



**Universidad de Jaén**

*Escuela Politécnica Superior de Jaén*

# **DISEÑO Y DESARROLLO HARDWARE Y SOFTWARE DE UN EQUIPO DE EXPERIMENTACIÓN PARA EL CULTIVO HIDROPÓNICO DE CIERTAS ESPECIES DE PLANTAS**

**Autor: David Jiménez Ramírez**

**Fecha: 27/06/2024**

Licencia CC



CREA





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Don ALEJANDRO SÁNCHEZ GARCÍA y SERGIO ILLANA RICO, tutor y cotutor del Proyecto Fin de Carrera titulado: DISEÑO Y DESARROLLO HARDWARE Y SOFTWARE DE UN EQUIPO DE EXPERIMENTACIÓN PARA EL CULTIVO HIDROPÓNICO DE CIERTAS ESPECIES DE PLANTAS, que presenta DAVID JIMÉNEZ RAMÍREZ, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, JUNIO de 2024

El alumno:

Los tutores:

DAVID JIMÉNEZ RAMÍREZ

ALEJANDRO GARCÍA SANCHEZ

SERGIO ILLANA RICO

## **AGRADECIMIENTOS**

Quisiera expresar mi más profundo agradecimiento a aquellas personas que han sido fundamentales en la realización de este trabajo. En primer lugar, a mi madre, por su amor incondicional y su constante apoyo en cada etapa de mi vida. Su fortaleza y dedicación han sido una fuente de inspiración constante.

A mi hermano, quien siempre ha estado a mi lado, brindándome su apoyo. Su confianza en mis capacidades y sus palabras de ánimo han sido esenciales para mantenerme motivado.

A Elena, por su apoyo y compañía. Su comprensión y amor han sido un pilar fundamental durante este tiempo.

Finalmente, a mis tutores, cuyos conocimientos, orientación y apoyo académico han sido cruciales para la culminación de este proyecto. Su compromiso y dedicación para guiarme a través de este proceso han sido invaluable, y su influencia perdurará más allá de este trabajo.

## Índice General

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| 1. ÍNDICE.....                                                                                 | 10 |
| 1.1. RESUMEN .....                                                                             | 10 |
| 1.2. ABSTRACT .....                                                                            | 10 |
| 1.3. CONTEXTO Y JUSTIFICACIÓN DEL PROYECTO.....                                                | 12 |
| 1.4. REVISIÓN DEL ESTADO DEL ARTE .....                                                        | 13 |
| 1.4.1. CULTIVO HIDROPÓNICO: FUNDAMENTOS Y APLICACIONES.....                                    | 13 |
| 1.4.2. EQUIPOS DE EXPERIMENTACIÓN EXISTENTES.....                                              | 14 |
| 1.4.3. TECNOLOGÍAS HARDWARE Y SOFTWARE APLICADAS AL CULTIVO<br>HIDROPÓNICO DE COMPETENCIA..... | 15 |
| 1.5. OBJETIVOS .....                                                                           | 16 |
| 2. DEFINICIÓN DE REQUISITOS.....                                                               | 18 |
| 2.1. REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL EQUIPO DE<br>EXPERIMENTACIÓN .....            | 18 |
| 3. IDENTIFICACIÓN DE COMPONENTES HARDWARE Y SOFTWARE NECESARIOS ..                             | 19 |
| 3.1. COMPONENTES UTILIZADOS EN LA PARTE HARDWARE .....                                         | 19 |
| 3.2. COMPONENTES UTILIZADOS EN LA PARTE SOFTWARE .....                                         | 26 |
| 4. DISEÑO DEL SISTEMA.....                                                                     | 31 |
| 4.1. DISEÑO HARDWARE.....                                                                      | 31 |
| 4.1.1. SENSORES Y ACTUADORES .....                                                             | 33 |
| 4.1.2. CONTROLADORES Y ESTRUCTURA .....                                                        | 34 |
| 4.2. DISEÑO DEL SOFTWARE .....                                                                 | 34 |
| 5. DESARROLLO DEL HARDWARE .....                                                               | 36 |
| 5.1. ADQUISICIÓN DE COMPONENTES.....                                                           | 36 |
| 5.2. DISEÑO E IMPRESIÓN DE ACOPLER 3D .....                                                    | 36 |
| 5.3. PROTOTIPADO DEL EQUIPO DE EXPERIMENTACIÓN.....                                            | 40 |
| 5.4. PROCESO DE ENSAMBLAJE.....                                                                | 40 |
| 5.5. PRUEBAS Y AJUSTES DEL HARDWARE .....                                                      | 41 |
| 5.5.1. PRUEBAS INICIALES.....                                                                  | 41 |
| 5.5.2. AJUSTES Y OPTIMIZACIÓN .....                                                            | 41 |
| 5.5.3. PRUEBAS CONTINUAS .....                                                                 | 42 |
| 6. DESARROLLO DEL SOFTWARE.....                                                                | 43 |
| 6.1. IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL .....                                              | 43 |
| 6.1.1. CONFIGURACIÓN INICIAL .....                                                             | 44 |

|         |                                                                              |    |
|---------|------------------------------------------------------------------------------|----|
| 6.1.2.  | CONFIGURACIÓN DE PINES .....                                                 | 44 |
| 6.1.3.  | BLOQUE MACRO CONEXIÓN WI-FI Y MQTT .....                                     | 46 |
| 6.1.4.  | BLOQUE MACRO LECTURA DE SENSORES Y CONTROL DE<br>ACTUADORES: .....           | 48 |
| 6.1.5.  | BLOQUE MACRO CONTROL DE ACTUADORES .....                                     | 49 |
| 6.2.    | DESARROLLO DE LA INTERFAZ DE USUARIO UTILIZANDO NODE-RED.....                | 51 |
| 6.3.    | EXPLICACIÓN DE LOS FLUJOS EN NODE-RED .....                                  | 59 |
| 6.3.1.  | FLUJO DE SENSORES .....                                                      | 60 |
| 6.3.2.  | FLUJO ACTUADORES .....                                                       | 63 |
| 6.3.3.  | FLUJO IMÁGENES POR MQTT .....                                                | 65 |
| 6.3.4.  | FLUJO WEBCAM.....                                                            | 66 |
| 6.3.5.  | FLUJO ÚLTIMA IMAGEN.....                                                     | 67 |
| 6.3.6.  | FLUJO ENVIAR TÓPICO ESPEFÍCO CON MENSAJE ESPEFÍCO.....                       | 67 |
| 6.3.7.  | FLUJO GRÁFICA TIEMPO EN FORMATO CHART DE DATOS DE<br>SENSORES .....          | 68 |
| 7.      | INTEGRACIÓN Y PRUEBAS .....                                                  | 69 |
| 7.1.    | INTEGRACIÓN DEL HARDWARE Y SOFTWARE.....                                     | 69 |
| 7.2.    | PRUEBAS FUNCIONALES Y DE RENDIMIENTO EN DIFERENTES<br>CONDICIONES .....      | 70 |
| 8.      | CONCLUSIONES.....                                                            | 72 |
| 9.      | REFERENCIAS.....                                                             | 74 |
| 10.     | ANEXOS.....                                                                  | 75 |
| 10.1.   | INSTALACIÓN NODE-RED .....                                                   | 75 |
| 10.2.   | INSTALACIÓN INFLUXDB .....                                                   | 76 |
| 10.3.   | INSTALACIÓN DEL BROKER MQTT SHIFTR.IO.....                                   | 76 |
| 10.4.   | INCLUSIÓN DE DATOS WIFI, BROKER Y DIRECTORIO DE<br>ALMACENAMIENTO LOCAL..... | 77 |
| 10.5.   | CÓDIGO PROGRAMACIÓN ESP32-CAM EN ENTORNO IDE ARDUINO .....                   | 77 |
| 10.5.1. | MAIN PRINCIPAL .....                                                         | 77 |
| 10.5.2. | INCLUSIÓN DE LIBRERÍAS .....                                                 | 78 |
| 10.6.   | CÓDIGO PROGRAMACIÓN DE NODE-RED EN JSON.....                                 | 78 |



## Índice de Figuras

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Fig 1. Sistema hidropónico .....                                | 12 |
| Fig 2. Tipos de cultivo Hidropónico.....                        | 14 |
| Fig 3. Módulo ESP32-CAM .....                                   | 19 |
| Fig 4. Programador ESP32-CAM .....                              | 20 |
| Fig 5. Sensor DTH11.....                                        | 21 |
| Fig 6. Tira LED (5V) .....                                      | 22 |
| Fig 7. Ventilador DC (5V) .....                                 | 23 |
| Fig 8. Pack 4 Relés .....                                       | 24 |
| Fig 9. Placa de Extensión de Pines .....                        | 25 |
| Fig 10. Batería portátil .....                                  | 25 |
| Fig 11. Protocolo MQTT .....                                    | 28 |
| Fig 12. Ejemplo de funcionamiento de shiftr.io .....            | 30 |
| Fig 13. Prototipo final .....                                   | 31 |
| Fig 14. Bandeja Hidropónica .....                               | 32 |
| Fig 15. Módulo de cámara OV2640.....                            | 33 |
| Fig 16. Diagrama interconexión del sistema.....                 | 35 |
| Fig 17. Soporte principal.....                                  | 37 |
| Fig 18. Soportes laterales.....                                 | 38 |
| Fig 19. Soporte ESP32-CAM y conjunto de relés .....             | 38 |
| Fig 20. Tapadera sujeta componentes .....                       | 39 |
| Fig 21. Soporte ESP32-CAM con bandeja.....                      | 39 |
| Fig 22. ESP32-CAM con Soporte .....                             | 39 |
| Fig 23. Diagrama de flujo de código ESP32-CAM .....             | 43 |
| Fig 24. Flujograma Conexión WIFI .....                          | 46 |
| Fig 25. Flujograma Callback.....                                | 47 |
| Fig 26. Lectura de Sensores y actuadores .....                  | 49 |
| Fig 27. Control independiente de Actuadores.....                | 50 |
| Fig 28. Interfaz usuario. "Datos Actuales".....                 | 52 |
| Fig 29. Selección de Dispositivo.....                           | 52 |
| Fig 30. Cámara en vivo .....                                    | 53 |
| Fig 31. Temperatura y Humedad última .....                      | 53 |
| Fig 32. Actuadores .....                                        | 54 |
| Fig 33. Interfaz usuario. "Historial de Datos" .....            | 54 |
| Fig 34. Almacenamiento Imágenes y selección de dispositivo..... | 55 |
| Fig 35. Gráficos Humedad y Temperatura frente a tiempo .....    | 55 |
| Fig 36. Interfaz de Usuario 'Opciones Avanzadas' .....          | 56 |
| Fig 37. Dirección MAC personalizable .....                      | 57 |
| Fig 38. Directorio de Imágenes.....                             | 57 |
| Fig 39. Historial específico de Temperatura y Humedad .....     | 58 |
| Fig 40. Node-Red. Flujo de Sensores.....                        | 60 |
| Fig 41. Nodo MQTT input.....                                    | 61 |
| Fig 42. Nodo Funcion .....                                      | 61 |
| Fig 43. Nodo Gauge .....                                        | 61 |
| Fig 44. Nodo Text Input.....                                    | 62 |
| Fig 45. Nodo Button .....                                       | 62 |
| Fig 46. Node-Red. Flujo de Actuadores.....                      | 63 |

|                                                           |    |
|-----------------------------------------------------------|----|
| Fig 47. Nodo MQTT output.....                             | 64 |
| Fig 48. Nodo Switch .....                                 | 64 |
| Fig 49. Node-Red. Flujo de Imágenes .....                 | 65 |
| Fig 50. Node-Red. Flujo de Webcam.....                    | 66 |
| Fig 51. Node-Red. Banco de Imágenes.....                  | 67 |
| Fig 52. Node-Red. Envío Tópico y mensaje Específico ..... | 67 |
| Fig 53. Node-Red. Inclusión de Gráfico por tiempo .....   | 68 |
| Fig 54. Datos .txt almacenado en microSD.....             | 77 |

## Índice de Tablas

|                                                  |    |
|--------------------------------------------------|----|
| Tabla 1. ESP32-CAM .....                         | 20 |
| Tabla 2. Características DHT11 .....             | 21 |
| Tabla 3. Características Tira LED.....           | 22 |
| Tabla 4. Características Ventilador DC (5V)..... | 23 |
| Tabla 5. Características Módulo Relés .....      | 24 |
| Tabla 6. Características Batería Portatil .....  | 26 |

## **1. ÍNDICE**

### **1.1. RESUMEN**

Este proyecto se centra en el monitoreo y control automatizado de un invernadero para optimizar el cultivo de plantas. Utilizando tecnologías del Internet de las Cosas (IoT) y sistemas de sensores, se recopilan datos en tiempo real sobre variables ambientales clave, como temperatura, humedad y luminosidad. Estos datos son procesados y analizados para proporcionar información útil sobre las condiciones del invernadero.

El sistema también incluye un sistema de control automatizado que puede ajustar variables ambientales según las necesidades específicas de las plantas en diferentes etapas de crecimiento. Esto se logra mediante el uso de actuadores conectados que pueden regular la luminosidad, la humedad y otros parámetros ambientales.

Además, se aborda la inclusión del uso de recursos como el monitoreo con cámara, tanto visión en vivo como almacenamiento de imágenes, mediante la implementación de la base de datos InfluxDB, con el objetivo de proporcionar un lugar útil, eficiente, con posibilidad de añadir periféricos de datos y extrapolable a otros sistemas para reducir el consumo de recursos.

En resumen, se propone un enfoque integral para el monitoreo y control de invernaderos, utilizando tecnologías avanzadas para mejorar la eficiencia y la sostenibilidad en la agricultura.

### **1.2. ABSTRACT**

The project focuses on the automated monitoring and control of a greenhouse to optimize plant cultivation. Using Internet of Things (IoT) technologies and sensor systems, real-time data is collected on key environmental variables, such as

temperature, humidity and luminosity. This data is processed and analyzed to provide useful information about greenhouse conditions.

The system also includes an automated control system that can adjust environmental variables according to the specific needs of the plants at different stages of growth. This is achieved through the use of connected actuators that can regulate illuminance, humidity and other environmental parameters.

In addition, the thesis addresses the inclusion of the use of resources such as camera monitoring, both live vision and image storage, through the implementation of the InfluxDB database, it is sought, to have a useful and extrapolable place to other systems to reduce resource consumption.

In summary, the thesis proposes a comprehensive approach for monitoring and control of greenhouses, using advanced technologies to improve efficiency and sustainability in agriculture.

### 1.3.CONTEXTO Y JUSTIFICACIÓN DEL PROYECTO

El cultivo hidropónico, una técnica agrícola que permite el crecimiento de plantas sin necesidad de suelo, utilizando soluciones minerales en un solvente acuoso, ha ganado considerable atención en los últimos años debido a sus numerosas ventajas [1]. Entre estas ventajas se encuentran el **uso eficiente del agua**, la **reducción de plagas y enfermedades del suelo**, y la posibilidad de **cultivar en áreas urbanas** o en **suelos infértiles**. En la Fig 1, se puede observar un ejemplo de un sistema hidropónico utilizado en la producción de un suelo no fértil. Sin embargo, la gestión de un sistema hidropónico puede ser compleja, ya que requiere un control preciso de variables ambientales como la temperatura, la humedad y la iluminación. [2]



Fig 1. Sistema hidropónico

La automatización de estos sistemas mediante el uso de tecnologías avanzadas ofrece una solución prometedora para simplificar el proceso y mejorar la eficiencia y productividad del cultivo. [2] En este contexto, la combinación de microcontroladores, sensores y actuadores, junto con plataformas de software para el monitoreo y control, se presenta como una herramienta poderosa para optimizar el cultivo hidropónico.

Este proyecto se centra en el diseño y desarrollo de un sistema automatizado de experimentación para el cultivo hidropónico utilizando una ESP32-CAM, un sensor DHT11, una tira LED y un ventilador, todo controlado mediante programación en el entorno Arduino. La integración de estas tecnologías permitirá un monitoreo constante y un control dinámico de las condiciones del cultivo, mejorando así la viabilidad y eficiencia del sistema.

## 1.4. REVISIÓN DEL ESTADO DEL ARTE

### 1.4.1. CULTIVO HIDROPÓNICO: FUNDAMENTOS Y APLICACIONES

El cultivo hidropónico es una técnica de cultivo que no utiliza suelo. En lugar de eso, las raíces de las plantas reciben una solución nutritiva equilibrada disuelta en agua, la cual contiene todos los elementos químicos esenciales para su desarrollo.

Existen varios tipos de sistemas hidropónicos, cada uno con sus propias características como se puede apreciar en la Fig 2. El sistema de **flujo y refluo** consiste en inundar periódicamente la bandeja de cultivo con una solución nutritiva, la cual luego se drena de vuelta al depósito. En el sistema de **aeroponía**, las raíces de las plantas están suspendidas en el aire y se rocían con una solución nutritiva para proporcionarles los nutrientes necesarios. Por otro lado, en el sistema de **raíz flotante**, las plantas flotan en una solución nutritiva con sus raíces sumergidas directamente en el agua rica en nutrientes. El sistema de **goteo** aplica la solución nutritiva lentamente sobre la base de cada planta, proporcionando nutrientes directamente a las raíces a través de un sistema de goteo controlado. En el método de **mecha**, las plantas absorben los nutrientes a través de una mecha que está sumergida en la solución nutritiva, permitiendo una absorción pasiva de nutrientes y agua. Finalmente, en el sistema **NFT (Nutrient Film Technique)**, las raíces de las plantas están en un canal poco profundo donde fluye constantemente una delgada película de solución nutritiva, proporcionando nutrientes y oxígeno de manera continua.



Fig 2. Tipos de cultivo Hidropónico

#### 1.4.2. EQUIPOS DE EXPERIMENTACIÓN EXISTENTES

Algunas de las soluciones existentes en el mercado son la **agricultura urbana** consiste en el cultivo de vegetales en zonas urbanas donde el espacio es limitado o el suelo está contaminado. Este método permite a las personas en las ciudades cultivar sus propios alimentos frescos y saludables.

Por otro lado, en el ámbito de la **investigación científica**, se realizan estudios enfocados en el crecimiento de plantas y la optimización de los nutrientes que estas requieren. Estos estudios son esenciales para mejorar las técnicas de cultivo y maximizar los rendimientos.

Además, la agricultura hidropónica se utiliza en el **espacio** y en **ambientes extremos**. La NASA, por ejemplo, lleva a cabo investigaciones sobre el crecimiento de plantas en el espacio y en condiciones extremas de la Tierra, buscando soluciones innovadoras para la alimentación en futuras misiones espaciales y en regiones inhóspitas de nuestro planeta.

Algunos de los equipos comerciales más destacados existentes son los siguientes:

- **AeroGarden:** Un sistema hidropónico de interior que utiliza tecnología aeropónica para cultivar hierbas y vegetales en casa.
- **Hydroponic Systems:** Variedad de sistemas hidropónicos comerciales, desde pequeños kits de inicio hasta grandes instalaciones industriales.
- **Tower Garden:** Sistema vertical que permite cultivar una gran cantidad de plantas en un espacio reducido, utilizando un sistema de aeroponía.

### Comparación con el proyecto:

En comparación con los sistemas comerciales, este proyecto presenta varias ventajas significativas. En cuanto al **costo**, los sistemas comerciales pueden ser bastante caros, mientras que este proyecto se enfoca en ofrecer una solución más accesible mediante el uso de componentes de bajo costo.

Otro aspecto importante es la **personalización**. Los sistemas comerciales suelen tener configuraciones fijas que limitan la flexibilidad del usuario. En cambio, este proyecto permite una mayor personalización tanto en el diseño como en el control del sistema, adaptándose mejor a las necesidades específicas de cada usuario.

Además, la **integración tecnológica** es una de las fortalezas de este proyecto. La combinación de ESP32-CAM, Node-RED e InfluxDB proporciona un enfoque moderno y altamente integrado para el control y monitoreo del sistema, ofreciendo una solución avanzada y eficiente para el manejo de cultivos hidropónicos.

### 1.4.3. TECNOLOGÍAS HARDWARE Y SOFTWARE APLICADAS AL CULTIVO HIDROPÓNICO DE COMPETENCIA

#### Hardware:

En el ámbito del hardware, se utilizan diversos **sensores y actuadores** para optimizar el cultivo hidropónico. Los **sensores** desempeñan un papel crucial, midiendo la temperatura, la humedad, el pH y la conductividad eléctrica (EC) para asegurar la

calidad del agua. Los **actuadores**, por otro lado, incluyen bombas de agua, ventiladores, luces LED y sistemas de riego automatizado, todos ellos esenciales para mantener el entorno ideal para las plantas.

Para el control de estos sensores y actuadores, se emplean microcontroladores como Arduino, ESP32 y Raspberry Pi, que facilitan la automatización y gestión eficiente del sistema hidropónico.

### **Software:**

En cuanto al software, las **plataformas de programación** son fundamentales, permitiendo la creación de entornos embebidos para los microcontroladores y el uso de lenguajes de programación que posibilitan una automatización avanzada. Los **sistemas de monitoreo y control** son también esenciales, proporcionando interfaces de usuario y flujos de trabajo de control complejos. Home Assistant, por ejemplo, es una herramienta popular para la integración domótica.

Las **bases de datos** juegan un papel importante en el almacenamiento y análisis de los datos recogidos por los sensores. Se utilizan tanto bases de datos privadas como bases de datos de series temporales, facilitando un análisis detallado y una gestión precisa de los cultivos.

## **1.5. OBJETIVOS**

El objetivo principal de este trabajo es diseñar y desarrollar un sistema de experimentación para el cultivo hidropónico, utilizando una combinación de hardware y software específico para automatizar el monitoreo y control de las condiciones ambientales. Este sistema buscará proporcionar una solución accesible y eficiente para la gestión de cultivos hidropónicos. En este proyecto, se tratará de diseñar y desarrollar un sistema automatizado de experimentación para el cultivo hidropónico, que permita monitorear y controlar las condiciones ambientales, optimizando el crecimiento de las plantas.

Para alcanzar este objetivo, se implementará un sistema capaz de **medir y modificar parámetros** como la temperatura, la humedad y la luz. Además, se desarrollará una **interfaz de usuario** interactiva basada en Node-RED que permitirá a los usuarios monitorear y controlar el sistema en tiempo real. También se utilizará InfluxDB para **almacenar los datos** de los sensores y proporcionar herramientas para su análisis. Finalmente, se realizarán **pruebas** en condiciones controladas para evaluar la eficiencia del sistema y su impacto en el crecimiento de las plantas.

Este objetivo puede desgranarse en otros objetivos más específicos, como son:

- **Diseñar e implementar un sistema de monitoreo y control basado en la ESP32-CAM:** La ESP32-CAM es un microcontrolador con capacidad Wi-Fi y una cámara integrada, ideal para el monitoreo remoto.
- **Desarrollar una interfaz de usuario utilizando Node-RED:** Node-RED es una herramienta de programación visual que permite conectar dispositivos hardware y servicios online, facilitando la creación de una interfaz de usuario interactiva y funcional.
- **Integrar una base de datos InfluxDB para el almacenamiento y análisis de datos:** InfluxDB es una base de datos de series temporales que permitirá almacenar y analizar los datos de los sensores para optimizar el control del sistema conectada a Node-red(Servidor Local).
- **Evaluar el rendimiento del sistema en condiciones controladas:** Se realizarán pruebas para verificar el correcto funcionamiento del sistema y su capacidad para mantener las condiciones óptimas para el cultivo hidropónico.

Con estos objetivos, el proyecto busca demostrar la viabilidad de una solución tecnológica integral para el cultivo hidropónico, que pueda ser utilizada tanto en entornos de investigación como en aplicaciones comerciales.

## 2. DEFINICIÓN DE REQUISITOS

### 2.1. REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL EQUIPO DE EXPERIMENTACIÓN

El diseño del sistema automatizado para el cultivo hidropónico incluye varios **requisitos funcionales** clave. Primero, el sistema debe medir y registrar la **temperatura y la humedad** del ambiente de cultivo, garantizando un monitoreo constante de las condiciones ambientales. Además, es fundamental que el sistema proporcione una **visualización en vivo** del entorno de cultivo, con la capacidad de **almacenar imágenes** tanto en un servidor local como en un directorio en base de datos online como InfluxDB. Para asegurar que las plantas reciban la cantidad adecuada de luz, se implementará una tira LED controlada.

La **circulación de aire** y el control de la temperatura y humedad se gestionarán mediante un ventilador. Para facilitar el control y la visualización de datos en tiempo real, se desarrollará una **interfaz de usuario** en Node-RED, permitiendo a los usuarios interactuar con el sistema tanto desde un servidor local como desde dispositivos conectados a la red. La integración de InfluxDB será esencial para **almacenar los datos** de los sensores, permitiendo su posterior análisis. Además, se aprovechará la capacidad Wi-Fi de la ESP32-CAM para la transmisión eficiente de datos entre los sensores, los actuadores y el servidor Node-RED.

Además de los requisitos funcionales, el sistema debe cumplir con **varios requisitos no funcionales** que garantizarán su usabilidad y efectividad a largo plazo. La **facilidad de uso** es primordial, por lo que la interfaz de usuario debe ser intuitiva y accesible para personas con diferentes niveles de conocimientos técnicos. La **fiabilidad** también es crucial; el sistema debe ser robusto y capaz de funcionar de manera continua y estable durante largos períodos. En términos de **escalabilidad**, el diseño debe permitir la adición de más sensores y actuadores en el futuro sin necesidad de grandes modificaciones, asegurando que el sistema pueda crecer junto con las necesidades del usuario. El **coste** es otro factor importante; el sistema debe ser económico, utilizando componentes y software de bajo costo. Finalmente, el

**mantenimiento** debe ser sencillo, permitiendo actualizaciones y reparaciones sin complicaciones. Estos requisitos no funcionales son esenciales para asegurar que el sistema sea práctico, accesible y sostenible a largo plazo.

### 3. IDENTIFICACIÓN DE COMPONENTES HARDWARE Y SOFTWARE NECESARIOS

#### 3.1. COMPONENTES UTILIZADOS EN LA PARTE HARDWARE

1. **ESP32-CAM:** Consiste en una PCB como se puede apreciar en la Fig 3 con un microcontrolador con capacidad Wi-Fi y cámara integrada para monitoreo visual y comunicaciones inalámbricas.. Con las siguientes características principales en la Tabla 1.

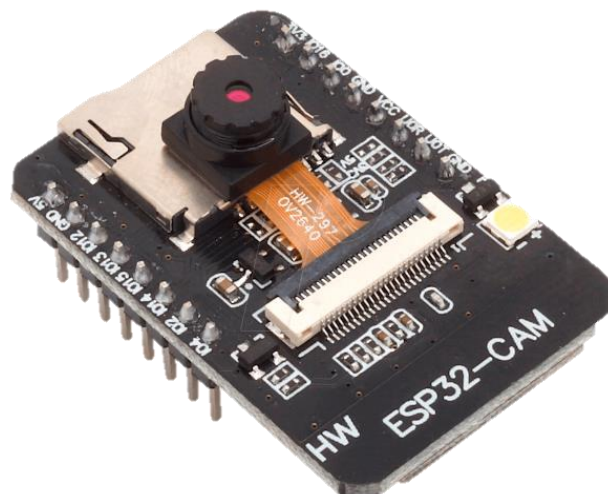


Fig 3. Módulo ESP32-CAM

Tabla 1. ESP32-CAM

| CARACTERÍSTICA   | DESCRIPCIÓN                                                                |
|------------------|----------------------------------------------------------------------------|
| MICROCONTROLADOR | Tensilica Xtensa LX6 dual-core a 240 MHz                                   |
| MEMORIA          | 520 KB SRAM                                                                |
| CONECTIVIDAD     | WiFi 802.11 b/g/n, Bluetooth v4.2 BR/EDR y BLE                             |
| CÁMARA           | OV2640, 2 megapíxeles, soporta JPEG                                        |
| ALMACENAMIENTO   | Soporte para tarjetas microSD                                              |
| GPIO             | 18 pines GPIO, algunos compartidos con funciones especiales como la cámara |

Esta información ha sido obtenida de la hoja de características datasheet del fabricante [3].

- 2. Programador ESP32-CAM:** PCB impreso con conexiones precisas como se puede ver en la Fig 4 para la programación del microcontrolador ESP32-CAM.



Fig 4. Programador ESP32-CAM

3. **Sensor DHT11:** Sensor de temperatura y humedad para monitoreo ambiental con placa de conexionado como se observa en la Fig 5. Cuyas características se resumen en la Tabla 2. Teniendo más información en: [4]

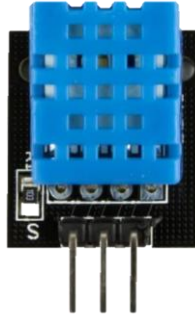


Fig 5. Sensor DTH11

Tabla 2. Características DHT11

| CARACTERÍSTICA           | DESCRIPCIÓN                             |
|--------------------------|-----------------------------------------|
| TIPO                     | Sensor digital de temperatura y humedad |
| RANGO TEMPERATURA        | 0 a 50 °C                               |
| PRECISIÓN DE TEMPERATURA | ±2 °C                                   |
| RANGO HUMEDAD            | 20% a 90% de humedad relativa           |
| PRECISIÓN DE HUMEDAD     | ±5%                                     |
| VOLTAJE DE OPERACIÓN     | 3.3V a 5.5V                             |
| TIEMPO DE RESPUESTA      | 1 segundo                               |
| SALIDA DE DATOS          | Señal digital en formato de pulso       |

4. **Tira LED:** componente electrónico flexible que consta de una serie de diodos emisores de luz (LED) montados en una tira flexible como se aprecia en la Fig 6. Estos diodos emisores de luz están dispuestos en una secuencia regular a lo largo de la tira y son capaces de emitir luz. Este componente está recubierto por un material transparente que protege los LED y difunde la luz de manera uniforme. Además, está equipada con adhesivo en la parte posterior para

facilitar su instalación en diversas superficies. A continuación se podrán visualizar las diferentes características principales en la Tabla 3.



Fig 6. Tira LED (5V)

Tabla 3. Características Tira LED

| CARACTERÍSTICA       | DESCRIPCIÓN                                                        |
|----------------------|--------------------------------------------------------------------|
| CONTROL              | A través de una señal digital accionando un Relé                   |
| VOLTAJE DE OPERACIÓN | 5V                                                                 |
| ALIMENTACIÓN         | 5V externos alimentando relé con fuente externa (batería portátil) |

5. **Ventilador:** Dispositivo electrónico diseñado para proporcionar circulación de aire mediante un motor alimentado por corriente continua (DC) utilizado comúnmente en aplicaciones de ventilación en ordenadores como el de la Fig 7. Ventilador DC (5V) Fig 7. Este tipo de ventilador se caracteriza por tener un tamaño compacto y ser alimentado típicamente con una tensión de 5 voltios, lo que lo hace adecuado para aplicaciones de baja potencia y portabilidad. Siendo sus principales características las que aparecen en la siguiente Tabla 4.



Fig 7. Ventilador DC (5V)

Tabla 4. Características Ventilador DC (5V)

| CARACTERÍSTICA       | DESCRIPCIÓN                                                           |
|----------------------|-----------------------------------------------------------------------|
| TIPO                 | Ventilador de enfriamiento DC                                         |
| VOLTAJE DE OPERACIÓN | 5V                                                                    |
| CONTROL              | Encendido y apagado mediante un relé, activado por una señal digital. |

6. **Relés:** Dispositivos electromecánicos que permiten controlar el flujo de corriente en un circuito eléctrico. En una PCB de 4 relés, hay cuatro relés montados en la placa, cada uno de los cuales puede ser controlado de forma independiente como se puede ver en la Fig 8. Cada relé en la PCB tiene una **bobina de control** que, cuando se energiza con una señal eléctrica de baja potencia, activa el interruptor interno del relé, permitiendo que pase la corriente a través de los contactos del relé. A continuación se mostrarán sus especificaciones en la Tabla 5.



Fig 8. Pack 4 Relés

Tabla 5. Características Módulo Relés

| CARACTERÍSTICA        | DESCRIPCIÓN                                                                                        |
|-----------------------|----------------------------------------------------------------------------------------------------|
| TIPO                  | Cuatro relés montados en una PCB, cada uno controlable de forma independiente                      |
| CONTACTOS DEL RELÉ    | Pueden ser configurados como normalmente abiertos (NA), normalmente cerrados (NC) o conmutados (C) |
| CONTROL               | Aplicando señal activa a cada relé                                                                 |
| CONEXIONES ELÉCTRICAS | Diseñadas para entrada de señales de control y conexiones de los circuitos de potencia             |
| INDICADORES LED       | Indican el estado de cada canal de relé (activado o desactivado) mediante luces LED                |

7. **Placa de expansión de pines:** Facilita la conexión de los diferentes componentes al microcontrolador y añade pines extra para facilitar la unión de conexiones como se observa en la Fig 9.

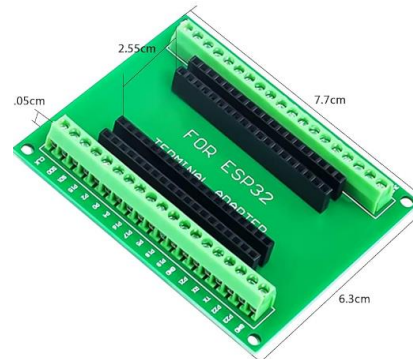


Fig 9. Placa de Extensión de Pines

8. **Batería portátil:** Fuente de alimentación para el sistema con dos salidas, con una capacidad de 10000mA. El Modulo compacto como se aprecia en la Fig 10. Posee las siguientes características principales como se aprecia en la Tabla 6.



Fig 10. Batería portátil

Tabla 6. Características Batería Portatil

| CARACTERÍSTICA    | DESCRIPCIÓN             |
|-------------------|-------------------------|
| CAPACIDAD         | 10000 mA                |
| VOLTAJE DE SALIDA | 5V                      |
| TIPO DE BATERÍA   | Iones de litio (Li-ion) |
| PUERTOS DE SALIDA | 2 Puertos USB           |
| ENTRADA DE CARGA  | microUSB                |
| DIMENSIONES       | 9,36 x 6,36 x 2,52 cm   |
| PESO              | 220                     |

### 3.2. COMPONENTES UTILIZADOS EN LA PARTE SOFTWARE

**Arduino IDE:** Entorno de desarrollo integrado para escribir, compilar y cargar programas en microcontroladores como el ESP32. Esta información ha sido encontrada en la web oficial del programa. [5]

- Lenguaje de Programación: C/C++.
- Bibliotecas Utilizadas:
- WiFi.h: Para conectividad WiFi.
- PubSubClient.h: Para la comunicación MQTT.
- esp\_camera.h: Para control de la cámara.
- FS.h y SD\_MMC.h: Para manejo del sistema de archivos y tarjetas SD.
- DHT.h: Para el sensor DHT11.

**Node-RED:** Plataforma de desarrollo basada en flujo para ensamblar aplicaciones conectando diferentes dispositivos, APIs y servicios en línea. [6]

Para programarla se utiliza el lenguaje de programación JavaScript (Node.js).  
Está formada por:

- **Flujos:** Interfaces gráficas para el diseño de flujos de datos.
- **Nodos:** Elementos funcionales que pueden representar entradas, salidas, funciones lógicas, etc.
- **Dashboard:** Herramienta para crear interfaces gráficas interactivas para la monitorización y control de dispositivos.

1. **InfluxDB:** Base de datos de series temporales optimizada para almacenar y analizar grandes volúmenes de datos a lo largo del tiempo. [7]

El lenguaje que utiliza es InfluxQL, similar a SQL pero diseñado para consultas de series temporales. Algunas de sus características principales son las siguientes:

- Alta escritura y lectura concurrente.
- Compresión de datos eficiente.
- Consultas avanzadas como agregaciones, derivaciones y transformaciones temporales.

- **MQTT**: Protocolo de comunicación ligero para la transmisión de datos entre dispositivos viendo como ejemplo el de la Fig 11.

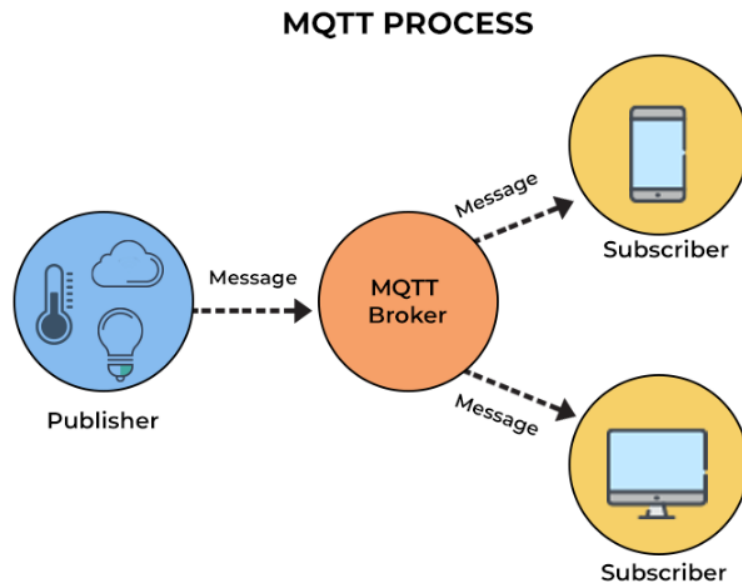


Fig 11. Protocolo MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación ligero y de alto rendimiento diseñado para la transferencia de datos en redes con ancho de banda limitado y alta latencia. Fue desarrollado por IBM y está optimizado para conectar dispositivos con recursos limitados en redes con poca capacidad de transmisión, como las utilizadas en el Internet de las Cosas (IoT).

### Características Principales de MQTT

#### 1. Modelo de Publicación/Suscripción:

- **PUBLICADOR:** Un dispositivo que envía mensajes (publica) a un tema (topic) específico.
- **SUSCRIPTOR:** Un dispositivo que recibe mensajes (se suscribe) a uno o más temas.
- **BROKER:** Un servidor que actúa como intermediario, recibiendo mensajes de los publicadores y enviándolos a los suscriptores.

#### 2. Temas (Topics):

- Los temas son jerárquicos y permiten organizar los mensajes en diferentes categorías.
- Ejemplo de tema: **sensor/temperatura/invernadero**.

### 3. Calidad del Servicio (QoS):

- QoS 0: Entrega del mensaje al menos una vez, sin confirmación.
- QoS 1: Entrega del mensaje al menos una vez, con confirmación de recepción.
- QoS 2: Entrega del mensaje exactamente una vez, con un protocolo más complejo de confirmación.

### 4. Retained Messages:

- Los mensajes retenidos permiten que los suscriptores reciban el último mensaje publicado en un tema al momento de suscribirse.

### 5. Will Message:

- Mensaje configurado para ser enviado por el broker si el cliente se desconecta inesperadamente.

## Ventajas de MQTT

- **Ligereza:** Uso mínimo de ancho de banda y baja sobrecarga en la red.
- **Escalabilidad:** Adecuado para redes con miles de dispositivos.
- **Fiabilidad:** Permite asegurar la entrega de mensajes con diferentes niveles de QoS.
- **Flexibilidad:** Los dispositivos pueden publicar y suscribirse a múltiples temas.

En este proyecto, el protocolo MQTT se utilizará para la comunicación entre la ESP32-CAM (cliente) y el servidor MQTT (broker). A continuación, se describen los pasos para implementar MQTT en el sistema.

- **Broker MQTT**

El broker MQTT puede estar alojado en un servidor local o en la nube. Para este proyecto, se utilizará un servidor local:

Para ello utilizamos el software **shiftr.io** [8] debido a su facilidad de uso y al útil entorno para ver la entrada y salida de datos. A continuación se puede ver un ejemplo de transmisión y recepción de datos en la Fig 12.

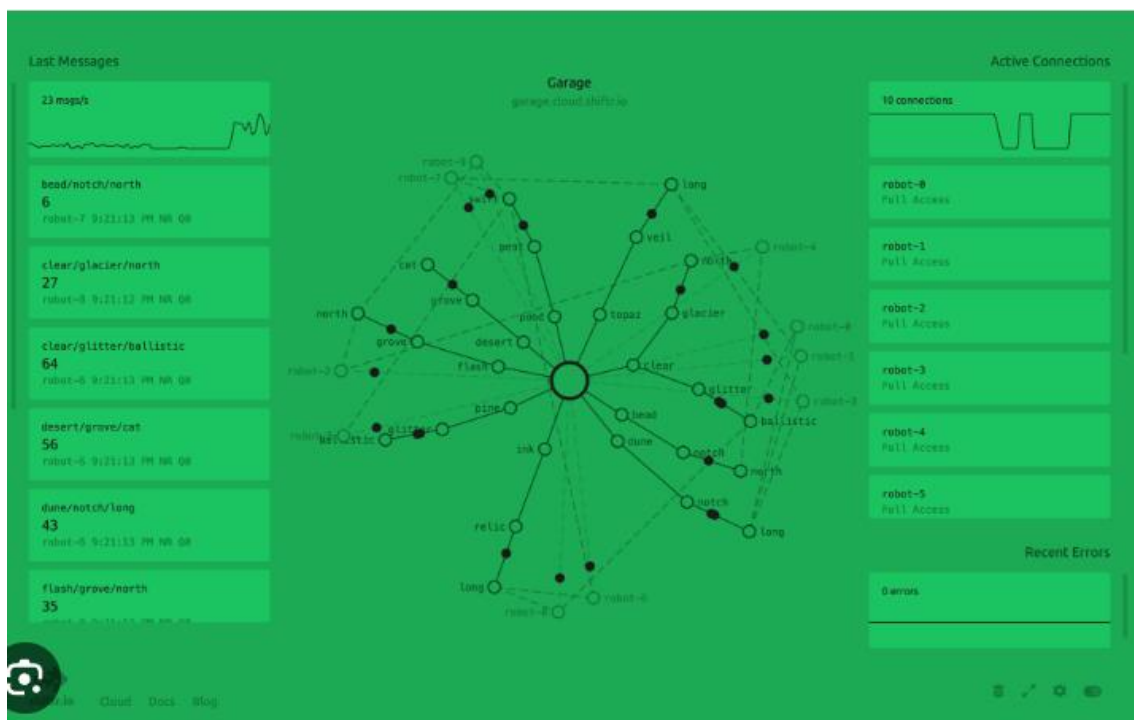


Fig 12. Ejemplo de funcionamiento de shiftr.io

- Conexión de la ESP32-CAM al Broker MQTT
- La ESP32-CAM se conectará al broker MQTT y publicará o se suscribirá a diferentes temas para controlar y monitorear el sistema hidropónico. (temperatura, humedad, actuadores, etc...).
- Publicación de Datos de Sensores  
La ESP32-CAM publicará los datos de temperatura y humedad del sensor DHT11 a través de MQTT.

## 4. DISEÑO DEL SISTEMA

### 4.1. DISEÑO HARDWARE

La estructura del sistema de cultivo hidropónico está diseñado para ser modular y flexible, montado en una bandeja de plástico que sirve como la base para el cultivo de plantas. Se utilizarán acoples impresos en 3D para sostener y organizar los diferentes componentes del sistema, incluyendo los sensores, actuadores y la ESP32-CAM. Siendo el prototipo final el mostrado en la Fig 13.

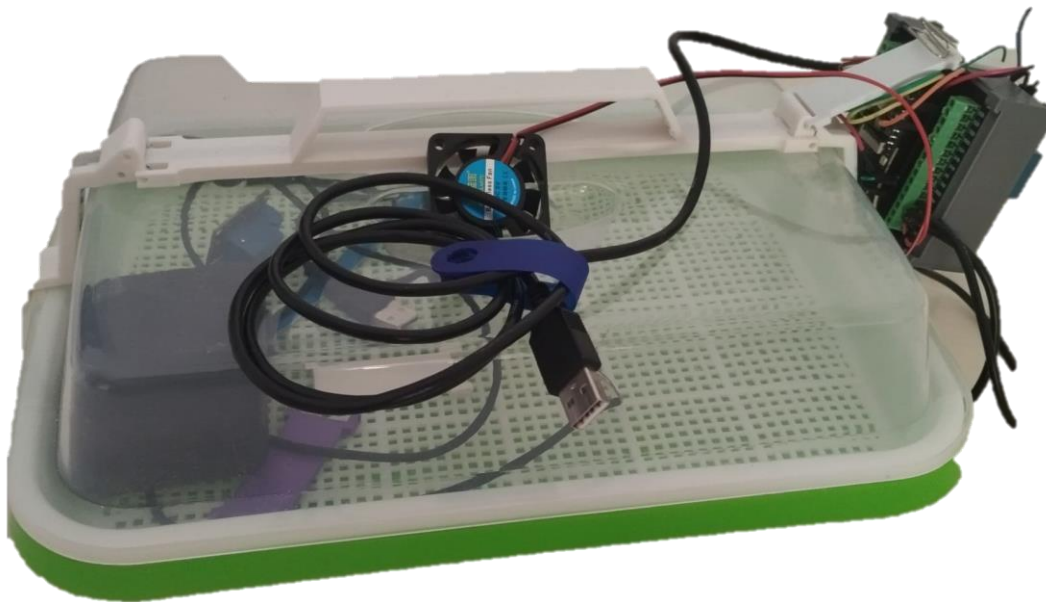
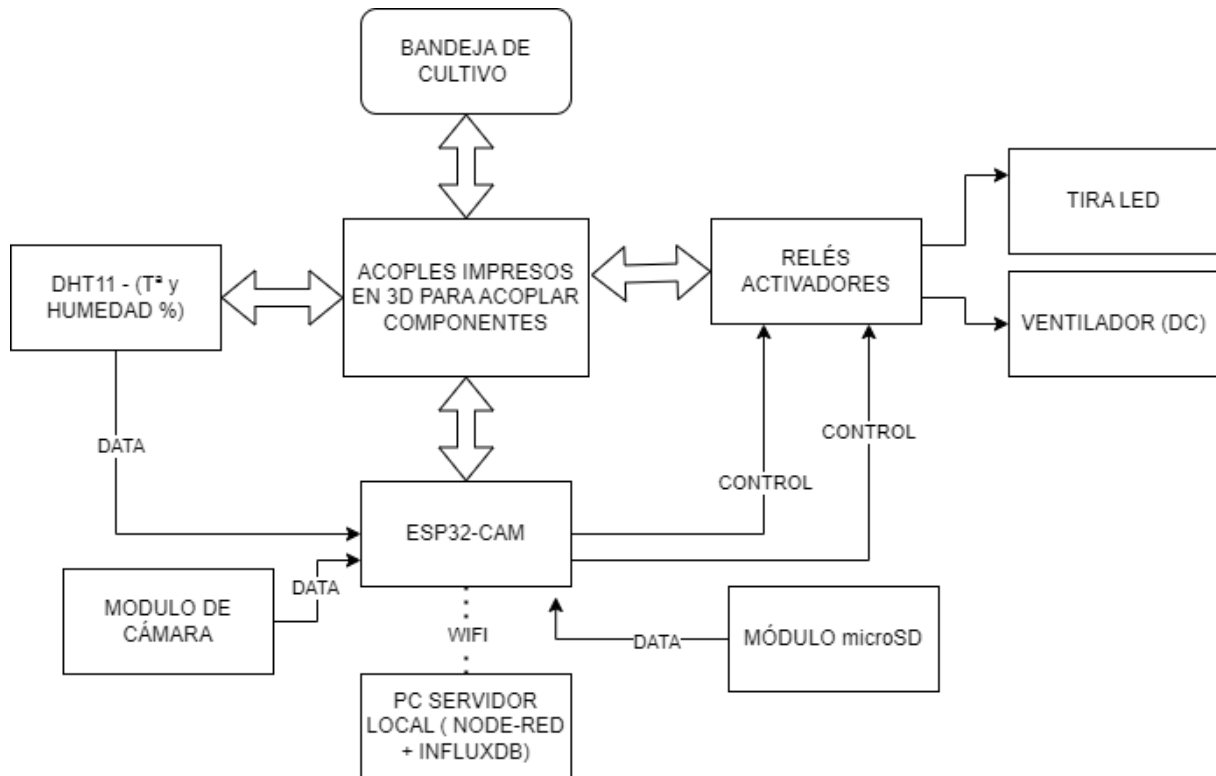


Fig 13. Prototipo final

**DIAGRAMA DE BLOQUES:**

- **Bandeja de cultivo:** Contiene las plantas y la solución nutritiva. Se usará como planta principal, siendo un ejemplo de la bandeja utilizada la Fig 14.



Fig 14. Bandeja Hidropónica

- **ESP32-CAM:** Controlador principal que gestiona la comunicación y procesamiento de datos.
- **Sensor DHT11:** Mide la temperatura y humedad del ambiente de cultivo.
- **Tira LED:** Proporciona la iluminación necesaria para el crecimiento de las plantas.
- **Ventilador:** Mantiene la circulación del aire y ayuda a controlar la temperatura y humedad.
- **Relés:** Controlan el encendido y apagado de la tira LED y el ventilador.
- **Componentes impresos en 3D** para conseguir un sistema compacto.

#### 4.1.1. SENSORES Y ACTUADORES

- **Sensor DHT11:** Seleccionado por su simplicidad y capacidad para medir tanto la temperatura como la humedad. Se conecta a la ESP32-CAM para enviar datos en tiempo real.
- **Módulo cámara OV2640 (integrado en ESP32-CAM):** Permite la captura de imágenes del estado de las plantas, lo que facilita el monitoreo visual remoto a buena calidad de imagen. Es un modelo flexible como se observa en la Fig 15 y se puede adaptar fácilmente a la planta.



Fig 15. Módulo de cámara OV2640

- **Tira LED:** Elegida por su eficiencia energética y capacidad de control preciso de la luz. Controlada por la ESP32-CAM a través de relés. Controlada mediante el pin GPIO 13. Se enciende y apaga según los comandos recibidos por MQTT.

- **Ventilador:** Ayuda a mantener la circulación del aire, esencial para evitar problemas de moho y mantener una temperatura estable. También controlado por relés. Controlado mediante el pin GPIO 12. Su funcionamiento es también regulado por comandos MQTT.

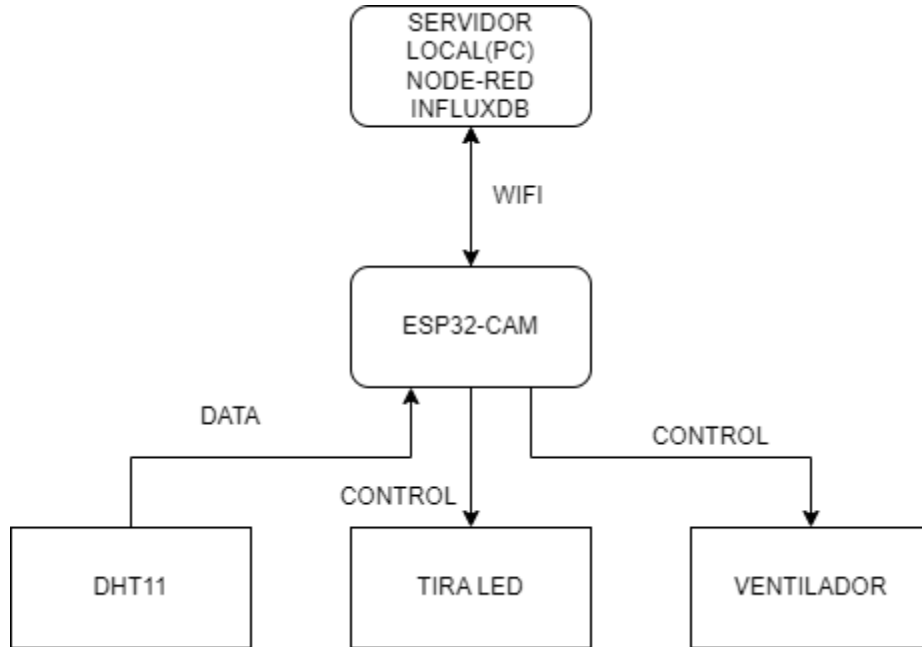
#### 4.1.2. CONTROLADORES Y ESTRUCTURA

- **ESP32-CAM:** Funciona como el cerebro del sistema, gestionando los datos de los sensores y controlando los actuadores. Alimentada por una batería portátil para asegurar la continuidad del funcionamiento incluso en caso de fallos de energía.
- **Placa de expansión de pines:** Utilizada para facilitar la conexión de los sensores y actuadores a la ESP32-CAM.
- **Bandeja de plástico y acoples impresos en 3D:** La bandeja de plástico actúa como el contenedor para el sistema de cultivo, mientras que los acoples impresos en 3D se utilizan para montar y organizar los componentes de manera ordenada y accesible.

#### 4.2. DISEÑO DEL SOFTWARE

El Sistema está configurado de la siguiente forma: Fig 16. Diagrama interconexión del sistema

- **Node-RED:** Implementado en un PC local, Node-RED es el **servidor local** del sistema. Sirve como la plataforma de control y monitoreo del sistema. Utiliza flujos visuales para recibir datos de los sensores, controlar los actuadores y proporcionar una interfaz de usuario accesible. Se hará uso de la **Dashboard**, que proporciona una interfaz gráfica para el usuario. Incluye visualizaciones de datos en tiempo real, gráficos históricos y controles interactivos para ajustar parámetros como la iluminación y la ventilación.
- **InfluxDB:** Base de datos de series temporales alojada en el mismo servidor local que Node-RED. Almacena los datos recopilados por los sensores, permitiendo su análisis y visualización histórica.



**Fig 16. Diagrama interconexión del sistema**

Este diagrama muestra la interconexión de los componentes principales del sistema y cómo se comunican entre sí. Se puede apreciar el diagrama de flujo en la Fig 16.

## 5. DESARROLLO DEL HARDWARE

### 5.1. ADQUISICIÓN DE COMPONENTES

La **ESP32-CAM** ofrece una serie de ventajas que la hacen destacar frente a otros microcontroladores. La **conectividad Wi-Fi** integrada simplifica la transmisión de datos y la conexión remota sin la necesidad de agregar módulos adicionales. En comparación con otras plataformas con características similares, la ESP32-CAM es una **opción económica**, lo que hace que el proyecto sea más accesible. El microcontrolador, Equipado con un procesador rápido y suficiente memoria, puede **manejar eficazmente múltiples tareas**, incluido el procesamiento de imágenes. La presencia de la **cámara OV2640** integrada permite el monitoreo visual de las plantas, agregando una capa adicional de control y análisis sin necesidad de componentes adicionales.

La elección del **DHT11** se basa en varios factores: La **simplicidad** que este sensor ofrece, es una forma sencilla de medir temperatura y humedad, con una precisión adecuada para proyectos de hobby y experimentación. El DHT11 es una opción muy **económica**, lo que lo hace ideal para proyectos con presupuestos limitados. Se **integra fácilmente** con la ESP32-CAM y otros microcontroladores mediante bibliotecas estándar disponibles en el entorno Arduino, lo que simplifica su uso y programación.

### 5.2. DISEÑO E IMPRESIÓN DE ACOPLERES 3D

Para el **diseño de piezas** se utilizan herramientas de diseño asistido por computadora (CAD) para crear modelos específicos que se adaptan a los componentes del sistema. Estos modelos aseguran que todos los elementos se mantengan firmemente en su lugar y se organizan de manera ordenada dentro de la bandeja de plástico.

Las piezas diseñadas se **imprimen** utilizando una **impresora 3D**, permitiendo la creación de acoples personalizados. Para esto se utiliza el software de laminación

*Prusa Slicer* para laminar las piezas [9]. Estos acoples facilitan el montaje y mantenimiento del sistema, asegurando que los componentes estén bien sujetos y accesibles.

A continuación se mostrarán las diferentes piezas diseñadas e impresas para ver con claridad el modelo final.

- **Soporte principal** para sujetar la bandeja: su principal función es facilitar el agarre de la bandeja y tener los acoples y agujeros necesarios para ensamblar con las siguientes piezas como se observa en la Fig 17.

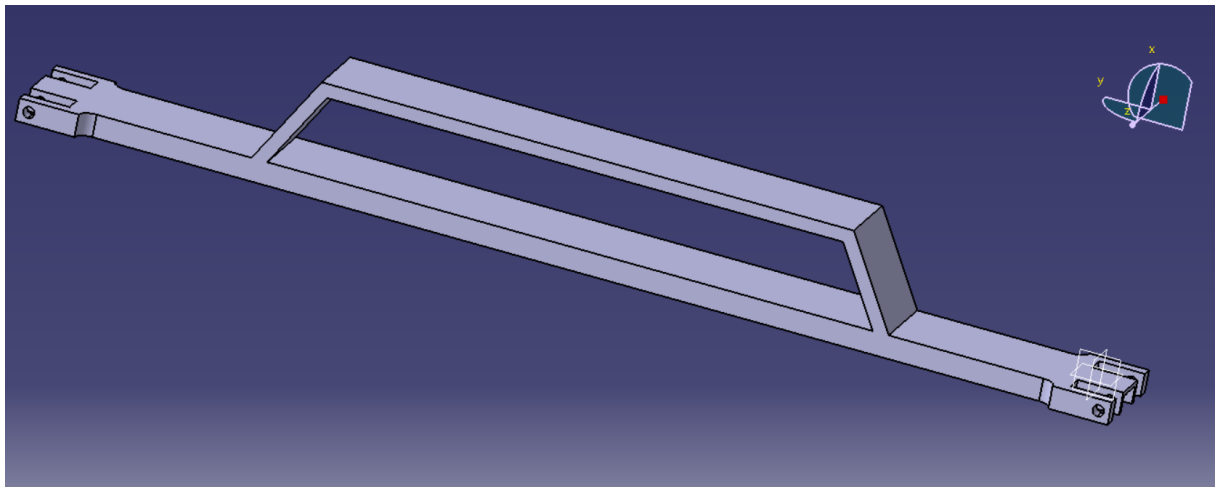


Fig 17. Soporte principal

- **Agarres laterales** para amarrar el soporte principal a la tapa transparente de la bandeja como se aprecia en la Fig 18. También será de gran ayuda para sostener el soporte para sostener la placa de expansión de pines y el conjunto de relés.

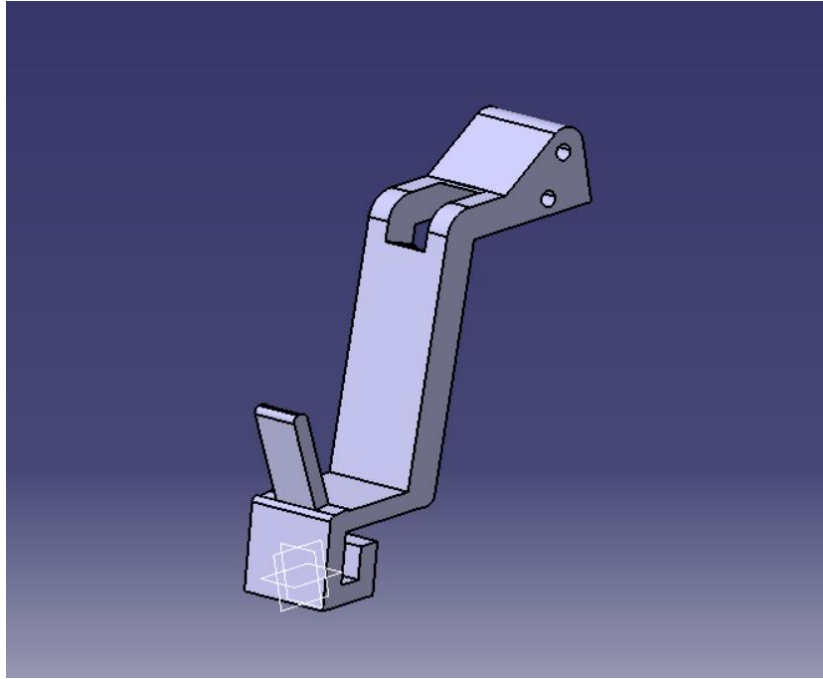


Fig 18. Soportes laterales

- **Soporte placa de expansión de pines con módulo de relés y tapa para cerrarlo.**  
Utilizando para sostener estos componentes el diseño de la Fig 19. Soporte ESP32-CAM y conjunto de relés para mantener el sistema compacto se utilizará la tapadera de la Fig 20.

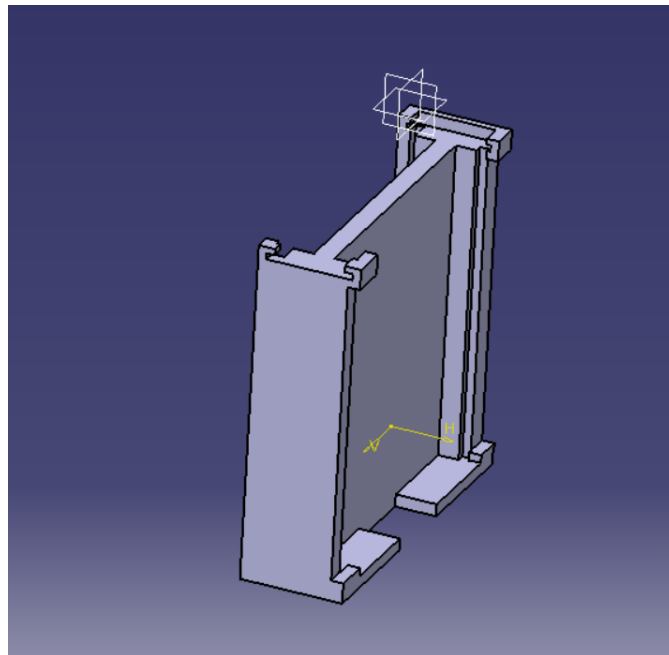


Fig 19. Soporte ESP32-CAM y conjunto de relés

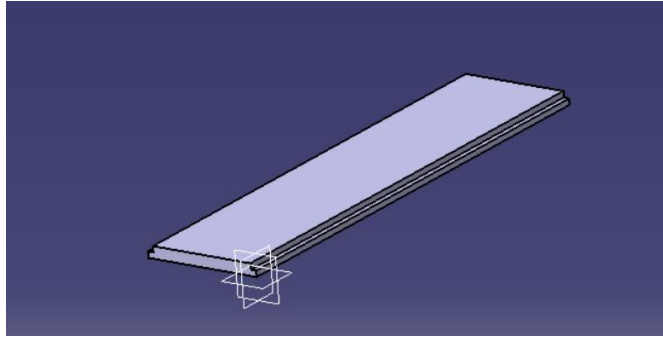


Fig 20. Tapadera sujeta componentes

- **Soportes extra** para mantener el sistema compacto y seguro se utilizarán los diseños mostrados en la Fig 21 y la Fig 22.

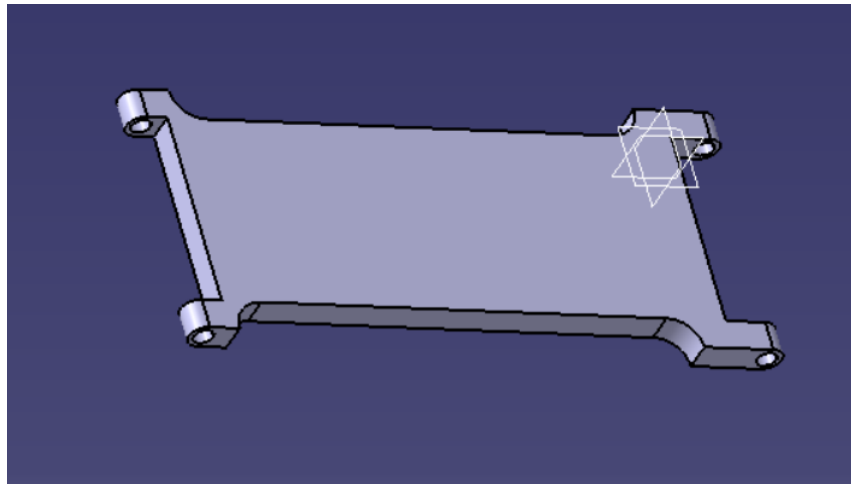


Fig 21. Soporte ESP32-CAM con bandeja

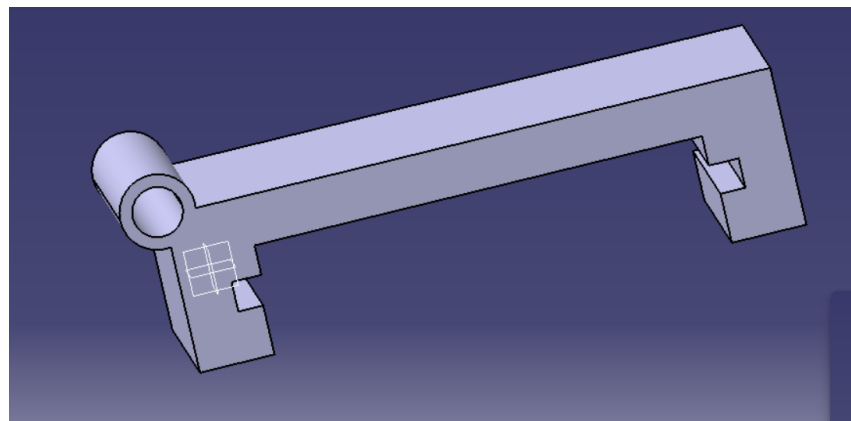


Fig 22. ESP32-CAM con Soporte

### 5.3. PROTOTIPADO DEL EQUIPO DE EXPERIMENTACIÓN

#### Montaje inicial:

El montaje inicial del sistema incluye varios elementos:

Se utiliza la **Placa de expansión de pines** donde la ESP32-CAM se conecta lo que facilita la conexión de los sensores (DHT11) y actuadores (tira LED y ventilador), permitiendo una integración ordenada y organizada.

Se diseñan unos **acoples impresos en 3D**, se fabrican, imprimiéndose en una impresora 3D para mantener firmemente en su lugar los componentes dentro de la bandeja de plástico. Esto asegura una disposición ordenada y accesible de los elementos del sistema.

Cómo planta se utiliza una **bandeja de plástico** que actúa como el contenedor principal para las plantas y el sistema de hardware, proporcionando un entorno controlado para el cultivo hidropónico, protegiendo los componentes electrónicos y facilitando la gestión del sistema.

### 5.4. PROCESO DE ENSAMBLAJE

El proceso de ensamblaje del sistema se lleva a cabo en varias etapas. Primero, se monta la **ESP32-CAM** en la **placa de expansión de pines** para asegurar una conexión sólida y estable. Luego, se procede a la **conexión** del sensor **DHT11** a los pines correspondientes de la **ESP32-CAM**, incluyendo los pines de alimentación, tierra y señal de datos, para garantizar una correcta lectura de temperatura y humedad. Acto seguido, se instala la **tira LED** y el **ventilador**, los cuales se conectan a los relés controlados por la ESP32-CAM, activados a través de los pines de control GPIO, permitiendo el control de la iluminación y la ventilación. Una vez completada esta etapa, se utiliza los **acoples impresos en 3D** para fijar todos los componentes en la bandeja de plástico, asegurando que estén firmemente sujetos y que los cables estén organizados de manera ordenada para evitar interferencias con el crecimiento de las plantas. Finalmente, se conecta una **batería portátil** a la ESP32-CAM para garantizar

una alimentación constante y portátil, asegurando el funcionamiento del sistema incluso en ausencia de una fuente de alimentación externa.

## 5.5. PRUEBAS Y AJUSTES DEL HARDWARE

### 5.5.1. PRUEBAS INICIALES

En las pruebas iniciales del sistema, se llevan a cabo varios pasos para garantizar su correcto funcionamiento y la integridad de los componentes. En primer lugar, se realiza una **verificación** minuciosa de todas las **conexiones** entre la ESP32-CAM, el sensor DHT11, la tira LED y el ventilador, asegurándose de que estén correctamente ensambladas y firmemente conectadas.

Una vez verificadas las conexiones, se procede al **encendido del sistema** y a la **prueba individual** de cada componente. Se verifica que el sensor DHT11 mida con precisión la temperatura y la humedad del ambiente, que la tira LED responda correctamente al encendido y apagado según lo programado, y que el ventilador funcione de manera adecuada.

Además de probar los componentes individualmente, se realizan **pruebas de comunicación** para confirmar que la ESP32-CAM establezca correctamente la conexión Wi-Fi con el servidor local. Se verifica que los datos recopilados por los sensores se transmitan de manera efectiva a **Node-RED** y que se almacenen de manera apropiada en la base de datos **InfluxDB** para su posterior análisis y procesamiento.

### 5.5.2. AJUSTES Y OPTIMIZACIÓN

Durante la etapa de ajustes y optimización del sistema, se realizan una serie de acciones para garantizar su eficiencia y mejorar su rendimiento. En primer lugar, se lleva a cabo la **calibración del sensor DHT11**, comparando sus lecturas con las de un termohigrómetro calibrado para corregir posibles desviaciones y mejorar la precisión de las mediciones.

Posteriormente, se procede a **ajustar la iluminación** proporcionada por la tira LED, variando tanto la intensidad lumínica como la duración de la iluminación de acuerdo con las necesidades específicas de las plantas en cada etapa de su ciclo de crecimiento.

Además, se realiza una **optimización del flujo de aire** dentro del invernadero, ajustando la velocidad y la ubicación del ventilador para garantizar una circulación adecuada sin causar estrés a las plantas. Se busca encontrar el equilibrio óptimo que proporcione una ventilación adecuada sin generar corrientes de aire excesivas que puedan perjudicar el desarrollo de las plantas.

### 5.5.3. PRUEBAS CONTINUAS

Durante la fase de pruebas continuas, se llevan a cabo pruebas prolongadas en diversas condiciones ambientales para garantizar la **fiabilidad** y **estabilidad** del sistema. Esto implica monitorear de cerca el comportamiento del hardware durante un período prolongado y ajustar los parámetros según sea necesario para optimizar su funcionamiento.

Esta documentación detallada garantiza un proceso de desarrollo y ajuste bien estructurado, facilitando futuras actualizaciones y mejoras en el sistema.

## 6. DESARROLLO DEL SOFTWARE

### 6.1. IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL

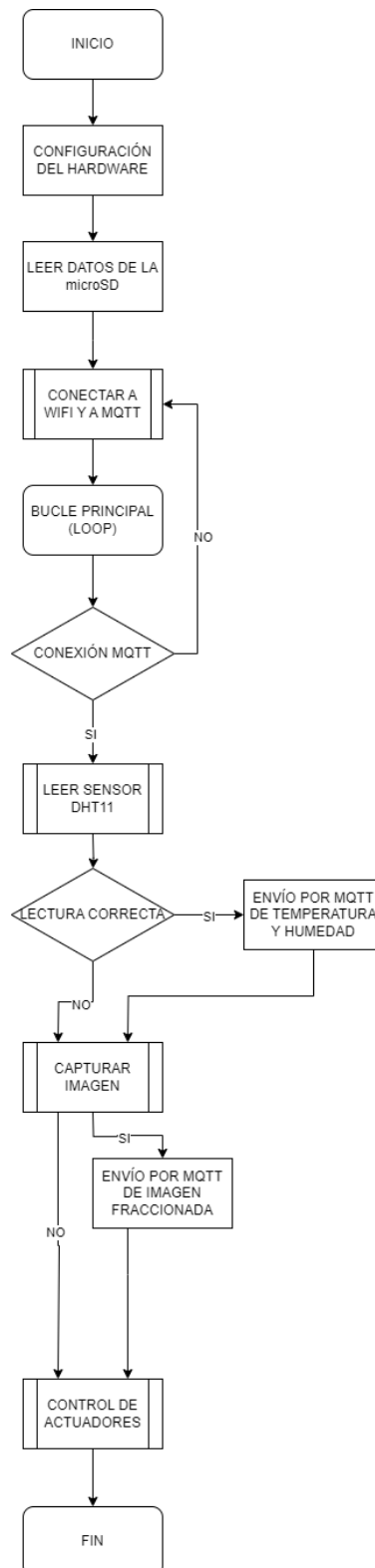


Fig 23. Diagrama de flujo de código ESP32-CAM

La implementación del software de control se realiza utilizando programación en entorno Arduino para la placa ESP32-CAM, tal y como se muestra el diagrama de flujo del programa principal en la Fig 23. A continuación se desglosará cada parte del código en distintos flujogramas. Toda la programación se aloja en un directorio almacenado en un repositorio de GITHUB. A este repositorio se podrá acceder en el siguiente enlace. [10]

## ENTORNORNO DE DESARROLLO

**Arduino IDE** Ha sido el elegido para para programar la ESP32-CAM. El entorno Arduino es ideal por su facilidad de uso y la abundancia de bibliotecas disponibles que simplifican la implementación de diversas funcionalidades.

### 6.1.1. CONFIGURACIÓN INICIAL

Instalación de bibliotecas: Instalar las bibliotecas necesarias en el Arduino IDE, incluyendo las bibliotecas para el DHT11, la cámara OV2640, Wi-Fi y MQTT.

Para ser capaz de modificar los valores de SSID como PASSWORD de la red Wi-Fi, así como el puerto y el servidor MQTT, y elegir el directorio específico del lugar de almacenamiento de imágenes en el servidor local, se siguen las siguientes instrucciones: INCLUSIÓN DE DATOS WIFI, BROKER Y DIRECTORIO DE ALMACENAMIENTO LOCAL

### 6.1.2. CONFIGURACIÓN DE PINES

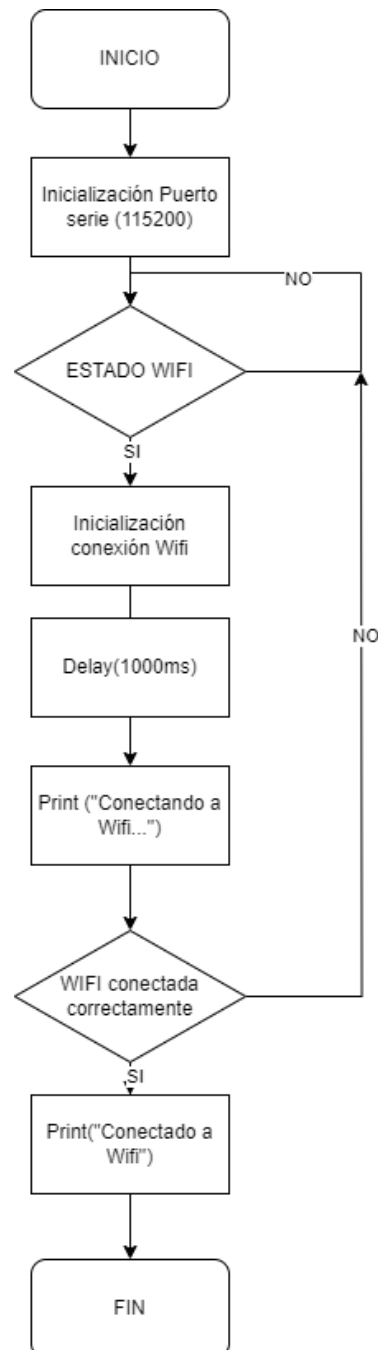
Para la correcta implementación del sistema de automatización de cultivo hidropónico, es esencial definir las conexiones entre los sensores, actuadores y el microcontrolador. Este proceso asegura que todos los componentes puedan comunicarse de manera eficiente y operar según lo esperado. En particular, asignar los pines adecuados del microcontrolador a cada dispositivo permite la lectura precisa de las condiciones ambientales y el control de los elementos que regulan el entorno

de cultivo. La siguiente sección presenta el código de configuración, que especifica las conexiones del sensor de temperatura y humedad DHT11, la tira LED y el ventilador, garantizando así una integración efectiva de todo el hardware en el sistema:

- Definición de **DHTPIN 11** : Pin al que está conectado el DHT11
- Definición **DHTTYPE DHT11** : Tipo de sensor DHT
- Definición **LED\_PIN 13** : Pin al que está conectada la tira LED
- Definición **FAN\_PIN 12** : Pin al que está conectado el ventilador

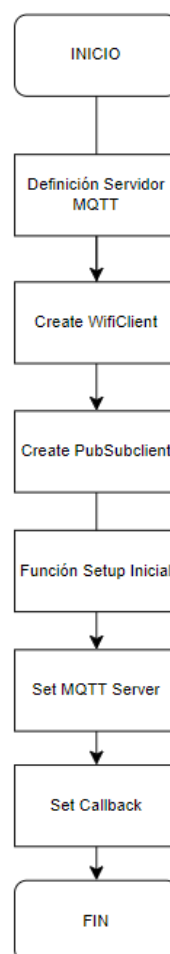
**6.1.3. BLOQUE MACRO CONEXIÓN WI-FI Y MQTT**

1. Conectar a Wi-Fi: Configurar la ESP32-CAM para que se conecte a la red Wi-Fi. Se adjunta el diagrama de desarrollo de la funcionalidad de la conexión Wi-Fi y MQTT en la Fig 24.

**Fig 24. Flujograma Conexión WIFI**

El flujograma se encarga de establecer una conexión WiFi en un microcontrolador, iniciando con la configuración de la comunicación serial a una velocidad de 115200 baudios para permitir la depuración y monitoreo. Luego, inicia el proceso de conexión a una red WiFi utilizando un SSID y una contraseña previamente definidos. Mientras la conexión no se haya establecido, el microcontrolador entra en un bucle que verifica el estado de la conexión cada segundo, imprimiendo "Connecting to WiFi..." en el monitor serial durante cada intento. Una vez que la conexión se ha establecido exitosamente, el bucle finaliza y el mensaje "Connected to WiFi" se imprime en el monitor serial, confirmando que el dispositivo está conectado a la red WiFi.

2. Configuración de MQTT: Configurar el cliente MQTT para la comunicación con Node-RED. [10]



**Fig 25. Flujograma Callback**

El fragmento de código proporcionado configura un cliente MQTT para permitir la comunicación con Node-RED. Primero, se define la dirección del servidor MQTT mediante la variable `mqtt_server`. A continuación, se crea un objeto `WiFiClient` llamado `espClient`, que sirve como cliente Wi-Fi, y un objeto `PubSubClient` llamado `client`, que utiliza `espClient` para la comunicación. En la función `setup()`, se configura el servidor MQTT especificando la dirección y el puerto (1883) mediante `client.setServer(mqtt_server, 1883)`. Además, se establece una función de devolución de llamada (callback) que manejará los mensajes entrantes desde el servidor MQTT utilizando `client.setCallback(callback)`. Este conjunto de configuraciones permite al dispositivo suscribirse y publicar mensajes en el servidor MQTT, facilitando así la comunicación con Node-RED. Se puede resumir y entender el funcionamiento con el diagrama de flujo mostrado en la Fig 25.

#### **6.1.4. BLOQUE MACRO LECTURA DE SENSORES Y CONTROL DE ACTUADORES:**

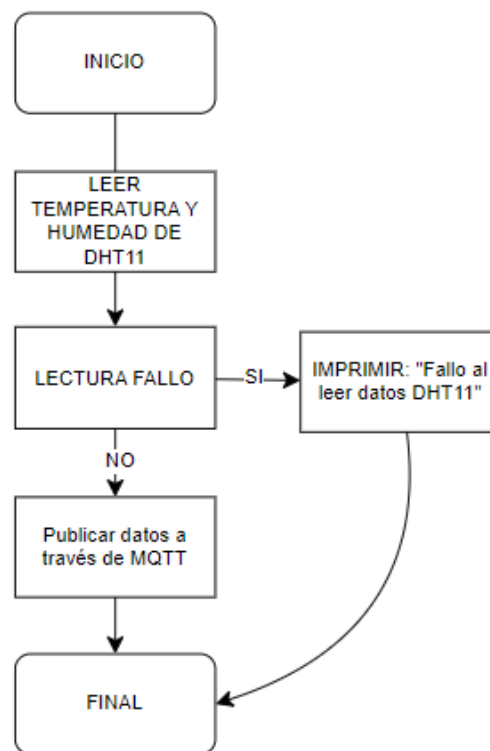


Fig 26. Lectura de Sensores y actuadores

El flujograma de código proporcionado en la Fig 26 se encarga de leer los datos de un sensor DHT11 y de publicar estos datos a través de MQTT. Dentro de la función `loop()`, primero se leen los valores de humedad (h) y temperatura (t) del sensor DHT11 utilizando los métodos `dht.readHumidity()` y `dht.readTemperature()`, respectivamente. Si alguna de estas lecturas falla, se verifica mediante la función `isnan()`, y se imprime un mensaje de error en el monitor serial, finalizando la ejecución de la función con `return`. Si las lecturas son exitosas, los valores obtenidos se publican a través de MQTT mediante la función `publishData(t, h)`. De esta manera, el código asegura la adquisición y transmisión continua de datos ambientales para su monitoreo remoto.

#### 6.1.5. BLOQUE MACRO CONTROL DE ACTUADORES

Para el control de la tira LED y el ventilador, se implementará el control de los actuadores basado en comandos recibidos a través de MQTT.

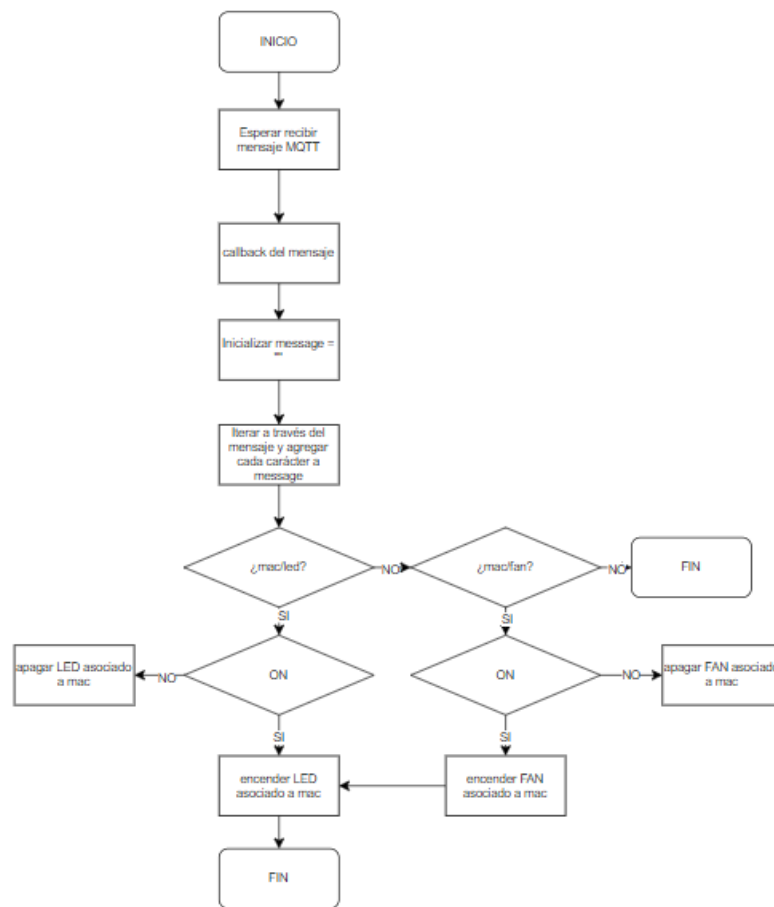


Fig 27. Control independiente de Actuadores

Este flujograma implementa el control de una tira LED y un ventilador basándose en los comandos recibidos a través del protocolo MQTT. En la Fig 27 se muestra el flujograma encargado de este fragmento de código. La función callback se ejecuta cada vez que llega un mensaje a un tópico MQTT al que el cliente está suscrito. Primero, el mensaje recibido se convierte en una cadena de texto para facilitar su manipulación. Luego, el código verifica si el tópico del mensaje es "home/led" o "home/fan". Si el tópico es "home/led" y el mensaje es "ON", el pin asociado con la tira LED se activa, encendiendo la luz. Si el mensaje es "OFF", el pin se desactiva, apagando la luz. De manera similar, si el tópico es "home/fan" y el mensaje es "ON", el pin asociado con el ventilador se activa, encendiendo el ventilador. Si el mensaje es "OFF", el pin se desactiva, apagando el ventilador. Este mecanismo permite controlar de manera remota los dispositivos conectados utilizando comandos enviados a través de MQTT.

## 6.2. DESARROLLO DE LA INTERFAZ DE USUARIO UTILIZANDO NODE-RED

El desarrollo de la interfaz de usuario utilizando Node-RED comienza con la instalación de Node-RED en un servidor local, como una PC personal, mediante el comando `sudo npm install -g --unsafe-perm node-red`. Para más información se puede consultar el anexo INSTALACIÓN NODE-RED . Después de la instalación, se realiza la configuración inicial que incluye la configuración de los nodos básicos para conectar con MQTT e InfluxDB. El nodo **MQTT** se configura para suscribirse a los temas relevantes y publicar comandos, facilitando la comunicación entre los dispositivos y el servidor. Asimismo, el nodo **InfluxDB** se configura para almacenar los datos recibidos de los sensores obteniendo la información de instalación en: INSTALACIÓN INFLUXDB, permitiendo la recopilación y el análisis de los datos ambientales del sistema hidropónico. La programación en formato JSON se almacena en un repositorio GIT al igual que la programación del microcontrolador ESP3-CAM. Este repositorio se encuentra en el siguiente enlace: [11]

- **EXPLIACIÓN DE LA DASHBOARD**

La Dashboard se divide en tres pantallas principales: 'DATOS ACTUALES', 'HISTORIAL DE DATOS' y 'OPCIONES AVANZADAS'.

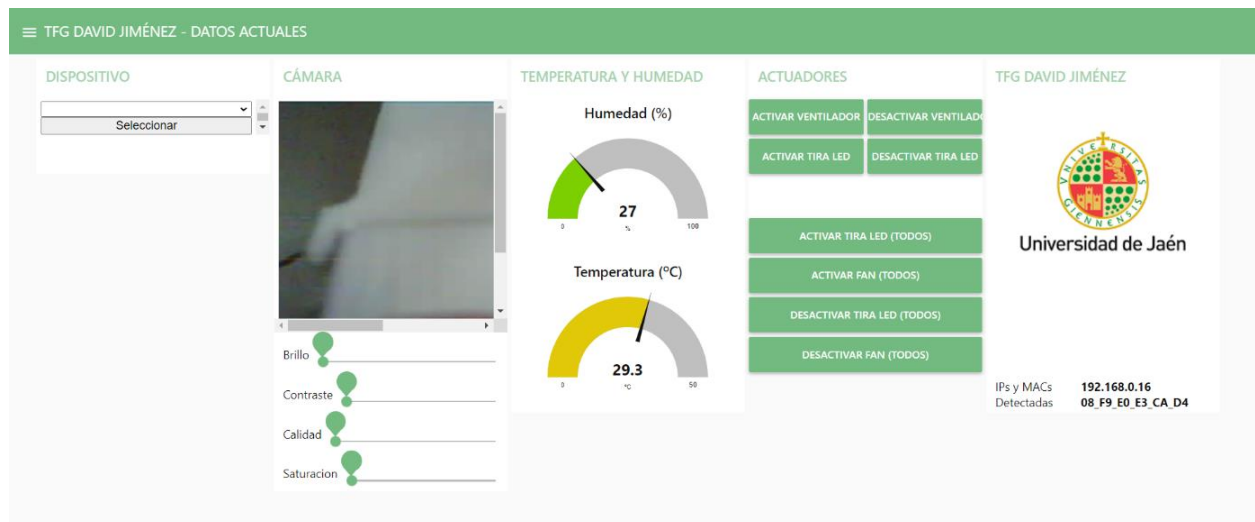


Fig 28. Interfaz usuario. "Datos Actuales"

En esta pantalla, como se observa en la Fig 28 se podrá acceder a los principales controles de la planta en el momento actual.

En primer lugar se deberá elegir el dispositivo a mostrar como en la Fig 29. El nombre del dispositivo se elige en la configuración inicial en el archivo data.txt donde se podrá configurar siguiendo la estructura de la Fig 54. Para acceder a la visión de la **cámara específica en vivo**, se actualizará una vez seleccionado el nombre de la bandeja. Debajo de esta opción tendremos la posibilidad de modificar los **parámetros** de la cámara de cada dispositivo independiente como se aprecia en la Fig 30.

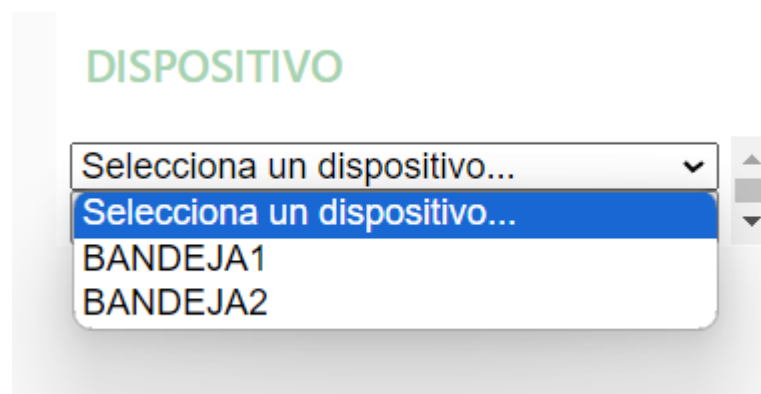


Fig 29. Selección de Dispositivo

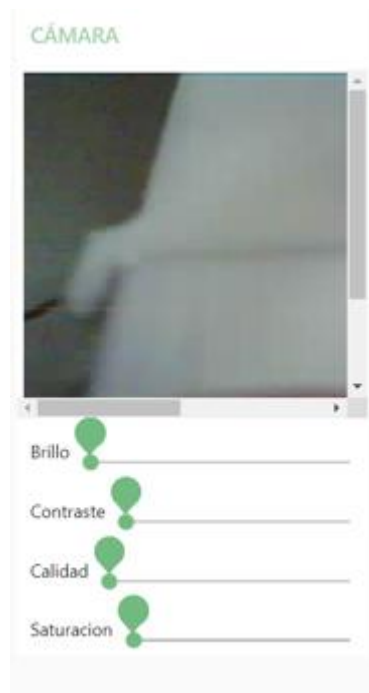


Fig 30. Cámara en vivo

Seguido a esto se podrá ver el último valor de **Temperatura y Humedad** recibido por el DHT11 de la bandeja seleccionada, teniendo como en el ejemplo de la Fig 31.

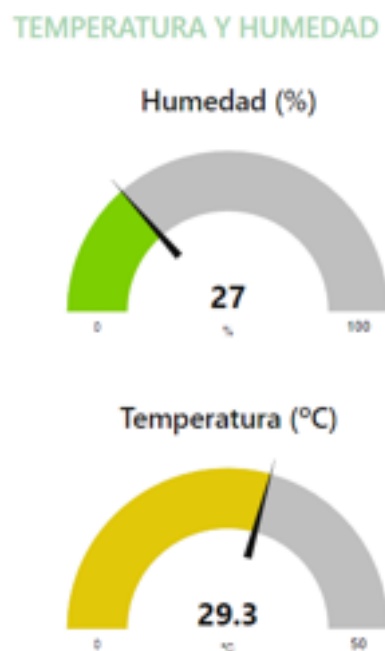
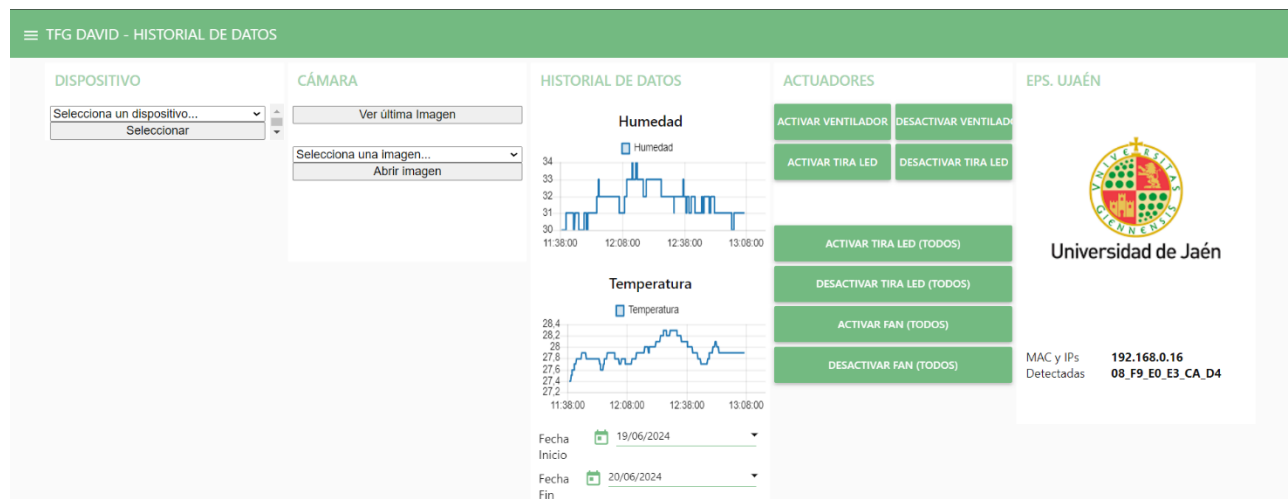


Fig 31. Temperatura y Humedad última

Por otro lado, se podrá interactuar con los diferentes actuadores entre los cuales se podrá controlar Tira Led, ventilador,... Teniendo la posibilidad de controlar la bandeja independiente seleccionada o manejar el control total de las bandejas como el ejemplo de la Fig 32.



**Fig 32. Actuadores**



**Fig 33. Interfaz usuario. "Historial de Datos"**

Aquí, se accederán a los valores de sensores y cámara históricos así como el control de la planta como se observa en la Fig 33.

En primer lugar, se tendrá acceso al **historial de las imágenes almacenadas** así como al último valor de imagen obtenida como se puede apreciar en la Fig 34. Para ello será indispensable la selección previa o posible ahora de la bandeja a visualizar.

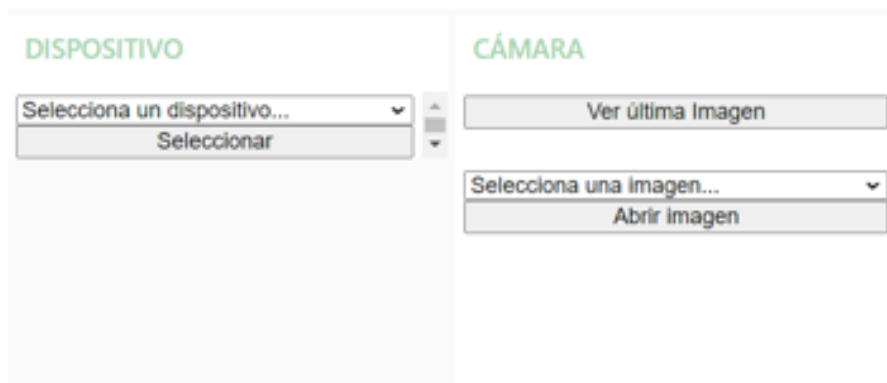


Fig 34. Almacenamiento Imágenes y selección de dispositivo

Seguido a esto se podrá obtener acceso al valor del historial de **temperatura y humedad específica** de la bandeja seleccionada en formato gráfico accediendo a los datos almacenados en influxDB como se aprecia en la Fig 35.

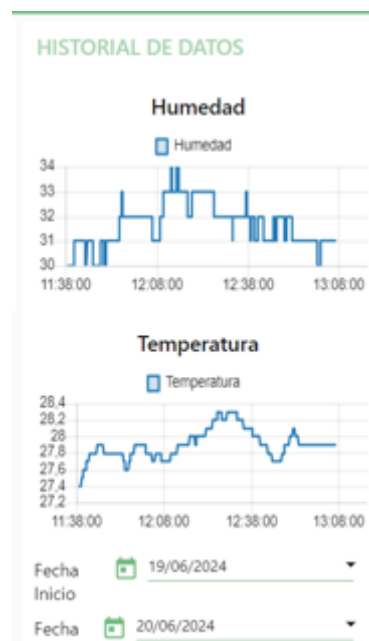
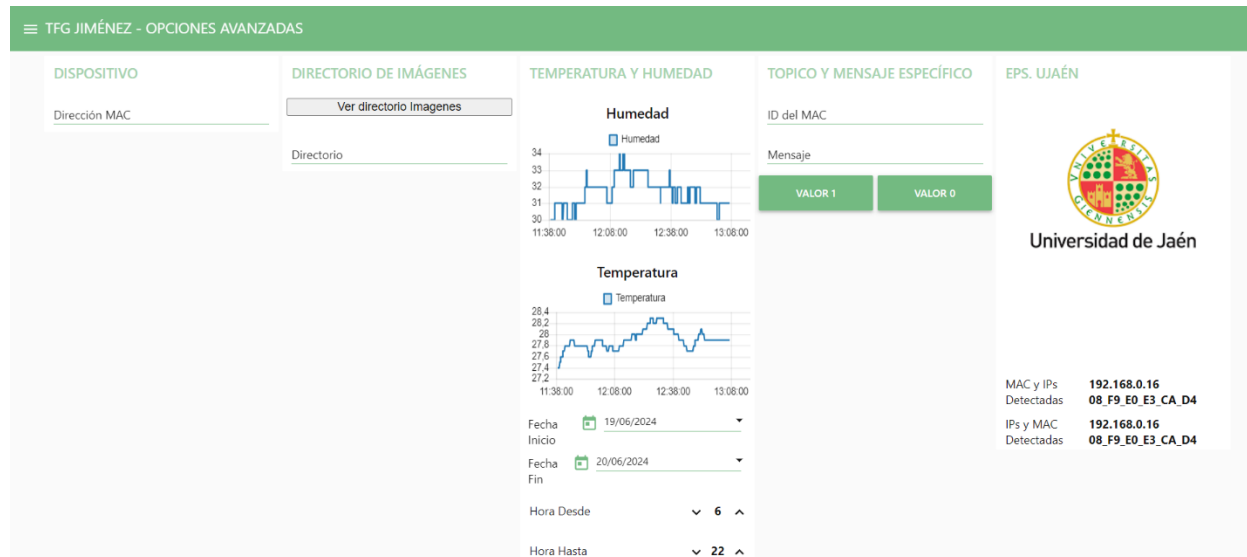


Fig 35. Gráficos Humedad y Temperatura frente a tiempo

Completando este menú, se podrá controlar los actuadores de igual forma que el MENÚ PRINCIPAL, Como se pudo ver en la Fig 32.



**Fig 36. Interfaz de Usuario 'Opciones Avanzadas'.**

Finalmente, en la última pantalla desplegable 'OPCIONES AVANZADAS' la cual se muestra en la Fig 36, se tendrá un acceso más personalizable tanto de la selección de dispositivo, como de la elección de datos a mostrar y el actuador concreto a activar o desactivar. A continuación se procederá a explicar cada función más detalladamente.

Al principio, se tendrá la posibilidad de elegir la dirección MAC para mostrar datos manualmente, introduciendo en formato: XX\_XX\_XX\_XX\_XX\_XX como se muestra en la Fig 37. Y teniendo las MAC detectadas disponibles en la parte derecha de la pantalla.

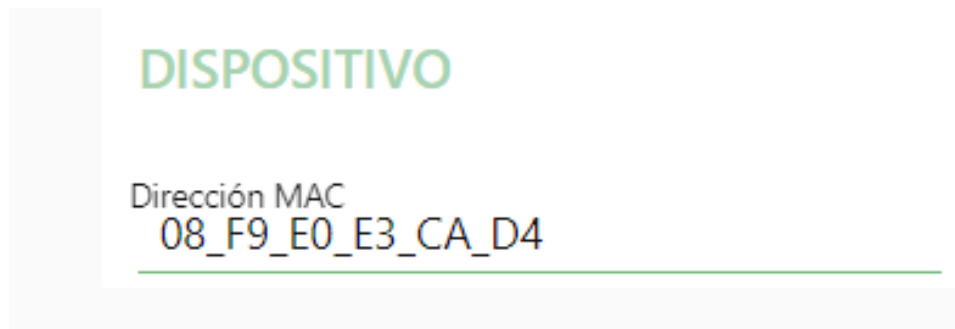


Fig 37. Dirección MAC personalizable

Seguido a esto, se podrá acceder al directorio donde están almacenadas todas las imágenes así como a una entrada de texto donde se dispondrá la posibilidad de elegir manualmente la dirección de acceso a donde se almacenan las mismas como se puede apreciar en la Fig 38.



Fig 38. Directorio de Imágenes

Seguido a esto, se dispondrá de una selección de histórico de datos de sensores tanto de Humedad como Temperatura con la posibilidad de elegir datos más precisos como el control de día y hora como se detalla en la Fig 39.

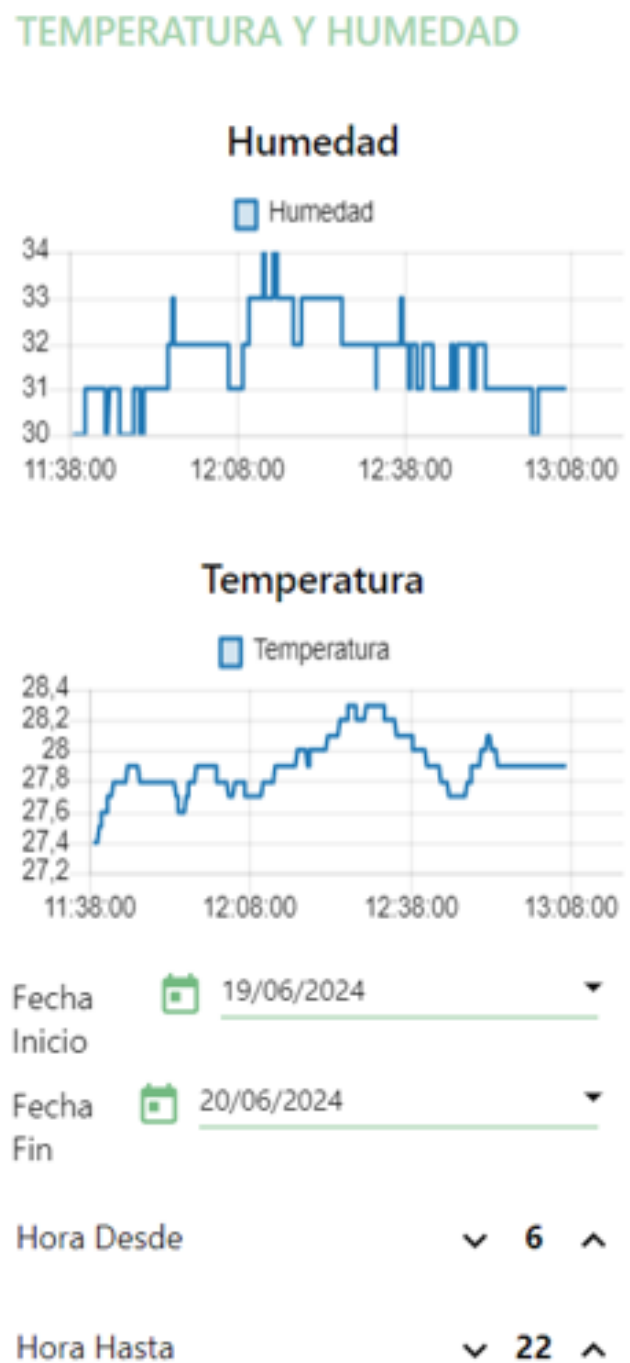


Fig 39. Historial específico de Temperatura y Humedad

### **6.3. EXPLICACIÓN DE LOS FLUJOS EN NODE-RED**

Para la correcta integración del microcontrolador con la parte del servidor local y con la propia dashboard, así como el control de actuadores, lectura de sensores y posterior envío hacia InfluxDB, almacenamiento de imágenes y la posibilidad de mostrar en vivo la planta, se hace uso de los distintos flujos utilizados en Node-Red explicados a continuación. Esta integración es crucial para el funcionamiento del sistema, permitiendo que los sensores recojan datos ambientales precisos y que los actuadores respondan de manera adecuada a las condiciones del cultivo, optimizando así el crecimiento de las plantas.

A continuación se desglosará cada parte del entorno node-red con su programación correspondiente:

6.3.1. FLUJO DE SENSORES

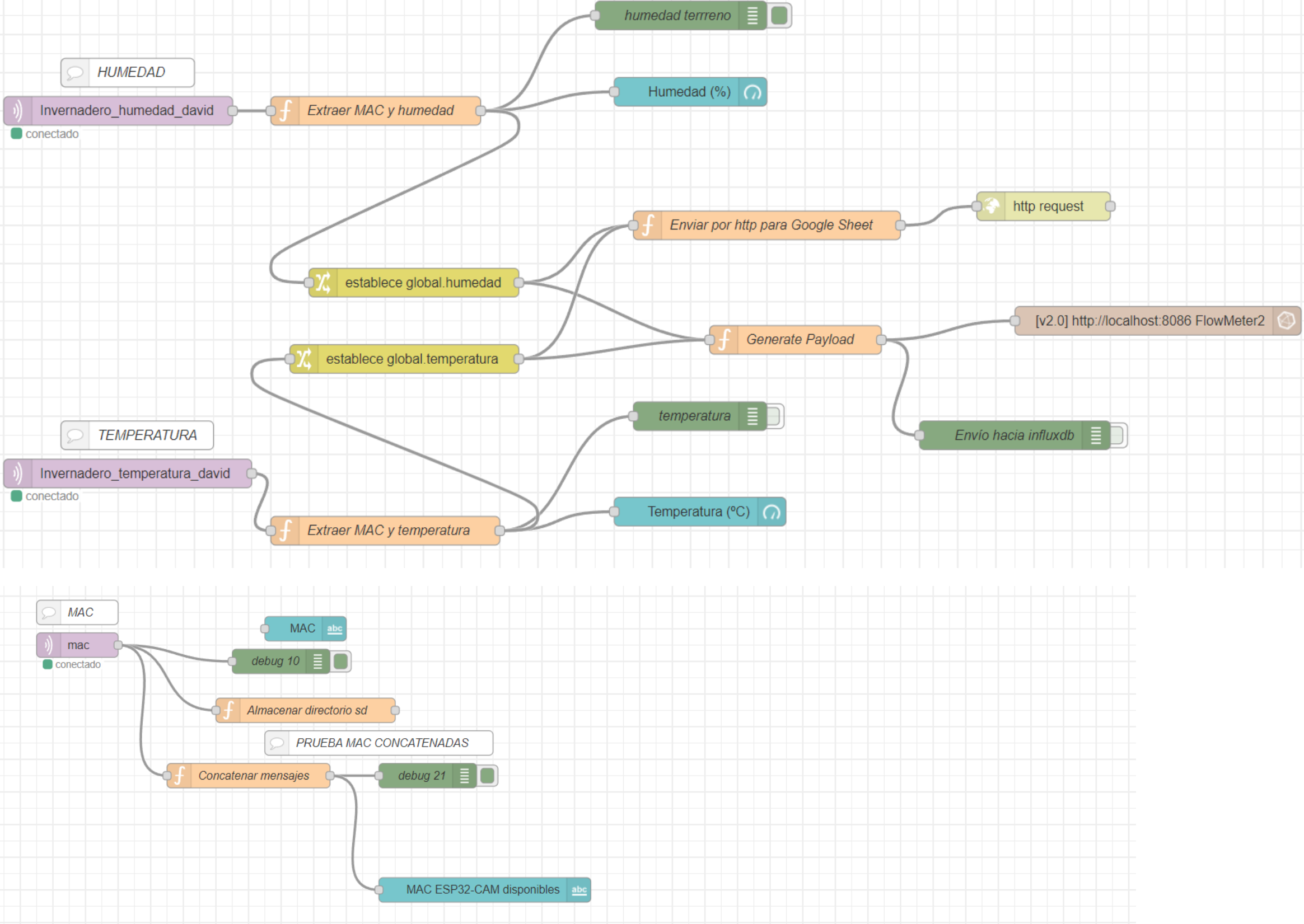


Fig 40. Node-Red. Flujo de Sensores

Estos flujos de Node-RED permiten recibir, procesar y mostrar datos de sensores de humedad y temperatura desde un ESP32, además de almacenar estos datos en una base de datos InfluxDB y en un formulario de Google Forms. Además, se gestionan las direcciones MAC de los dispositivos ESP32 conectados y se muestran en un tablero de control. En la Fig 40 se muestra el diagrama general del flujo de sensores.

Los principales nodos utilizados en este flujo son los siguientes:

1. **mqtt in**: Este nodo se utiliza para suscribirse a un tema MQTT y recibir mensajes publicados en ese tema. En el contexto del TFG, se emplea para recibir datos de los sensores instalados en el invernadero, como la temperatura, la humedad, etc. El nodo en cuestión es el que aparece como en la Fig 41.

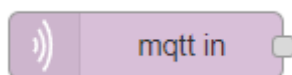


Fig 41. Nodo MQTT input

2. **function**: Los nodos de función se utilizan para realizar operaciones de procesamiento de datos. En el flujo de sensores, estos nodos pueden usarse para dividir los datos recibidos en partes, realizar cálculos específicos o formatear los datos para su visualización. El nodo función se muestra como en la Fig 42 en el entorno de Node-Red.

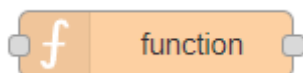


Fig 42. Nodo Funcion

3. **ui\_gauge**: Este nodo pertenece al dashboard de Node-RED y se utiliza para visualizar datos en forma de medidores. En el TFG, se emplea para mostrar la temperatura y la humedad en tiempo real. En el entorno Node-Red, el nodo se muestra de la forma de la Fig 43.

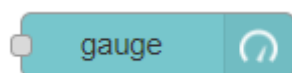


Fig 43. Nodo Gauge

4. **ui\_text\_input:** Estos nodos proporcionan campos de entrada de texto en el dashboard de Node-RED, siendo el nodo el que aparece en la Fig 44. Se pueden usar para ingresar información o configuraciones específicas, como la dirección MAC de un sensor en el flujo de sensores del TFG.

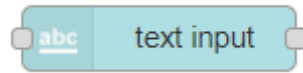


Fig 44. Nodo Text Input

5. **ui\_button:** Los nodos de botón se utilizan para crear botones interactivos en el dashboard de Node-RED, siendo el nodo utilizado el de la Fig 45. En el flujo de sensores, pueden usarse para enviar comandos de control o realizar acciones específicas, como la activación de un proceso de calibración.

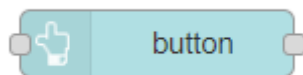


Fig 45. Nodo Button

### 6.3.2. FLUJO ACTUADORES

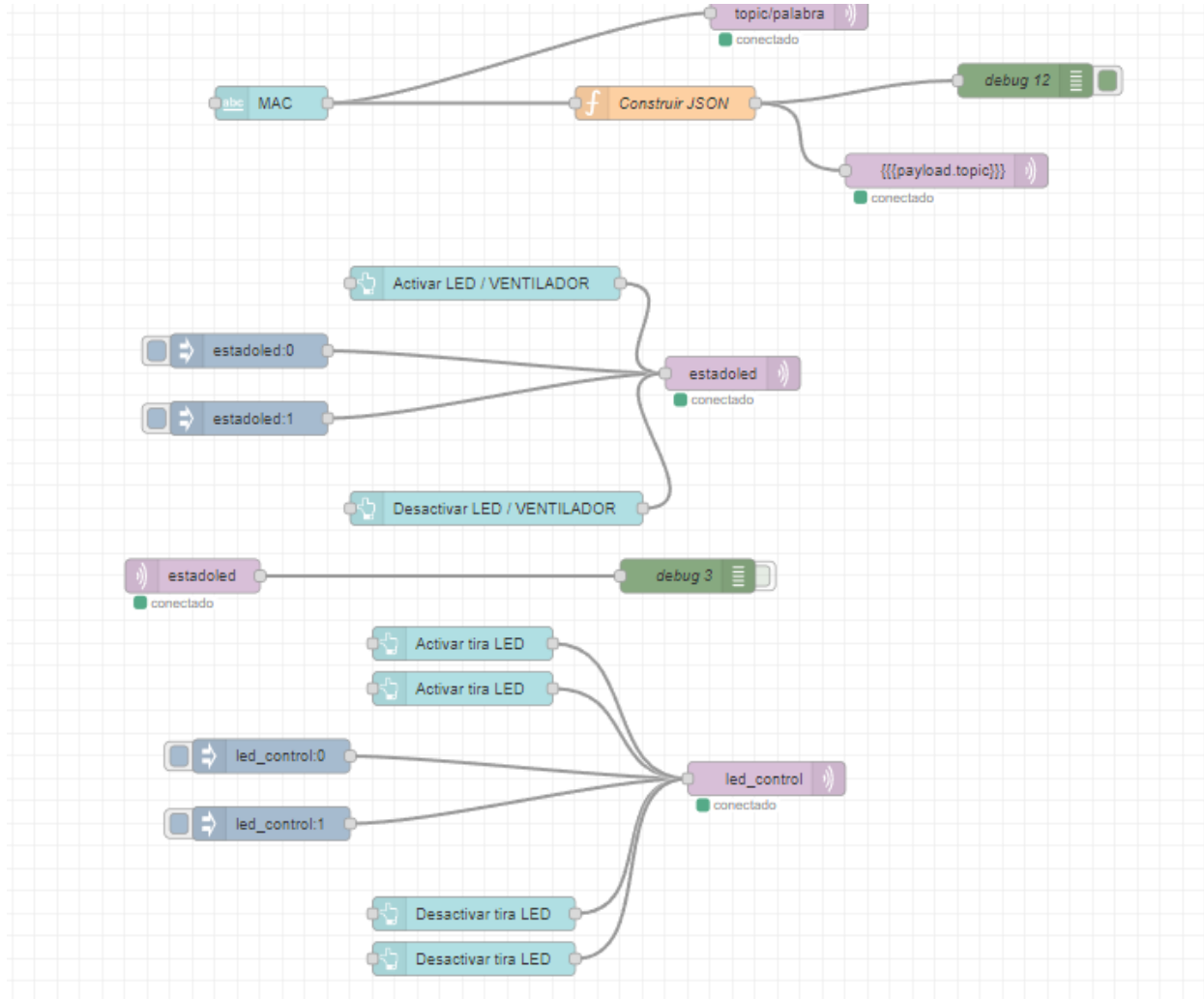


Fig 46. Node-Red. Flujo de Actuadores

Este flujo de Node-RED , mostrado en la Fig 46, permite controlar varios actuadores (LED, tira LED y ventilador) a través de mensajes MQTT, utilizando una interfaz de usuario con botones para facilitar la interacción. Además, permite introducir y enviar datos específicos como la MAC a través de la misma interfaz.

Los principales nudos utilizados en este flujo son los siguientes:

1. **mqtt out:** Este nodo, teniendo la apariencia que se muestra en la Fig 47, se utiliza para enviar mensajes MQTT que contienen comandos de control a los

actuadores del invernadero. Estos comandos pueden incluir instrucciones para activar sistemas de riego, controlar la iluminación o ajustar la ventilación, entre otros.

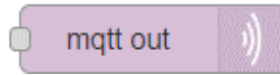


Fig 47. Nodo MQTT output

2. **ui\_switch**: Los nodos de interruptor, teniendo como ejemplo el de la Fig 48, se utilizan para crear interruptores interactivos en el panel de control del dashboard de Node-RED. Estos interruptores pueden estar vinculados a actuadores específicos y permiten a los usuarios activar o desactivar manualmente los dispositivos desde la interfaz de usuario.

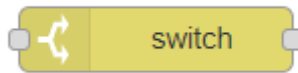


Fig 48. Nodo Switch

## Brokers MQTT

- **Broker MQTT**
  - Configurado con la dirección IP "192.168.0.14" y puerto "1883".
  - Se utiliza para enviar y recibir mensajes MQTT en los tópicos configurados en el flujo.

### 6.3.3. FLUJO IMÁGENES POR MQTT

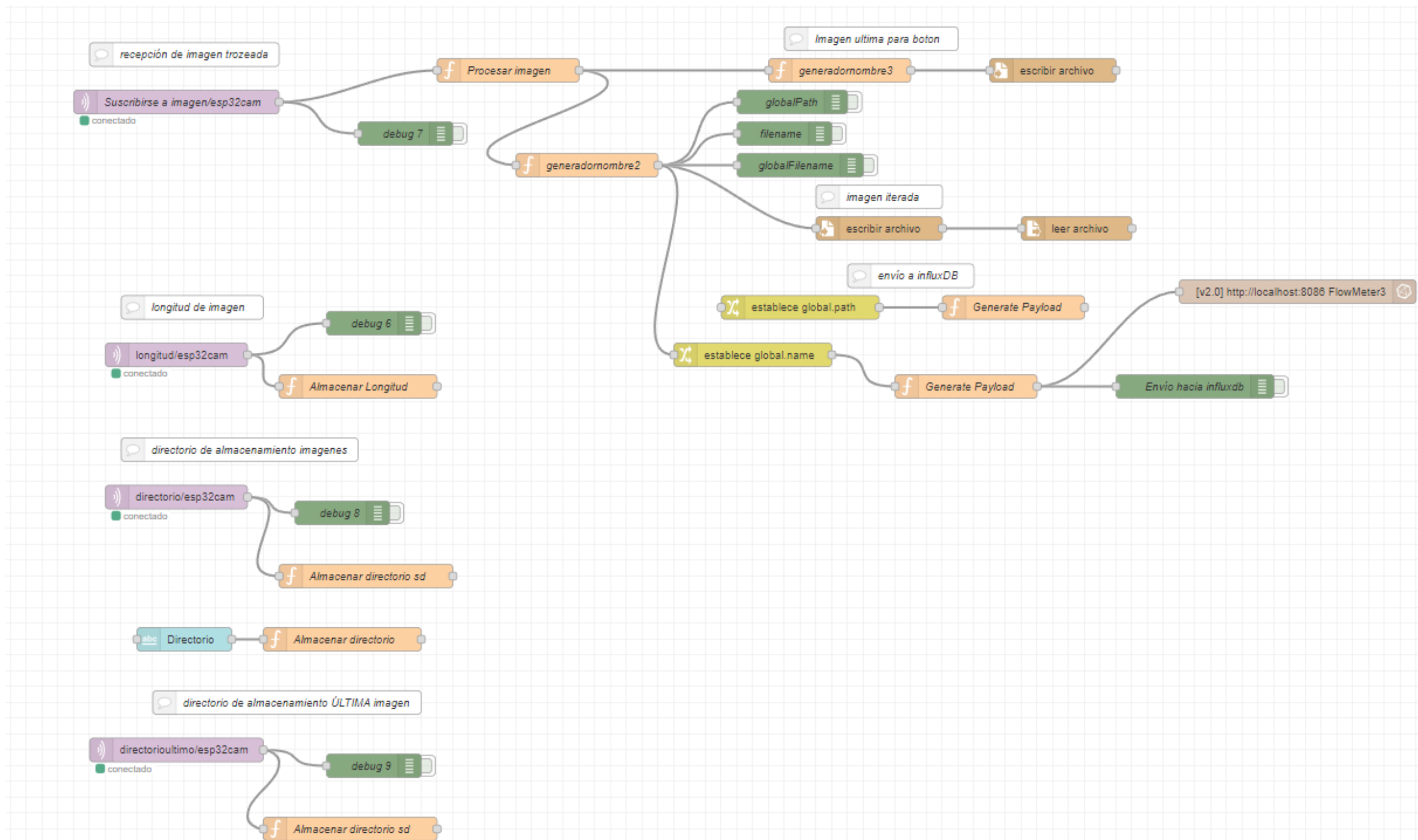
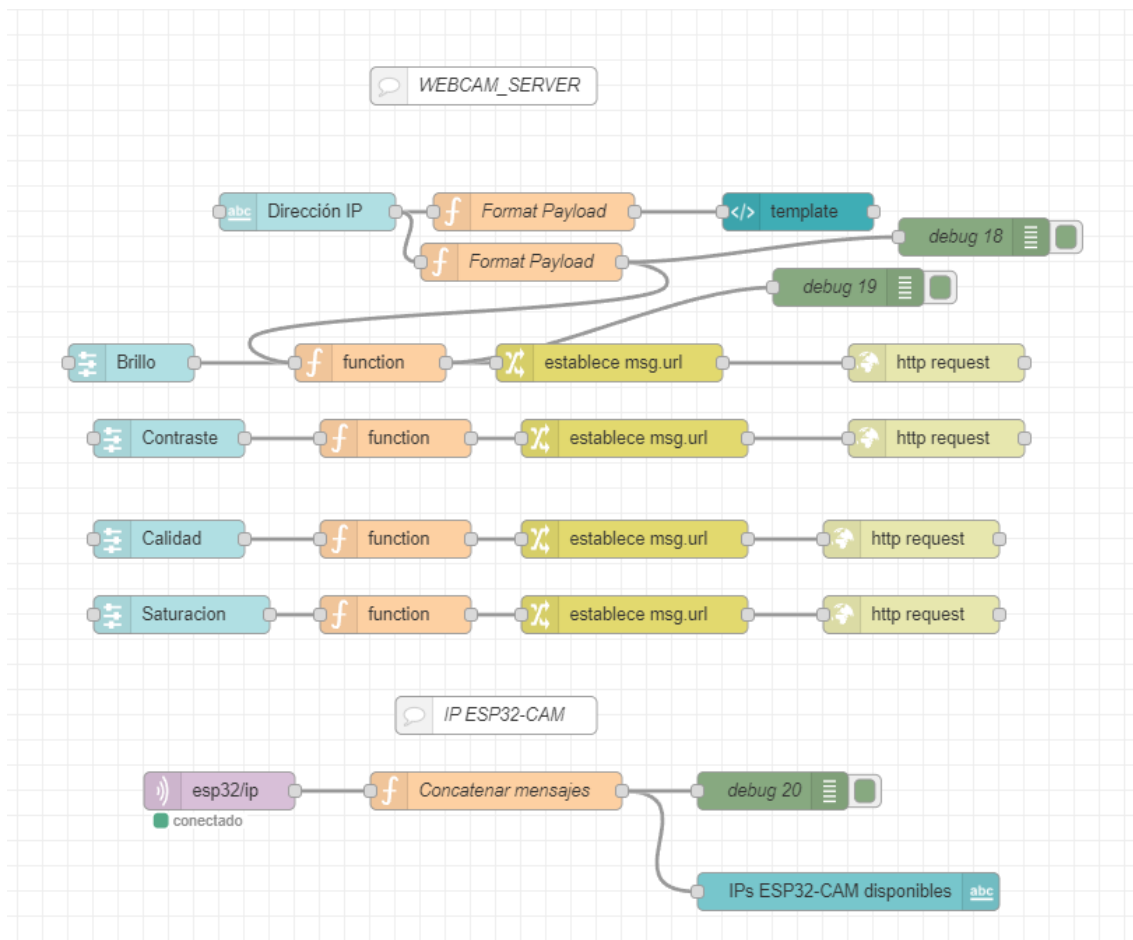


Fig 49. Node-Red. Flujo de Imágenes

El flujo de Node-RED llamado "IMAGENES\_MQTT" mostrado en la Fig 49, está diseñado para gestionar la recepción, procesamiento y almacenamiento de imágenes provenientes de un dispositivo ESP32CAM, utilizando el protocolo MQTT. Aquí se describe cada componente del flujo y su funcionamiento:

#### 6.3.4. FLUJO WEBCAM



**Fig 50. Node-Red. Flujo de Webcam**

Este flujo mostrado en la Fig 50, proporciona una interfaz web para controlar una cámara por IP, permitiendo ajustes en tiempo real del brillo, contraste, saturación y calidad de la imagen. Los ajustes se realizan enviando solicitudes HTTP a la cámara.

### 6.3.5. FLUJO ÚLTIMA IMAGEN

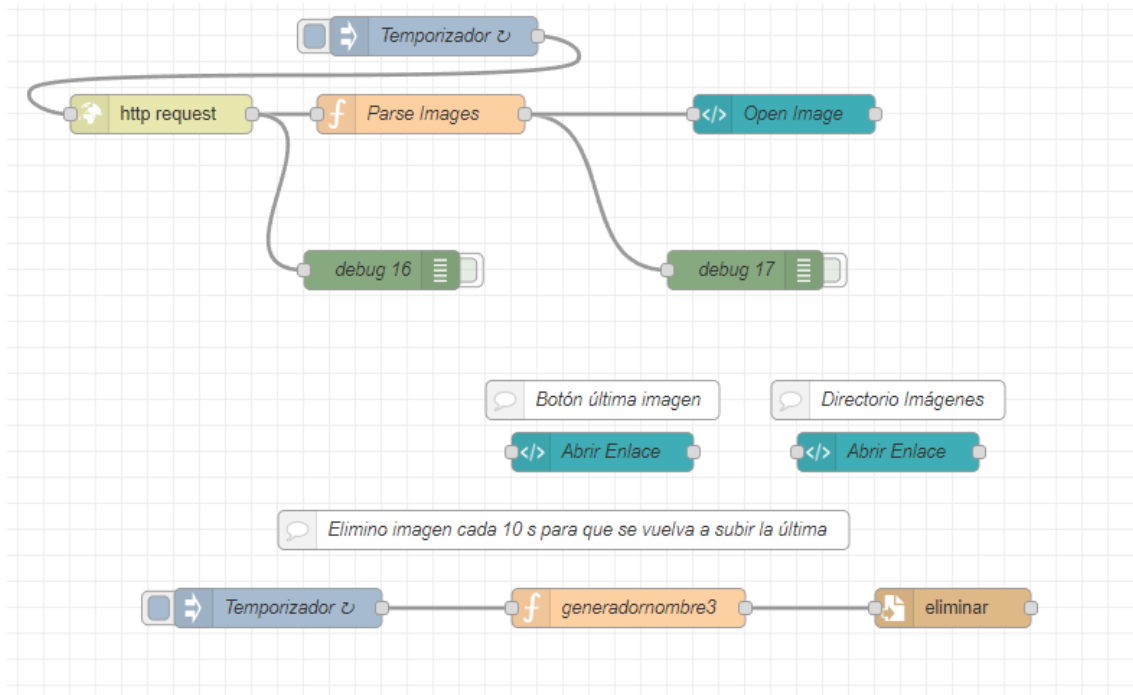


Fig 51. Node-Red. Banco de Imágenes

El flujo de Node-RED resumido en la Fig 51, está diseñado para interactuar con imágenes almacenadas localmente en el servidor web y proporciona una interfaz de usuario para seleccionar y abrir estas imágenes ya almacenadas en el servidor así como permitir acceso a su directorio.

### 6.3.6. FLUJO ENVIAR TÓPICO ESPEFÍCO CON MENSAJE ESPEFÍFICO

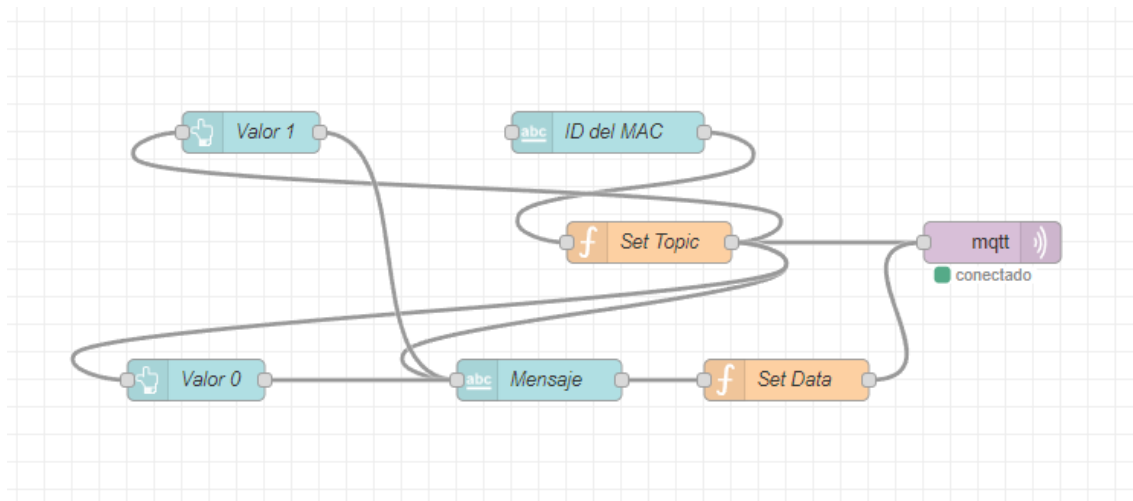
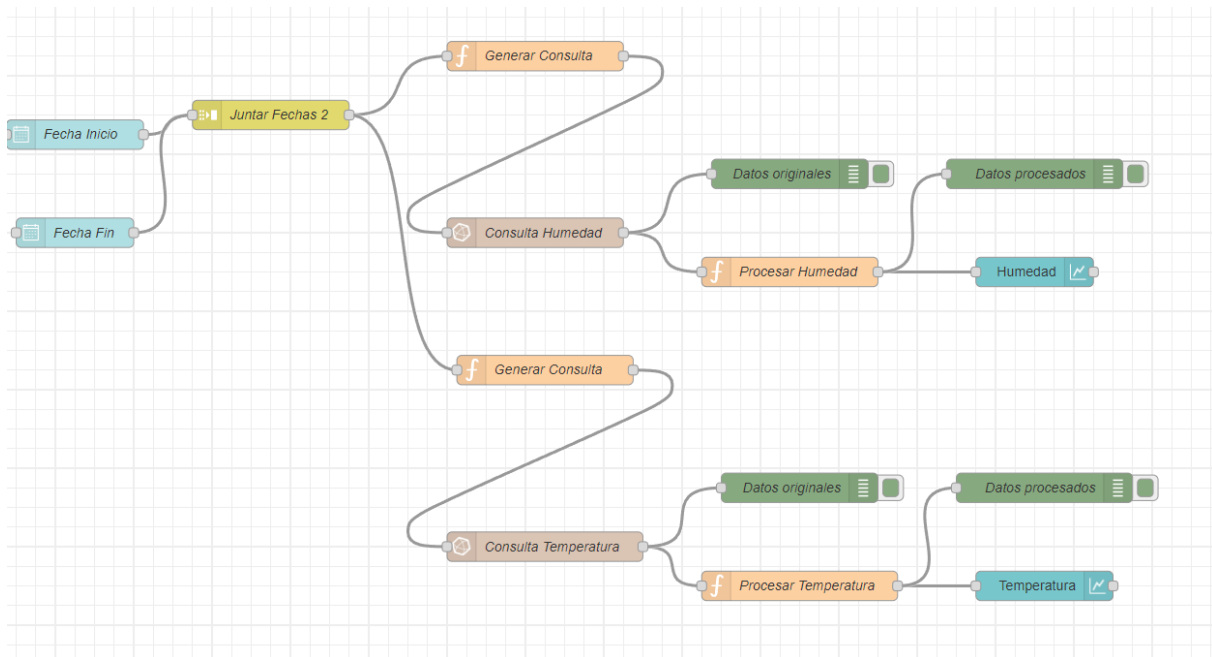


Fig 52. Node-Red. Envío Tópico y mensaje Específico

Este flujo de Node-RED compactado en la Fig 52, está diseñado para enviar mensajes MQTT con un tema específico y un valor asociado, utilizando la información proporcionada por el usuario a través de una interfaz de usuario situado en la Fig 36.

### 6.3.7. FLUJO GRÁFICA TIEMPO EN FORMATO CHART DE DATOS DE SENSORES



**Fig 53. Node-Red. Inclusión de Gráfico por tiempo**

El flujo de Node-RED que se muestra en la Fig 53, está diseñado para mostrar datos de temperatura y humedad específicos provenientes de un invernadero (BANDEJA), filtrados por la dirección MAC proporcionada a través del menú desplegable selector de dispositivos y proporciona al usuario en formato gráfica, los valores almacenados en la base de datos InfluxDB.

## 7. INTEGRACIÓN Y PRUEBAS

### 7.1. INTEGRACIÓN DEL HARDWARE Y SOFTWARE

#### Montaje final del sistema

Para asegurar el correcto funcionamiento del sistema, es esencial integrar de manera efectiva tanto el hardware como el software. El montaje final del sistema implica ensamblar y conectar cada componente de hardware de manera precisa. Esto incluye verificar que la ESP32-CAM esté firmemente conectada a la placa de expansión de pines y que todos los sensores y actuadores, como el DHT11, la tira LED y el ventilador, estén correctamente conectados. Además, es crucial configurar la red Wifi para garantizar que la ESP32-CAM se conecte sin problemas. Por último, la configuración del servidor local, que incluye la correcta instalación y configuración de Node-RED e InfluxDB en el PC local. A continuación, se explicará detalladamente cada punto citado anteriormente:

1. **Ensamblajes y conexiones finales:** Asegurar que todos los componentes de hardware estén bien montados y conectados correctamente. Verificar que la ESP32-CAM esté firmemente conectada a la placa de expansión de pines y que todos los sensores y actuadores (DHT11, tira LED y ventilador) estén bien conectados.
2. **Configuración de la red Wi-Fi:** Asegurarse de que la ESP32-CAM se conecte correctamente a la red Wifi del entorno donde se realizará la operación del sistema.
3. **Configuración del servidor local:** Verificar que Node-RED y InfluxDB estén correctamente instalados y configurados en el PC local. Asegurarse de que Node-RED esté suscrito a los temas MQTT correctos y que pueda almacenar datos en InfluxDB.

#### Pruebas iniciales de integración:

Antes de proceder con las fases siguientes del proyecto, es esencial llevar a cabo pruebas iniciales de integración para asegurar el funcionamiento óptimo del sistema. Estas pruebas son fundamentales para garantizar la eficacia y fiabilidad

del sistema en su conjunto. A continuación se presentan las principales pruebas realizadas:

1. **Prueba de comunicación:** Confirmar que la ESP32-CAM puede enviar datos de sensores (temperatura y humedad) a través de MQTT a Node-RED y que estos datos se almacenen en InfluxDB.
2. **Prueba de control:** Verificar que los comandos enviados desde Node-RED (encendido/apagado de la tira LED y el ventilador) lleguen a la ESP32-CAM y que los actuadores respondan adecuadamente.

## 7.2. PRUEBAS FUNCIONALES Y DE RENDIMIENTO EN DIFERENTES CONDICIONES

### PRUEBAS DE ESTABILIDAD Y RENDIMIENTO:

1. **Pruebas prolongadas:** Ejecutar el sistema durante un período prolongado en torno a 24 horas para asegurar la estabilidad del hardware y el software. Monitorizar las lecturas de temperatura y humedad, así como el comportamiento de la tira LED y el ventilador.
2. **Evaluación de la conectividad:** Verificar la fiabilidad de la conexión Wi-Fi durante las pruebas prolongadas. Asegurarse de que no haya caídas de conexión que puedan afectar el monitoreo y control del sistema.

### Pruebas en condiciones variables:

1. **Simulación de cambios de temperatura y humedad:** Introducir cambios controlados en la temperatura y humedad del ambiente de cultivo para evaluar la respuesta del sistema.
2. **Control de iluminación:** Variar los períodos de encendido de la tira LED para evaluar su impacto en el crecimiento de las plantas y la respuesta del sistema de control.

**Pruebas de robustez del hardware:**

1. **Pruebas de durabilidad:** Evaluar la durabilidad del hardware, especialmente de la ESP32-CAM y los actuadores. Simular situaciones de estrés, como cortes de energía, y observar cómo el sistema se recupera.
2. **Evaluación del consumo de energía:** Medir el consumo de energía del sistema completo para asegurar que sea eficiente y sostenible. Verificar la capacidad de la batería portátil para mantener el sistema operativo durante cortes de energía.

**Documentación de resultados:**

1. **Registro de datos:** Documentar todas las lecturas de temperatura y humedad, así como las imágenes registradas.

**Ejemplo de gráficos y análisis:**

- **Gráfico de temperatura y humedad:** Utilizar los datos almacenados en InfluxDB para crear gráficos de línea que muestren la evolución de la temperatura y la humedad a lo largo del tiempo.
- **Análisis de rendimiento:** Evaluar el tiempo de respuesta del sistema a los comandos de control y la estabilidad de la conexión Wi-Fi.

## 8. CONCLUSIONES

El objetivo principal de este proyecto fue diseñar y desarrollar un sistema automatizado para el cultivo hidropónico, capaz de monitorear y controlar las condiciones ambientales para optimizar el crecimiento de las plantas. Desde la fase inicial de investigación hasta la implementación final, cada etapa del proyecto se llevó a cabo meticulosamente para asegurar el cumplimiento de los objetivos establecidos.

Los resultados obtenidos han sido satisfactorios, cumpliendo con las expectativas planteadas al inicio del proyecto. El sistema desarrollado ha demostrado ser eficaz en el monitoreo y control de variables críticas como la temperatura, la humedad y la iluminación, utilizando la ESP32-CAM como núcleo del sistema. La integración con Node-RED e InfluxDB permitió una gestión eficiente y en tiempo real de los datos recogidos por los sensores, lo que facilitó un control preciso y automatizado de las condiciones del invernadero.

Durante el desarrollo del proyecto, se enfrentaron diversos desafíos técnicos, como el diseño de piezas en 3D para incluir un modelo compacto y útil para la inclusión de la ESP32-CAM y la extensión de pines con el módulo de relés y el permitir la funcionalidad de añadir dispositivos nuevos en el entorno de trabajo, siendo este un aspecto clave y complejo a la hora de trabajar en la programación del mismo. Sin embargo, mediante un proceso continuo de pruebas y ajustes, se lograron implementar soluciones efectivas que garantizaron el correcto funcionamiento del sistema.

El impacto de este proyecto en el ámbito de la agricultura hidropónica es positivo. La automatización de los procesos de monitoreo y control no solo facilita el manejo de cultivos, sino que también puede contribuir a un uso más eficiente de los recursos y a una mejora en la calidad de los productos cultivados. Los resultados obtenidos pueden servir como base para futuras investigaciones y desarrollos en sistemas de cultivo hidropónico, potenciando la innovación en este campo.

A nivel personal y profesional, este proyecto ha sido una oportunidad muy positiva para desarrollar habilidades técnicas en programación, electrónica y diseño de sistemas automatizados. Además, ha fomentado el aprendizaje de técnicas de

investigación y resolución de problemas, fundamentales para cualquier ingeniero. La experiencia adquirida será de gran utilidad en futuros proyectos y desafíos profesionales.

De cara al futuro, se identifican varias áreas que podrían beneficiarse de investigaciones adicionales. Por ejemplo, la implementación de sensores más avanzados para medir otros parámetros importantes del cultivo, o la integración de técnicas de inteligencia artificial para un control aún más preciso y autónomo del sistema. También sería interesante explorar la escalabilidad del sistema para su aplicación en instalaciones más grandes o en otros tipos de cultivos.

En definitiva, el proyecto ha sido exitoso en el cumplimiento de sus objetivos, demostrando la viabilidad y eficacia de un sistema automatizado para el cultivo hidropónico. Los aprendizajes obtenidos sientan una base sólida para futuras motivaciones y aplicaciones prácticas en la agricultura, promoviendo una producción más eficiente y sostenible.

## 9. REFERENCIAS

- [1] B. Hydroponics, «Introduction to Hydroponics». Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://briohydroponics.com/blog/introduction-to-hydroponics>
- [2] S. Khan, A. Purohit, y N. Vadsaria, «Hydroponics: current and future state of the art in farming», *J. Plant Nutr.*, vol. 44, pp. 1-24, dic. 2020, doi: 10.1080/01904167.2020.1860217.
- [3] «espressif/esp32-camera». Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://github.com/espressif/esp32-camera>
- [4] «DHT11, DHT22 and AM2302 Sensors», Adafruit Learning System. Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://learn.adafruit.com/dht/overview>
- [5] «Arduino - Home». Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://www.arduino.cc/>
- [6] «Node-RED». Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://nodered.org/>
- [7] «InfluxDB | Real-time insights at any scale», InfluxData. Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://www.influxdata.com/home/>
- [8] «shiftr.io». Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://www.shiftr.io/>
- [9] «PrusaSlicer | Impresoras 3D Original Prusa vendidas directamente por Josef Prusa». Accedido: 12 de junio de 2024. [En línea]. Disponible en: [https://www.prusa3d.com/es/pagina/prusaslicer\\_424/](https://www.prusa3d.com/es/pagina/prusaslicer_424/)
- [10] N. O'Leary, «knolleary/pubsubclient». 11 de junio de 2024. Accedido: 12 de junio de 2024. [En línea]. Disponible en: <https://github.com/knolleary/pubsubclient>
- [11] «asgarcia/hidroponico-nodered», GitHub. Accedido: 17 de junio de 2024. [En línea]. Disponible en: <https://github.com/asgarcia/hidroponico-nodered>
- [12] «TTGO-Camer-Mic/main/www/index\_ov2640.html.gz at master · Xinyuan-LilyGO/TTGO-Camer-Mic», GitHub. Accedido: 17 de junio de 2024. [En línea]. Disponible en: [https://github.com/Xinyuan-LilyGO/TTGO-Camer-Mic/blob/master/main/www/index\\_ov2640.html.gz](https://github.com/Xinyuan-LilyGO/TTGO-Camer-Mic/blob/master/main/www/index_ov2640.html.gz)
- [13] «Apache License, Version 2.0». Accedido: 17 de junio de 2024. [En línea]. Disponible en: <https://www.apache.org/licenses/LICENSE-2.0>

## 10. ANEXOS

### 10.1. INSTALACIÓN NODE-RED

Antes de instalar Node-RED, es necesario asegurarse de tener instalado **Node.js** en el sistema. Se puede verificar si ya está instalado ejecutando el siguiente comando en el terminal:

```
node -v
```

Si no está instalado, se deberá descargar e instalar desde la página oficial de Node.js: [6]

Para instala Node-RED globalmente en el sistema, se deberá ejecutar el siguiente comando en el terminal:

```
sudo npm install -g --unsafe-perm node-red
```

Una vez completada la instalación, se inicia Node-RED con el siguiente comando:

```
node-red
```

Esto iniciará el servidor de Node-RED y se podrá acceder a la interfaz web desde el navegador en la dirección: <http://localhost:1880>.

## 10.2. INSTALACIÓN INFLUXDB

Es importante asegurarse de tener acceso a un sistema operativo Windows y privilegios de administrador para realizar la instalación.

Para la instalación de InfluxDB, será necesario instalar el instalador de InfluxDB para Windows desde la página oficial: [7] así como ejecutarlo y seguir las instrucciones para su instalación.

Una vez instalado, se abre una ventana de terminal (CMD o PowerShell) y navega al directorio de instalación de InfluxDB.

Se podrá iniciar InfluxDB con el siguiente comando:

```
influxd.exe
```

Para acceder al shell de InfluxDB y crear una base de datos:

```
influx
```

Dentro del shell de InfluxDB, crea una base de datos con:

```
CREATE DATABASE database
```

Reemplazando "database" con el nombre de tu base de datos.

## 10.3. INSTALACIÓN DEL BROKER MQTT SHIFTR.IO

Será necesario visitar la web oficial del sitio: [8] donde se deberá registrar una cuenta y crear un espacio de proyecto. Aquí obtendremos la URL y las credenciales username y password que indicaremos en nuestro bróker instalado. Este bróker permite una visualización rápida y sencilla de los tópicos y elementos suscritos así como una visualización del contenido de los tópicos enviados y recibidos como se puede ver en la imagen: Fig 12. Ejemplo de funcionamiento de shiftr.io

#### 10.4. INCLUSIÓN DE DATOS WIFI, BROKER Y DIRECTORIO DE ALMACENAMIENTO LOCAL

Para una mayor facilidad de uso del sistema y ser capaz de modificar la SSID y CONTRASEÑA de la red Wi-Fi así como la IP y el PUERTO del BROKER, como los directorios donde se almacenarán las imágenes en el servidor local, se hace uso de la lectura de un .txt como se puede apreciar en la Fig 54, alojado en una microSD conectada en la ESP32-CAM. Así, siguiendo la siguiente estructura se podrá elegir cada valor independiente y almacenar en el .txt llamado "data.txt".

```
//-----DATOS INICIO ESP32-CAM-----//  
  
SSID = "vodafone1928"  
PASSWORD = "ZGZYQJN4AJNZQ2"  
MQTT_SERVER = "192.168.0.14"  
MQTT_PORT = "1883"  
IP_LOCAL = "192.168.0.14"  
DIRECTORIO = "C:\\Users\\David\\.node-red\\projects\\hidroponia\\ui-media\\lib\\images_tfg\\"  
DIRECTORIOULTIMO = "C:\\Users\\David\\.node-red\\projects\\hidroponia\\ui-media\\lib\\images_tfg\\"  
NOMBRE_DISPOSITIVO = "BANDEJA1"  
|
```

Fig 54. Datos .txt almacenado en microSD

#### 10.5. CÓDIGO PROGRAMACIÓN ESP32-CAM EN ENTORNO IDE ARDUINO

##### 10.5.1. MAIN PRINCIPAL

El desarrollo principal del código ha sido desarrollado en entorno IDE Arduino. Para ello se han hecho uso de principales configuraciones de uso de programación en C y C++ y el MAIN principal se encuentra en el siguiente enlace de repositorio en GITHUB: <https://github.com/asgarcia/hidropnico-arduino>

**10.5.2. INCLUSIÓN DE LIBRERÍAS**

El MAIN principal requiere de unas librerías para el correcto funcionamiento del programa como conexionado de pines, índices de cámara [12] y la conexión de la WEBCAM por HTTP [13]. Tanto el MAIN principal como las dichas librerías también se encuentran en el siguiente enlace de repositorio de GITHUB: <https://github.com/asgarcia/hidroponico-arduino>

**10.6. CÓDIGO PROGRAMACIÓN DE NODE-RED EN JSON**

El servidor Local y el entorno web ha sido creado con Node-Red. El JSON exportado del código se encuentra alojado en el repositorio siguiente de GITHUB: <https://github.com/asgarcia/hidroponico-nodered>