

Trabajo Fin de Grado

Ingeniería Electrónica, Robótica y Mecatrónica

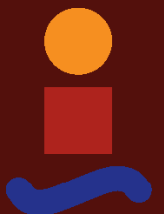
Implementación de sistema IoT para telemetría y riego automático de sustrato para el cultivo de plantas

Autor: Bruno Fernández Sánchez-Hermosilla

Tutor: Juan Antonio Becerra González

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2024



Trabajo Fin de Grado
Ingeniería Electrónica, Robótica y Mecatrónica

Implementación de sistema IoT para telemetría y riego automático de sustrato para el cultivo de plantas

Autor:

Bruno Fernández Sánchez-Hermosilla

Tutor:

Juan Antonio Becerra González

Profesor Titular

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2024

Trabajo Fin de Grado: Implementación de sistema IoT para telemetría y riego automático de sustrato para el cultivo de plantas

Autor: Bruno Fernández Sánchez-Hermosilla
Tutor: Juan Antonio Becerra González

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Aprovecho este espacio para tomar la oportunidad de compartir la autoría de este trabajo con aquellas personas que influyeron y me ayudaron tanto directamente como indirectamente, pues sin ellas jamás habría conseguido llegar a este punto.

Primero, debo agradecer los ánimos y consejos de mis padres y, en general, de mi familia, a lo largo de la carrera y durante este trabajo, ya que ellos siempre creyeron en mí aunque me encontrara baches en el camino.

También agradecer a Carmen, que me acompaña incluso antes de iniciar esta aventura y me ha reanimado, tranquilizado, cuidado y, sobre todo, querido durante mis altos y mis bajos, siempre con una sonrisa.

Durante la carrera he hecho amigos irremplazables y agradezco sobre todo a Mate y Jaime por acompañarme y hacerme más amenas todas esas clases, prácticas y semanas de preparación de exámenes que parecían interminables.

Por último, doy las gracias por todos aquellos profesores que, a lo largo de los años, me han enseñado los conceptos necesarios para que este trabajo no fuera una tarea imposible, en especial, me gustaría agradecer las atenciones de Juan Antonio Becerra, que ha sido mi tutor en este proyecto y creyó que podría hacer un buen trabajo.

A todos, ¡Gracias!

Resumen

En este trabajo se van a mostrar las diferentes formas de utilizar sistemas de sensores en conjunto con tecnologías IoT (*Internet of Things*), para la monitorización y subida a la nube de las diferentes variables del sustrato terrestre, muy relevantes en la agricultura moderna.

Además, sobre estos sistemas de sensores, se comentarán y analizarán los resultados de implementar, en base a sus medidas, un sistema de riego, que mantenga el sustrato terrestre de las plantas en un rango de humedad óptimo automáticamente.

Por último, se discutirá la conveniencia de usar uno u otro sistema de sensores en base a las capacidades materiales y objetivos de aquel que pretenda reproducir estos experimentos.

Abstract

In this assignment, different ways of using sensor systems alongside IoT (*Internet of Things*) technologies will be shown, with the objective of monitoring and uploading to the cloud the many variables present in the terrestrial substrate, quite relevant in modern agriculture.

Moreover, over these sensor systems, the results of the implementation of an irrigation system, based on their measures, that can maintain the terrestrial substrate of the plants in an optimal moisture range, will be commented and analyzed.

Lastly, the convenience of using one or another sensor system will be discussed regarding the material capacities and objectives of those who pretend to reproduce these experiments.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
1 Introducción	1
1.1 Objetivos	2
1.2 Estructura del Documento	2
2 Estado Actual de las Tecnologías IoT	5
2.1 Disposición y Conexión de Sensores	5
2.1.1 Topología Tipo Estrella	5
2.1.2 Redes en Malla Inalámbrica	5
2.1.3 Salida Local o <i>Local Gateway</i>	6
2.1.4 Jerárquica o en Árbol	6
2.1.5 Conexión Directo	7
2.2 Protocolos de Comunicación IoT	7
2.2.1 Protocolos de Bajo Rango de Operación	7
Bluetooth de Baja Energía, <i>Bluetooth Low Energy</i> o BLE	7
ZigBee	8
<i>IPv6 over Low power Wireless Personal Area Networks</i> o 6LoWPAN	9
2.2.2 Protocolos de Alto Rango de Operación	9
<i>Long Range Wide Area Network</i> o LoRaWAN	9
SigFox	11
<i>NarrowBand IoT</i> o NB-IoT	12
2.3 Aplicaciones de las Tecnologías IoT	14
2.3.1 Gestión Inteligente de Energía o <i>Smart Grids</i>	14
2.3.2 Industria y Producción Inteligente	15
2.3.3 Venta Minorista Inteligente	16
2.3.4 Monitorización de la Salud en Tiempo Real	16
2.3.5 Inteligencia Artificial en IoT	17
2.3.6 Agricultura Inteligente	18
Medición de Variables Agrícolas	18
Métodos de Irrigación Inteligente	20
Detección de Enfermedades de los Cultivos	21
3 Diseño de Experimentos	23
3.1 Elementos Comunes a Ambos Experimentos	23
3.1.1 Microcontrolador ESP8266 en Placa de Desarrollo D1 Mini	23
3.1.2 Recolección y Visualización de Datos en la Nube	24
3.1.3 Sistema de Irrigación	25
3.2 Sistema de Medida con Sonda <i>Capacitive Soil Moisture Sensor v2.0</i>	25
3.2.1 Convertidor Analógico-Digital ADS1115	26

3.2.2	Calibración de la Sonda	26
3.2.3	Conexión del Sistema Completo	27
3.2.4	Programa Utilizado	27
3.3	Sistema de Medida con Sonda <i>MEC10</i>	29
3.3.1	Capa Física RS-485 y Protocolo Modbus	29
3.3.2	Módulo MAX485	31
3.3.3	Conexión del Sistema Completo	31
3.3.4	Programa Utilizado	32
4	Resultados Experimentales	35
4.1	Medios para la Representación de Datos	35
4.2	Resultados del Experimento Inicial	35
4.3	Resultados del Experimento Avanzado	37
4.4	Otros Aspectos a Considerar	42
4.5	Diseño de Placas Personalizadas	44
5	Conclusiones	49
5.1	Líneas Futuras	51
	<i>Índice de Figuras</i>	53
	<i>Índice de Tablas</i>	55
	<i>Índice de Códigos</i>	57
	<i>Bibliografía</i>	59
	<i>Glosario</i>	67

1 Introducción

El objetivo principal de la agricultura moderna es el de generar la mayor cantidad de alimentos posible, de la mayor calidad posible y usando la menor cantidad de recursos posible. Una de las formas de conseguir esta meta es mediante la optimización de las tierras de cultivo disponibles, por ello, es cada vez más relevante el estudio de tecnologías que permitan conocer datos de estos terrenos y, por tanto, saber qué está ocurriendo en ellos. Además, es importante mencionar que el 70% del tiempo dedicado a la agricultura se emplea en la monitorización y estudio de las variables relevantes del terreno [1], por tanto, la recolección de estos datos automáticamente daría la oportunidad de aumentar en gran medida la eficiencia de las plantas agrícolas. Este conocimiento es el que permite estudiar experimentalmente los complejos sistemas naturales agrícolas y, tras este estudio, se puede actuar sobre estos sistemas de la manera óptima.

Muy posiblemente, la optimización de las tierras de cultivo disponibles sea uno de los desafíos más complicados y urgentes a los que se debe enfrentar la sociedad, esto se evidencia en mayor medida atendiendo a los datos de crecimiento de la población mundial, pudiendo llegar a la preocupante cifra de diez mil millones de personas en el año 2050 [2]. Teniendo en cuenta que las tierras de cultivos son limitadas, es necesario aprovechar cada terreno lo mejor que se pueda, con el objetivo de poder proveer al mundo del futuro.

También se debe dar importancia a las preocupaciones medioambientales tras la agricultura y actuar en consecuencia, por tanto, no sólo se debe tener en cuenta la optimización de la producción, sino que es necesario investigar en rumbo a la optimización de los recursos utilizados. Estos engloban tanto a la energía utilizada para las actuaciones necesarias en las plantaciones como a la cantidad de agua usada, otro bien limitado y del cual sería importante utilizar lo mínimo posible.

Otra restricción asociada es el precio de los sistemas que permitan esta optimización, ya que, a un menor precio, existen más probabilidades de que los responsables de instalaciones agrícolas puedan permitirse estas tecnologías. Por tanto, es de suma importancia que el medio por el que se realizan estas optimizaciones sea de bajo coste, con el objetivo de que el mayor número de instalaciones posibles puedan implementar estos sistemas.

Por último, se puede destacar la necesidad de que el conjunto de las tecnologías utilizadas no sólo no sean totalmente propietarias sino que también sea sencillo familiarizarse con ellas desde un nivel básico. Esto podría hacer posible un avance mayor de estas tecnologías basado en la cooperación, ofreciendo librerías, ejemplos de códigos o tutoriales creados enteramente por la comunidad que sirvan de base a conceptos nuevos.

Todos estos requisitos van de la mano del IoT, ya que estas tecnologías permiten la implementación de sistemas de monitoreo de variables del terreno que realizan mediciones de manera suficientemente precisa y con un consumo de energía lo suficientemente bajo como para poder desplegarse en todo tipo de situaciones y contextos.

1.1 Objetivos

Este Trabajo de Fin de Grado se ha centrado en la creación e implementación de sistemas de riego cuya actuación se rija por las necesidades de la planta en cada momento y que, además, mande los datos recogidos a la nube para hacer posible un análisis más profundo y especializado que los actualmente posibles mediante técnicas tradicionales.

Para llevar a cabo este objetivo, se han creado dos sistemas de sensores, cada uno con diferentes prestaciones, los cuales han realizado mediciones en el sustrato de una planta durante diferentes tiempos de ensayo. En ambos, la variable utilizada para determinar la actuación sobre el sustrato ha sido la humedad, ya que, en base a ella, se controla la cantidad de agua que fluye hacia el sustrato. Todos los datos recogidos durante estos tiempos de ensayo se han recopilado en la nube, permitiendo así ver todos estos datos organizados en el tiempo y pudiendo analizar el sistema conjunto formado por la planta y su sustrato.

Sin embargo, antes de hablar sobre estos dispositivos, se han investigado las diferentes tecnologías relevantes que engloban los sistemas IoT. Esto tiene la intención de mostrar las diferentes formas que estas toman y por qué es interesante investigar y realizar avances en este campo, además de servir como introducción al lector que no esté familiarizado con estas tecnologías.

En general, los resultados generados por estos experimentos han sido favorables, se ha conseguido mantener la humedad dentro de umbrales determinados por el usuario con ambos sistemas de sensores y se ha conseguido visualizar y recolectar correctamente todos los datos del sustrato de la planta que se habían propuesto. Se ha realizado un análisis del precio aproximado de despliegue de estos dispositivos para discutir su viabilidad y también se han diseñado componentes personalizados, con el objetivo de bajar el precio de implementación de estos.

1.2 Estructura del Documento

Este documento se organiza en cinco capítulos distintos, cada uno de ellos acomete diferentes aspectos a resaltar a la hora de exponer tanto la importancia y flexibilidad de las tecnologías IoT cómo la aplicación y comprobación de la eficacia de estas mediante experimentos en un sistema real.

A continuación, se presentan los capítulos antes mencionados, indicando su contenido y, por tanto, su función en el escrito:

El capítulo 2, titulado *Estado Actual de las Tecnologías IoT*, pretende realizar una breve revisión de la literatura y de las distintas tecnologías que engloban a los sistemas IoT. Esta revisión no sólo introduce los conceptos y restricciones básicas que conlleva el uso de las tecnologías IoT en general, sino que también se realiza una recopilación de topologías de los sensores dispuestos para aplicaciones IoT, de protocolos de comunicación relevantes y, además, de algunas de las muchas aplicaciones en las que el IoT consigue (o puede llegar a conseguir) optimizaciones imposibles de lograr con técnicas previas.

El capítulo 3, titulado *Diseño de Experimentos*, expone las diferentes preparaciones y procesos llevados a cabo para la creación de dos sistemas de medida distintos. En particular, en este capítulo se pueden encontrar:

- Listados de los diferentes componentes utilizados en ambos experimentos, indicando los elementos comunes y específicos de cada uno. Además, se exhiben brevemente sus características, el funcionamiento de estos y las diferentes justificaciones que llevaron a elegirlos en primer lugar. Por último, se han explicado los diferentes protocolos de comunicación necesarios para la lectura de los sensores utilizados.
- Esquemáticos que indican en un vistazo las conexiones realizadas en todos los componentes, además de justificaciones del por qué de la realización de estas.
- Diagramas de flujo que muestran la lógica y estructura de los programas creados, estos programas son los que controlan el comportamiento de los elementos del sistema.

También, a lo largo de este capítulo, se ha hecho referencia a todas las librerías y entornos utilizados para la programación del control de estos sistemas y para la subida de datos a la nube. Al existir tantos recursos a la hora de realizar este tipo de experimentos, se evidencia la facilidad de introducción a este tipo de tecnologías incluso a usuarios con poca experiencia.

El capítulo 4, titulado *Resultados Experimentales*, muestra, en primer lugar, los recursos utilizados para la exportación de datos de la nube y, posteriormente, los códigos creados para la representación de estos datos en diferentes gráficas, posteriormente, se muestran estas gráficas, explicando diferentes aspectos relevantes de los experimentos realizados en base a estas. En segundo lugar, se ha indicado el coste aproximado de cada uno de los componentes utilizados en ambos sistemas de medida, lo cual da lugar a una breve discusión de viabilidad a la hora de la implementación de estos en los emplazamientos agrícola. Por último, se han diseñado dos placas personalizadas, las cuales incorporan para cada sistema de medida el sensor, el microcontrolador y el modo de actuación en un solo módulo, con el objetivo de facilitar la realización de este tipo de experimentos y de, además, poder implementar estos sistemas con costes reducidos, los cuales también se han indicado y discutido.

Finalmente, el capítulo 5, titulado *Conclusiones*, muestra las conclusiones a las que se han llegado tras la realización de este trabajo, además, también se indican algunas de las líneas futuras que otros pueden tomar como punto de partida para la realización de nuevos experimentos basados en los sistemas de sensores diseñados.

2 Estado Actual de las Tecnologías IoT

El primer paso de una investigación es estudiar en qué se ha indagado antes sobre ese tema y, por tanto, descubrir si uno puede aportar nuevos conocimientos o añadir matices adicionales a algún desarrollo anterior.

Por ello, en este capítulo se van presentar los distintos acercamientos actuales tanto a las tecnologías IoT como al uso combinado de estas con sistemas de sensores, además de las distintas aplicaciones en las que estos dispositivos ofrecen mejoras que, en algunos casos, eran imposibles de implementar con técnicas anteriores.

2.1 Disposición y Conexión de Sensores

Los sistemas IoT normalmente están implementados en base a una gran cantidad de dispositivos, y cómo la única funcionalidad de estos es conectarse a la red y enviar datos de telemetría, precisan de una cantidad de energía ínfima para funcionar y, por tanto, suelen estar alimentados por baterías.

Sin embargo, aunque es beneficioso que estos dispositivos gasten tan poca energía, también esto tiene la restricción asociada de que la conexión de estos dispositivos a la red debe consumir la menor cantidad de energía posible. Esto hace que las tecnologías de comunicación que normalmente se usarían para la conexión de dispositivos sean inutilizables en este caso, por el consumo de energía que precisan para llevarse a cabo. También es necesario que los dispositivos puedan comunicarse entre sí a largas distancias, para, así, poder tomar medidas de la mayor extensión de terreno posible.

Estos requisitos hacen que tecnologías de transmisión de datos como el Wi-Fi o el Bluetooth convencional no sean las primeras opciones a considerar (aunque existan ejemplos de utilización específicos), utilizando alternativas de menor consumo de energía y mayor rango, pero con la desventaja de una significativa reducción de la velocidad de transmisión. A continuación, se exponen algunas de las topologías de conexión de sensores más relevantes usadas en las aplicaciones IoT:

2.1.1 Topología Tipo Estrella

Esta arquitectura se compone de un nodo o servidor central que recibe información de otros nodos emisores, estos nodos emisores sólo están conectados al nodo central. La comunicación con el nodo central es directa, por lo tanto, no se precisa de un algoritmo de enrutado, lo cual hace que el gasto de energía se concentre sólo en la emisión de los paquetes. Sin embargo, existe un punto de fallo fatal (*Single Point Of Failure* o *SPOF*), el servidor (o servidores) que se conecta a la red [3], es decir, si este falla, toda la red queda inservible y desconectada de la nube. En la Figura 2.1 se puede observar un esquema que muestra esta topología.

2.1.2 Redes en Malla Inalámbrica

La arquitectura de Redes en Malla Inalámbrica o WMN (*Wireless Mesh Network*) se compone de muchos dispositivos IoT conectados entre sí, con uno o varios puntos de conexión a Internet (Nodos de Puerta de Enlace, Nodos de Salida o *Gateway Nodes*), los cuales mandarían los datos de los dispositivos de la malla a la nube. En la Figura 2.2 se puede apreciar un esquema de este tipo de topología.

En esta arquitectura, si alguno de los puntos de la malla deja de funcionar, esta puede reorganizarse para mandar los datos a los otros puntos que funcionen correctamente, asegurando así la robustez del sistema (incluso si los *Gateway Nodes* fallaran, todavía podrían existir comunicaciones locales en la malla).

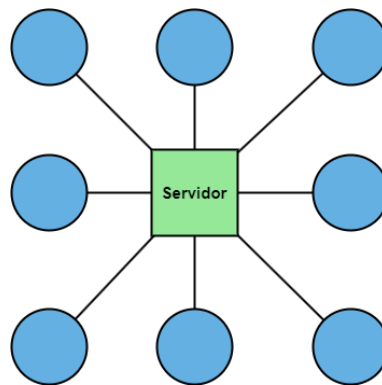


Figura 2.1 Topología tipo Estrella.

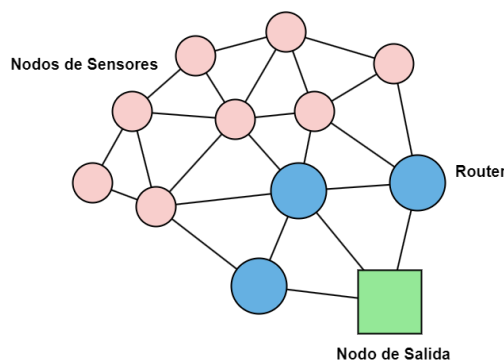


Figura 2.2 Topología tipo Red en Malla.

La mayor desventaja presente en esta arquitectura es el bajo throughput de datos, esto se debe a que el protocolo de red debe encontrar en cada momento el camino óptimo desde su posición en la red hasta el *Gateway Node* correspondiente. También, debido a la gran cantidad de dispositivos, es necesario un cambio constante del canal de radiofrecuencias con el objetivo de no interrumpirse las comunicaciones entre ellos. Sin embargo, el tiempo consumido por la obtención del camino óptimo disminuye una vez el sistema se estabiliza (una vez se encuentra el camino óptimo hasta el *Gateway Node* para todos los dispositivos), aunque esto puede cambiar si ocurren fallos en la red [4].

2.1.3 Salida Local o *Local Gateway*

Esta arquitectura está basada en la creación de grupos de sensores que, con topología tipo estrella, se conectan a *Gateway Nodes*, los cuales mandan los datos a Internet. Esto puede ser de gran utilidad para redes que se extienden espacialmente a lo largo de muchos kilómetros, en aplicaciones de estas características la conexión en malla es complicada de implementar, por tanto, se suele recurrir a esparcir estos grupos independientes a lo largo del terreno.

La gran desventaja de esta arquitectura es que, como se indicó anteriormente, la utilización de la topología tipo estrella tiene como resultado la adición de muchos *SPOFs*, que, tras fallar, pueden dejar de proporcionar datos de un grupo amplio de sensores, además de no dar la opción a estos dispositivos de comunicarse entre sí [4].

2.1.4 Jerárquica o en Árbol

Esta arquitectura es el resultado de aplicar a varios niveles (o jerarquías) la topología tipo estrella. Se empieza el "árbol" en el nodo conectado a internet, el cuál se puede considerar "profundidad 0" o "nivel 0" de la red de dispositivos, a este nodo se conectan unos nodos hijos a una "profundidad" o "nivel" mayor, estos a su vez estarán conectados a nodos de una mayor profundidad y, así hasta que se abarque a todos los dispositivos que se quieran conectar [5].

La ventaja principal de esta arquitectura es que aunque, igual que en la de salida local, el fallo de un nodo intermedio puede dejar a una región de sensores inutilizada, el gasto energético en algoritmos de enrutado es

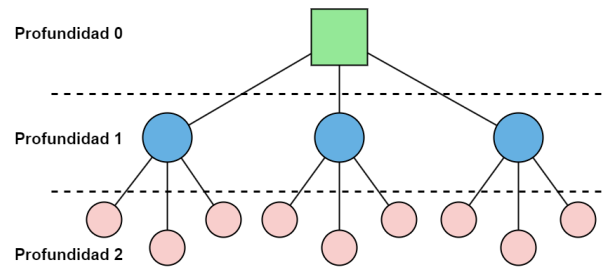


Figura 2.3 Topología Jerárquica o en Árbol.

ínfimo, ya que los paquetes de datos se mandan al nodo "padre" (de profundidad o nivel inferior) en cada uno de los casos hasta llegar al nodo conectado a internet.

2.1.5 Conexionado Directo

Esta arquitectura se basa simplemente en la conexión de los dispositivos IoT a antenas de telefonía móvil, por tanto, no es necesario crear una red, sino conectar los dispositivos a la red existente. Al conectarlos a estas redes, se consiguen anchos de banda mucho mayores que en las otras dos arquitecturas.

Sin embargo, estas conexiones tienen el coste añadido de que, además de que es preciso instalar un módulo que permita la conexión (por ejemplo, una tarjeta SIM), también se debe contratar, en algunos casos, servicios de datos móviles para cada dispositivo conectado. Esto puede hacer que el coste total crezca de manera desmesurada cuando se quieren utilizar muchos dispositivos de medida y recolección de datos [4].

El uso de una arquitectura u otra es algo que está condicionado por los objetivos y posibilidades del proyecto a realizar, por lo tanto, no es inusual encontrar implementaciones IoT que usan incluso varias arquitecturas distintas a la vez, con el objetivo de optimizar la obtención de datos.

2.2 Protocolos de Comunicación IoT

Como bien se ha comentado al principio del capítulo, es necesario tener restricciones como la energía consumida o el rango de las comunicaciones en cuenta, por ello, se debe elegir el protocolo de comunicación entre dispositivos IoT adecuado para cumplir los objetivos de cada aplicación. Se presentarán a continuación las opciones disponibles más destacadas.

2.2.1 Protocolos de Bajo Rango de Operación

Bluetooth de Baja Energía, *Bluetooth Low Energy* o BLE

Esta versión del Bluetooth convencional se centra más en el bajo consumo de energía por parte del dispositivo que lo utiliza a cambio de un sacrificio de velocidad de transmisión y rango. No es tanto una optimización del Bluetooth clásico, sino que es un paso en otra dirección y que, por tanto, se espera que se utilice cada tipo de Bluetooth para distintos objetivos [6].

Con el objetivo de disminuir el consumo de energía, se deben optimizar las acciones que los dispositivos van a realizar numerosas veces, es decir, descubrimiento de dispositivos, conexión a estos y envío de datos. Para ello, se hace uso de varias técnicas [6]:

- Cuando un dispositivo quiere ser descubierto, este transmite tres mensajes en tres frecuencias distintas, luego, debe esperar la recepción de un mensaje de algún dispositivo que esté intentando conectarse a él, si lo hubiera. El número de mensajes se decide en base al compromiso entre robustez del protocolo y la eficiencia de este, si sólo se mandara un mensaje, pero la frecuencia a la que se manda fuera inaccesible en ese momento, el protocolo fallaría a menudo, pero si se mandaran muchos más mensajes, el consumo de energía sería mayor aunque ocurrieran menos fallos.
- En la comunicación entre dos dispositivos que utilizan BLE siempre está definido cuál de ellos es el receptor y cuál el emisor, el descubierto emite y el que lo descubrió recibe. El receptor es el que más energía gasta, ya que debe buscar al dispositivo emisor y hacer espera activa de las transmisiones, por

tanto, se suele dejar el rol de recepción a dispositivos con un relativamente superior suministro de energía.

- Como en estos dispositivos se suele usar una batería de botón o similares como fuente de energía, es mucho más eficiente transmitir y recibir paquetes cortos de información de forma espaciada en el tiempo, ya que se estarían consumiendo pequeñas ráfagas de energía en vez de consumirla continuamente. Esto hace que la vida útil de la batería aumente porque, al dejar espacio entre ráfagas, se impide el sobrecalentamiento de esta.



Figura 2.4 Batería de Botón o *Button-Cell Battery*.

Aunque la eficiencia energética del BLE sea algo a tener en cuenta, este protocolo brilla sobre todo en instalaciones en las que comunicaciones en rangos de 15m a 30m [7] son suficientes, por tanto, puede ser interesante para usos de IoT en aplicaciones cotidianas e, incluso, en instalaciones agrícolas de tamaño reducido. Esta restricción de rango no es más que una consecuencia de la restricción energética mencionada, a mayor rango deseado, mayor es la potencia a la que se debe transmitir la señal que contiene los datos, ya que hay que tener en cuenta las pérdidas de potencia por la propagación de la señal por el aire o *Path Loss*.

ZigBee

Al igual que el BLE, ZigBee es un protocolo de comunicación que tiene como objetivo la transmisión de datos con bajo coste energético y monetario. El ahorro de energía se consigue gracias al bajo tiempo de actividad o *Duty Cycle* de estos dispositivos, en algunas aplicaciones, siendo de menos del 1%, lo cual se traduce en un gasto energético que permite un uso ininterrumpido de incluso años [5].

Aunque ZigBee se pueda disponer en topología estrella, árbol o redes en malla, para esta última es necesario crear un algoritmo de enrutado eficiente, para ello, se realiza una función de costes en base a las siguientes métricas [5]:

- El LQI o *Link Quality Indicator* indica la calidad de los paquetes que recibe un receptor al que queremos emitir un paquete. Esta propiedad se puede deducir a partir del SNR o *Signal-to-noise ratio*, el cual mide la relación entre la potencia de la señal de transmisión del paquete y la potencia de la señal de ruido dentro de la banda frecuencial de transmisión, a mayor SNR, mayor será la calidad de la señal y, por tanto, de la recepción del paquete.

$$SNR = \frac{P_{señal}}{P_{ruido}} \quad (2.1)$$

Al utilizar esta propiedad como métrica para el algoritmo de enrutado estamos asegurando paquetes de mejor calidad con un gasto de energía menor (menor potencia de transmisión de señal requerida debido a un menor ruido).

- El descriptor de energía residual es el que indica la batería restante de cada nodo además del estado de la propia batería. Tener en cuenta esta propiedad para el algoritmo de enrutado permite que la mayoría de dispositivos elijan enviar sus paquetes a otros con más batería restante. Esto hace que los nodos que se encuentran bajos en batería tarden más en descargarse por completo, además de darles tiempo para poder recargarse, si estuviera implementado un sistema de recarga de baterías. También se contempla el uso mixto de baterías recargables y no recargables, en ese caso, los dispositivos que gozaran de baterías recargables tendrían un menor coste de transmisión.
- La distancia física hasta el dispositivo de destino también se puede tener en cuenta a la hora de generar la función de costes. Se puede implementar de forma que la aportación a la función de costes es inversamente proporcional al progreso realizado en la dirección correcta al destino, es decir, cuanto más progreso hacia la dirección deseada, menor coste añadido.

IPv6 over Low power Wireless Personal Area Networks o 6LoWPAN

Aunque este protocolo comparte los mismos objetivos que en los anteriormente mencionados, 6LoWPAN da una solución a las redes de dispositivos IoT fuera del uso de tecnologías propietarias y privadas (como lo puede ser ZigBee).

Para cumplir ese objetivo, 6LoWPAN integra el protocolo IPv6, esto hace que el funcionamiento interno sea conocimiento público, además de permitir la comunicación de dispositivos 6LoWPAN con otras redes IP sin ningún intermediario [8].

Sin embargo, el protocolo IPv6 en un principio no está pensado para utilizarse en dispositivos de tan bajo consumo y de tan baja velocidad de transmisión. Por tanto, se deben hacer adaptaciones para que su funcionamiento sea más apto para IoT [8]:

- Debido al ínfimo *duty cycle* que suelen tener los dispositivos de bajo consumo, es muy probable que, cuando un servidor quiera acceder a un dispositivo 6LoWPAN, este esté en modo de bajo consumo y, por tanto, no disponible para enviar datos. Por ello, es recomendable que la comunicación sea iniciada por el nodo 6LoWPAN de forma asíncrona, lo cual es totalmente contrario al sondeo o *polling* por parte del servidor, que es lo que se suele utilizar en el protocolo IP.
- El protocolo de transporte utilizado es UDP, para el envío de paquetes de datos, debido a su sencillez y a su reducida cabecera en comparación con el protocolo TCP. Los datos deben enviarse de manera compacta (preferiblemente en binario) y, además, existen varias técnicas internas de compresión de cabeceras que hacen que los paquetes sean mucho menos pesados.

2.2.2 Protocolos de Alto Rango de Operación

Long Range Wide Area Network o LoRaWAN

Antes de explicar el protocolo de comunicación LoRaWAN, es conveniente explicar la capa física que lo hace posible, llamada *Long Range* o LoRa.

Muchos sistemas inalámbricos de bajo consumo suelen usar Modulación por Desplazamiento en Frecuencia *Frequency Shift Keying* o FSK, la cual consigue la transmisión de datos mediante el uso de dos o más señales portadoras con frecuencias distintas (dentro de una banda frecuencial acotada), su implementación más simple consiste en la transmisión de dos señales, una de ellas representa un "0" lógico y la otra un "1" lógico. Con esto, cuando el receptor intente descifrar la transmisión, sólo le es necesario conocer qué frecuencia corresponde a qué nivel lógico.

En LoRa se utiliza la tecnología de transmisión CSS o *Chirp Spread Spectrum*, esta se basa en la transmisión de datos a partir de señales que aumentan o disminuyen en frecuencia linealmente (o incluso dan saltos frecuenciales) llamadas "símbolos". Una propiedad de estos es el factor de dispersión, *Spreading Factor* o SF, el cual indica qué cantidad de símbolos diferentes se pueden transmitir, en LoRa se usa desde el SF7 hasta el SF12. Como la cantidad de símbolos viene dada por la siguiente ecuación [9]:

$$N_{sim} = 2^{SF} - 1 \quad (2.2)$$

Se podrán transmitir de 127 a 4095 símbolos distintos dependiendo del SF elegido.

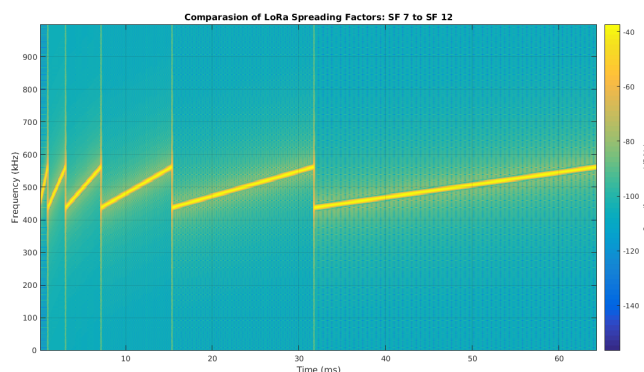


Figura 2.5 Espectrograma de los diferentes Factores de Dispersión [10].

Para que el proceso funcione de forma asíncrona y el receptor descifre de manera correcta los datos, se añaden unos símbolos de preámbulo (varios símbolos iguales) y otros de sincronización antes de los datos en sí, cómo se puede observar en la Figura 2.6 [10].

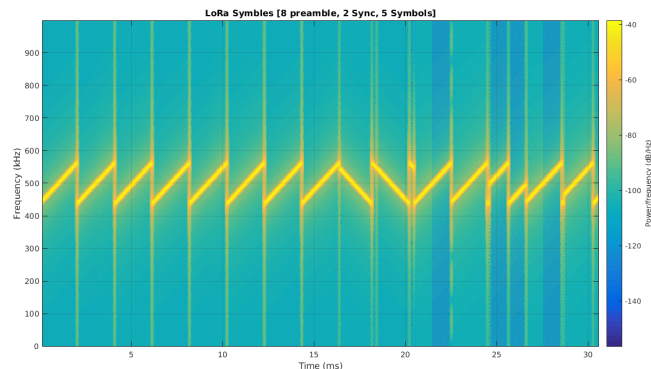


Figura 2.6 Ejemplo de transmisión de datos con símbolos CSS [10].

El receptor realiza la demodulación de estos símbolos en dos pasos [11], ilustrados en la Figura 2.7 y explicados a continuación:

1. Se multiplica la señal recibida por el símbolo CSS "0" complejo conjugado, también llamado *Pure Downchirp*. Esta operación se llama *Dechirping*.
2. Se calcula la DFT del resultado de la multiplicación y, posteriormente, se elige la frecuencia de mayor magnitud de la DFT realizada, que es la que corresponde al símbolo correcto.

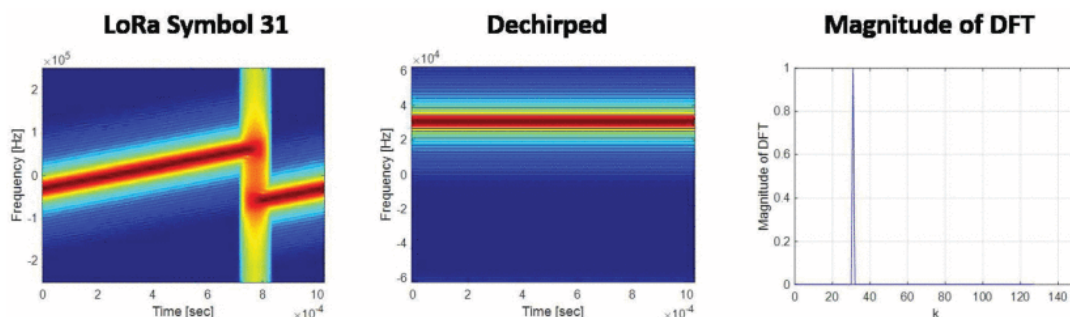


Figura 2.7 Pasos para la demodulación de un símbolo CSS [11].

Gracias a esta forma de demodulación, la transmisión CSS es muy resistente al ruido que pueda existir en el medio. La robustez de esta se puede medir mediante el Ratio de Bits Erróneos, *Bit Error Rate* o BER, que se define como el ratio de bits erróneos recibidos entre el número total de bits enviados. Esta métrica se puede comparar con el SNR de la señal para deducir la susceptibilidad de la transmisión ante al ruido [12].

Se puede observar en la Figura 2.8 que la transmisión de datos es muy resistente al ruido, incluso en niveles de SNR muy bajos, por ello, se posibilita la demodulación de estas transmisiones por receptores a mayor distancia del emisor y, a la vez, el envío de estas a una potencia de transmisión admisible para las aplicaciones de bajo consumo.

La decisión de usar uno u otro SF viene dada por las restricciones habituales de consumo de batería y de rango de alcance, ya que, al aumentar el SF, aumenta el tiempo de transmisión del paquete o *Airtime*, lo que conlleva un uso mayor de la batería [13, 14], pero también aumenta la robustez ante el ruido [15]. Además, un aumento del SF se traduce en una disminución del alcance de los dispositivos [14] y de la velocidad de transmisión de los paquetes [16].

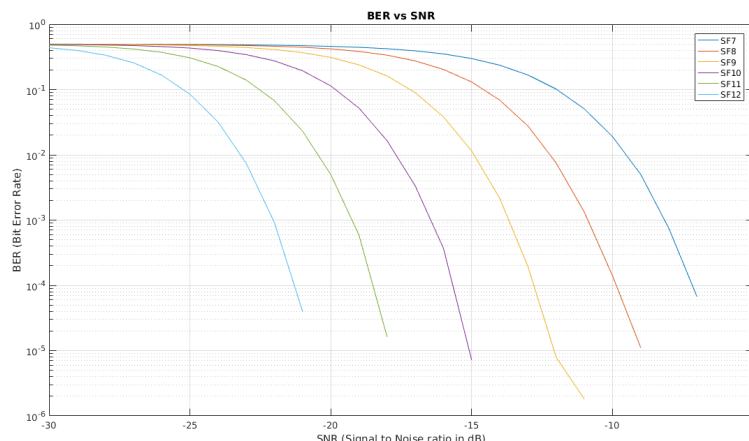


Figura 2.8 BER frente a SNR para todos los SF disponibles en LoRa [10].

Con un mayor entendimiento de las propiedades de LoRa, se puede empezar a explicar el protocolo de comunicación al que ofrece sus servicios: LoRaWAN.

Los dispositivos que usan LoRaWAN se disponen con arquitectura tipo estrella, es decir, no hay comunicación entre los sistemas de sensores, sólo entre estos y los Nodos de Salida. Un Nodo de Salida recibe paquetes de datos de todos los sensores en su rango de recepción, por tanto, es posible que varios Nodos de Salida reciban un mismo paquete, esto reduce las posibilidades de pérdida de estos [15].

Los sistemas de sensores mandan sus datos de forma asíncrona (o bien tras tomar una medida o bien tras un evento preprogramado), esto hace que no se gaste energía en la sincronización con el Nodo de Salida. Para que esto sea posible, los Nodos de Salida usan el protocolo ALOHA Puro [17], el cual deja "abierto" el canal de recepción, donde llegan los paquetes de los sistemas de sensores de forma aleatoria y sin organización alguna (por tanto, sin gastos de sincronización). Por todo esto, los paquetes recibidos tienen una alta posibilidad de interferir los unos con los otros [17, 18], realizando así la importancia de la redundancia en la recepción de paquetes, mencionada anteriormente.

Otra forma de reducir las interferencias es enviar señales ortogonales entre sí. Dos señales son ortogonales si no existe correlación entre ellas, esto permite que se puedan recibir varias señales a la vez sin interferencias provocadas por el envío simultáneo.

En LoRa, las señales ortogonales son aquellas que se envían con SF distinto, por tanto, para asegurar la mayor eficiencia en la recepción simultánea de paquetes, es necesario que el mayor número posible de sistemas de sensores envíen paquetes con SF distinto (siempre y cuando el SF le permita cumplir las especificaciones necesarias). En el caso en el que no sean ortogonales (mismo SF) es posible la colisión de los paquetes si estos son recibidos a la vez, sin embargo, si uno de ellos tiene una potencia de señal superior en 6dB al otro, sobrevivirá la colisión [15].

SigFox

SigFox es otra opción para la transmisión de datos a grandes distancias, realiza estas transmisiones mediante la Modulación por Desplazamiento de Fase Binaria, *Binary Phase Shift Keying* o BPSK [19, 20]. Esta consiste en la variación de la fase de una señal portadora entre dos valores (en el caso no binario, entre varios valores), estos cambios de fase indican, de forma parecida a la modulación FSK, el "0" lógico o el "1" lógico. Por tanto, el receptor sólo debe reconocer estos cambios en la fase de la portadora para demodular la transmisión.

Para cumplir las restricciones de consumo de batería y los objetivos de rango, los datos se transmiten en una Banda Ultra Estrecha, *Ultra Narrow Band* o UNB [19], de no más de unas decenas o cientos de hercios de ancho. Cuando el receptor demodule las transmisiones recibidas, podrá descartar todas las componentes de ruido fuera de esa pequeña banda, con certeza de que no contendrán componentes frecuenciales relevantes [21, 22], la simpleza a la hora de demodular hace que los dispositivos de recepción Sigfox sean mucho más económicos [19]. Esta resistencia ante el ruido y las interferencias [22] permite que el dispositivo emisor pueda transmitir los datos con una potencia menor, lo cual permite alcanzar altas distancias de transmisión con un consumo de energía por parte del emisor relativamente bajo [19, 21].

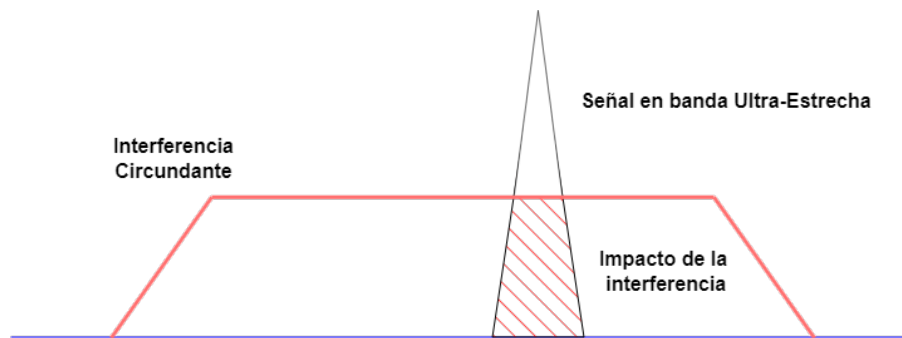


Figura 2.9 Representación de la resistencia al ruido y a las interferencias de las señales que hacen uso de UNB [22].

Sin embargo, el uso de UNB impone unas duras limitaciones de velocidad de transmisión, que normalmente está por debajo de los 100bps [19, 20]. Además, como Sigfox transmite en frecuencias sin necesidad de licencia, debe obedecer las restricciones en cuanto a duty cycle de los dispositivos [19, 23]. En particular, en la UE es del 1%, lo que hace que estos dispositivos sólo puedan transmitir datos durante 36 segundos cada hora, con un *airtime* por mensaje de 6 segundos [23], son 6 mensajes cada hora, lo cual puede llegar a ser un limitante para algunas aplicaciones que precisen de altas tasas de transmisión de datos [19].

Al igual que LoRa y, por las mismas razones, SigFox utiliza la arquitectura tipo estrella [19], además, también comparte la transmisión asíncrona de paquetes a unos receptores que reciben estos con el protocolo ALOHA [23]. En lo que se diferencia de LoRa es que, en vez de garantizar redundancia en el envío de paquetes mediante la transmisión de estos a varios Nodos de Salida, la garantiza gracias al envío de tres paquetes en secuencia, cada uno con una señal portadora distinta [19]. Esto da la posibilidad de que, aunque se pierdan paquetes debido a interferencias o colisiones, haya más posibilidades de recepción exitosa [23].

NarrowBand IoT o NB-IoT

NB-IoT es un protocolo de comunicación basado en el protocolo LTE [19, 24], en particular, se simplifica el protocolo LTE de tal manera que cumpla con las restricciones habituales de los dispositivos IoT (uso de Banda Estrecha o *Narrowband*, disminución del máximo de bits transmitidos cada vez... [25]), ya que este se propuso originalmente como protocolo de comunicación de altas prestaciones para telefonía móvil [26]. Una de estas modificaciones es el Modo de Ahorro de Energía, *Power Saving Mode* o PSM, que optimiza el uso de la energía del dispositivo, en este modo, el dispositivo apaga su módulo de radio y entra en sueño profundo, del que sólo podrá ser despertado cuando este transmita un paquete de datos, en ese caso, se deberá conectar al Nodo de Salida (al igual que LoRaWAN o SigFox, NB-IoT se utiliza en dispositivos con topología tipo estrella [19]), enviar el paquete y esperar, si, tras un tiempo, no está programado el envío de otro mensaje, vuelve al PSM [27, 28].

La transmisión de datos desde los sistemas de sensores hacia la red se realiza mediante Acceso Múltiple por División de Frecuencia de Portadora Única, *Single Carrier Frequency Division Multiple Access* o SC-FDMA [24, 29], la cual consiste en la realización de una DFT de un paquete de datos modulado (en el dominio del tiempo, mediante Modulación por Desplazamiento en Fase o PSK), esta operación convierte los datos al dominio de la frecuencia, esparciéndolos en una serie de subportadoras, que, entre todas, ocupan el ancho de banda asignado al emisor. El receptor sólo debe deshacer la operación con una DFT Inversa para acceder a los datos en el dominio del tiempo [30, 31].

Una de las ventajas de SC-FDMA es que la relación entre el pico y la media de la potencia de transmisión, *Peak-to-Average Power Ratio* o PAPR es menor que en otras técnicas como la Multiplexación por División de Frecuencias Ortogonales, *Orthogonal Frequency Division Multiplexing* u OFDM [31, 32]. Si esta relación es alta, el amplificador de potencia del dispositivo debe estar preparado para potencias mucho mayores de la que habitualmente se va a consumir y, como este es uno de los principales pozos de energía del dispositivo, se termina usando la batería de forma ineficiente. Es por eso que el uso de SC-FDMA permite un mejor aprovechamiento de la batería y, en consecuencia, un mayor tiempo de vida útil de los dispositivos [30, 33].

A diferencia de LoRaWAN o SigFox, los dispositivos NB-IoT mandan sus datos de forma síncrona, además, deben mantenerse en contacto con el Nodo de Salida [34]. Para ello, el Nodo de Salida manda señales de sincronización al dispositivo NB-IoT usando OFDMA (en las transmisiones en las que el Nodo de Salida es el emisor, o transmisiones *Downlink* [24], no es tan necesario el ahorro de energía mediante el uso de

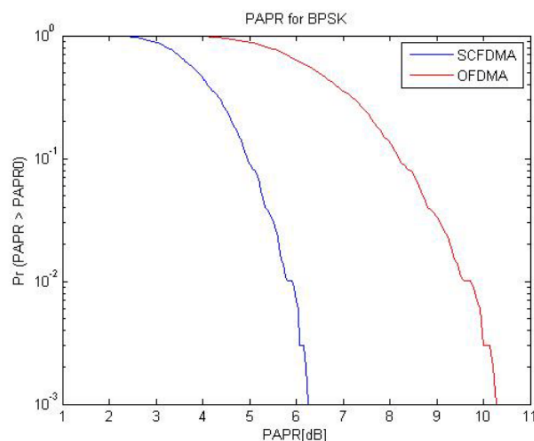


Figura 2.10 Probabilidad de que se dé cada valor de PAPR frente al valor de PAPR cuando se usa SC-FDMA y OFDM de Acceso Múltiple o OFDMA (con modulación BPSK en ambos casos) [32].

SC-FDMA, ya que, normalmente, son dispositivos conectados a la red eléctrica o con menores restricciones de consumo de batería):

- **Señal Primaria de Sincronización en Banda Estrecha o NPSS:** realiza la sincronización de tiempo y frecuencia con cada dispositivo de la red NB-IoT [35, 36], todos los dispositivos de la red usan la misma NPSS, por tanto, los dispositivos sólo deben buscar una NPSS para la sincronización (en LTE se utilizan tres señales de sincronización, por simpleza se han reducido en NB-IoT [35]). Esto es necesario para una recepción de paquetes con una menor tasa de error y con una gran disminución del retardo.
- **Señal Secundaria de Sincronización en Banda Estrecha o NSSS:** después de realizar la sincronización en tiempo y frecuencia con la NPSS, se envía esta señal para detectar la identidad de cada dispositivo emisor [35].

La sincronización en la transmisión permite disminuir las pérdidas y los errores de paquetes en comparación a otros protocolos asíncronos como el ALOHA. Se puede observar en la Figura 2.11 que, aunque se transmitan una gran cantidad de paquetes por minuto, el porcentaje de paquetes erróneos sube de manera mucho menos acelerada que con la misma tasa de transmisión en otras tecnologías.

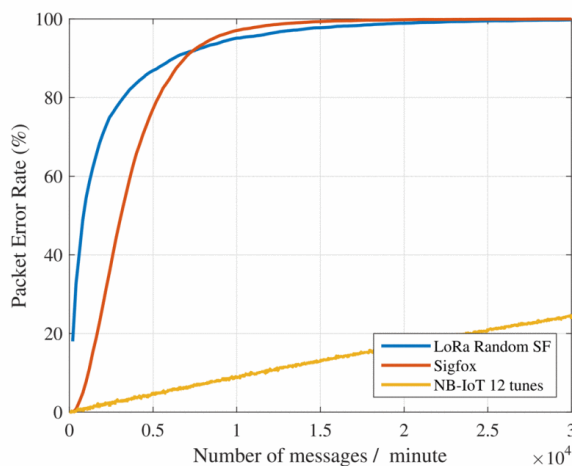


Figura 2.11 Comparación del Ratio de Error en Paquetes, *Packet Error Rate* o PER frente al número de paquetes por minuto transmitidos, comparación entre LoRaWAN, SigFox y NB-IoT [29]. .

Tabla 2.1 Comparación de Rango y Flujo de Datos.

Protocolo	Rango	Flujo de Datos Máximo
BLE	15-30m [7]	1Mbps [7]
ZigBee	10-100m [5]	250kbps [5]
6LoWPAN	10-100m [7]	250kbps [8]
LoRaWAN	3-18km [14]	50kbps [14, 19, 37]
Sigfox	10-50km [19, 20]	100bps [19, 20]
NB-IoT	1-10km [19]	250kbps [38]

2.3 Aplicaciones de las Tecnologías IoT

Tras comprender algo mejor el funcionamiento de algunas de las tecnologías más relevantes dentro del IoT, en esta sección se explican brevemente las posibles mejoras que introducen estas tecnologías en las distintas aplicaciones en las que se implementan.

2.3.1 Gestión Inteligente de Energía o *Smart Grids*

Con un monitoreo en tiempo real de la energía generada por las centrales eléctricas y la gastada por los consumidores se abre la posibilidad de analizar en mucha mayor profundidad datos de consumo y producción de energía. Esto permite realizar de forma mucho más informada y eficaz un conjunto de acciones [39] que mejoran la eficiencia energética de las redes:

- Optimización de la generación y consumición de energía, basada en datos anteriores y actuales.
- Identificación de fallos o irregularidades en la red y dónde se cometen estos, con objetivo de realizar una gestión de fallos mucho más inmediata (redireccionado de la energía para aislar la zona de fallo, por ejemplo).
- Gestión de la energía durante horas de alta generación, es decir, almacenamiento eficiente de la energía sobrante durante estos periodos.

Como la generación de energías renovables suele ser menor en comparación a otras formas de generación de energía y, en general, dependen de factores naturales, la optimización de la red eléctrica puede ser una forma de priorizar el uso de estas energías cuando estén disponibles, disminuyendo así la huella de carbono. Una forma de conseguir esto es mediante la comunicación entre varias plantas de energías renovables para poder dar suministrar de forma conjunta a la red [4], esto hace que sea más probable suplir de manera completa la demanda energética usando sólo energías renovables.

Según la escala en la que se aplique la gestión inteligente de energía, se pueden distinguir varios tipos de redes [40], estas redes se conectan y comunican entre sí para garantizar el funcionamiento óptimo de la *Smart Grid*:

- **Red de Área de Hogar, *Home Area Network* o HAN:** monitoriza la demanda energética de los distintos dispositivos de los consumidores (dispositivos inteligentes, electrodomésticos, vehículos eléctricos...) e informa con esos datos a los consumidores para que, por ejemplo, usen los electrodomésticos fuera de las horas de alto consumo. Esto puede desembocar incluso en una disminución del gasto doméstico en energía. Para este tipo de aplicaciones quizá tecnologías IoT como BLE o ZigBee [40] son las más indicadas, ya que cumplen los requisitos de rango y tasa de transmisión de datos para estas aplicaciones.
- **Red de Área Vecinal, *Neighborhood Area Network* o NAN:** en este caso, se realiza la monitorización de energía y el control de la distribución de esta entre las diferentes HANs de la zona. Además, se manda toda esta información a las redes de mayor escala (WAN). De ahora en adelante, el espacio físico que tienen que cubrir estas redes crea la necesidad de implementar protocolos de alto rango de operación como LoRaWAN o SigFox, además, es un claro ejemplo de la distribución en estrella, ya que los diferentes nodos HAN transmiten sus datos al Nodo de Salida NAN.
- **Red de Área Extensa, *Wide Area Network* o WAN:** vertebra las comunicaciones entre las diferentes redes de las *Smart Grids* y recibe los datos de las diferentes NANs, con el objetivo de usarlos para la distribución y generación inteligente de la energía en todas las escalas.

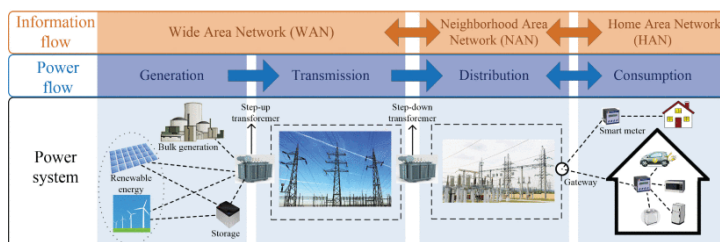


Figura 2.12 Representación de las diferentes escalas en las que funcionan las *Smart Grids* [40].

En un estudio realizado por la UE en países miembros, se estimó que el uso de estos sensores inteligentes puede llegar a ahorrar hasta un 10% de energía. Además, el ahorro monetario que suponen (por sensor instalado) supera el coste de instalación, por tanto, es una opción bastante asequible para el consumidor medio [41].

Otra cuestión sobre la incorporación de estos medidores es la protección de datos, de nada sirve tener la infraestructura necesaria para la monitorización de la energía si los consumidores no aceptan ceder los datos de consumo necesarios. Según algunos estudios, un gran porcentaje de los consumidores aceptarían que se enviaran sus datos de consumo (parcial o total), si, con ello, se consigue un ahorro significativo de energía y, en consecuencia, una reducción del coste energético mensual [42, 43].

2.3.2 Industria y Producción Inteligente

En un sector de tan alta competitividad como es el de la industria, es normal que se quieran optimizar cada uno de los aspectos relevantes en la producción, sin embargo, debido a la complejidad de las fábricas, no es fácil tener en cuenta todas las variables, incluso con los avanzados sistemas de control actuales [44].

El IoT puede introducirse en la industria y prestar servicios que puedan mejorar la producción en muchos aspectos, algunos ejemplos son:

- La implementación de un sistema de comunicación basado en sensores para la obtención en tiempo real de datos de actuadores, máquinas-herramientas y, en general, cualquier tipo de infraestructura relevante en la producción. Además de para recoger datos de la producción en sí, es también interesante la monitorización continua del estado de la infraestructura, lo cual podría permitir que se llevaran a cabo arreglos o sustituciones antes de que ocurriera un error crítico [4] (en contraposición a realizar revisiones periódicas).
- Este sistema de comunicación, al contrario que los sistemas alámbricos habituales, no es estático, ya que cada uno de sus componentes, además de la topología de la red, es reconfigurable. Por tanto, se da la posibilidad de que, con el mismo equipo, se puedan realizar una gran cantidad de proyectos distintos, ahorrando tiempo y recursos en la obtención y reorganización de los equipos de comunicación de la fábrica [44].
- El procesamiento de datos en la nube puede ser la solución para la gestión de datos de las redes de comunicaciones de la fábrica. Esto conlleva un ahorro en equipos físicos de gestión de datos [44], además de que permite que toda la información, ya procesada, sea visible y analizable de forma sencilla, facilitando así la toma de decisiones de operarios y supervisores [45].
- La agilidad y realimentación en la toma de decisiones brindada por las redes de comunicaciones IoT podrían proporcionar una personalización del producto nunca antes vista, incluso en etapas muy tardías del desarrollo. Además, dependiendo del producto, se podría personalizar incluso tras el desarrollo, mediante la descarga de actualizaciones de software. Por último, se podría monitorizar el propio estado del producto para su reemplazo o arreglo, así, los clientes tendrían también la posibilidad de realizar el mantenimiento del producto antes del fallo fatal [4].

La posibilidad de introducir estas tecnologías puede ser muy atractivo para las empresas, ya que es una forma de reducir los costes de producción y, así, poder comercializar sus productos a un precio más competitivo con el mismo o superior margen de beneficios. De hecho, a lo largo de los años, el porcentaje de empresas que se plantean acoger estas tecnologías ha ido subiendo aceleradamente [46], según una encuesta realizada en 2022 por la empresa CESMII junto a Smart Manufacturing Experience (SME), sólo un poco más de la mitad

de las empresas tenía pensado implementar estas tecnologías [47]. Al año siguiente, otro estudio realizado por Rockwell Automation apuntaba a que este dato había aumentado hasta un 97% de los encuestados [48].

Por último, la implementación de estas tecnologías ayuda a reducir el malgasto de energía y materiales [45], por ello, también supone una mejora en cuanto a la reducción de la huella de carbono derivada de la producción.

2.3.3 Venta Minorista Inteligente

Las tecnologías IoT son tan versátiles que pueden acompañar a un producto en todas sus etapas, desde la producción hasta la venta existen posibilidades de optimización que pueden cubrir estas tecnologías. En la venta minorista, se utiliza la gran capacidad de obtención y gestión de la información del IoT para recoger información que, con tecnologías anteriores, nunca antes se había podido recopilar [49]. Esto permite que se puedan ofrecer nuevos servicios tanto a vendedores minoristas como a sus clientes:

- Mediante una malla de sensores de presencia IoT, se puede hacer un estudio de las rutas más populares dentro de los establecimientos y, en base a esos datos, concentrar ofertas en aquellas rutas [4, 49]. También se pueden implementar sensores IoT que detecten qué productos se están vendiendo y en qué cantidad, pudiendo tener así una aproximación más certera de las preferencias de los clientes [4] y, así, orientar más acertadamente las campañas de marketing [49].
- La monitorización de los productos vendidos es, a la vez, información sobre qué productos se deben reponer [4]. Esta información en tiempo real puede ser usada para conseguir una comunicación fluida entre el establecimiento y los proveedores, disminuyendo así la complejidad de las operaciones de la cadena de suministro [49].
- La información sobre los productos también puede ser revelada a los clientes, estos, mediante el uso de aplicaciones en sus teléfonos móviles, podrían introducir el nombre del producto (o, por ejemplo, la lista de la compra completa) que necesitan y la aplicación podría indicar la ubicación en la que se encuentra [4, 49]. Este servicio, sobre todo en establecimientos de mayor extensión (grandes almacenes, centros comerciales...), sirve para agilizar las compras, para que el cliente encuentre lo que busca y para que, en general, se tenga una experiencia de compra más positiva y dinámica.
- La experiencia de compra del cliente se puede mejorar también al reducir los tiempos de espera para acceder al pago de los productos. Aunque las cajas de autopago llevan existiendo varios años, las tecnologías IoT puede dejarlas obsoletas ofreciendo alternativas más cómodas para el cliente. Una opción es que, con el mismo sistema que monitoriza la retirada de productos de su estantería, identificar los productos que lleva cada cliente, pudiendo así realizar el pago automáticamente al salir del establecimiento [49, 50]. Esto abre la posibilidad de implementar sitios de venta en las que no hagan falta trabajadores para realizar el cobro [49, 50], pudiendo enfocar el trabajo del personal en otras actividades.

La implementación de estas tecnologías supondría, según algunas encuestas, un aumento de la clientela por el simple hecho de ofrecer experiencias dinámicas [51, 52] (como pueden ser pantallas interactivas, programas de realidad aumentada, realidad virtual...) y, así, aumentar la cantidad de productos mostrados al cliente al poder capturar su atención por más tiempo.

2.3.4 Monitorización de la Salud en Tiempo Real

Las tecnologías IoT no sólo sirven para optimizar procesos, sino que también se pueden utilizar para facilitar el día a día de personas que precisen de algún tipo de monitorización. Al utilizarse tecnologías inalámbricas, estas personas pueden disponer de una mayor autonomía, lo que puede mejorar su estado de ánimo (o, incluso, su estado físico) y también ayudar a combatir el aislamiento que suelen sufrir [53]. Las redes inalámbricas portátiles que hacen esto posible reciben el nombre de Red de Área Corporal Portátil, *Wearable Body Area Network* o WBAN [54, 55], aunque también existe la posibilidad de implementar estaciones fijas al alcance del paciente [56] (con la idea de realizar una monitorización que, aunque periódica, mucho más frecuente que las visitas médicas).

Al tener en tiempo real los valores de las variables sanitarias del paciente, se puede implementar un sistema que avise a los servicios de emergencia ante la detección de valores inusuales en estas constantes. Además, estos datos pueden servir para analizar los efectos y la evolución de la condición que sufra el paciente, pudiéndose así modificar el tratamiento de forma más adecuada y personalizada. No tienen por qué medirse

sólo variables intrínsecamente corporales, sino también aquellas que puedan indicar distintas emociones, estrés o ansiedad mental [56, 57].

Aunque se ha hablado del uso de estos dispositivos mientras el paciente realiza vida normal, también es interesante considerar el caso en el que se utilicen junto a sistemas alámbricos más habituales, dentro de hospitales y otros centros sanitarios. Se podría realizar un monitoreo individual o por habitaciones (varios dispositivos conectados a un sólo Nodo de Salida en la habitación), reduciendo así costes y facilitando el análisis de datos [55].

A continuación, se presentan algunos ejemplos de variables cuya monitorización es de alta relevancia en este tipo de aplicaciones y cómo medirlas:

- **Ritmo Cardíaco:** permite obtener información sobre la salud cardiovascular del individuo, abriendo la posibilidad de realizar avisos si se está produciendo un comportamiento anómalo del corazón. Los sensores de ritmo cardíaco pueden ser eléctricos u ópticos, los primeros utilizan el potencial eléctrico generado por el corazón al bombear sangre para determinar los latidos, los segundos emiten una luz a través de la piel, en función de la luz que traspase, se determina la cantidad de sangre que está circulando [58].
- **Saturación de Oxígeno:** permite conocer la cantidad de oxígeno que transporta la sangre a los tejidos corporales, además de poder analizar el funcionamiento de los pulmones. Los sensores que miden esta magnitud funcionan de forma muy parecida al sensor óptico del ritmo cardíaco, en este caso, se proyectan haces de luces (sobre la piel, o en las uñas) de dos longitudes de onda distintas, dependiendo de la oxigenación, la sangre absorbe un tipo de luz u otra [59, 60].
- **Actividad Cerebral:** permite detectar cambios en la actividad cerebral y, así, poder diagnosticar trastornos cerebrales como la epilepsia, además de daños y tumores cerebrales. Esta actividad se determina mediante la medición de los pulsos eléctricos generados por las neuronas [61].
- **Actividad Electrodérmica:** a partir de los cambios en temperatura y electricidad de la piel (producidos por los nervios y las glándulas sudoríparas) se pueden detectar ciertos estados emocionales o físicos (por ejemplo, sofocos provocados por la menopausia o grandes excitaciones del sistema nervioso autónomo) [62]. Para calcularla, los sensores electrodérmicos miden la variación de una corriente eléctrica aplicada entre dos electrodos, los cuales se colocan pegados a la piel [63].

2.3.5 Inteligencia Artificial en IoT

La inteligencia artificial puede permitir que las tecnologías IoT no sólo sirvan para recoger información, sino que, además, esta pueda ser procesada e interpretada automáticamente. Esto es especialmente interesante para redes IoT que generen una gran cantidad de datos a altas velocidades, tanto que los humanos no podamos procesarlos adecuadamente [64].

La IA se puede implementar de muchas maneras, aunque normalmente se aplica, junto al IoT, con alguna de las siguientes técnicas [64]:

- **Redes Neuronales:** consiste en un conjunto de procesos interconectados llamados "neuronas", organizados en tres tipos de capas: la capa de entrada, la capa de salida y las capas ocultas.

Las neuronas de la capa de entrada son activadas por los sensores al tomar datos del entorno, ante esta activación, las neuronas de las otras capas también se activan en mayor o menor medida (mediante un sistema de "pesos"). Por último, las neuronas de la capa de salida actúan sobre el entorno, provocando una realimentación con las neuronas de la capa de entrada. Para que una red neuronal alcance un objetivo concreto, es necesario ajustar los "pesos" de cada neurona [65]. Este ajuste se puede realizar de varias formas, pero para la cantidad de datos que se analizan en aplicaciones IoT, quizá el camino a seguir sea el aprendizaje de refuerzo, este permite un entrenamiento autónomo o semiautónomo [64] en el que la propia red puede "comprobar" su rendimiento basándose en funciones de optimización [66]. Tras el entrenamiento, estas redes son capaces de aplicar varias operaciones no lineales en cadena [65] para llegar al resultado más óptimo.

- **Lógica Difusa o Fuzzy Logic:** la lógica clásica nos ayuda a identificar si un enunciado o una condición son verdaderos o falsos, sin embargo, esta clasificación binaria puede no ser suficiente para describir los siguientes pasos a seguir ante la entrada de datos ambiguos o distorsionados y, tampoco, a la hora de representar proposiciones imprecisas (como las que se podrían expresar en lenguaje natural, por ejemplo,

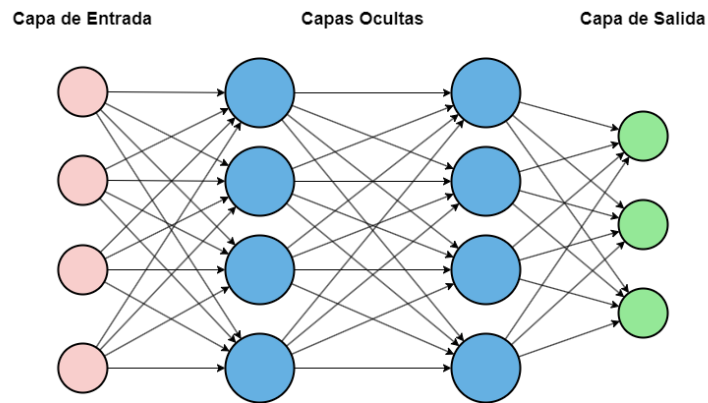


Figura 2.13 Distintas capas que componen las redes neuronales.

la expresión "A es un poco menor que B" es imposible de representar bajo lógica clásica [67, 68]. Para la gestión de datos en aquellos casos en los que exista ambigüedad o imprecisión, la lógica difusa implementa una aproximación de la lógica humana [67, 68], permite clasificar magnitudes en diferentes grupos mediante "grados de membresía", que varían en un rango entre 0 y 1, además de determinar el grado de membresía a un grupo en función de los grados de membresía a otros grupos [69].

Las redes neuronales implementadas pueden aprovechar la gran cantidad de datos recogidos en las aplicaciones IoT para entrenarse, pudiendo así mejorar continuamente y, por tanto, dar mejores resultados con el paso del tiempo [64].

También se hace posible implementar en una red IoT sistemas de reconocimiento de voz o formas, abriendo nuevas posibilidades que, con los métodos tradicionales, o bien eran imposibles, o bien eran mucho más complicadas de implementar [64].

2.3.6 Agricultura Inteligente

Con los pronósticos indicando un aumento significativo de la población mundial a lo largo de este siglo [2], se estima que será necesario doblar la producción de alimentos en los próximos 25 años [70, 71]. Sin embargo, las tierras de cultivos están limitadas por variables como la temperatura, calidad del suelo o la topografía [72]. Esta limitación hace que la opción más prometedora para aumentar la producción de alimentos sea la que propone una mejor gestión de las tierras cultivables disponibles, en contraposición a aumentar las tierras de cultivo pero con una gestión pobre [1, 72].

Los Sistemas de Agricultura Inteligente, *Smart Agriculture Systems* o SAS son aquellos que permiten un monitoreo remoto del estado de los cultivos y, en algunos casos, también permiten la actuación sobre estos de manera automática [71]. Estos sistemas están basados principalmente en tecnologías IoT para la obtención de estos datos y, por tanto, permite que los agricultores puedan analizar de manera más precisa y cómoda las necesidades de sus cultivos, mejorando la eficiencia del espacio cultivable disponible [71, 72]. Aunque la inversión inicial realizada a la hora de obtener estos equipos sea considerable, el ahorro de recursos que conllevan hacen que esta se recupere rápidamente, convirtiéndose en una opción razonable para los agricultores [71]. Tras la instalación, el consumo de energía de los sensores es un gasto incluso despreciable, como se ha indicado en varias ocasiones, estos dispositivos están creados de tal manera que funcionen con una batería pequeña durante meses o años. Sin embargo, también se pueden aprovechar las condiciones en las que se usan los dispositivos (en este caso, al aire libre) y utilizar las energías renovables disponibles para su alimentación energética, en la agricultura, es interesante considerar la instalación de paneles solares [73, 74] para dar energía a estos sistemas de forma autosuficiente y económica.

Medición de Variables Agrícolas

Lo que hace posible el monitoreo del estado de los cultivos es la instalación de sensores a lo largo de las tierras agrícolas. Estos sensores recogen datos no sólo para poder ser analizados en su conjunto por los agricultores, sino que también son enviados a los diferentes actuadores, cuando existen sistemas de actuación automática [71].

Existen muchas variables a tener en cuenta a la hora de garantizar el correcto desarrollo de los cultivos y, aunque distintas, suelen estar relacionadas entre sí. Con un control más preciso de estas, se pueden llegar al

equilibrio que permita el máximo desarrollo de la planta con el mínimo uso de recursos. A continuación, se muestran algunas de las variables más relevantes a monitorizar y por qué tienen tanta relevancia:

- **Temperatura:** la temperatura de la planta como del sustrato que la contiene tiene grandes efectos sobre su desarrollo, de hecho, para cada cultivo existen unos umbrales de temperatura en los que el cultivo puede desarrollarse correctamente, fuera de estos, la calidad disminuye de forma significativa o, simplemente, no se dan las condiciones para que el cultivo crezca [75, 76]. En general, a muy altas temperaturas se impide la correcta fertilización y germinación de las plantas [76], también, aceleran tanto el crecimiento de los cultivos que estos se pueden desarrollar de manera errónea [76], esto da lugar a plantas más pequeñas y, por tanto, a una menor producción [75, 77]. A temperaturas muy bajas, además de los efectos antes mencionados [75], pueden producirse roturas celulares por daños de congelación [76].
- **Humedad del Sustrato:** la cantidad de agua que el terreno tiene disponible en las diferentes etapas de desarrollo de las plantas es una métrica realmente importante, ya que indica si estas han podido realizar una correcta absorción de los nutrientes del sustrato, necesarios para llevar a cabo la fotosíntesis correctamente. Ante escenarios de sequía, los cultivos absorben una menor cantidad de nutrientes del suelo, disminuyendo drásticamente la producción [78], de hecho, en una situación conjunta de altas temperaturas y baja humedad, las consecuencias adversas son mayores que si actuaran por separado [79]. Sin embargo, si el sustrato está demasiado húmedo, el agua ocupa el espacio del oxígeno, provocando problemas de respiración en la planta, también puede propiciar enfermedades en las raíces y el aumento de la actividad microbial [80, 81].
- **Salinidad y Conductividad Eléctrica:** saber la cantidad de sal que contiene el sustrato es de gran importancia, un aumento de esta aumenta la fuerza osmótica del suelo, haciendo más difícil la absorción de agua [82, 83] por las raíces de la planta, además, cuanto más se seque el sustrato, mayor será la fuerza osmótica de este, dificultando aún más la absorción [82, 84]. Al limitar la absorción de agua de la planta, se tiene como consecuencia los mismos efectos que en el caso de sequía, es decir, una disminución del tamaño de las plantas y, por tanto, de la producción [82, 83]. Para poder medir la salinidad se precisa de una medición de la conductividad eléctrica del sustrato (aunque esta también se ve afectada por otros parámetros del suelo [85]), esta se puede medir de varias formas. Una manera es mediante sensores de inducción electromagnética, que se componen de dos bobinas separadas a una distancia conocida, una emite un campo magnético, el cual genera una corriente eléctrica que recibe la otra bobina, el ratio entre ambos campos magnéticos es linealmente proporcional a la salinidad del sustrato [86]. Otra manera es mediante los sensores por inyección de corriente, consiste en la introducción de dos electrodos en el sustrato que generan un campo eléctrico mediante inyección de corriente, otros dos electrodos miden las diferencias de voltaje creadas, la relación entre el voltaje y la intensidad de corriente determina la conductividad del suelo [83].
- **Calidad del Agua:** además de realizar mediciones en el sustrato, es importante también analizar el estado del agua con el que se riegan los cultivos. Un riego de mayor calidad puede aumentar la producción incluso con una menor utilización de agua [87]. Esta calidad se puede medir mediante diferentes magnitudes, entre ellas está el pH, que indica cuán ácido o básico es un medio, es relevante porque cada cultivo tiene un rango de valores de pH en los que el crecimiento es mayor [88], además de indicar qué nutrientes y elementos puede absorber la planta, esto puede ser positivo si los cultivos tienen la posibilidad de absorber más nutrientes, o negativo si en el suelo existen sustancias tóxicas a evitar [88, 89]. Otra medida de calidad del agua es el Total de Sólidos Disueltos, *Total Dissolved Solids* o TDS, que, como su nombre indica, mide la cantidad total de nutrientes y otros elementos disueltos en el agua [90, 91, 92], normalmente se deduce a partir de la conductividad eléctrica del medio (es linealmente proporcional a ella [91, 92]), por tanto, tiene que ver con la salinidad de este. En general, aunque un TDS bajo suele significar una baja cantidad de nutrientes, tampoco debe ser muy alto, ya que esto indica altos niveles de salinidad, con las desventajas que eso conlleva y que ya se han discutido [91].

Un factor a tener en cuenta al realizar estas mediciones es el ruido, es importante implementar técnicas de supresión de ruido para poder luego analizar los datos correctamente y sin muchas anomalías. Quizá el método más utilizado sea la implementación de un filtro paso-baja, este tiene la capacidad de filtrar ruido de alta frecuencia, dejando pasar las componentes relevantes en baja frecuencia. Un ejemplo de este tipo de filtros es el de Media Exponencial Móvil, al que se le introducen la medida actual ruidosa y la media de todas

las medidas anteriores, además de un factor de suavizado normalmente designado como α , para calcular la salida, que será menos ruidosa [71, 93]:

$$S = \alpha M_{actual} + (1 - \alpha) M_{media} \quad (2.3)$$

Otra manera de reducir el ruido es mediante un filtro de Kalman, este, a grandes rasgos, intenta predecir el siguiente estado del sistema en base a mediciones anteriores. Esta predicción se compara con la nueva medida tomada, dependiendo del error cometido, se modifica más o menos los parámetros para realizar una nueva predicción en el siguiente estado [94].

Métodos de Irrigación Inteligente

Aunque la monitorización de las variables mencionadas en sí son útiles y permiten cierto grado de optimización y análisis, también es interesante la posibilidad de implementar actuadores que respondan antes las necesidades de la planta automáticamente. En específico, el control realimentado de la irrigación permite un mayor ahorro del agua, ya que, tanto el momento de inicio del riego como el de finalización está basado en los datos de los sensores, esto hace que también se puedan ahorrar recursos en mano de obra, ya que sólo sería necesaria a la hora de mantener los dispositivos. Existen varias maneras de implementar estos sistemas de irrigación dependiendo de la naturaleza del cultivo o del mecanismo de riego, a continuación se exponen algunos ejemplos:

- **Riego Realimentado por Tiempo:** consiste en la activación de la irrigación por periodos cortos tras una orden de inicialización y hasta que se dé una orden de paro. Estas órdenes se basan en las medidas tomadas por los sensores de humedad, si estas medidas sobrepasan ciertos umbrales configurados por el usuario (al bajar de cierto umbral, se inicia el riego, al sobrepasar otro umbral, se para el riego). Tras cada periodo corto de irrigación es necesario esperar un tiempo considerablemente alto (en relación al tiempo del periodo de irrigación) antes de regar otra vez, ya que es necesario que el sustrato pueda absorber el agua y que, por tanto, el sensor de humedad pueda dar una medida lo más fiable posible (si este tiempo no es lo suficientemente alto, se puede realizar una medida cuando el sistema terrestre se encuentra todavía en una especie de "estado transitorio"). Por tanto, este tipo de irrigación es adecuada para sistemas de riego que tengan estados binarios de activación, es decir, que sólo puedan estar encendidos o apagados.
- **Riego por Voto Mayoritario:** en el caso en el que haya un sólo actuador para un grupo de sensores, este sistema tiene en cuenta la cantidad de estos que reportan que la humedad ha caído por debajo de cierto umbral, si esta cantidad es mayoritaria, se realiza el riego [71].
- **Riego Realimentado con PID:** para sistemas que sí tengan alguna forma de activación parcial (mediante una señal de control PWM [95], por ejemplo), un control PID o Proporcional, Integral y Derivativo puede ser una opción más robusta para la irrigación. El funcionamiento de un PID reside en el cálculo de una señal de control a partir del error actual (P), el acumulado (I) y la velocidad de cambio de este (D), este error se define como la resta entre la referencia o consigna (configurada por el usuario) y la medida actual del sensor.

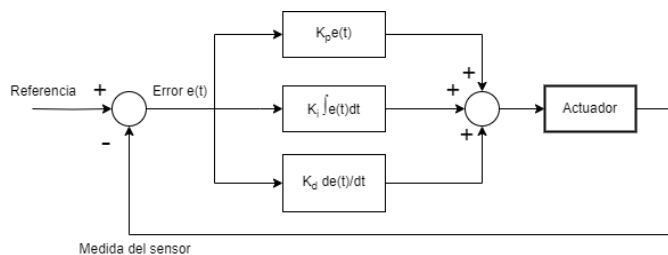


Figura 2.14 Esquema de un controlador PID.

La componente integral ayuda a llegar a un error nulo en régimen permanente, y la componente derivativa ayuda a disminuir la sobreoscilación [71]. En concreto, la disminución de la sobreoscilación es importante en este tipo de sistemas, ya que, debido a la naturaleza de estos, no es posible la retirada de agua si el sustrato ha recibido demasiada, obligando al controlador a esperar a que, por evaporación natural del agua, la humedad baje.

- **Irrigación por Lógica Difusa:** para el crecimiento adecuado de los cultivos, no es necesario que estos tengan un valor de humedad específico, sino que existe un rango de valores aceptables dónde el crecimiento es cercano al óptimo. Por tanto, las variables difusas, que permiten acercamientos probabilísticos y aproximados, pueden ser bastante adecuadas en un sistema de riego [96, 97], además, esto permite prescindir de modelos matemáticos complejos que describan el sistema [97]. A partir de las diferentes funciones de membresía, basadas en medidas de distintas variables, se puede determinar la cantidad de irrigación necesaria en cada momento.
- **Irrigación por Red Neuronal:** mediante el uso de una red neuronal se puede controlar el riego sin usar ningún algoritmo específico, sólo a través de los datos de los sensores (de variables distintas) y de una función a minimizar [98]. Para este tipo de irrigación, al igual que en la mayoría de las aplicaciones de las redes neuronales, es necesario una gran cantidad de datos de entrenamiento de los sensores [71, 99], lo que puede significar un aumento de las prestaciones de los sistemas utilizados para entrenar estas redes [71].

Detección de Enfermedades de los Cultivos

Hasta ahora se ha hablado sobre la monitorización de los cultivos de forma aislada, sin embargo, existen agentes externos a la planta y el sustrato que pueden hacer que la producción baje considerablemente. En el caso de las enfermedades, si son infecciosas, pueden dejar inutilizados campos enteros, por tanto, no es inusual que ciertas tecnologías IoT en la agricultura se hayan orientado a la detección de estas enfermedades, con el objetivo de minimizar los daños provocados por estas.

Los dispositivos que hacen posible la detección de estas enfermedades basan su funcionamiento en la captación de imágenes de las plantas mediante sistemas de percepción. Una forma de implementar estos sistemas es mediante el procesado avanzado de imágenes, atendiendo, por ejemplo, a la distribución de colores y la relación entre estos de la imagen captada, se puede distinguir, con mucha precisión, si una planta está infectada o no [100, 101].

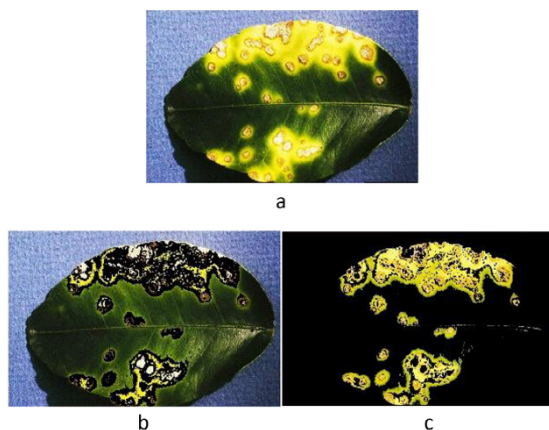


Figura 2.15 Separación mediante discriminación por color de las regiones sanas e infectadas de una hoja, (a) foto inicial, (b) área sana de la hoja, (c) área infectada de la hoja [100].

La principal desventaja de este tipo de métodos es que, para enfermedades o plantas distintas, quizá sea necesario realizar un procesamiento de la imagen totalmente diferente, por lo tanto es más adecuado para plantaciones monocultivo.

Otra forma de realizar estas clasificaciones es mediante el uso combinado de un pre-procesamiento de imagen y de redes neuronales, estas redes son entrenadas con imágenes pre-procesadas (aunque este procesado es más sencillo que si se hiciera sólo este paso, se realizan, por ejemplo, aumentos del contraste, cambios en el color, cambios en el tamaño de la imagen o recortes de las partes relevantes [102]). Dependiendo de las imágenes de entrada y de las capas que conforman la red neuronal, se podrán clasificar más tipos de plantas y enfermedades con mayor precisión [102, 103], ofreciendo una opción más flexible que en el caso en el que se sólo se utiliza procesado de imágenes.

En cualquier caso, es importante recalcar que los cálculos previamente comentados se realizan en sistemas que, aunque siguen siendo IoT, precisan de una alimentación mayor [104, 105]. Esto es debido a las tareas de procesamiento de imagen que deben realizar estos dispositivos, las cuales, en ocasiones, conllevan una carga computacional y un consumo de energía relativamente altos. Además, el precio de estas tecnologías de captación de imágenes son, en general, más costosas que los sensores y microprocesadores habituales en este tipo de aplicaciones.

3 Diseño de Experimentos

Una vez explicados tanto los principios básicos de las tecnologías IoT, sus posibilidades y sus distintas aplicaciones, en este capítulo se procederá a describir el funcionamiento de los experimentos desarrollados para la realización de este trabajo.

En cada experimento se implementa un sistema de sensores distinto, ambos con el objetivo de monitorizar las variables relevantes del sustrato terrestre, en el que hay una planta. Las medidas tomadas por estos sensores son utilizados para controlar un sistema de Riego Realimentado por Tiempo, cuyo funcionamiento ya se ha descrito en la sección 2.3.6.

3.1 Elementos Comunes a Ambos Experimentos

Antes de hablar de los distintos sistemas de medida implementados, se van a listar y exponer los elementos que comparten estos dos experimentos.

3.1.1 Microcontrolador ESP8266 en Placa de Desarrollo D1 Mini

El microcontrolador escogido para estos experimentos, y que sirve para gestionar ambos sistemas, es el ESP8266, este es usado principalmente por sus capacidades de conexión a internet por Wi-Fi y su coste relativamente bajo [106, 107, 108]. En concreto, se ha usado la placa de desarrollo D1 Mini, esta tiene integrada, además del propio microcontrolador ESP8266, algunos componentes que, en otras placas, tendrían que haberse instalado externamente: un módulo WLAN para poder realizar las conexiones a internet, un regulador de voltaje a 3.3V [108] para evitar la alimentación a 5V (si se llegaran a aplicar 5V sin el regulador en alguno de los pines, el microcontrolador podría dañarse o, directamente, dejar de funcionar) [108, 109] y un módulo que permita la utilización del driver CH340 para la conversión de bus USB a serie [106, 107, 108], entre otros.

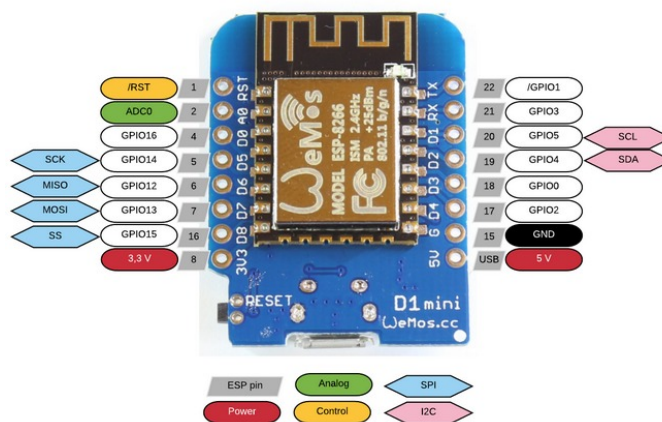


Figura 3.1 Pinout de la placa de desarrollo D1 Mini [106].

Como se puede observar en la Figura 3.1, esta placa cuenta con varios pines al alcance del usuario, todos ellos pueden ser de utilidad de una manera u otra en el desarrollo de sistemas de sensores:

- **Pines de Propósito General de Entrada/Salida, *General Purpose Input/Output pins* o pines GPIO:** como su nombre indica, sirven para realizar actuaciones o lecturas de carácter general. En específico, esta placa cuenta con 11 pines GPIO disponibles (los pines de este tipo del 6 al 11 están reservados para comunicaciones con la memoria flash del dispositivo [109]). Sin embargo, algunos de ellos están preparados para utilizarse de manera que tengan otras funcionalidades, por ejemplo, los pines GPIO1, GPIO3, GPIO15, GPIO13 y GPIO2 pueden usarse para la comunicación por puerto serie (por hardware) del tipo *Universal Asynchronous Receiver / Transmitter* o UART [110].
- **Pin de Entrada/Salida Analógica:** sirve para realizar lecturas o actuaciones analógicas (aunque en algunos pines GPIO se puede usar la técnica *Pulse Width Modulation* o PWM por software para generar un efecto similar [110]), sin embargo, sólo cuenta con 10 bits de resolución y un rango de entrada de 0 a 3.3V (algunas placas tienen un rango de 0 a 1V) [109, 110], lo cual no puede ser suficiente dependiendo de la aplicación, en particular, para aquellas que necesiten una precisión de medida alta.
- **Pines para la Implementación de comunicaciones *Serial Peripheral Interface* o SPI:** los pines GPIO14, GPIO12, GPIO13 y GPIO15 se pueden utilizar para enviar las señales *Serial Clock* o SCK, *Master Input Slave Output* o MISO, *Master Output Slave Input* o MOSI y *Slave Select* o SS, respectivamente, todos necesarios para la implementación del estándar de comunicaciones SPI. La comunicación SPI se basa en el envío de señales síncronas entre un dispositivo "maestro" y uno o varios dispositivos "esclavos", para la sincronización entre estos dispositivos se usa la señal SCK, al realizarse la sincronización, se utilizan las señales MISO y MOSI para enviar datos de "maestro" a "esclavo" y viceversa. Por último, en el caso en el que existan varios "esclavos", el "maestro" haría uso de la señal SS para elegir a qué "esclavo" enviar los datos [111].
- **Pines para la Implementación de comunicaciones *Inter-Integrated Circuit* o I²C:** los pines GPIO4 y GPIO5 se pueden utilizar para enviar las señales *Serial Clock* o SCL y *Serial Data* o SDA, respectivamente, necesarias en la implementación de las comunicaciones I²C. Al igual que en las comunicaciones SPI, la señal SCL sirve para sincronizar a un dispositivo "maestro" (varios en el caso de sistemas I²C multi-maestro) con uno o varios dispositivos "esclavos", la señal SDA sirve para la transmisión de datos, tanto el "maestro" como los "esclavos" la usan para el envío de información. Además, todos los dispositivos se conectan a estas dos señales en topología tipo estrella [112].

Aunque este microcontrolador se puede programar de varias formas, se ha elegido el entorno Arduino IDE, principalmente por su compatibilidad con otras librerías que hacen posible una programación sencilla que permita incorporar todas las características buscadas en estos experimentos y, así, poder completar los objetivos propuestos en cada uno. En Arduino IDE es necesario instalar *Board Cores* o Núcleos de Placa para poder programar microcontroladores que, por defecto, son incompatibles con este entorno. Debido a la accesibilidad pública de la información del hardware y software de los productos Arduino, existe una gran cantidad de *Board Cores* creados por la comunidad. Por tanto, se ha utilizado el *Board Core* "ESP8266", creado por la comunidad "ESP8266 Community" [113] para poder programar este microcontrolador de forma conveniente y sencilla. Por último, es importante añadir que, durante estos experimentos, el microcontrolador se ha alimentado mediante la conexión por USB a la red eléctrica.

3.1.2 Recolección y Visualización de Datos en la Nube

Aunque la placa de desarrollo D1 Mini es la que hace la conexión a internet, es necesario un dominio al que enviar los datos recolectados para guardarlos y poder analizarlos. Para ello, en estos experimentos se ha utilizado *ThingSpeak*, de la empresa MathWorks, este servicio ofrece una plataforma que permite cumplir con los requisitos antes nombrados y, además, facilita la visualización de los datos enviados. En *ThingSpeak* el almacenamiento de datos se hace en lugares llamados "canales", con una cuenta gratuita se pueden tener hasta 4 canales simultáneos, cada uno pudiendo mostrar hasta 8 variables al mismo tiempo (cada una almacenada en un "campo" distinto), por tanto, a la hora de recuperar los datos de la nube mediante funciones de Matlab (específicamente creadas para la plataforma) se debe indicar el canal del cual recoger los datos.

El envío de datos también se facilita usando esta plataforma, ya que existe una librería para Arduino IDE, creada por MathWorks, llamada "ThingSpeak Communication Library" [114], que permite, con algunas funciones especializadas, enviar datos de los sensores a la nube de forma sencilla y con una sola línea de código.

3.1.3 Sistema de Irrigación

Para implementar el sistema de riego, se ha utilizado un relé electromagnético conectado a una bomba de agua, esta última sumergida en un recipiente con agua. El relé recibe alimentación del microcontrolador además de la señal que permite el control del sistema de irrigación.

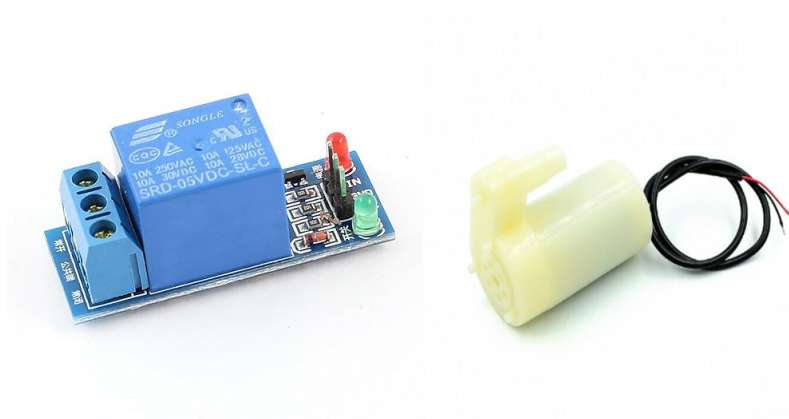


Figura 3.2 A la izquierda, módulo que incorpora el relé "SRD-05VDC-SL-C" de la empresa Ningbo Songle Relay [115], a la derecha, bomba de agua "Submersible 3VDC Vertical Pump" de la empresa Adafruit [116].

3.2 Sistema de Medida con Sonda *Capacitive Soil Moisture Sensor v2.0*

El objetivo de este primer experimento era el de controlar el sistema de irrigación antes expuesto mediante los datos de humedad recogidos de la sonda *Capacitive Soil Moisture Sensor v2.0*, esta cuenta con tres pines al alcance del usuario, uno de alimentación con rango de 3.3V a 5V, uno de tierra y otro con salida analógica en el rango de 0 a 2.3V [117].

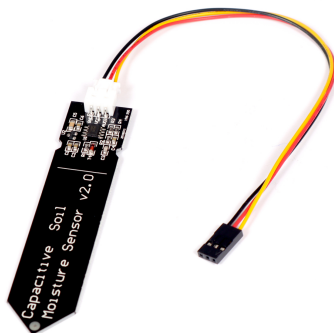


Figura 3.3 Sonda *Capacitive Soil Moisture Sensor v2.0*.

Esta sonda está compuesta por dos electrodos separados por un material dieléctrico, si cambia la humedad de este material dieléctrico, cambia la capacitancia entre los electrodos. Para realizar la medición de humedad en sí, se aplica una tensión en los electrodos y se mide el tiempo que tarda en cargarse, el cual es proporcional a la humedad del material dieléctrico [118, 119].

3.2.1 Convertidor Analógico-Digital ADS1115

Aunque sería posible conectar esta sonda al pin de entrada analógica del microcontrolador y, de hecho, se implementó así en un principio para realizar pruebas, sólo se obtendrían 10 bits de resolución, lo cual, en muchas aplicaciones, puede no ser suficiente. Por tanto, se decidió usar un convertidor analógico-digital externo, en concreto, se ha utilizado el modelo ADS1115. Perteneciente a la familia ADS111X, presenta características destacables y convenientes en su uso para la implementación de sistemas IoT [120]:

- Una resolución de 16 bits, para realizar mediciones precisas.
- Un tamaño relativamente pequeño (2 mm × 1.5 mm × 0.4 mm), que permite su instalación en sistemas en los que el espacio puede ser limitado, además de disminuir las dimensiones totales del sistema, habilitando la implementación de más dispositivos de medida en un mismo espacio.
- Un bajo consumo general, lo cual puede ayudar en el cumplimiento de las restricciones de consumo de energía, tan limitantes en este tipo de aplicaciones.

El ADS1115 cuenta con cuatro entradas analógicas para realizar conversiones analógico-digitales de hasta cuatro sensores distintos, esto es posible gracias a que envía los datos mediante unas salidas SCL y SDA, es decir, comunicándose por I²C. Además, cuenta con otro pin, llamado ADDR, que sirve para que el dispositivo con el que se está comunicando elija el sensor del que desea obtener los datos.

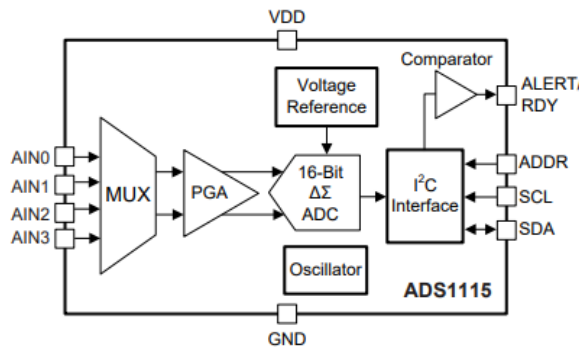


Figura 3.4 Esquema que muestra los distintos componentes del convertidor analógico-digital ADS1115 [120].

Para integrar este convertidor en el sistema de forma más sencilla, se ha usado un módulo que facilita el acceso del usuario a los pines de este (pines de alimentación, de tierra, de SDA/SCL y de las distintas entradas analógicas). Este módulo, de base, sólo facilita el acceso a dos entradas analógicas (aunque se pueden soldar pines para incorporar las otras dos entradas), sin embargo, cómo en este experimento se hacen lecturas de un único sensor, es más que suficiente.

Para poder recibir los datos de este convertidor con el entorno Arduino IDE, se han utilizado varias librerías:

- **"Adafruit_ADS1X15" de Adafruit:** es un driver tanto para el convertidor ADS1015 como para el ADS1115, ofreciendo funciones específicas para poder leer fácilmente de estos dispositivos [121].
- **"Adafruit_BusIO" de Adafruit:** es una librería de ayuda que sirve para simplificar tanto las comunicaciones SPI como las I²C [122].

3.2.2 Calibración de la Sonda

Los datos recibidos del convertidor analógico-digital no se pueden recopilar directamente como datos de humedad, por tanto, es necesario realizar un ajuste lineal de estos para poder analizarlos en las unidades deseadas (tanto por ciento de humedad). Para ello se realiza una calibración un tanto heurística pero eficaz, ya que, si no se hiciera de esta manera y hubieran unos valores fijos de calibración, es posible que se cometieran errores de medida provocados por las diferentes humedades atmosféricas, los diferentes materiales dieléctricos utilizados o las pequeñas variaciones en el diseño de la sonda. Los pasos a seguir para realizar la calibración son los siguientes [117, 119]:

1. Apuntar el valor dado por la sonda cuando esta mide la humedad del aire, este valor representará el 0% de humedad.
2. Apuntar el valor dado por la sonda cuando esta mide la humedad del agua, este valor representará el 100% de humedad.
3. Ajustar linealmente los datos leídos de la sonda entre estos dos valores, o bien con la función *map* de Arduino, o bien con la siguiente fórmula:

$$Humedad(\%) = (Val_{Medido} - Val_{Aire}) \frac{100}{Val_{Agua} - Val_{Aire}} \quad (3.1)$$

3.2.3 Conexión del Sistema Completo

Tras la calibración de la sonda, ya se pueden integrar todos los componentes en conjunto y realizar el conexionado entre ellos. Aunque en la Figura 3.5 se pueden observar todas las conexiones de forma inmediata, se explicarán igualmente:

- El microcontrolador ESP8266 se ocupa de dar alimentación a 3.3V a todos los demás componentes del sistema, cabe destacar que la sonda es alimentada a través del módulo del ADS1115 y que la bomba de agua es alimentada a través del pin común del relé, cuando este último se activa. Además, todos los componentes están conectados a la tierra del ESP8266.
- Para la lectura de datos de la sonda, primero se ha conectado la salida analógica de la sonda a la entrada analógica número 0 del ADS1115, luego, los pines SDA y SCL del ADS1115 se han conectado a los pines que pueden usarse como SDA (D2) y SCL (D1) en el ESP8266, respectivamente.
- Para el riego en sí, primero se ha conectado el positivo de la bomba al pin Normalmente Abierto, *Normally Open* o NO del relé, luego, la entrada que decide el estado del relé se ha conectado al pin D0 del ESP8266, este pin se usa en esta aplicación como salida digital.

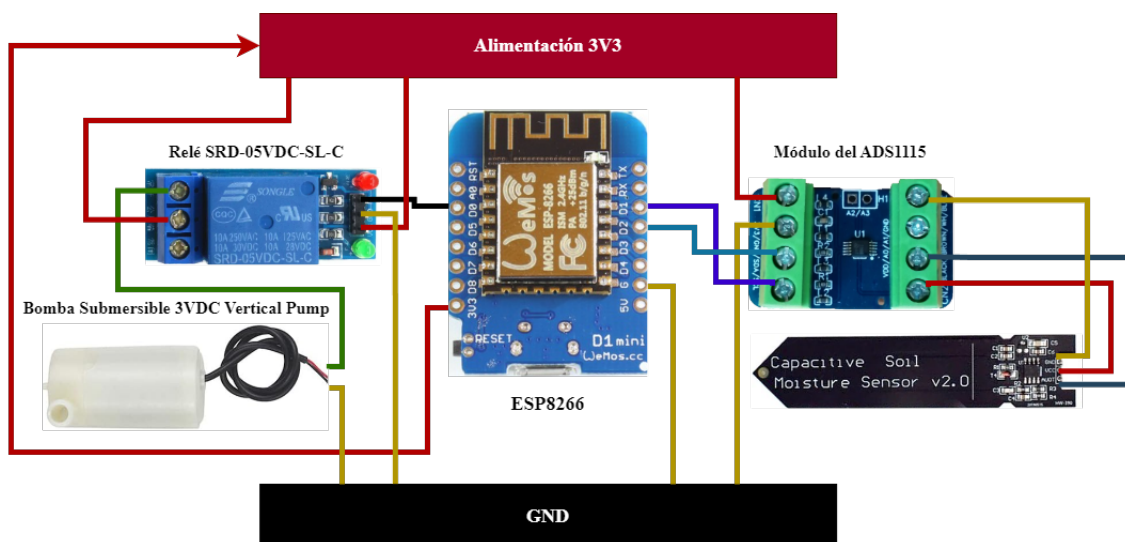


Figura 3.5 Esquema que muestra el conexionado del sistema de medida completo.

3.2.4 Programa Utilizado

A continuación, se ilustrará en la Figura 3.6, mediante un diagrama de flujo, el funcionamiento general del código programado en el microcontrolador ESP8266, encargado de gestionar la lectura de los sensores y el sistema de riego, además de mandar los datos recogidos a la nube:

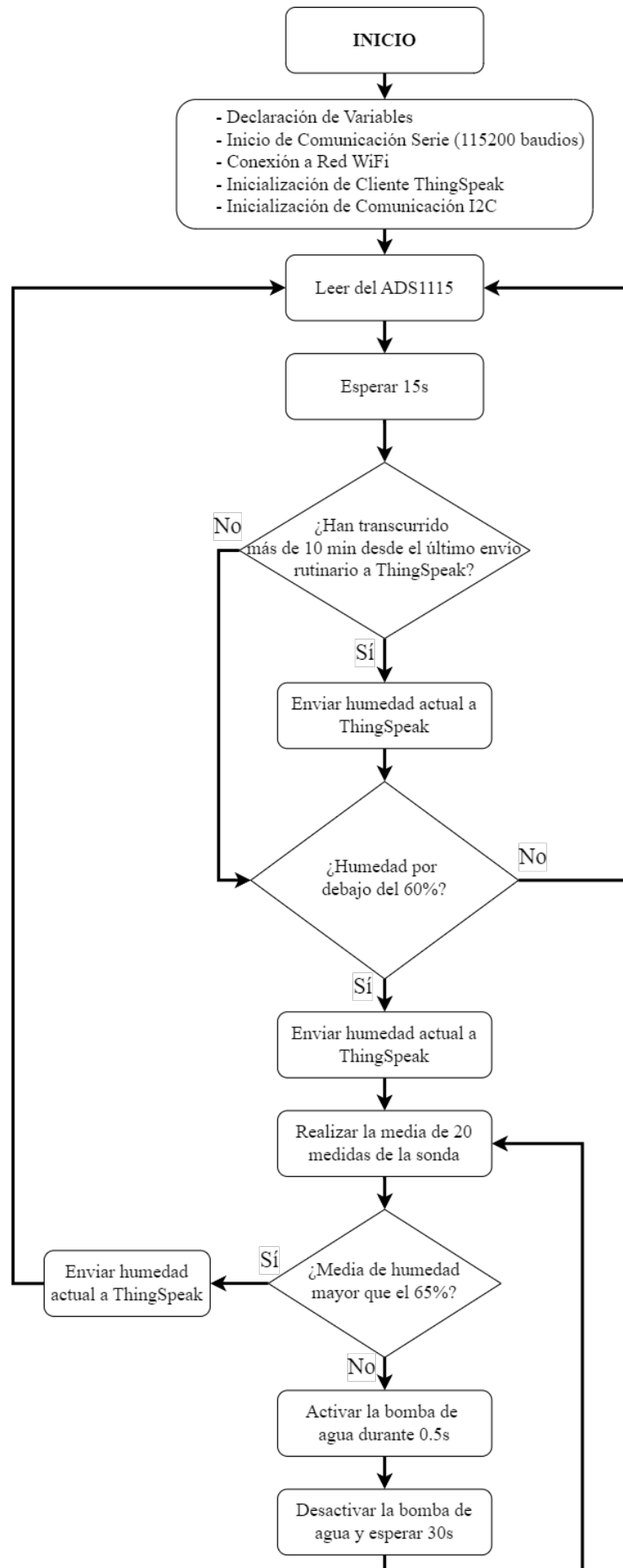


Figura 3.6 Diagrama de flujo que muestra el control del sistema realizado en este experimento.

Se procede a realizar algunos comentarios para justificar varias acciones llevadas a cabo:

- La espera de 15s se realiza para hacer un muestreo constante pero no excesivo de la humedad del sustrato, pensado para activar el sistema de riego de una forma rápida en cuanto esta baja del umbral especificado.
- El envío rutinario a la nube de los datos de humedad se realiza cada 10 minutos porque el sistema no evoluciona de tal forma que justifique un intervalo menor de envío. De esta manera, se reducen los envíos de datos repetidos a ThingSpeak, lo cual permite una visualización más limpia y concreta además de un análisis de los datos más sencillo.
- El envío de datos de humedad a ThingSpeak después de comprobar si la humedad está por debajo del umbral inferior se realiza con el objetivo de visualizar de manera correcta el último estado de la humedad antes de que se inicie el sistema de riego (esto se ha pensado de esta manera porque lo más probable es que el envío rutinario de la humedad no se active al mismo tiempo).
- Los umbrales de humedad (60% para empezar a regar y 65% para terminar de regar) se han escogido en base a datos experimentales tomados de la planta en dos situaciones: cuando precisaba de riego y cuando se había regado. Estas situaciones han sido identificadas mediante el riego manual realizado en los meses previos al experimento (sin embargo, existen algunos matices sobre la elección de estos umbrales, que se discutirán en el siguiente capítulo).
- Los tiempos en los que la bomba está activada y desactivada (durante el riego) se han obtenido mediante prueba y error.

3.3 Sistema de Medida con Sonda MEC10

El objetivo de este segundo experimento era el de controlar el sistema de irrigación descrito mediante los datos recogidos de la sonda MEC10, de la empresa INFWIN. Esta sonda es capaz de recoger datos de temperatura, humedad, conductividad eléctrica, salinidad, total de sólidos disueltos y constante dieléctrica del sustrato [123]. Por tanto, no sólo se han recogido datos de humedad para el control del sistema de irrigación, sino que también se han recopilado datos de las otras variables para tener la posibilidad de realizar un seguimiento más preciso del estado del sustrato y de la planta. En la Figura 3.7 se pueden observar unos electrodos, los cuales van enterrados en el sustrato y realizan las medidas utilizando diversos métodos explorados en la sección 2.3.6.



Figura 3.7 Sonda MEC10.

3.3.1 Capa Física RS-485 y Protocolo Modbus

Antes de exponer cada uno de los cables que tiene esta sonda y sus funciones, es necesario explicar la capa física que hace posible el envío de datos de la sonda al usuario: el RS485. El RS485 es un estándar de interfaz alámbrico de la capa física, de altas prestaciones en cuanto a rango de operación y velocidad de transmisión [124]. Este estándar se basa en la diferencia de potencial entre dos cables (esta pareja de cables es denominada "par trenzado"), uno de ellos da una señal y el otro la misma señal pero con polaridad inversa, el signo de la diferencia de estos dos voltajes determina si se ha transmitido un "1" lógico o un "0" lógico. Estos cables normalmente son nombrados como "A" y "B", definiendo los distintos estados lógicos de la siguiente manera [124, 125, 126]:

$$\text{Si } V_A - V_B < -0.2V \rightarrow \text{"1" lógico}$$

$$\text{Si } V_A - V_B > +0.2V \rightarrow \text{"0" lógico}$$

Estos márgenes de $\pm 0.2V$ son necesarios para tener en cuenta el ruido que pueda haber en la transmisión [124], sobre todo para evitar envíos erróneos de datos cuando el transmisor se encuentra en un estado de reposo. Aún así, debido a que estos estados se determinan mediante diferencias de potencial, si ambos cables del par trenzado están sometidos al mismo ruido, al realizar la diferencia, los componentes de este ruido se cancelarán en su mayoría [125].

Todos los dispositivos que utilizan este estándar se conectan entre sí mediante un bus, del cual se pueden extender hasta 32 derivaciones (es decir, 32 dispositivos). Esto es posible gracias a que cada uno de los emisores utiliza el bus sólo el instante en el que transmite los datos, evitando así las interferencias entre dispositivos [124, 125].

Es importante aclarar que, aunque de esta manera se permiten las comunicaciones entre varios dispositivos, el RS485 no tiene un protocolo de comunicación por defecto, en el caso de la sonda *MEC10*, se utiliza el protocolo Modbus, que, aunque es el más común a la hora de usar esta capa física, se pueden usar otros.

El protocolo Modbus implementa la comunicación entre un dispositivo "maestro" y otros "esclavos", en particular, el "maestro" envía mensajes de petición a los "esclavos", los cuales responden con los datos pedidos [127]. Por tanto, un mensaje Modbus contiene la dirección del "esclavo", un código de función (por ejemplo, lectura o escritura de un registro), la dirección del registro y el dato si lo hubiera [123]. En este experimento no ha sido necesario escribir en ningún registro, sólo se han realizado lecturas de los distintos registros de la sonda, los cuales vienen indicados por el fabricante y se pueden consultar en la Figura 3.8.

Parameters	Register Addr. (HEX/DEC)	Data Type	Modbus Function Code(DEC)	Range and Comments	Default Value
TEMPERATURE	0x0000 /0	INT16 RO	3/4	-4000-8000 for -40.00-80.00°C.	N/A
VWC-Volumetric Water Content	0x0001 /1	UINT16 RO	3/4	0-10000 for 0-100%	N/A
EC-Electrical Conductivity	0x0002 /2	UINT16 RO	3/4	0-20000 for 0-20000us/cm	N/A
SALINITY	0x0003 /3	UINT16 RO	3/4	0-20000 for 0-20000mg/L	N/A
TDS	0x0004 /4	UINT16 RO	3/4	0-20000 for 0-20000mg/L	N/A
EPSILON	0x0005 /5	UINT16 RO	3/4	88-8188 for 0.88-81.88	N/A

Figura 3.8 Tabla proporcionada por el fabricante que indica los registros a leer para obtener las distintas medidas tomadas por la sonda, además de los rangos y unidades de estas magnitudes [123].

Una vez visto tanto el estándar RS485 como el protocolo Modbus, es más sencillo explicar para qué sirven cada uno de los cables de la sonda [123]:

- Un cable rojo y otro negro, que sirven como conexión a alimentación y a tierra, respectivamente.
- Un cable amarillo y otro blanco, que sirven como cables "A" y "B" de RS485, respectivamente.
- Un cable de color verde, cuya función es la de imponer en la sonda unas condiciones genéricas (también llamado "modo de configuración") para la comunicación Modbus (condiciones como la dirección, la velocidad de transmisión, la paridad...), si se conecta a la alimentación. En este experimento, como no se necesitan conectar más sensores a un mismo bus RS485, se ha decidido, por sencillez, trabajar en estas condiciones genéricas en todo momento.

3.3.2 Módulo MAX485

Para que la sonda y el ESP8266 puedan comunicarse, es necesario añadir un módulo intermediario que traduzca paquetes de datos UART a un formato válido para RS485 y viceversa [128]. En este experimento se ha usado el módulo MAX485, que no es más que una placa que facilita el acceso por parte del usuario a los pines del transceptor integrado homónimo. Esta placa cuenta, además de con los pines utilizados para comunicaciones RS485, con los siguientes pines: *Receiver Enable* o RE, *Driver Enable* o DE, *Driver Input* o DI y *Receiver Output* o RO [126, 128]. Los pines RE y DE determinan el papel del dispositivo: si se desea configurarlo como receptor, se deben conectar ambos pines a tierra, si, por el contrario, se desea que sea emisor, ambos pines deben ir conectados a alimentación. Sin embargo, en este experimento, el ESP8266 debe funcionar de las dos maneras, como emisor para realizar la petición de datos a la sonda y como receptor para recoger los datos que envía la sonda, por tanto, ambos pines irán conectados a una salida digital. Los pines RO y DI se conectan a RX y TX, respectivamente, y permiten la transmisión y recepción de datos mediante UART por parte del ESP8266.

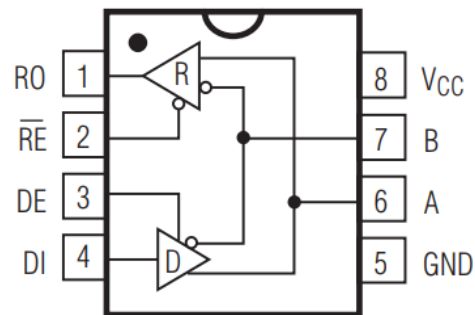


Figura 3.9 Esquemático que muestra los componentes del transceptor MAX485, fabricado por la empresa Analog Devices [129].

Para que el ESP8266, usando el entorno Arduino IDE, se pueda comunicar fácilmente con el módulo MAX485, se ha utilizado la librería "ModbusMaster", creada y mantenida por Doc Walker [130]. Esta librería ofrece, entre otros, funciones que permiten tanto realizar peticiones a la sonda cómo obtener los datos de cada registro por separado de forma sencilla y compacta.

3.3.3 Conexión del Sistema Completo

Tras realizar unas primeras pruebas de lectura de la sonda, ya se pueden integrar todos los componentes en conjunto y realizar el conexionado entre ellos. Aunque en la figura Figura 3.10 se pueden observar todas las conexiones de forma inmediata, se explicarán igualmente:

- Todos los elementos comunes al experimento anterior están dispuestos y conectados de la misma manera. Es decir, tanto la alimentación de los dispositivos, como el funcionamiento del relé y de la bomba, no tiene cambio alguno.
- Se han conectado los pines D7 (RX) y D8 (TX) del ESP8266 a los pines RO y DI del módulo MAX485, respectivamente. Estos pines pueden funcionar como pines de UART por hardware si, en el código, se ejecuta el comando *Serial.swap*. Este comando reasigna estos pines de GPIO1 y GPIO3 a GPIO15 y GPIO13, respectivamente [110].
- Se ha elegido el pin D6 del ESP8266 como salida digital que determina si este es emisor o receptor a la hora de realizar las comunicaciones con la sonda. Esta salida está conectada a los pines RE y DE del módulo MAX485.
- En apartados anteriores, se ha hecho referencia a que la sonda utilizada en este experimentos contaba con, entre otros, un cable blanco, sin embargo, en la Figura 3.10 se ha ilustrado como un cable gris, por claridad.

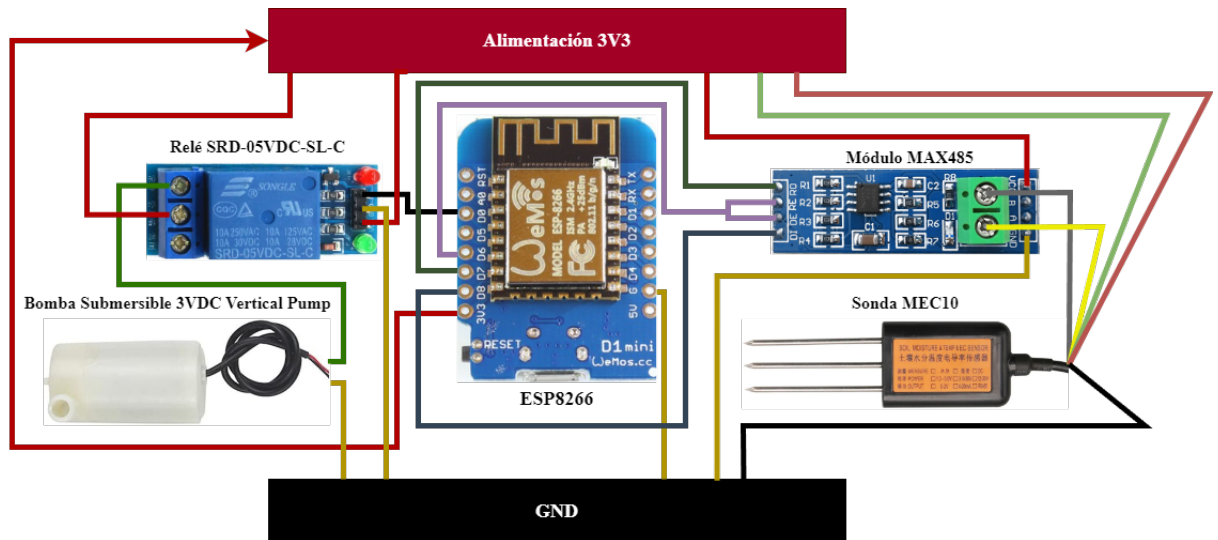


Figura 3.10 Esquema que muestra el conexionado del sistema de medida completo.

3.3.4 Programa Utilizado

Tras explicar cómo se han conectado los distintos elementos de este experimento, se realizará una explicación de ciertos aspectos del código programado en el ESP8266 para el control del sistema. Esta explicación se basará en el diagrama de flujo ilustrado en la Figura 3.11:

- La espera de 15s se realiza por las mismas razones que en el experimento anterior.
- A la hora de leer de la sonda, es necesario primero configurar al ESP8266 como emisor (poner la salida D6 a nivel alto), enviar la petición de lectura de los registros que contienen las magnitudes deseadas y, después, configurar al ESP8266 como receptor (poner la salida D6 a nivel bajo).
- En cada envío rutinario, la transmisión a la nube de cada dato por separado de la sonda se hace 20s después del envío del anterior, se ha hecho de esta manera debido a las limitaciones de *ThingSpeak*, por las cuales, la realización de transmisiones muy seguidas en el tiempo puede conllevar una pérdida de datos. Por ello, se ha reducido el tiempo entre envíos rutinarios de 10 a 8 minutos (hasta que no se envían todas las magnitudes, no se reinicia el temporizador)
- Los umbrales de humedad se han elegido de la misma manera que en el experimento anterior. La diferencia de los valores del umbral en comparación al experimento anterior puede deberse, entre otras cosas, al uso de diferentes métodos de medición de la humedad o a que la sonda capacitiva pudiera tener en cuenta en mayor medida la humedad ambiental, debido a que no podía enterrarse del todo en la tierra por razones de seguridad.
- Cuando la irrigación está activada, sólo se leen las medidas de humedad de la sonda, debido a que las demás magnitudes son irrelevantes a la hora de regar. El tiempo de riego de la bomba se ha aumentado respecto al experimento anterior.
- Cuando en el diagrama se hace referencia a que se haga una "Lectura Correcta", se refiere a que, en el caso de lectura errónea, la sonda devuelve un valor específico (en el experimento anterior no había una forma fiable para saber si la comunicación estaba funcionando correctamente). Por tanto, siempre se comprueba que la lectura sea correcta para evitar errores, sobre todo cuando el sistema de riego está activo.

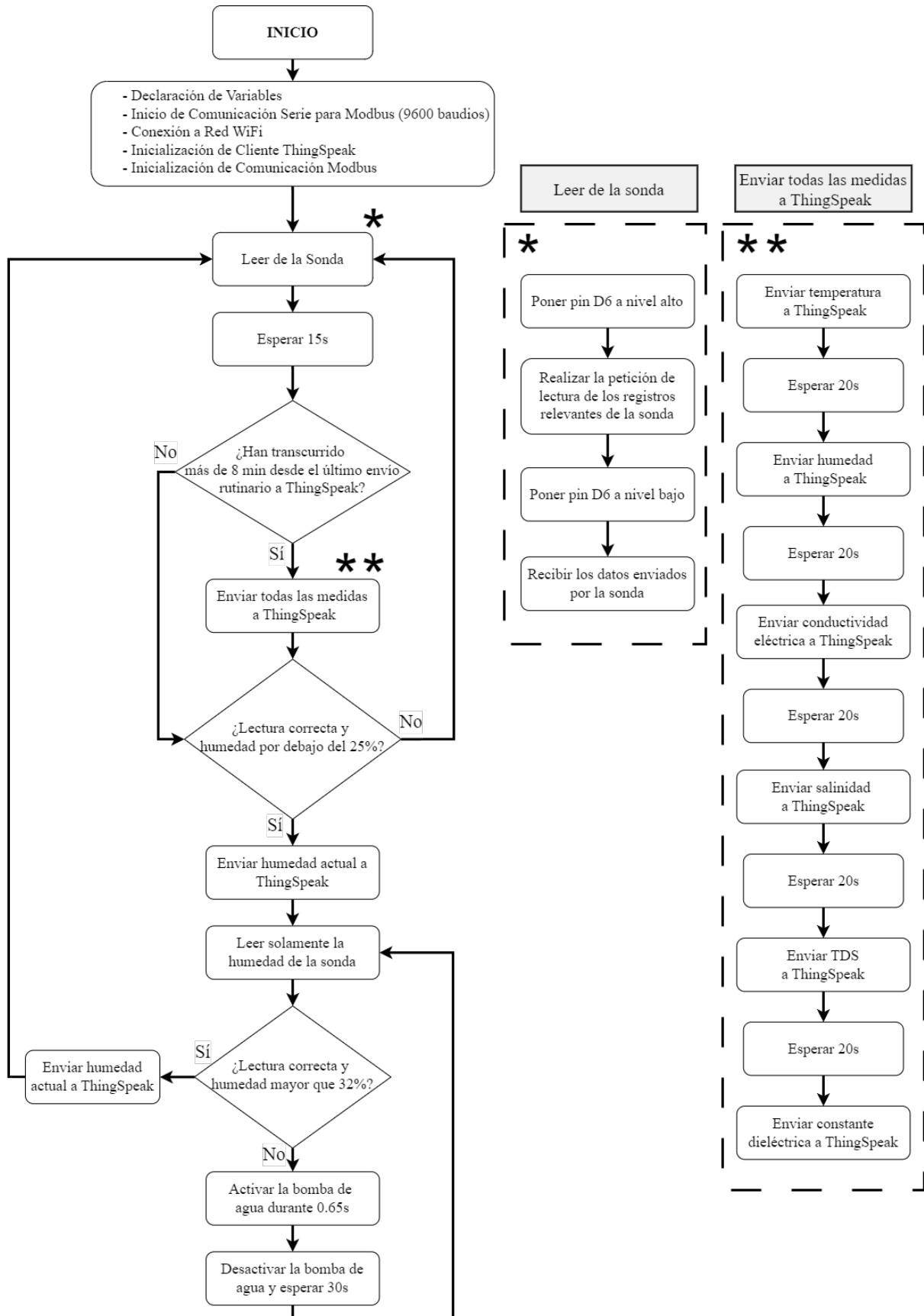


Figura 3.11 Diagrama de flujo que muestra el control del sistema realizado en este experimento.

4 Resultados Experimentales

Tras exponer tanto el conexionado como la lógica de control programada en ambos experimentos, se procederá a explicar las maneras en las que se han recogido los datos para su análisis y representación y, además, se discutirán los resultados en base a estas representaciones.

4.1 Medios para la Representación de Datos

Como ya se ha comentado en el capítulo anterior, se ha usado la plataforma *ThingSpeak* para el guardado de datos en la nube. En la misma página web de esta plataforma cualquier usuario puede observar los datos recogidos hasta el momento, incluso con algunas opciones de visualización, sin embargo, estas opciones son muy limitadas, por lo cual es conveniente exportar estos datos para poder representarlos de forma personalizada.

ThingSpeak ofrece distintas opciones para realizar la exportación de datos, en formato JSON, XML o CSV. Estas maneras pueden resultar útiles, sin embargo, para recoger los datos de los experimentos expuestos se ha utilizado una función del entorno Matlab que convierte directamente los datos en variables tipo tabla, facilitando su manipulación. En concreto, se ha usado la función *thingSpeakRead*, que tiene la siguiente estructura:

```
[datos, tiempo, channelInfo]=thingSpeakRead(ChannelID, 'Fields', [], ReadKey='', DateRange=[]);
```

En *ChannelID* se debe indicar el código identificador del canal del que se quiera leer y tras *ReadKey* se debe indicar la clave de lectura (en caso de que sea un canal privado). Después de *Fields* se deben indicar los campos que se quieren leer dentro del canal y en *DateRange* se deben indicar las fechas de inicio y final (en formato *datetime*) del intervalo de tiempo deseado.

La función devuelve los datos que se enviaron en el intervalo de tiempo especificado, una estructura tipo *datetime* que contiene las fechas en las que se envió cada dato y el *channelInfo*, que no es más que una estructura que contiene datos como: el código de identificación del canal, la fecha de creación del canal, su URL, la última vez que fue actualizado, entre otros.

4.2 Resultados del Experimento Inicial

La duración de este experimento inicial fue de, aproximadamente, un par de semanas, debido a que se decidió que era más interesante realizar el experimento avanzado durante más tiempo. Antes de discutir los resultados, se va a mostrar el código utilizado para representarlos, además de una gráfica que los exhibe. En este código simplemente se utiliza la función antes explicada para recoger los datos del canal de *ThingSpeak* donde se encuentran las medidas de la sonda, que luego se muestran en una gráfica utilizando unas opciones personalizadas de representación.

Código 4.1 Código para recoger datos de la sonda Capacitive Soil Moisture Sensor v2.0 y representarlos en una gráfica de Matlab.

```

%% Recoger datos de sonda Capacitive Soil Moisture Sensor v2.0
% Limpiar todas las variables

clear all

% Fecha de inicio y final de recolección de datos

startDate=datetime(2024,2,19,18,14,30);
endDate=datetime(2024,3,5,17,38,09);

% Lectura de todos los datos del canal de ThingSpeak

[sonda,tiempo,channelInfo]=thingSpeakRead(2393548,...
    DateRange=[startDate,datetime("now")],...
    ReadKey='YNQ14IEHFBQMF7HG');

%Representación de los datos del cuarto campo del canal, que es el que
%contiene los datos de humedad del sustrato de esta sonda

plot(tiempo,sonda(:,4),'LineWidth',1.75);

%Opciones de representación: título de la figura y descripción de los ejes,
% además de elección de la fuente y el tamaño de esta

title("Humedad del sustrato respecto al tiempo",'FontName','Times','FontSize',
    ,24);
xlabel("Fecha",'FontName','Times','FontSize',32);
ylabel("Humedad [%]","FontName','Times','FontSize',32);
ax = gca();
%Configuración de los límites y marcas del eje y (marca cada punto
%porcentual)

yticks(55:1:75)
ax.YAxis.FontSize = 14;
ylim([min(sonda(:,4)*0.98) max(sonda(:,4)*1.02)])

%Configuración de los límites y marcas del eje x (una marca diaria)

ax.XTick = datetime(2024,2,20,00,00,00):caldays(1):datetime(2024,3,5,00,00,00);
ax.XAxis.FontSize = 14;
xlim([startDate,endDate])

```

La gráfica que se genera tras la ejecución del código se puede observar en la Figura 4.1, en ella se muestran los datos de humedad del sustrato de la planta a lo largo de, aproximadamente, dieciséis días, y habiendo realizado dos ciclos de irrigación satisfactorios. Se pueden comentar algunos aspectos de esta gráfica:

- Aunque, como se indicó en el capítulo anterior, la irrigación se detiene cuando se alcanza un 65% de humedad, tras la acción de riego se puede observar que esta magnitud sube hasta, aproximadamente, un 72%. Esto se debe a la absorción de agua por parte del sustrato, como esta hace que suba el porcentaje de humedad a lo largo de varias horas, se ha optado por tener en cuenta este "offset" a la hora de regar. Este comportamiento se identificó en las pruebas anteriores a la implementación del sistema de irrigación, ya que ofrecía los mismos resultados cuando se regaba el sustrato de forma manual.

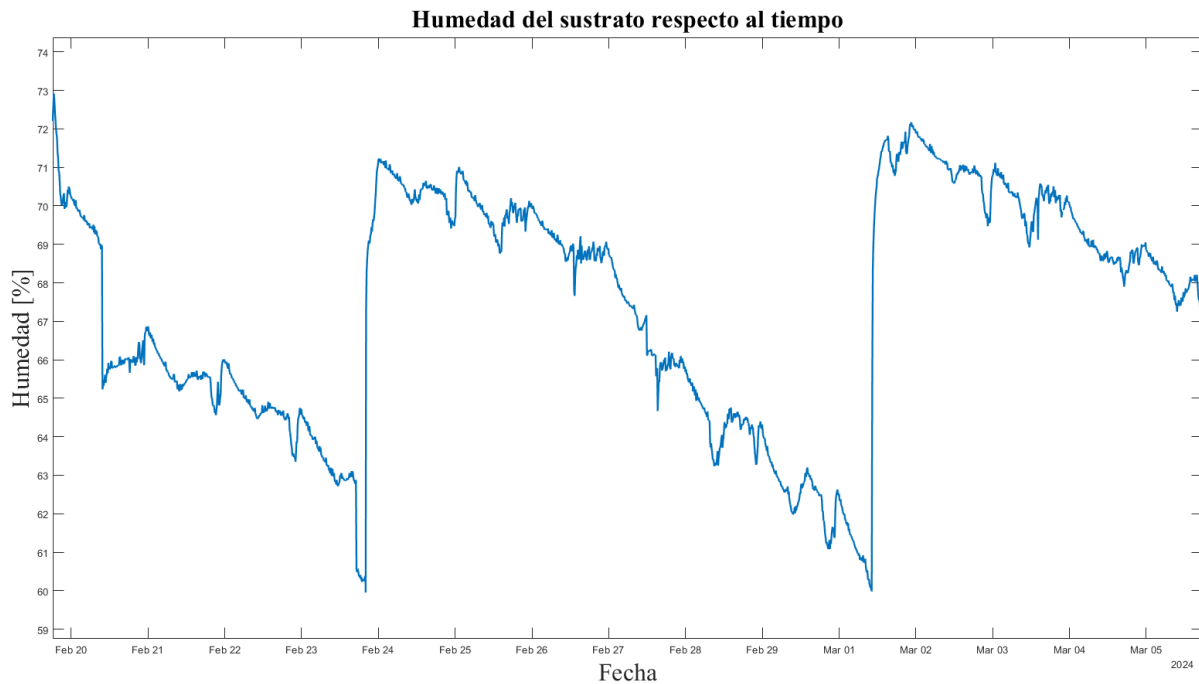


Figura 4.1 Gráfica que muestra los datos de humedad del sustrato tomados a lo largo del experimento inicial.

- Se puede discernir una especie de ruido en la medida, esto puede deberse a cambios en la humedad de los alrededores de dónde estaba colocado el experimento. Esta sonda, como ya se ha comentado, no puede enterrarse en el sustrato terrestre en su totalidad debido a que es necesario proteger sus conexiones eléctricas, por tanto, una parte de material capacitivo está expuesta directamente al aire. Al no estar enteramente dentro de la tierra, la humedad del ambiente tiene una influencia mayor en la medida, provocando el ruido observado (aunque se calibrara la sonda en un principio, la humedad del ambiente ha ido cambiando a lo largo del experimento).
- También se pueden observar un par de descensos bruscos de humedad antes del primer riego, la explicación posible de estos es que, debido a las pequeñas vibraciones que se pudieran haber dado en el lugar del experimento, la sonda se haya movido ligeramente, permitiendo así que más superficie de los electrodos estuviera expuesta al aire seco, bajando considerablemente la humedad medida. Esta hipótesis tiene más sentido si se considera lo expuesto en el párrafo anterior, al estar la sonda un poco desenterrada, es más probable que se dé esta situación.

4.3 Resultados del Experimento Avanzado

La duración de este experimento se ha prolongado durante un mes, aproximadamente, ya que, en este caso, se podían obtener más datos de relevancia de la planta. Aunque en el código para recoger los resultados se han utilizado las mismas funciones, se ha necesitado crear un algoritmo más complejo debido a que la función *thingSpeakRead* tiene un límite de lectura de 8000 datos. En particular, obtiene los 8000 últimos datos, por tanto, es necesario realizar varias lecturas, se ha optado por un bucle que contiene a esta función, cada vez que se lee, se toma en la siguiente iteración la fecha del dato más antiguo obtenido hasta el momento como fecha final de lectura, y se van organizando todos los datos en un vector total hasta que la fecha final de lectura sea igual a la fecha inicial de lectura dada por el usuario.

Cómo se explicó en el capítulo anterior, los diferentes datos que recoge la sonda se mandan con un tiempo aproximado entre envíos de 20 segundos, esto significa que en todos los instantes de tiempo sólo existen datos para una magnitud y todas las demás quedan como *Not a Number* o NaN, esto genera problemas a la hora de representar los datos, teniendo que reorganizar la matriz total que los contiene. Para ello, hay que tener en cuenta que el vector de tiempo sí que tiene una sola columna, conteniendo los instantes de tiempo en los que se haya enviado cualquier dato.

	1	2	3	4	5	6			1
1	22.5300	NaN	NaN	NaN	NaN	NaN		1	18-Mar-2024 00:02:04
2	NaN	26.1900	NaN	NaN	NaN	NaN		2	18-Mar-2024 00:02:24
3	NaN	NaN	76	NaN	NaN	NaN		3	18-Mar-2024 00:02:45
4	NaN	NaN	NaN	42	NaN	NaN		4	18-Mar-2024 00:03:05
5	NaN	NaN	NaN	NaN	38	NaN		5	18-Mar-2024 00:03:26
6	NaN	NaN	NaN	NaN	NaN	14.1300		6	18-Mar-2024 00:03:47
7	22.5300	NaN	NaN	NaN	NaN	NaN		7	18-Mar-2024 00:10:05
8	NaN	26.1900	NaN	NaN	NaN	NaN		8	18-Mar-2024 00:10:25
9	NaN	NaN	76	NaN	NaN	NaN		9	18-Mar-2024 00:10:46
10	NaN	NaN	NaN	42	NaN	NaN		10	18-Mar-2024 00:11:06
11	NaN	NaN	NaN	NaN	38	NaN		11	18-Mar-2024 00:11:27
12	NaN	NaN	NaN	NaN	NaN	14.1300		12	18-Mar-2024 00:11:47
13	22.4800	NaN	NaN	NaN	NaN	NaN		13	18-Mar-2024 00:18:05
14	NaN	26.1900	NaN	NaN	NaN	NaN		14	18-Mar-2024 00:18:25
15	NaN	NaN	76	NaN	NaN	NaN		15	18-Mar-2024 00:18:46
16	NaN	NaN	NaN	42	NaN	NaN		16	18-Mar-2024 00:19:06
17	NaN	NaN	NaN	NaN	38	NaN		17	18-Mar-2024 00:19:27
18	NaN	NaN	NaN	NaN	NaN	14.1300		18	18-Mar-2024 00:19:47
19	22.4600	NaN	NaN	NaN	NaN	NaN		19	18-Mar-2024 00:26:05
20	NaN	26.0700	NaN	NaN	NaN	NaN		20	18-Mar-2024 00:26:26
21	NaN	NaN	76	NaN	NaN	NaN		21	18-Mar-2024 00:26:46
22	NaN	NaN	NaN	42	NaN	NaN		22	18-Mar-2024 00:27:07
23	NaN	NaN	NaN	NaN	38	NaN		23	18-Mar-2024 00:27:27
24	NaN	NaN	NaN	NaN	NaN	14.0600		24	18-Mar-2024 00:27:48
25	22.4300	NaN	NaN	NaN	NaN	NaN		25	18-Mar-2024 00:34:05
26	NaN	26.2500	NaN	NaN	NaN	NaN		26	18-Mar-2024 00:34:26

Figura 4.2 A la izquierda, ejemplo de matriz de datos recogidos de *ThingSpeak* sin organizar, a la derecha, ejemplo de vector de tiempos, en formato *datetime*.

Teniendo en cuenta estos formatos, se han creado dos matrices, una que contenga los datos de todas las variables sin datos tipo NaN y otra que contenga los tiempos en los que se enviaron cada una de estas variables. Para rellenarlas se han creado dos bucles, uno que aumenta un índice de 1 a 6, representando las distintas variables recogidas y otro anidado en el anterior, que recorre todas las filas del vector total de datos, si el dato no es tipo NaN (se utiliza la función *isnan* de Matlab para comprobar esto), se guarda el dato en la matriz de datos total y también su tiempo correspondiente. Las demás funciones del código sí que son similares a las del código utilizado en el experimento anterior:

Código 4.2 Código para recoger datos de la sonda MEC10 y representarlos en una gráfica de Matlab.

```

%% Recoger datos de sonda MEC10
% Limpiar todas las variables

clear all

% Fecha de inicio y final de recolección de datos

startDate=datetime(2024,3,19,23,59,58);
endDate=datetime("now");

% Lectura de datos del canal de ThingSpeak

[sonda,tiempo,channelInfo]=thingSpeakRead(2434217,...
    DateRange=[startDate,endDate],...
    ReadKey='YB8YCS9DJELKCC36');

% Necesitamos hacer varias lecturas, definimos estado inicial de los
% vectores de datos y tiempo totales

tiempototal=tiempo;
sondatotal=sonda;

% Hacemos un bucle dónde leemos los datos hasta que la primera fecha

```

```

% registrada sea la fecha de inicio

while(tiempo(1)~=startDate)

    [sonda,tiempo,channelInfo]=thingSpeakRead(2434217,...
        DateRange=[startDate,tiempo(1)],...
        ReadKey='YB8YCS9DJELKCC36');

    %Actualizamos los vectores de datos totales recolectados

    tiempototal=[tiempo;
                tiempototal];
    sondatotal =[sonda;
                sondatotal];

end

%% Representar los datos recolectados

%Bucle para guardar cada una de las variables sin que haya ningún dato tipo
% "NaN"

for j=1:6
    index=0;
    for i=1:length(sondatotal)
        if(~isnan(sondatotal(i,j)))
            index=index+1;
            datos(index,j)=sondatotal(i,j);
            time(index,j)=tiempototal(i);
        end
    end
end

%Definición de nombres para titular las gráficas de forma más cómoda

nombres=["Contenido Volumétrico de Agua" "Contenido Volumétrico de Agua [%]";
        "Temperatura" "Temperatura [°C]";
        "Conductividad Eléctrica" "Conductividad Eléctrica [us/cm]";
        "Salinidad" "Salinidad [mg/L]";
        "Total de Sólidos Disueltos" "Total de Sólidos Disueltos [mg/L]";
        "Constante Dieléctrica" "Constante Dieléctrica"];

%Reestructuración de las matrices totales para facilitar la creación de gráficas

datos(:,7)=datos(:,2);
datos(:,2)=datos(:,1);
datos(:,1)=datos(:,7);

time(:,7)=time(:,2);
time(:,2)=time(:,1);
time(:,1)=time(:,7);

% Primera figura, hacemos un bucle que rellena una gráfica compuesta por
% varias subgráficas

figure(1)

```

```

for j=1:5

    %Número de subgráfica

    if(j==5)
        subplot(3,2,5.5);
    else
        subplot(3,2,j);
    end

    %Generamos la gráfica con todas las especificaciones personalizadas

    plot(time(:,j+1),datos(:,j+1),'LineWidth',1.75);
    title(nombres(j+1,1),'FontName','Times','FontSize',18);
    ylabel(nombres(j+1,2),'FontName','Times','FontSize',15);
    xlabel("Fecha",'FontName','Times','FontSize',18);

    %Límites del eje y

    ylim([min(nonzeros(datos(:,j+1)*0.98)) max(datos(:,j+1)*1.02)]);

    %Límites y marcas del eje x (marca cada dos días)

    ax = gca();
    ax.XTick = datetime(2024,3,18,00,00,00):caldays(2):datetime
        (2024,4,21,00,00,00);
    xlim([startDate,endDate])
end

% Segunda figura, sólo se representa una magnitud, por tanto, hacemos una
% gráfica sin subgráficas

figure(2)

%Generamos la gráfica con todas las especificaciones personalizadas

plot(time(:,1),datos(:,1),'LineWidth',1.75);
title(nombres(1,1),'FontName','Times','FontSize',24);
ylabel(nombres(1,2),'FontName','Times','FontSize',24);
xlabel("Fecha",'FontName','Times','FontSize',24);

%Límites del eje y

ylim([min(datos(:,1)*0.98) max(datos(:,1)*1.02)]);

%Límites y marcas del eje x (marca cada dos días)

ax = gca();
ax.XTick = datetime(2024,3,18,00,00,00):caldays(2):datetime
        (2024,4,21,00,00,00);
xlim([startDate,endDate])

```

Tras ejecutar este código, se generan las gráficas que se pueden observar en la Figura 4.3 y la Figura 4.4. Se ha decidido hacer una gráfica con sólo la variable de control del sistema (la humedad) y otra con las demás variables que mide la sonda:

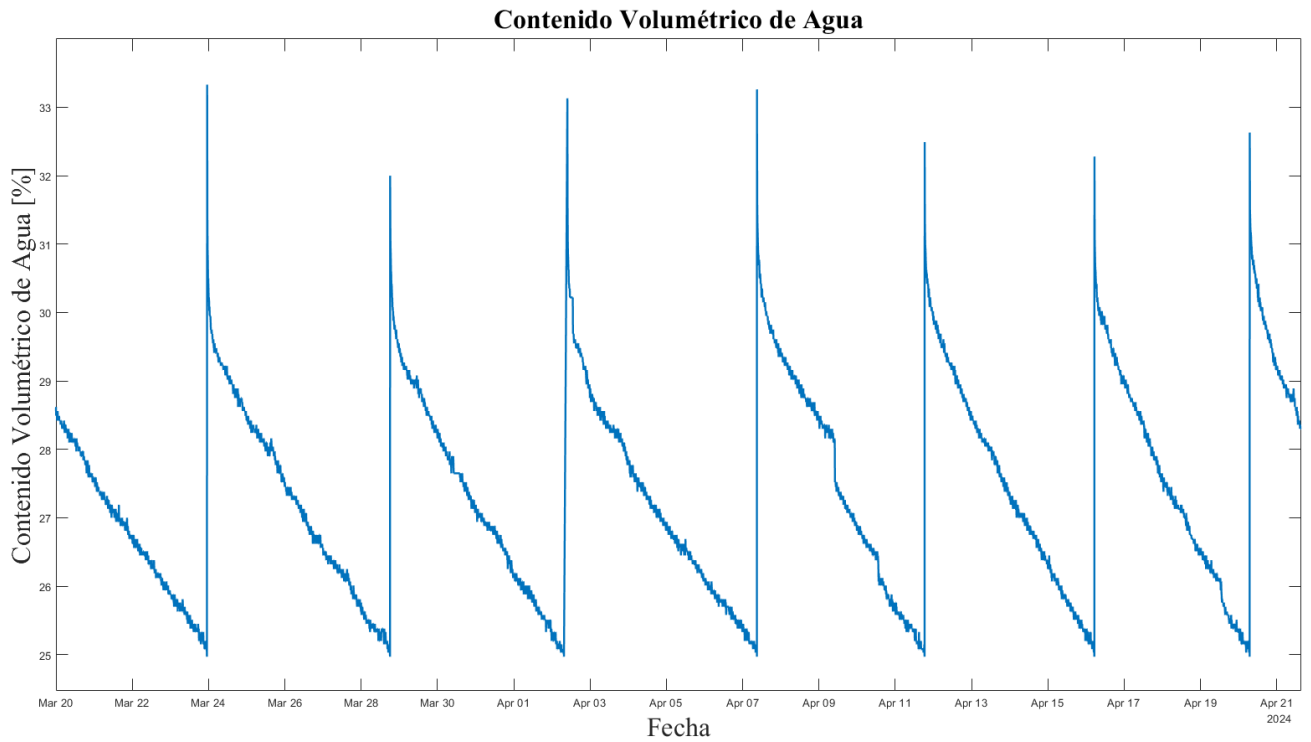


Figura 4.3 Humedad del sustrato a lo largo del experimento.

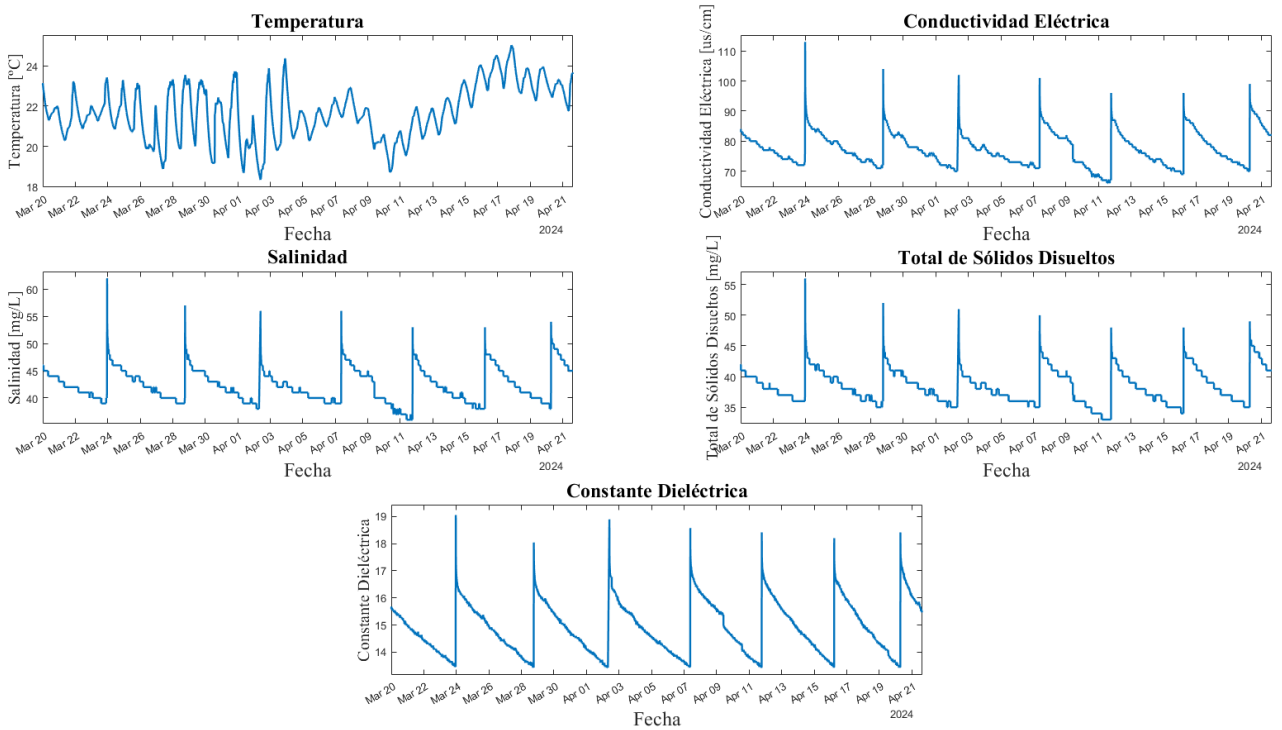


Figura 4.4 Temperatura, Conductividad Eléctrica, Salinidad, Total de Sólidos Disueltos y Constante Dieléctrica a lo largo del experimento.

Usando estas gráficas como referencia, se procederá a comentar algunos aspectos relevantes sobre los resultados obtenidos:

- Al contrario que en el experimento anterior, se puede observar que, en las medidas de humedad, no se observa un ruido tan pronunciado. Esto se puede deber a que, como la sonda usada es lo suficientemente estanca como para poder clavarse en el sustrato completamente, la humedad del ambiente no genera una perturbación a la hora de tomar la medida.
- Se observa una evolución distinta de la humedad del sustrato respecto al experimento anterior, se puede ver que, tras el riego, la humedad disminuye de forma significativa un par de puntos porcentuales, este cambio de comportamiento entre experimentos se puede deber al cambio en el método de medición entre las dos sondas.
- La medida de temperatura refleja de forma aproximada la temperatura dada en el emplazamiento del experimento, pudiéndose identificar claramente la diferencia entre las nocturnas y diurnas.
- Observando las gráficas de la conductividad eléctrica, la salinidad y el total de sólidos disueltos, se puede comprobar que, como se ha comentado en capítulos anteriores, existe una fuerte relación de proporcionalidad entre ellas. Además, todas estas magnitudes están también fuertemente ligadas a la humedad del sustrato.

4.4 Otros Aspectos a Considerar

Una característica sobre la que no se ha hecho mucho hincapié hasta ahora es el precio de los materiales necesarios para realizar estos experimentos, este es un aspecto importante a considerar para cumplir las restricciones de precio dadas en los sistemas IoT. Por ello, aquí se va a exponer el precio aproximado de cada componente por separado (en el momento en el que se está redactando esta sección), empezando por el experimento inicial:

Tabla 4.1 Precio aproximado de los componentes necesarios para la realización del experimento inicial.

Componentes	Precio Aproximado
Sonda <i>Soil Capacitive Moisture Sensor v2.0</i>	0.9-2€
Placa de Desarrollo D1 Mini	5-7€
Módulo de Relé SRD-05VDC-SL-C	1-4.5€
Bomba Submersible 3VDC Vertical Pump	2.95€
Convertidor ADS1115	3.8-6€
Precio Total	13.65-22.45€

Es importante aclarar que estos precios son los ofertados cuando se intenta adquirir cada uno de los componentes de forma individual, por tanto, si se requiriera producir una gran cantidad de sistemas de medida, seguramente el precio sería considerablemente menor. Además, también se han utilizado módulos que incorporan ciertos componentes, en vez de los componentes en sí, lo cual aumenta el precio, esto se podría solucionar en cierta medida produciendo una placa personalizada que incorpore todos los componentes. Sin embargo, se van a realizar comentarios sobre los precios dados en la Tabla 4.1, ante estos, se puede indicar que, claramente, no es muy asequible en el caso en el que se quiera implementar uno de estos sistemas en cada planta de una plantación de gran tamaño, por tanto, este sistema es más adecuado para plantaciones de menor tamaño (o, como en el experimento, de una única planta) o para plantaciones de un tamaño mayor pero implementando este sistema de sensores para varias plantas a la vez (es decir, suponiendo que un grupo de plantas tienen, de forma aproximada, las mismas necesidades de riego). A continuación, se exponen los precios aproximados de los componentes utilizados en el experimento avanzado:

Tabla 4.2 Precio aproximado de los componentes necesarios para la realización del experimento avanzado.

Componente	Precio Aproximado
Sonda <i>MEC10</i>	59-68€
Placa de Desarrollo D1 Mini	5-7€
Módulo de Relé SRD-05VDC-SL-C	1-4.5€
Bomba Submersible 3VDC Vertical Pump	2.95€
Módulo de Transceptor MAX485	0.3-2€
Precio Total	68.25-84.45€

En este caso, aunque se bajen los precios de producción mediante la creación de una placa personalizada o mediante la compra mayorista, claramente este sistema de sensores debe utilizarse o bien en plantaciones de pequeño tamaño o en plantaciones de tamaño mayor pero con un sistema para varias plantas. Incluso aunque se aprovecharan las cualidades de la capa física RS485 y se utilizaran varias sondas en un mismo sistema, no cambiaría mucho el precio medio por planta porque la sonda es el componente más costoso a la hora de implementar este sistema. Sin embargo, también es interesante resaltar la posible función que este sistema puede tener en cultivos que precisen de mayores cuidados, ya que, como se obtienen muchas más variables relevantes del sustrato, se puede realizar un seguimiento más preciso, necesario para cumplir requisitos de crecimiento de algunas especies de plantas.

Por último, un resultado sobre el que tampoco se ha insistido mucho es el estado de salud de la planta al implementar estos sistemas de irrigación. Sin embargo, aunque realizar ensayos rigurosos de la salud de la planta se sale del enfoque y los objetivos de este trabajo, se puede, mediante la observación a simple vista, determinar si el estado de salud de la planta ha mejorado. En este caso, cómo se puede ver en la Figura 4.5, aunque no se dispongan de imágenes antes de la implementación del sistema de riego, se observa que han nacido nuevos brotes de las ramas de la planta, pudiendo este ser un indicativo de que, con los sistemas de irrigación implementados, se han dado las condiciones necesarias para el crecimiento saludable de la planta.

**Figura 4.5** Detalle de las ramas de la planta cuyo sustrato se ha estado midiendo a lo largo de estos experimentos.

4.5 Diseño de Placas Personalizadas

Como se ha comentado en el apartado anterior, la creación de placas personalizadas podría disminuir de forma significativa el coste de implementación de los experimentos expuestos (al menos, el del experimento inicial), por tanto, se ha decidido realizar el diseño de estas placas para comprobarlo.

Estos diseños se han realizado en la plataforma online *EasyEDA*, la cual permite tanto la realización de esquemáticos como de las Placas de Circuito Impreso, *Printed Circuit Boards* o PCBs, dando también la opción de visualizar estos circuitos en tres dimensiones. Además, si se escogen componentes dentro de la base de datos de la plataforma (que coinciden con componentes ofertados por distintos vendedores), esta te permite realizar automáticamente una Factura de Materiales, *Bill Of Materials* o BOM, que no es más que una lista de todos los componentes usados en el esquemático, con su precio correspondiente. En los esquemáticos se han integrado los siguientes componentes:

- La placa de desarrollo *Wemos D1 Mini* desde un esquemático dado por la propia empresa *Wemos* [131], aunque este esquemático construye la placa desde el propio microcontrolador ESP8266, se ha decidido realizar el diseño a partir del módulo ESP-12F, que ya implementa, entre otros, un módulo Wi-Fi. Por lo tanto, a este módulo se le han tenido que añadir: la interfaz con el USB, el módulo CH340C para realizar las comunicaciones por USB y un regulador de tensión, entre otros.
- El módulo del relé *SRD-05VDC-SL-C*, que consiste principalmente en el propio relé, al que se le añade un transistor para controlar la alimentación y un Diodo Emisor de Luz, *Light Emitting Diode* o LED que indica si el relé está dando alimentación.
- En el caso del experimento inicial, se ha implementado también el convertidor analógico digital *ADS1115*, de tal manera que sólo se han conectado los pines de este a unos pines al alcance del usuario.
- En el caso del experimento avanzado, se ha implementado también el módulo *MAX485*, que no es más que el transceptor *MAX485*, algunas resistencias de *pull-up* y un LED que indica el estado de funcionamiento.

Se pueden observar en la Figura 4.6 y la Figura 4.7 los esquemáticos de la placa del experimento inicial y avanzado, respectivamente.

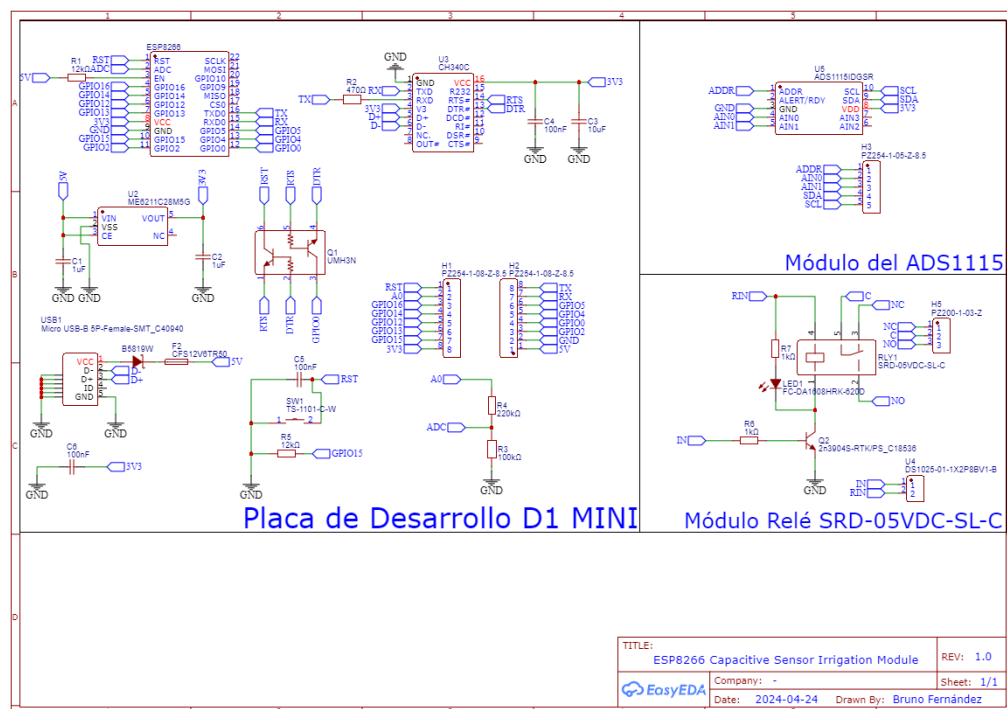


Figura 4.6 Esquemático diseñado para la creación de una placa personalizada para el experimento inicial.

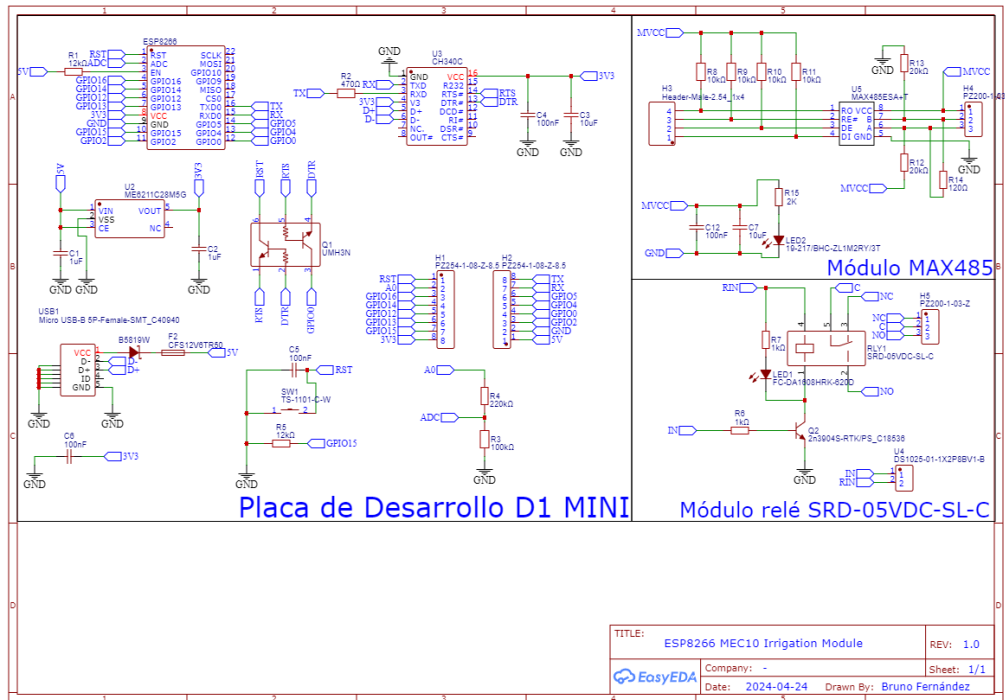


Figura 4.7 Esquemático diseñado para la creación de una placa personalizada para el experimento avanzado.

Como se ha mencionado antes, *EasyEDA* da la opción de visualizar las PCBs en tres dimensiones, en la Figura 4.8 y Figura 4.9 se pueden observar tanto el anverso como el reverso de estas representaciones en tres dimensiones.

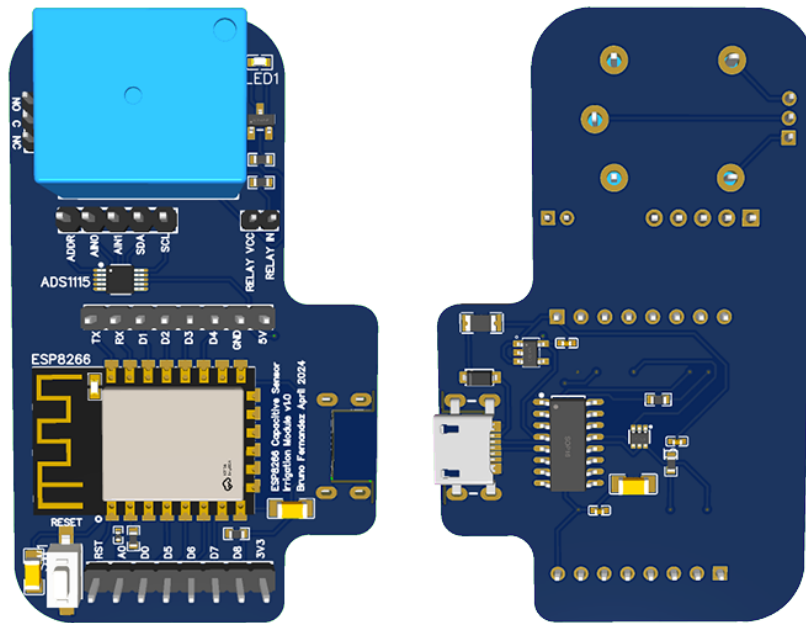


Figura 4.8 Visualización en tres dimensiones de la placa personalizada creada para el experimento inicial.

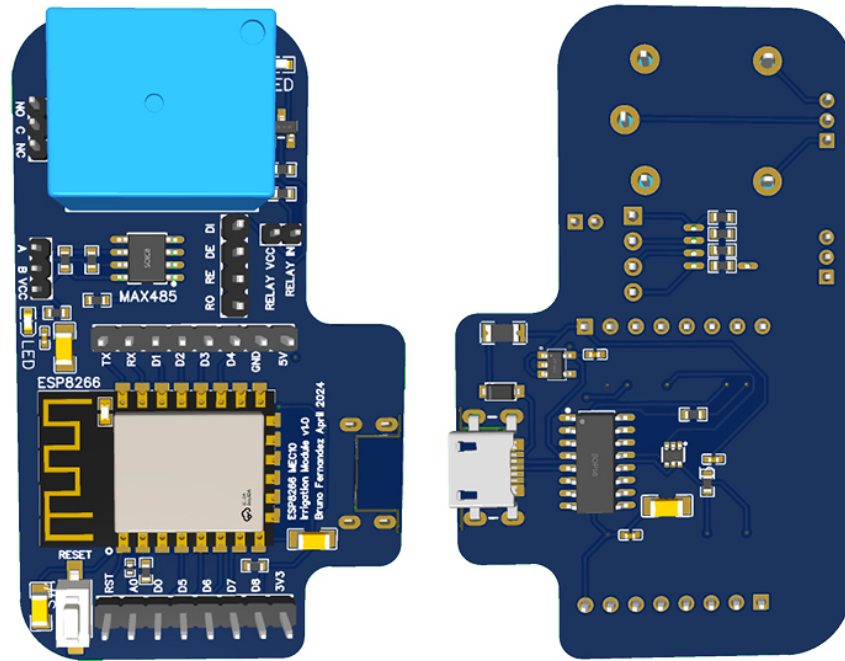


Figura 4.9 Visualización en tres dimensiones de la placa personalizada creada para el experimento avanzado.

Por último, se mostrarán mediante tablas los precios actualizados de los experimentos a la hora de utilizar estas placas personalizadas. Cabe destacar que estas placas no se han llegado a fabricar, sino que sólo se han diseñado, por tanto, es posible que, de nuevo, los precios de estas placas no sean totalmente correctos, ya que no se tienen en cuenta factores como el volumen de fabricación o los gastos de transporte. Para el experimento inicial los precios quedan de la siguiente manera:

Tabla 4.3 Precio aproximado de los componentes necesarios para la realización del experimento inicial utilizando una placa personalizada.

Componentes	Precio Aproximado
Sonda <i>Soil Capacitive Moisture Sensor v2.0</i>	0.9-2€
Bomba Submersible 3VDC Vertical Pump	2.95€
Placa Personalizada	4.6€
Precio Total	8.45-9.55€

Como se puede observar, los límites del rango de precios son significativamente menores a los calculados sin usar una placa personalizada. Si a esto se le añade la opción de obtener todos estos componentes en grandes cantidades, disminuyendo aún más el precio por unidad de estos, su implementación podría ser más asequible incluso a la hora de utilizar uno de estos sistemas de sensores por planta (o por cada par de plantas).

Tabla 4.4 Precio aproximado de los componentes necesarios para la realización del experimento avanzado utilizando una placa personalizada.

Componente	Precio Aproximado
Sonda <i>MEC10</i>	59-68€
Bomba Submersible 3VDC Vertical Pump	2.95€
Placa Personalizada	4.1€
Precio Total	66-75€

Sin embargo, cómo se puede observar en la Tabla 4.4, la introducción de una placa personalizada en el experimento avanzado es verdad que disminuye el precio, pero no en gran medida, esto es debido a que, como se comentó anteriormente, el producto de mayor valor de este sistema de riego es la propia sonda. Esto significa que, aunque se introduzca una placa personalizada, las opciones de implementación de este sistema no diferirán mucho de las indicadas en apartados anteriores.

5 Conclusiones

En este trabajo, en primer lugar, se ha realizado una revisión de la literatura disponible, investigando las diferentes tecnologías que consiguen implementar los sistemas IoT y las aplicaciones en las que pueden realizar una mejora considerable. Tras realizar esta investigación, queda claro que estos sistemas no sólo han tenido un gran recorrido y crecimiento en los últimos años, sino que también los proyectos y conceptos de aplicación de este tipo de tecnologías crece día a día, siendo un campo que seguramente alcanzará un gran protagonismo de forma transversal, desde su uso en plantas industriales y agrícolas hasta su instalación en los hogares.

La mayor ventaja de que el uso de estas tecnologías haya crecido y continúe creciendo hasta tal punto es que existe una comunidad de usuarios a escala internacional que facilita recursos de distintos tipos, desde microcontroladores, sensores o plataformas orientadas e intuitivas para las aplicaciones IoT hasta ejemplos de código y tutoriales explicatorios. De hecho, la realización de este trabajo se ha simplificado en gran medida gracias a la cantidad de recursos que se pueden encontrar y consultar, lo cual es un claro indicativo de que el entorno al que pertenece este tipo de tecnologías favorece la creación de nuevos proyectos y experimentos mediante este tipo de facilidades.

Posteriormente, se han diseñado e implementado dos sistemas de riego distintos, ambos se han basado en la humedad del sustrato de una planta para determinar cuando se activaba la irrigación, pudiendo así realizarse un riego automático que mantenía la humedad del sustrato entre unos umbrales predeterminados para el correcto desarrollo de la planta. Aunque el experimento inicial sólo realizaba mediciones de la humedad del sustrato, el experimento avanzado realizaba además de estas otras mediciones de diferentes variables relevantes del terreno.

Estas mediciones se han subido a la nube, dando la oportunidad de recolectar los datos de las variables medidas a lo largo del tiempo, pudiendo generar gráficas que muestra la evolución del sustrato terrestre de la planta estudiada en los experimentos realizados. Aunque el experimento avanzado ha realizado mediciones de más variables, es interesante comparar las gráficas que recojen los datos de humedad medidos por ambos experimentos.

Como se puede observar en la Figura 5.1, aunque existen ciertas diferencias entre el desempeño de estos sistemas, el objetivo principal, que es el de mantener los niveles de humedad entre unos límites determinados por el usuario, se ha cumplido en ambos de manera satisfactoria. Los resultados del experimento avanzado podrían considerarse de mayor calidad por las siguientes razones:

- Un menor ruido en las medidas tomadas, debido a que la humedad del ambiente no afecta a estas, al contrario que en el experimento inicial.
- Una mayor precisión y una mayor fiabilidad a la hora de mantener la humedad dentro del umbral especificado en el código de control.
- Una menor sensibilidad ante las perturbaciones de la medida, mostrando una mayor resistencia a las vibraciones que en el sistema del experimento inicial.

Sin embargo, atendiendo a los precios de implementación de estos dos sistemas expuestos en la Tabla 5.1, se puede observar que, incluso teniendo en cuenta el uso de placas personalizadas, el coste monetario del experimento avanzado aumenta hasta en ocho veces el del experimento inicial. Tras exponer esto, se puede llegar a la conclusión de que, como el sistema de sensores del experimento inicial ha demostrado un desempeño similar y ha cumplido el mismo objetivo que el sistema de sensores del experimento avanzado, el uso de este último está limitado, debido a su elevado coste, a cultivos de precisión que necesiten un seguimiento más preciso no sólo de la humedad sino también de otras variables relevantes del sustrato.

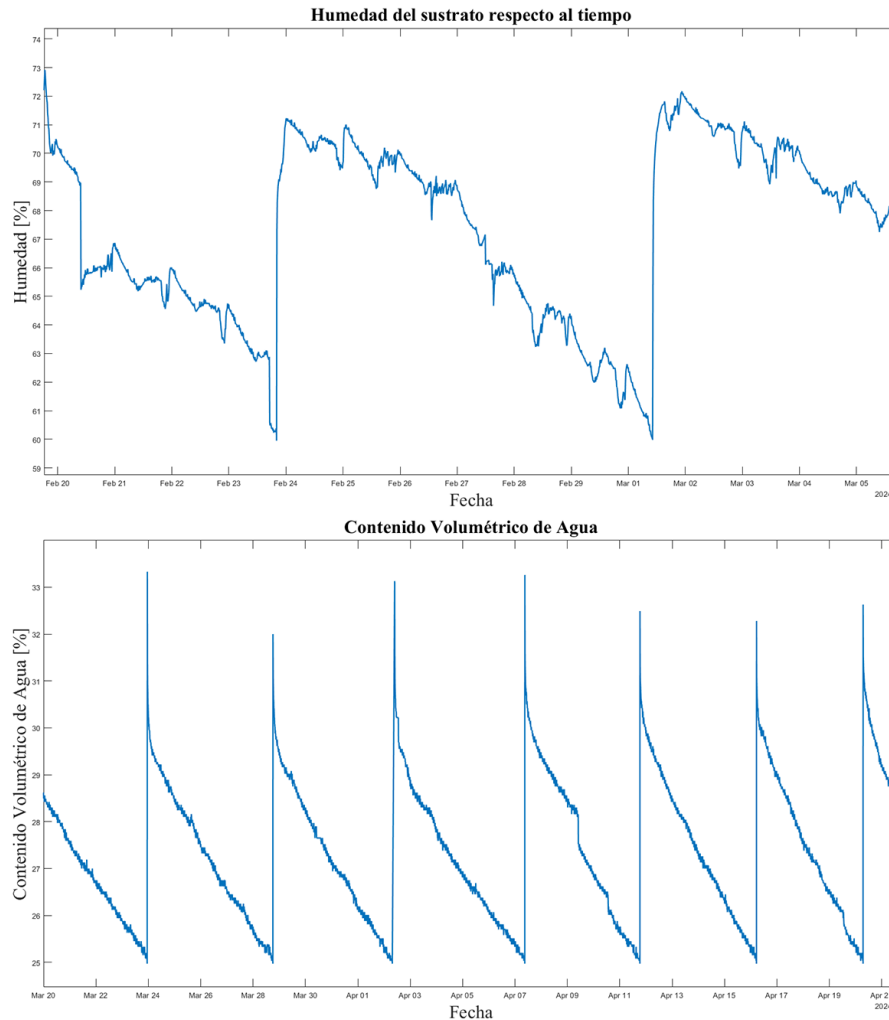


Figura 5.1 Arriba, datos de humedad recogidos durante el experimento inicial, abajo, datos de humedad recogidos durante el experimento avanzado.

Tabla 5.1 Comparación de precios de implementación de los experimentos realizados.

Tipo de Experimento	Precio Aproximado
Inicial	8.45-9.55€
Avanzado	66-75€

Además, aunque ambos sistemas de sensores tienen la capacidad de incorporar varias sondas en un mismo dispositivo (sin realizar modificaciones o realizando modificaciones mínimas), debido a que el precio del sistema del experimento avanzado es tan elevado principalmente por el coste monetario de la sonda MEC10, es mucho más asequible adquirir varias sondas como las utilizadas en el experimento inicial. Esto vuelve a evidenciar que, en casos de uso general o a grandes escalas, es mucho más barato utilizar el sistema del experimento inicial por su mayor escalabilidad.

5.1 Líneas Futuras

Aunque las conclusiones que se han sacado en este trabajo se pueden considerar interesantes, siempre se puede investigar y ahondar más, sobre todo en este tipo de tecnologías y con la cantidad de recursos disponibles para los usuarios. A continuación, con idea de ofrecer un punto de partida a aquellos que decidan basar un nuevo proyecto parcial o totalmente en los contenidos de este trabajo, se ofrecen algunos conceptos para la innovación:

- Aunque en capítulos anteriores se ha mostrado de forma heurística que el estado de salud de la planta ha mejorado, sería interesante que esto se hiciera de forma metódica y que la monitorización de estas variables se tratara como un objetivo desde el principio de los experimentos. Esto se podría realizar, por ejemplo, mediante medidas como el grosor de las ramas tomadas mediante dendrómetros y mediante la inspección visual continua que identifique si el color de las hojas se encuentra en un rango saludable, incluso, si fuera necesario, utilizando técnicas de percepción automática.
- Se podría explorar la idea de crear un sistema de sensores que midiera las mismas variables que el sistema del experimento avanzado pero con un coste menor. Esto se podría conseguir implementando sensores de humedad, temperatura y conductividad eléctrica por separado, pudiendo realizar incluso técnicas de supresión de ruido en estas medidas, para después comparar los resultados a los obtenidos con la sonda MEC10.
- Se podrían mandar a construir los diseños de las placas personalizadas tanto para el experimento inicial como para el experimento avanzado, así se podría probar su funcionamiento y hacer una mejor aproximación del coste total de implementación.
- También se ha mencionado en varias ocasiones la relación de las tecnologías IoT con la IA, por tanto, se propone que se utilicen modelos de redes neuronales entrenados mediante los datos tomados del sustrato terrestre. Esto podría servir para predecir con precisión el comportamiento del terreno, incluso llegando al punto de no tener que utilizar sistemas de sensores en algunas partes de la plantación y basar las actuaciones del sistema de riego en las predicciones del modelo de redes neuronales.
- Se pueden explorar otros tipos de irrigación fuera del riego temporizado utilizado en estos experimentos, por ejemplo, utilizando sistemas de riego que permitan una actuación modulada (con técnicas como el PWM o con salidas analógicas que permitan aplicar una tensión suficiente) que consigan una mayor precisión a la hora de alcanzar el umbral de riego previsto en el código de control. Otra idea a aplicar en sistemas que contengan varias sondas y, por tanto, tomen datos de varios sustratos, es el riego mayoritario por voto mayoritario, lo cual serviría como un recurso para poder utilizar un sólo sistema de sensores para varias plantas.

Para finalizar, es necesario indicar que las líneas futuras expuestas anteriormente no son todas las posibles en absoluto, los sistemas que rodean a las tecnologías IoT tienen una gran cantidad de áreas de estudio que explorar e, incluso, descubrir. El avance en estas áreas servirán para que este tipo de tecnologías actúen con mayor precisión, robustez y resiliencia, y así poder resolver desafíos cada vez de una forma más eficaz.

Índice de Figuras

2.1	Topología tipo Estrella	6
2.2	Topología tipo Red en Malla	6
2.3	Topología Jerárquica o en Árbol	7
2.4	Batería de Botón o <i>Button-Cell Battery</i>	8
2.5	Espectrograma de los diferentes Factores de Dispersión [10]	9
2.6	Ejemplo de transmisión de datos con símbolos CSS [10]	10
2.7	Pasos para la demodulación de un símbolo CSS [11]	10
2.8	BER frente a SNR para todos los SF disponibles en LoRa [10]	11
2.9	Representación de la resistencia al ruido y a las interferencias de las señales que hacen uso de UNB [22]	12
2.10	Probabilidad de que se dé cada valor de PAPR frente al valor de PAPR cuando se usa SC-FDMA y OFDM de Acceso Múltiple o OFDMA (con modulación BPSK en ambos casos) [32]	13
2.11	Comparación del Ratio de Error en Paquetes, <i>Packet Error Rate</i> o PER frente al número de paquetes por minuto transmitidos, comparación entre LoRaWAN, SigFox y NB-IoT [29].	13
2.12	Representación de las diferentes escalas en las que funcionan las <i>Smart Grids</i> [40]	15
2.13	Distintas capas que componen las redes neuronales	18
2.14	Esquema de un controlador PID	20
2.15	Separación mediante discriminación por color de las regiones sanas e infectadas de una hoja, (a) foto inicial, (b) área sana de la hoja, (c) área infectada de la hoja [100]	21
3.1	Pinout de la placa de desarrollo D1 Mini [106]	23
3.2	A la izquierda, módulo que incorpora el relé "SRD-05VDC-SL-C" de la empresa Ningbo Songle Relay [115], a la derecha, bomba de agua "Submersible 3VDC Vertical Pump" de la empresa Adafruit [116]	25
3.3	Sonda <i>Capacitive Soil Moisture Sensor v2.0</i>	25
3.4	Esquema que muestra los distintos componentes del convertidor analógico-digital ADS1115 [120]	26
3.5	Esquema que muestra el conexionado del sistema de medida completo	27
3.6	Diagrama de flujo que muestra el control del sistema realizado en este experimento	28
3.7	Sonda <i>MEC10</i>	29
3.8	Tabla proporcionada por el fabricante que indica los registros a leer para obtener las distintas medidas tomadas por la sonda, además de los rangos y unidades de estas magnitudes [123]	30
3.9	Esquemático que muestra los componentes del transceptor MAX485, fabricado por la empresa Analog Devices [129]	31
3.10	Esquema que muestra el conexionado del sistema de medida completo	32
3.11	Diagrama de flujo que muestra el control del sistema realizado en este experimento	33
4.1	Gráfica que muestra los datos de humedad del sustrato tomados a lo largo del experimento inicial	37
4.2	A la izquierda, ejemplo de matriz de datos recogidos de <i>ThingSpeak</i> sin organizar, a la derecha, ejemplo de vector de tiempos, en formato <i>datetime</i>	38
4.3	Humedad del sustrato a lo largo del experimento	41
4.4	Temperatura, Conductividad Eléctrica, Salinidad, Total de Sólidos Disueltos y Constante Dieléctrica a lo largo del experimento	41
4.5	Detalle de las ramas de la planta cuyo sustrato se ha estado midiendo a lo largo de estos experimentos	43

4.6	Esquemático diseñado para la creación de una placa personalizada para el experimento inicial	44
4.7	Esquemático diseñado para la creación de una placa personalizada para el experimento avanzado	45
4.8	Visualización en tres dimensiones de la placa personalizada creada para el experimento inicial	45
4.9	Visualización en tres dimensiones de la placa personalizada creada para el experimento avanzado	46
5.1	Arriba, datos de humedad recogidos durante el experimento inicial, abajo, datos de humedad recogidos durante el experimento avanzado	50

Índice de Tablas

2.1	Comparación de Rango y Flujo de Datos	14
4.1	Precio aproximado de los componentes necesarios para la realización del experimento inicial	42
4.2	Precio aproximado de los componentes necesarios para la realización del experimento avanzado	43
4.3	Precio aproximado de los componentes necesarios para la realización del experimento inicial utilizando una placa personalizada	46
4.4	Precio aproximado de los componentes necesarios para la realización del experimento avanzado utilizando una placa personalizada	46
5.1	Comparación de precios de implementación de los experimentos realizados	50

Índice de Códigos

4.1	Código para recoger datos de la sonda Capacitive Soil Moisture Sensor v2.0 y representarlos en una gráfica de Matlab	36
4.2	Código para recoger datos de la sonda MEC10 y representarlos en una gráfica de Matlab	38

Bibliografía

- [1] S. Navulur, A. Sastry, and M. Prasad, “Agricultural Management through Wireless Sensors and Internet of Things,” *International Journal of Electrical and Computer Engineering*, vol. 7, pp. 3492–3499, 12 2017.
- [2] United Nations, “World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100,” <https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html>, accessed: March 2024.
- [3] Y. Liu, K.-F. Tong, X. Qiu, Y. Liu, and X. Ding, “Wireless Mesh Networks in IoT networks,” in *2017 International Workshop on Electromagnetics: Applications and Student Innovation Competition*, 2017, pp. 183–185.
- [4] B. Adryan, D. Obermaier, and P. Fremantle, *The Technical Foundations of IoT*. USA: Artech House, Inc., 2017.
- [5] S. Farahani, *ZigBee Wireless Networks and Transceivers*. USA: Newnes, 2008.
- [6] R. Heydon, *Bluetooth Low Energy: The Developer’s Handbook*. Prentice Hall, 2012.
- [7] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of Things (IoT) communication protocols: Review,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690.
- [8] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley Publishing, 2009.
- [9] L. Vangelista, “Frequency Shift Chirp Modulation: The LoRa Modulation,” *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, 2017.
- [10] S. Ghosly, “All About LoRa and LoRaWAN,” <https://www.sghosly.com>, accessed: February 2024.
- [11] P. Edward, M. El-Aasser, M. Ashour, and T. Elshabrawy, “Interleaved Chirp Spreading LoRa as a Parallel Network to Enhance LoRa Capacity,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3864–3874, 2021.
- [12] O. Afisiadis, M. Cotting, A. Burg, and A. Balatsoukas-Stimming, “On the Error Rate of the LoRa Modulation With Interference,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1292–1304, 2020.
- [13] J. Liando, A. Jg, A. Tengourtius, and M. Li, “Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study,” *ACM Transactions on Sensor Networks*, vol. 15, pp. 1–35, 02 2019.
- [14] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. F. Skarmeta, “Performance Evaluation of LoRa Considering Scenario Conditions,” *Sensors*, vol. 18, no. 3, 2018.
- [15] “LoRa® and LoRaWAN®: A Technical Overview,” Semtech Corporation, Tech. Rep., 2019.

- [16] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, 2018.
- [17] T. Polonelli, D. Brunelli, and L. Benini, "Slotted ALOHA Overlay on LoRaWAN - A Distributed Synchronization Approach," in *2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2018, pp. 129–132.
- [18] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Boston, MA: Pearson, 2016.
- [19] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 197–202.
- [20] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [21] M. Anteur, V. Deslandes, N. Thomas, and A.-L. Beylot, "Ultra Narrow Band Technique for Low Power Wide Area Communications," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.
- [22] N. Naik, "LPWAN Technologies for IoT Systems: Choice Between Ultra Narrow Band and Spread Spectrum," in *2018 IEEE International Systems Engineering Symposium (ISSE)*, 2018, pp. 1–8.
- [23] B. Vejlggaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sorensen, "Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017, pp. 1–5.
- [24] R. Ratasuk, B. Vejlggaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, pp. 1–5.
- [25] R. Ratasuk, N. R. Mangalvedhe, Y. Zhang, M. Robert, and J.-P. Koskinen, "Overview of narrowband IoT in LTE Rel-13," *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–7, 2016.
- [26] R. P. Jover and I. Murynets, "Connection-less communication of IoT devices over LTE mobile networks," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2015, pp. 247–255.
- [27] K. Liu, G. Cui, Q. Li, S. Zhang, W. Wang, and X. Li, "An Optimal PSM Duration Calculation Algorithm For NB-IoT," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, 2019, pp. 447–452.
- [28] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow Band Internet of Things," *IEEE Access*, vol. 5, pp. 20 557–20 577, 2017.
- [29] H. Mroue, A. Nasser, S. Hamrioui, B. Parrein, E. Motta-Cruz, and G. Rouyer, "Mac layer-based evaluation of iot technologies: Lora, sigfox and nb-iot," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018, pp. 1–5.
- [30] H. G. Myung, "Introduction to single carrier FDMA," in *2007 15th European Signal Processing Conference*, 2007, pp. 2144–2148.
- [31] B. Benammar, N. Thomas, M.-L. Boucheret, C. Poulliat, and M. Dervin, "Analytical expressions of power spectral density for general spectrally shaped sc-fdma systems," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.
- [32] A. S. Shaikh and K. C. Kumar, "Performance Evaluation of LTE Physical Layer Using SC-FDMA & OFDMA," 2010.
- [33] Techplayon, "What is PAPR (Peak to average power ratio), Why it matters to Power Amplifier?" <https://www.techplayon.com/papr-peak-average-power-ratio-matters-power-amplifier/>, accessed: March 2024.

- [34] A. Ali and W. Hamouda, "On the Cell Search and Initial Synchronization for NB-IoT LTE Systems," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1843–1846, 2017.
- [35] RF Wireless World, "LTE NB-IoT Preamble And Reference Signals | NB-IoT NPSS, NSSS, NRS, DMRS," <https://www.rfwireless-world.com/Terminology/LTE-NB-IoT-Preamble-and-Reference-signals.html>, accessed: March 2024.
- [36] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A Primer on 3GPP Narrowband Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, 2017.
- [37] LoRa Alliance, "What is LoRaWAN Specification," <https://lora-alliance.org/about-lorawan/>, accessed: February 2024.
- [38] F. Michelinakis, A. S. Al-Selwi, M. Capuzzo, A. Zanella, K. Mahmood, and A. Elmokashfi, "Dissecting Energy Consumption of NB-IoT Devices Empirically," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1224–1242, 2021.
- [39] S. Pradeep, S. Krishna, M. S. Reddy, D. D. Sri, and M. S. Sri, "Analysis and Functioning of Smart Grid for Enhancing Energy Efficiency Using Optimization Techniques with IoT," in *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, 2023, pp. 316–321.
- [40] Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, "Internet of Things-Aided Smart Grid: Technologies, Architectures, Applications, Prototypes, and Future Research Directions," *IEEE Access*, pp. 62 962–63 003, 2019.
- [41] European Commission and Directorate-General for Energy and Alaton, C and Tounquet, F, *Benchmarking smart metering deployment in the EU-28 – Final report*. Publications Office, 2020.
- [42] L. Gomes, A. Coelho, and Z. Vale, "Assessment of Energy Customer Perception, Willingness, and Acceptance to Participate in Smart Grids—A Portuguese Survey," *Energies*, vol. 16, no. 1, 2023. [Online]. Available: <https://www.mdpi.com/1996-1073/16/1/270>
- [43] D. N. yin Mah, J. M. van der Vleuten, P. Hills, and J. Tao, "Consumer perceptions of smart grid development: Results of a Hong Kong survey and policy implications," *Energy Policy*, vol. 49, pp. 204–216, 2012, special Section: Fuel Poverty Comes of Age: Commemorating 21 Years of Research and Policy. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0301421512004685>
- [44] Z. Bi, L. D. Xu, and C. Wang, "Internet of Things for Enterprise Systems of Modern Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.
- [45] V. U. Ihekoronye, C. I. Nwakanma, G. O. Anyanwu, D.-S. Kim, and J.-M. Lee, "Benefits, Challenges and Practical Concerns of IoT for Smart Manufacturing," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, 2021, pp. 827–830.
- [46] Plex, "Plex, 7th Annual State of Smart Manufacturing Report," https://www.plex.com/sites/default/files/2022-05/10289_7_SOSM_Report.pdf?tcid=1211, 2022.
- [47] CESMII and SME, "CESMII & SME, 2022 Smart Manufacturing Sustainability Study," <https://www.cesmii.org/wp-content/uploads/2022-CESMII-SME-Smart-Manufacturing-Sustainability-Study-Report.pdf>, 2022.
- [48] Rockwell Automation, "Rockwell Automation, 8th 2023 State of Smart Manufacturing Report," https://www.rockwellautomation.com/content/dam/rockwell-automation/documents/pdf/campaigns/state-of-smart-2023/State-of-Smart-Manufacturing_RockwellAutomation_INFO-BR027B-EN-P.pdf, 2023.
- [49] N. N. Dlamini and K. Johnston, "The use, benefits and challenges of using the Internet of Things (IoT) in retail businesses: A literature review," in *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2016, pp. 430–436.

- [50] J. Xu, Z. Hu, Z. Zou, J. Zou, X. Hu, L. Liu, and L. Zheng, "Design of smart unstaffed retail shop based on iot and artificial intelligence," *IEEE Access*, vol. 8, pp. 147 728–147 737, 2020.
- [51] A. Esposito, "Analyzing The Mobile Consumer Part II: Personalization Vital To Customer Loyalty And Engagement," <https://www.retailtouchpoints.com/features/industry-insights/analyzing-the-mobile-consumer-part-ii-personalization-vital-to-customer-loyalty-and-engagement>, accessed: March 2024.
- [52] K. Damask, "Shoppers seeking online experience in the store," <https://www.digitalsignagetoday.com/articles/shoppers-seeking-online-experience-in-the-store/>, accessed: March 2024.
- [53] National Institute of Aging, "Loneliness and Social Isolation — Tips for Staying Connected," <https://www.nia.nih.gov/health/loneliness-and-social-isolation/loneliness-and-social-isolation-tips-staying-connected/>, accessed: March 2024.
- [54] F. Wu, T. Wu, and M. R. Yuce, "Design and Implementation of a Wearable Sensor Network System for IoT-Connected Safety and Health Applications," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 87–90.
- [55] M. R. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators A: Physical*, vol. 162, no. 1, pp. 116–129, 2010.
- [56] M. K. Kim, H. T. Jung, S. D. Kim, and H. J. La, "A Personal Health Index System with IoT Devices," in *2016 IEEE International Conference on Mobile Services (MS)*, 2016, pp. 174–177.
- [57] H. Cai, X. Liu, A. Jiang, R. Ni, X. Zhou, and A. Cangelosi, "Combination of EOG and EEG for emotion recognition over different window sizes," in *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*, 2021, pp. 1–6.
- [58] V. B. Shalini, "Smart health care monitoring system based on internet of things (iot)," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 1449–1453.
- [59] American Lung Association, "Pulse Oximetry," <https://www.lung.org/lung-health-diseases/lung-procedures-and-tests/pulse-oximetry#:~:text=The%20pulse%20oximeter%20uses%20a,in%20the%20red%20blood%20cell./>, accessed: March 2024.
- [60] Infinium, "How Do Pulse Oximeters Work? A Complete Guide," <https://infiniummedical.com/how-do-pulse-oximeters-work-a-complete-guide/>, accessed: March 2024.
- [61] Mayo Clinic, "Electroencefalografía (EEG)," <https://www.mayoclinic.org/es/tests-procedures/eeg/about/pac-20393875#:~:text=Un%20electroencefalograma%20puede%20detectar%20cambios,Tumores%20cerebrales/>, accessed: March 2024.
- [62] National Cancer Institute, "Definición de respuesta galvánica de la piel," <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/respuesta-galvanica-de-la-piel>, accessed: March 2024.
- [63] Brain Signs, "Respuesta galvánica de la piel (GSR)," <https://www.brainsigns.com/es/science/s2/technologies/gsr>, accessed: March 2024.
- [64] S. C. Mukhopadhyay, S. K. S. Tyagi, N. K. Suryadevara, V. Piuri, F. Scotti, and S. Zeadally, "Artificial intelligence-based sensors for next generation iot applications: A review," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 24 920–24 932, 2021.
- [65] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>
- [66] IBM, "¿Qué es Deep Learning?" <https://www.ibm.com/es-es/topics/deep-learning>, accessed: March 2024.
- [67] L. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.
- [68] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

- [69] MathWorks, “Foundations of Fuzzy Logic,” <https://in.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html>, accessed: March 2024.
- [70] United Nations, “Food Production Must Double by 2050 to Meet Demand from World’s Growing Population,” <https://press.un.org/en/2009/gaef3242.doc.htm>, accessed: March 2024.
- [71] S. Qazi, B. A. Khawaja, and Q. U. Farooq, “IoT-Equipped and AI-Enabled Next Generation Smart Agriculture: A Critical Review, Current Challenges and Future Trends,” *IEEE Access*, vol. 10, pp. 21 219–21 235, 2022.
- [72] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, “Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk,” *IEEE Access*, vol. 7, pp. 129 551–129 583, 2019.
- [73] M. Konain, R. Janapati, S. M. Ahmed, and M. M. Ahmed, “IoT based solar-powered mushroom farming for sustainable agriculture,” in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2023, pp. 944–948.
- [74] P. Palniladevi, T. Sabapathi, D. A. Kanth, and B. P. Kumar, “IoT based smart agriculture monitoring system using renewable energy sources,” in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, 2023, pp. 1–6.
- [75] J. L. Hatfield and J. H. Prueger, “Temperature extremes: Effect on plant growth and development,” *Weather and Climate Extremes*, vol. 10, pp. 4–10, 2015, USDA Research and Programs on Extreme Events.
- [76] K. Pramanik, J. Pradhan, P. Mohapatra, C. Jena, and L. Acharya, *Impact of Temperature: An Important Climate Changing Factor on Vegetable Crops*, 12 2020, pp. 105–134.
- [77] L. Gałęzewski, I. Jaskulska, D. Jaskulski, A. Lewandowski, A. Szyplowska, A. Wilczek, and M. Szczepańczyk, “Analysis of the need for soil moisture, salinity and temperature sensing in agriculture: a case study in Poland,” *Scientific Reports*, vol. 11, 08 2021.
- [78] S. Daryanto, L. Wang, and P.-A. Jacinthe, “Global Synthesis of Drought Effects on Maize and Wheat Production,” *PLOS ONE*, vol. 11, 05 2016.
- [79] M. Lamaoui, M. Jemo, R. Datla, and F. Bekkaoui, “Heat and Drought Stresses in Crops and Approaches for Their Mitigation,” *Frontiers in Chemistry*, vol. 6, p. 26, 02 2018.
- [80] S. Irmak and W. Rathje, “Plant Growth and Yield as Affected by Wet Soil Conditions Due to Flooding or Over-Irrigation,” University of Nebraska–Lincoln, Tech. Rep.
- [81] Canadian Government, “Managing Crops For Excess Water Stress,” <https://www.gov.mb.ca/agriculture/crops/crop-management/managing-crops-for-excess-water-stress.html>, accessed: March 2024.
- [82] R. M. A. Machado and R. P. Serralheiro, “Soil Salinity: Effect on Vegetable Crop Growth. Management Practices to Prevent and Mitigate Soil Salinization,” *Horticulturae*, vol. 3, no. 2, 2017.
- [83] HIBA, “Sensores de Conductividad Eléctrica del Suelo,” <https://learningdata.hubiberiaagrotech.eu/sensores-de-conductividad-electrica-del-suelo/>, accessed: March 2024.
- [84] A. Sheldon, N. Menzies, H.-B. So, and R. Dalal, “The effect of salinity on plant available water,” 01 2004.
- [85] W. Bai, L.-W. Kong, and A. Guo, “Effects of physical properties on electrical conductivity of compacted lateritic soil,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 5, p. 406–411, 10 2013.
- [86] M. Hardie and R. Doyle, “Measuring soil salinity,” *Methods in molecular biology (Clifton, N.J.)*, vol. 913, pp. 415–25, 08 2012.
- [87] T. Temesgen, “Systematic review: Effect of irrigation water quality and deficit irrigation on crop yield and water use efficiency,” *Turkish Journal of Agriculture: Food Science and Technology*, vol. 8, 06 2020.

- [88] F. Poyen Phd, P. Kundu, and A. Ghosh, “pH Control of Untreated Water for Irrigation,” *Journal of The Institution of Engineers (India): Series A*, vol. 99, pp. 1–8, 05 2018.
- [89] A. Taniguchi, L. Okumura, G. Spinelli, J. Silva, J. Uyeda, , and J. Sugano, “The Importance of Testing pH and EC in Agriculture,” University of Hawaii at Mānoa, 2019.
- [90] K. Rawat, S. Singh, and S. Gautam, “Assessment of groundwater quality for irrigation use: a peninsular case study,” *Applied Water Science*, vol. 8, 12 2018.
- [91] PennState Extension, “Interpreting Irrigation Water Tests,” <https://extension.psu.edu/interpreting-irrigation-water-tests>, accessed: March 2024.
- [92] A. Purohit and U. Gokhale, “Real Time Water Quality Measurement System based on GSM,” *IOSR Journal of Electronics and Communication Engineering*, vol. 9, pp. 63–67, 01 2014.
- [93] T.-h. Kim, V. S. Solanki, H. J. Baraiya, A. Mitra, H. Shah, and S. Roy, “A Smart, Sensible Agriculture System Using the Exponential Moving Average Model,” *Symmetry*, vol. 12, no. 3, 2020.
- [94] A. Winursito, I. Masngut, and G. N. P. Pratama, “Development and implementation of kalman filter for iot sensors: Towards a better precision agriculture,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2020, pp. 360–364.
- [95] X. He, Z. Xiong, and L. Zhang, “Design of incremental pid intelligent irrigation control system,” in *2023 4th International Conference on Mechatronics Technology and Intelligent Manufacturing (ICMTIM)*, 2023, pp. 45–49.
- [96] M. Bahat, G. Inbar, O. Yaniv, and M. Schneider, “A fuzzy irrigation controller system,” *Engineering Applications of Artificial Intelligence*, vol. 13, no. 2, pp. 137–145, 2000.
- [97] H. Benyezza, M. Bouhedda, and S. Rebouh, “Zoning irrigation smart system based on fuzzy control technology and iot for water and energy saving,” *Journal of Cleaner Production*, vol. 302, p. 127001, 2021.
- [98] P. Xuanrong, Y. Tingdong, and W. Yuesheng, “Research and design of precision irrigation system based on artificial neural network,” in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 3865–3870.
- [99] A. B., J. Pandi, M. V., S. Shalini, and S. R., “Design and Implementation of a Smart Solar Irrigation System Using IoT and Machine Learning,” *E3S Web of Conferences*, vol. 387, 05 2023.
- [100] H. Ali, M. Lali, M. Nawaz, M. Sharif, and B. Saleem, “Symptom based automated detection of citrus diseases using color histogram and textural descriptors,” *Computers and Electronics in Agriculture*, vol. 138, pp. 92–104, 2017.
- [101] M. Bhangе and H. Hingoliwala, “Smart Farming: Pomegranate Disease Detection Using Image Processing,” *Procedia Computer Science*, vol. 58, pp. 280–288, 2015, second International Symposium on Computer Vision and the Internet (VisionNet’15).
- [102] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, 2016.
- [103] H. Wang, G. Li, Z. Ma, and X. Li, “Application of neural networks to image recognition of plant diseases,” in *2012 International Conference on Systems and Informatics (ICSAI2012)*, 2012, pp. 2159–2164.
- [104] R. D. Devi, S. A. Nandhini, R. Hemalatha, and S. Radha, “IoT Enabled Efficient Detection and Classification of Plant Diseases for Agricultural Applications,” in *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, 2019, pp. 447–451.
- [105] A. Thorat, S. Kumari, and N. D. Valakunde, “An iot based smart solution for leaf disease detection,” in *2017 International Conference on Big Data, IoT and Data Science (BIG)*, 2017, pp. 193–198.
- [106] AV Electronics, “WeMos D1 mini ESP8266,” <https://avelectronics.cc/producto/wemos-d1-mini-esp8266/>, accessed: April 2024.

- [107] AZ-Delivery, “D1 Mini NodeMcu con módulo WLAN ESP8266-12F compatible con Arduino,” <https://www.az-delivery.de/es/products/d1-mini>, accessed: April 2024.
- [108] Naylamp mechatronics, “WEMOS D1 MINI ESP8266 WIFI,” <https://naylampmechatronics.com/espressif-esp/291-wemos-d1-mini-esp8266-wifi.html>, accessed: April 2024.
- [109] Pieter P, “A Beginner’s Guide to the ESP8266,” <https://tttpa.github.io/ESP8266/Chap01%20-%20ESP8266.html>, accessed: April 2024.
- [110] ESP8266 Community Forum, “Documentation for ESP8266 Arduino Core. Installation instructions, functions and classes reference.” <https://arduino.esp8266.com/Arduino/versions/2.1.0-rc2/doc/reference.html>, accessed: April 2024.
- [111] Scott Campbell, “BASICS OF THE SPI COMMUNICATION PROTOCOL,” <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>, accessed: April 2024.
- [112] V. M. Ayora, “ANÁLISIS Y CONTROL DE UN VEHÍCULO BASADO EN PÉNDULO INVERTIDO,” 2009.
- [113] ESP8266 Community, “Arduino core for ESP8266 WiFi chip,” <https://github.com/esp8266/Arduino>.
- [114] MathWorks, “ThingSpeak Communication Library for Arduino, ESP8266 and ESP32,” <https://github.com/mathworks/thingspeak-arduino>.
- [115] Ningbo Songle Relay, “SRD-05VDC-SL-C Datasheet,” <https://www.alldatasheet.es/datasheet-pdf/pdf/1132639/SONGLERELAY/SRD-05VDC-SL-C.html>.
- [116] Adafruit, “Submersible 3V DC Water Pump,” <https://www.adafruit.com/product/4547>.
- [117] ProBots, “How to Use Soil Moisture Sensor Capacitive V2.0 Module With Arduino,” <https://tutorials.probots.co.in/using-soil-moisture-sensor-capacitive-v2-0-module-for-arduino/#calibration>.
- [118] Electricity-Magnetism.org, “Sensores de humedad capacitivos,” <https://www.electricity-magnetism.org/es/sensores-de-humedad-capacitivos/>.
- [119] Last Minute Engineers, “Interfacing Capacitive Soil Moisture Sensor with Arduino,” <https://lastminuteengineers.com/capacitive-soil-moisture-sensor-arduino/>.
- [120] *ADS111x Ultra-Small, Low-Power, I2C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Comparator*, Texas Instruments, 5 2009.
- [121] Adafruit, “Adafruit_ADS1X15,” https://github.com/adafruit/Adafruit_ADS1X15.
- [122] —, “Adafruit_BusIO,” https://github.com/adafruit/Adafruit_BusIO.
- [123] *MEC10 Soil Moisture, EC and Temperature Sensor User Manual*, INFWIN, 4 2018.
- [124] Novus Automation, “CONCEPTOS FUNDAMENTALES DE RS485 Y RS422,” <https://cdn.novusautomation.com/downloads/conceptos%20fundamentales%20de%20rs485%20y%20rs422%20-%20espa%20sol.pdf>.
- [125] W&T, “Sistemas de bus RS485: Procedimientos de transmisión física, longitudes de cables, polaridad y mediciones,” <https://www.wut.de/e-6www-11-apes-000.php>.
- [126] Luis Llamas, “Comunicación RS485 con Arduino y conversor MAX485,” <https://www.luisllamas.es/arduino-rs485-max485/>.
- [127] Olga Weis, “Guía de comunicación Modbus RTU,” <https://www.virtual-serial-port.org/es/articles/modbus-rtu-guide/#4>.
- [128] Jason Kelly, “RS-485 Serial Interface Explained,” <https://www.cuidevices.com/blog/rs-485-serial-interface-explained#:~:text=At%20the%20data%20layer%20RS,serial%20communication%20in%20full%20duplex.>

- [129] *MAX481/MAX483/MAX485/MAX487–MAX491/MAX1487 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers*, Analog Devices, 9 2014.
- [130] Doc Walker, Ag Primatic and Marius Kintel, “ModbusMaster,” <https://github.com/4-20ma/ModbusMaster>.
- [131] *Wemos D1 Mini V3.0.0*, https://www.wemos.cc/en/latest/_static/files/sch_d1_mini_v3.0.0.pdf, Wemos, 1 2018.

Glosario

6LoWPAN IPv6 over Low power Wireless Personal Area Networks. 9, 14

BER Bit Error Rate. 10, 11, 53

BLE Bluetooth Low Energy. 7, 8, 14

BOM Bill Of Materials. 44

BPSK Binary Phase-Shift Keying. 11, 13, 53

CSS Chirp Spread Spectrum. 9, 10, 53

CSV Comma-Separated Values. 35

DE Driver Enable. 31

DFT Discrete Fourier Transform. 10, 12

DI Driver Input. 31

FSK Frequency Shift Keying. 9, 11

GPIO General Purpose Input/Output. 24, 31

HAN Home Area Network. 14

IA Inteligencia Artificial. 17, 51

IoT Internet of Things. 1, 2, 5, 7, 12, 14–18, 21–23, 26, 42, 49, 51, III, V

IP Internet Protocol. 9

I²C Inter-Integrated Circuit. 24, 26

JSON JavaScript Object Notation. 35

LED Light Emitting Diode. 44

LoRa Long Range. 9, 11, 12, 53

LoRaWAN Long Range Wide Area Network. 9, 11–14, 53

LTE Long Term Evolution. 12, 13

MISO Master Input Slave Output. 24

MOSI Master Output Slave Input. 24

- NAN** Neighborhood Area Network. 14
- NaN** Not a Number. 37, 38
- NB-IoT** Narrowband Internet of Things. 12–14, 53
- NO** Normally Open. 27
- NPSS** Narrowband Primary Synchronization Signal. 13
- NSSS** Narrowband Secondary Synchronization Signal. 13
-
- OFDM** Orthogonal Frequency-Division Multiplexing. 12, 13, 53
- OFDMA** Orthogonal Frequency-Division Multiple Access. 12, 13, 53
-
- PAPR** Peak-to-Average Power Ratio. 12, 13, 53
- PCB** Printed Circuit Board. 44, 45
- PER** Packet Error Rate. 13, 53
- PSK** Phase-Shift Keying. 12
- PSM** Power Saving Mode. 12
- PWM** Pulse Width Modulation. 24, 51
-
- RE** Receiver Enable. 31
- RO** Receiver Output. 31
-
- SAS** Smart Agriculture Systems. 18
- SC-FDMA** Single Carrier Frequency Division Multiple Access. 12, 13, 53
- SCK** Serial Clock. 24
- SCL** Serial Clock. 24, 26, 27
- SDA** Serial Data. 24, 26, 27
- SF** Spreading Factor. 10, 11, 53
- SIM** Subscriber Identity Module. 7
- SNR** Signal-to-Noise Ratio. 8, 10, 11, 53
- SPI** Serial Peripheral Interface. 24, 26
- SPOF** Single Point Of Failure. 5, 6
- SS** Slave Select. 24
-
- TCP** Transmission Control Protocol. 9
- TDS** Total Dissolved Solids. 19
-
- UART** Universal Asynchronous Receiver / Transmitter. 24, 31
- UDP** User Datagram Protocol. 9
- UE** Unión Europea. 12, 15
- UNB** Ultra Narrow Band. 11, 12, 53
- URL** Uniform Resource Locator. 35
- USB** Universal Serial Bus. 23, 24, 44
-
- WAN** Wide Area Network. 14
- WBAN** Wearable Body Area Network. 16

Wi-Fi Wireless Fidelity. 23, 44

WLAN Wireless Local Area Network. 23

WMN Wireless Mesh Network. 5

XML Extensible Markup Language. 35