**ESCUELA DE INGENIERÍAS INDUSTRIALES**

**ARQUITECTURA DE COMPUTADORES**

**ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES**

**TRABAJO FIN DE GRADO**

DISPOSITIVO INTELIGENTE PARA MONITORIZACIÓN INALÁMBRICA DE PILA DE COMPOSTAJE

SMART WIRELESS COMPOST PILE MONITORING SYSTEM

Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Autor: **Javier Gil Palacios**

Tutor: **Andrés Rodriguez Moreno**

Cotutor: **Rubén Delgado Escaño**

MÁLAGA, SEPTIEMBRE de 2023

**RESUMEN**

Este proyecto surge por la necesidad de automatizar, monitorizar y controlar de forma eficiente todo el proceso de tratamiento y creación del compost de la estación ubicada en el huerto docente de la Universidad de Ciencias de Málaga.

Se ha fabricado un dispositivo portátil e inalámbrico, alimentado mediante baterías recargadas por el sol, este se introduce en una pila de compostaje a monitorizar, cada dispositivo consta de tres sensores de temperatura y un sensor de humedad. El objetivo es recopilar datos a distintos niveles de profundidad dentro de la pila de compost, con los datos de humedad se gestiona el aporte de agua para mantener las condiciones óptimas y con los datos de temperatura se podrá conocer cuando se ha alcanzado la fase de volteo de la pila, en la cual, se reactiva el proceso de compostaje al aportar oxígeno.

El dispositivo estará basado en microcontroladores con conexión WiFi ESP8266 y ESP32, los cuales usarán el protocolo de comunicación ESP-NOW para ser lo más eficiente posible en el consumo de energía.

La gestión del dispositivo y datos se integrarán al sistema existente en el huerto docente de Ciencias en el Campus. Esta solución permitirá un proceso de compostaje de forma eficiente y controlada.

**PALABRAS CLAVES**

Compostaje, Sostenibilidad, Microcontroladores, ESP-Now, MQTT, Monitorización, IoT.

**Abstract**

Contenido

[Índice de figuras 6](#_Toc143367766)

[Referencias 6](#_Toc143367767)

[Capítulo 1. Introducción 7](#_Toc143367768)

[1.1 Antecedentes 7](#_Toc143367769)

[1.2 Proceso de compost 7](#_Toc143367770)

[1.3 Objetivos 8](#_Toc143367771)

[Capítulo 2. Fases del trabajo 8](#_Toc143367772)

[2.1 Diseño, elección del Hardware y materiales 9](#_Toc143367773)

[2.2 Estudio de protocolos de comunicación de los distintos dispositivos 10](#_Toc143367774)

[2.2.1 ESP-Now 10](#_Toc143367778)

[2.2.2 MQTT 12](#_Toc143367779)

[2.3 Diseño del esquema eléctrico y conexionado del sistema 13](#_Toc143367780)

[2.4 Diseño del software y gestión de envío de datos 15](#_Toc143367781)

[2.4.1 Programación entre estaciones móviles ESP8266 y estación fija ESP32. 16](#_Toc143367782)

[2.4.2 Programación entre el ESP32 y el servidor Web. 17](#_Toc143367783)

[2.5 Diseño y programación de la Interfaz de monitorización 17](#_Toc143367784)

[Capítulo 3. Software y hardware empleado 18](#_Toc143367785)

[3.1 Software 18](#_Toc143367787)

[3.1.1 Arduino 18](#_Toc143367788)

[3.1.2 Node-RED 18](#_Toc143367789)

[3.2 Hardware 19](#_Toc143367790)

[3.2.1 ESP8266 19](#_Toc143367791)

[3.2.2 ESP32 20](#_Toc143367792)

[3.2.3 Sensores 21](#_Toc143367793)

[3.2.3.1 Sensor digital de temperatura DS18B20 21](#_Toc143367797)

[3.2.3.2 Sensor de humedad capacitivo soil moistureV1.0 23](#_Toc143367798)

[Capítulo 4. Desarrollo del software 24](#_Toc143367799)

[4.1 Software ESP32 24](#_Toc143367804)

[4.2 Software ESP8266 25](#_Toc143367805)

[4.3 Node-RED 29](#_Toc143367806)

[Capítulo 5. Resultados y simulaciones 30](#_Toc143367807)

[Capítulo 6. Conclusiones y líneas futuras 30](#_Toc143367808)

[Anexos 30](#_Toc143367809)

# Índice de figuras

[Figura 1.Diseño de La Pica de Toma de Datos. [1] 13](#_Toc143367813)

[Figura 2. Conexionado de los Dispositivos de La Estación Móvil. [2] 14](#_Toc143367814)

[Figura 3. Diseño red ESP-Now 16](#_Toc143367815)

[Figura 4. Comunicación mqtt ESP32 con servidor web 17](https://uma365-my.sharepoint.com/personal/0619117020_uma_es/Documents/tfg2023/TFG-JAVIER-GIL-PALACIOS-ESTACION-DE-COMPOST/Documentación/Dispositivo%20inteligente%20para%20monitorización%20inalámbrica%20de%20pila%20de%20compostaje.docx#_Toc143367816)

[Figura 5. Logo Arduino. [3] 18](#_Toc143367817)

[Figura 6. Logo Node-RED. [4] 19](#_Toc143367818)

[Figura 7. Sonda sensor de temperatura DS18B20 22](#_Toc143367819)

[Figura 8. Sonda sensor de humedad Capacitive Soil Moisture v1.0 23](#_Toc143367820)

# Referencias

Arduino. (2023). *Arduino Home*. Obtenido de https://www.arduino.cc/

ESP-Now, E. S. (2023 ). *esp-now*. Obtenido de https://www.espressif.com/en/solutions/low-power-solutions/esp-now

MQTT, I. (2023). *MQTT*. Obtenido de https://mqtt.org/

Node-RED. (2023). *Node-RED*. Obtenido de https://nodered.org/

Randomnerdtutorials. (2013-2023). *Randomnerdtutorials*. Obtenido de https://randomnerdtutorials.com/projects-esp8266/

Systems, Espressif. (2023). Obtenido de http://esp32.net/

# Capítulo 1. Introducción

## Antecedentes

En la era actual, el desarrollo de prácticas sostenibles se ha vuelto estrictamente necesario para hacer frente a los desafíos ambientales a los que se enfrenta nuestro planeta. La economía circular es esencial en este contexto, en ella se busca reducir al mínimo el desperdicio y maximizar la utilización de recursos, haciendo frente al modelo lineal tradicional de "tomar, hacer y desechar", la economía circular fomenta la reutilización, el reciclaje y la regeneración, con el objetivo de mantener los materiales en uso durante el mayor tiempo posible.

En este contexto, el compostaje emerge como un método valioso para cerrar el ciclo de vida de los materiales orgánicos. Transformar los residuos orgánicos en compost beneficia la fertilidad del suelo, reduce la acumulación de desechos y fortalece la producción de alimentos. La combinación de estos enfoques contribuye a la conservación de recursos, la reducción de la contaminación y la creación de sistemas más resistentes y equilibrados, asegurando un entorno más saludable y sostenible para el futuro.

## Proceso de compost

El compostaje es un proceso natural de deterioro de materiales orgánicos, como restos de alimentos y residuos vegetales, en condiciones controladas. Se lleva a cabo mediante la acción de microorganismos, como bacterias y hongos, que descomponen los materiales en compuestos más simples y estables. El proceso de compostaje implica la creación de un entorno adecuado para que estos microorganismos prosperen, he aquí la importancia de llevar a cabo este proyecto. A continuación, se describen las fases del proceso de generación de compost.

1. Preparación: Se toman los materiales orgánicos adecuados, como restos de frutas, vegetales, hojas, recortes de césped y se mezclan para lograr una combinación equilibrada de carbono y nitrógeno en la descomposición.
2. Descomposición: Crecen microorganismos, como bacterias y hongos, los cuales, comienzan a descomponer los materiales orgánicos, como resultado genera calor y libera dióxido de carbono y vapor de agua.
3. Mezcla y Volteo: Es importante voltear o mezclar los materiales para asegurar que haya una distribución uniforme de oxígeno y humedad. Este proceso acelera la descomposición y evita la formación de malos olores.
4. Maduración: Con el tiempo, los materiales se descomponen en una sustancia oscura y rica en nutrientes llamada compost. Este proceso puede durar desde varias semanas a meses, dependiendo de las condiciones y los materiales utilizados.
5. Cribado: Una vez que el compost está listo, se puede cribar para eliminar objetos no deseados y obtener un producto uniforme y fino.

El resultado final del compostaje es un producto valioso que puede usarse para enriquecer el suelo en jardines, huertos y áreas verdes, además aumenta la capacidad de retención de agua, proporcionando nutrientes esenciales a las plantas.

## Objetivos

El objetivo principal del proyecto es conocer a través de los índices de temperatura y humedad la fase en la que se encuentra el proceso de compostaje de la pila, con el fin de controlar de manera eficiente la generación del compost y estudiar las propiedades en las mezclas de los materiales orgánicos.

# Capítulo 2. Fases del trabajo

A continuación, se enumeran y detallan las distintas etapas que se han seguido en la realización del proyecto.

## Diseño, elección del Hardware y materiales

En primer lugar, se realiza un estudio de la zona donde serán alojados los equipos de medición y toma de datos, así mismo, se pretende encontrar la forma más adecuada atendiendo a la sostenibilidad y eficiencia para llevar a cabo el montaje. Por ello, se utilizan microcontroladores con pequeños consumos eléctricos con el fin de poder alimentarlos por medio de placas solares y baterías.

Respecto a la elección del material, se han buscado sensores de temperaturas y de humedad compuestos por materiales resistentes, con la finalidad de que perduren en el tiempo y sean de fácil integración en el sistema.

Pensando en la posibilidad de controlar y monitorizar múltiples pilas de compost, se diseña un sistema que podemos dividir en dos partes:

1. Estación fija principal: Este es el dispositivo principal encargado de recibir la información de varias estaciones móviles, actúa de pasarela con el servidor del huerto docente de la Universidad de Ciencias, comunicándose y transmitiendo los datos de todas las estaciones móviles para posteriormente monitorizarlos y almacenarlos en la base de datos. Está compuesto por un microcontrolador ESP32 alimentado por una fuente de tensión externa de 5v. Todo el tratamiento de datos se realiza por comunicación inalámbrica sin un solo cable.
2. Estación móvil: Este dispositivo se encarga de tomar las medidas de los sensores y enviarlas a la estación fija. Por otro lado, se ha diseñado y fabricado el prototipo de una estación, puede verse en el anexo de imágenes (imagen). Estos equipos se introducen en las distintas pilas de compost alojadas en el huerto. La estación móvil está compuesta por:
   1. Pica de toma de datos: consiste en una varilla compuesta por un sensor de humedad y tres sensores de temperatura, los cuales están situados a distinta distancia. La pica irá soterrada en la pila de compost y transmitirá los datos de todos los sensores por un único cable al módulo de control, puede verse en el anexo de imágenes (imagen).
   2. Módulo de control: Consiste en una caja estanca compuesta por:
      * Microprocesador ESP8266 con antena WIFI: Se alimenta mediante una batería recargada con placas solares. El dispositivo se ha programado para comunicar y transmitir de forma inalámbrica los datos de los sensores a la estación fija.
      * Sistema de alimentación: Se encarga de alimentar tanto al ESP8266 como a los sensores. Se compone de:
        1. Placas solares de 5w.
        2. Batería de 1200 mAh, 4,44w y 3,7v.
        3. Circuito electrónico de control de carga de batería y estabilizador de tensión.
      * Pulsador (NO): Su función, al pulsar, es el envío de datos de temperatura y humedad de forma instantánea.

Puede verse en el anexo de imágenes (imagen)

La información detallada de todos estos dispositivos mencionados queda reflejada en el capítulo 2, en el apartado 2.2 Hardware.

## Estudio de protocolos de comunicación de los distintos dispositivos

Los microcontroladores ESP32 y ESP8266 se caracterizan por su versatilidad a la hora de realizar comunicaciones y por admitir varios protocolos de comunicación. A continuación, se nombran y explican los protocolos de comunicación utilizados en el proyecto.



## ESP-Now

Es un protocolo de comunicación inalámbrica desarrollado por Espressif Systems. El principal propósito es permitir la comunicación directa y eficiente entre dispositivos con microcontroladores ESP8266 o ESP32 en un rango cercano sin la necesidad de un enrutador Wi-Fi. Como principal característica de funcionamiento, este protocolo ofrece una comunicación de baja latencia y bajo consumo de energía, gracias a que trabaja en la capa de enlace de datos, permitiendo que los dispositivos se comuniquen directamente entre sí, utilizando direcciones MAC únicas, esto hace que sea ideal para el proyecto. Sus características de funcionamiento en términos generales son:

* Creación de nodos: En una red ESP-Now, hay al menos dos tipos de dispositivos: el "nodo maestro" y los "nodos secundarios". El nodo maestro es responsable de coordinar la comunicación (en nuestro caso es el ESP32), mientras que los nodos secundarios envían y reciben datos (en nuestro caso el ESP8266).
* Se establecen parejas de dispositivos: Los dispositivos ESP8266 o ESP32 deben ser emparejados antes de poder comunicarse entre sí utilizando ESP-Now. Esto implica que los nodos secundarios deben registrarse en el nodo maestro, estableciendo una relación de comunicación.
* Identificación MAC Address e ID: Cada dispositivo en la red ESP-Now tiene una dirección MAC única. El nodo maestro y los nodos secundarios utilizan estas direcciones para identificarse entre sí. Además, los nodos secundarios pueden tener un identificador único asignado por el usuario.
* Publicación y suscripción: ESP-Now utiliza un modelo de comunicación de "publicación-suscripción". Un nodo secundario puede publicar datos, y el nodo maestro reenviará esos datos a otros nodos secundarios que estén interesados en recibirlos. Los nodos secundarios pueden "suscribirse" a ciertos tipos de datos o dispositivos específicos.
* Transmisión de paquetes: Los datos se encapsulan en paquetes de datos y se envían mediante el protocolo ESP-Now. El nodo maestro tiene la tarea de reenviar los paquetes entre los nodos secundarios según sus suscripciones. Permitiendo una comunicación eficiente y rápida entre los dispositivos.
* Manejo de errores y retransmisiones: ESP-Now ofrece confiabilidad en la transmisión de datos. Si un paquete se pierde o se corrompe durante la transmisión, se pueden implementar mecanismos de retransmisión para garantizar que los datos lleguen correctamente e identificar nodos que han dejado de funcionar.

Para más información puede acceder a (ESP-Now, 2023 )

## MQTT

Fue desarrollado por IBM en la década de 1990, MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación eficiente y fiable, utilizado para enviar mensajes entre dispositivos en redes con ancho de banda limitado, es popularmente utilizado para las comunicaciones en el mundo del internet de las cosas (IoT).

Los tres elementos principales que forman este protocolo de comunicación son:

* Broker: Es un servidor central que actúa como intermediario entre los clientes. Gestiona la comunicación entre los clientes, recibiendo mensajes de los publicadores y reenviándolos a los suscriptores adecuados, además asegura la entrega según el nivel de calidad de servicio (QoS) especificado.
* Clientes MQTT: Son todos los dispositivos que se comunican a través de MQTT, tienen un identificador único denominado “Cliente ID”. Pueden ser:
  + Publicadores: Son los dispositivos que envía un mensaje a un topic específico en el bróker, este mensaje, puede tener diferentes niveles de calidad de servicio (QoS) para garantizar la entrega.
  + Suscriptores: Son los dispositivos que tras suscribirse en el bróker e indicar qué topics les interesan, reciben los mensajes e información contenida en dichos topics.
* Topics: Son cadenas de texto que se utilizan para organizar y enrutar los mensajes entre los dispositivos. Se organizan jerárquicamente, similar a una estructura de carpetas.

Para más información sobre el protocolo de comunicación MQTT pueden acceder al siguiente enlace (MQTT, 2023)

## Diseño del esquema eléctrico y conexionado del sistema

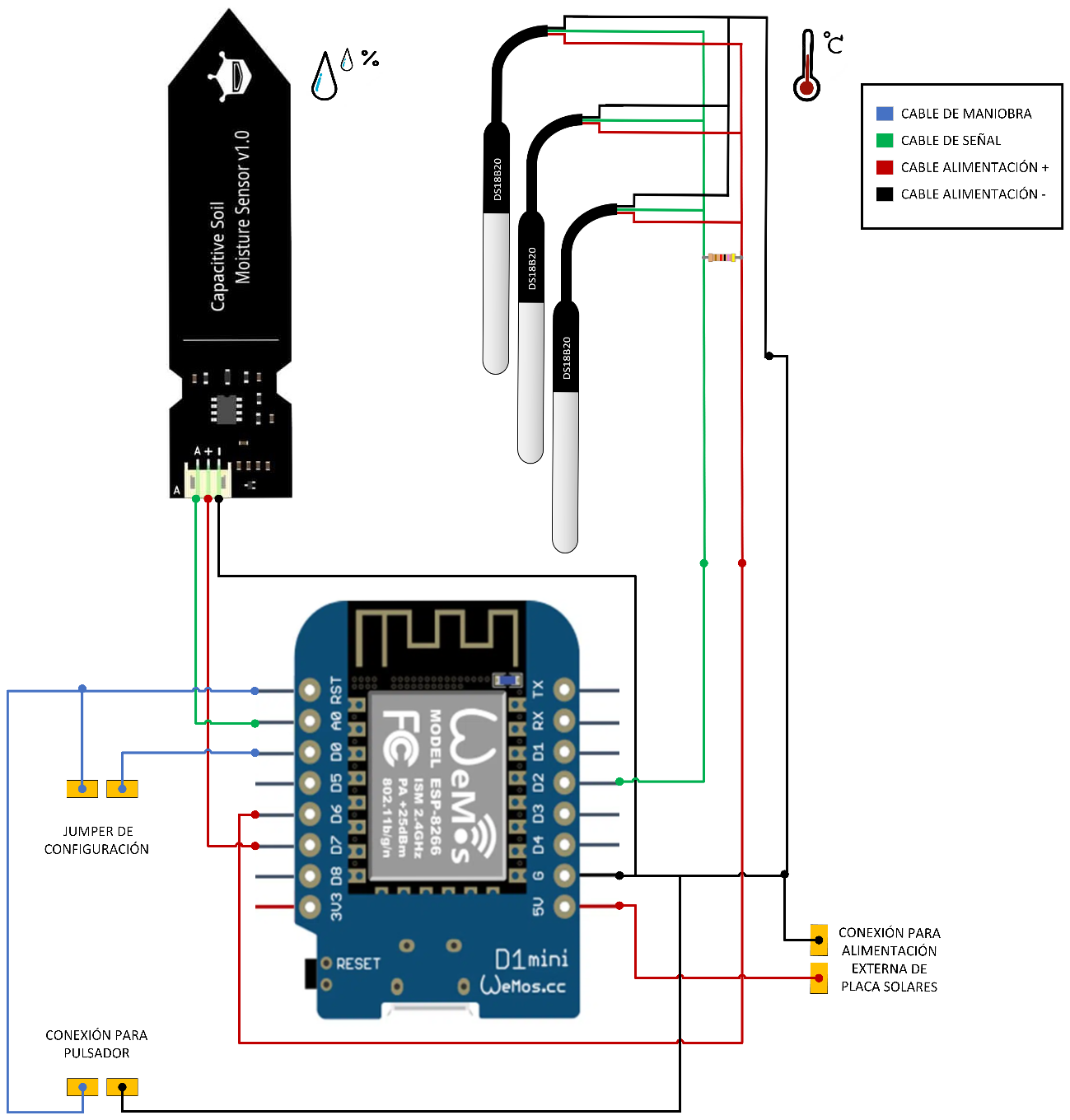
En primer lugar, se analiza la configuración y conexionado de la pica de toma de datos (ver *Figura 1*).

Diagrama

Descripción generada automáticamente

Figura 1.Diseño de La Pica de Toma de Datos. [1]

En la *Figura 2*, se muestra el esquema eléctrico del módulo de control de la estación móvil, este circuito, ha sido diseñado y fabricado por el alumno en una placa electrónica, la cual, se puede ver en el anexo (imagen de la placa).



Resistencia pull-up 4,7 KΩ

Figura 2. Conexionado de los Dispositivos de La Estación Móvil. [2]

## Diseño del software y gestión de envío de datos

La estructuración del proyecto a nivel de diseño de software y protocolos de comunicación se puede dividir en dos partes.

## Programación entre estaciones móviles ESP8266 y estación fija ESP32.

Diagrama

Descripción generada automáticamente

Figura 3. Diseño red ESP-Now

## Diagrama Descripción generada automáticamenteProgramación entre el ESP32 y el servidor Web.

Figura 4. Comunicación mqtt ESP32 con servidor web

## Diseño y programación de la Interfaz de monitorización

# Capítulo 3. Software y hardware empleado

En esta sección, detallaremos el software y hardware seleccionados para alcanzar los requisitos y objetivos del proyecto, garantizando un rendimiento óptimo.



## Software

En la realización del proyecto se han utilizado dos softwares diferentes y un sistema de gestión de bases de datos gestionado a través de Node-RED, llamado MongoDB.

## Arduino

El software de Arduino es una herramienta que ha tomado gran importancia en el mundo de la electrónica y la programación, ya que además de ser open source, ha sido diseñado para ser accesible y amigable, con una interfaz intuitiva, permite escribir, cargar y depurar código en microcontroladores de manera eficiente. Brinda la flexibilidad necesaria para crear proyectos complejos de automatización y robótica. Para más información el lector puede dirigirse a la siguiente web (Arduino, 2023).



Figura 5. Logo Arduino. [3]

## Node-RED

Node-RED es una plataforma gráfica de programación, cuya interfaz de usuario corre a través de un navegador. Los usuarios pueden arrastrar y soltar nodos prediseñados con funciones específicas para construir flujos que pueden proporcionar soluciones complejas, tanto en la recopilación de datos, como en la toma de decisiones y ejecución de acciones. Node-RED se ha convertido en una gran herramienta para la creación rápida de prototipos y la implementación de proyectos en el ámbito del internet de las cosas (IoT).

Los factores que han llevado a utilizar esta herramienta han sido:

* + - * Interfaz visual intuitiva: Su entorno gráfico facilita la creación y gestión de flujos de datos y lógica.
      * Rápido prototipado: Facilita la creación rápida de prototipos y pruebas de concepto para proyectos IoT y automatización.
      * Integración con plataformas: Puede conectarse a diversas plataformas y servicios populares, mejorando la interoperabilidad.
      * Código abierto: Al ser de código abierto, es gratis, accesible y adaptable a diferentes entornos y requisitos.

Para más información pueden acceder a la web (Node-RED, 2023).

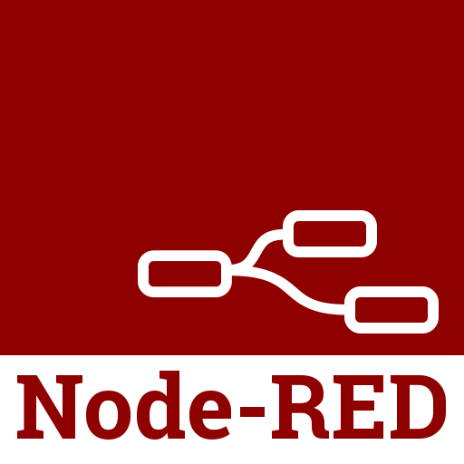


Figura 6. Logo Node-RED. [4]

## Hardware

La realización del proyecto se ha llevado a cabo principalmente con los dispositivos expuestos a continuación.

## ESP8266

El ESP8266 es un microcontrolador que ha evolucionado hasta convertirse en una plataforma completa de desarrollo. Entre sus características principales, destacamos:

* + - * Conectividad Wi-Fi integrada: Incluye un módulo Wi-Fi que permite la conexión a redes inalámbricas, facilitando la comunicación y la transmisión de datos.
      * Procesador de bajo consumo: Tiene un tamaño compacto y cuenta con un procesador eficiente que equilibra el rendimiento y el consumo de energía.
      * Amplia gama de pines: Ofrece una variedad de pines GPIO (Entrada/Salida de Propósito General) que permiten la conexión y control de diversos componentes electrónicos.
      * Memoria flash integrada: Tiene memoria flash incorporada que se utiliza para almacenar el firmware, programas y datos.
      * Modos de suspensión: Ofrece modos de suspensión para ahorrar energía cuando no se requiere una operación activa.
      * Comunicación serial: Admite comunicación serial (UART) para la interacción con otros dispositivos.
      * Interfaz I2C y SPI: Ofrece interfaces I2C y SPI que facilitan la conexión a sensores y otros dispositivos.
      * Soporte para servidor web: Puede funcionar como un servidor web, lo que permite crear interfaces de usuario y aplicaciones en línea.
      * Programación versátil: Se puede programar utilizando lenguajes y entornos como el Arduino.

Todo ello, ha hecho que nos hayamos decantado por este dispositivo para crear el sistema de toma de datos portátil que recoge la información de los sensores y se comunica de forma inalámbrica a través de la pasarela para transmitir los datos con el ESP32. En la siguiente web podrán consultar todo tipo de información del dispositivo, así como ejemplos de diversos proyectos (Randomnerdtutorials, 2013-2023).

## ESP32

El ESP32 es la evolución del microcontrolador ESP8266, ha sido diseñado para ofrecer un mayor rendimiento y capacidades de conectividad más amplias. Este dispositivo integra un potente procesador de doble núcleo, junto con conectividad Wi-Fi y Bluetooth, lo que lo convierte en una herramienta ideal para programar y gestionar aplicaciones de Internet de las cosas (IoT) y proyectos electrónicos. Las principales características que lo diferencian del ESP8266 son:

* + - * Doble núcleo: Ofrece un procesador de doble núcleo que permite realizar múltiples tareas de manera eficiente.
      * Wi-Fi y Bluetooth: Integra conectividad Wi-Fi y Bluetooth clásica y de baja energía (BLE), lo que facilita la comunicación inalámbrica.
      * Potencia de procesamiento: Su procesador más rápido permite un mejor rendimiento en comparación con el ESP8266.
      * Memoria Flash y RAM: Cuenta con mayor capacidad de memoria flash y RAM para almacenamiento y ejecución de programas.
      * Soporte para pantallas y sensores: Puede manejar pantallas y sensores de manera eficiente, lo que lo hace adecuado para aplicaciones multimedia y de detección.
      * Seguridad: Ofrece capacidades de seguridad mejoradas, incluido el cifrado de datos y las funciones de seguridad de hardware.

Toda la información acerca del dispositivo se puede encontrar en la web (Systems, Espressif, 2023).

## Sensores

El sistema se compone de tres sensores de temperatura DS18B20 y un sensor de humedad capacitivo, las características principales se analizan a continuación.



## Sensor digital de temperatura DS18B20

Es un sensor de temperatura altamente preciso, fabricado por Maxim Integrated, utiliza la tecnología de medición digital de temperatura para proporcionar lecturas confiables y exactas mediante. La comunicación la realiza a través del protocolo 1-Wire, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus. Sus características son las siguientes:

* + - * Medición digital: proporciona lecturas digitales precisas de temperatura, lo que elimina la necesidad de tener que realizar conversiones de medidas.
      * Comunicación por un cable: Utiliza un protocolo de comunicación por un cable, lo que facilita la integración en proyectos con limitaciones de cableado.
      * Precisión: Ofrece una alta precisión en las mediciones de temperatura, lo que lo hace adecuado para aplicaciones sensibles a cambios pequeños en la temperatura.
      * Rango amplio de temperatura: Puede medir temperaturas en un rango amplio, esto la hace idóneo para diversas condiciones ambientales.
      * Calibración interna: Incluye una función de calibración interna que mejora la precisión de las mediciones.
      * Alimentación por el Bus: Puede ser alimentado directamente a través del cable de comunicación, simplificando el diseño del circuito.
      * Amplia compatibilidad: Soporta diversas plataformas y microcontroladores, lo que facilita su integración en proyectos.
      * Robusto y duradero: Presenta un encapsulado resistente con gran durabilidad y confiabilidad.

A continuación, se muestra una imagen de la sonda.



Figura 7. Sonda sensor de temperatura DS18B20

## Sensor de humedad capacitivo soil moistureV1.0

Es un dispositivo diseñado para medir la humedad evaluando con alta sensibilidad los cambios en la capacitancia causados ​​por la presencia de agua en el suelo. Presenta una sonda de detección que se inserta en el suelo, diseñada para soportar condiciones ambientales variables y entornos de uso al aire libre. Es un sensor a tres hilos, que trabaja con una tensión que va desde 3,3 v a 5,5 v, la comunicación se realiza mediante una salida analógica.

Por estas características y su fácil integración a la hora de programación y conexión se ha elegido para de toma de medidas de humedad del compost.

En la siguiente imagen podemos ver la sonda capacitiva.



Figura 8. Sonda sensor de humedad Capacitive Soil Moisture v1.0

# Capítulo 4. Desarrollo del software



## Software ESP32

El microcontrolador ESP32 se ha programado para que actúe como nodo intermediador entre los dispositivos ESP8266 y el servidor web del huerto docente. El código consiste en una pasarela ESP-Now a MQTT con autoemparejamiento entre dispositivos. A continuación, se explica de forma detallada las partes principales del código y las acciones llevadas a cabo en cada una de ellas.

* Función principal de configuración “void setup()”

Esta función se ejecuta solamente una vez al inicio del dispositivo. Las acciones que implementa son:

* + Inicializa la comunicación en serie. Se inicia la comunicación con el monitor serie a una velocidad de 115200 baudios para la depuración y visualización de mensajes.
  + Establece la conexión WiFi. Antes de realizar la conexión, ejecuta la función “sprintf(ID\_PLACA, "ESP\_%d", ESP.getEfuseMac())”, con el fin de adquirir la dirección MAC del dispositivo y almacenarla en la variable “ID\_PLACA” para utilizarla en el protocolo de comunicación ESP-Now. A continuación, Se llama a la función “conecta\_wifi” para establecer la conexión WiFi utilizando las credenciales proporcionadas (SSID y contraseña).
  + Realiza tareas de configuración. En este punto, se configura:
    - Configuración del Canal WiFi: Se obtiene el número del canal WiFi en el que se encuentra el dispositivo ESP32 y se almacena en la variable “myChannel”.
    - Configuración del Cliente MQTT: Se configura el cliente MQTT (mqtt\_client) para que se conecte al servidor MQTT en la dirección y puerto especificados. También se establece el tamaño del búfer de mensajes a 512 bytes y se registra la función “procesa\_mensaje()” como callback para manejar los mensajes MQTT entrantes.
  + Configura ESP-NOW. Se llama a la función “initESP\_NOW()” para inicializar la comunicación utilizando el protocolo ESP-NOW. En ella se definen otras funciones y callbacks relacionados con ESP-NOW, como addPeer para agregar dispositivos emparejados, OnDataSent para manejar la confirmación de envío y OnDataRecv para procesar los datos recibidos.
* Función “encuentra\_mensaje()”

Se ejecuta antes de la función principal “void loop()”. Busca un mensaje en la cola de mensajes MQTT para un dispositivo específico. Para ello se ha declarado como un iterador “std::list<TmensajeMQTT>::iterator”, esto indica que la función encuentra\_mensaje(),recorre la lista de mensajes MQTT (TmensajeMQTT), buscando y captando mensajes procedente del dispositivo cuya MAC se le pase como argumento(uint8\_t \*\_mac) para poder procesarlos posteriormente.

* Función principal del programa “void loop()”

Es el bucle principal del programa, en esta función se implementa la pasarela ESP-Now a MQTT. Se encarga de mantener en funcionamiento las comunicaciones MQTT y ESP-NOW, gestionando los dispositivos listos para ser emparejados o los ya emparejados y activos, para enviar o revivir mensajes pendientes a través de ambos protocolos de comunicación.

## Software ESP8266

El programa desarrollado para el dispositivo ESP8266 se divide y analiza en seis secciones:

1. Declaración de librerías

En esta sección se incluyen las librerías necesarias para el programa. Estas librerías proporcionarán las funciones y clases necesarias para la comunicación con los sensores, la conexión a redes WiFi y la actualización del firmware a través de OTA (Over-the-Air). Anexo

1. Declaración de pines de alimentación de los sensores, variables, estructura y objeto

En primer lugar, se definen los pines para controlar la alimentación de los sensores de temperatura y humedad. A continuación, se declaran variables para almacenar los valores de humedad y de temperatura (las variables “S1\_T”, “S2\_T” y “S3\_T” representan las temperaturas individuales de cada sensor, mientras que el array “temp” se utiliza para almacenar los valores de temperatura de los tres sensores).

Se instancia un objeto de la clase AUTOpairing\_t llamado “clienteAP” para manejar la comunicación a través de ESP-NOW.

Se define una estructura llamada “mi\_configuración” que tiene dos campos sleep y timeout, ambos de tipo uint32\_t. La estructura se utiliza para almacenar la configuración del programa, definida por el tiempo de sueño profundo y el tiempo de espera de conexión.

1. Declaración de variables para la actualización vía FOTA (Firmware Over-the-Air)

Aquí se definen variables relacionadas con la actualización del firmware a través de FOTA. Se define el nombre de la red WiFi (ssid) y la contraseña (password) para la conexión, la URL para la actualización del firmware (OTA\_URL) y la versión del firmware (HTTP\_OTA\_VERSION). Además, se crea un objeto WiFiClientSecure llamado “WClient” para la conexión segura con el servidor.

1. Declaración de la función “procesa mensaje”

Es una función compleja, la cual se encarga de procesar los mensajes recibidos a través de ESP-NOW. Está compuesta por dos argumentos:

* Topic: Es el primer campo del mensaje recibido y nos indica cual es el tema.
* Payload: Es la parte del mensaje que contiene los datos.

La función “procesa mensaje” dependiendo del topic del mensaje, se realiza dos acciones:

* La actualización del firmware vía FOTA.
* Configuración y establecimiento de los parámetros de tiempo de sueño profundo y tiempo máximo de establecimiento de conexión.

A continuación, para una mayor comprensión del funcionamiento y del código de la función, se desglosan y analizan los pasos que se ejecutan en la transición del código de forma secuencial ver código anexo. Estos son:

* Imprime información sobre el mensaje recibido, el tema (topic) y el mensaje (payload).
* Comprueba el tema del mensaje, utilizando una estructura de control if-else. Hay dos posibles temas:

1. Si el tema es "fota", significa que se ha recibido un mensaje para realizar una actualización de firmware. En este caso, corre el código de actualización FOTA.
   * Realiza la conexión wifi utilizando la librería “<ESP8266WiFi.h>”. Si no conecta transcurrido el tiempo establecido, el nodo pasa a estado de sueño profundo, si conecta, a través de la clase WClient realiza la actualización llamando a la función “ESPhttpUpdate.update(WClient,OTA\_URL, HTTP\_OTA\_VERSION)”.
   * Si no realiza la conexión el nodo entra en sueño profundo.
2. Si el tema es "config", significa que se recibió un mensaje para actualizar la configuración del dispositivo. En este caso, ejecuta el código correspondiente para realizar la configuración.
   * Se crea un StaticJsonDocument con un tamaño adecuado para cargar el mensaje JSON recibido.
   * Se descompone el mensaje JSON usando “deserializeJson” para convertir el mensaje en un documento JSON manejable.
   * Se comprueba si la deserialización fue exitosa.
   * Se verifica si el documento JSON contiene los campos "sleep" y "timeout". Si están presentes, significa que se recibió una configuración válida.
   * Se extraen los valores de "sleep" y "timeout" del documento JSON y se almacenan en la estructura de configuración “mi\_configuracion”.
   * Finalmente, se llama al método “set\_config” del objeto “clienteAP” para actualizar la configuración con los nuevos valores.
3. Entrada en el bucle de Setup

El código de este bucle se ejecuta en el arranque e inicialización del dispositivo. A continuación, se desarrolla su funcionamiento de forma secuencial ejecutando los siguientes pasos:

* Se configura la alimentación de los sensores de temperatura y humedad mediante los pines “Power\_s\_temp” y “Power\_s\_hum”. Se establece su modo como salida y se encienden inicialmente.
* Se inicia la comunicación serie (puerto serie) con una velocidad de transmisión de 115200 baudios.
* Se imprimen mensajes en la consola para indicar que la configuración está comenzando.
* Se inicializa el tamaño de la configuración del objeto “clienteAP” para que coincida con el tamaño de la estructura “mi\_configuracion”.
* Se verifica si hay una configuración previamente guardada utilizando el método “get\_config” de la clase “clienteAP”. Si no hay configuración guardada, se establecen valores por defecto para “sleep” y “timeout” en la estructura “mi\_configuracion”.
* Se imprimen en la consola los valores de “sleep” y “timeout” obtenidos o establecidos.
* Se configuran parámetros de la clase “clienteAP” utilizando los métodos:

1. “set\_timeOut”, establece el tiempo máximo de conexión.
2. “set\_deepSleep”, establece el tiempo en sueño profundo (en segundos).
3. “set\_channel, set\_debug”, establece el canal donde empieza el escaneo de dispositivos en la red wifi para el autoemparejamiento.
4. “set\_callback”, llama a la función “procesa\_mensajes”.

* Se inicia la comunicación a través de ESP-NOW llamando al método “begin” de “clienteAP”. Ver explicación del método “begin” en el anexo, ya que es una de las funciones principales del proyecto para autoemparejar a los dispositivos.
* Se realiza una adquisición de las direcciones de los sensores de temperatura llamando a la función “adquisicion\_direcciones\_temp” (función perteneciente a la librería funciones.cpp).

1. Entrada en el bucle principal.

En el bucle, se retoma la conexión cada vez que el nodo despierta y se comprueba si hay mensajes disponibles de actualización o configuración del dispositivo. A continuación, obtienen los datos de temperaturas y humedad de los sensores, una vez recibidos deja de alimentar a los sensores y se crea un mensaje con los datos en formato JSON. Luego se utiliza en método “espnow\_send\_check()” de la clase “clienteAP” para enviar el mensaje a través de ESP-NOW. Después de enviar los datos, el dispositivo entra en modo sueño profundo (deepsleep) hasta que se active nuevamente transcurrido en el tiempo fijado por configuración.

## Node-RED

# Capítulo 5. Resultados y simulaciones

# Capítulo 6. Conclusiones y líneas futuras

# Anexos