadero en tiempo real.

iv. MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligera para la comunicación eficaz entre dispositivos en redes restringidas. Se emplea con frecuencia en aplicaciones del Internet de las Cosas (IoT) y en otras circunstancias en las que son cruciales un ancho de banda y un consumo de energía reducidos.

Cuando se utiliza MQTT, los dispositivos pueden publicar mensajes en temas concretos y otros dispositivos pueden suscribirse a esos temas para recibir los mensajes publicados. Esto se conoce como patrón de mensajería publicar-suscribir. Este enfoque de comunicación desacoplada permite una conectividad de dispositivos escalable y asíncrona, así como una entrega de datos eficaz.

La arquitectura del protocolo es cliente-servidor, con un broker MQTT como intermediario principal. Para publicar o suscribirse a temas, los clientes (como dispositivos IoT o aplicaciones) se conectan al broker. Basándose en las suscripciones a temas de los clientes, el broker distribuye las comunicaciones que recibe de los editores a sus clientes suscriptores.

El reducido peso de MQTT es una de sus características importantes. Es apropiado para dispositivos con recursos limitados, como los que tienen poca capacidad de procesamiento, memoria o ancho de banda. Debido a su eficacia, MQTT puede utilizarse en situaciones en las que el acceso a la red puede ser errático o cuando hay dispositivos alimentados por batería.

Para la entrega de mensajes, MQTT ofrece tres niveles de calidad de servicio (QoS)

QoS 0 (máximo una vez): La comunicación sólo se envía una vez sin ningún tipo de confirmación o garantía.

QoS 1 (como mínimo una vez): La entrega del mensaje está asegurada, pero existe la posibilidad de duplicación.

QoS 2 (exactamente una vez): Mediante la aplicación de un protocolo handshake, el mensaje sólo se entregará exactamente una vez.

MQTT ha ganado popularidad en aplicaciones IoT por su sencillez, escalabilidad e interoperabilidad. Se utiliza en diversos ámbitos, como la domótica, la supervisión y el

control industrial, las ciudades inteligentes y los sistemas de telemetría. Su diseño ligero y su eficaz distribución de mensajes lo hacen idóneo para aplicaciones con un gran número de dispositivos y redes con poco ancho de banda.

En este trabajo, se ha utilizado el protocolo MQTT para el envío de datos entre los dispositivos integrados en el invernadero y el servidor de visualización y control. En el apartado de implementación se detallará su uso en este proyecto.

3.2.3 Accesorios

Al sistema diseñado se le han añadido, además de todo lo ya incorporado, una planta, una maceta, un pequeño invernadero y un sistema de riego por goteo, con el fin de dar al trabajo un enfoque más práctico y realista. La incorporación de estos componentes permite realizar pruebas precisas del sistema desarrollado, permitiendo evaluar su funcionamiento en un entorno controlado como el de un invernadero real.

Cabe destacar que el uso de estos accesorios no sólo hizo posible probar el sistema desarrollado de una manera más fiable y precisa, sino que también ayudó a aumentar la comprensión de los diversos factores que intervienen en el diseño, construcción y funcionamiento de los invernaderos, permitiendo una evaluación más exhaustiva de los resultados.

i. Estructura del invernadero

El invernadero pequeño adquirido tiene una estructura de acero y cubierta de PVC transparente, con unas dimensiones de 120x60x60 cm. Este tipo de invernaderos son ideales para proteger las plantas y cultivos de las perturbaciones meteorológicas y controlar la temperatura, humedad y luz que reciben. Además, su tamaño compacto lo hace ideal para su uso en espacios reducidos. El invernadero también cuenta con una puerta con cremallera que facilita el acceso y ventilación de la estructura.



Figura 3.20 Estructura del Invernadero utilizada

ii. Maceta

La maceta es un recipiente utilizado para cultivar plantas que está hecho para proporcionar a la planta suficiente espacio y nutrientes para crecer. La maceta rectangular utilizada en este proyecto está compuesta de un material robusto y duradero. La maceta contiene sustrato, una sustancia que se utiliza para mantener la humedad y los nutrientes del suelo cerca de las raíces de la planta. Para este proyecto se empleó como sustrato una combinación de tierra, arena y material orgánico, que ofrece los nutrientes necesarios para el desarrollo de la planta.

iii. Planta

Lavandula angustifolia, conocida como lavanda, es una planta aromática y decorativa autóctona de la región mediterránea que se emplea en este proyecto. Es una planta perenne de unos 60 cm de altura que se desarrolla en pequeños arbustos.

La lavanda es una planta resistente que requiere un mantenimiento mínimo y puede prosperar en diversas situaciones climáticas. Es conocida por su aroma calmante y sus beneficios terapéuticos, que incluyen la reducción de la ansiedad y la mejora del sueño.

La lavanda se ha utilizado como planta modelo para medir los efectos de la gestión medioambiental en el crecimiento y la salud de las plantas en el contexto del proyecto de invernadero. También se ha utilizado para examinar la eficacia de los sensores de humedad del suelo y del sistema de riego por goteo.

iv. Sistema de riego

El tipo de sistema de riego empleado en este proyecto es un pequeño sistema de riego por goteo. Este método utiliza una tubería de PVC para transportar el agua a través de un depósito de agua que está conectado a una bomba sumergible. Para regar las plantas se han instalado pequeños goteros que liberan el agua de forma gradual y controlada a lo largo de este conducto.

El sistema de riego es un componente esencial del proyecto, ya que permite mantener un nivel adecuado de humedad en el sustrato, lo que favorece el crecimiento y desarrollo de la planta.



Figura 3.21 Sistema de Riego

3.3 Implementación

Esta es la sección más importante de todo el trabajo de este invernadero, la cual implica el diseño y la construcción de hardware y software personalizados para el control y monitorización, así como la integración de los módulos solares en el sistema.

Para cumplir con el objetivo general del trabajo, se ha dividido esta sección en “Objetivos Específicos” (OE), que, en su conjunto, dan respuesta al problema planteado en este proyecto.

Cada OE presentará diferentes “Actividades”, que son las realizadas para cumplir con dicho OE.

OE.1. Framework IoT

Este objetivo específico consiste en definir y desplegar una estructura IoT.

Actividad 1.1. Identificación de sensores y sistema IoT

El objetivo de esta actividad es identificar los distintos sensores a utilizar y su funcionalidad.

En primer lugar, se define el término “Nodo”. Un nodo es el dispositivo encargado de capturar, procesar y comunicar los datos leídos por diferentes sensores, así como interactuar con los actuadores en caso de existir. En este trabajo se presentan cuatro nodos:

- **Nodo 1:** Compuesto por un sensor de humedad y temperatura de aire, un sensor de temperatura y humedad de la tierra, un actuador de riego, y un actuador de ventilación. Este nodo se comunica con el servidor mediante WiFi, a través del protocolo MQTT. El microcontrolador es un Arduino Nano IoT.

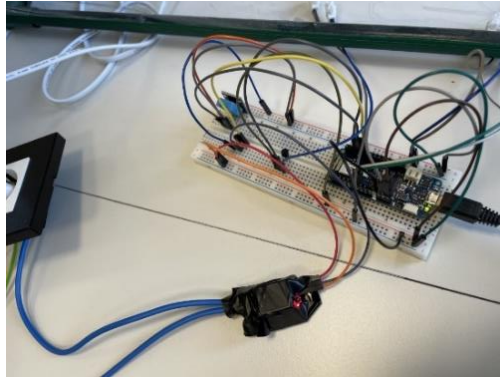


Figura 3.22 Nodo 1

- **Nodo 2:** Compuesto por un sensor óptico (cámara). Este sensor captura en forma de vídeo lo que ocurre en el invernadero. Este nodo se comunica con el servidor mediante WiFi, permitiendo una visualización en tiempo real de la cámara (streaming). Este nodo es comercial y corresponde a la cámara Tapo.



Figura 3.23 Nodo 2 (Cámara Tapo)

- **Nodo 3:** Este nodo no tiene ningún sensor físico. Su función es hacer de "Gateway". Recibirá la información de otro nodo mediante LoRa y la transmitirá al servidor mediante WiFi, usando el protocolo MQTT. El microcontrolador es un ESP32.



Figura 3.24 Nodo 3 (Gateway)

- **Nodo 4:** Compuesto por un sensor de humedad y temperatura del aire. Este nodo representa una estación de clima y envía los datos del exterior del invernadero al Nodo 3 mencionado anteriormente mediante LoRa. El microcontrolador es un ESP 32. Al no disponer de un sensor de este tipo, este nodo enviará exclusivamente el estado de la batería que utiliza como alimentación, es decir, su capacidad.

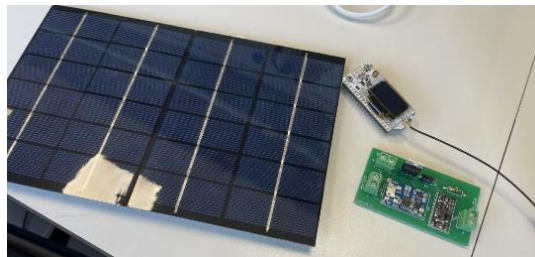


Figura 3.25 Nodo 4

Actividad 1.2. Frontend

El objetivo de esta actividad es instalar Home Assistant y presentar un panel de control para la visualización de los datos.

En primer lugar, se tratará la instalación de Home Assistant. Home Assistant permitirá agrupar todos los sensores y actuadores desplegados en el proyecto permitiendo una visualización “friendly” para el usuario.

Se ha elegido Home Assistant Supervised entre las cuatro opciones de instalación mencionadas en el apartado de Software. Uno de los motivos de la selección de esta instalación es la habilitación para poder instalar complementos o Add-Ons que se utilizarán más tarde.

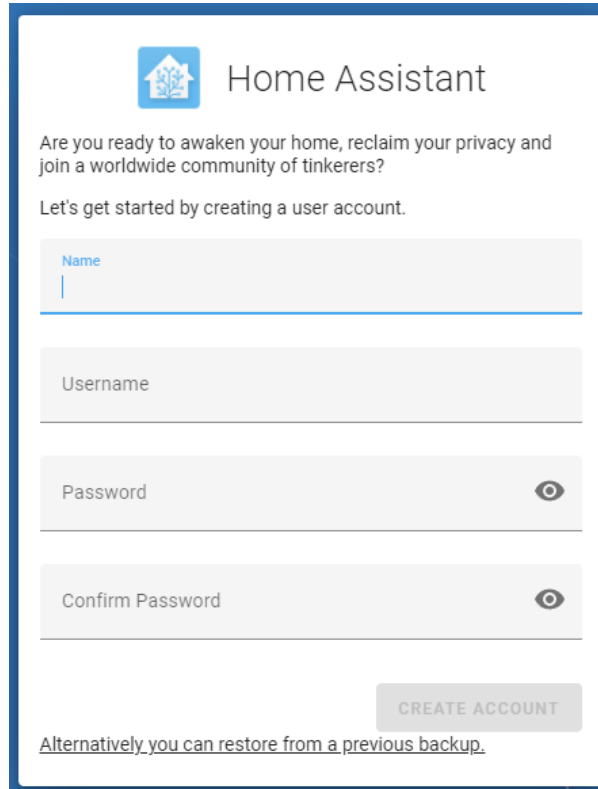
Cabe destacar que los Add-Ons o complementos son en sí contenedores Docker.

Para la instalación, se ha obtenido una Raspberry Pi 3 y una tarjeta microSD de al menos 64 GB de capacidad. A continuación, se ha descargado la imagen de Home Assistant desde su sitio web oficial. La imagen está disponible en varios formatos, pero para una Raspberry Pi 3 se debe descargar la imagen en formato "Raspberry Pi 3 Model B/B+". Se ha preparado la tarjeta microSD para poder instalar Home Assistant en la Raspberry Pi 3. Para ello, se ha utilizado un software especializado en grabar imágenes en tarjetas microSD, Balena Etcher.

Antes de insertar la tarjeta en la Raspberry Pi, se ha preparado un USB para realizar la configuración inicial del Wi-Fi ([Anexo I](#)).

Una vez que se ha preparado la tarjeta microSD con la imagen de Home Assistant, se inserta la en la Raspberry Pi 3 y se enciende. La Raspberry Pi 3 arrancará desde la tarjeta

microSD y se iniciará Home Assistant. Para acceder al panel de control, simplemente hay que dirigirse a la siguiente url: `http://<ip_raspberry>:8123` Una vez ahí, aparecerá una ventana como la siguiente figura:



The screenshot shows the Home Assistant initial configuration interface. At the top, there is the Home Assistant logo and the text "Home Assistant". Below this, a message asks if the user is ready to awaken their home and join a community of tinkerers. The main instruction is to create a user account. The form contains four input fields: "Name", "Username", "Password", and "Confirm Password". Each password field has an eye icon to toggle visibility. A "CREATE ACCOUNT" button is located at the bottom right of the form. Below the button, there is a link: "Alternatively you can restore from a previous backup."

Figura 3.26 Configuración inicial de Home Assistant

Una vez instalado e inicializado Home Assistant, se pueden configurar paneles de control (en Home Assistant conocidos como dashboards) como los de la siguiente figura:

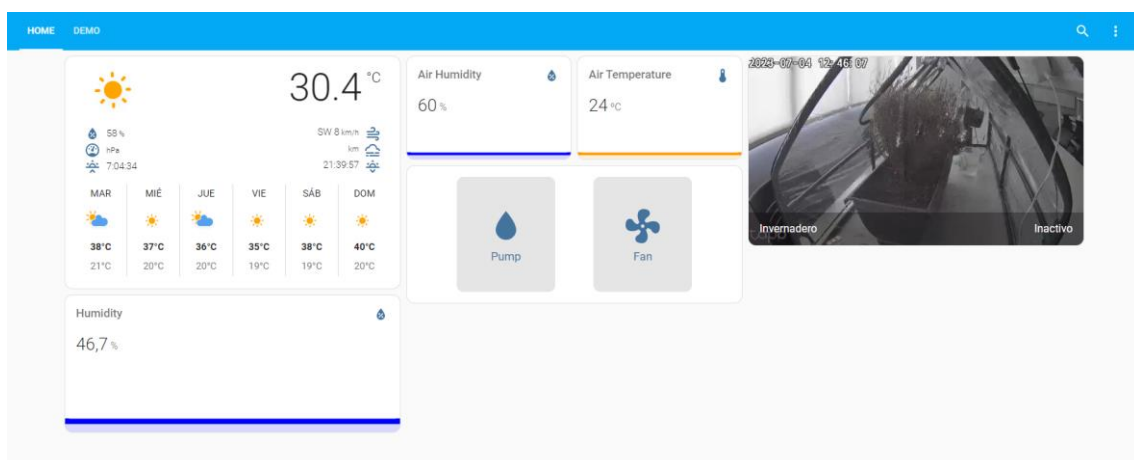


Figura 3.26 Dashboard v1 de Home Assistant

En posteriores objetivos específicos y actividades se detallarán las diferentes configuraciones realizadas para lograr un panel de control como el anterior.

Actividad 1.3. Integración

El objetivo de esta actividad es integrar y comunicar los distintos nodos mencionados en la Actividad 1. Esta actividad requiere de una configuración en el servidor.

Nodo 1 y 3: Para integrar estos nodos, se ha configurado en primer lugar un bróker MQTT. Este bróker es el encargado de capturar los mensajes enviados por el nodo. En Home Assistant, tal y como se ha explicado en la sección de Software, se puede integrar Mosquitto, un bróker MQTT. Simplemente se agrega un complemento. En el siguiente objetivo específico se detallará esta configuración.

Nodo 2: Para la integración del segundo nodo (cámara), simplemente se requiere de la integración de Tapo. Al navegar a las integraciones de Home Assistant, se puede añadir una llamada “Tapo Camera”.

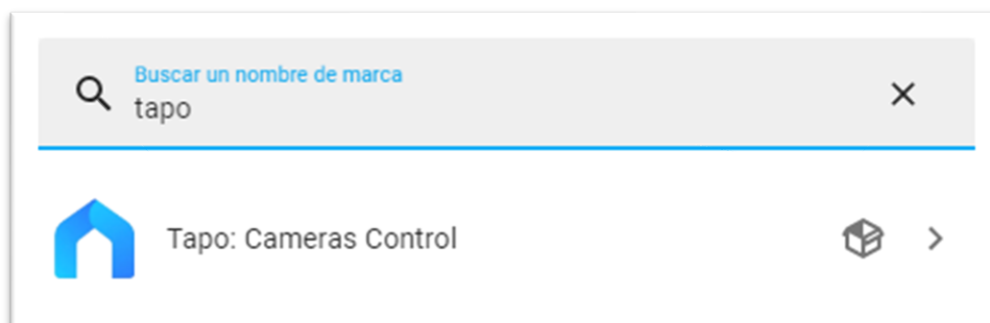


Figura 3.27 Integración Tapo

Al habilitarla, se puede configurar una cámara Tapo introduciendo su dirección IP y las credenciales definidas en la aplicación oficial de Tapo. Esta integración utiliza el protocolo RTSP para la comunicación de las cámaras.

Actividad 1.4. Acceso exterior

El objetivo de esta actividad es acceder desde cualquier parte del mundo al panel de control, y poder monitorizar y controlar el estado de las plantas desde cualquier lugar.

Tal y como se ha explicado en la Actividad 1.2, para acceder al panel de control es necesario dirigirse a una url que contiene la IP de la raspberry. Para ello, hay que estar en el mismo rango de IPs que el dispositivo.

La solución propuesta se denomina Tunnel, y se ha implantado utilizando Cloudflare, una empresa estadounidense que proporciona una red de entrega de contenido, servicios de seguridad de Internet y servicios de servidores de nombres de dominio distribuidos.

Cloudflare Tunnel es una herramienta proporcionada por Cloudflare que le permite establecer una conexión segura y cifrada entre sus servidores de origen y la red de Cloudflare. Proporciona una forma segura de acceder a aplicaciones y servicios que normalmente se ejecutan detrás de un cortafuegos o en una red privada.

El objetivo principal del Túnel de Cloudflare es proporcionar acceso a servicios que se ejecutan en una infraestructura de red privada o detrás de un cortafuegos sin exponer estos servicios directamente a Internet. Esto permite acceder a estos servidores privados de forma remota y segura.

El funcionamiento de Cloudflare Tunnel consiste en crear un "túnel" seguro que conecta un servidor de origen a la red de Cloudflare.

Una vez establecido el túnel, el tráfico se enruta a través de la red de Cloudflare y se aplican una serie de funciones de seguridad como el filtrado de contenido malicioso, la mitigación de ataques DDoS y la protección de la capa de aplicación.

Para establecer un túnel, es necesario adquirir un dominio, en el contexto de este trabajo, se ha obtenido un dominio con extensión gratuita: `invernaderoloyola.tk`, a través del proveedor de dominios Freenom.

Una vez adquirido el dominio, se debe crear una cuenta en Cloudflare y cambiar los DNS del dominio a los proporcionados por Cloudflare.

Gestión de DNS para **invernaderoloyola.tk** [Importar y exportar](#) [Configuración de la pantalla del Panel de control](#)

Todos los cambios realizados en el cajón de edición se implementan una vez guardados.

Tipo	Nombre	Contenido	Estado de proxy	TTL	Acciones
CNAME	invernaderoloyola.tk		Redirigido por proxy	Automático	Editar

Servidores de nombres de Cloudflare

Para usar Cloudflare, asegúrese de que se hayan cambiado los servidores DNS autoritativos o los servidores de nombre. Estos son los servidores de nombre asignados de Cloudflare.

Tipo	Valor
NS	xxxx.xx.cloudflare.com
NS	xxxx.xx.cloudflare.com

Figura 3.28 Gestión de DNS en Cloudflare

Cuando se vinculan los DNS, ya se puede configurar en Home Assistant, para ello, se instala un nuevo complemento llamado Cloudflared. Ahí, se configura el dominio adquirido. Una vez configurado, ya se puede acceder al panel de control desde: <https://invernaderoloyola.tk>.

OE.2. Sistema de comunicación asíncrona

Para una correcta comunicación entre Nodo y servidor, se ha hecho uso de MQTT. Para comunicarse con Home Assistant usando MQTT es necesario la configuración de un bróker.

Actividad 2.1. Bróker MQTT

El objetivo de esta actividad es instalar un bróker para permitir la comunicación MQTT entre los nodos y Home Assistant.

Un bróker es el servidor con el que se comunican los clientes: recibe comunicaciones de unos y se las envía a otros. Los clientes no se comunican directamente entre sí, sino que se conectan con el bróker. Cada cliente puede ser un editor, un suscriptor o ambos.

La instalación de un bróker en Home Assistant se hace a través de los complementos. En este caso, se ha utilizado Mosquitto como el bróker a utilizar.



Figura 3.29 Mosquitto Broker

Una vez instalado el complemento, aparecerá una nueva integración que permitirá la correcta configuración del bróker. En esta integración, se define el nombre del bróker, el usuario y la contraseña que será utilizada para comunicarse con él.

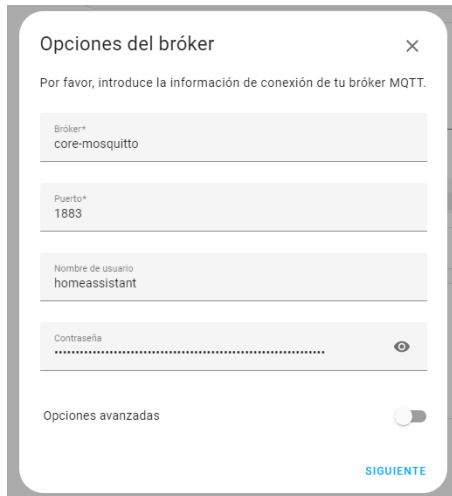


Figura 3.30 Configuración del bróker

Una vez configurado el bróker, se programa el nodo para, primero leer los datos de los sensores, y luego publicar esos datos en el topic correspondiente. Todos los códigos utilizados para programar los nodos se encuentran en el [Anexo III](#).

Además, el nodo puede estar suscrito a un topic y activar o desactivar los actuadores en función de lo que reciba en ese topic.



Figura 3.31 Ejemplos de lectura de Topics

Actividad 2.2. Topics

El objetivo de esta actividad es definir y explicar los diferentes topics a los que se suscriben y publican los clientes.

Un topic MQTT es un filtro que el broker utiliza en las entregas de mensajes MQTT. En pocas palabras, el broker filtra todos los clientes conectados según sus suscripciones y reenvía el mensaje a estos suscriptores (clientes). Tanto los editores como los suscriptores pueden crear topics MQTT.

En Home Assistant, se definen tres topics. El primero para recibir la configuración de un dispositivo, el segundo para recibir los datos de ese dispositivo, y el tercero para comunicarse con el dispositivo (si aplica).

- Topic 1: Este topic permite a Home Assistant definir un dispositivo o sensor en base al mensaje que captura. Este topic se define finalizando en /config. El mensaje se envía y recibe en formato json. En la siguiente figura se presenta un mensaje en un topic de configuración. En este caso, para la ventilación del invernadero.

```
Mensaje 6 recibido en homeassistant/fan/78a76dbd9e7c/ventilation/config a las 13:14:

{
  "name": "Fan",
  "uniq_id": "ventilation",
  "dev": {
    "ids": "78a76dbd9e7c",
    "name": "Arduino",
    "sw": "1.0.0"
  },
  "stat_t": "aha/78a76dbd9e7c/ventilation/stat_t",
  "cmd_t": "aha/78a76dbd9e7c/ventilation/cmd_t"
}

QoS: 0 - Retain: true
```

Figura 3.32 Topic de configuración

Como se puede observar, se definen varios parámetros, así como dos topics más.

- Topic 2: /stat_t. En este topic se envían los datos del sensor. En el caso de la ventilación sería un mensaje de estado (Bool), en el que se indica si el ventilador este encendido o no. En el caso de un sensor de humedad, sería un mensaje con un valor para la humedad.

```
Mensaje 12 recibido en aha/78a76dbd9e7c/SensorHumDHT/stat_t a las 13:14:  
  
60  
  
QoS: 0 - Retain: true
```

Figura 3.33 Topic de estado

- Topic 3: /cmd_t. En este topic, se envían comandos desde Home Assistant al nodo, en el caso del ventilador, permite encenderlo o apagarlo mediante la publicación de “ON” y “OFF” en este tópic

```
Mensaje 15 recibido en aha/78a76dbd9e7c/ventilation/cmd_t a las 13:22:  
  
ON  
  
QoS: 0 - Retain: false
```

Figura 3.34 Topic de comandos

OE.3. Comunicación por radiofrecuencia

Para comunicar los nodos 3 y 4 descritos en la Actividad 1.1, es necesario plantear una red LoRa.

Actividad 3.1. Emisor

El objetivo de esta actividad es configurar el emisor (Nodo 4), encargado de captar la humedad y temperatura del aire exterior, para enviar esta información al receptor (Nodo 3).

Para la integración del nodo 4, se ha programado el microcontrolador para leer los sensores y enviar los datos al nodo 3 a través de LoRa. El código se encuentra en el [Anexo III](#).

El nodo presenta un sensor DHT11 que mide estas dos variables. Mediante pines digitales, se puede leer el voltaje que proporciona el sensor y calibrarlo.

El nodo también presenta el chip lora SX1276/SX1278. Con el que se puede realizar la comunicación utilizando este protocolo.

Actividad 3.2. Receptor

El objetivo de esta actividad es configurar el receptor (Nodo 3), encargado de captar la información enviada por el emisor (Nodo 4). Además, una vez captada la información, se pretende enviarla a Home Assistant a través de MQTT. Por tanto, es conveniente expresar el nodo receptor como Gateway.

Para la integración de este nodo (Gateway LoRa - WiFi), se ha programado el microcontrolador para captar mensajes a través de LoRa, y se ha definido una función "Callback" para que, al captar un mensaje, publique en el topic correspondiente (definido en la Actividad 2.2) ese mensaje a través de MQTT. El código se encuentra en el [Anexo III](#).

El nodo presenta el chip lora SX1262. Con el que se puede realizar la comunicación utilizando este protocolo.

OE.4. Fuentes de datos

Este objetivo específico pretende definir e identificar las diferentes fuentes de datos disponibles por el usuario que se han integrado en este trabajo. Se incluyen tanto fuentes de datos internas como externas.

Actividad 4.1. Nodos

El objetivo de esta primera actividad es describir la primera fuente de datos incorporada, los nodos.

Como se ha mencionado en el primer OE, se han incorporado cuatro nodos, pero solo 3 de ellos envían datos al servidor. A continuación, se presenta una tabla resumen con las magnitudes disponibles.

Nodo 1	Nodo 2	Nodo 3
Humedad relativa (aire)	Captura de vídeo	Porcentaje de batería
Temperatura (aire)		Humedad exterior (no implementado)
Humedad relativa (tierra)		Temperatura exterior (no implementado)

Tabla 3.2 Resumen de datos de los Nodos

Actividad 4.2. Aemet

El objetivo de esta actividad es incorporar una fuente de datos externa de una estación meteorológica de AEMET cercana al invernadero. Para ello, se ha solicitado un token a AEMET para poder acceder a su API.

Esta API permite acceder a diferentes datos como pueden ser pronósticos de clima, temperatura actual, humedad actual, velocidad de viento, etc.

Cada uno de estos datos han sido incorporados en el dashboard principal de Home Assistant, para tener una visión global del clima de la zona.

La siguiente figura muestra una captura de esta información.

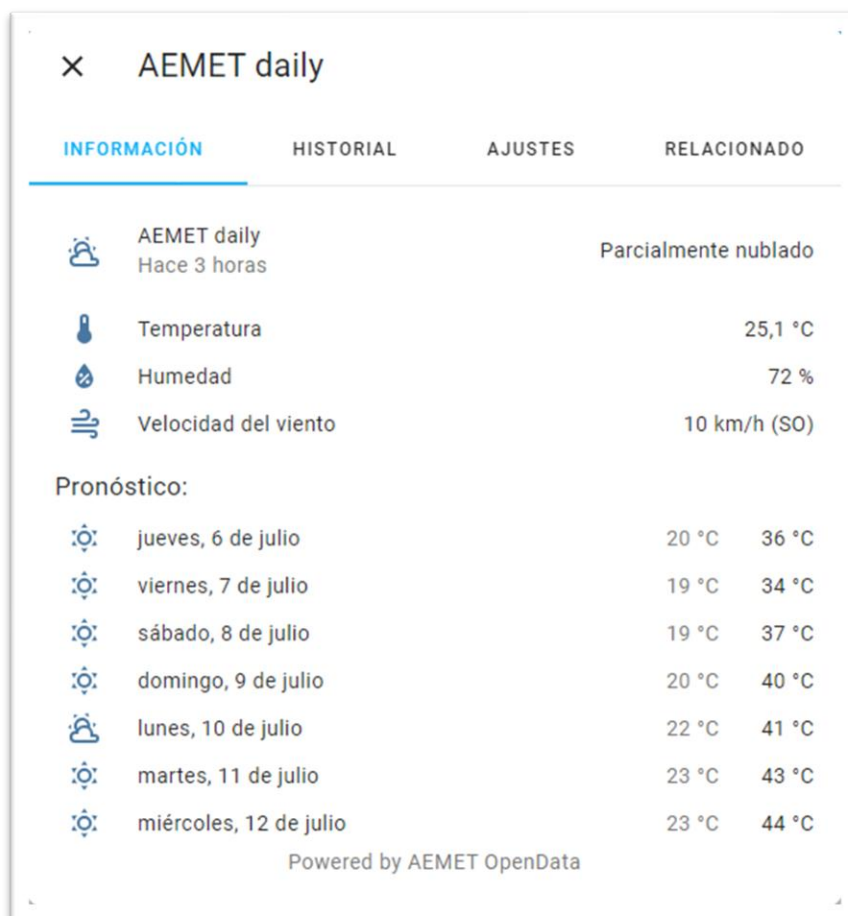











Figura 3.35 Tarjeta Aemet

Esta información permite utilizar estos datos a favor del usuario, por ejemplo, si se prevén altas temperaturas, se puede acondicionar el invernadero para esos días de fuerte calor.

Actividad 4.3. Dispositivo Móvil

La última fuente de datos incorporada es el dispositivo móvil del usuario. Al instalar la APP, puedes otorgar diferentes permisos a la APP, estos permisos permiten monitorizar

la actividad del dispositivo móvil del usuario. Estos datos se visualizan en la siguiente figura.

Sensores		
	iPhone Activity	Walking
	iPhone Average Active Pace	0 m/s
	iPhone BSSID	Not Connected
	iPhone Battery Level	71%
	iPhone Battery State	Not Charging
	iPhone Connection Type	Cellular
	iPhone Distance	445 m
	iPhone Floors Ascended	0 floors
	iPhone Floors Descended	0 floors









	iPhone Focus	Encendido
	iPhone Geocoded Location	xxxxx
	iPhone Last Update Trigger	Siri
	iPhone SIM 1	xxx
	iPhone SIM 2	N/A
	iPhone SSID	Not Connected
	iPhone Steps	604 steps
	iPhone Storage	18,81 % available

Figura 3.36 Datos del dispositivo móvil

Con esta información, se pueden realizar diferentes automatizaciones y cruzamientos de datos, por ejemplo, se podría diseñar una automatización en la que al detectar que el usuario se acerca el invernadero, enviarle una notificación del estado del mismo, incluyendo humedad de la planta y condiciones climáticas. También se podría configurar una automatización que permita que Alexa diga estas condiciones al acercarse al invernadero.

OE.5. Toma de decisiones

En este objetivo se definen diferentes acciones a realizar en función de los datos obtenidos, como pueden ser el riego según la humedad de la planta, o un sistema de notificaciones.

Actividad 5.1 Riego automático

El objetivo de esta actividad es diseñar y programar un sistema de riego automático en función de las condiciones de la planta.

Para lograrlo, se ha hecho uso de las automatizaciones de Home Assistant. Las automatizaciones se pueden programar mediante una herramienta visual o YAML. En este caso, se ha utilizado YAML.

Como se ha explicado en la sección correspondiente, una automatización se compone de un desencadenante, condiciones y acciones. Si ocurre el desencadenante y se cumple la condición, se ejecuta la acción.

A continuación, se muestra la configuración de la automatización.

```
1 alias: Riego Inteligente
2 description: ""
3 trigger:
4   - type: value
5     platform: device
6     device_id: 6710a2a959bcd6b9aa25a5f21a860c05
7     entity_id: sensor.humidity
8     domain: sensor
9     below: 50
10 condition: []
11 action:
12   - type: turn_on
13     device_id: 6710a2a959bcd6b9aa25a5f21a860c05
14     entity_id: switch.pump
15     domain: switch
16   - delay:
17     hours: 0
18     minutes: 0
19     seconds: 5
20     milliseconds: 0
21   - type: turn_off
22     device_id: 6710a2a959bcd6b9aa25a5f21a860c05
23     entity_id: switch.pump
24     domain: switch
25 mode: single
26
```

Figura 3.37 Automatización del riego

Como se observa, el desencadenante o “trigger” es el sensor de humedad, si éste se encuentra por debajo de 50%, se ejecuta el desencadenante.

En cuanto a las condiciones, en este caso no existen condiciones para que se ejecute la acción, pero podría haberse definido, por ejemplo, que el sol se haya ocultado, de tal manera que, al ocurrir el desencadenante, solo se ejecute la acción si es de noche.

Por último, se define la acción. En este caso, se activa la bomba de agua durante cinco segundos, y luego se desactiva. Con esta automatización, se regará la planta en función de la humedad relativa de su tierra.

Actividad 5.2 Sistema de notificaciones

El objetivo de esta actividad es establecer un sistema de notificaciones con el fin de avisar al usuario final si se ha producido un evento importante, como puede ser la activación del riego, de la ventilación, etc.

Se han diseñado notificaciones para la activación y desactivación del riego, así como para la ventilación. Para ilustrarlo, se tomará de ejemplo la construcción de la notificación de activación de riego.

Al igual que en el riego automático, para las notificaciones se utilizan automatizaciones. A continuación, se presenta el YAML correspondiente a esta automatización.

```
1 alias: Riego ON
2 description: ""
3 trigger:
4   - platform: device
5     type: turned_on
6     device_id: 6710a2a959bcd6b9aa25a5f21a860c05
7     entity_id: switch.pump
8     domain: switch
9 condition: []
10 action:
11   - service: notify.mobile_app_iphone_12_de_ivan
12     data:
13       title: Invernadero
14       message: >-
15         Se ha empezado a regar. Humedad de la tierra: {{
16           states('sensor.humidity') }}%
17 mode: single
18
```

Figura 3.38 Automatización de notificaciones

El desencadenante en este caso es la bomba de agua. Al activarse, se aplica el “trigger” y se pasaría a la condición. De nuevo, en este caso no se ha establecido una condición.

La acción es la parte más importante de esta automatización. En la acción se llama a un “servicio”, llamado “notify”. Este servicio vincula Home Assistant con los usuarios que tienen la APP descargada, permitiendo enviar notificaciones. Al establecer el servicio, se define el título de la notificación y el mensaje. En este caso, se avisa que el riego ha comenzado y se envía el valor de humedad actual de la tierra.

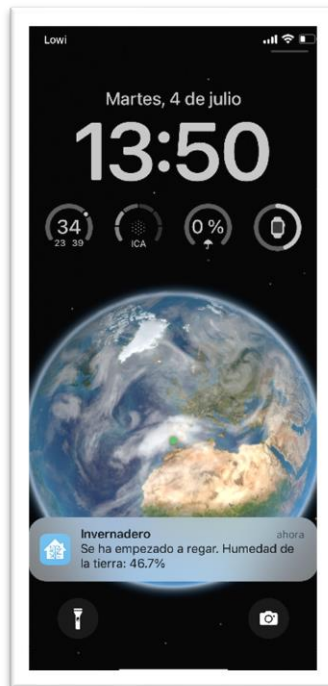


Figura 3.39 Captura de notificación recibida

OE.6. Sistema de control por voz

Este objetivo específico se establece con el fin de realizar consultas y acciones mediante comandos de voz, tales como encender el ventilador o regar la planta. De tal manera que no se tenga que hacer uso de ningún dispositivo móvil u ordenador. Para ello, se utilizará Alexa.

Actividad 6.1 AWS

El objetivo de esta actividad es realizar el primer paso de configuración de Amazon Alexa.

Amazon Web Services (AWS) es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com.

Dentro de AWS, se utilizará Lambda. AWS Lambda es una plataforma informática sin servidor basada en eventos proporcionada por Amazon como parte de Amazon Web Services. Es un servicio informático que ejecuta código en respuesta a eventos y administra automáticamente los recursos informáticos requeridos por medio de código.

Lambda, permitirá vincular todos los dispositivos integrados en Home Assistant con la APP Alexa, y, por tanto, podrán ser controlados por voz.

El código se proporciona en el [Anexo III](#).

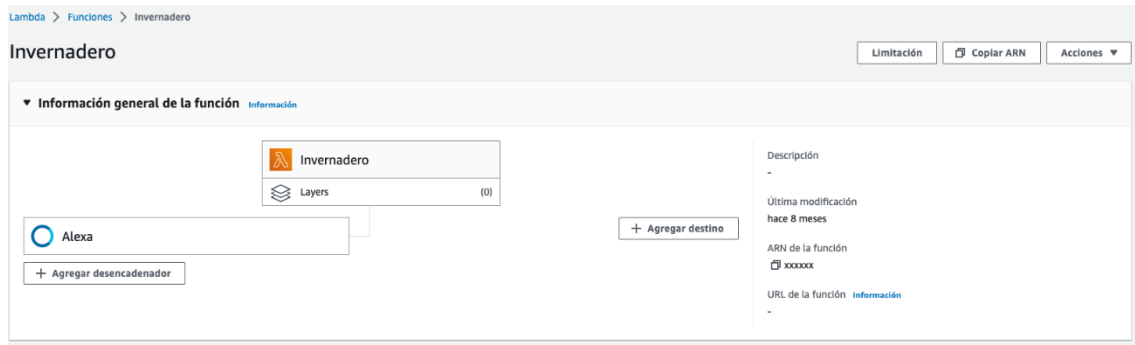


Figura 3.40 Despliegue AWS Lambda

Actividad 6.2 Alexa Developer Console

El objetivo de esta actividad es desarrollar una “Skill” que, junto con la función Lambda definida anteriormente, permita vincular los dispositivos integrados en Home Assistant con la APP de Alexa.

Una skill es una aplicación de voz que permite a los usuarios interactuar con dispositivos habilitados para Alexa, como los altavoces Echo de Amazon. Las skills son el equivalente a las aplicaciones móviles, pero están diseñadas específicamente para ser controladas y accedidas a través de comandos de voz.

Para la creación de una skill, se requerirá una cuenta en Alexa Developer Console. Alexa Developer Console es una plataforma en línea proporcionada por Amazon que permite a los desarrolladores crear, configurar y administrar skills para dispositivos habilitados para Alexa, como los altavoces Echo y otros dispositivos compatibles con Alexa.

La consola proporciona una interfaz gráfica de usuario (GUI) que simplifica el proceso de desarrollo y permite a los desarrolladores diseñar skills interactivas para interactuar con los usuarios a través de comandos de voz. Los desarrolladores pueden utilizar la consola para crear diálogos, configurar interacciones y respuestas, probar sus skills y publicarlas en la tienda de habilidades de Alexa.

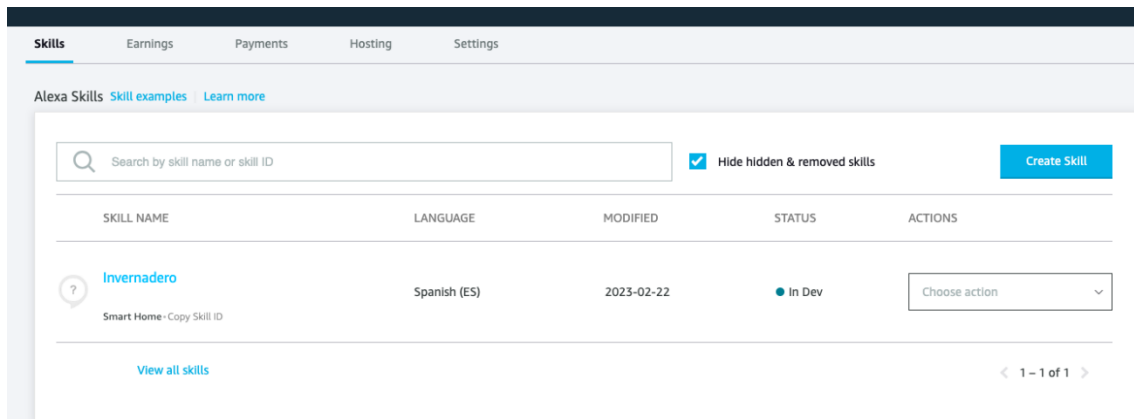


Figura 3.41 Skill Desarrollada

Se ha creado una skill en modo desarrollo, es decir, solo puede ser utilizada por el usuario propietario de la skill.

Se ha configurado una skill de tipo "Smart Home", y se han establecido los permisos y credenciales necesarios para acceder a la instancia de Home Assistant del invernadero.

La skill aparecerá configurada en el apartado "Mis Skills" en la APP de Alexa, tal y como se muestra en la siguiente figura.

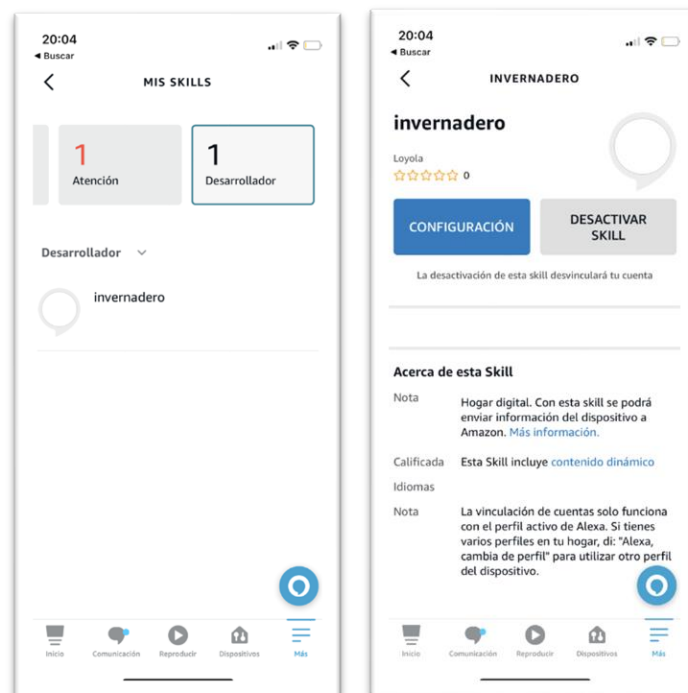


Figura 3.42 Captura de la Skill desarrollada en la APP Alexa

Una vez localizada la skill, se sincronizarán automáticamente todos los dispositivos. Para tenerlos todos localizados y ordenados, se ha creado un grupo llamado “laboratorio” y se han vinculado a él todos los dispositivos, tal y como muestra la siguiente figura.

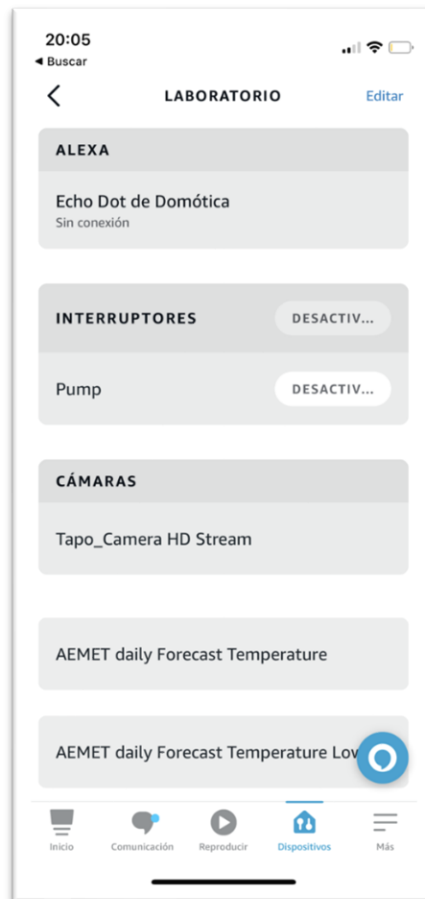


Figura 3.43 Captura de dispositivos vinculados con Alexa

Actividad 6.3 Rutinas

El objetivo de esta actividad es configurar “rutinas” de Alexa para definir comandos de voz y realizar acciones en base a estos.

Una rutina de Alexa es una serie de acciones automatizadas que se activan con un solo comando de voz. Permite a los usuarios personalizar y programar una secuencia de acciones para que Alexa las realice de forma consecutiva, como reproducir música, encender luces, proporcionar noticias y realizar otras tareas simultáneamente.

Por ejemplo, se ha configurado una rutina para el siguiente comando de voz: “Alexa, riega”. Cuando este comando es lanzado por el usuario, el sistema de riego comenzará a regar la planta. La siguiente figura muestra una captura de la rutina configurada.

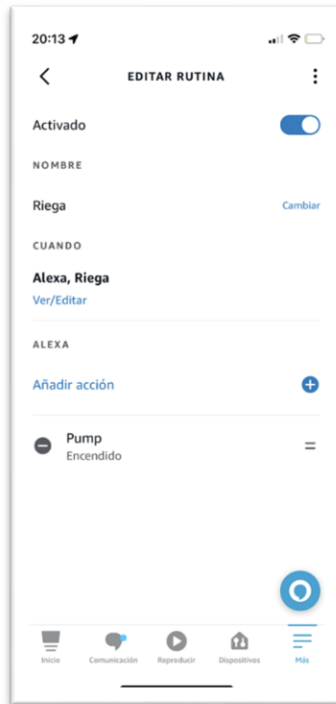


Figura 3.44 Captura de rutina de Alexa

OE.7. Uso de energía solar

En este objetivo específico se abarca la implementación de módulos solares para cumplir con uno de los objetivos principales de este trabajo, reducir la huella de carbono asociada a la instalación de dispositivos IoT en invernaderos.

Actividad 7.1 Flujo de trabajo

Se ha decidido implementar en uno de los nodos mencionados anteriormente, concretamente para el Nodo 4, un sistema de alimentación autónoma. El sistema consiste en una batería, un módulo de carga, y un panel solar.

El flujo de trabajo de este sistema pretende alimentar el Nodo 4 durante el día mediante placas solares, el excedente de energía producida por estas placas se destina a la carga de la batería, para que por la noche esté suficientemente cargada y pueda alimentar al microcontrolador. Esta gestión de carga se realiza gracias a un módulo de carga diseñado específicamente para ello. En la siguiente actividad se detallará su función.

Actividad 7.2 Módulo de carga

El objetivo de esta actividad es diseñar un dispositivo para gestionar la carga de baterías de forma eficiente, permitiendo redirigir la energía en función de las condiciones ambientales actuales, es decir, si es de día, las placas solares alimentarán al microcontrolador, y el excedente de energía producida por estas placas, se destinará a la carga de la batería de litio para que, por la noche, el microcontrolador pueda seguir funcionando.

El modulo de carga lo confortan diferentes componentes de hardware, todos conectados entre sí en una PCB diseñada (ver schematic en [Anexo II](#)).

El principal componente del módulo de carga es el chip TP4056. Este chip es el encargado de gestionar la carga de la batería de litio y proporcionar la corriente necesaria en cada momento. Además, el chip incorpora protecciones adicionales tales como protección contra sobrecargas y protección contra cortocircuito, previniendo así posibles daños a la batería. El chip contiene dos entradas, donde se conectan el positivo y negativo de la placa solar; y cuatro salidas, donde se conectan el positivo y negativo de la batería, y el positivo y negativo a utilizar.

El principal inconveniente de utilizar este chip es que el voltaje a la salida no es apto para alimentar un ESP32, además, este chip está ideado para cargar baterías de litio y, una vez cargadas, utilizar la salida proporcionada. Esto no es apto para esta aplicación ya que es necesario utilizar el microcontrolador a la vez que las baterías son cargadas. Utilizar este chip conectando el microcontrolador mientras se carga la batería es una mala práctica ya que puede dañar las baterías. El chip incorpora una lógica para cortar la corriente una vez el voltaje de la batería es el adecuado, si se conecta un microcontrolador en paralelo, este voltaje se puede alterar y el chip puede seguir intentando cargar la batería incluso cuando está ya está totalmente cargada.

Para solventar este problema, se ha diseñado un “Powerpath”, el cual incluye un Mosfet P-Channel, un diodo Schottky, un boost converter, una resistencia y el chip tp4056.

La idea de este circuito es poder utilizar el microcontrolador a la vez que se carga la batería, para ello, el mosfet es encargado de cambiar entre alimentación por panel solar y alimentación por batería en función del voltaje que le llega del panel solar. Si el voltaje baja de un umbral, lo que quiere decir que es de noche, se aplica el cambio, permitiendo al microcontrolador utilizar la batería como fuente de alimentación. El diodo es utilizado para prevenir una posible corriente inversa hacia las placas solares cuando se produce este cambio. Como aclaración, se ha utilizado un diodo de tipo Schottky ya que proporciona conmutaciones muy rápidas entre los estados de conducción directa e inversa y muy bajas tensiones umbral.

En la siguiente figura aparece un diagrama con el conexionado de este “Powerpath”.

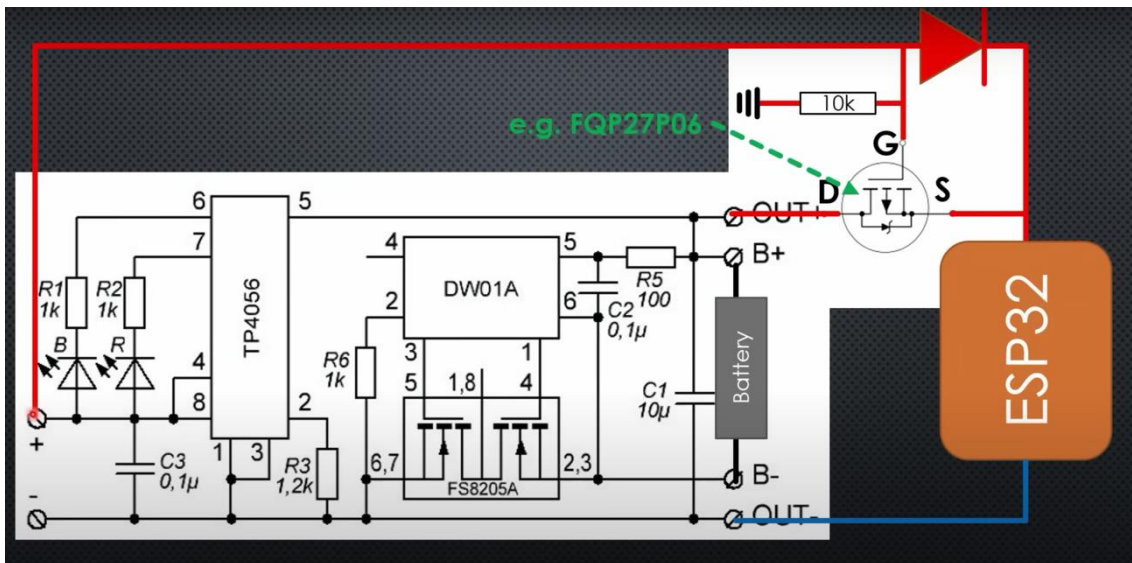


Figura 3.45 PowerPath

El siguiente aspecto por cubrir es el voltaje de entrada al microcontrolador. Los ESP32 funcionan a 3.3V, pero las placas de desarrollo contienen un LDO que puede reducir una entrada de 5V a 3.3V. En este caso, se ha decidido optar por alimentar el microcontrolador con 5V, ya que, en un futuro, es posible que se deban conectar sensores con un voltaje de funcionamiento de 5V.

Para alcanzar los 5V necesarios para el funcionamiento del microcontrolador, se ha utilizado un pequeño boost converter (Step-up) de 5V 1A. Este módulo eleva la tensión a la salida del módulo de carga a 5V. La tensión a la salida del módulo de carga es variable, porque a medida que la batería se descarga, su voltaje disminuye y, el ESP 32, necesita un voltaje constante.

En la siguiente figura se muestra el módulo de carga soldado y con el "PowerPath" y boost converter incorporados.

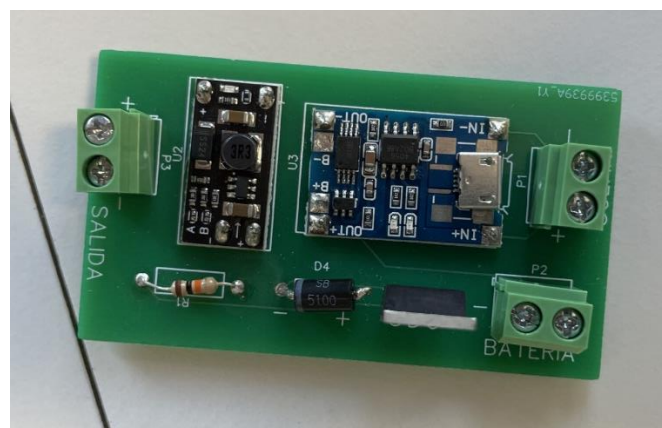


Figura 3.46 Módulo de carga

Actividad 7.3 Placas solares

Esta actividad tiene como objetivo incorporar placas solares para integrar con el módulo de carga diseñado en la actividad anterior.

Las placas solares juegan un papel muy importante en la alimentación del nodo 4. Permiten construir un sistema de alimentación autónomo al aprovechar la energía solar.

Se han utilizado dos paneles solares de 7V y 250mA cada uno. Conectados en paralelo, permiten alcanzar hasta 500mA en pleno rendimiento. Un ESP32 tiene un consumo máximo de 240mA (Sin utilizar WiFi) lo que permite cargar las baterías al mismo tiempo que se alimenta el microcontrolador con la energía excedente.

En la siguiente figura se observan los paneles solares, junto con el módulo de carga y el nodo 4.

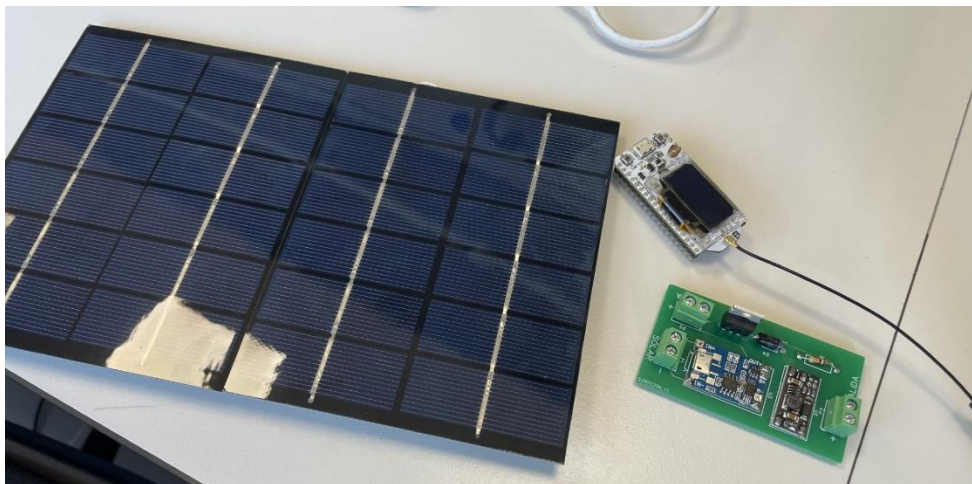


Figura 3.47 Módulo de carga junto con Nodo 4 y Placa Solar

4. Resultados y Conclusiones

En conclusión, en este TFG se ha desarrollado un invernadero inteligente que incorpora tecnologías innovadoras para mejorar la eficiencia y la sostenibilidad de la producción agrícola.

Los agricultores podrán controlar el riego a distancia a través de una plataforma en línea, supervisar las condiciones ambientales del invernadero en tiempo real y ahorrar agua y energía evitando el derroche en zonas donde no es necesario mediante la implantación de tecnologías del Internet de las Cosas (IoT) y sistemas de automatización.

Estos invernaderos podrán producir su propia energía renovable gracias al uso de módulos solares, lo que reducirá su dependencia de la red y su huella de carbono. El servidor extremadamente modular y fácil de usar que se ha desarrollado facilitaría el escalado de datos en función de las necesidades, al tiempo que resolvería los problemas de complejidad que plantea la conexión de los nodos de comunicación y los sensores a los servidores.

En general, esta innovadora solución para invernaderos proporciona un método sostenible de producción de cultivos que es respetuoso con el medio ambiente y eficaz para satisfacer las necesidades de los cultivos. Esta solución podría tener un impacto global porque aborda cuestiones como el cambio climático, que repercute en la productividad agrícola.

4.1 Evidencias

A continuación, se mostrarán algunas evidencias y resultados de este trabajo. En primer lugar, se mostrará la interfaz web (APP) en su pleno funcionamiento (figura 4.1).

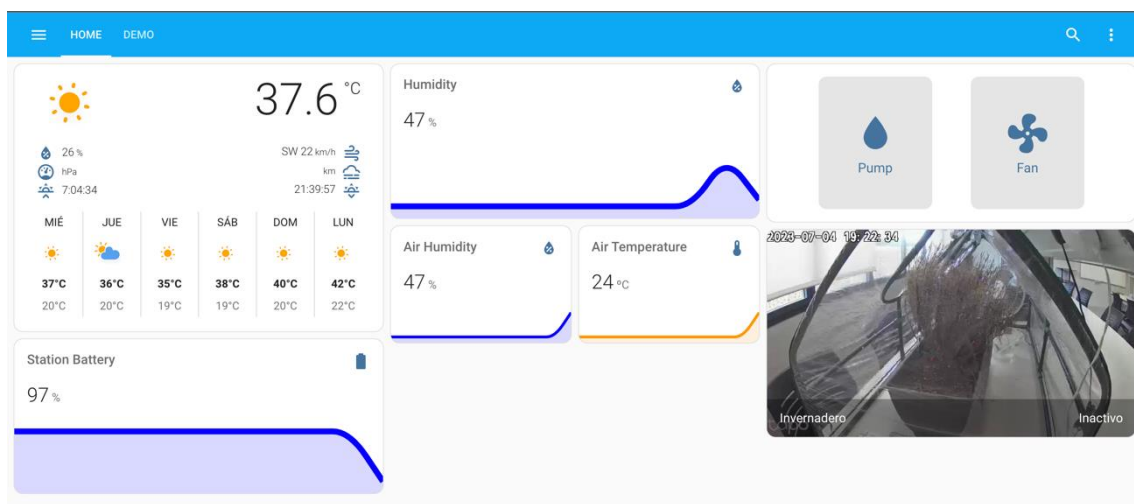


Figura 4.1: Interfaz Web (APP)

La interfaz de usuario, desplegada en un servidor y accesible desde cualquier parte del mundo, contiene datos de agencias de meteorología (AEMET), datos de sensores físicos, actuadores como la bomba de riego o el sistema de ventilación, y la cámara de video.

Esta interfaz también está disponible para dispositivos móviles descargando la aplicación de Home Assistant. En la figura 4.2 se muestra una captura de la interfaz vista desde un dispositivo móvil.

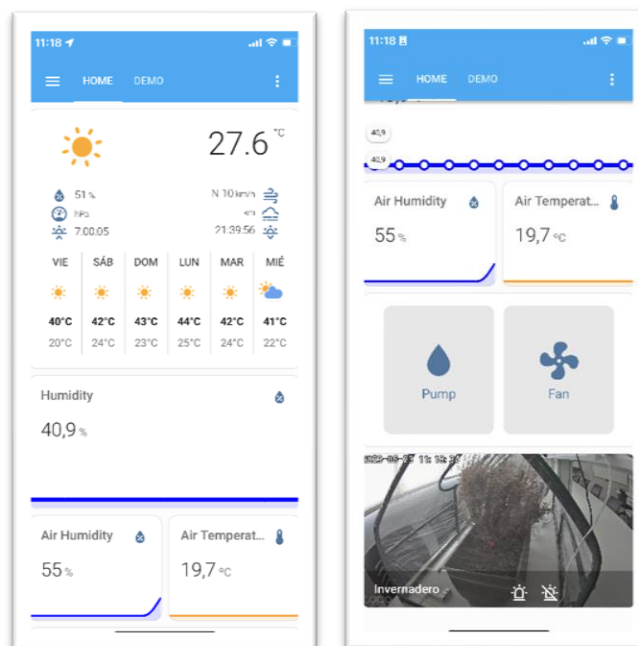


Figura 4.2 Aplicación Móvil

También, desde la interfaz, se pueden observar histórico de datos.

A continuación, se mostrarán algunos datos recogidos considerados relevantes a la hora de tomar decisiones sobre el invernadero por parte de usuario.

Estos datos pueden ser accesibles tanto desde la APP como desde el navegador. Se encuentran en la pestaña “historial” y pueden ser filtrados por fecha y hora. Estos datos se almacenan automáticamente en la base de datos interna de Home Assistant, la cual se puede establecer una persistencia.

En este caso, se ha establecido una persistencia de datos de 60 días, es decir, los datos recogidos se borrarán tras transcurrir 60 días. Esto se ha realizado con el objetivo de no saturar la memoria del servidor.

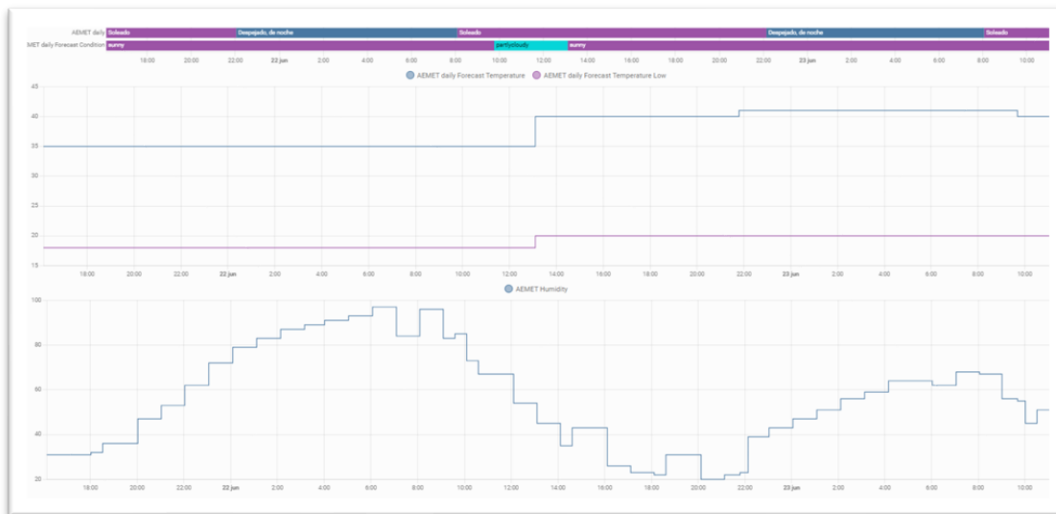


Figura 4.3 Histórico de datos AEMET

La figura 4.3 muestra la evolución de las condiciones meteorológicas de la zona en los últimos 3 días. Se observan datos cuantitativos como la humedad relativa, la temperatura media, o las temperaturas mínimas alcanzadas.

Por otro lado, también se observan datos cualitativos, como la condición del clima, que indica si está soleado durante el día (o despejado por la noche), nublado, parcialmente nublado, etc.

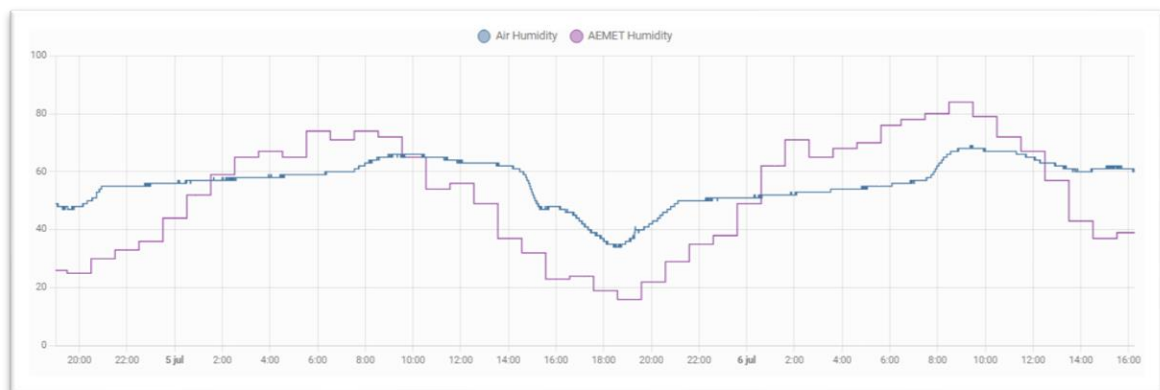


Figura 4.4 Comparación Humedad del Aire (Invernadero) vs Humedad del Exterior (AEMET)

La figura anterior (figura 4.4) presenta datos de 3 días comparando la humedad relativa del aire dentro del invernadero y la humedad relativa del aire en el exterior usando los datos de Aemet. Se observan grandes fluctuaciones en el exterior, en cambio, en el interior fluctúa menos.

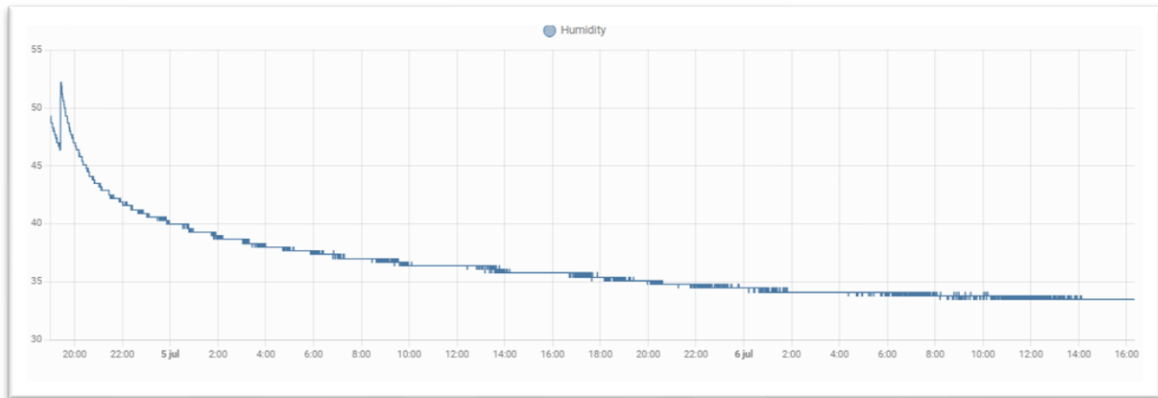


Figura 4.5 Evolución de la humedad del suelo

En la figura anterior (figura 4.5) se observa como ha evolucionado la humedad del suelo de la planta tras ser regada. Se observa una asíntota en torno al 33.5%. A continuación, se presentará una figura focalizándose en el momento de riego (figura 4.6).



Figura 4.6 Cambio de humedad ante riego

Se observa en la figura anterior como la humedad incrementa justo cuando se activa la bomba de agua, indicando que el riego ha sido iniciado correctamente.

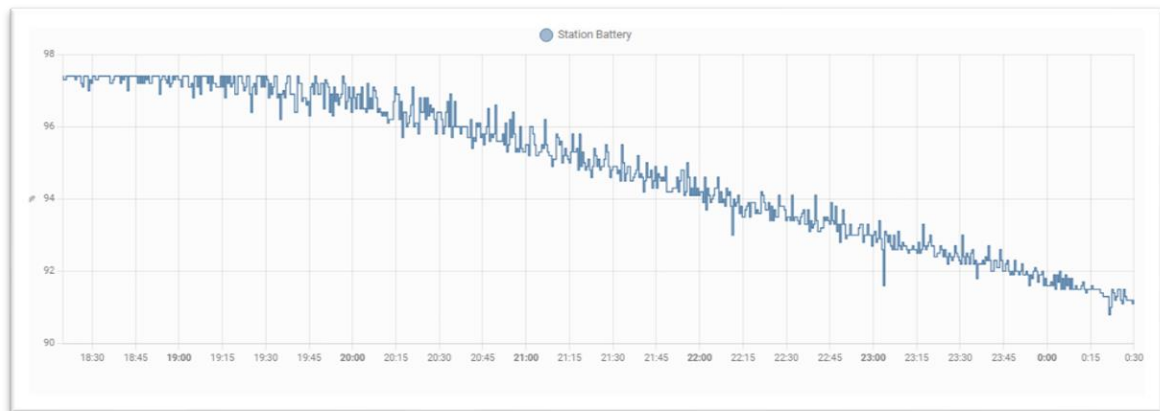


Figura 4.7 Nivel de batería del Nodo 4

En la figura anterior (figura 4.7) se observa el nivel de batería del microcontrolador (Nodo 4) alimentado por placas solares. Se observa como al utilizar solo la batería, esta va decayendo conforme pasa el tiempo. Cabe recordar que estos datos son recibidos por LoRa al Gateway (Nodo 3) y posteriormente enviados al servidor mediante MQTT.

4.2 Problemas encontrados

En esta sección se detallarán algunos de los problemas encontrados durante la realización de este trabajo. Al exponerlos, puede ser útil en futuros despliegues de arquitecturas como estas como referencia.

El primer inconveniente surgió durante el arranque de Home Assistant, y fue la sincronización con el kernel del tiempo (Failed to start Wait Until Kernel Time Synchronized). Este problema surge cuando Home Assistant intenta comunicarse con el servidor de fecha y hora establecido por defecto (Google) a través de NTP (Network Time Protocol) utilizando el puerto 123/UDP. El problema reside en que la red Wi-Fi utilizada (en este caso la proporcionada por la universidad) bloquea ese puerto. Para solucionarlo, se debe preparar un USB para conectarlo durante el arranque del sistema, de forma parecida a la configuración Wi-Fi inicial mencionada en el apartado de configuración, pero en este caso se utiliza una nueva configuración NTP (Anexo I)

Este problema presentaba consecuencias como la desincronización entre dispositivos, o instantes de tiempo registrados erróneamente, así como tiempo de inicio y carga elevados debido a la constante petición al servidor de fecha y hora fallida.

El segundo problema surge en fases tardías del desarrollo del proyecto, concretamente cuando se estaban recogiendo datos. El problema surge cuando desde la universidad, asignan la IP utilizada por el servidor a otro dispositivo, por tanto, la raspberry pi no podía conectarse a internet ya que la IP solicitada ya estaba en uso. Esto provocó dejar de recoger datos ya que el servidor no estaba disponible. El problema tuvo dos fases, la

primera fue la detección del error, ya que a simple vista es complicado conocer el motivo de la falla de conexión, la segunda fase fue la solicitud de una nueva IP al servicio TIC, ya que la que habían cambiado no podían devolverla. Ya obtenida la nueva IP para la Raspberry, se configuró de nuevo con la nueva IP. El inconveniente asociado a este problema es que todos los Nodos conectados al servidor apuntaban a la IP antigua, y hubo que reprogramar todos para que apuntasen a la nueva IP.

El tercer problema, en relación con el segundo, fue encontrado en los últimos días de desarrollo del proyecto, problemas con el WiFi de la universidad. El Wifi utilizado hasta el momento de la detección de este problema fue el proporcionado por la universidad. El router configurado en la universidad permite conexiones a la red Wifi filtrando por MACs, muchos dispositivos, concretamente Alexa y los nodos, fueron desconectados de la red de repente y no pudieron volver a ser conectados. Esto involucra que, sin conexión a internet, los datos de los sensores no pueden ser enviados al servidor y, por tanto, no pueden ser ni recogidos ni visualizados.

Finalmente, ante la cantidad de fallos producidos por un mismo motivo (red Wifi), y ante la posibilidad de falla, se decidió adquirir un router 4G y contratar una tarjeta SIM con transferencia de datos por cuenta ajena. La nueva arquitectura de red fue montada sin problemas y todos los inconvenientes han sido resueltos.

4.3 Trabajo futuro

Para finalizar, esta sección pretende establecer nuevos campos de investigación y nuevas integraciones a partir de la infraestructura IoT montada en este trabajo.

Para comenzar, uno de los campos en los que investigar puede centrarse en la detección de anomalías en la planta usando visión artificial. Al haber incorporado una cámara para la visualización en tiempo real de las condiciones físicas de la planta, se plantea realizar un algoritmo que involucre realizar capturas periódicas con el fin de analizar la evolución del estado de la planta. Por ejemplo, se puede analizar el nivel de verde, el movimiento de las ramas (si decaen o levantan), el crecimiento, etc.

Por otro lado, otro campo abierto es diseñar un sistema de riego automático complejo. A partir de los datos adquiridos (humedad del suelo, aire, exterior, etc.) se podría realizar un algoritmo o clúster para regar en el momento y la cantidad óptimos. Este algoritmo podría cruzar los diferentes datos extraídos y tomar decisiones en base a ellos. Por ejemplo, se podría implementar un algoritmo de riego que tenga en cuenta la humedad de la planta, pero también si es de día o de noche o si lloverá en breve.

Por último, se propone investigar en el despliegue de una red LoRa de mayor alcance. Esto puede incluir la implementación de una malla (mesh) que permita desplegar

numerosos nodos y comunicarlos entre sí, de tal manera que el Gateway pueda adquirir los datos de todos los nodos de forma óptima. Se plantea la siguiente figura:

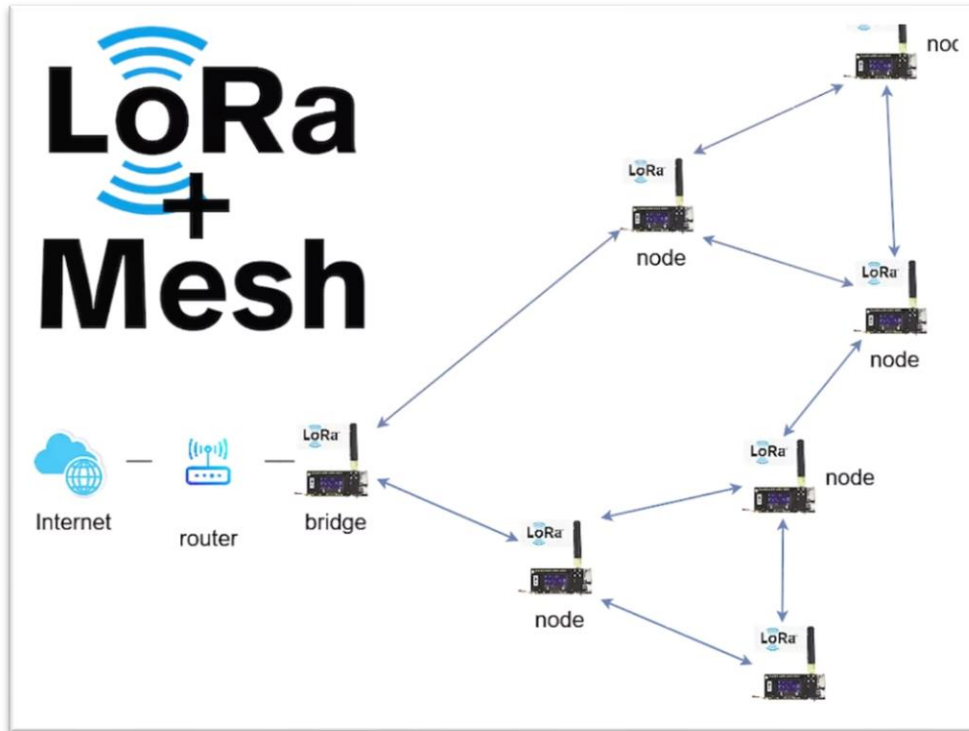


Figura 4.8 Propuesta de malla LoRad

En definitiva, este trabajo involucra todo lo mencionado en apartados anteriores y satisface con creces el objetivo planteado al inicio, no obstante, se abren nuevas líneas de investigación, tales como las propuestas en este apartado.

Bibliografía

- [1] FAO. (2017). The future of food and agriculture: Trends and challenges. Food and Agriculture Organization of the United Nations.
- [2] Rayhana, R., Xiao, G., & Liu, Z. (2020). Internet of Things Empowered Smart Greenhouse Farming. *IEEE Journal of Radio Frequency Identification*, 4(3), 195-211.
- [3] Wu, X., Liu, S., Zhang, Y., Chen, L., & Chen, X. (2021). Smart greenhouse system based on an Internet of Things platform. *Applied Sciences*, 11(1), 14.
- [4] H. Lia, Y. Guo, H. Zhao, Y. Wang, and D. Chow, "Towards Automated Greenhouse: A State of the Art Review on Greenhouse Monitoring Methods and Technologies Based on Internet of Things," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3157-3168, May 2020.
- [5] A. Pardossi, F. Tognoni, and L. Incrocci, "Mediterranean greenhouse technology," *Chronica Horticulturae*, vol. 44, no. 2, pp. 28-34, 2004.
- [6] R. Dagar, S. Som, and S. K. Khatri, "Smart farming—IoT in agriculture," in *Proc. IEEE Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Coimbatore, India, 2018, pp. 1052-1056.
- [7] T. Wiangtong and P. Sirisuk, "IoT-based versatile platform for precision farming," in *Proc. IEEE 18th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Bangkok, Thailand, 2018, pp. 438-441.
- [8] T. J. Jeanita, V. Sarasvathi, M. Harsha, B. Bhavani, and T. Kavyashree, "An automated greenhouse system using agricultural Internet of Things for better crop yield," in *Proc. Smart Cities Symp.*, 2018, pp. 129-134.
- [9] Good Agricultural Practices for Greenhouse Vegetable Crops Principles for Mediterranean Climate Areas, FAO, Rome, Italy, 2013. [Online]. Available: <http://www.fao.org>
- [10] U. Çakır and E. Şahin, "Using solar greenhouses in cold climates and evaluating optimum type according to sizing, position and location: A case study," *Comput. Electron. Agr.*, vol. 117, pp. 245-257, Sep. 2015.
- [11] J. Xie et al., "Gobi agriculture: An innovative farming system that increases energy and water use efficiencies. A review," *Agronomy Sustain. Develop.*, vol. 38, no. 6, p. 62, 2018.

- [12] F. Tognoni, A. Pardossi, and G. Serra, "Strategies to match greenhouses to crop production," in Proc. Int. Symp. Growing Media Hydroponics 481, 1997, pp. 451–462.
- [13] J. Kipp, "Optimal climate regions in Mexico for greenhouse crop production," Wageningen UR Greenhouse Horticult., Bleiswijk, The Netherlands, Rep. GTB.
- [14] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," Comput. Electron. Agr., vol. 156, pp. 467–474, Jan. 2019.
- [15] A. Medela, B. Cendón, L. González, R. Crespo, and I. Nevares, "IoT multiplatform networking to monitor and control wineries and vineyards," in Proc. Future Netw. Mobile Summit, Lisboa, Portugal, 2013, pp. 1–10.
- [16] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin, and K. Ellis, "IoT in agriculture: Designing a Europe-wide large-scale pilot," IEEE Commun. Mag., vol. 55, no. 9, pp. 26–33, Sep. 2017.
- [17] Z. Pang, Q. Chen, W. Han, and L. Zheng, "Value-centric design of the Internet-of-Things solution for food supply chain: Value creation, sensor portfolio and information fusion," Inf. Syst. Front., vol. 17, no. 2, pp. 289–319, 2015.
- [18] J. M. Talavera et al., "Review of IoT applications in agro-industrial and environmental fields," Comput. Electron. Agr., vol. 142, pp. 283–297, Nov. 2017.
- [19] S. L. Li, Y. Han, G. Li, M. Zhang, L. Zhang, and Q. Ma, "Design and implementation of agricultural greenhouse environmental monitoring system based on Internet of Things," in Proc. Appl. Mech. Mater., vol. 121, 2012, pp. 2624–2629.
- [20] N. Kaewmard and S. Saiyod, "Sensor data collection and irrigation control on vegetable crop using smart phone and wireless sensor networks for smart farm," in Proc. IEEE Conf. Wireless Sens. (ICWiSE), Subang, Malaysia, 2014, pp. 106–112.
- [21] Q. T. Minh et al., "A cost-effective smart farming system with knowledge base," in Proc. 8th Int. Symp. Inf. Commun. Technol., 2017, pp. 309–316.
- [22] A. Mittal, S. Sarangi, S. Ramanath, P. V. Bhatt, R. Sharma, and P. Srinivasu, "IoT-based precision monitoring of horticultural crops— A case-study on cabbage and capsicum," in Proc. IEEE Global Humanitarian Technol. Conf. (GHTC), San Jose, CA, USA, 2018, pp. 1–7.
- [23] J. Park, J.-H. Choi, Y.-J. Lee, and O. Min, "A layered features analysis in smart farm environments," in Proc. Int. Conf. Big Data Internet Thing, 2017, pp. 169–173.

[24] S. R. Rupanagudi, B. Ranjani, P. Nagaraj, V. G. Bhat, and G. Thippeswamy, "A novel cloud computing based smart farming system for early detection of borer insects in tomatoes," in Proc. IEEE Int. Conf. Commun. Inf. Comput. Technol. (ICCICT), Mumbai, India, 2015, pp. 1–6.

[25] J. Aleotti, M. Amoretti, A. Nicoli, and S. Caselli, "A smart precision agriculture platform for linear irrigation systems," in Proc. IEEE 26th Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM), Split, Croatia, 2018, pp. 1–6.

[26] D. S. Domingues, H. W. Takahashi, C. A. Camara, and S. L. Nixdorf, "Automated system developed to control PH and concentration of nutrient solution evaluated in hydroponic lettuce production," *Comput. Electron. Agr.*, vol. 84, pp. 53–61, Jun. 2012.

[27] D. Saraswathi, P. Manibharathy, R. Gokulnath, E. Sureshkumar, and K. Karthikeyan, "Automation of hydroponics green house farming using IoT," in Proc. IEEE Int. Conf. Syst. Comput. Autom. Netw. (ICSCA), Pondicherry, India, 2018, pp. 1–4.

[28] K. Ferentinos and L. Albright, "Predictive neural network modeling of ph and electrical conductivity in deep–trough hydroponics," *Trans. ASAE*, vol. 45, no. 6, pp. 2007–2015, 2002.

[29] M. Mehra, S. Saxena, S. Sankaranarayanan, R. J. Tom, and M. Veeramanikandan, "IoT based hydroponics system using deep neural networks," *Comput. Electron. Agr.*, vol. 155, pp. 473–486, Dec. 2018.

[30] B. Paulchamy, N. Balaji, S. D. Pravatha, P. H. Kumar, and T. J. Frederick, "A novel approach for automating & analyzing hydroponic farms using Internet of Things," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3, no. 3, pp. 1230–1234, 2018.

[31] J. Pitakphongmetha, N. Boonnam, S. Wongkoon, T. Horanont, D. Somkiadcharoen, and J. Prapakornpilai, "Internet of Things for planting in smart farm hydroponics style," in Proc. IEEE Int. Comput. Sci. Eng. Conf. (ICSEC), Chiang Mai, Thailand, 2016, pp. 1–5.

[32] R. Vidhya and K. Valarmathi, "Survey on automatic monitoring of hydroponics farms using IoT," in Proc. IEEE 3rd Int. Conf. Commun. Electron. Syst. (ICCES), Coimbatore, India, 2018, pp. 125–128.

[33] S. Bhowmick, B. Biswas, M. Biswas, A. Dey, S. Roy, and S. K. Sarkar, "Application of IoT-enabled smart agriculture in vertical farming," in *Advances in Communication, Devices and Networking*. Singapore: Springer, 2019, pp. 521–528.

[34] M. I. H. Bin Ismail, and N. M. Thamrin, "IoT implementation for indoor vertical farming watering system," in Proc. IEEE Int. Conf. Elect. Electron. Syst. Eng. (ICEESE), Kanazawa, Japan, 2017, pp. 89–94. [52] F. C. L. Belista, M. P. C. Go, L. L. Luceñara, C. J.

G. Policarpio, X. J. M. Tan, and R. G. Baldovino, "A smart aeroponic tailored for IoT vertical agriculture using network connected modular environmental chambers," in Proc. IEEE 10th Int. Conf. Humanoid Nanotechnol. Inf. Technol. Commun. Control Environ. Manag. (HNICEM), Baguio City, Philippines, 2018, pp. 1–4.

[35] Zu, X., Bai, Y., Yao, X., 2016. Data-centric publish-subscribe approach for Distributed Complex Event Processing deployment in smart grid Internet of Things. In: in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 710–713

[36] Murugesan, P., et al., 2017. Adopting Attribute-Based Access Control to Data Distribution Service. In: in 2017 International Conference on Software Security and Assurance (ICSSA), pp. 112–115.

[37] Badenhop, C.W., et al., 2017. The Z-Wave routing protocol and its security implications. Comput. Secur. 68, 112–129

[38] Tournier, J., et al., 2021. A survey of IoT protocols and their security issues through the lens of a generic IoT stack. Internet of Things 16, 100264

[39] da Cruz, M.A.A., et al., 2018. A Proposal for Bridging the Message Queuing Telemetry Transport Protocol to HTTP on IoT Solutions. In: in 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1–5.

[40] Al-Masri, E., et al., 2020. Investigating messaging protocols for the internet of things (IoT). IEEE Access 8, 94880–94911.

[41] Sanjuan, E.B., et al., 2020. Message queuing telemetry transport (MQTT) security: a cryptographic smart card approach. IEEE Access 8, 115051–115062.

[42] Syafarinda, Y., Akhadin, F., Fitri, Z. E., Yogiswara, Widiawanl, B., and Rosdiana, E. 2018. The Precision Agriculture Based on Wireless Sensor Network with MQTT Protocol, IOP Conference Series: Earth and Environmental Science, 207(1), pp. 1–9.

[43] Saxena, S., Farag, H.E.Z., El-Taweel, N., 2021. A distributed communication framework for smart Grid control applications based on data distribution service. Electr. Power Syst. Res. 201, 107547.

[44] Hakiri, A., et al., 2015. Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications. IEEE Commun. Mag. 53 (9), 48–54.

- [45] Tekinerdogan, B., Koksal, " O. " and Çelik, T. (2017) 'Data Distribution Service-Based Architecture Design for the Internet of Things Systems BT - Connected Environments for the Internet of Things: Challenges and Solutions', in Mahmood, Z. (ed.). Cham: Springer International Publishing, pp. 269–285.
- [46] Beckmann, K., Dedi, O., 2015. sDDS: A portable data distribution service implementation for WSN and IoT platforms. In: in 2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES), pp. 115–120.
- [47] Sinha, A., Shrivastava, G., Kumar, P., 2019. Architecting user-centric internet of things for smart agriculture. Sustain. Comput. Inf. Syst. 23, 88–102.
- [48] Peladarinos, N.; Cheimaras, V.; Piromalis, D.; Arvanitis, K.G.; Papageorgas, P.; Monios, N.; Dogas, I.; Stojmenovic, M.; Tsaramirsis, G. Early Warning Systems for COVID-19 Infections Based on Low-Cost Indoor Air-Quality Sensors and LPWANs. Sensors 2021, 21, 6183.
- [49] Raspberry Pi Foundation. "What is a Raspberry Pi?" Raspberry Pi Documentation, 2021.

ANEXO I – Configuración Avanzada HA

Se puede utilizar una unidad USB con HassOS para configurar las opciones de red, el acceso SSH al host y para instalar actualizaciones. Para ello se debe formatear una memoria USB con FAT32/EXT4/NTFS y nombrarla CONFIG (en mayúsculas).

Alternativamente se puede crear una carpeta CONFIG dentro de la partición de arranque. Se debe utilizar la siguiente estructura de directorios dentro de la unidad USB:

```
network/  
modules/  
modprobe/  
udev/  
authorized_keys  
timesyncd.conf  
hassos-xy.raucb
```

- La carpeta de red puede contener cualquier tipo de archivos de conexión NetworkManager.
- La carpeta modules es para los archivos de configuración de modules-load.
- La carpeta modprobe es para archivos de configuración de módulos (/etc/modprobe.d).
- La carpeta udev es para los archivos de reglas udev.
- El archivo authorized_keys activa el acceso SSH de depuración en el puerto 22222.
- El archivo timesyncd.conf permite configurar diferentes servidores NTP. HassOS no arrancará sin servidores de tiempo que funcionen correctamente.
- El archivo hassos-*.raucb es una actualización OTA de firmware que será instalada. Estos se pueden encontrar en la página de lanzamiento.

Los archivos de texto de la memoria USB deben tener caracteres de fin de línea Unix (LF). Si crea la memoria USB en una máquina Windows, asegúrese de usar Notepad++, Visual Studio Code o cualquier otro editor que soporte diferentes finales de línea. En Notepad++ LF EOL se puede activar con la opción Editar -> Conversión EOL -> Unix (LF). Puedes poner esta memoria USB en el dispositivo y será leída en el arranque y los archivos escritos en los lugares correctos. También puedes activar este proceso más tarde usando `ha os import` desde el CLI o llamando a `systemctl restart hassos-config` en el shell del SO. La memoria USB sólo necesita ser insertada en el dispositivo durante este proceso de configuración y puede ser retirada después.

Bootargs

Se puede editar o crear un `cmdline.txt` en la partición de arranque. Que será leído desde el gestor de arranque.

Kernel-Module

La carpeta de módulos del kernel `/etc/modules-load.d` es persistente y se puede añadir ahí archivos de configuración.

Udev rules

La carpeta de reglas udev `/etc/udev/rules.d` es persistente y se puede añadir archivos de configuración.

NTP

Se puede editar manualmente el archivo `systemd timesync` en `/etc/systemd/timesyncd.conf`.

La configuración NTP por defecto es la siguiente:

```
[Time]
NTP=time.cloudflare.com
FallbackNTP=0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org 3.pool.ntp.org
```

Network

Se puede añadir, editar o eliminar manualmente configuraciones de conexiones desde `/etc/NetworkManager/system-connections`.

El sistema operativo Home Assistant utiliza NetworkManager para controlar la red host.

Configurar Red

Por defecto el dispositivo estará en estado DHCP.

La configuración básica de red puede establecerse a través del frontend del Supervisor en la pestaña Sistema. Las configuraciones avanzadas como VLAN también están disponibles a través del comando `ha network CLI`.

Para restaurar la configuración por defecto también se puede utilizar el comando `ha network CLI`:

```
ha network update default --ipv4-method auto
```

Si se necesitan configuraciones de red más avanzadas, los archivos de conexión de red pueden colocarse en una unidad USB e importarse al host como se describe anteriormente.

Configuración de Red Manual

Si el frontend o ha network CLI no puede satisfacer algún caso de uso, todavía es posible configurar el NetworkManager subyacente manualmente.

Se puede leer el manual de NetworkManager o encontrar muchos ejemplos de configuración en internet. Hay que tener en cuenta que el sistema es de sólo lectura. Si no se quiere que la dirección IP cambie en cada arranque, se debe modificar la propiedad UUID a un UUID4 genérico.

Dentro del directorio `\CONFIG\network\` en la unidad USB o tarjeta SD, se puede crear un archivo llamado `my-network` y añadir el contenido apropiado a continuación:

Por defecto, está activo un perfil de conexión preinstalado para la red por cable:

```
[connection]
id=Home Assistant OS default
uuid=f62bf7c2-e565-49ff-bbfc-a4cf791e6add
type=802-3-ethernet
llmnr=2
mdns=2
```

```
[ipv4]
method=auto
```

```
[ipv6]
addr-gen-mode=stable-privacy
method=auto
```

Wired connection to the LAN

```
[connection]
id=my-network
uuid=d55162b4-6152-4310-9312-8f4c54d86afa
type=802-3-ethernet
llmnr=2
mdns=2
```

```
[ipv4]
method=auto
```

```
[ipv6]  
addr-gen-mode=stable-privacy  
method=auto
```

Wireless LAN WPA/PSK

```
[connection]  
id=my-network  
uuid=72111c67-4a5d-4d5c-925e-f8ee26efb3c3  
type=802-11-wireless
```

```
[802-11-wireless]  
mode=infrastructure  
ssid=MY_SSID  
# Uncomment below if your SSID is not broadcasted  
#hidden=true
```

```
[802-11-wireless-security]  
auth-alg=open  
key-mgmt=wpa-psk  
psk=MY_WLAN_SECRET_KEY
```

```
[ipv4]  
method=auto
```

```
[ipv6]  
addr-gen-mode=stable-privacy  
method=auto
```

Static IP

Reemplazar la siguiente configuración:

```
[ipv4]  
method=manual  
address=192.168.1.111/24;192.168.1.1  
dns=8.8.8.8;8.8.4.4;
```

Para *address*, el valor antes del punto y coma es la dirección IP y la longitud de bits del prefijo de subred. El segundo valor (después del punto y coma) es la dirección IP del gateway

Network Commands

Usando nmcli para establecer una dirección IPv4 estática

Iniciar sesión en el sistema base Home Assistant OS a través de una consola:

```
Welcome to Home Assistant  
homeassistant login:
```

- Iniciar sesión como root (no se necesita contraseña). En: ha > , escribir login.

A partir de ahí se utiliza la herramienta de configuración *nmcli*.

- # nmcli con show will list the "Home Assistant OS default" connection in use.
- # nmcli con show "Home Assistant OS default" will list all the properties of the connection.

Para empezar a editar los ajustes de configuración de "Home Assistant OS default":

```
# nmcli con edit "Home Assistant OS default"
```

Para añadir una dirección IP estática (seleccionar "yes" para el método manual)

```
nmcli> set ipv4.addresses 192.168.100.10/24
```

```
Do you also want to set 'ipv4.method' to 'manual'? [yes]:
```

Además, se recomienda configurar el servidor DNS y la puerta de enlace local. En la mayoría de los routers domésticos, el servidor DNS tendrá la misma dirección IP que el propio router:

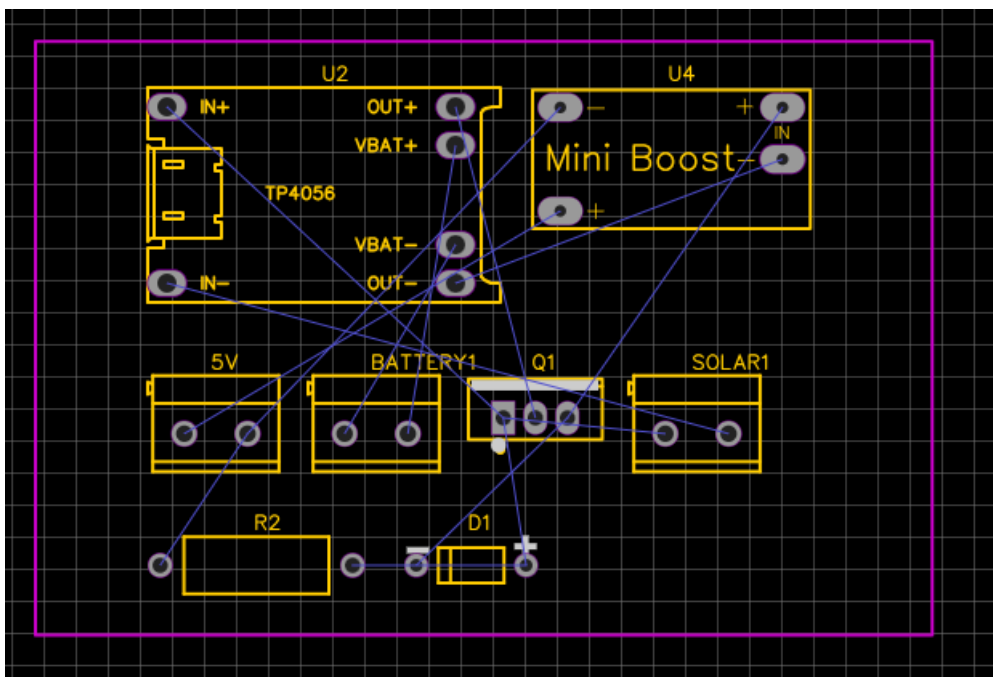
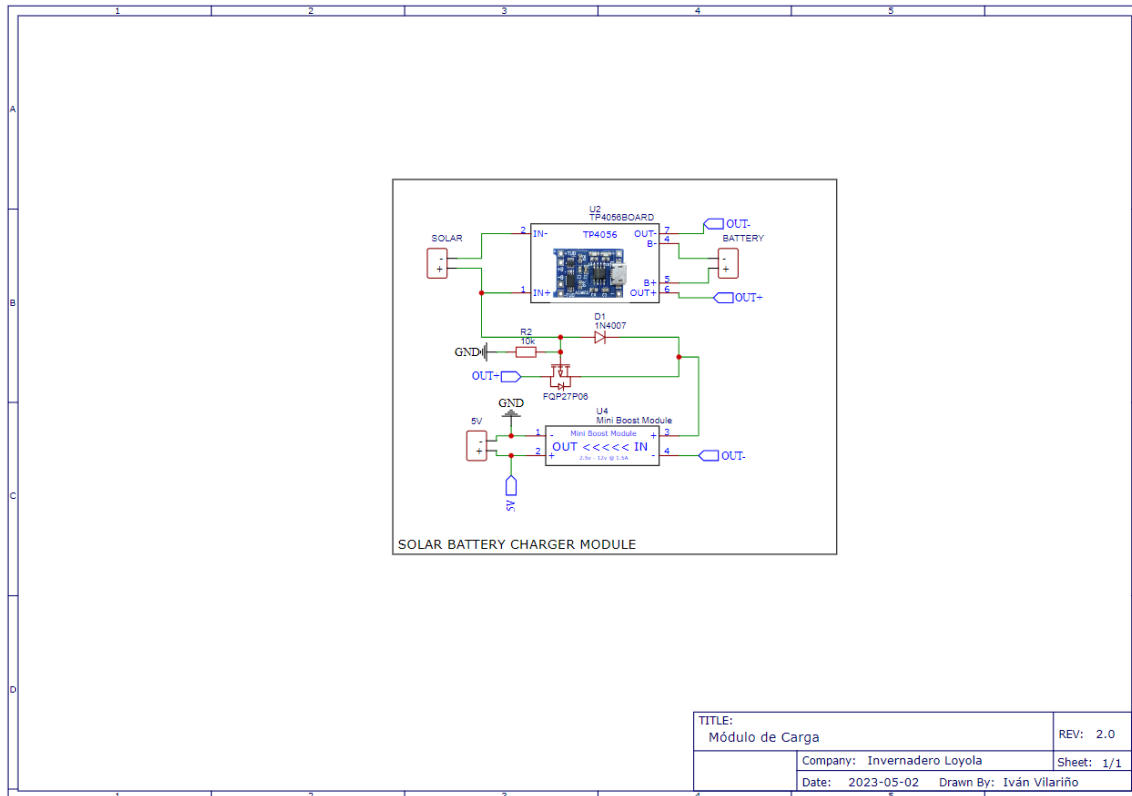
```
nmcli> set ipv4.dns 192.168.100.1
```

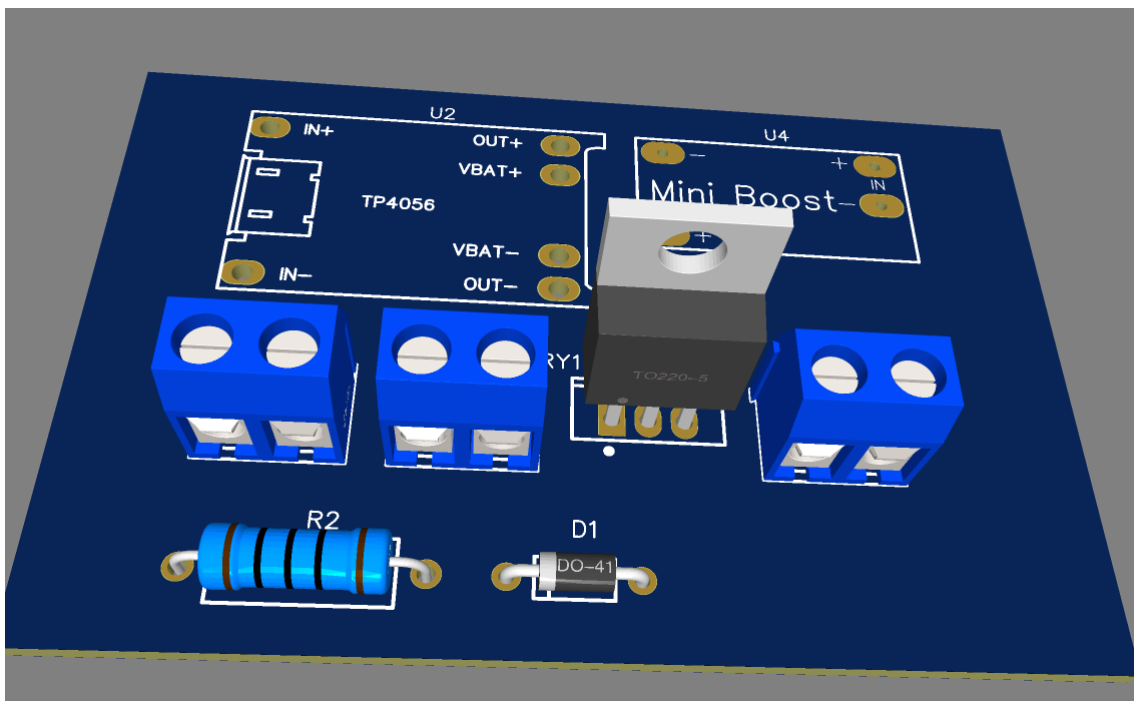
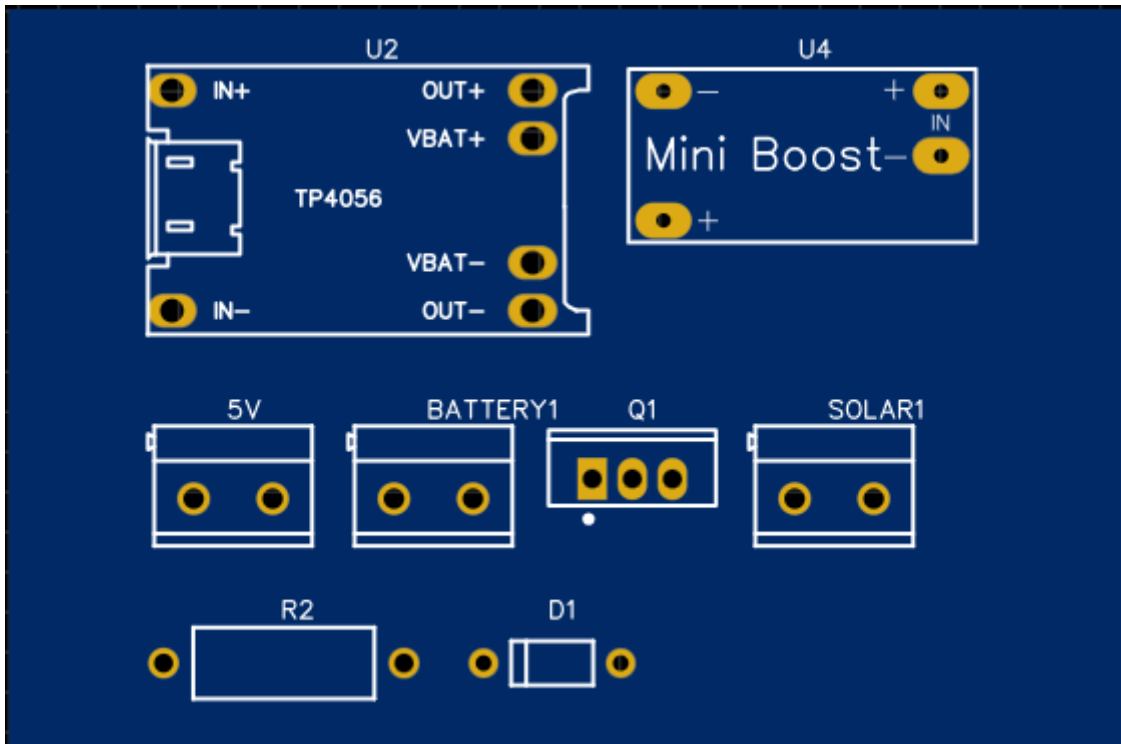
```
nmcli> set ipv4.gateway 192.168.100.1
```

nmcli> print ipv4 mostrará las propiedades IPv4 de esta conexión. Con nmcli> save se guardarán los cambios.

Más información en: <https://github.com/home-assistant/operating-system/tree/dev/Documentation>

ANEXO II – Schematics y Diagramas





ANEXO III - Códigos

Código correspondiente al Nodo 1 (Arduino)

```
#include <WiFiNINA.h>
#include <SPI.h>
#include <I2CSoilMoistureSensor.h>
#include <Wire.h>
#include <cstdio>
#include <Adafruit_Sensor.h>
#include "DHT.h"
#include <ArduinoHA.h>

WiFiClient client;
HADevice device;
HAMqtt mqtt(client, device);

//HASwitch led("led");
HASensorNumber analogSensor("SensorHum", HASensorNumber::PrecisionP1);
//HASensorNumber analogSensor2("SensorTem", HASensorNumber::PrecisionP1);

HASensorNumber analogSensor3("SensorTemDHT", HASensorNumber::PrecisionP1);
HASensorNumber analogSensor4("SensorHumDHT", HASensorNumber::PrecisionP1);

HAFan fan("ventilation");

HASwitch led("pump");

unsigned long lastUpdateAt = 0;

I2CSoilMoistureSensor sensor(0x21);

// char ssid[] = "W_SalaO"; // your network SSID (name)
// char pass[] = "xxxxx"; // your network password (use for WPA, or use as key for WEP)

char ssid[] = "invernaderoLOYOLA"; // your network SSID (name)
char pass[] = "xxxxxx"; // your network password (use for WPA, or use as key for WEP)

int status = WL_IDLE_STATUS; // the Wifi radio's status

int capacitancia = 0;
float humidity = 0.0;

//Sensor Humedad
#define DHTPIN 2
```

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

void printWifiData();
void printCurrentNet();
void printMacAddress(byte mac[]);

void onStateCommand(bool state2, HAFan* sender) {
    digitalWrite(1, (state2 ? HIGH : LOW));
    Serial.print("State: ");
    Serial.println(state2);

    sender->setState(state2); // report state back to the Home Assistant
}

void onSwitchCommand(bool state, HASwitch* sender)
{
    digitalWrite(3, (state ? HIGH : LOW));
    sender->setState(state); // report state back to the Home Assistant
}

void setup() {

    byte mac[6];
    WiFi.macAddress(mac);
    device.setUniqueId(mac, sizeof(mac));

    Wire.begin();

    pinMode(3, OUTPUT);
    digitalWrite(3, LOW);

    pinMode(1, OUTPUT);
    digitalWrite(1, LOW);

    pinMode(2, INPUT);

    // set device's details (optional)
    device.setName("Arduino");
    device.setSoftwareVersion("1.0.0");

    // set icon (optional)
    led.setIcon("mdi:lightbulb");
    led.setName("Pump");
```

```
// handle switch state
led.onCommand(onSwitchCommand);

// configure sensor (optional)
analogSensor.setICon("mdi:ink-percent");
analogSensor.setName("Humidity");
analogSensor.setUnitOfMeasurement("%");

//analogSensor2.setICon("mdi:ink-percent");
//analogSensor2.setName("Temperature");
//analogSensor2.setUnitOfMeasurement("°C");

analogSensor3.setICon("mdi:ink-percent");
analogSensor3.setName("Air Temperature");
analogSensor3.setUnitOfMeasurement("°C");

analogSensor4.setICon("mdi:ink-percent");
analogSensor4.setName("Air Humidity");
analogSensor4.setUnitOfMeasurement("%");

// configure fan (optional)
fan.setName("Fan");
fan.onStateCommand(onStateCommand);

//Initialize serial and wait for port to open:

Serial.begin(9600);

// check for the WiFi module:

if (WiFi.status() == WL_NO_MODULE) {

  Serial.println("Communication with WiFi module failed!");

  // don't continue

  while (true);

}

String fv = WiFi.firmwareVersion();

if (fv < WIFI_FIRMWARE_LATEST_VERSION) {

  Serial.println("Please upgrade the firmware");
```

```
}

// attempt to connect to Wifi network:

while (status != WL_CONNECTED) {

  Serial.print("Attempting to connect to WPA SSID: ");

  Serial.println(ssid);

  // Connect to WPA/WPA2 network:

  status = WiFi.begin(ssid, pass);
  Serial.print(status);

  // wait 10 seconds for connection:

  delay(10000);

}

// you're connected now, so print out the data:

Serial.print("You're connected to the network");

printCurrentNet();

printWifiData();

//Moisture Sensor Startup
sensor.begin(); // reset sensor
delay(1000); // give some time to boot up
Serial.print("I2C Soil Moisture Sensor Address: ");
Serial.println(sensor.getAddress(),HEX);
Serial.print("Sensor Firmware version: ");
Serial.println(sensor.getVersion(),HEX);
Serial.println();

//Temperature Sensor Startup
dht.begin();
Serial.println(F("DHTxx Unified Sensor Example"));

mqtt.begin("192.168.0.177", "homeassistant",
"aiB0tieM5shiez0Niephus1oSh4ooseph7ieDoogaed3ib2ieghei5eikiliexei");
```

```
// mqtt.begin("invernaderoloyola.tk", "homeassistant",
"aiB0tieM5shiez0Niephus1oSh4ooseph7ieDoogaed3ib2ieghei5eikiliexei");
Serial.println(F("Nice!!"));

}

void loop() {
  mqtt.loop();

  if (mqtt.isConnected() == false){
    NVIC_SystemReset();
  }

  if ((millis() - lastUpdateAt) > 30000) { // 1000ms debounce time

    capacitancia = sensor.getCapacitance();
    humidity = (100.0 / 310.0) * capacitancia + (100.0 - (100.0 / 310.0) * 560.0);

    // float temperatura = sensor.getTemperature()/(float)10;

    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Compute heat index in Celsius (isFahreheit = false)
    //float hic = dht.computeHeatIndex(t, h, false);

    analogSensor.setValue(humidity);
    //analogSensor2.setValue(temperatura);
    analogSensor3.setValue(t);
    analogSensor4.setValue(h);

    lastUpdateAt = millis();

    Serial.print("Soil Moisture Capacitance: ");
    Serial.print(capacitancia); //read capacitance register

    Serial.print(", Humidity (%): ");
    Serial.print(humidity); //show humidity
    Serial.println();

    Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("% Temperature: "));
    Serial.print(t);
    Serial.print(F("°C "));
    Serial.println();
  }
}
```

```
}  
}  
  
void printWifiData() {  
  
    // print your board's IP address:  
  
    IPAddress ip = WiFi.localIP();  
  
    Serial.print("IP Address: ");  
  
    Serial.println(ip);  
  
    Serial.println(ip);  
  
    // print your MAC address:  
  
    byte mac[6];  
  
    WiFi.macAddress(mac);  
  
    Serial.print("MAC address: ");  
  
    printMacAddress(mac);  
}  
  
void printCurrentNet() {  
  
    // print the SSID of the network you're attached to:  
  
    Serial.print("SSID: ");  
  
    Serial.println(WiFi.SSID());  
  
    // print the MAC address of the router you're attached to:  
  
    byte bssid[6];  
  
    WiFi.BSSID(bssid);  
  
    Serial.print("BSSID: ");  
  
    printMacAddress(bssid);  
  
    // print the received signal strength:
```

```
long rssi = WiFi.RSSI();

Serial.print("signal strength (RSSI):");

Serial.println(rssi);

// print the encryption type:

byte encryption = WiFi.encryptionType();

Serial.print("Encryption Type:");

Serial.println(encryption, HEX);

Serial.println();
}

void printMacAddress(byte mac[]) {

for (int i = 5; i >= 0; i--) {

    if (mac[i] < 16) {

        Serial.print("0");

    }

    Serial.print(mac[i], HEX);

    if (i > 0) {

        Serial.print(":");

    }

}

Serial.println();
}
```

Código correspondiente al Nodo 4 (Emisor LoRa):

```
// include the library
```

```

#include <Arduino.h>
#include <SPI.h>
#include <Wire.h>
#include <RadioLib.h>

// SX1262 has the following connections:
// NSS pin: 10
// DIO1 pin: 2
// NRST pin: 3
// BUSY pin: 9

#define LoRa_MOSI 10
#define LoRa_MISO 11
#define LoRa_SCK 9

#define LoRa_nss 8
#define LoRa_dio1 14
#define LoRa_nrst 12
#define LoRa_busy 13

SX1262 radio = new Module(LoRa_nss, LoRa_dio1, LoRa_nrst, LoRa_busy);

void setFlag();

int transmissionState = RADIOLIB_ERR_NONE;
float valor = 0.0;
int pin = 19;
float bateria = 0.0;

void setup() {
  Serial.begin(9600);

  // initialize SX1262 with default settings
  Serial.print(F("[SX1262] Initializing ... "));
  int state = radio.begin();
  if (state == RADIOLIB_ERR_NONE) {
    Serial.println(F("success!"));
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }

  // set the function that will be called
  // when packet transmission is finished
  radio.setPacketSentAction(setFlag);

```



```
// start transmitting the first packet
Serial.print(F("[SX1262] Sending first packet ... "));

// you can transmit C-string or Arduino string up to
// 256 characters long
transmissionState = radio.startTransmit("Hello World!");

}

// flag to indicate that a packet was sent
volatile bool transmittedFlag = false;

// this function is called when a complete packet
// is transmitted by the module

#ifdef ESP8266 || defined(ESP32)
  ICACHE_RAM_ATTR
#endif
void setFlag(void) {
  // we sent a packet, set the flag
  transmittedFlag = true;
}

// counter to keep track of transmitted packets
int count = 0;

void loop() {
  // check if the previous transmission finished
  if(transmittedFlag) {
    // reset flag
    transmittedFlag = false;

    if (transmissionState == RADIOLIB_ERR_NONE) {
      // packet was successfully sent
      Serial.println(F("transmission finished!"));

    } else {
      Serial.print(F("failed, code "));
      Serial.println(transmissionState);

    }

  }

  // clean up after transmission is finished
  // this will ensure transmitter is disabled,
  // RF switch is powered down etc.
  radio.finishTransmit();
}
```

```
// wait 30 seconds before transmitting again
delay(30000);

// send another one
Serial.print(F("[SX1262] Sending another packet ... "));

// you can transmit C-string or Arduino string up to
// 256 characters long
valor = analogRead(pin);
bateria = (100.0 * ((valor * 3.144)/4095)) / 3.228;

String str = String(bateria);
Serial.println(valor);
Serial.println(str);
transmissionState = radio.startTransmit(str);

}
}
```

Código correspondiente al Nodo 3 (Receptor LoRa / Gateway):

```
// include the library
```

```

#include <Arduino.h>
#include <SPI.h>
#include <Wire.h>
#include <RadioLib.h>
#include <ArduinoHA.h>
#include <WiFi.h>

#define LED_PIN      D0
#define BROKER_ADDR  IPAddress(192,168,0,177)
#define WIFI_SSID    "invernaderoloyola"
#define WIFI_PASSWORD "lvM5653560!"

WiFiClient client;
HADevice device;
HAMqtt mqtt(client, device);

// "bateria" is unique ID of the switch. You should define your own ID.
HASensorNumber analogSensor("BatteryLevel", HASensorNumber::PrecisionP1);

// SX1278 has the following connections:
// NSS pin: 10
// DIO0 pin: 2
// RESET pin: 9
// DIO1 pin: 3
SX1276 radio = new Module(18, 26, 14, 35);

void setFlag();

void setup() {
  Serial.begin(9600);

  byte mac[6];
  WiFi.macAddress(mac);
  device.setUniqueId(mac, sizeof(mac));

  // initialize SX1278 with default settings
  Serial.print(F("[SX1278] Initializing ... "));
  int state = radio.begin();
  if (state == RADIOLIB_ERR_NONE) {
    Serial.println(F("success!"));
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }
}

```

```

// set the function that will be called
// when new packet is received
radio.setPacketReceivedAction(setFlag);

// connect to wifi
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500); // waiting for the connection
}
Serial.println();
Serial.println("Connected to the network");

// set device's details (optional)
device.setName("OutsideStation");
device.setSoftwareVersion("1.0.0");

// configure sensor (optional)
analogSensor.setICon("mdi:battery");
analogSensor.setName("Station Battery");
analogSensor.setUnitOfMeasurement("%");

mqtt.begin("192.168.0.177", "homeassistant",
"aiB0tieM5shiez0Niephus1oSh4ooseph7ieDoogaed3ib2ieghei5eikiliexei");

// start listening for LoRa packets
Serial.print(F("[SX1278] Starting to listen ... "));
state = radio.startReceive();
if (state == RADIOLIB_ERR_NONE) {
  Serial.println(F("success!"));
} else {
  Serial.print(F("failed, code "));
  Serial.println(state);
  while (true);
}
}

// flag to indicate that a packet was received
volatile bool receivedFlag = false;

// this function is called when a complete packet
// is received by the module
#ifdef ESP8266 || defined(ESP32)
  ICACHE_RAM_ATTR
#endif
#endif

```

```
void setFlag(void) {
  // we got a packet, set the flag
  receivedFlag = true;
}

void loop() {
  mqtt.loop();

  if (mqtt.isConnected() == false){
    Serial.println("NO!");
    ESP.restart();
  }

  // check if the flag is set
  if(receivedFlag) {
    // reset flag
    receivedFlag = false;

    String str;
    int state = radio.readData(str);

    if (state == RADIOLIB_ERR_NONE) {
      Serial.println(str);

      float battery = str.toFloat();
      Serial.println(battery);

      analogSensor.setValue(battery);

    } else if (state == RADIOLIB_ERR_CRC_MISMATCH) {
      // packet was received, but is malformed
      Serial.println(F("[SX1278] CRC error!"));

    } else {
      // some other error occurred
      Serial.print(F("[SX1278] Failed, code "));
      Serial.println(state);

    }
  }
}
}
```

Código correspondiente a la función AWS Lambda:

.....

Copyright 2019 Jason Hu <awaregit at gmail.com>

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

```
"""
```

```
import os
```

```
import json
```

```
import logging
```

```
import urllib3
```

```
_debug = bool(os.environ.get('DEBUG'))
```

```
_logger = logging.getLogger('HomeAssistant-SmartHome')
```

```
_logger.setLevel(logging.DEBUG if _debug else logging.INFO)
```

```
def lambda_handler(event, context):
```

```
    """Handle incoming Alexa directive."""
```

```
    _logger.debug('Event: %s', event)
```

```
    base_url = os.environ.get('BASE_URL')
```

```
    assert base_url is not None, 'Please set BASE_URL environment variable'
```

```
    base_url = base_url.strip("/")
```

```
    directive = event.get('directive')
```

```
    assert directive is not None, 'Malformatted request - missing directive'
```

```
    assert directive.get('header', {}).get('payloadVersion') == '3', \
```

```
        'Only support payloadVersion == 3'
```

```
    scope = directive.get('endpoint', {}).get('scope')
```

```
    if scope is None:
```

```
        # token is in grantee for Linking directive
```

```
        scope = directive.get('payload', {}).get('grantee')
```

```
    if scope is None:
```

```
        # token is in payload for Discovery directive
```

```
        scope = directive.get('payload', {}).get('scope')
```

```
    assert scope is not None, 'Malformatted request - missing endpoint.scope'
```

```
    assert scope.get('type') == 'BearerToken', 'Only support BearerToken'
```

```
    token = scope.get('token')
```

```

if token is None and _debug:
    token = os.environ.get('LONG_LIVED_ACCESS_TOKEN') # only for debug purpose

verify_ssl = not bool(os.environ.get('NOT_VERIFY_SSL'))

http = urllib3.PoolManager(
    cert_reqs='CERT_REQUIRED' if verify_ssl else 'CERT_NONE',
    timeout=urllib3.Timeout(connect=2.0, read=10.0)
)

response = http.request(
    'POST',
    '{}/api/alexa/smart_home'.format(base_url),
    headers={
        'Authorization': 'Bearer {}'.format(token),
        'Content-Type': 'application/json',
    },
    body=json.dumps(event).encode('utf-8'),
)

if response.status >= 400:
    return {
        'event': {
            'payload': {
                'type': 'INVALID_AUTHORIZATION_CREDENTIAL'
                if response.status in (401, 403) else 'INTERNAL_ERROR',
                'message': response.data.decode("utf-8"),
            }
        }
    }

_logger.debug('Response: %s', response.data.decode("utf-8"))
return json.loads(response.data.decode('utf-8'))

```