TRABAJO FIN DE GRADO

INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas

Autor

Jesús Castillo Izquierdo

Directores

Jorge Navarro Ortiz

Sandra Sendra Compte

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, junio de 2017

Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas

Autor
Jesús Castillo Izquierdo

Directores
Jorge Navarro Ortiz
Sandra Sendra Compte

Departamento de Teoría de la Señal, Telemática y Comunicaciones

—

Granada, junio de 2017

Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas

Jesús Castillo Izquierdo

Palabras clave: monitorización ambiental, agricultura, *Internet of Things*, ESP8266, Arduino, sensores, ThingSpeak.

Resumen

El documento presente se centra en la inclusión de soluciones *Internet of Things* dentro de entornos agrícolas. Tras el estudio del mercado actual se ha detectado la carencia de un dispositivo de bajo coste plenamente adaptable a las necesidades del usuario. Por tanto, el objetivo principal es diseñar e implementar un dispositivo de bajo coste que registre la fluctuación de los factores ambientales. Asimismo, el instrumento desarrollado debe generar alertas cuando se presenten condiciones perjudiciales para el cultivo. Se utiliza el microcontrolador ESP8266 como sustento del resto de módulos que componen el dispositivo. El desarrollo del código controlador se realiza haciendo uso del entorno de programación Arduino. Los registros obtenidos podrán ser consultados a través de la plataforma web ThingSpeak. Tras la realización de las pruebas correspondientes, se ha confirmado el funcionamiento adecuado de la solución implementada. También se han reflejado con claridad las limitaciones que introducen algunos elementos en la autonomía del dispositivo. Para finalizar se presentan posibles líneas futuras de evolución.

Design and implementation of a IoT low cost device for agricultural environments.

Jesús Castillo Izquierdo

Keywords: environmental monitoring, agriculture, Internet of Things, ESP8266, Arduino, sensors, ThingSpeak.

Abstract

The present document focuses on the inclusion of Internet of Things solutions in agricultural environments. After a study of the current market, we detected the lack of a low cost device fully adaptable to the user necesities. Because of that, the main objective is to design and implement a low cost device that records the flutuation of environmental factors. Also, the developed instrument must generate alerts when harmful conditions for crop appear. The ESP8266 microcontroller is used as base of the rest of modules that build the device. The development of the controller code is made using the programming environment Arduino. The obtained registers can be consulted through the web platform ThingSpeak. After the realization of the corresponding tests, it has been confirmed the right performance of the implemented solution. Also, it has became clear the limitations that bring some of the elements to the device autonomy. Finally there are presented different future lines of evolution.

Yo, **Jesús Castillo Izquierdo**, alumno de la titulación Grado en Ingeniería de Tecnologías de Telecomunicación de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI XXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Jesús Castillo Izquierdo

Granada a 21 de junio de 2017 .

D. **Jorge Navarro Ortiz**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Dª. **Sandra Sendra Compte**, Profesora del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas***, ha sido realizado bajo su supervisión por **D. Jesús Castillo Izquierdo**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 21 de junio de 2017.

Los directores:

Jorge Navarro Ortiz **Sandra Sendra Compte**

Agradecimientos

Quiero mostrar mi agradecimiento a todas las personas que me han ayudado durante el desarrollo de este trabajo, en especial al director del mismo, Jorge Navarro Ortiz, por proporcionarme su ayuda en todo momento.

También me gustaría mostrar mi agradecimiento a mi familia y amigos, por apoyarme durante el transcurso de la carrera.

Índice general

1. Introducción **1**
1.1. Motivación 1
1.2. Objetivos 2
1.3. Metodología 3
1.4. Estructura de la memoria 4

2. Estado del arte **7**
2.1. Sistemas de monitorización agrícola 7
2.1.1. Libelium 7
2.1.2. Nazaríes IT 10
2.1.3. CropX 11
2.1.4. BioAgro Technologies 12
2.2. Crítica del estado del arte 14
2.3. Propuesta 15

3. Análisis del problema **17**
3.1. Análisis de requisitos 17
3.1.1. Requisitos funcionales 17
3.1.2. No funcionales 18
3.2. Planificación 19
3.3. Recursos 21
3.3.1. Recursos hardware 21
3.3.2. Recursos software 22
3.3.3. Recursos humanos 22
3.3.4. Coste del prototipo 23
3.3.5. Coste global del trabajo 24

4. Tecnologías empleadas **27**
4.1. IDE de Arduino 27
4.1.1. Interfaz gráfica 27
4.1.2. Configuración inicial 28
4.1.3. Código 28
4.2. ThingSpeak 29

4.2.1. Interfaz gráfica 30
4.2.2. Primeros pasos 31
4.3. Placa de desarrollo 32
4.3.1. Arduino 32
4.3.2. Raspberry Pi 33
4.3.3. ESP8266 34
4.3.4. Justificación de la elección 36
4.4. Sensor de temperatura 37
4.4.1. Alternativas disponibles 38
4.4.2. DS18B20 39
4.5. Sensor de humedad 40
4.5.1. Alternativas disponibles 40
4.5.2. Higrómetro YL-69 41
4.5.3. Sensor de lluvia MH-RD 42
4.6. Sensor de temperatura y humedad 42
4.6.1. Alternativas disponibles 43
4.6.2. Sensor de temperatura y humedad DHT11 43
4.7. Sensor detector de humos 44
4.7.1. Alternativas disponibles 45
4.7.2. Sensor MQ2 46
4.8. Módulo detector de movimiento 47
4.8.1. Alternativas disponibles 47
4.8.2. Módulo HC-SR501 49
4.9. Sensor fotoeléctrico 50
4.9.1. Alternativas disponibles 51
4.9.2. Fotorresistor GL5516 52
4.10. Multiplexor 53

5. Diseño e implementación 55
5.1. Algoritmo de funcionamiento 55
5.2. WiFiManager 56
5.3. ThingSpeak 58
5.4. Flexibilidad 59
5.5. Montaje físico 60
5.5.1. Alternativas consideradas 61
5.5.2. Implementación escogida 62

6. Resultados 65
6.1. Calibración 65
6.1.1. Calibración de alertas 65
6.1.2. Calibración de los sensores de humedad 66
6.1.3. Calibración del módulo detector de movimiento 67
6.1.4. Calibración de los elementos restantes 68
6.2. Consumo 68

6.2.1. Caracterización de los sensores 69
6.2.2. Recogida de datos 71
6.2.3. Deep Sleep 73
6.2.4. Estimación de la autonomía 74
6.3. Pruebas de funcionamiento 75
6.3.1. Primera prueba 75
6.3.2. Segunda prueba 78
6.3.3. Tercera prueba 80
6.3.4. Prueba auxiliar: sensor de humos 82
6.3.5. Análisis de los resultados 83

7. Conclusiones y líneas futuras 85
7.1. Conclusiones 85
7.2. Líneas futuras 86
7.2.1. Mejora de la autonomía 86
7.2.2. Actuadores 89
7.2.3. Nueva tecnología inalámbrica 90

Bibliografía 95

Índice de figuras

2.1. Waspmote. 8
2.2. Meshlium. 9
2.3. Dtalogger L1 2015. Fuente: Nazaríes IT 10
2.4. CropX 12
2.5. Dispositivo de Seguimiento BioAgro. Fuente: BioAgro 13

3.1. Diagrama de Gantt 20
3.2. Planificación. 21

4.1. Interfaz gráfica de Arduino. 29
4.2. Interfaz gráfica de ThingSpeak. 30
4.3. Arduino Uno. 32
4.4. Raspberry Pi 3. 33
4.5. WeMos D1 Mini. 37
4.6. Sensor de temperatura Ds18b20. 39
4.7. Higrómetro YL-69. 41
4.8. Sensor de lluvia MH-RD. 42
4.9. Sensor de temperatura y humedad DHT11. 44
4.10. Sensor detector de gases MQ2. 47
4.11. Módulo HC-SR51. 50
4.12. Fotorresistor GL5516. 52
4.13. Multiplexor CD74HC4067. 53

5.1. Diagrama de flujo. 56
5.2. Interfaz gráfica WiFiManager. 57
5.3. Claves de la API. 59
5.4. Wemos D1 Mini pinout. 61
5.5. Conexionado de los dispositivos. 63
5.6. Esquema de conexionado. 63
5.7. Implementación final. 64

6.1. Potenciómetro. 66
6.2. Módulo HC-SR501. 67
6.3. USB tester. 69

6.4. Factores ambientales. Primera prueba. 76
6.5. Factores ambientales. Primera prueba. 77
6.6. Alertas. Primera prueba. 78
6.7. Factores ambientales. Segunda prueba. 79
6.8. Factores ambientales. Segunda prueba. 79
6.9. Alertas. Segunda prueba. 80
6.10. Factores ambientales. Tercera prueba. 81
6.11. Factores ambientales. Tercera prueba. 82
6.12. Alertas. Tercera prueba. 82
6.13. Prueba de detección de gas. 83

7.1. Protector de batería para WeMos. 87
7.2. Empleo del *shield* sobre la placa WeMos. 88
7.3. Modos de alimentación del *shield*. 89
7.4. Electroválvula. 90

Índice de cuadros

3.1. Coste de los componentes. 24
3.2. Coste de los recursos hardware 24
3.3. Coste de los recursos software. 25
3.4. Coste de los recursos humanos. 25
3.5. Coste total. 25

4.1. Especificaciones Técnicas de la placa Arduino Uno 33
4.2. Especificaciones Técnicas de las principales placas Raspberry 34
4.3. Especificaciones Técnicas ESP8266 35
4.4. Especificaciones Técnicas de la placa WeMos D1 Mini 37
4.5. Especificaciones Técnicas del DS18B20. 40
4.6. Especificaciones Técnicas del YL-69. 41
4.7. Especificaciones Técnicas del MH-RD. 42
4.8. Especificaciones Técnicas del DHT11 44
4.9. Especificaciones Técnicas del MQ2. 47
4.10. Especificaciones Técnicas del HC-SR501. 50
4.11. Especificaciones Técnicas del GL5516. 52

5.1. Conexionado de la placa WeMos con los sensores 62

6.1. Resultados de las primeras pruebas de consumo 70
6.2. Consumo individual de los sensores 70
6.3. Influencia del tiempo entre muestras 71
6.4. Influencia del tiempo entre muestras con sensor MQ2 incluido 72
6.5. Consumo en modo Deep Sleep 73
6.6. Estimación de la autonomía. 74
6.7. Estimación de la autonomía sin MQ2. 75

Capítulo 1

Introducción

La influencia de las condiciones climáticas en el desarrollo de la flora es bien conocido. Debido al cambio climático, dichas condiciones están cambiando notablemente, lo cual modifica el comportamiento de la vegetación.

En lo relacionado a la agricultura, el cambio de las condiciones climáticas puede suponer la inhabilitación de ciertas zonas para el cultivo. Gracias a la monitorización y el análisis de dichos factores, es posible realizar las actuaciones necesarias para favorecer el desarrollo adecuado de un cultivo.

1.1. Motivación

El proyecto actual nace de la necesidad de monitorizar las condiciones ambientales de un cultivo agrícola, ya que afectan de forma directa al comportamiento del mismo, haciendo uso de un dispositivo de precio reducido.

El mercado actual ofrece varias soluciones IoT *(Internet of Things)* que permiten a sus usuarios analizar dichos factores. Dichas alternativas cuentan con características muy diversas y en ocasiones pueden llegar a encontrarse formando parte de sistemas más complejos, que a su vez pueden integrar circuitos de riego.

Existen diversos factores ambientales que influyen directamente en el desarrollo de una planta. Tal y como comenta Scott Shelton en Sweating High Humidity [1], la humedad es un factor que afecta notablemente. Dicho factor influye en dos funciones elementales de una planta, la transpiración y la fotosíntesis. Si la humedad es muy elevada, la transpiración se reducirá y la planta dejará de adquirir los nutrientes del suelo necesarios mediante absorción. En el extremo opuesto, una transpiración elevada puede provocar que la planta se marchite. La fotosíntesis se ve afectada de forma análoga.

La temperatura es otro de los factores más influyentes. El artículo Influencia de la temperatura ambiental en las plantas [2] nos ayuda a tomar noción de ello. Por norma general, las temperaturas elevadas suelen favorecer la aceleración de los procesos biológicos. Sin embargo, si la temperatura es demasiado elevada la transpiración aumentará en exceso.

También es importante monitorizar la duración del día solar. En el artículo La influencia de la luz en el crecimiento del cultivo [3] podemos observar de qué forma afecta dicho factor al desarrollo de una planta. Gracias al aporte de la energía solar, una planta es capaz de convertir materia inorgánica en orgánica mediante la fotosíntesis.

La principal motivación de este proyecto es diseñar e implementar un dispositivo que sea capaz de registrar adecuadamente dichos factores. Gracias a ello, será posible tomar las decisiones oportunas que permitan favorecer el correcto desarrollo de los cultivos.

Asimismo, se incluirán elementos que permitan garantizar la integridad de la plantación mediante la detección de posibles incendios y la detección de intrusos que puedan sabotear los cultivos.

1.2. Objetivos

Tal y como recoge el título del documento, el principal objetivo del proyecto será diseñar e implementar un dispositivo IoT de bajo coste para la monitorización agrícola. Será necesario reunir todos nuestro esfuerzos para cumplir dicho objetivo.

Es posible derivar una serie de objetivos secundarios de los cuales dependerá el cumplimiento de nuestra tarea. A continuación se enumeran dichos objetivos.

- Análisis del estado actual del mercado en relación con la monitorización de las condiciones ambientales mediante dispositivos IoT.
- Estudio y comparación de las principales placas de desarrollo.
- Utilización del entorno y lenguaje de programación Arduino.
- Estudio y comparación de los sensores ambientales disponibles en el mercado.
- Diseñar e implementar un dispositivo de bajo coste capaz de integrar sensores ambientales.

- Configuración de la plataforma ThingSpeak. Dicha plataforma nos permitirá almacenar los datos obtenidos mediante el dispositivo de monitorización.
- Instalación del dispositivo desarrollado para comprobar que cumple con los requisitos planteados.

1.3. Metodología

Entendemos la metodología como el conjunto de tareas o procedimientos que nos permiten alcanzar un objetivo determinado. A continuación nos centraremos en definir las fases que deberemos seguir para implementar el dispositivo deseado, al mismo tiempo que cumplimos con los requisitos planteados en la sección anterior.

En nuestro caso emplearemos una metodología lineal. Esto nos permitirá definir de forma clara y sencilla las distintas fases que deberemos cumplir para desarrollar el dispositivo. También se facilitará la gestión de cada una de las fases.

En primer lugar definiremos las distintas fases del proyecto para así poder estructurar adecuadamente el modo de trabajo, tras lo cual será posible iniciar el proyecto. Para ello será necesario tener muy en cuenta los objetivos que deseamos alcanzar. Tras finalizar una tarea será posible volver a trabajar en ella posteriormente para modificar lo que creamos conveniente, en caso de que sea necesario.

Si queremos desarrollar una alternativa competente, debemos analizar el estado del mercado actual. Con tal objetivo, estudiaremos y compararemos las principales soluciones que pueden ser encontradas actualmente. Esto nos permitirá en fases posteriores agregar a nuestro dispositivo funcionalidades que lo distingan de las demás alternativas. Tras ello será necesario definir claramente el comportamiento del dispositivo, para lo cual será necesario analizar y establecer una serie de requisitos.

Llegados a este punto deberemos sondear nuevamente el mercado, en esta ocasión en busca de los componentes que formarán parte de nuestro dispositivo. Será necesaria una comparación individualizada para adquirir la mejor opción en cada situación. Dicha elección deberá tener en cuenta los requisitos planteados.

Antes de comenzar a programar los distintos elementos será necesario comenzar a trabajar con el IDE de Arduino, para aprender cómo funciona dicho entorno. También será necesario documentarnos acerca de la pro-

gramación en Arduino. A partir de aquí nos centraremos en programar y configurar cada uno de los sensores de forma independiente, de modo que logremos que todos ellos funcionen adecuadamente.

La siguiente fase consistirá en diseñar e implementar la configuración final del dispositivo. Para ello, deberemos tener en cuenta todas las peculiaridades que puedan haber mostrado los sensores durante la fase de configuración individual. Posteriormente, conectaremos todos los elementos para que funcionen de forma simultánea. Tras verificar que todo funciona adecuadamente, nos centraremos en la plataforma ThingSpeak. Dicha plataforma será empleada para almacenar los datos sondeados.

Antes de finalizar el proyecto será necesario realizar una serie de pruebas que nos permitan concluir si el dispositivo cumple los requisitos deseados o no. Se realizarán pruebas para analizar el consumo y para ver el comportamiento del dispositivo en una situación real. Tras ello, se presentarán las conclusiones obtenidas y las principales líneas de desarrollo que puede sufrir el producto en fases futuras.

1.4. Estructura de la memoria

El desarrollo del proyecto será documentado mediante la realización de una memoria estructurada tal y como se indica a continuación.

- **Capítulo 1. Introducción.** Breve descripción de la motivación que mueve la realización de dicho trabajo, así como los principales objetivos que se pretenden alcanzar y la metodología seguida.
- **Capítulo 2. Estado del arte.** Estudio del estado actual del mercado para comparar las distintas soluciones disponibles. Tras observar las lagunas existentes en el sector se realizará una propuesta, la cual justificará el desarrollo de una nueva alternativa.
- **Capítulo 3. Análisis del problema.** Definición del comportamiento del dispositivo mediante el establecimiento de requisitos. Descripción de la opción propuesta y la planificación y presupuesto estimados para su realización.
- **Capítulo 4. Tecnologías empleadas.** Análisis y comparación de los elementos que ofrece el mercado. Elección de los elementos más adecuados para el desarrollo del dispositivo de monitorización.
- **Capítulo 5. Diseño e implementación.** Justificación del modo de funcionamiento del dispositivo. Ejemplo de uso de la plataforma

ThingSpeak y presentación las posibilidades de configuración ofrecidas a los usuarios. Justificación de la implementación escogida.

- **Capítulo 6. Resultados.** Puesta a punto de los distintos elementos para obtener registros fiables. Estudio del consumo y simulación de funcionamiento en condiciones reales.

- **Capítulo 7. Conclusiones y líneas futuras.** Exposición de las conclusiones obtenidas tras la elaboración del proyecto. Análisis de las futuras actuaciones sobre el dispositivo para mejorar sus cualidades y evitar la obsolescencia.

Capítulo 2

Estado del arte

A lo largo del siguiente capítulo se analizarán las principales soluciones existentes en el mercado que monitorizan las condiciones ambientales para aplicar los resultados obtenidos en la mejora de los cultivos agrícolas. Dicho análisis nos permitirá tener una visión clara del estado actual.

Tras ello seremos capaces de diferenciar claramente las cualidades y carencias del mercado, por lo que podremos realizar una crítica constructiva del mismo. Finalmente, describiremos la solución propuesta para mejorar y subsanar las carencias detectadas.

2.1. Sistemas de monitorización agrícola

Dentro del sector agrícola existen empresas tecnológicas que ofrecen a sus clientes la posibilidad de instalar una red sensórica dedicada a la monitorización. En esta sección nos centraremos en describir las características de las principales soluciones encontradas.

2.1.1. Libelium

Libelium [4] es una empresa tecnológica española fundada en 2006 por ingenieros españoles. Diseña y fabrica hardware y un kit de desarrollo de software (SDK) para redes de sensores inalámbricos. Sus productos ofrecen soluciones para dispositivos IoT, smart cities y M2M, que proporciona conectividad con la nube. La empresa surge tras detectar la necesidad de monitorizar parámetros ambientales de forma inalámbrica. Actualmente es una de las referencias dentro del sector.

Waspmote

Waspmote es una plataforma modular inalámbrica desarrollada por Libelium. Su uso se orienta a la implementación de redes de sensores de bajo consumo. Ofrece un tiempo de vida útil entre 1 y 5 años, que variará dependiendo del ciclo de funcionamiento del dispositivo.

Figura 2.1: Waspmote.

Libelium pretendía desarrollar una red de sensores haciendo uso de Arduino y XBee. Durante los primeros pasos del desarrollo encontraron dos problemas significativos. La alimentación de Arduino no permitía ser desconectada, lo cual evitaba que se pudiese desarrollar un modo de bajo consumo. Por otro lado, necesitaban que la plataforma radio estuviese certificada, ya que operaría en entornos reales.

A raíz de esto, surge la idea de desarrollar un dispositivo nuevo, especialmente diseñado para trabajar en redes sensoriales de bajo consumo. Actualmente, Waspmote es una plataforma estándar para IoT. Se trata de una solución escalable y modular, que integra 110 sensores diferentes para adaptarse a los requerimientos de cada uno de sus usuarios. También ofrece 15 tecnologías radio distintas.

La comunidad de desarrollo ha jugado un papel fundamental en el desarrollo de Waspmote. Libelium ofrece un sitio web con el objetivo de facilitar la colaboración y mejorar el soporte. Esto ha beneficiado la existencia de numerosos ejemplos de código para multitud de aplicaciones. Waspmote puede ser programado de forma sencilla, ya que es compatible con el IDE Arduino.

El precio de una unidad varía dependiendo de los módulos que deseemos incorporar. El precio mínimo de una placa es de 228 €. Podemos adquirir el kit de inicio por un precio de 360 € cada unidad. El lote incluye una placa

Waspmote, una pasarela, una batería de 2300 mAh y un cable miniUSB. Es posible incluir módulos ZigBee, GSM, 3G/GPRS, GPS, tarjetas de memoria y módulos de sensores.

Como ya se ha comentado, Libelium ofrece una gran gama de sensores. El precio de cada uno de los módulos variará desde un precio mínimo entorno a 20 euros hasta alcanzar los 600 en algunas ocasiones. Es posible adquirir un kit de evaluación por 4.000 €. Dicho kit incluye todo lo necesario para realizar una estación meteorológica con cuatro nodos y una pasarela.

Meshlium

Meshlium es otro de los dispositivos ofertados por Libelium. Se trata de un router multiprotocolo diseñado para conectar sensores ZigBee, Wi-Fi y bluetooth a la nube gracias a la conectividad 3G. La pasarela es capaz de enviar la información recogida por la red sensorial al mismo tiempo, ya que cuenta con una velocidad de subida de 5.5 Mb/s. También es posible compartir dicha información a través de Ethernet.

Figura 2.2: Meshlium.

Meshlium puede ser configurado con facilidad mediante una aplicación de control web de código libre, accesible desde cualquier navegador o smartphone. También ofrece a los desarrolladores de ejecutar sus propias aplicaciones, ya que contiene una distribución Linux que puede ejecutar códigos en C++, Java o PHP.

La plataforma cuenta con certificaciones CE y FCC, lo que permite usarlo en todo el mundo. El precio de una router Meshlium va desde los 690 €, la unidad más básica, a los más de 1300 € que cuesta la unidad más completa.

2.1.2. Nazaríes IT

Nazaríes IT [5] es una empresa tecnológica especializada en ofrecer soluciones a medida, tanto hardware como software. Acumula más de cinco años de experiencia en el desarrollo de soluciones tecnológicas en el ámbito de la monitorización mediante el empleo de redes de sensores. Asimismo, también ofrecen su experiencia en el empleo de sistemas de gran volumen de datos.

Ofrece varios productos orientados a distintos mercados. SKADE es un sistema que permite monitorizar el estado de la red vial, de modo que se facilite la gestión, conservación y explotación de la misma. CERES es un producto orientado a la monitorización agrícola, con el objetivo de optimizar la producción. Port Monitor es una herramienta on-line que permite a sus usuarios monitorizar el estado de sus páginas webs y servidores cada 60 segundos. Por último, ERPagro es una solución orientada a empresas que busca facilitar los procesos de producción y gestión.

CERES

CERES es un sistema de sensorización agrícola desarrollado por Nazaríes IT. Hace uso de redes sensoriales para ofrecer a sus usuarios información detallada de las condiciones ambientales, lo cual permite optimizar el proceso productivo.

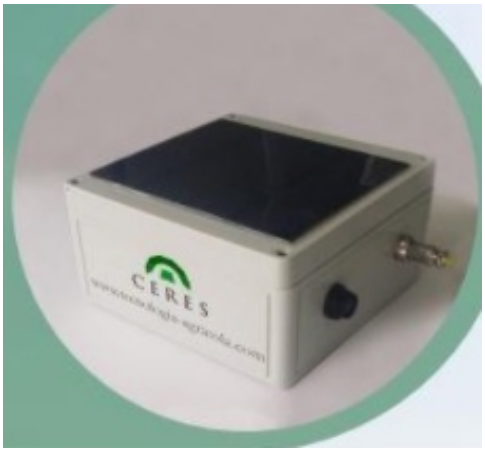

Figura 2.3: Dtalogger L1 2015. Fuente: Nazaríes IT

El objetivo de Nazaríes IT es ayudar a los agricultores a maximizar su producción mediante el uso de tecnología de última generación. Se ofrece un producto sencillo que permite la monitorización remota a través de diversas plataformas. Además de ello, permite el procesado y la posterior exportación de los datos a Excel así como su visualización mediante gráficas multivaria-

bles. CERES incluye un sistema novedoso mediante el cual se avisa al usuario de las alertas vía SMS o e-mail.

El sistema de monitorización se basa en el Datalogger L1 2015. Dicho dispositivo se encarga de registrar los datos obtenidos. Se trata de un elemento móvil, de tamaño reducido y con gran autonomía, gracias al uso de un panel solar. Cuenta con un microprocesador y una memoria interna que le permite almacenar los datos.

Las capacidades de monitorización son adaptables a las necesidades del consumidor, ya que es posible elegir los sensores que serán incorporados a dicho elemento. Dependiendo de los módulos que sean incorporados, tendremos la posibilidad de monitorizar los siguientes factores.

- Nitrato y potasio en suelo.
- Humedad en suelo.
- PH en agua y suelo.
- Humedad ambiental.
- Conductividad eléctrica en suelo.
- Punto de rocío.
- Constante dieléctrica imaginaria y real.
- Radiación solar.

CERES dispone de un servicio de pago por uso para aquellos usuarios que ya dispongan de una red de sensores propia. Gracias a esta herramienta les será posible trabajar con sus datos desde un portal web personalizado.

2.1.3. CropX

CropX [6] es una compañía americana que desarrolla soluciones software basadas en la nube que integran redes inalámbricas de sensores. Mediante la monitorización de las condiciones ambientales, ofrecen un servicio capaz de actuar sobre el riego de los cultivos, de tal modo que se ahorre energía y agua mientras se maximiza la producción de la explotación.

La compañía genera los mapas de riego de sus clientes teniendo en cuenta la orografía del terreno. Cuando adquiere el servicio, cada cliente recibe las

coordenadas en las cuales deberá instalar los diferentes nodos de la red para lograr un funcionamiento óptimo.

Actualmente, la empresa ofrece un único dispositivo, encargado de registrar la temperatura y humedad. Su principal ventaja es sencillez en la instalación. Basta con implantar el dispositivo en una determinada zona y escanear el código QR que tiene en la parte superior para que el nodo comience a funcionar. Por contra, es imposible añadir nuevas funcionalidades al producto, ya que no permite incluir nuevos sensores.

Figura 2.4: CropX

El precio del dispositivo que se muestra en la figura 2.4 es de 600 $. CropX ofrece sus servicios en dos paquetes distintos. El paquete básico de un año tiene un coste de 275 $ y el paquete de tres años cuesta 700 $. Es indispensable adquirir uno de estos paquetes, ya que es aquí donde la empresa nos proporciona la capacidad de monitorizar y realizar las actuaciones pertinentes.

2.1.4. BioAgro Technologies

BioAgro Technologies [7] nació en 2015 con el objetivo de resolver una plaga que afectaba a las palmeras. Se desarrolló un dispositivo capaz de detectar la presencia del insecto que afectaba a las palmeras. Dicha solución no tuvo el éxito previsto debido a que tras la detección era necesario un proceso de fumigación demasiado costoso.

Aprovechando la experiencia acumulada, BioAgro Technologies desarrolló una solución capaz de monitorizar las condiciones ambientales de un cultivo. Incluye un elemento capaz de monitorizar varios factores ambientales, un módulo de comunicaciones y una plataforma que recibe, almacena y procesa los datos.

En principio, la red fue desarrollada para trabajar en invernaderos, donde resulta más sencillo realizar actuaciones que alteren los factores ambientales. Tras comprobar el éxito de la solución inicial, esta se ha extendido a cultivos de exterior, tanto extensivos como intensivos.

Actualmente, todas las soluciones ofrecidas por BioAgro Technologies se basan el el Dispositivo de Seguimiento BioAgro (DSB). Dicho elemento recoge información en tiempo real de temperatura, humedad relativa, luminosidad, velocidad y dirección del viento, nubosidad, humedad a la profundidad de las raíces, conductividad y DPV (Déficit de Presión de Vapor). Los datos pueden ser consultados a través de la plataforma web o la aplicación ofrecida por la empresa. También pueden programarse alertas vía email o app.

Figura 2.5: Dispositivo de Seguimiento BioAgro. Fuente: BioAgro

Además del sistema de monitorización, la empresa ofrece dos soluciones de riego automatizado, que dependen directamente del DSB. La primera de ellas, BioAgro Aqua Airway, está pensada para cultivos de exterior y BioAgro Aqua se emplea en invernaderos. Ambos sistemas regulan la cantidad de agua y fertilizantes que se suministra a los cultivos.

2.2. Crítica del estado del arte

Llegados a este punto es posible extraer las primeras conclusiones. A pesar de lo que se podría pensar en un primer momento, la monitorización de las condiciones ambientales con fines agrícolas es un mercado en el cual existen una gran cantidad de productos.

La mayoría de las soluciones encontradas se centran en los cultivos en invernaderos. Es una decisión estratégica lógica y adecuada, ya que dentro de un invernadero el agricultor tiene la capacidad necesaria para modificar las condiciones ambientales. Gracias a la información recogida, se pueden modificar parámetros como la humedad, la temperatura o el aporte de nutrientes, lo cual mejorará la producción de la explotación.

En cuanto a las explotaciones situadas fuera de dichos entornos controlados, la situación cambia notablemente. En esta ocasión las redes de monitorización sitúan los nodos mucho más espaciados y los datos registrados no suelen ser tan exhaustivos. Esto se debe a que resulta imposible modificar parámetros como la temperatura, por tanto no es necesario un registro tan pormenorizado.

El poco margen de trabajo que poseen dichas soluciones se encuentra en la modificación de la humedad. Como hemos podido observar, es relativamente común que los elementos de monitorización sean integrados dentro de un sistema mayor. Dichos sistemas son capaces de reaccionar frente a los datos obtenidos y actuar sobre circuitos de riego para corregir efectos perjudiciales que se encuentren relacionados con la humedad.

En lo relativo al proceso de instalación, observamos alternativas muy diversas. En algunos casos es el propio agricultor el que decide donde colocar sus dispositivos, mientras que en otras ocasiones la instalación se realiza por parte del proveedor, debido a la mayor complejidad de la infraestructura. Todas las empresas observadas ofrecen a sus clientes la posibilidad de consultar sus datos de forma remota a través de navegadores web.

El elevado precio de los productos analizados supone una gran barrera para agricultores con explotaciones pequeñas. Al coste de los dispositivos es necesario añadir en muchas ocasiones un coste extra para acceder a las funcionalidades que permiten visualizar y analizar los registros. Resulta evidente que muchos usuarios potenciales dejarán de adquirir dichos productos debido a su precio.

A esto último es necesario añadir la escasa flexibilidad que ofrecen las empresas a sus clientes. Es posible que un usuario se decida finalmente a realizar una inversión en una de estas tecnologías y que posteriormente esta

no se adapte a sus necesidades. En algunas ocasiones el producto no permite ninguna modificación que permita ajustarse mejor a los deseos del consumidor. Aunque existen alternativas que permiten añadir nuevos módulos, lo cual incrementará aún más el precio, tampoco está del todo claro que dichos módulos permitan alguna configuración por parte del usuario.

2.3. Propuesta

Tras analizar el estado actual, nos centraremos en describir la alternativa que proponemos diseñar e implementar durante la realización del trabajo actual. Dicha solución deberá aglutinar las características que han hecho de los dispositivos anteriores unas herramientas de gran utilidad en el sector agrícola. Asimismo se incluirán varias modificaciones con el objetivo de solventar las carencias observadas.

La solución propuesta debe ser capaz de registrar adecuadamente factores ambientales como la temperatura y la humedad, tal y como hacen las soluciones comentadas. Además de ello, será posible medir la duración del día solar y detectar los periodos de lluvia sobre el cultivo. Los datos se almacenarán en una plataforma que permita a los usuarios observar la evolución de cada uno de ellos. Para ello, el dispositivo deberá ser capaz de conectarse a Internet y enviar los datos recogidos.

Asimismo el sistema contará con la posibilidad de configurar alertas para avisar cuando ocurran determinadas condiciones, como la aparición de lluvia o exceso de agua en el suelo. El sistema de alertas contará con dos novedades más respecto a sus competidores.

En primer lugar se instalará un sensor de humos que permita detectar la presencia de gases inflamables. Esto dará a los usuarios la posibilidad de actuar con rapidez frente a un posible fuego para reducir los daños sobre la cosecha. Con el fin de proteger la integridad del cultivo se instalará también un sensor detector de movimiento. Gracias a ello el sistema avisará de la llegada de posibles intrusos que puedan ocasionar daños o sustraer los productos.

Como se ha podido comprobar, el dispositivo cuenta con varias mejoras respecto a las soluciones observadas. Sin embargo, las principales ventajas competitivas no han sido comentadas aún. El sistema desarrollado podrá ser plenamente configurable por el usuario, permitiendo cambiar el número de registros realizados por hora, el nivel al que se activan las alarmas, el comportamiento ante dichas alertas y actuar sobre el consumo de los nodos.

Para concluir, el dispositivo debe ser capaz de contar con todas las cualidades previstas manteniendo un coste de producción muy reducido. Cuando revisamos las alternativas previas, se observó que los precios eran excesivamente elevados. En nuestro caso se pretende implementar un dispositivo con un coste de producción situado entre 10 y 20 € como máximo.

Capítulo 3

Análisis del problema

Antes de comenzar a desarrollar un proyecto, resulta indispensable una fase previa de análisis. Gracias a ello nos será posible obtener una visión global del procedimiento, lo cual nos será de ayuda en fases posteriores.

A lo largo del siguiente capítulo, nos centraremos sentar las bases de nuestro proyecto.

3.1. Análisis de requisitos

La definición de los requisitos nos ayudará a entender con mayor facilidad el funcionamiento del dispositivo. A lo largo de la siguiente sección se describen los requisitos que debe cumplir nuestro dispositivo para que pueda ser considerado como una buena alternativa dentro del mercado.

3.1.1. Requisitos funcionales

Los requisitos funcionales de un sistema definen las funciones que debe llevar a cabo el dispositivo. El incumplimiento de dichos requisitos supondrá la perdida de cualidades por parte de la solución desarrollada.

A continuación se incluyen los requisitos funcionales establecidos para el dispositivo.

- Medida de la temperatura. El dispositivo debe ser capaz de registrar la temperatura ambiental del entorno donde se encuentre situado.

- Medida de la humedad atmosférica. Cada elemento de la red debe ser capaz de registrar la evolución de la humedad atmosférica.
- Medida de la humedad terrestre. Asimismo, el dispositivo también debe registrar la evolución de la humedad en el suelo.
- Detección de lluvia. Cada uno de los nodos de la red debe ser capaz de detectar la aparición de lluvia.
- Detección de inundaciones. Cuando el nivel de humedad del suelo supere un cierto umbral, el nodo deberá generar una alerta de inundación.
- Medida del día solar. Cada nodo de la red debe ser capaz de registrar las horas de luz, ya que es un factor que influye directamente en el desarrollo de una planta.
- Detección de gases inflamables. Se debe registrar la evolución de los gases inflamables en el entorno para prevenir posibles incendios.
- Detección de incendios. Cuando el nivel de gas supere el límite establecido, el dispositivo debe generar una señal de alarma.
- Detección de intrusos. El dispositivo debe ser capaz de detectar la presencia de personas en su entorno.
- Sistema de alerta. El usuario debe disponer de una plataforma que le avise en caso de que se produzca alguna de las alertas anteriores.
- Presentación de los resultados. Asimismo cada usuario debe ser capaz de visualizar los datos recogidos por cada uno de los nodos de la red sin tener que encontrarse físicamente en la zona de operación.
- Autonomía. Dado que los nodos se situarán de forma dispersa, deben disponer de una fuente de alimentación que les conceda la suficiente autonomía.
- Resistencia a condiciones ambientales. Los nodos de la red serán situados en entornos donde tendrán que soportar fenómenos atmosféricos adversos. Deben ser capaces de operar en dichas condiciones.

3.1.2. No funcionales

Un requisito no funcional es aquel que se conoce y especifica criterios que pueden emplearse para juzgar el modo de operación de un sistema en lugar de sus componentes físicos. En definitiva, los requisitos no funcionales son

aquellos que no modifican el comportamiento del sistema que se pretende evaluar.

Estudiando el dispositivo que nos ocupa, podemos encontrar los siguientes requisitos no funcionales:

- Entorno de desarrollo. Es necesario disponer de un IDE para el desarrollo del código que controlará el funcionamiento del dispositivo. En este caso se ha elegido Arduino.

- Placa de desarrollo. Se necesita un elemento hardware al cual conectaremos los diferentes sensores y que será el encargado de la lectura y procesamiento de los datos. En este punto, el tutor indicó que la placa escogida debía utilizar el microcontrolador ESP8266. Este dispositivo es especialmente adecuado para nuestras necesidades ya que presenta un coste reducido y es adaptable a multitud de aplicaciones. Asimismo, el uso de un microcontrolador, ofrece un tiempo de arranque menor que otras opciones estudiadas, como podría ser una Raspberry Pi.

- Coste de producción reducido. Para favorecer la competitividad del dispositivo en el mercado actual, será necesario que su coste sea reducido.

- Flexibilidad. Las soluciones existentes se caracterizan por su poca flexibilidad. El dispositivo desarrollado debe ser configurable por los usuarios, para adaptarse plenamente a sus necesidades.

3.2. Planificación

Para llevar a cabo el desarrollo del proyecto actual será necesario seguir una planificación que nos permita ejecutar de manera ordenada las tareas necesarias para el cumplimiento de nuestros objetivos.

El diagrama de Gantt correspondiente a la planificación realizada se muestra en la imagen 3.1. En la figura 3.2 se registran el número de días que se deben emplear previsiblemente en el desarrollo de cada una de las fases.

Como se puede apreciar, existen etapas que serán realizadas simultáneamente. Asimismo, el desarrollo del código que nos permitirá leer los sensores se llevará a cabo en primer lugar sin comprobar su correcto funcionamiento. Esto se debe a que tras comprar los dispositivos, estos tardarán un tiempo en llegar hasta nosotros, ya que serán adquiridos a vendedores internacionales.

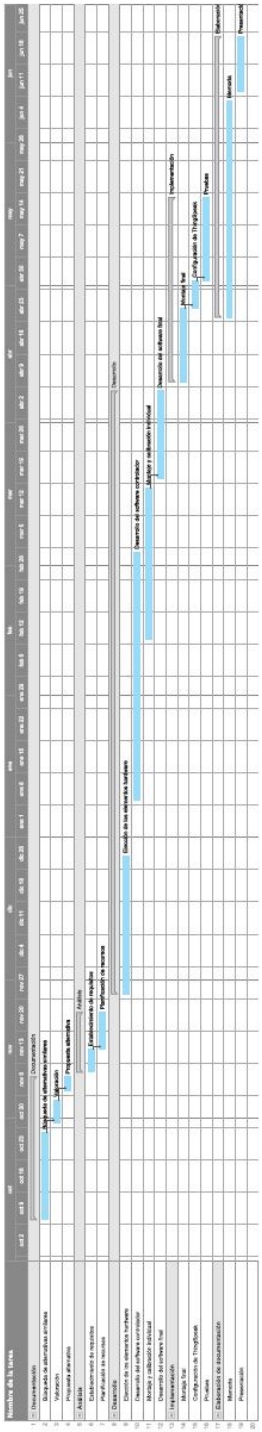

Figura 3.1: Diagrama de Gantt

Nombre de la tarea	Fecha de Inicio	Fecha final	Duración
Documentación	**10/10/16**	**09/11/16**	**23d**
Búsqueda de alternativas similares	10/10/16	28/10/16	15d
Valoración	31/10/16	04/11/16	5d
Propuesta alternativa	07/11/16	09/11/16	3d
Análisis	**11/11/16**	**23/11/16**	**9d**
Establecimiento de requisitos	11/11/16	15/11/16	3d
Planificación de recursos	16/11/16	23/11/16	6d
Desarrollo	**28/11/16**	**07/04/17**	**95d**
Elección de los elementos hardware	28/11/16	27/12/16	22d
Desarrollo del software controlador	09/01/17	03/03/17	40d
Montaje y calibración individual	13/02/17	17/03/17	25d
Desarrollo del software final	20/03/17	07/04/17	15d
Implementación	**10/04/17**	**19/05/17**	**30d**
Montaje final	10/04/17	25/04/17	12d
Configuración de ThingSpeak	26/04/17	01/05/17	4d
Pruebas	02/05/17	19/05/17	14d
Elaboración de documentación	**24/04/17**	**23/06/17**	**45d**
Memoria	24/04/17	09/06/17	35d
Presentación	12/06/17	23/06/17	10d

Figura 3.2: Planificación.

3.3. Recursos

Cuando nos encontramos frente a un problema, siempre es necesario realizar una estimación de los recursos necesarios para desarrollar la solución apropiada, lo cual nos ayudará a determinar si se trata de una alternativa viable o no. Durante la siguiente sección analizaremos los recursos que necesitaremos para cumplir nuestro objetivo.

3.3.1. Recursos hardware

Para desarrollar el software controlador del dispositivo será necesario hacer uso de un ordenador personal. Dicho dispositivo también será empleado durante la búsqueda de documentación y la elaboración de la memoria. El único requisito que debe albergar es que sea capaz de ejecutar con fluidez el IDE de Arduino, lo cual no es un problema ya que se trata de un software multiplataforma.

Se ha empleado un ordenador Acer Aspire E1-571G, cuyo precio es de 626 €. Fijando el tiempo de vida útil en 3 años y teniendo en cuenta que ha sido usado durante nueve meses, contar con este elemento supondrá un

total de 156.5 €.

También deberemos adquirir una placa de desarrollo y tantos sensores como sean necesarios para registrar adecuadamente los factores ambientales necesarios. En secciones posteriores se desarrollará con mayor profundidad el conjunto de elementos hardware necesarios para implementar el dispositivo de monitorización.

3.3.2. Recursos software

Los recursos software nos permitirán desarrollar el código capaz de controlar nuestro dispositivo y también serán utilizados para generar la documentación necesaria.

- IDE de Arduino. Entorno de programación que nos permitirá desarrollar el código controlador. Se puede obtener de forma gratuita en la página web de Arduino.
- ThingSpeak. Plataforma *online* empleada para almacenar los datos. Ofrece una versión gratuita que será empleada en nuestro caso, ya que cumple con los requisitos planteados.
- Overleaf. Plataforma web que permite desarrollar documentos en latex de forma gratuita. Será empleada para redactar la memoria.
- Fritzing. Programa gratuito empleado para realizar el diseño del esquema de conexión de los elementos que forman parte del dispositivo de monitorización.
- Smartsheet. Plataforma web empleada para realizar la planificación temporal del proyecto. Dispone de una versión de prueba gratuita suficiente para nuestras necesidades.
- Smartdraw. Plataforma *online* que utilizaremos para realizar un diagrama de flujo que refleje el comportamiento del dispositivo. Al igual que los recursos anteriores, dispone de una versión de prueba gratuita.

3.3.3. Recursos humanos

Dentro de este apartado encontramos a las personas que han trabajado en el desarrollo del proyecto. En primer lugar, los tutores del trabajo serán los encargados de ofertar, asignar y controlar el correcto desarrollo del

mismo. También serán los encargados de verificar el cumplimiento de los objetivos planteados. El alumno desarrollará el software de control y realizará la implementación del dispositivo, contando con la ayuda de los tutores para solucionar posibles problemas que puedan ocurrir durante el desarrollo.

Si repasamos brevemente la sección de planificación de nuevo, veremos que se ha planificado realizar el trabajo durante 202 días. La carga de trabajo en cada uno de estos días variará dependiendo de la tarea que se esté realizando en cada momento. El objetivo es que al final del trabajo el promedio de horas trabajadas por día sea en torno a 2 horas y media. Por tanto, el desarrollo del proyecto ocupará 505 horas.

Para calcular el salario del alumno se ha tenido en cuenta que se trata de un trabajador con poca experiencia, por lo que tendrá una retribución de 25 € por hora. El salario del tutor será el doble que el del alumno, ya que posee una mayor experiencia y es el encargado de la dirección del trabajo.

3.3.4. Coste del prototipo

En esta sección se presenta el coste de producción del prototipo inicial implementado. En capítulos anteriores se estableció como requisito que el coste del dispositivo debía ser tan reducido como fuese posible.

Sin embargo, el cumplimiento de dicho objetivo no debe perjudicar las cualidades del dispositivo, por lo cual es muy difícil establecer un margen a priori y sin tener en cuenta las características de los elementos que necesitaremos. Al comenzar el proyecto se fijó el límite presupuestario dedicado a este apartado en torno a 20 €.

En la tabla 3.1 se refleja el coste detallado de cada uno de los elementos que forman parte del dispositivo. Tal y como se pude observar, hemos sido capaces desarrollar un dispositivo de monitorización de bajo coste, lo cual era uno de nuestro requisitos.

Se ha decidido no introducir los gastos de envío ya que desvirtúan el coste real de los elementos. Dependiendo de las ofertas de los proveedores es posible obtener todos los elementos con los gastos de envío gratuitos. En caso de no disponer de ofertas el coste de envío puede suponer entorno a 10 euros adicionales.

Elemento	Precio €	IVA €	Cantidad	Precio total €
WeMos D1 Mini	2.28	0.48	1	2.76
MQ2	0.7	0.15	1	0.85
HC-SR501	0.63	0.13	1	0.76
GL5516	0.04	0.01	1	0.05
DHT11	0.56	0.12	1	0.68
Protoboard	1.69	0.35	1	2.04
DS18B20	0.79	0.16	1	0.95
Multiplexor	0.79	0.16	2	1.9
Higrómetro	0.32	0.07	1	0.39
Resistencia	0.03	0	3	0.09
Cable	0.25	0.05	2	0.6
Jumper	0.06	0.01	31	2.17
Coste final				**13.24**

Cuadro 3.1: Coste de los componentes.

3.3.5. Coste global del trabajo

Para concluir reuniremos en este apartado todos los costes que han sido enumerados a lo largo del capítulo, gracias a lo cual podremos obtener el coste total del proyecto.

En primer lugar se recoge el coste de los recursos hardware en el cuadro 3.2.

Elemento	Coste (€)
Ordenador personal	156.5
Coste del prototipo	13.24
TOTAL	**169.74**

Cuadro 3.2: Coste de los recursos hardware

El conjunto de elementos software listados en la sección 3.3.2. se encuentran recogidos en la tabla 3.3 junto a su coste.

Elemento	**Coste (€)**
IDE Arduino	0
ThingSpeak	0
Overleaf	0
Fritzing	0
Smartsheet	0
Smartdraw	0
TOTAL	**0**

Cuadro 3.3: Coste de los recursos software.

El cuadro 3.4 presenta los gastos derivados de las personas encargadas de la realización de trabajo.

Empleado	**Horas**	**Precio hora (€)**	**Total (€)**
Alumno	505	25	12625
Profesor	15	50	750
TOTAL			**13375**

Cuadro 3.4: Coste de los recursos humanos.

Por último, en el cuadro 3.5, se agrupan los costes anteriores para indicar el coste final.

Recursos	**Coste (€)**
Hardware	169.74
Software	0
Humanos	13375
TOTAL	**13544.74**

Cuadro 3.5: Coste total.

Capítulo 4

Tecnologías empleadas

A lo largo del siguiente capítulo se analiza, compara y explica el funcionamiento de las distintas soluciones empleadas en el dispositivo IoT. Asimismo se pueden observar tecnologías alternativas que pueden emplearse con el mismo fin, junto con la justificación de su descarte para la tarea actual.

4.1. IDE de Arduino

Un IDE, o entorno de desarrollo, es un entorno de programación que ha sido empaquetado como un programa de aplicación y que cuenta con las herramientas necesarias para facilitar a los desarrolladores el desarrollo de software. Cuenta con un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. El IDE Arduino permite desarrollar código y subirlo a la placa Arduino con facilidad. Es posible obtener el IDE Arduino de forma gratuita desde su página web.

4.1.1. Interfaz gráfica

A continuación se describe brevemente para qué sirve cada una de las opciones.

- **Archivo.** Esta pestaña funciona de manera análoga a cualquier otro programa. Permite crear, abrir, guardar, imprimir y cerrar cualquier proyecto. También cuenta con las opciones de configurar página y establecer preferencias. Resulta de gran utilidad para los usuarios la opción ejemplos. Dentro podemos encontrar muestras de código que orientan

sobre el modo de trabajar con las entradas/salidas, configurar los distintos modos de funcionamiento y escanear redes Wi-Fi, entre otras cosas.

- **Editar.** Cuenta con las herramientas de edición que dispone habitualmente un IDE, como cortar, pegar, deshacer, comentar, buscar, ir a la línea o aumentar sangría.
- **Programa.** Permite compilar y subir el programa a la placa. La gestión de las librerías también se realiza desde esta pestaña. Podemos descargar e incluir nuevas librería en el código desde aquí.
- **Herramientas.** Se trata de una de las pestañas más importantes para poder utilizar el IDE sin problemas. Posee diferentes opciones donde deberemos seleccionar la placa con la que estamos trabajando, frecuencia de CPU, velocidad de subida, puerto serie en el que se conecta la placa y tamaño de la memoria flash. También ofrece la posibilidad de dar autoformato y obtener información sobre la placa.
- **Ayuda.** Observamos distintas posibles elecciones donde obtener información acerca del entorno gráfico y los problemas que puedan ocurrir durante el trabajo con el programa.
- **Menú de botones.** Justo debajo del menú principal, podemos encontrar el menú de botones. Permite crear, abrir, guardar, compilar y subir un programa, además de abrir el monitor serie.

4.1.2. Configuración inicial

Antes de comenzar a programar, es necesario realizar la siguiente configuración. Nos dirigimos a Archivo $\prec$ Preferencias. Dentro del apartado Gestor de URLs Adicionales de Tarjetas incluimos la siguiente dirección.

`http://arduino.esp8266.com/stable/package_esp8266com_index.json`

Posteriormente, accedemos a Herramientas y en la opción Placa seleccionamos "WeMos D1 R2 & mini". Dentro de la misma pestaña deberemos seleccionar el puerto en el cual conectaremos posteriormente la placa.

4.1.3. Código

Antes de comenzar a desarrollar nuestro código, es necesario tener en cuenta cómo funcionan las estructuras setup y loop.

- **setup()** El código que incluyamos dentro de este apartado sólo será ejecutado cuando se inicie el programa. Debemos declarar las entradas y salidas que utilizaremos posteriormente. La inicialización y calibración de los módulos y sensores a emplear también se realiza dentro de este apartado.

- **loop()** Se trata de un bucle que ejecutará de forma continua el código que deseemos. Dentro de él incluiremos las funciones que permitan la correcta ejecución de nuestro programa.

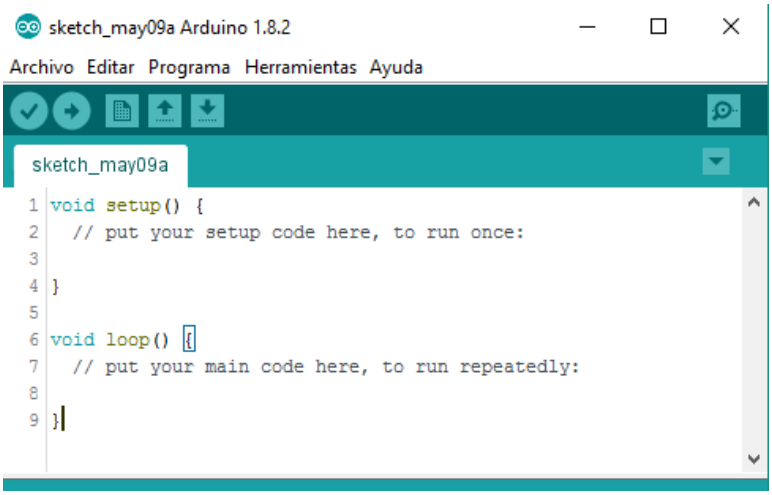

Figura 4.1: Interfaz gráfica de Arduino.

Dichas funciones son indispensables para el correcto funcionamiento de un programa. Cada usuario deberá incluir las librerías necesarias, en caso de que algún elemento lo requiera. La programación en Arduino es similar a otros lenguajes como C o Java. ‘

4.2. ThingSpeak

ThingSpeak [8] es una plataforma IoT que permite a sus usuarios recoger y almacenar datos y desarrollar aplicaciones IoT. Pertenece a MathWorks y ofrece aplicaciones que permiten analizar y visualizar los datos en MATLAB. Puede almacenar datos enviados desde dispositivos Arduino, Raspberry y Beaglebone, entre otros más. Crear una cuenta en ThingSpeak es totalmente gratuito y es posible mejorar las cualidades de la versión inicial mediante la adquisición de una versión de pago.

4.2.1. Interfaz gráfica

Cuando accedemos a ThingSpeak podemos observar el menú de bienvenida que aparece en la imagen 4.2. Las opciones que ofrece dicho panel serán comentadas a continuación.

Figura 4.2: Interfaz gráfica de ThingSpeak.

- **Channels.** Contiene tres secciones que nos permiten visitar nuestros canales, canales que hayamos visto con anterioridad o canales públicos que son ofrecidos por la comunidad.

- **Apps.** Reúne las herramientas ofertadas para trabajar con los datos. Es en este apartado donde podemos emplear las herramientas de MathWorks para realizar el análisis de los datos registrados, así como crear gráficos. También es posible programar acciones. Es posible configurar dichas acciones para enviar alertas mediante Twitter cuando los datos cumplan una determinada condición. También es posible enviar dichas alertas a una página web de nuestra elección mediante el protocolo HTTP.

- **Community.** Acceso directo al blog de la comunidad ThingSpeak. Dentro podremos consultar artículos, tutoriales y demás documentación relativa a ThingSpeak.

- **Support.** Ofrece un contenido similar a la sección community. En este caso podremos encontrar información relacionada con MathWorks.

- **How to buy.** La versión gratuita permite enviar en torno a 8200 mensajes diarios. Dentro de dicha pestaña aumentar el número de mensajes disponibles previo pago. Existen diversos precios, dependiendo del número de mensajes que se quieran enviar.

- **Account.** Ofrece información relativa a la cuenta del usuario. Podemos encontrar estadísticas de uso y modificar nuestra cuenta.
- **Sign out.** Empleada para salir de la cuenta.

4.2.2. Primeros pasos

Cuando accedemos por primera vez a la plataforma, tras realizar el registro, debemos ir a la pestaña Channel y seleccionar la opción New Channel. Una vez dentro podremos observar dos bloques de información claramente diferenciados.

En la parte izquierda de la pantalla se muestran una serie de campos vacíos donde podremos indicar las características del canal como el nombre, descripción, localización del nodo y campos de información, entre otros. En la parte derecha podemos encontrar un información de ayuda sobre cada uno de los campos anteriores. También aparecen enlaces a tutoriales sobre el uso adecuado del canal para obtener el máximo provecho.

No es necesario completar el formulario anterior al crear el canal, ya que esta información puede ser modificada en cualquier momento posterior. Tras crear el canal, podremos disponer de las siguientes opciones:

- **Private View.** Dentro de este apartado podremos visualizar gráficamente los datos recogidos por cada canal. Contiene información sobre el canal en cuestión (fecha de creación, última actualización, última entrada de datos y número de datos recogidos) y accesos directos a las herramientas de análisis.
- **Public View.** El funcionamiento es idéntico al de la pestaña anterior. Sin embargo, solo podremos observar dicha información si hemos definido nuestro canal como público.
- **Channel Settings.** Muestra el Channel ID, que será necesario cuando queramos enviar o recibir información del canal en cuestión. También se muestra el formulario que pudimos encontrar durante la creación del canal.
- **API Keys.** En esta sección se muestran las claves del canal. Debemos hacer uso de dichas claves cuando queramos escribir o leer información en el canal. También aparecen ejemplos de código que indican cómo realizar ambas operaciones.
- **Data Import/Export.** Permite importar o exportar información mediante archivos CSV.

4.3. Placa de desarrollo

Cuando nos enfrentamos a un proyecto de estas características, es indispensable contar con una placa de desarrollo. Este será el elemento fundamental entorno al cual se añadirán los demás elementos. Debido a su gran importancia, comenzaremos describiendo las dos opciones más conocidas del mercado. Posteriormente nos centraremos en analizar las placas que cuentan con el microcontrolador ESP8266, ya que es uno de los requisitos de nuestro proyecto.

4.3.1. Arduino

Arduino [9] es una plataforma electrónica libre basada en el empleo de hardware y software de gran simplicidad. Nace en el año 2005, dentro del entorno educativo, como respuesta a la necesidad de trabajar con un dispositivo de bajo coste, multiplataforma y que cuente con la suficiente documentación para permitir que un usuario pueda comenzar desde cero. Inicialmente fue desarrollado en un instituto de la ciudad italiana de Ivrea y más tarde fue liberado y abierto a la comunidad.

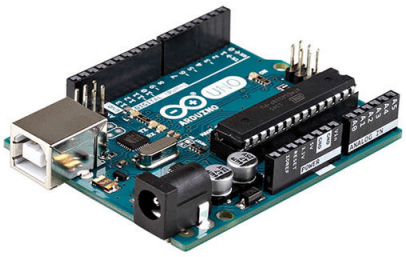

Figura 4.3: Arduino Uno.

Arduino cuenta con una familia de placas hardware que disponen de un microprocesador programable y una serie de pines. Tras grabar el código previamente desarrollado, los pines podrán comportarse como entradas o salidas y nos permitirán trabajar con diferentes sensores. En la tabla 4.1 se recogen las características de la placa Arduino Uno.

Una de las principales ventajas de Arduino frente a sus competidores es que posee una comunidad de usuarios activa muy amplia. Cualquier usuario puede modificar libremente el diseño de una placa, el entorno de programación o incluso el lenguaje de programación. Debido a ello, la documentación

Modelo A	**2 Modelo B**
Microcontrolador	Atmega328
Voltaje de operación	5 V
Voltaje de alimentación	6 - 20 V
Corriente por pin IO	40 mA
Corriente por pin 3.3 V	50 mA
Pines digitales	14
Pines analógicos	6
Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Cuadro 4.1: Especificaciones Técnicas de la placa Arduino Uno

que podemos encontrar es mucho mejor que la de cualquier competidor. Las placas son muy baratas y cada usuario puede optar por comprar los componentes por separado según sus necesidades particulares. Son fáciles de reprogramar, permitiendo así que puedan ser empleadas en varios proyectos distintos.

4.3.2. Raspberry Pi

Raspberry Pi es una placa computadora de bajo coste desarrollada en el Reino Unido por la fundación con el mismo nombre [10], que forma parte de la Universidad de Cambridge. Al igual que Arduino, surge en el entorno educativo, en este caso con la finalidad de estimular la enseñanza de la informática. Los primeros diseños de Raspberry Pi se basan en el microcontrolador Atmel ATmega644.

Figura 4.4: Raspberry Pi 3.

En cuanto al hardware, se trata de un producto registrado pero de uso libre. Desde el lanzamiento delas primeras 50 placas en agosto de 2011, el hardware que emplean estos dispositivos no ha parado de evolucionar. Actualmente existen varios modelos distintos a la venta. En el cuadro 4.2 se mencionan las principales características de algunos de ellos.

	Modelo A	**2 Modelo B**	**3 Modelo B**
Soc	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837
CPU	ARM1176JZF-S 700 MHz	ARM Cortex-A7 900 MHz Quad-core	QUAD ARM Cortex -A53 1.2 GHz
GPU	VideoCore IV	VideoCore IV	VideoCore IV
RAM	256 Mb	512 Mb	1 Gb
USB	1	2	4
Vídeo	RCA, HDMI	RCA, HDMI	Jack ,HDMI
Audio	Jack ,HDMI	Jack ,HDMI	Jack ,HDMI
Boot	SD	SD	MicroSD
Red	-	Ethernet 10/100	Ethernet 10/100, Wifi, BT
Consumo	300mA/1.5w/5V	800mA/4w/5V	2.5A/12.5w/5V
Precio	25 $	35 $	35 $

Cuadro 4.2: Especificaciones Técnicas de las principales placas Raspberry

Raspberry emplea mayoritariamente sistemas operativos open source como GNU/Linux. Su sistema operativo oficial, Raspbian, es una versión adaptada de Debian. También soporta las distribuciones ligeras multipropósito Moebius, Squeezed Arm Puppy y Minibian. Estas están especialmente concebidas para reducir al mínimo el consumo de recursos necesarios para funcionar. Por último, es posible instalar distribuciones ligeras que cuenten con un único propósito.

El gran éxito de Raspberry en la comunidad ha favorecido la aparición en el mercado de diversos módulos. Dichos elementos agregan funcionalidades a la placa base. Mediante estos módulos es posible doblar el número de pines GPIO de la Raspberry, agregar una cámara, añadir la tecnología NFC mediante una placa de expansión y añadir el número de entradas y salidas.

4.3.3. ESP8266

A lo largo de la siguiente sección nos centraremos en aquellas placas que cuentan con el microcontrolador ESP8266 [11]. Este dispositivo es el encargado del procesamiento y el control de la conexión Wi-Fi. Puede ser

empleado para albergar la aplicación o para otorgar conectividad a otro procesador en el que se encuentre la misma. El ESP8266 es uno de los chips más integrados del mercado. Requiere una circuitería externa mímima y ha sido diseñado para ocupar el menor espacio posible.

Algunos de las características que podemos encontrar en el datasheet, vienen recogidas la tabla 4.3.

Parámetro	**Condiciones técnicas**
Voltaje de operación	3.3 V - 3.6 V
Corriente de operación	80 mA
Protocolos Wi-Fi	802.11 b/g/n
Rango de frecuencia	2.4 GHz - 2.5 GHz
Consumo Deep Sleep	$\prec$ 10 μA

Cuadro 4.3: Especificaciones Técnicas ESP8266

Existen una gran variedad de opciones disponibles en el mercado. En nuestro caso nos centraremos únicamente en las placas de desarrollo, para lo cual nos ayudaremos del artículo de Todd Henderson Top 6 ESP8266 Modules for IoT Projects [12].

- **Adafruit Feather HUZZAH.** Puede ser programada mediante el IDE de Arduino. Entre sus ventajas destaca su sencillez de configuración, gracias al soporte y la gran cantidad de documentación existente. También dispone de un conector para baterías, lo cual añade portabilidad. Su precio es de 16.95 $.

- **NodeMCU Dev Kit.** NodeMCU es una plataforma de código abierto, gracias a lo cual existe mucha documentación. Utiliza el lenguaje de programación LUA. Cuenta con varios modelos cuyos precios oscilan entre 4 y 9 $.

- **KNEWRON smartWIFI.** Es una de las opciones más baratas. Destaca la sencillez de su configuración. Cuenta con un cargador de batería LiPo, un led RGB y utiliza el lenguaje de programación Lua.

- **SparkFun ESP8266 Thing.** Propiedad de SparkFun lanzada al mercado como solución para proyectos IoT. Muy similar a la primera opción en coste y características. La principal diferencia es que HUZZAH es más intuitivo y la flash es más robusta en el caso actual. Ambos modelos necesitan que soldemos los pines a la placa.

- **WeMos D1 Mini.** Es la mejor alternativa del mercado para proyectos simples. Aunque es una placa de desarrollo de tamaño reducido, cuenta

con un número decente de pines GPIO y suficiente memoria flash. Permite ser programada mediante Arduino. Su configuración simple, fiabilidad y buena documentación la han convertido en una de las placas más populares del mercado. Su precio se encuentra en torno a los 4 $. También es posible adquirir la versión estándar, que cuenta con un mayor número de pines, lo cual incrementa el tamaño y el precio.

4.3.4. Justificación de la elección

Los requisitos de nuestro dispositivo fueron concretados en el tercer capítulo. Si analizamos la información recogida durante esta sección, podremos entender por qué uno de dichos requisitos es el empleo de una placa que cuente con el ESP8266.

En primer lugar nos centraremos en la alternativa que a priori puede parecer más completa. Raspberry Pi nos ofrece conectividad Wi-Fi y un sistema operativo en el cual desarrollar nuestra aplicación. Esta segunda ventaja no es tal en nuestro caso, ya que el SO sería infrautilizado. Además de ello, el arranque del sistema conlleva un mayor tiempo de inicio y, por lo tanto, un mayor consumo del dispositivo, lo cual va en contra de nuestros objetivos. Por último, las placas Raspberry poseen el precio más elevado de entre los elementos comparados.

Las placas Arduino son una solución más económica que la anterior. El principal problema que encontramos es que no disponen de un módulo Wi-Fi, por lo que sería necesario adquirir este elemento por separado. Además de ello, Arduino no fue diseñado inicialmente para aplicaciones *Internet of Things*, debido a lo cual el consumo de las placas no es tan reducido como desearíamos.

Por último, el microcontrolador cumple perfectamente con nuestras necesidades. Ofrece los dispositivos con el coste más reducido y varias opciones de ahorro energético. Además de ello, cuenta con un número más que suficiente de pines. Como se ha visto, existen diversos elementos elementos en el mercado que cuentan con el ESP, lo que nos permite elegir la opción que más se adapte a nuestros requisitos.

Dentro de las placas que incorporan el ESP8266, la opción escogida es la WeMos D1 Mini. Posee el precio más reducido y cuenta con una gran cantidad de documentación disponible, lo cual nos facilitará su configuración. El tamaño de la placa es el más reducido de entre todos los elementos comparados en la sección actual y dispone de una cantidad suficiente de pines GPIO. A todo esto hay que sumar que su consumo en modo de ahorro de

energía es muy reducido.

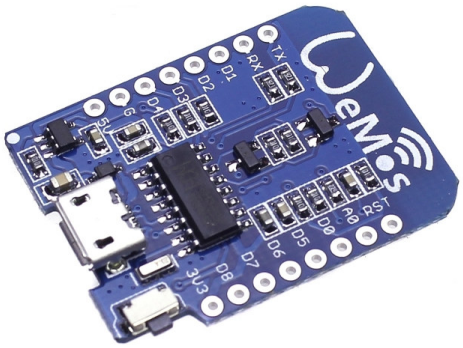

Figura 4.5: WeMos D1 Mini.

Las principales características de la placa aparecen recogidas en la tabla 4.4. Dicha información ha sido recogida de [13] D1 MINI [WEMOS ELECTRONICS].

Parámetro	**Condiciones técnicas**
Microcontrolador	ESP-8266EX
Voltaje de operación	3.3 V
Pines digitales (E/S)	11
Pines analógicos (E)	1
Velocidad de reloj	80 MHz / 160 MHz
Memoria flash	4 Mbytes
Longitud	34.2 mm
Anchura	25.6 mm
Peso	10 g

Cuadro 4.4: Especificaciones Técnicas de la placa WeMos D1 Mini

4.4. Sensor de temperatura

En la siguiente sección nos centraremos en analizar cómo añadir a nuestra red de sensores la capacidad de monitorizar la temperatura. Para ello, compararemos las principales alternativas existentes en el mercado y posteriormente detallaremos las cualidades de la opción empleada.

4.4.1. Alternativas disponibles

El mercado ofrece multitud de dispositivos capaces de obtener la temperatura. En nuestro caso nos centraremos en analizar aquellos con un menor coste.

Termistores

Si pretendemos medir la temperatura es indispensable considerar los termistores. Son elementos electrónicos cuya resistencia varía con la temperatura. Podemos encontrar termistores que aumenten su resistencia con la temperatura y otros que la disminuyan. Son fáciles de usar, baratos, rápidos, duraderos y relativamente precisos. Su principal inconveniente es que son dispositivos poco lineales. El precio varia dependiendo de la marca y el modelo, aunque se pueden encontrar por un precio en torno a 0.1 €.

LM35

El LM35 es un módulo con un circuito integrado que proporciona una tensión de salida proporcional a la temperatura. Se trata, por tanto, de un sensor analógico. Funciona con un voltaje de alimentación entre 3 y 5.5 V y dispone de tres pines. Su rango de medida se encuentra entre -55 ºC y 150 ºC y cuenta con una precisión de 0.5 ºC. Es un sensor bastante común, por lo que puede ser adquirido por un precio entorno a 0.6 € cada unidad.

TMP36

Nos encontramos nuevamente ante un sensor analógico, muy similar al anterior. Puede ser alimentado con un voltaje entre 2.7 V y 5.5 V. Su rango de operación va desde los -40 ºC a 125 ºC. Posee una precisión de 1 ºC. Como podemos ver, es algo más limitado que el anterior, a pesar de lo cual el precio es similar.

DS18B20

Se trata de un sensor que proporciona la salida mediante un bus que puede ser leído en cualquiera de las entradas digitales de un Arduino. Cuenta con tres terminales, dos de alimentación y un tercero donde se devuelve el valor obtenido. Utiliza la comunicación OneWire. Su rango de medición ocupa desde -55 ºC a 125 ºC, con una precisión de 0.5 ºC. El precio por unidad ronda los 0.5 € y si adquirimos el módulo sumergible el precio asciende hasta el euro.

4.4.2. DS18B20

En primer lugar se descartan los termistores debido a su baja linealidad. Si nos centramos en los sensores, el TMP36 es el que cuenta con las peores características y no ofrece una reducción considerable en cuanto al precio, por lo cual también será descartado. A pesar de que el sensor LM35 ofrece un rango de medición superior, el sensor elegido es el DS18B20. Puede ser adquirido por un precio menor y cuenta con una versión incluida en un módulo resistente al agua, lo cual será necesario en nuestra mota.

Figura 4.6: Sensor de temperatura Ds18b20.

Una ventaja muy importante, y que puede pasar desapercibida, es el hecho de que utilice comunicación OneWire. Se trata de un protocolo que nos permite enviar y recibir datos a través de un único cable. Dentro de un mismo bus OneWire pueden ser instalados tantos dispositivos como consideremos necesarios. Para ello, cada sensor dispone de una memoria ROM que es grabada de fábrica con un número de 64 bits. También es posible grabar en la memoria no volátil del DS18B20 los límites a partir de los cuales el dispositivo generará una alarma.

El sensor puede ser alimentado mediante el pin dedicado a ello o a través del pin de datos. El bus *OneWire* requiere una resistencia de 4.7 KΩ entre Vcc y la línea de datos para funcionar adecuadamente. Algunas de las características que podemos encontrar en el *datasheet* [14] aparecen recogidas en la tabla 4.5.

Parámetro	Condiciones técnicas
Voltaje de alimentación	3 VDC - 5.5 VDC
Señal de salida	digital
Rango de medida	-55 ºC - 125 ºC
Resolución	0.5 ºC
Tiempo de captura	$\prec$ 750 ms
Resolución	de 9 a 12 bits
Diámetro del cable	4 mm
Longitud del cable	91 cm

Cuadro 4.5: Especificaciones Técnicas del DS18B20.

4.5. Sensor de humedad

La siguiente sección incluye los elementos mediante los cuales nuestro dispositivo IoT será capaz de monitorizar adecuadamente la humedad ambiental.

4.5.1. Alternativas disponibles

No es habitual que una aplicación requiera registrar la humedad con gran exactitud. Debido a ello, existen pocos dispositivos desarrollados con tal fin. Con el objetivo de mejorar tanto como sea posible los resultados obtenidos incluiremos varios dispositivos.

En primer lugar se incluirá un sensor dedicado a medir la humedad del suelo. Como ya hemos visto, este es un factor fundamental en el desarrollo de una planta. Además, nos permitirá detectar posibles inundaciones que puedan afectar al cultivo. Dentro de este sector sólo ha sido posible encontrar los higrómetros YL-69 y FC-28. Son dispositivos de diferentes fabricantes que cuentan con unas características similares.

También se incluirá un módulo detector de lluvia. Al igual que en el caso anterior, las opciones que ofrece el mercado son reducidas. Podemos encontrar los módulos MH-RD, FC-37 e YL-83. Las diferencias entre ellos son meramente estéticas (variaciones en el tamaño o el color). En cuanto al funcionamiento, los tres elementos se comportan de forma idéntica.

En secciones posteriores se desarrollará con más detalle la inclusión de un tercer sensor de humedad, en este caso con el objetivo de registrar la humedad del aire con una mayor precisión.

4.5.2. Higrómetro YL-69

El higrómetro YL-69 [15] es capaz de medir la humedad del suelo aplicando una pequeña tensión entre sus terminales. La corriente que circula entre ellos depende de la resistencia que se genera en el suelo, que a su vez depende de la humedad. La sonda es alimentada mediante dos cables por el módulo YL-38. Dicho módulo incluye un comparador LM393. Dispone de dos pines de alimentación y otros dos donde podremos leer la salidas digital y analógica.

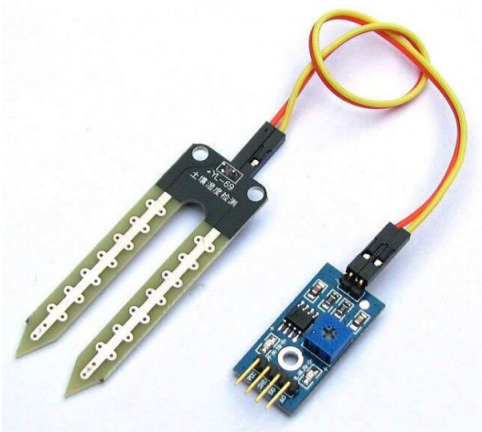

Figura 4.7: Higrómetro YL-69.

Resulta algo complejo obtener información de este módulo, ya que su uso no es muy habitual. A pesar de ello, son dispositivos muy baratos que pueden ser adquiridos por 0.4 € a suministradores internacionales. En el cuadro 4.6 se describen sus principales características.

Parámetro	**Condiciones técnicas**
Voltaje de alimentación	3 VDC - 5.5 VDC
Voltaje de salida	0 - 4.2 V
Dimensiones YL-69	60 x 30 mm
Dimensiones YL-38	30 x 16 mm

Cuadro 4.6: Especificaciones Técnicas del YL-69.

4.5.3. Sensor de lluvia MH-RD

El sensor MH-RD es capaz de detectar la presencia de lluvia. Contiene una placa con un circuito impreso. Cuando las gotas de lluvia caen sobre ella, se ponen en contacto los dos conductores del circuito, lo cual puede ser detectado por un sensor. Dicho sensor se conecta a un módulo que incorpora el comparador LM393. Dicho elemento se comporta de igual forma que cuando se encuentra conectado al higrómetro anterior.

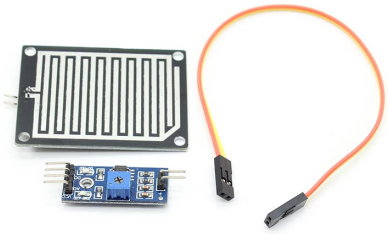

Figura 4.8: Sensor de lluvia MH-RD.

El sensor ha sido diseñado para detectar fenómenos atmosféricos como la lluvia o la nieve. En ningún caso podremos emplearlo para medir la cantidad de agua acumulada. Nuestro dispositivo se ayudará de este elemento para detectar y registrar los periodos de lluvia que tienen lugar en un determinado lugar. Es posible adquirir uno de estos módulos por un precio de 0.53 €. El cuadro 4.7 recoge algunas de sus características [16].

Parámetro	Condiciones técnicas
Voltaje de alimentación	5 VDC
Dimensiones YL-69	55 x 40 mm
Dimensiones LM393	32 x 14 mm

Cuadro 4.7: Especificaciones Técnicas del MH-RD.

4.6. Sensor de temperatura y humedad

Tal y como se comentó brevemente en la sección anterior, necesitamos incluir un sensor capaz de registrar la humedad ambiental. Durante el proceso de búsqueda y comparación, nos encontramos con una serie de sensores capaces de detectar la humedad y la temperatura simultáneamente. Esto nos permitirá cumplir dos requisitos mediante la inclusión de un único elemento.

4.6.1. Alternativas disponibles

La familia DHTXX cuenta con dos modelos capaces de obtener simultáneamente el valor de temperatura y humedad. Para ello, hace uso de un sensor capacitivo y un termistor. Cada módulo cuenta con un procesador interno que controla el proceso de medición y devuelve el valor obtenido mediante una señal digital. Gracias a ello, el proceso de sondeo de ambos sensores resulta muy cómodo desde Arduino. Ambos sensores poseen un encapsulado de plástico, para proteger sus componentes internos.

DHT11 vs DHT22

Pueden ser diferenciados a simple vista por el color de su recubrimiento, azul en el caso del DHT11 y blanco en el DHT22. Ambos sensores son fiables, ya que han sido calibrados en laboratorio. El sensor DHT11 presenta un tiempo de respuesta de 1 s mientras que el DHT22 tarda 2 s en responder. El sensor DHT22 cuenta con unas características superiores al otro modelo de la familia. Su rango de medidas es mayor, tanto en la medida de temperatura como de humedad. Asimismo, también cuenta con una mayor precisión en las medidas.

La diferencia de características se ve reflejada en el coste de ambos productos. Es posible encontrar el DHT11 por un precio de 0.8 € en vendedores internacionales. El precio del sensor DHT22 es notablemente superior, rondando los 2.5 € cada unidad.

4.6.2. Sensor de temperatura y humedad DHT11

En esta ocasión nos decantaremos por el sensor DHT11 [17], ya que su precio es algo más reducido. A pesar de que el sensor posee un rango de medición más reducido, dicho rango de operación coincide con las condiciones climatológicas bajo las cuales, a priori, funcionará el dispositivo IoT. A esto hay que sumar que la temperatura del aire no suele sufrir grandes cambios en periodos cortos de tiempo, por lo que no será necesaria una gran precisión.

El sensor DHT11 puede adquirirse en solitario o formando parte del un módulo. En la primera opción, cuenta con un encapsulado de plástico de color azul y cuatro pines. Como suele ocurrir con la mayoría de los sensores, dos de estos pines son empleados en alimentación y tierra. Un tercer pin se usa para devolver los valores obtenidos y el pin restante carece de uso. En caso de adquirir esta opción será necesario añadir una resistencia para proteger el sensor, de entre 4.7 kΩ y 10 KΩ.

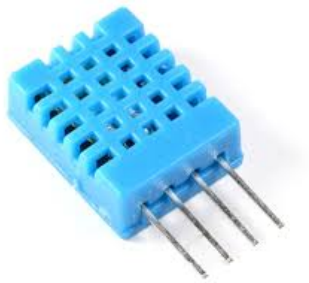

Figura 4.9: Sensor de temperatura y humedad DHT11.

En la segunda opción el módulo incorpora la resistencia necesaria y sólo cuenta con los tres pines útiles. También puede incorporar un condensador de filtrado de 100 nF. El coste de optar por esta opción supone un coste añadido de 0.1 € por unidad. En nuestro caso, nos hemos decantado por la primera opción ya que el precio es menor y contábamos con un paquete de resistencias, debido a que fue necesario adquirir una para el montaje del anterior sensor de temperatura.

El cuadro 4.8 incluye algunas de las características del DHT11.

Parámetro	**Condiciones técnicas**
Voltaje de alimentación	3 VDC - 5 VDC
Señal de salida	digital
Rango de medida de T	0 ºC - 50 ºC
Resolución T	0.1 ºC
Precisión T	± 2 ºC
Rango de medida de H	20 % - 90 % RH
Resolución H	1 % RH
Precisión H	± 4 % RH

Cuadro 4.8: Especificaciones Técnicas del DHT11

4.7. Sensor detector de humos

A lo largo de la siguiente sección nos centraremos en analizar las soluciones existentes en el mercado que pueden añadir al dispositivo la capacidad de detectar gases inflamables.

4.7.1. Alternativas disponibles

Es posible encontrar una gran cantidad de elementos de este estilo. En nuestro caso nos centraremos sólo en aquellos que presenten un precio más competitivo.

Familia MQ

Ofrece sensores capaces de detectar una gran variedad de gases y que pueden ser utilizados con facilidad junto con un Arduino. La familia MQ destaca por su alta sensibilidad, respuesta rápida ante cambios en la concentración, amplio rango de detección, circuitería simple, estabilidad y elevado periodo de vida útil. Algunos de los sensores que podemos encontrar dentro de esta familia son los siguientes.

Sensor MQ2. Adecuado para detectar GLP (Gas Licuado del Petróleo), propano, metano, alcohol, hidrógeno y humo. Es especialmente sensible al GLP y propano. El precio de una unidad de estos sensores es de 0.85 €.

Sensor MQ3. Es especialmente sensible al alcohol y etanol. Detecta gases como GLP, CO, Hexano y CH14, pero con muy poca sensibilidad, por lo que dicha cualidad es despreciable en condiciones normales. También se puede emplear para detectar humo, aunque su sensibilidad es muy reducida en este aspecto. Es posible adquirir un sensor MQ3 por 1.11 €.

Sensor MQ135. Usado principalmente en equipos de control de calidad del aire en edificios. Apto para la detección de NH3, NOx, CO2, alcohol y humos. Su sensibilidad es similar ante cualquiera de los gases nombrados. Se emplea para controlar la calidad del aire. El precio de un sensor MQ135 es de 1.27 € por unidad.

Serie MG

Actualmente esta serie solo cuenta con el sensor MG811. Se emplea como sistema de detección de incendios y cuenta con un detector de gas y una alarma, que se dispara tras superar un umbral. Caracterizado por su gran sensibilidad, estabilidad y consumo reducido. Para la detección de incendios, monitoriza la presencia de CO2 en el aire. El hecho de que el gas objetivo sea el CO2 puede llegar a afectarnos. Sería necesario estudiar con detenimiento si el proceso de fotosíntesis podría falsear los registros obtenidos, especialmente en entornos cerrados como los invernaderos. El precio por unidad ronda los 24 euros.

Familia MC

Cuenta con tres modelos para la aplicación en hogares y cinco para

uso industrial. Destaca por ser dispositivos con una señal de salida muy lineal. Son elementos estables y con una respuesta rápida, sin embargo su funcionamiento se ve muy afectado por la temperatura y la humedad. Debido a ello, es una gama de sensores poco empleada, por lo cual resulta difícil encontrar suministradores. Dentro de la familia, el sensor que más se ajusta a nuestras necesidades es el MC105, que detecta gases combustibles. El precio de este sensor se encuentra en torno a 2.5 € por unidad.

Familia MP

Las principales características de la familia MP son la alta sensibilidad, respuesta rápida, amplio rango de detección, estabilidad, circuitería simple, bajo consumo y larga vida útil. El principal problema que presenta esta familia es que cada uno de los sensores tiene como objetivo un solo gas, dos a lo sumo, lo cual condiciona el proceso de detección en un entorno donde pueden coexistir varios gases. El precio mínimo de un sensor de esta serie es de 2.5 euros.

4.7.2. Sensor MQ2

Teniendo en cuenta las especificaciones de cada familia, las dos mejores alternativas son la familias MQ y MP. La serie MG queda descartada por su elevado precio y la familia MC porque sus dispositivos se ven muy afectados por las condiciones ambientales. La familia MP es muy poco conocida, lo cual dificulta encontrar algunos de sus dispositivos. Otro inconveniente que presenta es que el sensor no cuenta con un módulo que ayude a configurar los niveles de alerta. La familia MQ funciona con un mayor consumo, algo más del doble que la familia MG. El último aspecto a comparar es el precio, donde la familia MQ presenta opciones más asequibles.

Finalmente se ha optado por la familia MQ. Los factores determinantes han sido el menor coste, el hecho de que se ofrezca un módulo que ayude a configurar y calibrar los dispositivos y la amplia documentación existente, que sin duda facilitará la integración en el dispositivo. Dentro de esta familia, se ha escogido el sensor MQ2. Se trata del sensor más barato de la familia y el más indicado para la detección de humos.

Está compuesto por un sensor electro-químico que varía su resistencia al estar en contacto con ciertos gases. Como ya se ha comentado, presenta una rápida respuesta en la detección. Sin embargo, el tiempo para volver a estabilizarse tras una detección es elevado, ya que es necesario que el gas abandone las zonas sensibles del sensor. Para funcionar adecuadamente, necesita que sus componentes alcancen una temperatura adecuada, lo cual

requiere unas 24 horas. Cuenta con un calentador para elevar la temperatura.

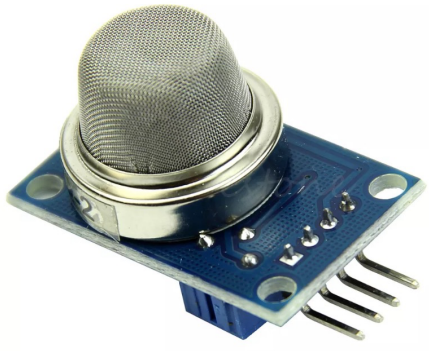

Figura 4.10: Sensor detector de gases MQ2.

El sensor adquirido forma parte de un módulo que también cuenta con un comparador. Este elemento nos ayuda a establecer un nivel de gas a partir del cual se generará una alerta. El módulo dispone de cuatro pines, dos para alimentación y tierra y dos más para obtener las lecturas digital y analógica. En el cuadro 4.9 se recogen algunas de sus características [18].

Parámetro	**Condiciones técnicas**
Voltaje de alimentación	5 V
Consumo	$\prec$ 800 mW
Temperatura de uso	20 °C - 50 °C
Humedad relativa	$\prec$ 95 %
Tasa de concentración	0.6

Cuadro 4.9: Especificaciones Técnicas del MQ2.

4.8. Módulo detector de movimiento

En el tercer capítulo contemplamos la necesidad de contar con un dispositivo detector de presencia. A continuación compararemos las mejores soluciones existentes que nos permitan lograr dicho objetivo.

4.8.1. Alternativas disponibles

El mercado ofrece una gran variedad de opciones en lo relacionado a la detección de movimiento. A continuación se describen las diferentes tecnologías consideradas en la elección de este elemento.

Sensor PIR.

Cualquier cuerpo cuya temperatura supere el cero absoluto emite radiación. Los sensores PIR detectan dicha la radiación infrarroja. Para ello, contienen dos zonas sensibles que reaccionan cuando un cuerpo se sitúa frente a ellas. Si se detecta movimiento, el sensor emite una alerta. Por sí solos, estos sensores poseen un rango de detección muy limitado y no son configurables.

Adquirir uno de estos sensores suele costar en torno a 0.4 € cada unidad. Existen ofertas para grandes pedidos que reducen el coste, las cuales no serán tenidas en cuenta ya que sólo se necesitaría uno de estos sensores para configurar la mota inicial. Para aumentar el rango de detección, sería necesario adquirir una lente de Fresnel por separado, lo cual añadiría un coste entorno a 0.1 €, dependiendo del modelo y proveedor.

Mini sensor PIR HC-SR505.

Basa su funcionamiento en la tecnología infrarroja, haciendo uso de un sensor PIR. Destaca por ser uno de los dispositivos de menor tamaño en su categoría. En cuanto a las especificaciones, observamos que puede ser alimentado con nuestro Arduino, ya que opera en un rango entre 4.5 V y 20 V. Como la mayoría de los dispositivos de su categoría, cuenta con tres pines, alimentación, tierra y salida. El pin de salida cambiará su estado de baja a alta cuando detecte movimiento. Su rango de detección alcanza un máximo de tres metros. El uso de este dispositivo no está muy extendido. La mejor oferta encontrada consta de un lote de 5 unidades por un precio final de 1.35 €/unidad.

Sensor HC-SR501.

Al igual que la solución anterior, se trata de un módulo que cuenta con un sensor piroeléctrico dividido en dos zonas sensibles. El funcionamiento también depende de la tecnología infrarroja. Cuenta con tres pines que funcionan de forma análoga al sensor HC-SR505. El uso de este módulo está muy extendido. Su rango de detección varía entre 3 m y 7 m. Podemos encontrar estos dispositivos por un precio en torno a 0.8 €/unidad más gastos de envío.

Sensor de ultrasonidos HC-SR04.

Su funcionamiento se basa en la emisión de un pulso de alta frecuencia, que rebota en los objetos situados frente a él. El módulo dispone de un micrófono para captar los pulsos reflejados. Haciendo una comparativa entre el tiempo entre los pulsos, es capaz de calcular la distancia hasta un cuerpo. En definitiva, se trata de un dispositivo medidor de distancia que podríamos

adaptar para detectar movimiento. En principio, el hecho de que no esté diseñado para cumplir el objetivo que perseguimos presenta las primeras reticencias para su elección.

Son sensores muy comunes, cuyo rango de medición va desde 2 cm a 4 m. Dispone de cuatro pines, dos de ellos empleados en alimentación y tierra. Los dos restantes se usan para generar lo pulsos y recoger el pulso reflejado. El usuario debe programar en Arduino el código que le advierta cuando se ha producido una alerta. Su precio ronda el euro, siendo la mejor oferta de 0.97 €/unidad.

Sensor HB100.

Sensor de movimiento fundamentado en el efecto doppler. Compuesto por un oscilador, un mezclador de microondas y una antena da recepción. Alimentado a una tensión de 5 V, que no tiene que por qué ser constante (mediante trenes de pulsos). Como en los casos anteriores dispone de tres pines. No permite variar la frecuencia ni la potencia de emisión. Su precio es superior a los tres euros debido a su escasa aplicación.

4.8.2. Módulo HC-SR501

Existen otros dispositivos cuya distancia de detección supera los 10 metros, capaces de aumentar el grado de detección hasta los 360º y con una mayor precisión y rapidez en la detección, entre otras características. La elección de alguno de estos módulos provoca que el precio por unidad supere los 10 € por unidad. Ya que uno de los principales objetivos es obtener un dispositivo con un coste reducido, no se estudiará el uso de una opción con un coste tan elevado.

El sensor elegido para realizar la tarea de detección es el módulo HC-SR501. Es uno de los más empleados en este tipo de aplicaciones, debido principalmente a su sencillez, y la existencia de múltiples suministradores. Además, es posible encontrar una gran cantidad de webs que contienen documentación relacionada. Cuenta con el mayor rango de detección de entre los dispositivos comparados. El coste por unidad es el segundo más reducido, siendo el coste del producto adquirido de 0.76 €/unidad.

El módulo HC-SR501 cuenta con dos potenciómetros y un jumper que nos permiten alterar el funcionamiento del dispositivo. Gracias a ellos es posible regular la precisión, el comportamiento ante las detecciones y el tiempo de activación. La resistencia RL2 permite modificar el rango de detección entre 3 y 7 metros y la resistencia Ch1 modifica el tiempo durante el cual la salida se mantiene en alta. Este valor puede variar entre 5 segundos y 5 mi-

nutos. Esto último resulta muy interesante, ya que el hecho de que podamos mantener un tiempo la alerta de detección permite usar este dispositivo sin ir acompañado de un microcontrolador.

Figura 4.11: Módulo HC-SR51.

Cuenta con dos modos de funcionamiento. En el modo continuo el sensor mantiene la señal de salida activa de manera ininterrumpida mientras detecta movimiento frente a él. Para activar dicho modo es necesario puentear los contactos entre las resistencias RL2 y Ch1. En el modo de repetición el sensor se activa al detectar movimiento y después vuelve al estado de reposo. Si sigue detectando movimiento volverá a repetir el mismo ciclo. El tiempo que el sensor está activo en este modo pude regularse con la resistencia Ch1.

El cuadro 4.10 recoge algunas de las características [19] del módulo HC-SR501.

Parámetro	**Condiciones técnicas**
Rango de detección	3 m a 7 m
Lente de Fresnel de 19 zonas	$\prec$ 100º
Salida activa alta	3.3 V
Consumo de corriente en reposo	50 μA
Voltaje de alimentación	4.5 VDC a 20 VDC

Cuadro 4.10: Especificaciones Técnicas del HC-SR501.

4.9. Sensor fotoeléctrico

Un sensor fotoeléctrico es un elemento electrónico que reacciona frente a cambios en la iluminación del entorno en el cual se encuentre situado. El

funcionamiento se basa en la emisión de luz frente a un objeto y el estudio del mismo haz de luz tras interferir con el objeto. En nuestro caso no es necesario la caracterización de objetos y disponemos de fuente de luz, la luz solar. Debido a ello, nos centraremos sólo en el componente de estos sensores que reacciona frente a los cambios luminosos.

4.9.1. Alternativas disponibles

Un fotorreceptor es un sensor capaz de convertir la energía solar en energía eléctrica. La corriente producida en la salida varía con la intensidad con la que incide la luz sobre la superficie sensible. El funcionamiento se basa en que la incidencia de fotones, con una determinada energía, provoca la transición de electrones de la banda de valencia a la de conducción. Existen varios tipos de fotorreceptores.

Fotorresistor

El fotorresistor o fotoconductor cambia el valor de su resistencia dependiendo de la energía que recibe. A mayor intensidad de la luz incidente, menor es su resistencia. Dentro de los fotorreceptores, son los elementos que reaccionan con mayor lentitud a los cambios. Su salida no es muy lineal, por lo que solo se pueden emplear para caracterizar la luz que reciben, nunca para cuantificar la intensidad.

Fotodiodo

El fotodiodo varía el valor de la corriente que circula por él en sentido inverso. Cuanto mayor es la excitación, mayor es la corriente que circula. Son elementos muy lineales, cuyo principal inconveniente es que no dejan circular corriente en un sentido, lo cual sólo afecta al modo en que deben ser alimentados

Fototransistor

Por último, el fototransistor conduce más o menos corriente de colector dependiendo de la incidencia de luz. Su principal ventaja es que amplifica la corriente producida. Sin embargo, es necesario polarizarlos para que funcionen correctamente y, aún en ese caso, su salida no es del todo lineal, lo cual hace más complejo el proceso de medida. Suelen ser empleados en aplicaciones como interruptores u otros elementos cuyo estado sea todo o nada.

4.9.2. Fotorresistor GL5516

Las características especiales de nuestra fuente de luz (no presenta grandes variaciones en periodos de tiempo cortos) hacen que cualquiera de los dispositivos que estamos comparando puedan ser usados indistintamente. Los fototransistores son los dispositivos que presentan más inconvenientes, ya que su funcionamiento es más complejo. El hecho de que los fotodiodos posean más precisión que los fotorresistores no es diferencial en este caso, ya que lo que buscamos es una caracterización de la luz recibida. Finalmente, se ha optado por el uso de una fotorresistencia, ya que su precio suele ser la mitad que el de un fotodiodo.

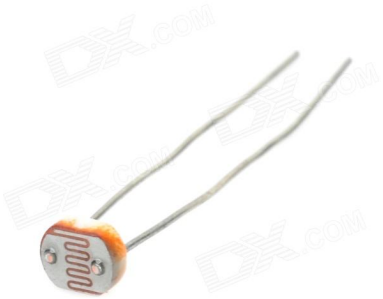

Figura 4.12: Fotorresistor GL5516.

A la hora de adquirir los fotorresistores nos centraremos en la familia GL55. La mayoría de los dispositivos que podemos encontrar forman parte de ella. Cuenta con 6 tipos distintos de fotorresistencias dentro de su serie. Todos ellos poseen características de funcionamiento similares. La principal diferencia reside en el valor de resistencia que presentan dependiendo de la iluminación. En este caso nos hemos decantado por el modelo GL5516, que tiene los valores más bajos de resistencia. El lote adquirido dispone de un total de 20 fotorresistencias por un precio de 5 cents/unidad. Sus principales características [20] se encuentran recogidas en el cuadro 4.11.

Parámetro	**Condiciones técnicas**
Voltaje máximo	150 V
Temperatura ambiental	30 ºC - 70 ºC
Tiempo de respuesta	30 ms
Mínima resistencia	5-10 KΩ
Máxima resistencia	0.5 MΩ

Cuadro 4.11: Especificaciones Técnicas del GL5516.

4.10. Multiplexor

Un multiplexor es un dispositivo electrónico que nos permite ampliar el número de entradas o salidas disponibles de una placa de desarrollo como WeMos D1 Mini. La justificación del empleo de dicho elemento será realizada posteriormente en el capítulo 5. En este punto nos centraremos en definir el comportamiento y las cualidades de dicho elemento.

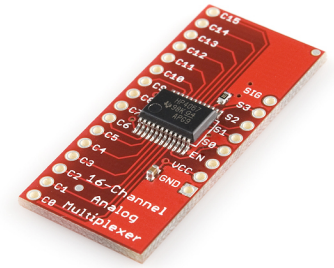

Figura 4.13: Multiplexor CD74HC4067.

El multiplexor CD74HC4067 [21] cuenta con 16 canales bidireccionales. Es posible controlar estos canales mediante solo 5 pines. Se necesitan cuatro pines para seleccionar el canal elegido y uno más en el cual se recoge la señal leída o escrita. El CD74HC4067 funciona con señales analógicas o digitales indistintamente. También puede ser empleado en dispositivos de comunicación como el puerto serie, bus i2c o bus SPI.

El máximo de corriente que puede proporcionar coincide con el máximo que puede proporcionar un pin de Arduino, 20 mA. Debe ser alimentado con una tensión entre 2 y 6 V. La conexión es muy sencilla. En primer lugar alimentamos el multiplexor mediante los pines VCC y GND. Posteriormente conectamos 4 salidas de Arduino en los pines S0, S1, S2 y S3, que controlan la selección de los canales. La señal obtenida se recoge en el pin SIG. En este punto ya solo queda conectar los sensores correspondientes en los canales disponibles.

Los multiplexores CD74HC4067 son muy comunes y pueden ser encontrados por un precio de 0.95 € en vendedores internacionales.

Capítulo 5

Diseño e implementación

Durante el proceso de diseño se analizarán las posibles estrategias a seguir para ofrecer a los usuarios una solución óptima que cumpla con sus requerimientos. En primer lugar se esboza el comportamiento predeterminado de cada uno de los dispositivos, para posteriormente detallar las posibles modificaciones que se pueden realizar dependiendo de la voluntad del usuario final. Por último se describe el proceso de conexión de los componentes.

5.1. Algoritmo de funcionamiento

En primer lugar es necesario indicar el modo de funcionamiento predeterminado del dispositivo. Es muy importante presentar esta información de la forma más clara y sencilla posible, para facilitar la comprensión del lector.

Con tal objetivo, nos ayudaremos de un diagrama de flujo. En la figura 5.1 se muestra el funcionamiento del código cargado en el microcontrolador.

Tal y como se puede comprobar, el dispositivo se encontrará de forma predeterminada en estado de bajo consumo. Tras un determinado periodo, se despertará y comenzará a realizar las acciones pertinentes.

En primer lugar, el dispositivo se conectará a la red Wi-Fi configurada mediante *WiFiManager*. Este proceso será detallado con mayor profundidad en la siguiente sección. A continuación, se realiza la lectura del estado de las salidas de todos los sensores y se almacena en una variable tipo *float*.

Tras realizar los correspondientes sondeos, se procederá a la impresión de un mensaje por el puerto serial. Dicho mensaje informará de cada uno de

los parámetros recogidos. Posteriormente, los mismos datos serán enviados a los canales de ThingSpeak configurados previamente. Una vez realizadas las acciones anteriores, el dispositivo volverá al estado de bajo consumo.

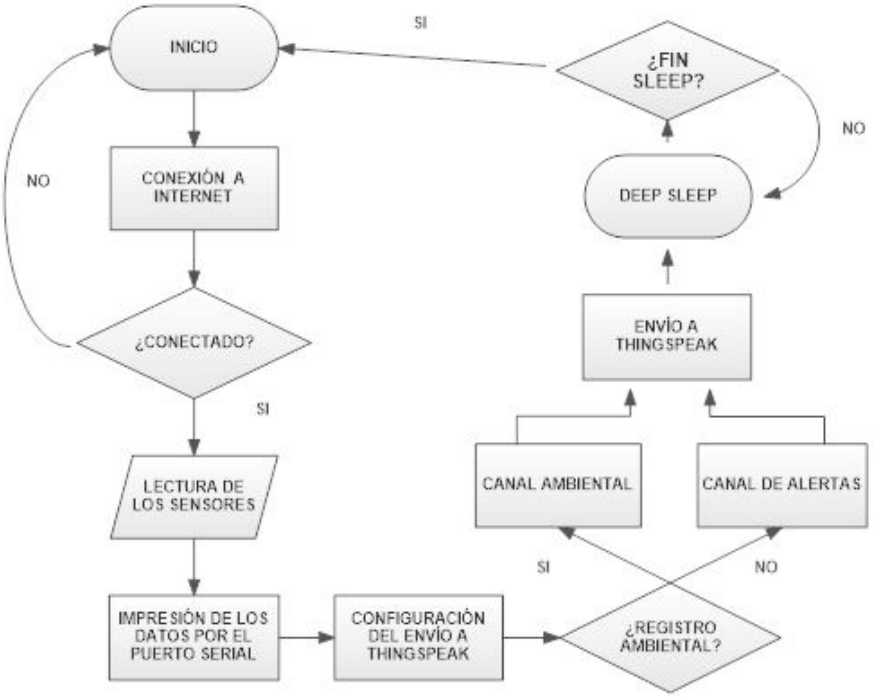

Figura 5.1: Diagrama de flujo.

Antes de finalizar, debemos resaltar lo siguiente. Durante el periodo de instalación y verificación el dispositivo deberá estar el menor tiempo posible en ahorro de consumo. Esto se debe a que durante dicho periodo es necesario realizar una gran cantidad de sondeos para verificar el correcto funcionamiento de todos los elementos. Tras finalizar el proceso anterior, se adaptará el tiempo de *sleep* a las necesidades de cada usuario.

5.2. WiFiManager

WiFiManager es una librería de Arduino que permite gestionar las conexiones Wi-Fi del ESP8266 mediante un portal web accesible desde cualquier dispositivo que soporte una conexión de este tipo, como ordenadores, móviles o tabletas.

El funcionamiento de WiFiManager es muy sencillo. Cuando el micro-

controlador se enciende por primera vez, comienza a funcionar como punto de acceso. Debemos conectarnos con un dispositivo auxiliar a la red Wi-Fi creada por el ESP. Tras ello, nos dirigimos al navegador web y accedemos a la dirección IP 192.168.4.1.

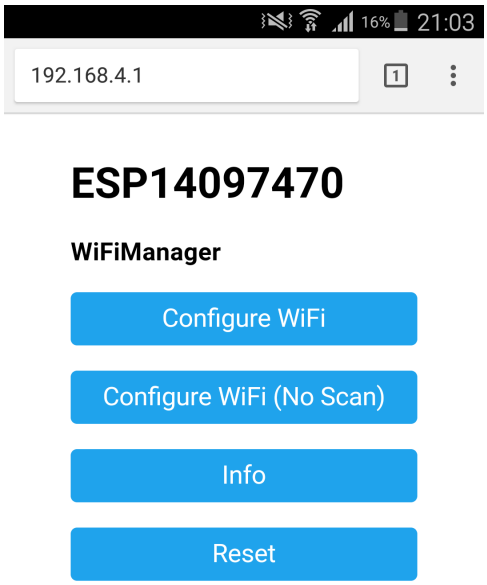

Figura 5.2: Interfaz gráfica WiFiManager.

La imagen 5.2 muestra el panel de configuración que aparece al acceder a la dirección anterior. Se pueden observar las siguientes opciones:

- Configure WiFi. Permite escanear las redes Wi-Fi disponibles y, tras elegir una, conectarnos mediante la introducción de la contraseña.
- Configure WiFi (No Scan). Permite conectarse a una red si realizar un escaneo previo. En este caso será necesario introducir el SSID de la red y la contraseña.
- Info. Proporciona información acerca del módulo Wi-Fi.
- Reset. Reinicia el ESP8266.

Tras realizar una conexión los parámetros indicados será almacenados en la memoria, por lo que no es necesario ninguna configuración adicional en conexiones posteriores. En el caso de que el microcontrolador no detecte la red establecida, volverá a trabajar nuevamente como punto de acceso para permitir la configuración de una nueva conexión.

El uso de WiFiManager permite configurar cada mota de manera independiente sin que sea necesario cargar los parámetros de la red a la que nos queremos conectar durante la fase de compilación. Para valernos de ella, es necesario incluir las librerías *ESP8266WiFi*, *DNSServer*, *ESP8266WebServer* y *WiFiManager*. Tras ello, se inicializa la librería WiFiManager y se incluye la siguiente línea dentro del *SETUP*:

- *wifiManager.autoConnect( );*

Es posible modificar los parámetros del punto de acceso creado por el microcontrolador, pudiendo cambiar el nombre, la contraseña y la dirección donde se puede acceder al panel de configuración. En nuestro caso esto no se llevará a cabo, ya que no aporta ninguna ventaja adicional al dispositivo.

5.3. ThingSpeak

ThingSpeak es una plataforma fundamental para el correcto funcionamiento de nuestro dispositivo. Durante el cuarto capítulo se realizó una pequeña introducción sobre su funcionamiento. En este punto nos centraremos en explicar cómo trabajar con la API (*Application Programming Interface*).

En nuestro caso haremos uso de la librería ThingSpeak.h en Arduino, la cual nos facilita las operaciones de lectura y escritura. En ambas situaciones, resulta indispensable conocer el identificador del canal y la clave del mismo. Dentro de la pestaña API Keys será posible encontrar dicha información, tal y como refleja la imagen 5.3.

A continuación se describen brevemente el funcionamiento de los comandos básicos que incluye la librería anterior.

- ThingSpeak.begin(cliente). Inicializa el cliente que enviará la información recogida a la plataforma.
- ThingSpeak.writeFields(canal, clave). Permite escribir en un canal hasta 8 valores simultáneamente.
- ThingSpeak.setField(campo,valor). Almacena en uno de los 8 campos

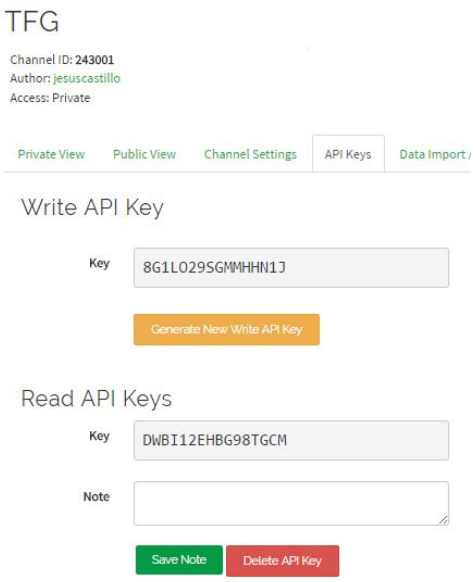

Figura 5.3: Claves de la API.

posibles el valor que deseemos. Se utiliza junto con la expresión anterior.

- ThingSpeak.readFloatField(canal, campo). Su funcionamiento es análogo al anterior. En este caso se usa para leer uno de los 8 campos disponibles en un canal.
- ThingSpeak.readFloatField(canal, campo, clave). Si queremos leer información de un canal privado, será necesario indicar la clave de dicho canal.

Dentro de la misma pestaña, es posible encontrar información acerca de cómo realizar las misma operaciones indicadas anteriormente en caso de no disponer de la librería ThingSpeak.h.

5.4. Flexibilidad

Tal y como se comentó durante el tercer capítulo, la flexibilidad es uno de los requisitos que debe cumplir nuestro dispositivo. El objetivo es presentar un elemento totalmente configurable por el usuario.

Durante el proceso de instalación y verificación, el usuario puede conectar el dispositivo a la red Wi-Fi que desee. Además, es posible configurar los

niveles de disparo de las alertas meteorológicas mediante el potenciómetro que dispone el comparador LM93. También será posible ajustar la sensibilidad del módulo detector de presencia para aumentar o disminuir el rango de detección. El sensor detector de humos también puede ser calibrado según las necesidades requeridas.

Es recomendable realizar las actuaciones anteriores durante el proceso de instalación, ya que así no será necesario volver a actuar sobre los dispositivos. Puede ocurrir que la calibración no se realice de manera adecuada o que varíen las condiciones ambientales (por ejemplo, las alertas relacionadas con la lluvia serán menos frecuentes en verano). En tal caso, cada usuario puede volver a recalibrar los sensores mencionados, con el objetivo de aumentar o disminuir el número de alertas.

Asimismo, es posible elegir el número de sondeos realizados por unidad de tiempo, mediante la variación del tiempo que pasa el dispositivo en modo de bajo consumo. Gracias a ello podrá aumentar la autonomía o, en caso contrario, aumentar la precisión de los resultados.

El capítulo posterior incluye toda la información necesaria para realizar la calibración de los sensores. También se aporta una estimación de la autonomía del dispositivo dependiendo del modo de funcionamiento, acompañada de las pruebas de consumo que permiten calcular dicha estimación.

5.5. Montaje físico

En la siguiente sección se describe el modo en el que se ha llevado a cabo la conexión entre los diferentes dispositivos implicados. Para favorecer la comprensión del lector se aportan imágenes que aportan una mayor nitidez. Durante el desarrollo del epígrafe se irán aportando las decisiones que justifican que el montaje final sea de un modo determinado.

En primer lugar, es necesario conocer el número de salidas de los distintos sensores empleados. En conjunto, contamos con cuatro señales analógicas procedentes del sensor de humos MQ2, la célula fotoeléctrica, el higrómetro y el sensor de lluvia. Por otro lado, tenemos seis salidas digitales aportadas por el sensor detector de humos, el módulo PIR, el sensor de temperatura y humedad DHT11, las sondas de humedad del suelo y lluvia y el termómetro DS18B20.

En este punto debemos repasar el pinout del nuestra placa Wemos Mini D1 para decidir cuál es la mejor forma de conectar las diferentes entradas.

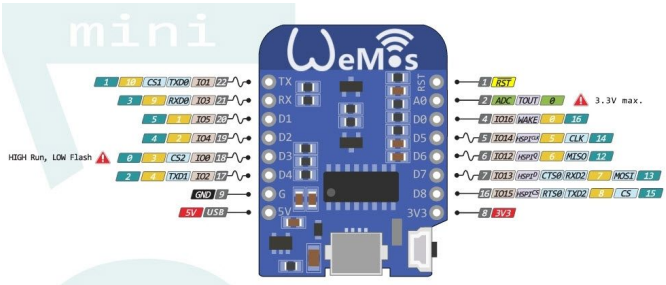

Figura 5.4: Wemos D1 Mini pinout.

Observamos que disponemos de sólo una entrada analógica y nueve entradas digitales. Para leer las cuatro entradas analógicas en el pin A0 usaremos un multiplexor. El uso de este nuevo elemento implica que debemos reservar dos pines para poder seleccionar el canal de entrada correspondiente. En definitiva, para leer los distintos sensores analógicos se emplea el pin analógico A0 y dos pines digitales para controlar el multiplexor.

5.5.1. Alternativas consideradas

A continuación, encontramos varias posibilidades para leer las entradas digitales. La primera de ellas es conectar cada una las salidas digitales a uno de los pines libres de la placa. Esta configuración conlleva que sólo quede libre el pin D0, que además presenta algunas limitaciones frente a los otros pines digitales. La decisión indicada no supone ningún problema para el desarrollo inicial de la mota, pero limita las posibles evoluciones futuras. Con esta configuración sólo se podría añadir una nueva entrada digital o cuatro sensores analógicos, si se hace uso del pin libre para direccionar cuatro canales más del multiplexor.

La segunda de las opciones pasa por hacer uso de un nuevo multiplexor para leer las entradas digitales en un solo pin. A priori, esta solución se antoja mejor ya que solo sería necesario un pin para leer la salida del multiplexor y otros tres para direccionar a cada uno de los 6 canales disponibles. Sin embargo, el hecho de que algunos sensores presenten características especiales provoca que no todos los sensores digitales puedan ser leídos en la misma posición, tal y como ocurriría al elegir este método.

Los dispositivos que presentan estas peculiaridades son el sensor DHT11, que hace uso de una librería especial, y el sensor de temperatura DS18B20, que hace uso del protocolo de comunicación OneWire. Los restantes cuatro sensores no presentan ningún problema. Teniendo esta consideración en cuenta, para direccionar los cuatro canales restantes, serían necesarios dos

pines de la placa, en nuestro caso D3 y D4. Además, son necesarios tres pines digitales más, uno para la conexión con la salida del multiplexor y otras dos más para los dispositivos DHT11 y DS18B20.

5.5.2. Implementación escogida

Tras comparar las alternativas disponibles, la opción elegida para el montaje es la segunda. La principal desventaja frente a la primera es que se aumenta el coste de fabricación sin ser estrictamente necesario. Sin embargo, se presenta una gran ventaja al dejar dos pines de la placa WeMos sin utilizar. Estos podrían ser usados en mejoras posteriores para direccionar nuevos canales en los multiplexores instalados.

En nuestro caso, se utilizará el pin D0 para desperar al microcontrolador tras un tiempo de *sleep*. El pin restante quedará libre para permitir posibles mejoras en un futuro. En cuanto al uso del dispositivo DS18B20, al emplear OneWire en una entrada se nos permite conectar todos los dispositivos que sean necesarios en el mismo lugar, siempre que soporten dicho protocolo.

El cuadro 5.1 resume el empleo de los pines de la placa de desarrollo.

Pin Wemos	**Conexión**
A0	Salida del multiplexor A
D0	Wake up
D1	S1 multiplexor D
D2	S0 multiplexor D
D3	S0 multiplexor A
D4	S0 multiplexor A
D5	Salida del multiplexor D
D6	DHT11
D7	DS18B20
D8	Reserva

Cuadro 5.1: Conexionado de la placa WeMos con los sensores

Tanto los sensores como la placa WeMos Mini D1 son alimentados con una tensión de 5V. Para ello se utilizará la salida a que ofrece la placa a dicha tensión. De este modo, estos dispositivos solo estarán activos cuando se vayan a realizar las medidas.

El conexionado empleado se recoge en la figura 5.5. Cada componente cuenta con una etiqueta con el nombre del sensor para facilitar la comprensión. En algunos módulos sólo se ha representado la conexión hasta el

comparador LM393, para evitar la acumulación de demasiados elementos. A partir de dicho punto cada sensor se une al comparador mediante dos cables.

El esquema se ha realizado gracias a la herramienta *Fritzing*. Se trata de una herramienta gratuita, por lo que a veces la figura empleada no se asemeja mucho al elemento al que representa. A pesar de ello, la figura resultante resulta de gran ayuda para entender el conexionado de los elementos.

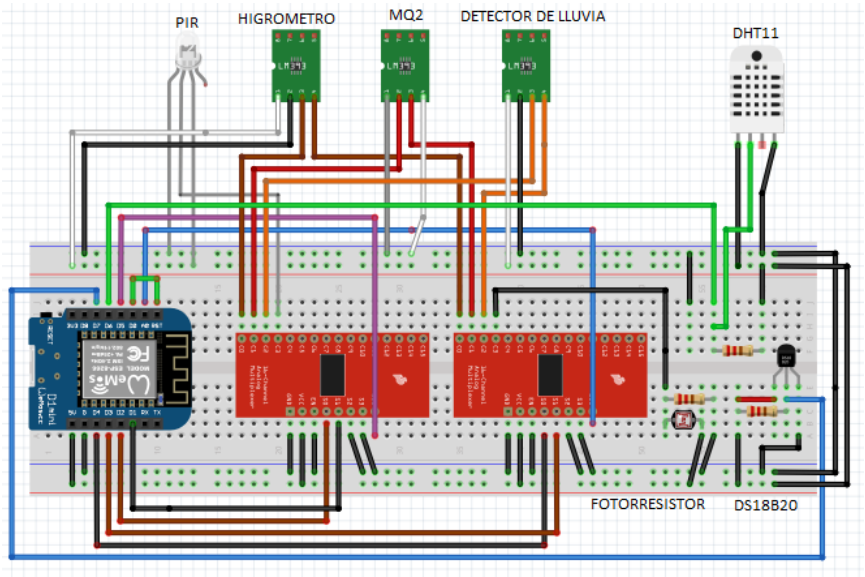

Figura 5.5: Conexionado de los dispositivos.

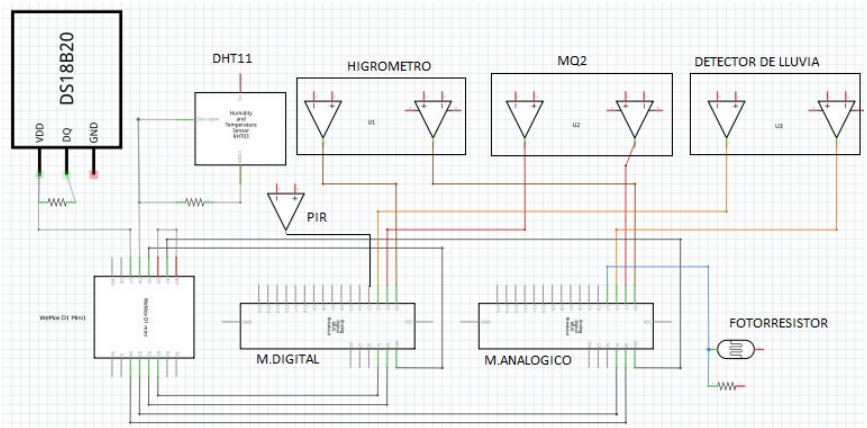

Figura 5.6: Esquema de conexionado.

La figura 5.6 también ha sido realizada gracias a la herramienta anterior. En esta ocasión solo se ha incluido el conexionado correspondiente a los pines de datos, para presentar la información de una forma más nítida.

Por último, se muestra la implementación final del dispositivo en la figura 5.7. Durante el proceso de conexionado se han seguido los esquemas anteriores.

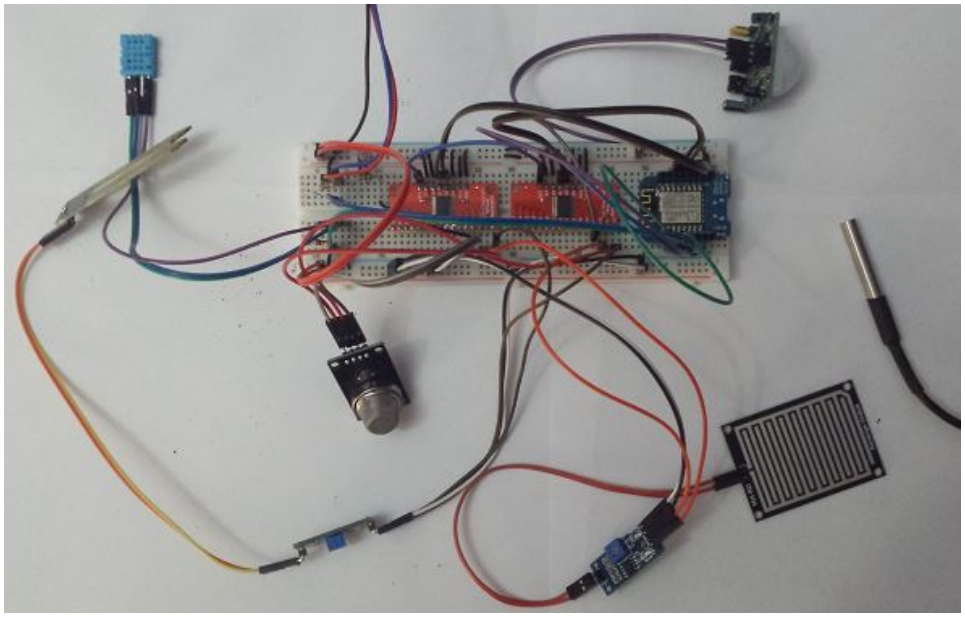

Figura 5.7: Implementación final.

Capítulo 6

Resultados

A lo largo de este capítulo se analizará el rendimiento ofrecido por nuestro dispositivo. En primer lugar será necesario calibrar los sensores para obtener unos registros adecuados. A continuación, comprobaremos cómo afecta el modo de bajo consumo al funcionamiento. Finalmente, se realizará una simulación, mediante la cual observaremos el comportamiento en una situación real. Con todo ello se pretende comprobar que el dispositivo funciona según lo previsto.

6.1. Calibración

Durante el proceso de instalación del dispositivo es imprescindible calibrar adecuadamente los sensores para poder recoger posteriormente registros fiables.

6.1.1. Calibración de alertas

Tres de los sensores presentes en el dispositivo cuentan con un comparador (higrómetro y detectores de gas y lluvia). Tal y como se comentó durante el cuarto capítulo, los módulos que poseen dicho elemento ofrecen dos salidas de datos. La primera de ellas devuelve una lectura analógica que aumenta o disminuye dependiendo del estado del sensor. La segunda varía su valor digital cuando se supera un nivel de alerta preestablecido.

Los módulos anteriores cuentan con un potenciómetro como el de la figura 6.1, que nos permite ajustar el valor a partir del cual el comparador cambiará el estado de la salida digital. Como podemos observar, el dispositi-

vo cuenta con diez muescas alrededor del tornillo que nos permite ajustarlo. Girando en el sentido de las agujas del reloj, cada muesca que avancemos significará que el nivel a partir del cual se genera la alerta se incrementa un 10 %.

Figura 6.1: Potenciómetro.

6.1.2. Calibración de los sensores de humedad

Tanto el higrómetro como el sensor detector de lluvia requieren atención especial en la obtención de la lectura analógica. Durante el proceso de prueba se comprobó que ambos sensores presentaban una particularidad que los distingue de los demás. Ambos sensores devuelven el valor analógico más alto posible cuando no detectan ninguna humedad. Sin embargo, cuando se encuentran sumergidos en agua su valor no llega al mínimo que podemos registrar en el pin A0.

Debido a ello, será necesario determinar cual es el valor mínimo que devuelven dichos sensores para calcular posteriormente la humedad relativa. Tras realizar dicha conversión, la humedad será presentada al usuario en un intervalo de 0 a 100 %.

Como ya hemos comentado, para realizar la calibración se sumergirán por completo ambos sensores en agua. Con ello pretendemos determinar el valor analógico devuelto cuanto la humedad es máxima. En ambos casos se observa un comportamiento similar. Inicialmente ambos módulos ofrecen un nivel de tensión muy bajo, que aumenta continuamente hasta alcanzar un valor estable.

En primer lugar nos centraremos en el higrómetro. Se trata de un dispositivo muy lento en su periodo de estabilización tras la primera puesta en funcionamiento, tal y como se podrá comprobar más tarde en este capítulo. Debido a ello, su calibración es algo compleja.

Tras el proceso de lectura del valor analógico, se realiza una conversión para ofrecer un valor de humedad relativa entre 0 y 100 %. En dicha con-

versión se pasa de un rango de 1024 valores posibles a 614, ya que el sensor no devuelve valores en un rango tan amplio. Se ha elegido este rango tras la realización de la primera simulación de operación. En esta primera simulación se pudo comprobar que el rango establecido en dicha ocasión era demasiado reducido.

El caso del sensor detector de lluvia no es tan extremo, por lo que tendremos una mayor sensibilidad. Tras el proceso de estabilización inicial, el módulo devuelve un valor fijo de 274 cuando está sumergido, dejando un rango de 750 valores posibles. Al igual que en el caso anterior, existe una función para adaptar el valor leído a un valor que se encuentre en un rango entre 0 y 100 %.

6.1.3. Calibración del módulo detector de movimiento

La calibración de este módulo se lleva a cabo mediante potenciómetros. En la figura 6.2 podemos observar los dos potenciómetros de color naranja.

Figura 6.2: Módulo HC-SR501.

El potenciómetro de la izquierda permite ajustar la sensibilidad de detección en un rango entre 3 y 7 metros. El potenciómetro de la derecha permite cambiar el tiempo durante el cual la salida se mantiene en alta tras una detección. Podemos modificar dicho valor desde 5 segundos hasta 5 minutos.

Nuevamente, el valor mínimo se obtiene cuando la rueda se encuentra en la posición más a la izquierda. Girando los potenciómetros hacia la derecha podremos aumentar el valor de ambos factores.

6.1.4. Calibración de los elementos restantes

Los sensores restantes no presentan particularidades especiales y el proceso de calibración se basa generalmente en comprobar que la lectura se ajusta a la realidad.

Durante la realización de las pruebas pertinentes se ha podido comprobar que el sensor detector de humos no necesita calibración. El único aspecto a reseñar es que, en este caso, también se realiza una conversión del valor leído a un nuevo valor, que se encuentre en un rango entre 0 y 100 %.

Los sensores de la familia DHT son calibrados en laboratorio y devuelven una lectura digital, por lo que en principio no sería necesario calibrarlos. Sin embargo, es conveniente comprobar que los valores leídos se corresponden con la realidad haciendo uso de un termómetro y un higrómetro auxiliares, para comprobar que, efectivamente, registran ambos factores adecuadamente. En caso de que el dispositivo se haya descalibrado por algún extraño motivo, se debe realizar el ajuste correspondiente.

En la lectura del fotorresistor también se convierte el registro obtenido, igual que con el resto de lecturas analógicas. El sensor de temperatura ds18b20 tampoco necesita ser calibrado, ya que devuelve la temperatura en grados centígrados mediante una señal digital.

Llegados a este punto, solo queda recordar que para funcionar adecuadamente los tanto el DHT como el fotorresistor necesitan una resistencia de 10 kΩ. El sensor ds18b20 requiere el empleo de una resistencia de 4.7 kΩ. El modo en el que deben ser conectadas se indica en la última figura del capítulo anterior.

6.2. Consumo

Una red de sensores debe garantizar un consumo energético reducido, con el objetivo de aumentar la autonomía de sus nodos. Por norma general, los sensores que adquirimos en el mercado no tienen en cuenta la eficiencia energética. En nuestro caso particular, el coste reducido de los sensores podría provocar que la elección de componentes de un precio menor presente efectos negativos en cuanto al consumo energético. El objetivo de esta sección es analizar cómo afecta cada sensor al consumo de la mota.

Antes de comenzar a realizar las medidas de consumo, es necesario indicar las condiciones bajo las cuales se van a realizar las mismas. Durante las tandas de pruebas, la placa Wemos D1 Mini será alimentada mediante

USB. De este modo nos aseguramos que tendremos siempre una tensión fija de entrada. Para obtener los resultados, se alimenta la placa con un USB tester como el de la figura 6.3.

Figura 6.3: USB tester.

Este dispositivo nos proporciona los valores de corriente y voltaje instantáneos, así como el tiempo transcurrido en cada tanta de pruebas y el consumo energético acumulado. La sensibilidad del dispositivo es de un minuto en el cronometro, 1 mAh en el medidor de consumo y 0.01 A y 0.01 V en los valores de corriente y voltaje instantáneos.

6.2.1. Caracterización de los sensores

En primer lugar, es necesario caracterizar el consumo, de manera individual, de cada uno de los sensores empleados. La prueba consistirá en medir el tiempo que tarda cada elemento en consumir 3 mAh. Para ello, se hará uso de un cronómetro auxiliar con mayor sensibilidad que el USB *Tester*. Posteriormente, los datos recogidos serán tratados, de modo que será posible observar el consumo por minuto y también por hora.

Para comenzar la prueba se ejecutará un *sketch* vacío, lo cual nos permitirá ver el consumo de la placa mientras no se realiza ninguna operación. A continuación se irán añadiendo de forma individual cada uno de los sensores. Dichos sensores serán alimentados mediante el pin de 5V de la placa WeMos. La lectura del estado de cada sensor se realizará cada 5 segundos.

Los datos obtenidos durante las pruebas se presentan en la tabla 6.1. En la primera columna se indica el sensor que se conecta en cada caso junto a la WeMos. El las siguiente columna, se muestra el tiempo transcurrido hasta que el consumo alcanza los 3 mAh y, por último, el consumo por minuto y hora de cada una de las configuraciones.

Elemento	Tiempo (min:s)	mAh (1min)	mAh (1h)
WeMos	3:37	0.839	49.769
PIR	3:32	0.849	50.943
Fotorresistor	3:34	0.841	50.467
DHT11	3:36	0.833	50
Suelo	3:33	0.845	50.704
LLuvia	3:34	0.841	50.467
Humos	0:59	3.051	183.051
Ds18b20	3:35	0.837	50.232

Cuadro 6.1: Resultados de las primeras pruebas de consumo

Gracias a los registros obtenidos seremos capaces de cuantificar el incremento de consumo que provoca cada uno de los sensores. Para ello, calcularemos la diferencia de consumo entre la primera medida, en la que solo se conecta la placa WeMos, y el resto. El incremento de consumo por cada hora de funcionamiento se incluye el el cuadro 6.2.

Elemento	Incremento (mAh)
PIR	1.174
Fotorresistor	0.698
DHT11	0.231
Suelo	0.935
LLuvia	0.698
Humos	133.282
Ds18b20	0.463

Cuadro 6.2: Consumo individual de los sensores

Tras realizar las pruebas, podemos concluir que el consumo de los sensores es despreciable frente al consumo energético de la placa. El consumo de los módulos es muy similar en la mayoría de los casos y no parece aportar efectos demasiado perjudiciales para la autonomía del dispositivo final.

El sensor de humos MQ2 merece una mención aparte. Como hemos podido observar, al alimentar el dispositivo el consumo de energía se aumenta casi al cuádruple. Si recordamos las especificaciones del elemento en cuestión, podremos encontrar los motivos de este incremento. Para funcionar correctamente el módulo necesita mantener una temperatura elevada, lo cual exige un gran aporte energético.

Debemos recordar que uno de los principales objetivos que debe cumplir el diseño de la mota es ser eficiente en términos de energía. Mantenerlo alimentado supone un gasto de 4393.224 mAh diarios, lo cual condiciona

enormemente el cumplimiento de dicho objetivo.

6.2.2. Recogida de datos

En esta ocasión se pretende conocer cómo afecta al consumo el hecho de que los sensores realicen más o menos medidas en un tiempo determinado. La prueba consiste en medir el tiempo que tarda la mota en consumir 5 mAh. El número de medidas a tomar se varía aumentando el valor del retardo entre muestras.

Dicha prueba se realiza en dos ocasiones distintas, la primera de ellas sin el sensor MQ2 y la segunda con todos los sensores conectados. Con ello, se pretende que la distorsión que introduce este dispositivo en el consumo no afecte a las decisiones que se tomarán tras conocer los datos. Los resultados de la primera tanda de pruebas se muestran en la tabla 6.3.

Delay(s)	**Sondeos/h**	**Tiempo(min:s)**	**mAh(1min)**	**mAh(1h)**
5	720	5:09	0.971	58.252
25	144	5:20	0.938	56.25
45	80	5:16	0.949	56.962
65	55	5:23	0.929	55.728

Cuadro 6.3: Influencia del tiempo entre muestras

Antes de analizar cómo afecta el número de sondeos por unidad de tiempo al consumo, nos detendremos un instante para comentar la primera medida. En este primer caso, las medidas se toman con la misma frecuencia que en la prueba inicial, pero el consumo es mayor. El incremento se debe a que en esta ocasión se están alimentando seis sensores simultáneamente. A ello hay que sumar los dos multiplexores empleados para trabajar con todos los sensores al mismo tiempo.

Tras este pequeño inciso, comentaremos los resultados de la segunda tanda de testeos. En primer lugar se observa una caída del consumo reseñable cuando pasamos de tomar muestras cada 5 segundos a 25 segundos. Resulta evidente que aumentar el tiempo entre sondeos ayuda a reducir el consumo energético, como se puede ver si comparamos la primera muestra con cualquiera de las demás. No obstante, para observar esta dependencia debemos modificar sustancialmente el número de sondeos.

En ocasiones donde el número de muestras cambia poco respecto de una prueba anterior, aparecen otros factores con mayor influencia en los resultados. Podemos ver que el consumo cuando se toman 80 muestras/hora es mayor que el medido al registrar 144 muestras/hora. Los sensores digitales

modifican el estado de su salida cuando realizan una detección. Asimismo algunos de ellos poseen un led que iluminan al realizar este cambio. Dichos leds se mantienen iluminados ininterrumpidamente siempre que se supere un umbral.

A la vista de los resultados, se podría decir que el estado de los sensores también influye en el consumo. Manteniendo el número de muestras, el consumo puede variar dependiendo de cómo reaccionen los sensores ante las condiciones atmosféricas.

En definitiva, esta segunda tanda de pruebas ha ayudado a constatar lo siguiente. Podemos reducir el consumo de nuestra mota si nos limitamos a tomar muestras cada cierto tiempo. El estado ambiental tiene influencia en el consumo, ya que afecta al estado de los leds. Para corregir este efecto, se recomienda eliminar los leds de los sensores, ya que en nuestro caso las alertas luminosas no añaden ninguna funcionalidad extra, debido a que el dispositivo trabajará de forma remota.

A continuación incluiremos el medidor de humos MQ2 a la mota y lo alimentaremos mediante la salida de la placa a 5 V. Con ello deseamos cuantificar la influencia de este elemento, volviendo a medir el tiempo que se tarda en consumir 5 mAh. Se prevé un gran aumento del consumo. Los resultados se incluyen en el cuadro 6.4.

Delay(s)	**Sondeos/h**	**Tiempo(min:s)**	**mAh(1min)**	**mAh(1h)**
5	720	1:36	3.125	187.5
25	144	1:37	3.093	185.567
45	80	1:35	3.158	189.474
65	55	1:35	3.158	189.474

Cuadro 6.4: Influencia del tiempo entre muestras con sensor MQ2 incluido

En esta ocasión, la tanda de pruebas carece de toda utilidad. Los resultados obtenidos son insuficientes para obtener conclusiones. Vuelve a quedar patente el aumento de consumo producido por el sensor de humos. Siempre que este sensor se encuentre en funcionamiento, modificar el número de sondeos no tendrá ninguna utilidad, ya que el ahorro de consumo es despreciable frente al consumo del MQ2.

En definitiva, queda claro que en este caso el consumo de la mota en esta situación será constante, ya que todas las consideraciones tenidas en cuenta en la anterior tanda de medidas tienen una influencia despreciable frente al consumo del MQ2.

6.2.3. Deep Sleep

El microcontrolador ESP8266 dispone de tres modos de ahorro de energía. Con el objetivo de consumir el mínimo posible, emplearemos el modo más agresivo. Para ello es necesario llamar a la función ESP.deepSleep(). Inmediatamente se activa este modo, lo cual provoca que se apague la conexión WIFI y la conexión de datos. Sólo el módulo RTC sigue funcionando, ya que es el responsable de que el módulo despierte.

Con la siguiente prueba pretendemos medir el consumo de la mota cuando el microcontrolador se encuentra en modo *Deep Sleep*. Para ello, pondremos la placa en modo *sleep* y analizaremos el consumo mientras tanto. En primer lugar observaremos el ahorro de consumo de la mota con todos los sensores juntos. Para ello, mediremos el tiempo que tarda en consumir 5 mAh. Los resultado se muestran en el cuadro 6.5.

Tiempo(min:s)	**mAh(1min)**	**mAh(1h)**
2:32	1.974	118.421

Cuadro 6.5: Consumo en modo Deep Sleep

Si comparamos la tabla con el cuadro 6.4 observamos como el consumo se reduce en torno a 70 mAh por cada hora de funcionamiento, lo que supone un ahorro de energía cercano al 40 % cuando se usa el modo *Deep Sleep*. Es interesante ver como el consumo no se reduce a cero, lo cual nos indica que los sensores se siguen alimentando.

Al medir el consumo del dispositivo sin el sensor de humos, volvemos a encontrar problemas con la sensibilidad del USB Tester. Cuando el microcontrolador entra en modo *Deep Sleep*, la corriente que consume la mota es inferior a 0.01 A. Por tanto, nos es imposible medir el consumo con dicho dispositivo.

Teniendo en cuenta los resultados anteriores, podemos extraer las siguientes conclusiones. Cuando el dispositivo trabaja en modo *Deep Sleep*, sin el sensor de humos, el consumo dependerá del tiempo que se encuentre fuera de dicho modo para realizar los sondeos. En caso de contar con el sensor de humos el consumo será siempre superior a 118 mAh. Este valor se incrementará dependiendo del tiempo que el dispositivo se encuentre fuera del modo de ahorro de energía.

6.2.4. Estimación de la autonomía

Gracias a los cálculos anteriores podemos realizar una estimación de la autonomía de la cual dispondrá el dispositivo implementado. Obviamente el tiempo dependerá de la capacidad de la batería con que se alimente el dispositivo, además del propio funcionamiento del mismo.

Para calcular la estimación es necesario tener en cuenta el tiempo que pasa el dispositivo en *sleep* y el ocupa en la ejecución del código. Gracias a la función *millis()* se ha podido medir el tiempo de ejecución del código. Este tiempo oscila dependiendo principalmente del tiempo que se tarde en establecer la conexión Wi-Fi. El valor máximo de ejecución se encuentra en torno a 14 segundos.

Se ha decidido realizar los cálculos para el caso en el cual el dispositivo se alimente mediante una *power bank* de 10000 mAh. En la tabla 6.6 se muestra la estimación del consumo del dispositivo cuando cuenta con todos los sensores incorporados.

En primer lugar se muestran en número de sondeos que realizará el dispositivo cada hora. A continuación se incluye el consumo que se realiza durante el periodo de sondeos acompañado del consumo durante el tiempo en *sleep*. En la siguiente columna se indica el consumo total del dispositivo por hora, resultado de la suma de los anteriores. Por último se realiza la estimación del tiempo de funcionamiento en horas.

Sondeos por hora	**Consumo sondeos(mAh)**	**Consumo *sleep*(mAh)**	**Consumo total(mAh)**	**Autonomía (h)**
1	0.735	117.96	118.695	84.249
2	1.47	117.499	118.969	84.055
3	2.205	117.039	119.244	83.862
6	4.41	115.657	120.067	83.287
12	8.82	112.894	121.714	82.159
60	44.1	90.784	407.236	24.556

Cuadro 6.6: Estimación de la autonomía.

Como se puede ver en el cuadro 6.6, la autonomía del dispositivo es muy reducida ya que en el mejor caso no alcanza los 4 días de funcionamiento. Esta autonomía será incluso menor, ya que en usa situación real cualquier batería de estas características dejará de suministrar la corriente necesaria para mantener el dispositivo activo antes de agotar su carga.

A continuación realizaremos un nuevo cálculo para observar la autonomía sin el sensor de humos. Para realizar el cálculo se tendrán en cuenta las mis-

mas consideraciones que en la ocasión anterior. Los resultados se muestran en la tabla 6.7. En la tabla se observa el consumo por hora debido al tiempo de ejecución del código, el consumo estimado en *Deep Sleep*, el consumo total y a continuación la estimación de la autonomía.

Sondeos por hora	**Consumo sondeos(mAh)**	**Consumo *sleep*(mAh)**	**Consumo total(mAh)**	**Autonomía (días)**
1	0.221	0.0199	0.2409	1729.63
2	0.443	0.0198	0.4628	900.32
3	0.665	0.0197	0.6847	608.54
6	1.33	0.0195	1.3495	308.76
12	2.66	0.019	2.679	155.53
60	13.3	0.0153	13.3153	31.29

Cuadro 6.7: Estimación de la autonomía sin MQ2.

En esta segunda ocasión se ha empleado el consumo teórico en *sleep*, ya que durante la realización de las pruebas de consumo fue imposible obtener dicho valor. Se ha tenido en cuenta que el microcontrolador ESP8266 consume en torno a 20 μA en dicho modo, lo cual supone 20 μA por hora en *sleep*. Dicha información puede ser consultada en la hoja de características que informa de los modos de ahorro de consumo del ESP8266 [22].

La estimación recogida en el cuadro 6.7 nos sirve para observar el gran incremento de autonomía que se puede lograr al eliminar el sensor de humos.

6.3. Pruebas de funcionamiento

Tras configurar el dispositivo y caracterizar su funcionamiento, sólo queda realizar una prueba en la que el dispositivo trabaje como lo haría una vez instalado en un cultivo.

Para ello, el dispositivo será instalado en el balcón de una vivienda, ya que así se podrá observar con mayor claridad la evolución de las condiciones ambientales. Esto implica que se registrarán un mayor número de detecciones de presencia de las que se podrían dar en una situación real, especialmente durante el día.

6.3.1. Primera prueba

Durante la realización de la prueba, el dispositivo será alimentado mediante una power bank cuya capacidad es de 6600 mAh. Se activará de for-

ma predeterminada el modo *deep sleep*. En intervalos de 5 minutos, la placa WeMos abandonará dicho modo para realizar los sondeos correspondientes y subir los datos a los canales de ThingSpeak configurados previamente.

La primera prueba nos sirve para volver al constatar lo que ya se pudo observar durante la sección anterior, el consumo del sensor de humos limita enormemente la autonomía del dispositivo. Durante el desarrollo de la misma, el dispositivo no alcanzó las 12 horas de funcionamiento. Esto se debe a que, antes de agotar su carga por completo, la *power bank* es incapaz de suministrar los 5 V necesarios para la alimentación del dispositivo. A ello hay que sumar que la batería comentada dispone de 4 leds que se iluminan para indicar la carga restante, lo cual afecta de manera importante a la autonomía.

Figura 6.4: Factores ambientales. Primera prueba.

En cuanto a los resultados, podemos considerar que son satisfactorios. En la figura 6.4 podemos encontrar los primeros factores ambientales recogidos por el dispositivo. Se puede observar que el registro de los valores se ha realizado adecuadamente.

Durante el apartado de calibración se comentó que el higrómetro necesitaba un tiempo de estabilización para comenzar a registrar valores confiables. Gracias a esta prueba podemos observar dicho proceso de estabilización. En un suelo húmedo, el sensor no detecta la presencia de humedad al comenzar la prueba. Conforme avanza la prueba, podemos observar como se registra dicho factor adecuadamente.

En la figura 6.5 se recogen el resto de factores ambientales. En esta

nueva imagen observamos un comportamiento anómalo en la gráfica de la luz solar. Los picos de luminosidad iniciales se deben a que el dispositivo comenzó a enviar datos a la plataforma mientras que aún no había concluido la instalación, por lo que los registros no son del todo exactos. También resulta curioso que la luminosidad nunca llegue a cero durante la noche, efecto provocado a causa de la iluminación que recibía el dispositivo por parte del alumbrado público.

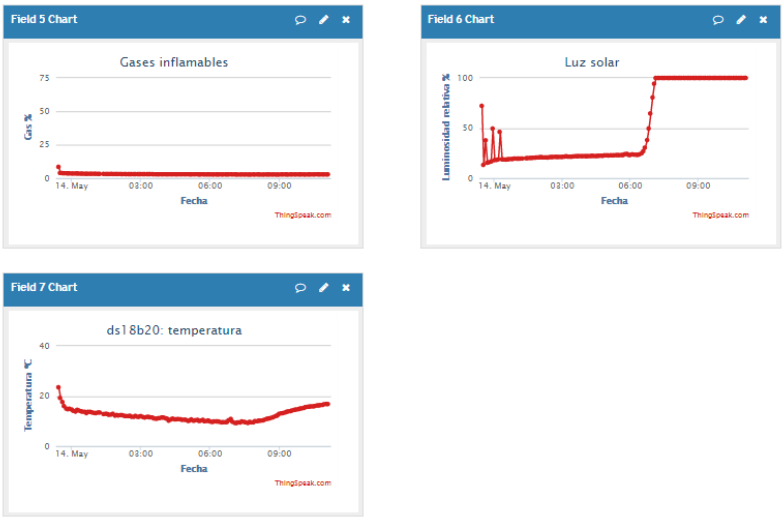

Figura 6.5: Factores ambientales. Primera prueba.

En cuanto al resto de factores monitorizados, no se aprecian registros que puedan indicar que el dispositivo no funciona adecuadamente. Los sensores detectores de lluvia y gases no presentan ningún registro significativo ya que no se dieron las condiciones necesarias para que ambos reaccionasen durante el desarrollo de la prueba.

Las alertas configuradas en algunos de los elementos se muestran en la imagen 6.6. Nuevamente, los sensores de gas y lluvia no presentan cambios durante la prueba. Observamos que el higrómetro activa la alerta tras transcurrir el tiempo de estabilización comentado anteriormente.

En cuanto a las alertas por detección de presencia, se detectan una mayor cantidad de lo que se podría esperar en un principio. Dicha prueba se realizó durante un día ventoso lo cual hace pensar que las alertas recogidas se deben al movimiento de la propia planta a causa del aire.

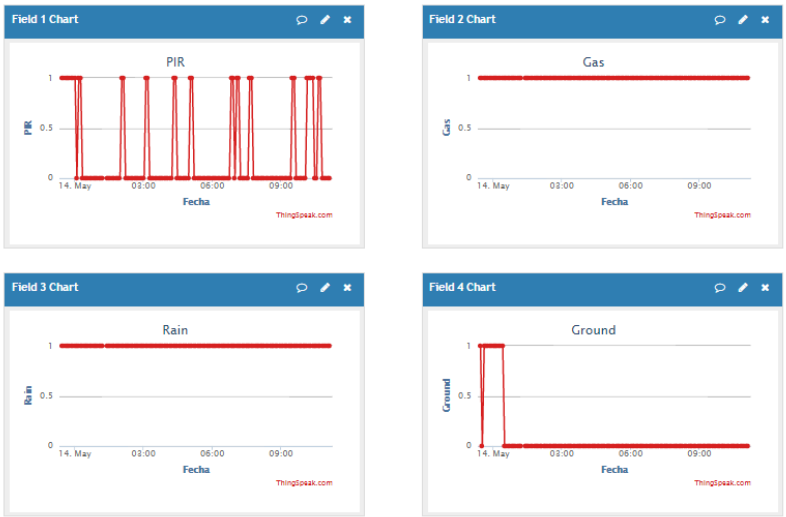

Figura 6.6: Alertas. Primera prueba.

6.3.2. Segunda prueba

Para obtener una mejor visión del funcionamiento del dispositivo, se realizará una segunda prueba en la cual se mantendrá en activo durante un periodo superior a 24 horas. En este caso la alimentación se realizará mediante una fuente de corriente continua, para evitar que deje de funcionar antes de lo deseado.

Al término de la segunda prueba, observamos como los datos son aún más precisos, ya que se han realizado modificaciones respecto a la prueba anterior. Tal y como se comentó en la sección de calibración, antes de esta segunda prueba el higrómetro fue recalibrado ampliando su rango. Como podemos ver en la figura 6.7, los resultados obtenidos mejoran significativamente y el valor de la humedad en el suelo no sufre variaciones drásticas como ocurría en la prueba inicial.

Al aumentar el tiempo de funcionamiento es posible observar con mayor claridad la evolución de la temperatura y la humedad durante el día. Como vemos en la figura 6.7, al final se la prueba cambia el registro de detección de lluvia. Esto se debe a que, para corroborar el funcionamiento del módulo, se vertió agua de manera artificial sobre el detector.

En la figura 6.8 se muestran el resto de factores ambientales. Se puede ver como en este caso la iluminación también se ve afectada por el tendido público durante la noche. En esta ocasión tampoco se han detectado gases inflamables.

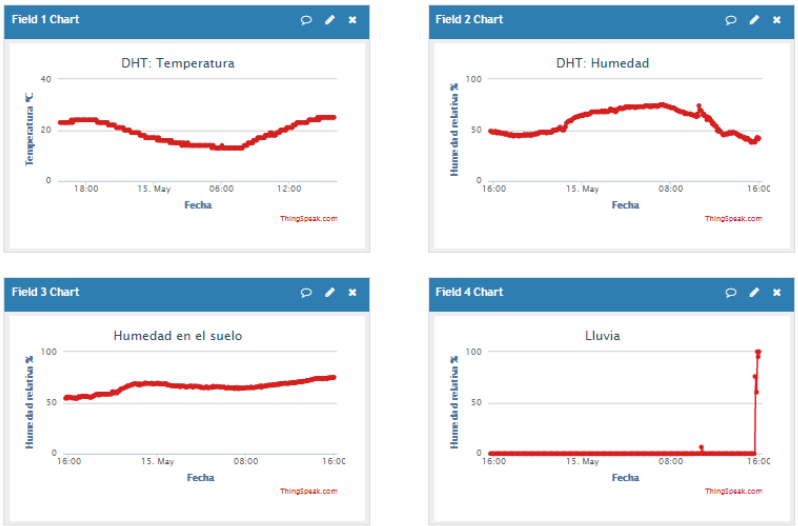

Figura 6.7: Factores ambientales. Segunda prueba.

Figura 6.8: Factores ambientales. Segunda prueba.

Para terminar de comentar la segunda prueba, se incluyen los valores de las alertas en la figura 6.9. En este caso, se observa como las alertas por detección de presencia sólo se realizan durante el día, donde hay una mayor actividad en dicha zona de la casa. Gracias a ello, podemos confirmar que las detecciones realizadas durante la noche en la primera prueba se debían al movimiento de la planta causado por el aire. Esto deberá ser tenido en cuenta durante la instalación del dispositivo para evitar falsos positivos.

Dentro de las alertas por lluvia encontramos dos cambios. El segundo de ellos se debe al vertido de agua comentado con anterioridad. En cambio, la primera detección resulta algo extraña. Si observamos los valores aledaños a esa muestra, observamos que no existe presencia de lluvia. Volviendo a la figura 6.7 apreciamos una detección mínima de agua sobre el módulo. Ya que las condiciones climáticas no apuntan a la presencia de lluvia, dicha detección habría sido causada por algún factor externo que no podemos determinar.

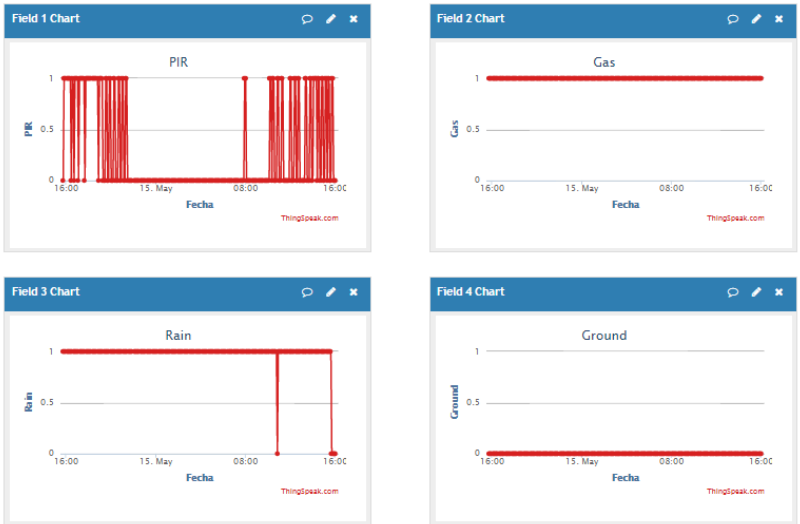

Figura 6.9: Alertas. Segunda prueba.

6.3.3. Tercera prueba

Llegados a este punto, el correcto funcionamiento del dispositivo ha sido registrado adecuadamente. Por ello, el objetivo principal de la prueba actual no es observar la monitorización de las condiciones climáticas. En esta ocasión pretendemos demostrar el aumento de autonomía producido al eliminar el sensor de humos.

Con tal fin, el sensor MQ2 será eliminado antes de comenzar la prueba. El dispositivo se alimentará nuevamente con la *power bank* anterior, cuya capacidad es de 6600 mAh. Con el objetivo de que la prueba no se prolongue en exceso, se realizarán dos sondeos de todos los sensores cada minuto. Los resultados obtenidos se reflejan a continuación.

La prueba comenzó justo después de regar la planta donde fue colocado el dispositivo. Como se observa en la figura 6.10, la humedad del suelo

comienza con un valor muy elevado. Conforme se seca la tierra, se observa cómo disminuye dicho factor. En cuanto a los valores de temperatura y humedad ambientales captados por el DHT11, observamos pequeñas fluctuaciones debidas al corto espacio entre muestras. Gracias a un intervalo entre sondeos tan reducido, se pueden registrar las pequeñas variaciones que se producen en el entorno.

Figura 6.10: Factores ambientales. Tercera prueba.

La figura 6.11 no presenta demasiados factores dignos de mención. Los valores de iluminación han fluctuado como en ocasiones anteriores y el sensor de humos no se encontraba conectado. En esta ocasión la sonda que mide la temperatura del suelo fue colocada a una mayor profundidad. Debido a ello, la variación de temperatura es menor, así como los valores registrados.

Por último, se muestran las alertas en la imagen 6.12. No se observan registros anómalos ya que en esta ocasión los factores ambientales que disparan las alertas se mantuvieron estables.

El principal objetivo de la prueba era constatar el aumento de autonomía conseguido al eliminar el sensor de humos. La realización de la misma se prolongó durante un periodo de 32 horas. Al igual que en la primera prueba, la autonomía se vio afectada por los leds que incorpora la power bank, así como por los propios leds que incorporan los sensores, ya que algunos de ellos se mantuvieron iluminados durante todo el proceso. También es necesario indicar que la eficiencia de la batería no es del 100 %, por lo que en realidad contamos con una capacidad de alimentación inferior a la indicada.

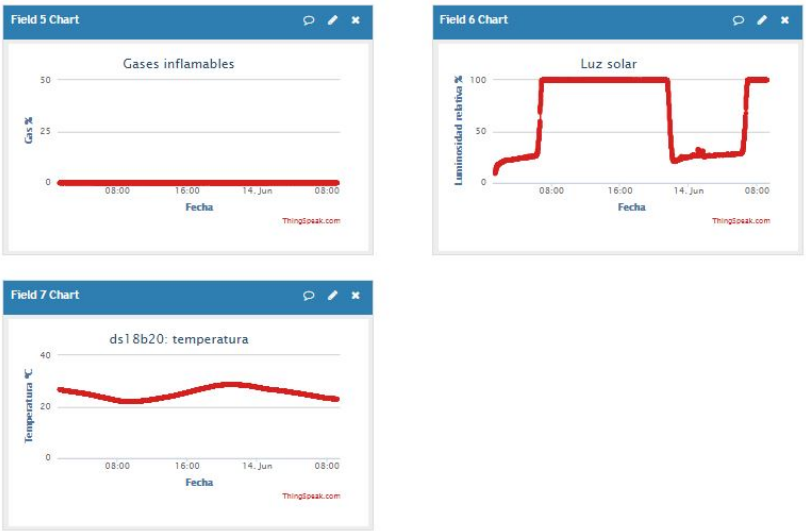

Figura 6.11: Factores ambientales. Tercera prueba.

Figura 6.12: Alertas. Tercera prueba.

6.3.4. Prueba auxiliar: sensor de humos

Durante la realización de las pruebas anteriores se ha podido comprobar correcto funcionamiento de todos los sensores excepto uno, el sensor de humos. Antes de analizar los resultados se realizará una prueba en la que quede constatado que dicho elemento funciona adecuadamente.

Para ello se alimentará el dispositivo como en la segunda prueba y se hará uso de un mechero para liberar gases inflamables cerca del sensor. Al activar el mechero podremos observar cómo aumenta el nivel de gas y se dispara la alerta digital. En esta ocasión visualizaremos los datos a través del puerto serial, ya que el envío de datos a *ThingSpeak* ya ha sido verificado con anterioridad.

```
****************************************** ****************************************** ******************************************
**         WEATHER CONDITIONS           ** **         WEATHER CONDITIONS           ** **         WEATHER CONDITIONS           **
****************************************** ****************************************** ******************************************
****************************************** ****************************************** ******************************************
**            -- RAIN SOIL --           ** **            -- RAIN SOIL --           ** **            -- RAIN SOIL --           **
** Percent: 0.00                        ** ** Percent: 0.00                        ** ** Percent: 0.00                        **
**                -- MQ2 --             ** **                -- MQ2 --             ** **                -- MQ2 --             **
** Gas percent: 5.18                    **  ** Gas percent: 39.94                   ** ** GAS DETECTED                         **
**        -- PRESENCE DETECTOR --       ** **        -- PRESENCE DETECTOR --       ** ** Gas percent: 100.00                   *
** PRESENCE DETECTED                    ** ** PRESENCE DETECTED                    ** **        -- PRESENCE DETECTOR --        **
**         -- GROUND HUMIDITY --        ** **         -- GROUND HUMIDITY --        ** ** PRESENCE DETECTED                    **
** Percent: 0.00                       **  ** Percent: 0.00                        ** **          -- GROUND HUMIDITY --        **
**          -- PHOTORESISTOR --         ** **          -- PHOTORESISTOR --         ** ** Percent: 0.00                       **
** Luminosity: 100.00 percent            **** Luminosity: 100.00 percent            *** **          -- PHOTORESISTOR --         **
**               -- DHT11 --            ** **               -- DHT11 --            ** ** Luminosity: 100.00 percent            *
** Temperature: 28.00 Celsius           ** ** Temperature: 28.00 Celsius           ** **               -- DHT11 --             **
** Humidity: 43.00 percent              ** ** Humidity: 45.00 percent              ** ** Temperature: 29.00 Celsius            **
**            -- TEMPERATURE --         ** **            -- TEMPERATURE --         ** ** Humidity: 43.00 percent               **
** Temperature: 29.81 Celsius           ** ** Temperature: 30.00 Celsius           ** **            -- TEMPERATURE --          **
****************************************** ****************************************** ** Temperature: 30.00 Celsius             **
                                                                                      ******************************************
```

Figura 6.13: Prueba de detección de gas.

En la figura 6.13 se pueden observar tres mensajes distintos que muestran la variación del gas. Dichos mensajes son mostrados por serial cada vez que el dispositivo realiza un sondeo de cada dispositivo. En la figura indicada se observa como aumenta la proporción de gas cuanto más tiempo pasa el mechero liberando gas. En la última captura se observa que aparece un nuevo mensaje junto al porcentaje, que indica que se ha activado la alerta digital.

6.3.5. Análisis de los resultados

Llegados a este punto es necesario evaluar los resultados obtenidos durante las pruebas. A la vista de los mismos, podemos afirmar que el dispositivo implementado cumple con los requisitos planteados al inicio del proyecto.

Como se ha podido comprobar, no existen grandes variaciones entre registros consecutivos. En un escenario real debemos tener en cuenta para qué queremos utilizar el dispositivo. Si nos centramos en monitorizar las condiciones en cultivos en el exterior, sería recomendable establecer un tiempo de *sleep* de entre 15 y 20 minutos. Este intervalo será más que suficiente para registrar adecuadamente las fluctuaciones.

En caso de implantar el dispositivo dentro de un invernadero es recomendable mantener el tiempo de *sleep* en 5 minutos. Dentro de estos entornos es necesario mantener unas condiciones muy concretas, por lo cual deberemos estar muy atentos a las variaciones. Esto es aún más importante si se usan

los datos para activar sistemas complementarios, como pueda ser un circuito de riego.

Para observar una diferencia notable entre los sensores de temperatura se debe colocar el sensor terrestre a una profundidad considerable. Cuanto más profundo se coloque, menor será la temperatura y también la fluctuación. Será el agricultor el encargado de elegir la profundidad a la que coloca dicha sonda, dependiendo de las condiciones que desea monitorizar.

Comparando la primera y la tercera prueba, debemos estudiar la inclusión o no del sensor de humos. En este punto debemos tener en cuenta la influencia de diversos factores que afectan al rendimiento, como son los leds que disponen tanto la batería como el resto de sensores y el hecho de que la capacidad real de la *power bank* es inferior a los 6600 mAh indicados.

En la primera prueba, la autonomía del dispositivo no alcanzó las 12 horas de funcionamiento. Dicha duración puede ser aumentada modificando el tiempo entre sondeos, aunque en esta ocasión esto no suponga una diferencia reseñable. En la segunda prueba se realizaban sondeos con una frecuencia 10 veces superior. A pesar de ello, la autonomía ascendió 20 horas, lo cual es muy reseñable. En esta segunda ocasión sí se lograría un aumento muy significativo al establecer un tiempo entre sondeos superior.Dicho tiempo podría incrementarse un más si se elimina la influencia de los leds y se incluye una fuente de alimentación con mayor eficiencia.

Tras analizar los resultados se presenta una alternativa diferente. La solución final ofrecida dispondría de dos modelos distintos. El primero sería instalado junto al punto de acceso de la red Wi-Fi que ofrece conectividad a los nodos. Ya que en este punto disponemos de corriente continua, se utilizará dicha forma de alimentación. Este modelo contaría con el sensor de humos, porque que en esta ocasión no supondría ningún problema. Los dispositivos repartidos por el cultivo carecerán de sensor de humos, con lo cual será posible mejorar notablemente la autonomía.

Para finalizar, se desaconseja el uso de una *power bank* como fuente de alimentación del dispositivo. Dichos elementos han sido desarrollados para realizar una carga rápida de un terminal como pueda ser un móvil. Debido a ello, no se suele tener en cuenta la influencia que puedan tener los elementos luminosos indicadores de la batería. Estos elementos aportan efectos muy perjudiciales a la autonomía del dispositivo desarrollado. Por tanto, se recomienda emplear una batería de litio.

Capítulo 7

Conclusiones y líneas futuras

Antes de finalizar, será necesario comentar las conclusiones obtenidas tras la realización del trabajo. En este capítulo se pretende transmitir al lector la experiencia obtenida durante la elaboración del proyecto. Con tal objetivo, se comentarán las líneas principales del desarrollo del dispositivo de monitorización.

Por último, se describen las principales líneas de actuación que se pueden seguir con el dispositivo implementado en fases posteriores. Esto ayudará a mejorar el dispositivo y combatir su obsolescencia.

7.1. Conclusiones

Tras realizar finalizar el trabajo, podemos concluir que es posible implementar un dispositivo IoT de coste reducido sin que ello afecte a las prestaciones del mismo. Se ha comprobado que los registros obtenidos gracias a dicho instrumento son completamente fiables.

Ha quedado patente que la placa de desarrollo WeMos D1 Mini es una alternativa excelente como base sobre la cual desarrollar un dispositivo *Internet of Things*. Sus cualidades fueron analizadas en el cuarto capítulo de la memoria, destacando frente a las demás alternativas. Más tarde se comprobó su escaso consumo energético, confirmando la elección realizada.

Gracias a las pruebas realizadas se ha demostrado la influencia de los sensores en la autonomía del dispositivo. Se ha podido comprobar que la existencia de un sensor de humos hace imposible la obtención de un tiempo de funcionamiento prolongado. El elevado consumo de este elemento impide

que la autonomía alcance un periodo aceptable.

En esta misma línea, se ha demostrado la influencia de las condiciones ambientales en el consumo del dispositivo, por lo cual se ha recomendado la eliminación de todos los elementos luminosos que forman parte de él. El periodo entre sondeos también afecta al tiempo de funcionamiento hasta que se agota la fuente de alimentación.

El modo en el que se coloque el dispositivo es fundamental a la hora de obtener unos registros adecuados. El posicionamiento y calibración de los sensores influye en las muestras recogidas. Gracias a la flexibilidad que ofrece el dispositivo, cada usuario puede adaptar la configuración a sus necesidades particulares, logrando así unos registros más precisos.

Por último, ha quedado comprobado que ThingSpeak es una plataforma ideal para analizar los datos obtenidos mediante nuestro dispositivo. Además de ello, no presenta ningún coste para trabajar con el volumen de datos que se generan mediante la monitorización.

7.2. Líneas futuras

El principal problema al que se enfrenta un dispositivo tecnológico es la obsolescencia. La rápida evolución del mercado puede provocar que nuestro dispositivo quede obsoleto en un periodo de tiempo no muy largo. Durante el siguiente capítulo, se presentan diversas mejoras que permiten añadir nuevas funcionalidades al elemento desarrollado. Gracias a esto será posible combatir la obsolescencia y ofrecer un mejor producto a los usuarios.

7.2.1. Mejora de la autonomía

Hasta ahora el dispositivo era alimentado mediante una power bank. Esto supone que cada cierto tiempo el nodo quedará sin batería y el usuario se verá obligado a cargarla. Como se pudo observar durante el capítulo anterior, el sensor de humos provoca que este proceso se tenga que repetir muy frecuentemente.

Para dotar al dispositivo de mayor autonomía se añadirá una placa sobre la placa de desarrollo WeMoS D1 Mini. Dicha placa posee dos enchufes, uno para una batería y el segundo para una conexión USB. Este elemento nos permitirá alimentar la placa mediante la batería y emplear la conexión USB para cargar la batería mediante un pequeño panel solar.

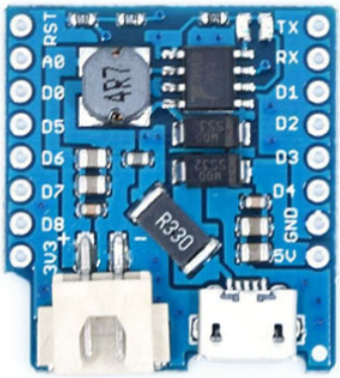

Figura 7.1: Protector de batería para WeMos.

En el mejor de los casos, lograremos dotar de autonomía completa a nuestro dispositivo. En caso de no emplear el sensor de humos es muy probable que podamos alcanzar esta situación, aunque será necesario realizar las pertinentes pruebas de consumo en su momento. Si se lograse alcanzar dicha situación, no sería necesario realizar ninguna acción sobre el dispositivo tras su instalación.

Obviamente la anterior situación será muy difícil de alcanzar. Previsiblemente, lograremos un aumento significativo del tiempo de operación del dispositivo, pero en algún momento será necesario cargar la batería manualmente o reemplazarla por otra cargada. Esto facilitará el mantenimiento del dispositivo por parte de los usuarios.

En cuanto a la placa solar, existen algunas que poseen conector USB. También es posible adquirir un panel solar independiente y adaptar un cable USB para poder conectar ambos elementos. La primera alternativa es significativamente más cara, siendo difícil encontrar uno de estos dispositivos por un precio inferior a 10 €. La segunda opción puede ser encontrada por un precio inferior a 1 € cada unidad. El *shield* puede ser adquirido a vendedores internacionales por un precio en torno a 3 €.

Toma de contacto inicial

Durante la fase final de nuestro trabajo, recibimos el elemento comentado, lo cual nos dejó sin tiempo para realizar el número de pruebas suficientes para confirmar con seguridad el aumento de autonomía comentado.

En la figura 7.2 se muestra el conexionado de los dispositivos comentados durante la sección.

Figura 7.2: Empleo del *shield* sobre la placa WeMos.

Tras realizar el montaje comentado nos encontramos con que el panel solar adquirido no suministraba energía suficiente para nuestras necesidades. Debido a ello sería necesario adquirir uno nuevo con mejores prestaciones. Al no obtener la alimentación suficiente del panel, el dispositivo se comportará del mismo modo que lo hacía anteriormente, solo que en lugar de una *power bank*, se usa una batería Li-ion para alimentarlo.

La figura 7.3 muestra el estado del led que incorpora el *shield* dependiendo del modo de alimentación. En la primera imagen observamos que no posee ninguna iluminación, ya que el led iluminado forma parte de la placa WeMos. Esto indica que el dispositivo se está alimentado mediante la batería.

Cuando se suministra una tensión adecuada mediante USB, el dispositivo puede actuar de dos modos distintos. En primer primer lugar se emplea dicha tensión para cargar la batería, lo cual se indica mediante el color rojo. Cuando la batería está cargada, o en el caso en el cual no dispongamos de batería, el dispositivo funciona gracias a la tensión del puerto USB. Esto queda reflejado cuando el led se ilumina en color verde.

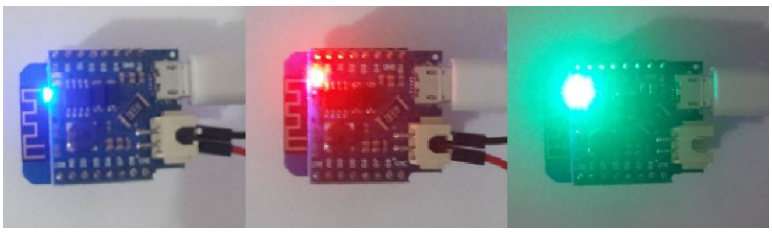

Figura 7.3: Modos de alimentación del *shield*.

Como ya se ha comentado, sería necesario seguir realizando pruebas con estos elementos en fases futuras hasta que consigamos alcanzar nuestro objetivo de aumentar la autonomía.

7.2.2. Actuadores

Ya que nuestro dispositivo es capaz de monitorizar las condiciones climáticas, una posible evolución sería dotar al dispositivo de las capacidades necesarias para que reaccione ante ellas.

Tal y como se comentó en el quinto capítulo, disponemos de dos pines digitales en reserva. En caso añadir actuadores no podríamos controlarlos mediante los multiplexores, ya que los pines que leen dichos elementos han sido declarados como entradas.

Utilizando uno de los pines libres, se añadirá una electroválvula conectada al circuito de riego. Dichas válvula se mueven mediante una bobina solenoide y están diseñadas para controlar el paso de un fluido por un conducto. Generalmente solo cuentan con dos posiciones, abierto y cerrado. La válvula se encontraría de forma predeterminada cerrada. Cuando el higrómetro detectase que la humedad del suelo está bajo un determinado umbral, se abriría la electroválvula para regar el cultivo.

La válvula también podría ser abierta en caso de que el sensor de humos detectase niveles de gas muy elevados. La apertura del riego dificultaría la expansión de un posible incendio. Esto sería especialmente útil si el riego se realizase mediante aspersores.

El control de una electroválvula es muy sencillo. Basta con cambiar el estado de la salida digital para abrirla o cerrarla. Es posible encontrar uno de estos dispositivos por un precio cercano a los 3 € por unidad en proveedores internacionales. El único problema que presentan dichos dispositivos es que generalmente suelen necesitar un voltaje de alimentación de 12 V. Debido a ello sería necesaria una fuente de energía complementaria para

dicho elemento.

Figura 7.4: Electroválvula.

7.2.3. Nueva tecnología inalámbrica

Cuando hablamos de tecnologías inalámbricas, la opción escogida habitualmente es la tecnología Wi-Fi. Esta elección se debe principalmente a que el uso de dichas redes es muy común en entornos domésticos y comerciales. Las infraestructuras actuales permiten la transferencia de grandes flujos de datos a velocidades elevadas.

El hecho de que nuestro dispositivo cuente con dicha tecnología también presenta algunos efectos no tan beneficiosos. En primer lugar, no es habitual contar con la infraestructura necesaria en entornos agrícolas y el alcance no es demasiado elevado, lo cual obligará a instalar diversos puntos de acceso. Por otro lado, la tecnología Wi-Fi consume demasiada potencia en el proceso de transmisión para una aplicación IoT.

Sería recomendable estudiar el empleo de una nueva tecnología inalámbrica en una posible evolución futura del dispositivo. Actualmente las tecnologías disponibles ofrecen una velocidad de transmisión más que suficiente para nuestras necesidades. Las principales diferencias se encuentran en el apartado de consumo y en el alcance.

A continuación se comentarán brevemente las principales tecnologías a considerar, para lo cual se han tenido en cuenta aquellas con mayor alcance.

- Red de telefonía móvil. Gracias a la tecnología 4G es posible enviar grandes cantidades de datos a velocidades entre 3 y 10 Mbps. El alcance de estas redes es superior a 30 Km. Por contra, esta opción puede

provocar que aumente el coste y tampoco se optimiza el consumo en la transmisión.

- LoRaWAN. Permite implementar redes WAN para aplicaciones de IoT, M2M y ciudades inteligentes. Se encuentra optimizada para necesitar el menor consumo energético posible. Soporta comunicaciones móviles, económicas, bidireccionales y seguras. La velocidad de transmisión varía entre 0.3 y 50 Kbps.

- Sigfox. Tecnología emergente de largo alcance. Utiliza la tecnología Ultra Narrow Band diseñada para trabajar con bajas velocidades de transferencia. En ambientes rurales el alcance de dichas redes supera los 30 Km. Se trata de una tecnología robusta, muy eficiente y escalable.

En caso de emplear el dispositivo en explotaciones de pequeña extensión, también pueden considerarse alternativas como ZigBee, Bluetooth o NFC.

Bibliografía

[1] *Scott Shelton*, Sweating High Humidity, Greenhouse Product News, Dec. 2005. Disponible en http://www.gpnmag.com/article/sweating-high-humidity/ (accedida el 19 de junio de 2017).

[2] Influencia de la temperatura ambiental en las plantas, *CANNA Research*, 2017. Disponible en http://www.canna.es/influencia_temperatura_ambiental_en_las_plantas (accedida el 19 de junio de 2017).

[3] *Jose Chen López*, La influencia de la luz en el crecimiento del cultivo, *Premier Tech Horticulture*, July 14, 2016. Disponible en http://www.pthorticulture.com/es/centro-de-formacion/la-influencia-de-la-luz-en-el-crecimiento-del-cultivo/ (accedida el 19 de junio de 2017).

[4] Web de Libelium - Connecting Sensors to the Cloud, disponible en http://www.libelium.com (accedida el 19 de junio de 2017).

[5] Web de Nazaríes IT - Su partner tecnológico, disponible en http://www.nazaries.com/ (accedida el 19 de junio de 2017).

[6] Web de CropX Inc., disponible en https://cropx.com/ (accedida el 19 de junio de 2017).

[7] Web de Brioagro Technologies. Inteligencia móvil para el cultivo, disponible en http://brioagro.es/ (accedida el 19 de junio de 2017).

[8] Web de ThingSpeak: IoT Analytics, disponible en https://thingspeak.com/ (accedida el 19 de junio de 2017).

[9] Web de Arduino, disponible en https://www.arduino.cc/ (accedida el 19 de junio de 2017).

[10] Web de Raspberry Pi - Teach, Learn, and Make with Raspberry Pi, disponible en https://www.raspberrypi.org/ (accedida el 19 de junio de 2017).

[11] ESP8266EX Datasheet, *Espressif Systems IOT Team*, version 5.4, May 2017. Disponible en http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (accedida el 19 de junio de 2017).

[12] *Todd Henderson*, Top 6 ESP8266 Modules for IoT Projects, *Losant | Losant Enterprise IoT Platform*, April 19, 2016. Disponible en https://www.losant.com/blog/top-6-esp8266-modules (accedida el 19 de junio de 2017).

[13] D1 mini [WEMOS Electronics], disponible en https://wiki.wemos.cc/products:d1:d1_mini (accedida el 19 de junio de 2017).

[14] DS18B20. Programmable Resolution 1-Wire Digital Thermometer, *Maxim Integrated*, Rev 4, Jan. 2015. Disponible en https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf (accedida el 19 de junio de 2017).

[15] Guide for Soil Moisture Sensor YL-69 or HL-69 with Arduino, *Random Nerd Tutorials*. Disponible en https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/ (accedida el 19 de junio de 2017).

[16] Rain sensor module, *OpenHacks*. Disponible en https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf (accedida el 19 de junio de 2017).

[17] DHT11 Humidity & Temperature Sensor, *D-Robotics*, July 30, 2010. Disponible en http://www.micropik.com/PDF/dht11.pdf (accedida el 19 de junio de 2017).

[18] Technical data mq-2 gas sensor, *Hanwei Electronics CO*. Disponible en http://sandboxelectronics.com/files/SEN-000004/MQ-2.pdf (accedida el 19 de junio de 2017).

[19] HC-SR501 Pir Motion Detector, *Marlin P. Jones & Assoc. Inc.*. Disponible en https://www.mpja.com/download/31227sc.pdf (accedida el 19 de junio de 2017).

[20] GL55 Series Photoresistor, *Senba Optical & Electronic CO*. Disponible en http://akizukidenshi.com/download/ds/senba/GL55%20Series%20Photoresistor.pdf (accedida el 19 de junio de 2017).

[21] CD74HC4067, CD74HCT4067 High-Speed CMOS Logic 16-Channel Analog Multiplexer/Demultiplexer, *Texas Instruments*, rev July 2003. Disponible en

http://www.ti.com/lit/ds/symlink/cd74hc4067.pdf (accedida el 19 de junio de 2017).

[22] ESP8266 Low Power Solutions, *Espressif IOT Team*. Version 1.3. Aug. 2016. Disponible en https://espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf (accedida el 19 de junio de 2017).

[23] *Óscar Torrente Artero*, ARDUINO Curso práctico de formación, Primera edición. 2013.

[24] *Charalampos Doukas*, Building Internet of Things with the Arduino, Volumen 1. 2012.

[25] Web de Luis Llamas - Ingeniería, informática y diseño (Zaragoza), *Luis Llamas*. Disponible en https://www.luisllamas.es/ (accedida el 19 de junio de 2017).