

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

CONTROL AUTOMÁTICO DE UN SISTEMA DE
AIRE ACONDICIONADO

GRADO EN INGENIERÍA DE
SISTEMAS ELECTRÓNICOS

BORJA LEBRÓN GARCÍA
MÁLAGA, 2022

Control automático de un sistema de aire acondicionado.

Autor: Borja Lebrón García.

Tutor: Francisco Javier Vizcaíno Martín.

Departamento: Departamento de Tecnología Electrónica.

Titulación: Grado en Ingeniería de Sistemas Electrónicos.

Palabras clave: Sistema basado en microcontroladores, ESP32, Señales inalámbricas, MQTT, ESP-NOW.

Resumen

Los sistemas de aire acondicionado son instrumentos cada vez más usados por la sociedad, con una gran variedad de dispositivos con diferentes tecnologías en cada uno de ellos.

Este proyecto pretende hacer un uso más eficiente de los equipos de aire acondicionado, además de permitir, en lo posible, alargar la vida media de uso de un equipo, al poder incorporar una serie de microcontroladores que serán los encargados de realizar el control del *split* del sistema de aire acondicionado. Este control del funcionamiento se realizará con un módulo principal, el cual será el encargado de mandar los comandos que indican la acción que debe realizarse. Además, cuenta con otros dos módulos que están compuestos por distintos sensores que enviarán la información recogida por estos al módulo principal, mediante ESP-NOW.

Por otro lado, nos encontramos con la emisión de mensajes desde una interfaz de usuario hacia el módulo principal mediante el protocolo de comunicación MQTT, que se encarga de realizar el intercambio de mensajes a través de la red de internet, lo que provoca que siempre vamos a necesitar que estos dispositivos se encuentren conectados a la red, tanto el módulo principal como el que contenga la interfaz de usuario, para que se pueda producir el intercambio de mensajes entre ellos.

Automatic control of an air conditioning system.

Author: Borja Lebrón García.

Supervisor: Francisco Javier Vizcaíno Martín.

Department: Departamento de Tecnología Electrónica.

Degree: Grado en Ingeniería de Sistemas Electrónicos.

Keywords: System based on microcontrollers, ESP32, Inalambrics signals, MQTT, ESP-NOW.

Abstract

Air conditioning systems are increasingly used by society, with a wide variety of devices with different technologies.

This project aims to make more efficient use of air-conditioning equipment and, in addition, to extend as far as possible the average life of use of an equipment. This control of the operation will be carried out with a main module, which will be responsible for sending the commands that indicate the action to be performed. It also has two others modules that are composed of sensors that send the information collected by these to the main module, through ESP-NOW.

On the other hand, we are faced with the transmisión of messages from a user interface to the main module by the protocol MQTT, which is responsible for exchanging messages over the internet network, which means that we will always need our devices to be connected to the internet, the main module and the one containing the user interface, in order to be able to send the messages between them.

Agradecimientos

Me gustaría agradecer en primer lugar a mi tutor, Francisco Javier Vizcaino Martín por haberme dado la oportunidad de realizar el proyecto bajo su supervisión y ayuda. También quisiera agradecer a todos aquellos profesores que me ofrecieron distintas alternativas de proyectos para poder realizarlos con ellos.

Quiero continuar agradeciendo su labor a todos los docentes que en algún momento han coincidido conmigo en alguna clase, ya que todos ellos han conseguido aportar algo a la persona que he llegado a ser con el paso de los años.

También a mis padres y mi hermano, porque siempre me han estado apoyando desde el primer día que comencé a estudiar y han dado su mejor esfuerzo por estar siempre a mi lado y darme todo lo que me fuera necesario para sacarlo todo adelante. Además, agradecer también a mi familia por todo lo que se han preocupado por mí y por saber cómo me iba todo en cada momento.

Por último, pero no menos importante quiero agradecerles a mis amigos por la paciencia que han tenido conmigo y por estar siempre que he tenido algún momento de bajón o de incertidumbre para que siguiera luchando por aquello que quería. Añadir también a todos aquellos compañeros que han pasado por la Escuela y con los que he tenido la oportunidad de compartir aula con ellos, porque cada uno ha sido cada capaz de aportar lo que necesitaba en cada momento y me han hecho aprender cosas a través de sus vivencias o consejos para poder sacar este grado adelante.

Con todo ello, sólo me queda estar eternamente agradecido a cada uno de ellos por hacerme ser quien soy con todo lo que me habéis llegado a aportar.

*Dedicado a mis padres y
hermano por ser la fuente
de mis fuerzas y mi dedicación.*

Contenido

Lista de Acrónimos.....	iii
Capítulo 1. Introducción	1
1.1. Contexto tecnológico.....	1
1.1.1. Productos de otros fabricantes	6
1.2. Objetivos del Trabajo Fin de Grado.....	7
1.3. Estructura de la memoria	8
Capítulo 2. Especificaciones del sistema	11
2.1. Requisitos	13
2.2. Especificaciones.	14
Capítulo 3. Desarrollo del sistema	15
3.1. Desarrollo hardware	16
3.1.1. Microcontrolador.....	16
3.1.2. Sensores	17
3.1.3. Diseño de las placas.....	20
3.2. Desarrollo software	26
3.2.1. Código de funcionamiento.....	26
3.2.2. Aplicación de usuario	35
Capítulo 4. Verificación y pruebas.....	41
4.1. Sistema de pruebas	43
4.2. Pruebas realizadas	44
Capítulo 5. Manual de instalación y uso	55
5.1. Manual de instalación.....	55
5.2. Manual de uso.....	60

Capítulo 6. Conclusiones y trabajo futuro	69
6.1. Conclusiones	69
6.2. Líneas futuras	70
Apéndice A. Presupuesto de elaboración	71
Apéndice B. Esquemáticos de las PCBs	73
Referencias	77

Lista de Acrónimos

PCB	Printed Circuit Board (Placa de Circuito Impreso)
MQTT	Message Queing Telemetry Transport
IR	Infrared (Radiacion Infrarroja)
BJT	Bipolar Junction Transistor
PIR	Passive Infrared (Infrarrojo Pasivo)
COM	Common or Ground (Común o Tierra)
NC	Normal Close (Normalmente cerrado)
NO	Normal Open (Normalmente Abierto)

Capítulo 1. Introducción

1.1. Contexto tecnológico

Los aparatos de aire acondicionado, desde que se empezaron a comercializar en el formato que hoy en día se les conoce, siempre han tenido una buena acogida dentro de la sociedad, aunque con el paso de los años se les ha ido introduciendo diferentes mejoras en cuanto a su tecnología, los materiales y los componentes que los conforman, haciendo que estos se tengan que cambiar al cabo de un tiempo para conseguir que cumplan con las nuevas normativas que se puedan aplicar en cada época [1]. En un principio los sistemas de aire acondicionado tenían una refrigeración controlada con el gas R-12 (desde 1930). Posteriormente se pasó a utilizar el gas R-22 (desde 1950) y, por último, se están utilizando en la actualidad los gases R-410A y R-404A (desde 1990). Aunque actualmente ha salido una nueva normativa que provocará que se produzcan nuevos cambios en su diseño y en la composición del material que utilizan para su refrigeración, de manera que se tendrá que volver a cambiar el suministro de los gases de refrigeración, debiendo utilizarse los gases: R452A, R407A, R410A, R407F, R134A o R449A [2].



Figura 1.1. Composición de un sistema de aire acondicionado. [3]

Los sistemas de aire acondicionado están compuestos principalmente por dos unidades, una que se encuentra en el exterior de la vivienda y otra que se encuentra en el interior, esta última recibe el nombre de *split*, que es la palabra con la que se hará referencia a esta unidad. Aunque también se pueden encontrar sistemas que están compuestos por un único módulo que integra los componentes del sistema exterior y del *split*. En el *split* se puede observar que tiene integrado los sistemas de funcionamiento y de recepción de las señales que son enviadas desde los dispositivos de control remoto. Su composición se puede observar en la Figura 1.2, donde se muestran todos aquellos componentes que van integrados dentro de este módulo, aunque para esta ocasión, el más importante será el receptor de las señales. Sobre esta parte del dispositivo será sobre la que más se va a interactuar, al utilizarlo para enviarle la información deseada de cómo debe funcionar en cada momento.

Unidad interior

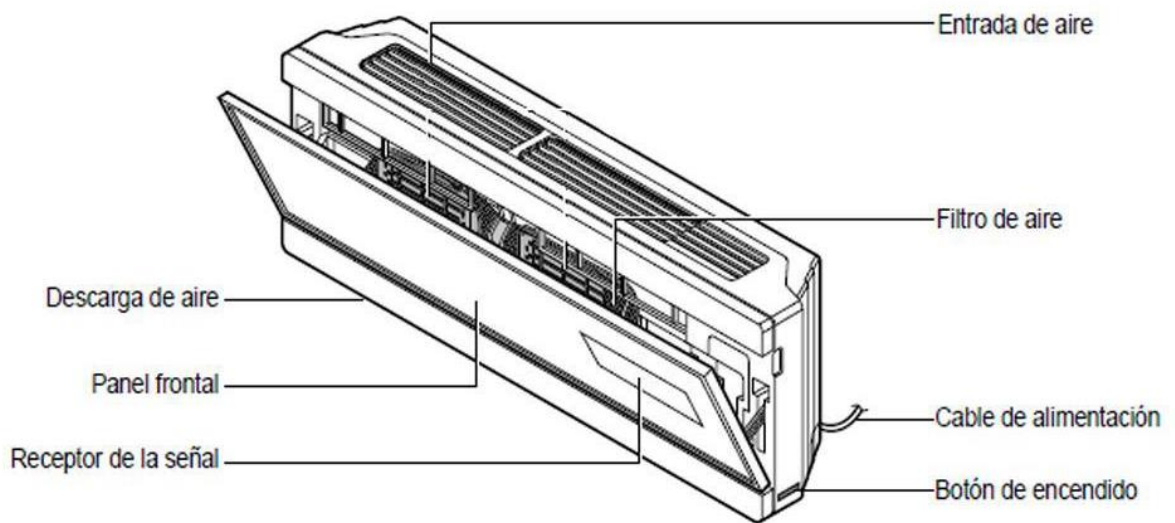


Figura 1.2. Composición de un Split. [4]

En la actualidad se encuentran diversos problemas que se pueden solucionar con la implementación de este proyecto. El más importante de ellos vendría dado por el consumo de energía. Desde que comenzó el cambio en la manera con la que se calcula el consumo energético, entre ellos el uso de los tramos horarios que alteran el precio de la energía consumida, como se puede observar en la Figura 1.3, hasta la subida del coste de dicha energía han provocado que el gasto producido por la energía eléctrica que consume cada vivienda sea mayor. En este caso, se ha vivido que, en el mes de marzo de 2022, el valor medio de la energía eléctrica llegó a ser de 545€/MWh [5]. Estos costes están muy por encima de los valores que esta había tomado en años anteriores. En este contexto se encuentra que progresivamente se están teniendo valores de récord en el precio de la energía eléctrica [6].

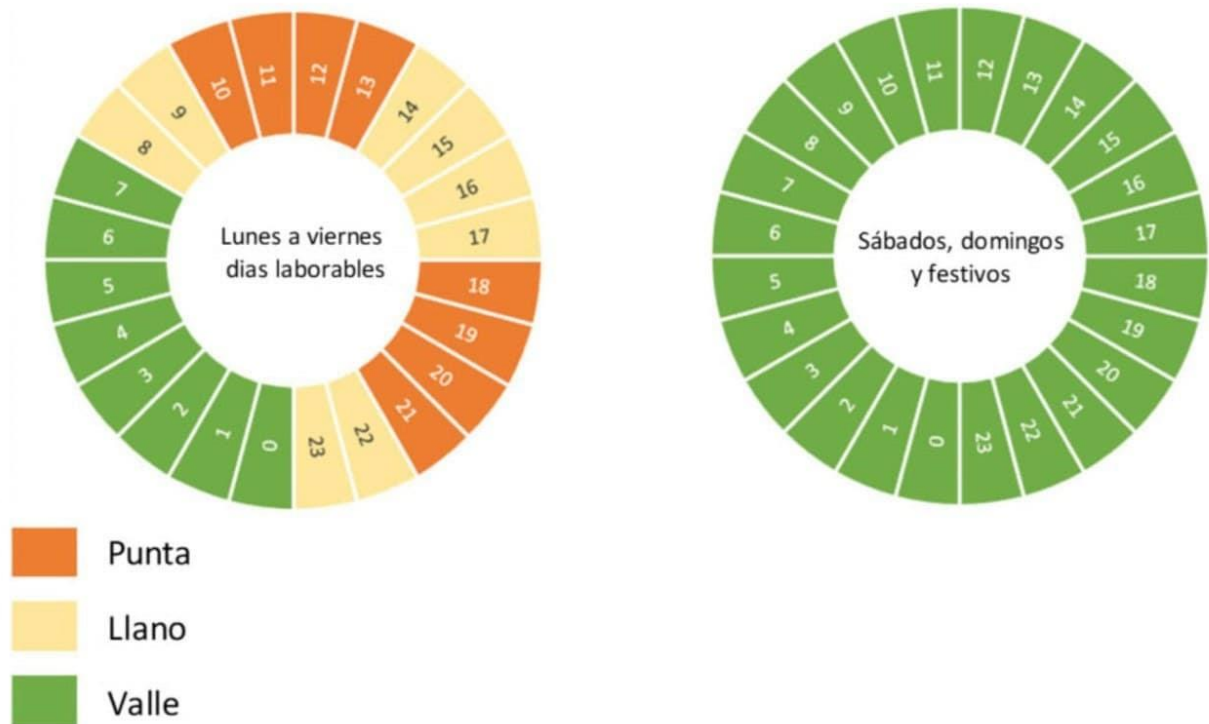


Figura 1.3. Diferenciación de los tramos horarios entre los días laborales y los fines de semana. [6]

Junto al problema del consumo eléctrico se encuentra otro problema, normalmente pasado por alto, es el de los residuos electrónicos que se producen anualmente en todo el mundo debido a que las tecnologías de los dispositivos se quedan obsoletas o que, al fallar un componente, se prefiere desechar a poder arreglarlo o volver a reutilizar el resto de los materiales en otro lugar. En el año 2019 se llegaron a generar 53.6 millones de toneladas de residuos electrónicos en todo el mundo, de los cuales sólo se reciclaron un 17.4%, un valor relativamente bajo [7]. Con este proyecto, al pretender que la vida de uso de estos componentes sea mayor, se conseguirá que estos niveles de contaminación electrónica sean menores, al menos en el apartado de los aires acondicionados. Esto se conseguirá mediante la implantación de un nuevo sistema de control, el cual permite que un usuario desde fuera de su estancia sea capaz de indicar qué acciones tiene que realizar el sistema de aire acondicionado, es decir, consigue que el usuario tenga una manera más cómoda de utilizar el sistema de control de un aire acondicionado.

Este proyecto se va a basar en la comunicación principal entre dos dispositivos, que serán el *split* del aire acondicionado y el microcontrolador principal, que estará constituido por un sensor de infrarrojos de transmisión y recepción de datos. Por otro lado, se encuentran dos módulos, ambos de ellos compuestos por un microcontrolador, aunque cada uno conectado a distintos sensores. Uno de los cuales se unirá a un sensor térmico y a un sensor de presencia, con los que podremos detectar si hay alguien en la estancia y así mantenerla en un estado de temperatura confortable para el usuario. El segundo microcontrolador se encontrará conectado a un sensor de proximidad, colocado en las ventanas, de modo que se pueda detectar si están abiertas y en consecuencia determinar si es conveniente tener encendido el *split*. Por último, el sistema se encontrará en continua conexión y transmisión de datos con la interfaz de usuario, que se encontrará en una página web diseñada para el caso. En la Figura 1.4 se muestra de una forma más esquematizada como sería la composición de este proyecto.

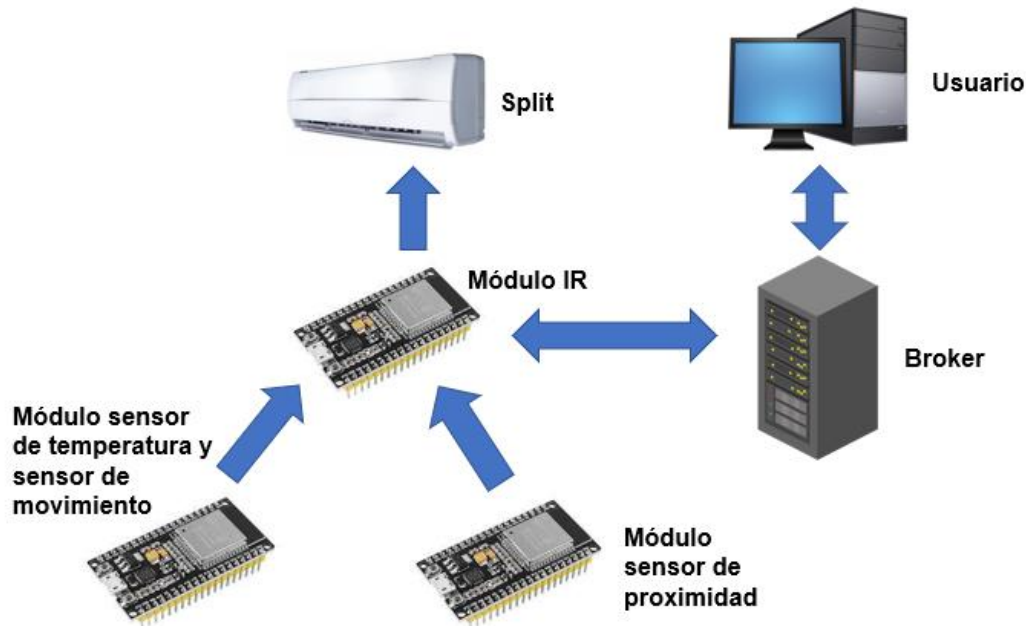


Figura 1.4: Modelo del proyecto.

Con esto, se pretende obtener una mayor eficiencia de los dispositivos con respecto a su gasto energético y el tiempo que estos necesitan estar conectados para mantener un estado de confort en la estancia para los usuarios. Esto se conseguirá mediante el conjunto de sensores de los que dispondrá el sistema, para saber si hay alguien en la estancia o para que no se desperdicie energía al tener fugas por las ventanas. De igual manera, se trata de conseguir que el periodo de vida de los aires sea mayor y no se produzca el cambio de los dispositivos debido a que estos se han quedado anticuados respecto a la tecnología que poseen, es decir, no se pueden conectar a ningún otro dispositivo electrónico y solo se puede realizar su control desde un mando remoto. Por último, al contar con que el control del aire lo puede tener el usuario desde su dispositivo móvil o desde un ordenador, el sistema permite una mayor comodidad para este al no exigir el empleo de un mando específico a distancia para poder modificar los parámetros del dispositivo y el sistema.

1.1.1. Productos de otros fabricantes

Frente a este sistema, se encontrarán por un lado con las propias aplicaciones de los fabricantes de los sistemas de aire. Estas aplicaciones vienen integradas directamente dentro de los *split* del sistema, los cuales se comunican únicamente con aquellas que son de su misma marca y no se pueden conectar con otros modelos de distintos fabricantes. Esto provoca que su compatibilidad sea limitada y que requiera comprar todos los sistemas de aires acondicionados del mismo fabricante para la estancia al completo y no tener que estar utilizando distintas aplicaciones dependiendo del aparato que se desee de controlar. Dentro de estas compañías que están comenzando a realizar sistemas de aire acondicionado controlados con su propia aplicación se encuentran: Toshiba, Daikin, Fujitsu y Mitsubishi entre otras [8] [9] [10]. Aunque dentro de estas, un caso particular lo constituye Mitsubishi, que permite incorporar un dispositivo dentro de sus *split* para poder realizar su control, aunque esto podría traer problemas si no se instala adecuadamente. Estos problemas vendrían dados con el hecho de que la garantía del producto no cubre la manipulación inadecuada

del equipo, por lo que esta se podría perder si la instalación de estos dispositivos no fuera realizada por un personal adecuado.

Por otro lado, podemos observar que también hay aplicaciones genéricas de diferente naturaleza que posibilitan conectarse con un *split* para realizar las funciones de control sobre ellos. Dentro de estos dispositivos, hay algunos que se encuentran integrados en el *split*, lo que podría generar conflictos con el fabricante, otros que se comunican con el *split* mediante infrarrojos y reciben la información para efectuar los cambios en él mediante wifi. Algunos de ellos al estar conectados a la red mediante wifi, dan la posibilidad de obtener datos del tiempo atmosférico que hará en la ubicación donde se encuentre el usuario al que informan de estos valores para poder ajustar la temperatura óptima para la estancia. Estos últimos también aprovechan que la vivienda donde se encuentren pueda estar domotizada, es decir, compuesta por más dispositivos electrónicos que puedan interconectar entre sí, consiguiendo un mayor control de la estancia [11].

1.2. Objetivos del Trabajo Fin de Grado

En este proyecto se pretende perseguir una serie de objetivos que se diferenciaron entre software y hardware.

Para el objetivo hardware, nos encontramos con la generación de un sistema funcional constituido por diferentes módulos compuestos por sensores con los que se permita el control de un equipo de aire acondicionado.

Como objetivo software, el sistema contará con un diseño de una interfaz de usuario, la cual permitirá la comunicación bidireccional con el sistema de forma que en ella se podrán observar y configurar los distintos parámetros de control del dispositivo.

1.3. Estructura de la memoria

La estructura de la memoria que se va a seguir estará constituida por siete apartados en los que se irán describiendo diferentes conceptos que son importantes para comprender cómo será el proyecto en su totalidad.

- Primer capítulo: Requisitos del proyecto.

Para este primer apartado, definiremos cuáles son los requisitos que se deberán cumplir a lo largo de todo el proyecto. Dentro de estos requisitos se encontrarán algunos con mayor prioridad que otros, otros que serán fundamentales para el funcionamiento del sistema y otros que serán opcionales, dependiendo de las necesidades y deseos del cliente.

- Segundo capítulo: Diseño Hardware.

El apartado describe el diseño hardware del sistema. En primer lugar, el capítulo se centra en indicar cuáles son todos los componentes hardware que se van a utilizar e indicar cuáles son las características principales que justifican su elección. Dentro de estos componentes se encontrarán distintos sensores, algunos microcontroladores y alguna placa de circuito impreso. Además de indicar los materiales a usar, también se explicarán cuáles han sido los pasos que se han seguido para conseguir que todos los materiales se hayan podido integrar para realizar las funciones que deseamos de este proyecto.

- Tercer capítulo: Diseño Software.

Se comentará el diseño software incluyendo las distintas plataformas que se van a utilizar a lo largo de toda la implementación de la interfaz de usuario y el código necesario para programar los distintos microcontroladores. Tras tener un conocimiento de estas plataformas, procederemos a mostrar cómo se han ido realizando las distintas fases de implementación de los distintos subsistemas, interfaz de usuario y la funcionalidad de los microcontroladores.

- Cuarto capítulo: Pruebas de funcionamiento.

En este apartado se mostrarán todas las pruebas que se han tenido que realizar para determinar el correcto funcionamiento del sistema diseñado. Comenzando por las pruebas de cada uno de los distintos sensores por separado y continuando por las pruebas de cada módulo con todos los sensores, correspondientes a su parte, incorporados. Más adelante se evalúa el funcionamiento de todos los subsistemas en conjunto, transmitiendo datos entre las tres partes de las que se compondrá dicho subsistema. Por último, se presentan las pruebas de transmisión con la interfaz de usuario desde el microcontrolador principal, para comprobar que los datos transmitidos y recibidos en cada punto son los deseados.

- Quinto capítulo: Manual de funcionamiento.

Continuaremos en este apartado con la descripción del manual de funcionamiento de este sistema. En él se mostrará cómo será la interacción con la interfaz de usuario, desde la que se podrán realizar dos configuraciones principales, el modo automático y el modo manual.

- Sexto capítulo: Conclusiones y líneas futuras.

Este punto lo dedicaremos para exponer los resultados finales que se han obtenido a lo largo de todo el proceso de diseño e implementación del proyecto, de modo que este sirva para mostrar cómo será al final el proyecto. En él se podrá observar cuál sería la disposición típica del producto para una vivienda típica y el precio final por el que se podría realizar su implementación.

- Séptimo capítulo: Anexos.

Para este último apartado, se van a realizar distintos anexos en los que se mostrarán varios conjuntos de elementos. Se empezará por indicar cual ha sido el presupuesto que se ha necesitado para generar el proyecto. En el último, se podrán visualizar los esquemáticos y diseños de las distintas PCBs, realizando aclaraciones de lo que representa cada figura.

Capítulo 2. Especificaciones del sistema

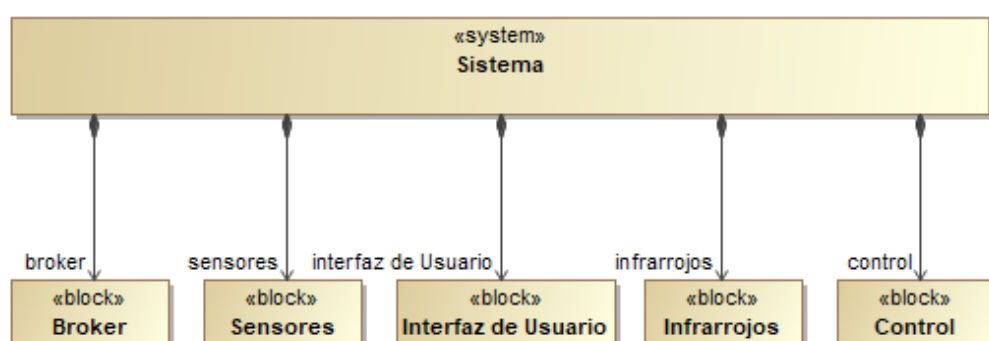


Figura 2.1. Diagrama genérico de bloques del sistema

El prototipo en el que se basa este proyecto se encuentra constituido por un conjunto de microcontroladores que realizarán una comunicación de datos continua entre ellos y por una aplicación que hará las funciones de interfaz de usuario con el *split*, con el que se comunicará tras enviar un mensaje a través del *broker*, ubicado en el servidor de MQTT en la dirección “broker.hivemq.com”, que a su vez envía los datos al módulo principal que le indicará la acción a realizar al *split*.

En la Figura 2.1 se observan diferentes bloques que hacen referencia a las distintas partes que forman parte del sistema general. Empezando de izquierda a derecha, se encuentra el *broker*, que es el punto intermedio en la comunicación entre el módulo principal y la interfaz de usuario. A continuación, se observa el nombre de sensores, el cual hace referencia a los dos módulos compuestos por

diversos sensores que se encargan de transmitir los datos recibidos por los sensores hacia el módulo principal. La siguiente parte en aparecer es la interfaz de usuario, que es una aplicación que se ha generado para que el usuario sea capaz de controlar las acciones que se tienen que ejecutar sobre el sistema de aire acondicionado. Más adelante se encuentra el infrarrojo, el cual hace referencia al módulo principal de microcontroladores, que se encuentra compuesto por un interfaz de infrarrojos con el que envía los datos al *split*. Por último, se puede observar el control, que hace mención del *split* al cual se le realizará la transmisión de mensajes para indicar el estado en el que se debe de encontrar según lo desee el usuario.

Junto a este módulo, se encontrarán otros dos más que se encargan de realizar las lecturas de los valores de los sensores que tienen incorporados y transmitirán estos datos hacia el módulo principal para que este los trate y decida la acción que ha de realizar.

Además de estos módulos se encuentra una aplicación, desarrollada mediante QT, que será encargada de comunicarse con el *bróker* ubicado en el servidor de MQTT, estos datos serán los que a su vez enviará y recibirá el módulo principal. Todo esto servirá para poder realizar el control del *split* sin necesidad de realizar ninguna interacción interna con el dispositivo.

El módulo principal está pensado para poder realizar las acciones que se muestran en la Figura 2.2, en ella se representan los distintos casos de uso que tiene el sistema, desde sus configuraciones manuales como las que son programables por el usuario. En las funciones manuales se conseguirá que el usuario pueda realizar el apagado del *split*, el encendido de este con una temperatura asignada o directamente realizar un cambio en la temperatura que hay establecida. Para el caso de las configuraciones programables, se podrá realizar el encendido o apagado del dispositivo en la fecha y hora que se haya decidido o en el caso de que ya no sea necesario realizar la acción correspondiente, se podrá cancelar el funcionamiento que hubiera previsto.

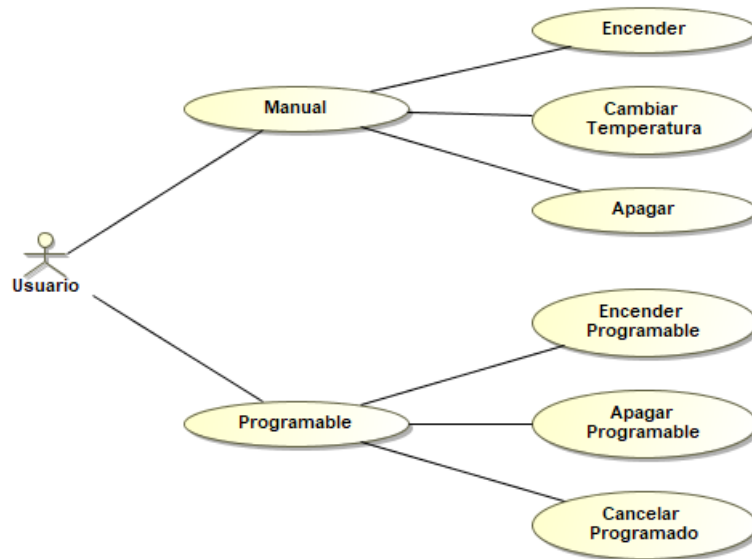


Figura 2.2. Casos de uso.

2.1. Requisitos

A continuación, se detalla en la Tabla 2.1 un listado de requisitos que se deben cumplir para poder realizar el diseño del prototipo. Este diseño se encontrará basado y restringido por los elementos que se muestran en la Figura 2.1.

Id.	Nombre	Descripción	Tipo	Prioridad
1.1	Sensores	Los sensores deben ser capaz de realizar las mediciones correspondientes de cada uno de ellos	Funcional	Alta
1.1.1	Sensor temperatura y humedad	de Realizar las lecturas de los valores de temperatura, en un rango de -20°C a 50°C en el entorno		
1.1.2	Sensor movimiento	de Ser capaz de detectar la presencia de personas en una estancia en una distancia mínima de 2m desde el techo		
1.1.3	Sensor proximidad	de Ser capaz de detectar el estado de apertura en el que se encuentra una ventana		
1.2	Comunicación entre microcontroladores	Se debe conseguir una comunicación entre los distintos microcontroladores	Funcional	Alta
1.3	Comunicación entre la interfaz de usuario y el módulo principal	El módulo principal ha de ser capaz de comunicarse con la interfaz de usuario ubicado en el ordenador del usuario	Funcional	Alta
1.4	Aplicación	Esta aplicación ha de ser capaz de comunicarse con un servidor de manera bidireccional	Funcional	Alta

1.4.1	Encendido manual	Realizar el encendido instantáneo del <i>split</i> del dispositivo
1.4.2	Apagado manual	Realizar el apagado instantáneo del <i>split del dispositivo</i>
1.4.3	Cambio de temperatura	Realizar el cambio de temperatura del <i>split del dispositivo</i>
1.4.4	Encendido programado	Permitir la configuración del encendido del <i>split</i> del dispositivo para el momento indicado
1.4.5	Apagado programado	Permitir la configuración del apagado del <i>split</i> del dispositivo para el momento indicado
1.4.6	Cancelar programado	Permitir anular las configuraciones de encendido y apagado programado

Tabla 2.1. Tabla de descripción de requisitos.

2.2. Especificaciones.

En este apartado, se marcarán en la Tabla 2.2 los hitos que se deberán cumplir durante el desarrollo del sistema, lo que impondrá la elección final de los componentes y el diseño que seguirán los distintos módulos.

Id.	Nombre	Descripción	Prioridad
2.1	Alimentación	Proporcionar una fuente de alimentación constante para los distintos microcontroladores.	Media
2.2	Montaje	Realizar el diseño de distintas PCBs para colocar los sensores en el microcontrolador correspondiente	Alta
2.3	Ubicación	Conocer dónde se van a colocar los distintos subsistemas de los que se va a componer el proyecto.	Media

Tabla 2.2. Tabla de descripción de especificaciones.

Capítulo 3. Desarrollo del sistema

Una vez se han descrito las condiciones de contorno sobre las que debe trabajar el sistema, se procederá a realizar el diseño de este en sus apartados software y hardware. En el diagrama de bloques propuesto en la Figura 3.1 se muestra cómo son las conexiones que se van a producir entre los agentes externos al dispositivo y los propios que están incorporados dentro de cada módulo.

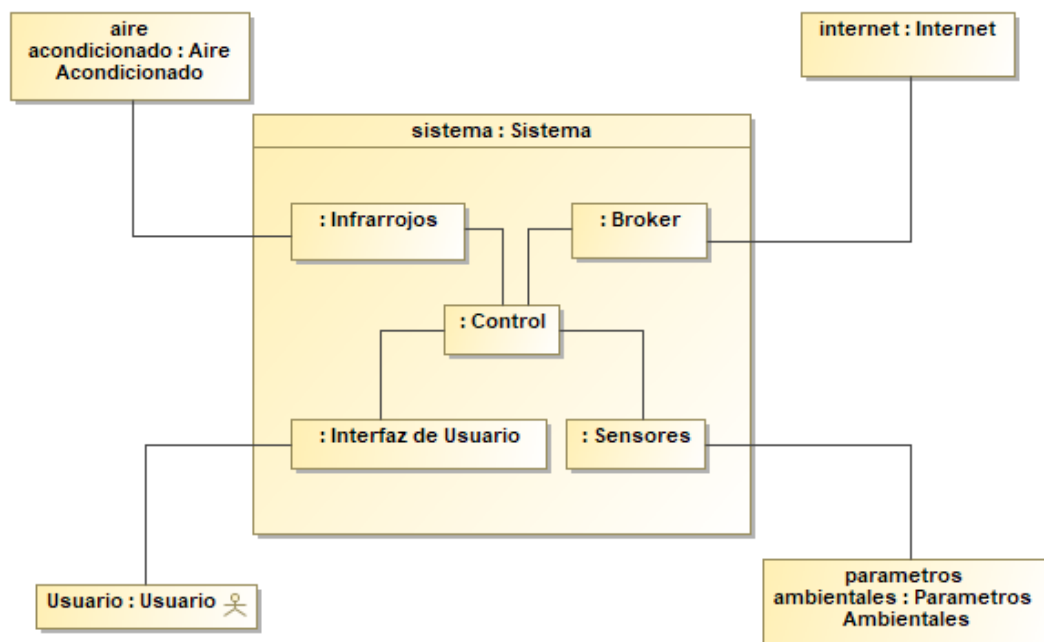


Figura 3.1. Diagrama de bloques del sistema.

Comenzaremos realizando la explicación de los parámetros externos con los que se encontrará el sistema. Siguiendo el orden la Figura 3.1, el primer agente externo sería el *split* del aire acondicionado con el que se va a comunicar mediante un interfaz de infrarrojos, el cual enviará pulsos de distinto tamaño y duración que permitirá el control del aire acondicionado. En segundo lugar, se observará al usuario, que se encargará de transmitir mensajes al sistema indicando qué acción es la que desea realizar en cada momento, esto tendrá lugar mediante una aplicación. A continuación, se usará la red de internet para transmitir los mensajes entre los distintos módulos del sistema y para la transmisión del módulo principal con el *broker*, que se encargará de la comunicación con la interfaz de usuario. Por último, se encuentran los parámetros ambientales, estos estarán compuestos por la temperatura y humedad que haya en la estancia donde se encuentre el dispositivo, si está sucediendo algún movimiento por la misma, o para detectar el estado de las ventanas o puertas donde se encuentre el último módulo.

3.1. Desarrollo hardware

En este punto se procederá a realizar el estudio de la placa ESP32 y de los sensores que se ha decidido colocar en los distintos módulos para el desarrollo del proyecto. Por último, se realizará un desarrollo de cómo se han diseñado las distintas PCBs para poder enviarlas a impresión o para realizar su montaje sobre una placa pretaladrada.

3.1.1. Microcontrolador

Se ha seleccionado el microcontrolador ESP32 ESP-WROOM-32 NodeMCU [12], Figura 3.2, la cual posee las características necesarias para desarrollar todas las funciones que van a ser desarrolladas por los sistemas. Entre ellas se encuentra el módulo wifi y bluetooth que tiene integrado el dispositivo, esto permite que no sea necesario añadir ningún componente externo para poder tener una conexión estable con la red de internet, por tanto, tener unas

comunicaciones fluidas con el servidor que se encuentra en ella. Por otro lado, posee un mecanismo interno del sistema que contiene integrado en su chip, se trata de las comunicaciones mediante ESP-NOW, el cual permite convertir a la placa en un esclavo o en un máster para la comunicación entre ellos. El módulo principal será el máster, es decir, recibirá toda la información procedente de los otros microcontroladores, que en este caso serán los esclavos.



Figura 3.2. ESP32 ESP-WROOM-32 NodeMCU. [12]

3.1.2. Sensores

Los sensores que se han decidido usar son los siguientes:

- Sensor de temperatura: DHT22.
- Sensor PIR de movimiento: HC-SR501.
- Sensor magnético de proximidad: MC-31B.

Se ha decidido utilizar el sensor DHT22 como sensor de temperatura y humedad al ser el último modelo y mejorar en prestaciones a su antecesor el DHT11. Este sensor está compuesto por dos partes, una primera parte que es la que nos interesa en este proyecto, es un termistor, el cuál detectará la temperatura de su alrededor en un rango de temperatura desde los -40°C hasta los 80°C con una

precisión de medición de $\pm 0.5^{\circ}\text{C}$. En segundo lugar, un sensor de humedad capacitivo, el cual también puede proporcionar unos datos interesantes de cara a su control. En la Figura 3.3 se puede observar cómo sería la estructura de este módulo y las conexiones mediante las cuales se conectará a la PCB que se ha diseñado para el módulo que está compuesto por este dispositivo y por el sensor de movimiento.



Figura 3.3. Sensor DHT22. [13]

El segundo sensor que se va a utilizar es el HC-SR501, que es un detector de movimiento por medio de la detección de la luz infrarroja transmitida por otros dispositivos o por las propias personas, lo cual, permitirá saber si hay algún ser vivo en la estancia donde se encuentre el dispositivo. Este sensor cuenta con dos potenciómetros ubicados en el lateral que permite un mejor ajuste de las características del dispositivo. Si se observa en la Figura 3.4 el dispositivo visto desde arriba, la resistencia variable de la izquierda permite configurar el tiempo entre disparos del sensor si ha detectado algún movimiento, mientras que la resistencia variable de la derecha será la encargada de ajustar la distancia hasta la que es posible que el sensor detecte una señal infrarroja. Además, este dispositivo permite un ángulo de visión, hasta la distancia selecciona, de unos 110° .



Figura 3.4. Sensor PIR HC-SR501. [14]

Por último, se ha seleccionado el sensor de proximidad magnético MC-31B, Figura 3.5, por la simplicidad de su uso, su posibilidad de colocación en cualquier lugar y su bajo precio. Este dispositivo está constituido por dos piezas, una de las cuales está compuesta por unas pletinas, que hacen la función de interruptor para indicar el estado en el que este se encuentra el sensor, en esta pieza se encuentran los pines en los cuales se realizan las conexiones para transmitir los datos, mientras que la otra, contiene un imán en su interior. En la pieza donde se encuentran las pletinas hay tres puntos para conexiones, los cuales son: COM, NC y NO.

El sensor cambia su estado dependiendo de donde se encuentre la pieza con el imán respecto a la pieza con las pletinas y a su vez en relación a la conexión que se haya decidido usar entre NC y NO. En este caso, se ha realizado una conexión con el modo NO, el cual devuelve que la ventana o puerta está abierta cuando las piezas se encuentran separadas, mientras que cuando se encuentran juntas indica que está cerrada.

Este sensor se ubicará principalmente en ventanas o puertas dependiendo de lo que se desee controlar.



Figura 3.5. Sensor magnético de proximidad MC-31B. [15]

3.1.3. Diseño de las placas

Por último, se concluirá este apartado realizando la explicación del proceso de desarrollo para la creación de las distintas PCBs que se han generado para cada módulo.

La primera placa que se ha diseñado ha sido aquella que va en el módulo principal, es decir, la que está compuesta por distintos componentes eléctricos y se encarga de realizar las comunicaciones con el *split* del aire acondicionado al que se quiera controlar. Los componentes electrónicos que forma parte de esta placa son los siguientes:

- Led de luz infrarroja.
- Resistencia de 100Ω .
- Resistencia de $3.9k\Omega$.
- Condensador de $100\mu F$.
- Transistor BJT 2N2222.

Estos componentes han sido elegidos para poder diseñar el circuito eléctrico de un amplificador en colector común, que sea capaz de transmitir señales mediante la interfaz de infrarrojos a una distancia de aproximadamente 2m respecto del *split*. Esto se ha conseguido al incorporar al emisor del transistor una resistencia de 100Ω para limitar la corriente que llega al led y que éste pueda transmitir a la distancia indicada. También se ha colocado una resistencia de

3.9k Ω para limitar la tensión que cae en el transistor y que este solo pueda estar en corte o en activa para producir en el envío de los pulsos IR. Por último, se ha decidido de colocar un condensador de 100uF para realizar las funciones de desacoplo en relación a la señal de alimentación del circuito proveniente del microcontrolador. En la Figura 3.6 se puede visualizar cómo sería el diseño final del circuito que se ha implementado para el módulo principal.

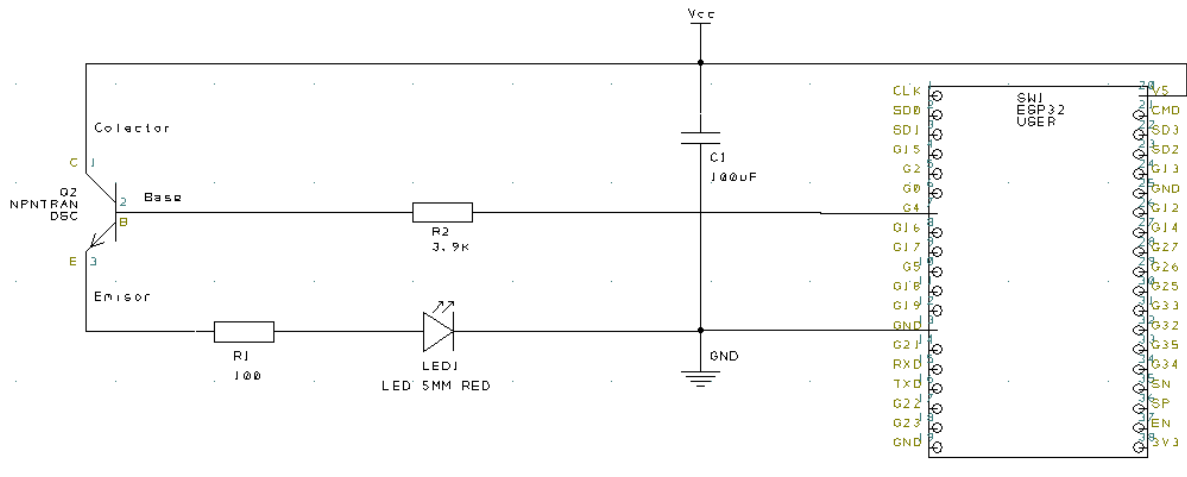


Figura 3.6. Esquemático del módulo IR.

En la Figura 3.7 y la Figura 3.8 se puede observar cómo ha quedado el montaje de la placa tras realizar las soldaduras en una placa pretaladrada de los componentes, junto con la colocación del microcontrolador en los pines de inserción que se han soldado a la placa pretaladrada, de manera que se pueda realizar la extracción y el cambio del microcontrolador de una manera más cómoda en caso de que se produjera un problema en él. A su vez, la Figura 3.9 muestra cómo sería el esquema de las conexiones eléctricas que se verían en el diseño de la PCB.



Figura 3.7. Montaje real del módulo IR por la cara de la placa.

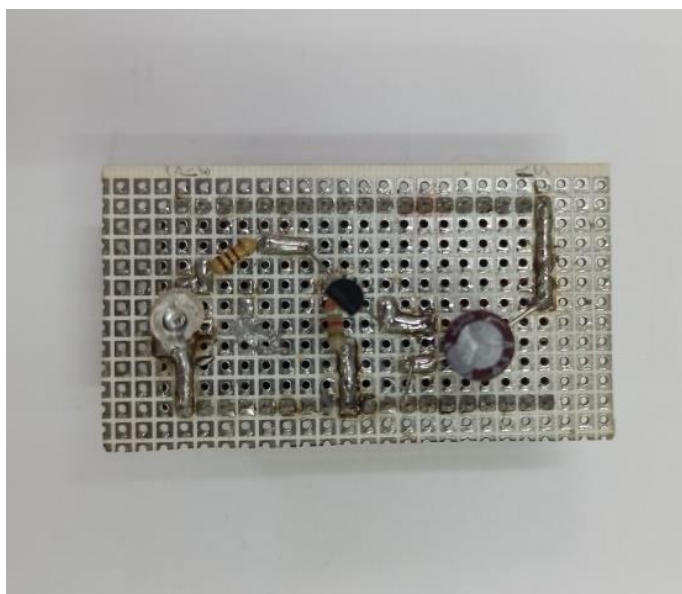


Figura 3.8. Montaje real del módulo IR por la cara de los componentes.

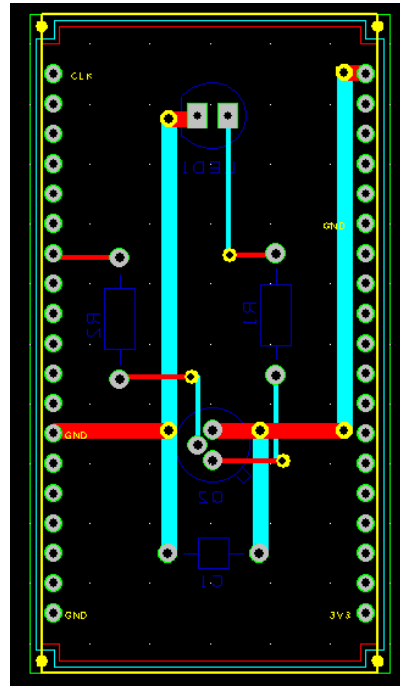


Figura 3.9. Esquema del módulo IR.

El segundo diseño, que se ha realizado está compuesto por dos sensores, los cuales son el sensor de temperatura y humedad y el sensor de movimiento con los que se ha pretendido realizar un diseño sencillo para su montaje, el cual ha quedado como se puede observar en la Figura 3.10. En la Figura 3.11 se puede visualizar cómo es el esquemático del que se compone el diseño de la placa y que a su vez es trasladado a la PCB para que quede un esquema con las conexiones eléctricas con la forma que se encuentra en la Figura 3.12.

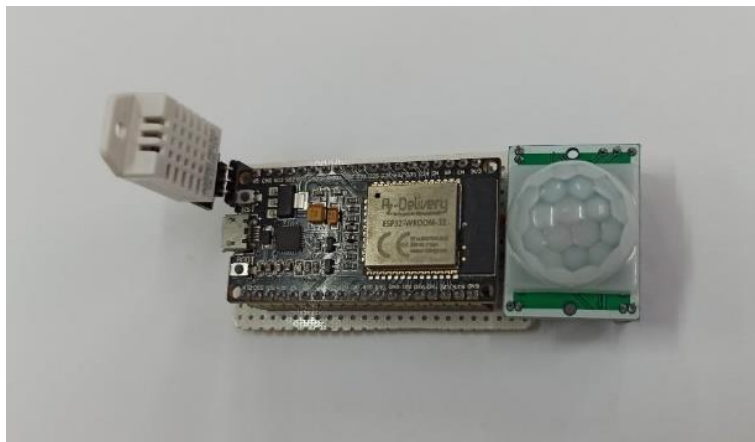


Figura 3.10. Montaje real del módulo con sensor de temperatura y de movimiento.

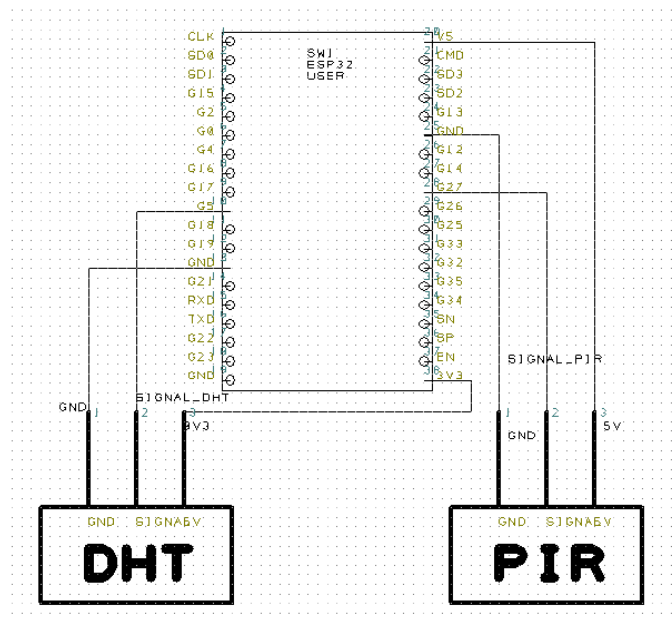


Figura 3.11. Esquemático del módulo con sensor de temperatura y movimiento.

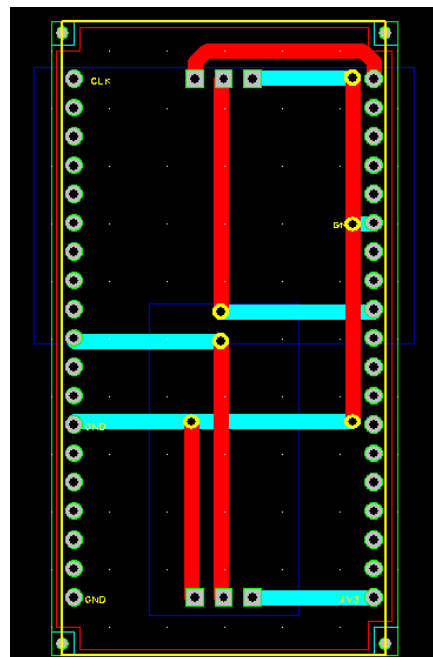


Figura 3.12. Esquema del módulo con el sensor de temperatura y movimiento.

El último módulo, se puede considerar el más sencillo de diseñar, debido a que solo está compuesto por un sensor que, además, se debe encontrar a una cierta distancia del microcontrolador que recibe los valores del sensor y no va directamente fijado a la placa pretaladrada que se utilice para el montaje. En la

Figura 3.13 se muestra cómo se ha realizado el montaje físico del módulo sobre una placa pretaladrada. También se encuentra el esquemático diseñado para este módulo en la Figura 3.14, mientras que el esquema con las conexiones eléctricas para el diseño de la PCB se puede observar en la Figura 3.15.

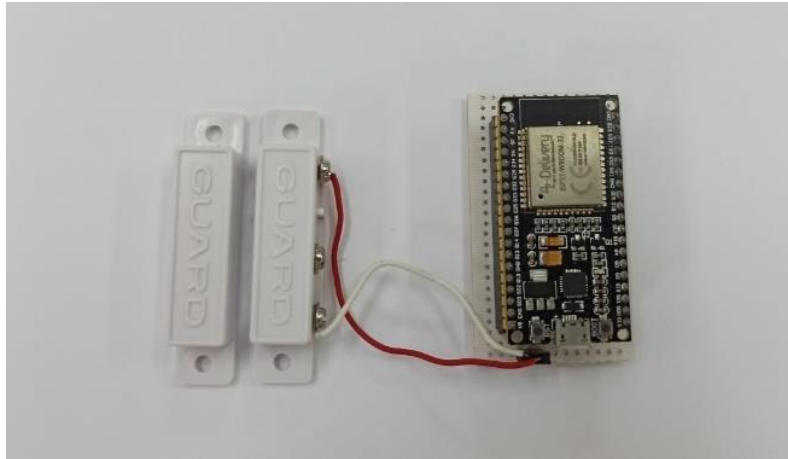


Figura 3.13. Montaje real del módulo con sensor de proximidad.

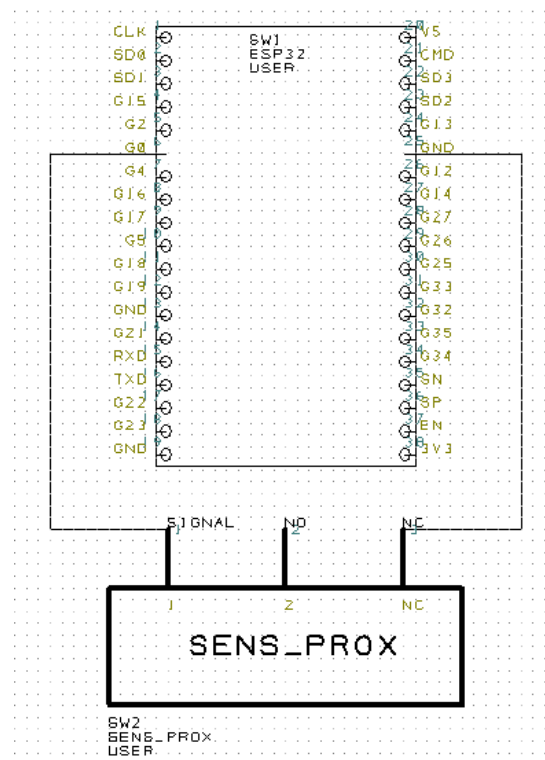


Figura 3.14. Esquemático del módulo con sensores de proximidad.

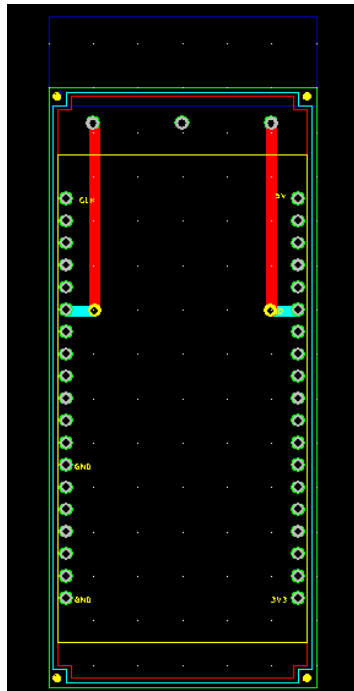


Figura 3.15. Esquemático del módulo con sensores de proximidad.

3.2. Desarrollo software

El diseño software se dividirá en varios apartados, entre los que se encontrará el código que se ha desarrollado para las distintas placas ESP32, cada una realiza una función diferente, por lo que necesitan un código distinto. También se contará con el código diseñado para la aplicación que hará las funciones de interfaz entre el usuario y el sistema del que se compone el proyecto.

3.2.1. Código de funcionamiento

Cada placa de ESP32 tiene integrado un código de funcionamiento diferente dependiendo del módulo al que pertenezca y las acciones que tenga que realizar este. La primera placa estará compuesta por el circuito eléctrico que determina la usabilidad del led infrarrojo. Este a su vez, será el código más complejo de los que se desarrollarán, debido a que contiene las acciones a realizar por el sistema, desde controlar cuál es la fecha de cada momento, hasta realizar lecturas de datos del *broker* que está en la red, el tratamiento de los datos y

posterior código de acción que debe enviar al *split* para que este funcione. A su vez, también posee un modo de configuración interno mediante el cual el dispositivo realizará un apagado del *split*, esto sucederá si se da alguno de los siguientes casos: no haya movimiento en la estancia durante un periodo de tiempo de 15 minutos, si la estancia se encuentra a la temperatura a la que se ha programado el dispositivo durante el mismo periodo de tiempo y si se abre una ventana o puerta, dependiendo de donde se ubique el módulo con el sensor de proximidad, y esta se mantuviera abierta durante un periodo de tiempo de 5 minutos. Estas configuraciones de protección se realizan para que el gasto energético de la vivienda sea lo más reducido posible.

A continuación, se realizará una explicación de las partes más importantes e interesantes de las que está compuesta el código del módulo principal. Se comenzará con las estructuras que poseen los campos de datos con los que se va a estar trabajando tanto en el código de las placas como en el código de la aplicación. Estas estructuras están definidas en la Figura 3.16 y Figura 3.17. Además de estas estructuras, también nos encontramos con las definidas para la transmisión de datos entre las distintas placas, Figura 3.18, poseyendo cada una de ellas una parte de código donde se marcará cuál es el dato que envía, mientras que en el módulo principal estará el código perteneciente a las tres placas ya que será la encargada de recibir los valores transmitidos y realizar acciones en función de los datos.

```

33 typedef struct{
34     uint8_t val_fecha_dia;
35     uint8_t val_fecha_mes;
36     uint16_t val_fecha_ano;
37     uint8_t val_reloj_hora;
38     uint8_t val_reloj_min;
39     uint8_t val_temp;
40     uint8_t val_ejec;
41 }MESSAGE_ENCENDIDO_PROGRAMADO_PARAMETROS;
42
43 typedef struct{
44     uint8_t val_fecha_dia;
45     uint8_t val_fecha_mes;
46     uint16_t val_fecha_ano;
47     uint8_t val_reloj_hora;
48     uint8_t val_reloj_min;
49     uint8_t val_ejec;
50 }MESSAGE_APAGADO_PROGRAMADO_PARAMETROS;

```

Figura 3.16. Estructura de los mensajes enviados y recibidos del servidor. Parte 1.

```

15 typedef struct {
16     uint8_t command;
17 } MESSAGE_REJECTED_PARAMETROS;
18
19 typedef struct{
20     uint8_t val_ejec;
21     uint8_t est_ventana;
22 }MESSAGE_ENCENDIDO_MANUAL_PARAMETROS;
23
24 typedef struct{
25     uint8_t val_ejec;
26 }MESSAGE_APAGADO_MANUAL_PARAMETROS;
27
28 typedef struct{
29     uint16_t value_temp;
30     uint8_t val_ejec;
31 }MESSAGE_CAMBIO_TEMPERATURA_PARAMETROS;

```

Figura 3.17. Estructura de los mensajes enviados y recibidos del servidor. Parte 2.

```

18 typedef struct message_IR{
19     int id;
20 }message_IR;
21
22 typedef struct message_temp {
23     int id;
24     int temp;
25     int hum;
26     int mov;
27     int aviso;
28 }message_temp;
29
30 typedef struct message_prox {
31     int id;
32     int sens_prox;
33     int aviso;
34 }message_prox;

```

Figura 3.18. Estructura de los mensajes enviados y recibidos entre placas.

En la Figura 3.19, Figura 3.20 y Figura 3.21, se puede observar una parte de código, en las primeras líneas de cada una de las figuras se encuentran las acciones de apagado de las distintas marcas de aire acondicionado, Mundo Clima, Panasonic y Fujitsu respectivamente, y a continuación, sus comandos propios para realizar el cambio de temperatura al valor deseado, el cual se encuentra indicado en el nombre de las variables de manera que se pueda seleccionar principalmente entre los valores de 18°C a 30°C, con un paso de 1°C entre cada uno. Sin embargo, algunos modelos poseen valores de configuración

para la temperatura que permiten modificar los valores en 0.5° y que tenga un rango de acción desde los 16°C a los 32°C. Los valores utilizados son los códigos raw, estos códigos están representados por el tiempo que permanece la señal en sus valores de tensión alto y bajo, normalmente estos datos se presentan agrupados en grupos de 2 y se coloca un doble espacio tras estos valores. Estos códigos serán los que se transmitirán por el interfaz de IR que está ubicado en el módulo principal del sistema.

```

70 //MUNDO_CLIMA MODO:HEAT FAN:MAX
71 //Protocolo de envio de los mensajes de NEC LIKE
72 uint32_t address = 0x84C0; //Address para el envio de los datos por el sistema NEC
73
74 uint64_t MC_PARA_APAGADO = 0x032168A5; //(32 Bits)
75 uint16_t raw_MC_PARA_APAGADO[67] = {8990, 4528, 616, 478, 618, 474, 534, 584, 620, 478,
76
77 uint32_t MC_TEMP_16 = 0x032008D4; //(32 Bits)
78 uint16_t raw_MC_TEMP_16[67] = {9018, 4448, 592, 524, 570, 550, 558, 534, 568, 524, 572,
79
80 uint32_t MC_TEMP_17 = 0x03208854; //(32 Bits)
81 uint16_t raw_MC_TEMP_17[67] = {9044, 4442, 596, 498, 626, 490, 536, 556, 594, 498, 598,
82
83 uint32_t MC_TEMP_18 = 0x03204894; //(32 Bits)
84 uint16_t raw_MC_TEMP_18[67] = {9016, 4472, 566, 526, 566, 554, 558, 536, 564, 528, 568,
85
86 uint32_t MC_TEMP_19 = 0x0320C814; //(32 Bits)
87 uint16_t raw_MC_TEMP_19[67] = {9020, 4472, 568, 524, 568, 552, 564, 528, 568, 522, 568,
88
89 uint32_t MC_TEMP_20 = 0x032028E4; //(32 Bits)
90 uint16_t raw_MC_TEMP_20[67] = {9046, 4470, 534, 560, 566, 528, 568, 552, 566, 526, 566,
91
92 uint32_t MC_TEMP_21 = 0x0320A864; //(32 Bits)
93 uint16_t raw_MC_TEMP_21[67] = {8986, 4508, 590, 524, 566, 528, 566, 552, 566, 528, 566,
94
95 uint32_t MC_TEMP_22 = 0x032068A4; //(32 Bits)
96 uint16_t raw_MC_TEMP_22[67] = {9010, 4484, 590, 526, 566, 528, 532, 586, 532, 560, 568,

```

Figura 3.19. Código de un mando de la marca Mundo Clima.

```

1 //PANASONIC MODO:AUTO FAN:AUTO
2 //Protocolo de envio de los mensajes de PANASONIC_AC
3 //define PAN PARA APAGADO 0x0220E0040000000060220E00400082C80AF00000EE00000890000E0 //(216
4 uint8_t PAN PARA APAGADO[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20,
5 uint16_t raw_PAN PARA APAGADO[439] = {3528, 1722, 492, 374, 444, 1288, 444, 422, 444, 4
6
7 //define PAN TEMP_16 0x0220E0040000000060220E00400092080AF00000EE00000890000D5 //(216 Bits)
8 uint8_t PAN_TEMP_16[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
9 uint16_t raw_PAN_TEMP_16[439] = {3496, 1754, 412, 454, 412, 1320, 414, 452, 438, 428,
10
11 //define PAN_TEMP_17 0x0220E0040000000060220E00400092280AF00000EE00000890000D7 //(216 Bits)
12 uint8_t PAN_TEMP_17[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
13 uint16_t raw_PAN_TEMP_17[439] = {3500, 1750, 416, 452, 412, 1318, 414, 452, 416, 450,
14
15 //define PAN_TEMP_18 0x0220E0040000000060220E00400092480AF00000EE00000890000D9 //(216 Bits)
16 uint8_t PAN_TEMP_18[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
17 uint16_t raw_PAN_TEMP_18[439] = {3530, 1720, 444, 424, 442, 1288, 444, 422, 444, 422,
18
19 //define PAN_TEMP_19 0x0220E0040000000060220E00400092680AF00000EE00000890000DB //(216 Bits)
20 uint8_t PAN_TEMP_19[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
21 uint16_t raw_PAN_TEMP_19[439] = {3580, 1672, 494, 372, 444, 1288, 444, 422, 494, 372,
22
23 //define PAN_TEMP_20 0x0220E0040000000060220E00400092880AF00000EE00000890000DD //(216 Bits)
24 uint8_t PAN_TEMP_20[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
25 uint16_t raw_PAN_TEMP_20[439] = {3530, 1722, 444, 422, 444, 1288, 494, 372, 444, 422,
26
27 //define PAN_TEMP_21 0x0220E0040000000060220E00400092A80AF00000EE00000890000DF //(216 Bits)
28 uint8_t PAN_TEMP_21[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
29 uint16_t raw_PAN_TEMP_21[439] = {3530, 1696, 468, 422, 446, 1288, 444, 398, 468, 422,
30
31 //define PAN_TEMP_22 0x0220E0040000000060220E00400092C80AF00000EE00000890000E1 //(216 Bits)
32 uint8_t PAN_TEMP_22[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0
33 uint16_t raw_PAN_TEMP_22[439] = {3530, 1722, 442, 424, 442, 1288, 444, 422, 444, 422,

```

Figura 3.20. Código de un mando de la marca Panasonic.

```

133 //FUJITSU MODE:AUTO FAN:AUTO
134 //define FUJ APAGADO 0x146300101002FD //(56 Bits)
135 uint8_t FUJ APAGADO[7] = {0x14, 0x63, 0x00, 0x10, 0x10, 0x02, 0xFD};
136 uint16_t raw_FUJ APAGADO[115] = {3342, 1592, 454, 396, 430, 400, 430, 1204, 432, 398,
137
138 //define FUJ TEMP_18 0x1463001010FE0930210010000000208F //(128 Bits)
139 uint8_t FUJ_TEMP_18[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x21, 0x00, 0x00
140 uint16_t raw_FUJ_TEMP_18[259] = {3378, 1580, 490, 336, 464, 364, 462, 1172, 462, 368,
141
142 //define FUJ TEMP_19 0x1463001010FE0930310010000000207F //(128 Bits)
143 uint8_t FUJ_TEMP_19[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x31, 0x00, 0x00
144 uint16_t raw_FUJ_TEMP_19[259] = {3378, 1576, 436, 390, 464, 366, 436, 1200, 464, 366,
145
146 //define FUJ TEMP_20 0x1463001010FE0930410010000000206F //(128 Bits)
147 uint8_t FUJ_TEMP_20[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x41, 0x00, 0x00
148 uint16_t raw_FUJ_TEMP_20[259] = {3348, 1610, 436, 392, 434, 394, 434, 1200, 434, 396,
149
150 //define FUJ TEMP_21 0x1463001010FE0930510010000000205F //(128 Bits)
151 uint8_t FUJ_TEMP_21[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x51, 0x00, 0x00
152 uint16_t raw_FUJ_TEMP_21[259] = {3348, 1606, 436, 390, 436, 394, 436, 1200, 434, 394,
153
154 //define FUJ TEMP_22 0x1463001010FE0930610010000000204F //(128 Bits)
155 uint8_t FUJ_TEMP_22[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x61, 0x00, 0x00
156 uint16_t raw_FUJ_TEMP_22[259] = {3348, 1608, 434, 392, 434, 396, 434, 1200, 434, 396,
157
158 //define FUJ TEMP_23 0x1463001010FE0930710010000000203F //(128 Bits)
159 uint8_t FUJ_TEMP_23[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x71, 0x00, 0x00
160 uint16_t raw_FUJ_TEMP_23[259] = {3348, 1608, 434, 392, 434, 368, 460, 1200, 434, 396,

```

Figura 3.21. Código de un mando de la marca Fujitsu.

Se seguirá con la explicación de las configuraciones de *setup* de los distintos tipos de comunicaciones que se van a ejecutar en el módulo principal. En primer lugar, se configura el wifi de la manera que se puede observar en la Figura 3.22, es decir, en ella se realiza una conexión con la red wifi que haya en la estancia donde se encuentre el dispositivo. En la siguiente imagen, Figura 3.23, se puede observar la conexión con el servidor que se encuentra en la red, además se produce la activación del *callback* de ejecución para los momentos en los que llega algún mensaje al *broker* y se realiza una lectura de los datos que se encuentran en él, esto se realiza en la línea 80 descrito en la Figura 3.23. El último *setup* consistirá en la activación del protocolo ESP-NOW, Figura 3.24, para poder efectuar la comunicación entre los distintos módulos con el módulo principal, también se configurará el *callback* para la recepción de los datos de las otras placas, esta acción se muestra en la línea 134 de la Figura 3.24.

```
91 void setup_wifi() {
92   WiFi.mode(WIFI_AP_STA);
93
94   delay(10);
95   Serial.println();
96   Serial.print("Connecting to ");
97   Serial.println(ssid);
98   WiFi.begin(ssid, password);
99   while (WiFi.status() != WL_CONNECTED) {
100     delay(500);
101     Serial.print(".");
102   }
103   Serial.println("");
104   Serial.println("WiFi connected");
105   Serial.println("IP address: ");
106   Serial.println(WiFi.localIP());
107   Serial.print("Wi-Fi Channel: ");
108   Serial.println(WiFi.channel());
109 }
```

Figura 3.22. Configuración setup_wifi.

```
77 void setup_cliente() {
78     irsend.begin();
79     client.setServer(mqtt_server, mqttPort);
80     client.setCallback(callback_json);
81     client.setBufferSize(1024);
82     if (!client.connected()) {
83         reconnect();
84     }
85
86     Serial.print("ESP Board MAC Address: ");
87     Serial.println(WiFi.macAddress());
88     delay(2000);
89 }
```

Figura 3.23. Configuración setup_cliente.

```
129 void setup_esp_now() {
130     if (esp_now_init() != ESP_OK) {
131         Serial.println("Error initializing ESP-NOW");
132         return;
133     }
134     esp_now_register_recv_cb(OnDataRecv);
135 }
```

Figura 3.24. Configuración setup_esp_now.

En los módulos que están compuestos por los sensores también se necesita realizar la configuración de algunos parámetros como son la activación del protocolo ESP-NOW, para la transmisión de datos de estas placas hacia el módulo principal y la configuración de la conexión al wifi y la selección del canal de comunicación que van a utilizar, como se puede observar en la Figura 3.25, cuyo código pertenece a “Random Nerd Tutorial” [16]. Si no se realiza adecuadamente esta configuración puede producirse una aparición de problemas en la comunicación, ya que el protocolo ESP-NOW toma unos canales para la transmisión que pueden ser a su vez los mismos canales que se utilicen en el módulo principal para la conexión al wifi, por lo tanto, hay que indicar en esta configuración los canales que se van a utilizar para cada uno de los tipos de comunicación.


```

55 // Set device as a Wi-Fi Station
56 WiFi.mode(WIFI_STA);
57
58 int32_t channel = getWiFiChannel(WIFI_SSID);
59
60 WiFi.printDiag(Serial); // Uncomment to verify channel number before
61 esp_wifi_set_promiscuous(true);
62 esp_wifi_set_channel(channel, WIFI_SECOND_CHAN_NONE);
63 esp_wifi_set_promiscuous(false);
64 WiFi.printDiag(Serial); // Uncomment to verify channel change after
65
66 // Init ESP-NOW
67 if (esp_now_init() != ESP_OK) {
68     Serial.println("Error initializing ESP-NOW");
69     return;
70 }
71
72 // Once ESPNow is successfully Init, we will register for Send CB to
73 // get the status of Trasnmitted packet
74 esp_now_register_send_cb(OnDataSent);
75
76 // Register peer
77 memcpy(peerInfo.peer_addr, broadcastAddress, 6);
78 peerInfo.channel = 0;
79 peerInfo.encrypt = false;
80
81 // Add peer
82 if (esp_now_add_peer(&peerInfo) != ESP_OK){
83     Serial.println("Failed to add peer");
84     return;
85 }

```

Figura 3.25. Configuración Wifi y ESP-NOW en los módulos 2 y 3.

La Figura 3.26 muestra cómo se realiza la configuración de la fecha que hay en la red en la placa del módulo principal, ya que esta al iniciarse adquiere una fecha que es distinta a la actual, por lo tanto, hay que realizar esta acción para poder inicializar con el valor actual.

```

228 void Actualiza_Fecha_Hora() {
229     configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
230     SetLocalTime();
231     fecha_actual = now();
232     Serial.println("Genero la fecha inicial");
233 }

```

Figura 3.26. Configura la fecha y hora de la ESP32.

Dentro de las acciones principales a realizar por este módulo, se comenzará mostrando cómo se genera un mensaje de tipo JSON para poder enviar los datos al *broker* sobre el *topic* indicado, el cual se observa en la línea 310 que es “/TFG/ConectarAA”, el cual es el recibido por la aplicación que se ha desarrollado, esto se puede visualizar en la Figura 3.27. Otra función importante se encontraría en la Figura 3.28, la cual mostrará cómo será la determinación de la acción que debe realizar el módulo dependiendo de los parámetros que se hayan podido leer del *broker* y/o de los datos que lleguen desde los sensores ubicados en las otras placas.

```

305 void Envio_Datos_Broker() {
306     String payload2 = "";
307     doc_envio["EstadoVentana"] = board_prox.sens_prox;
308     doc_envio["EstadoAire"] = message_enc_man.val_ejec;
309     serializeJson(doc_envio, payload2);
310     client.publish("/TFG/Estados", (char*)payload2.c_str());
311 }

```

Figura 3.27. Envío de datos hacia el *broker*.

```

382 void Realizo_Accion_Aire() {
383     Seleccion_Dispositivo();
384     if (nombre_disp == "PANASONIC") {
385         Comprueba_Encendido_Apagado_Man();
386         if(message_cam_temp.value_temp == 0){
387             temp_ejecucion = last_temp;
388         }else if(message_cam_temp.value_temp != 0){
389             temp_ejecucion = message_cam_temp.value_temp;
390         }
391         Accion_Panasonic(estado_encendido, temp_ejecucion);
392     }
393     } else if (nombre_disp == "MUNDO CLIMA") {
394         Comprueba_Encendido_Apagado_Man();
395         if(message_cam_temp.value_temp == 0){
396             temp_ejecucion = last_temp;
397         }else if(message_cam_temp.value_temp != 0){
398             temp_ejecucion = message_cam_temp.value_temp;
399         }
400         Accion_Mundo_Clima(estado_encendido, temp_ejecucion);
401     }
402     } else if (nombre_disp == "FUJITSU") {
403         Comprueba_Encendido_Apagado_Man();
404         if(message_cam_temp.value_temp == 0){
405             temp_ejecucion = last_temp;
406         }else if(message_cam_temp.value_temp != 0){
407             temp_ejecucion = message_cam_temp.value_temp;
408         }
409         Accion_Fujitsu(estado_encendido, temp_ejecucion);

```

Figura 3.28. Determina la acción a realizar.

Por último, se puede observar en la Figura 3. 29 cómo sería el envío de datos a las distintas marcas de aire acondicionado mediante el led IR ubicado en el módulo principal. El caso de la marca Fujitsu es diferente debido a que los tamaños del mensaje a enviar para el apagado y el encendido o cambio de temperatura son diferentes.

```

293 void Envio_Datos_IR_Panasonic(uint16_t code[]) {
294     irsend.sendRaw(code, TAM_MES_PAN, FREQ);
295 }
296
297 void Envio_Datos_IR_Mundo_Clima(uint16_t code[]) {
298     irsend.sendRaw(code, TAM_MES_MC, FREQ);
299 }
300
301 void Envio_Datos_IR_Fujitsu(uint16_t code[], uint16_t tam) {
302     irsend.sendRaw(code, tam, FREQ);
303 }

```

Figura 3. 29. Envía los datos mediante IR.

3.2.2. Aplicación de usuario

Se ha desarrollado la aplicación que va a utilizar el usuario a través de la herramienta de desarrollo QT, la cual permite la generación de pestañas sobre las que se pueden incluir diferentes tipos de figuras para generar una interfaz gráfica sobre la que interactuar para realizar las acciones que sean necesarias. Esto se puede observar en la Figura 3.30 y la Figura 3.31, las cuales representan los modos de actuación manual y programable respectivamente. La generación que se ha realizado del programa está basada en dos aspectos, uno primero en el que se encontrará la interfaz gráfica sobre la que trabajará el usuario que estará compuesta por distintos tipos de componentes sobre los que pueda actuar, como son los botones, selectores o modificadores de datos e incluso leds que cambian de intensidad. Por otro lado, estará la parte de código que indicará la acción que se debe realizar tras el uso que el usuario haga sobre la interfaz o en el caso en que la aplicación reciba un mensaje del *bróker* indicando que se ha producido algún cambio en los parámetros y que, por lo tanto, no se debe usar el comando que estaba indicado.

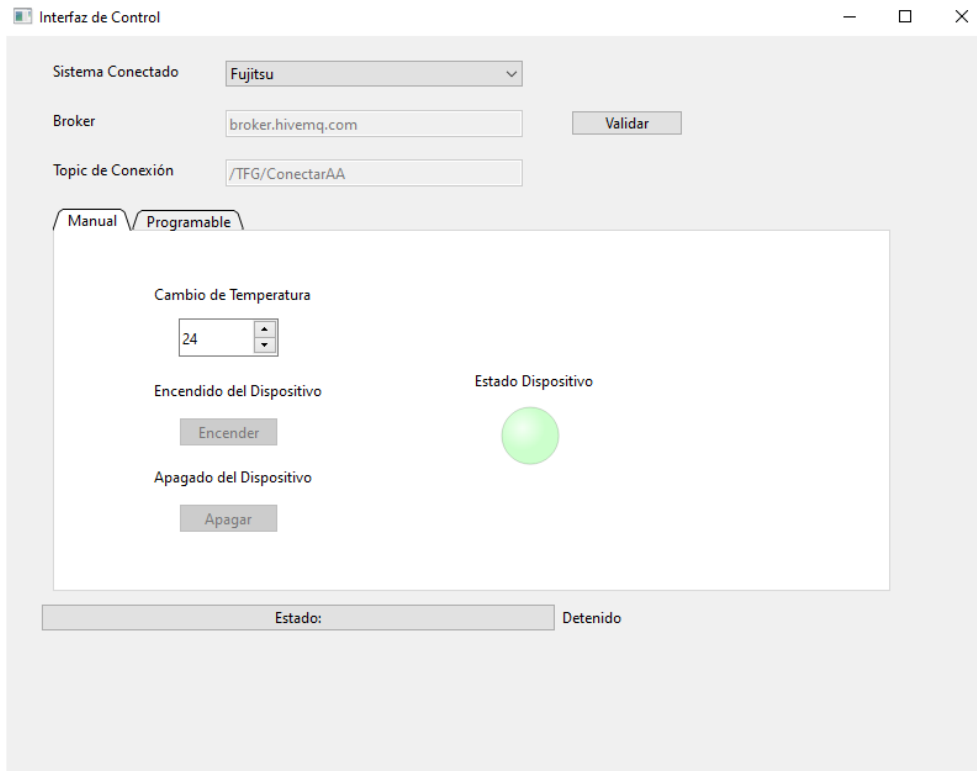


Figura 3.30. Estado inicial de la interfaz de usuario en modo manual.

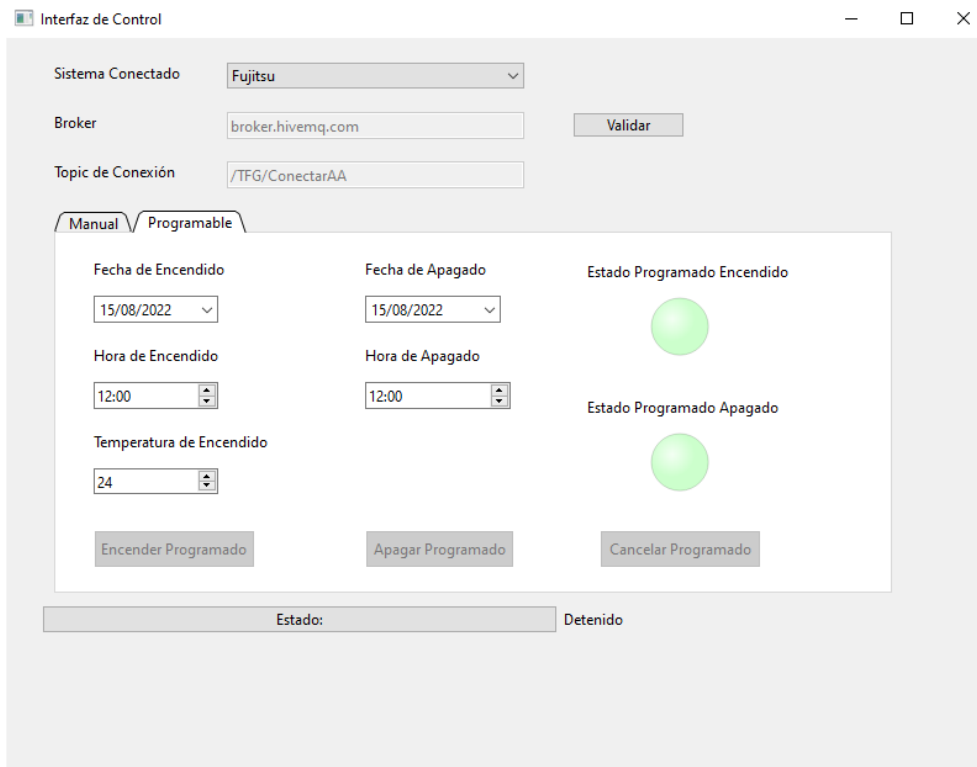


Figura 3.31. Estado inicial de la interfaz de usuario en modo manual.

La aplicación consistirá en una ventana en el ordenador del usuario, la cual permitirá realizar las funciones descritas en los casos de uso que aparecen en la Figura 3.32. Esta aplicación constará de dos partes de control, una primera que será la manual y que permitirá el envío de las acciones directas que tiene que realizar el *split*. Por otro lado, contará con un modo de funcionamiento programable que permitirá realizar una configuración anticipada del dispositivo indicando la fecha y la hora a la que se deberá realizar la función de encendido o de apagado del dispositivo. Estos mensajes que se envían desde la interfaz hacia el módulo encargado de la recepción de los mensajes se realizarán mediante un *broker* que permitirá la comunicación bidireccional entre estos dispositivos, ya que la interfaz también será capaz de recibir mensajes de la placa y con los datos recibidos permitir que se realice la acción requerida o denegar su ejecución y provocar que salte un aviso para el usuario del motivo que le impide que esto suceda.

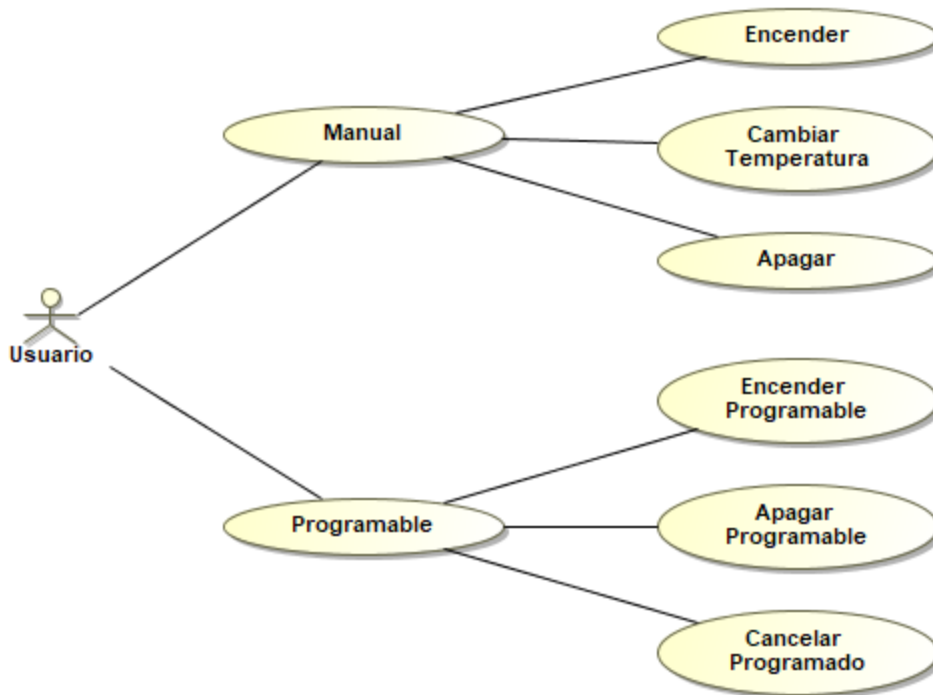


Figura 3.32. Casos de uso.

Más adelante, se expondrán pequeños fragmentos de código que ayudarán a definir el funcionamiento que posee la aplicación en sus diferentes modos de funcionamiento. En la Figura 3.33 se muestra la manera en la que se han generado las estructuras de los mensajes de tipo JSON para poder enviar los datos necesarios desde la aplicación de usuario hacia el *broker* y posteriormente desde él hacia el módulo principal, que será el encargado de recibirlo y realizar las acciones necesarias en función de los datos recibidos. En esta imagen se observa cuáles son los campos que contiene este tipo de archivo JSON y cómo se inicializa cada uno de ellos.

```

89     QJsonObject objeto_json;
90     objeto_json["Marca_Aire"] = marca_aire;
91
92     if(mes_enc_man.val_ejec == 1 && mes_cam_temp.val_ejec == 1){
93         objeto_json["Encendido"] = mes_enc_man.val_ejec;
94         objeto_json["Cambio_Temperatura"] = mes_cam_temp.value_temp;
95     }
96     else if(mes_apa_man.val_ejec == 1){
97         objeto_json["Apagado"] = mes_apa_man.val_ejec;
98         objeto_json["Cambio_Temperatura"] = 0;
99     }
100
101     if(mes_enc_prog.val_ejec == 1){
102         objeto_json["Encendido_Programado_Dia"] = mes_enc_prog.val_fecha_dia;
103         objeto_json["Encendido_Programado_Mes"] = mes_enc_prog.val_fecha_mes;
104         objeto_json["Encendido_Programado_Ano"] = mes_enc_prog.val_fecha_ano;
105         objeto_json["Encendido_Programado_Hora"] = mes_enc_prog.val_reloj_hora;
106         objeto_json["Encendido_Programado_Min"] = mes_enc_prog.val_reloj_min;
107         objeto_json["Encendido_Programado_Temp"] = mes_enc_prog.val_temp;
108         objeto_json["Encendido_Programado_Ejec"] = mes_enc_prog.val_ejec;
109     }
110
111     if(mes_apa_prog.val_ejec == 1){
112         objeto_json["Apagado_Programado_Dia"] = mes_apa_prog.val_fecha_dia;
113         objeto_json["Apagado_Programado_Mes"] = mes_apa_prog.val_fecha_mes;
114         objeto_json["Apagado_Programado_Ano"] = mes_apa_prog.val_fecha_ano;
115         objeto_json["Apagado_Programado_Hora"] = mes_apa_prog.val_reloj_hora;
116         objeto_json["Apagado_Programado_Min"] = mes_apa_prog.val_reloj_min;
117         objeto_json["Apagado_Programado_Ejec"] = mes_apa_prog.val_ejec;
118     }
119
120     QJsonDocument mensaje(objeto_json); //crea un objeto de tipo QJsonDocument c
121     MQTT::Message msg(0, ui->Topic_Conex->text(), mensaje.toJson()); //Crea el

```

Figura 3.33. Generación de un archivo JSON.

En cambio, en Figura 3.34 se puede observar cómo se realizaría la lectura de los datos proveniente desde el servidor y la forma en la que se deben tratar estos mensajes para poder obtener la información que se transmiten en ellos para poder luego realizar las acciones correspondientes.

```
151     bool previousblockinstate;  
152     double checked_ventana, checked_estado;  
153     if (connected)  
154     {  
155         QJsonParseError error;  
156         QJsonDocument mensaje=QJsonDocument::fromJson(message.payload(),&error);  
157  
158         if ((error.error==QJsonParseError::NoError)&&(mensaje.isObject()))  
159         { //Tengo que comprobar que el mensaje es del tipo adecuado y no hay errores de parseo...  
160  
161             QJsonObject objeto_json = mensaje.object();  
162             QJsonValue entrada_ventana = objeto_json["EstadoVentana"]; //Devuelve el estado en el que se encuentra la ventana  
163             QJsonValue estado_aire = objeto_json["EstadoAire"]; //Devuelve el estado en el que esta el aire,  
164                             //teniendo en cuenta si esta encendido/apagado y  
165                             //la temperatura a la que esta en ese momento  
166  
167             mes_enc_man.est_ventana = entrada_ventana.toDouble();
```

Figura 3.34. Lectura de datos del bróker.

Capítulo 4. Verificación y pruebas

En este capítulo se va a mostrar cómo se han realizado las pruebas para determinar el correcto funcionamiento de todo el sistema, desde el funcionamiento de cada uno de los módulos hasta las comunicaciones desde la aplicación con el módulo principal a través del *broker* para poder controlar el *split* del aire acondicionado. Para conocer si todo el sistema realiza todas las acciones que debe se ha preparado un sistema de pruebas sobre el cual se realizarán las verificaciones de cumplimiento de todos los requisitos.

El sistema comenzará su funcionamiento tras realizar la conexión de los distintos módulos a una fuente de alimentación. Una vez las distintas placas se encuentran alimentadas, se procederá a configurarse cada uno de los parámetros internos indicados en cada uno de ellos, además, se esperará a aproximadamente 1 minuto para determinar que las lecturas realizadas por el sensor de movimiento son las correctas, ya que este necesita de ese tiempo para adaptarse a las condiciones del entorno para comenzar a funcionar adecuadamente. Por otro lado, la aplicación generada para el control del *split* se encontrará en su estado inicial, Figura 4.1 para el modo manual y Figura 4.2 para el modo programable, desde el que se podrá empezar a realizar las configuraciones deseadas sobre el dispositivo. Tras haberse configurado los módulos y los sensores se puede comenzar a realizar las pruebas para verificar que se cumplen todos los requisitos propuestos para el proyecto.

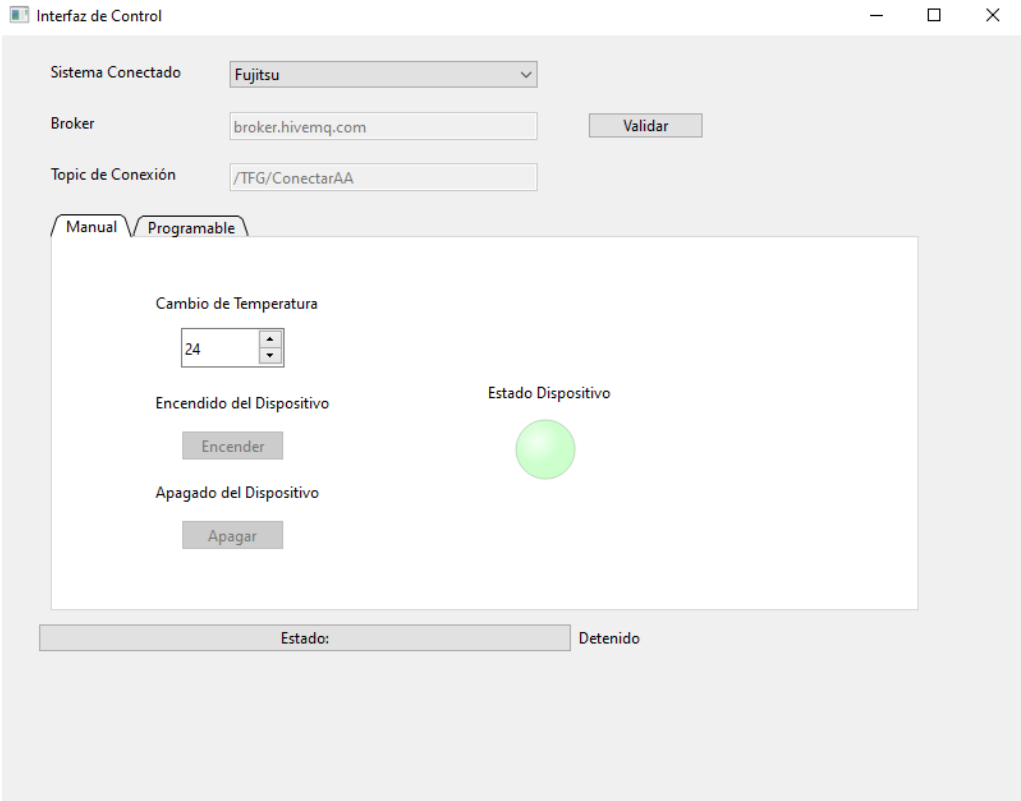


Figura 4.1. Estado inicial de la interfaz de usuario en modo manual.

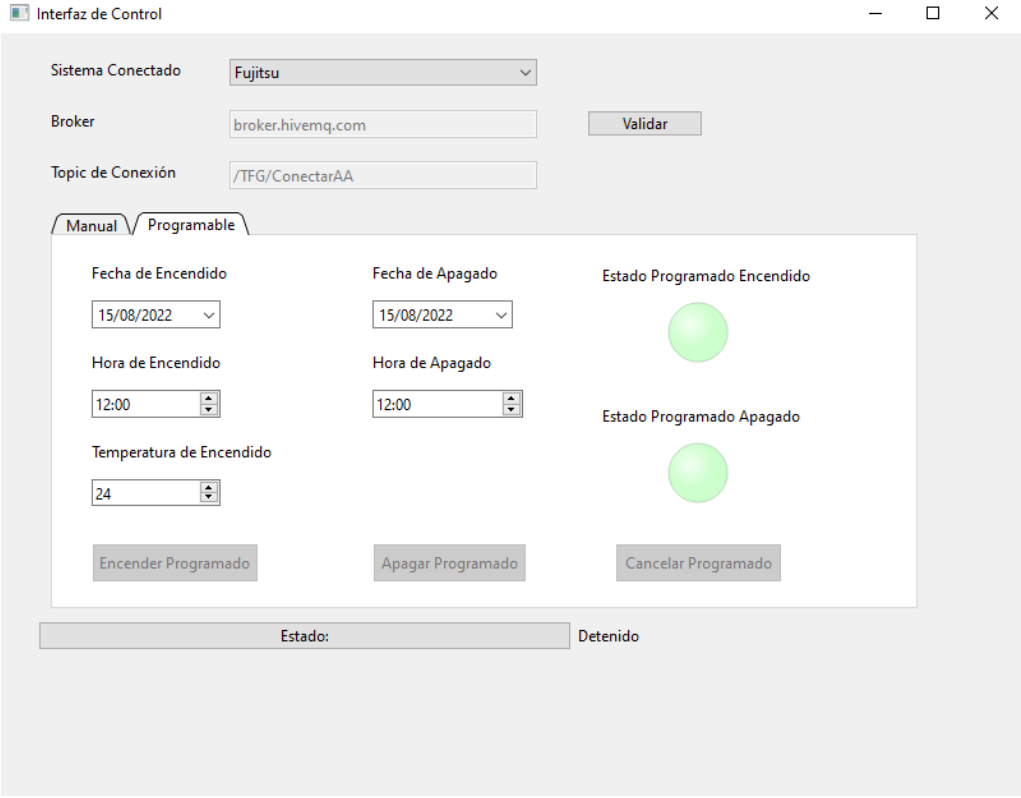


Figura 4.2. Estado inicial de la interfaz de usuario en modo programado.

4.1. Sistema de pruebas

El sistema de pruebas estará constituido por varios bloques que determinarán que tanto la aplicación cómo los distintos módulos que componen el sistema realizan las acciones pertinentes de cada uno.

El primer bloque estará constituido por la aplicación de compilación de código de Arduino, mediante el cual se han podido generar todos los códigos necesarios para los distintos microcontroladores y asignarles así su funcionamiento. A través de este programa, se ha podido visualizar mediante una ventana generada por el monitor serie, y tras incluir algunas líneas en el código que imprimieran los distintos valores que se querían observar mediante esta función, se pueden observar si los datos que están saliendo son los deseados o si es necesario realizar nuevas modificaciones.

Junto a esta aplicación, también ha sido necesario la utilización de otro programa, llamado MQTTfx, mediante el que se ha podido visualizar cuáles son los datos que se envía en cada iteración hacia el *broker* ubicado en el servidor de MQTT mediante los dos *topics* que se generaron para la transmisión de datos, uno de ellos que solo es recibido por el módulo principal, el cuál es “/TFG/ConectarAA”, y otro de ellos que solo es recibido por la aplicación que hace de interfaz de usuario, que es “/TFG/Estados”.

Por último, se ha utilizado para la realización de pruebas un diseño externo, compuesto por un receptor IR, como se puede observar en la Figura 4.3, el cuál ha servido para dos funciones, por un lado, para la generación de los códigos leídos de los distintos mandos de control remoto de las marcas que se ha decidido utilizar, y por otro lado, para este sistema de pruebas, determinando si los valores que se envían desde el módulo principal donde está situado el led IR son aquellos que corresponden a las ejecuciones determinadas por el mando de cada dispositivo.

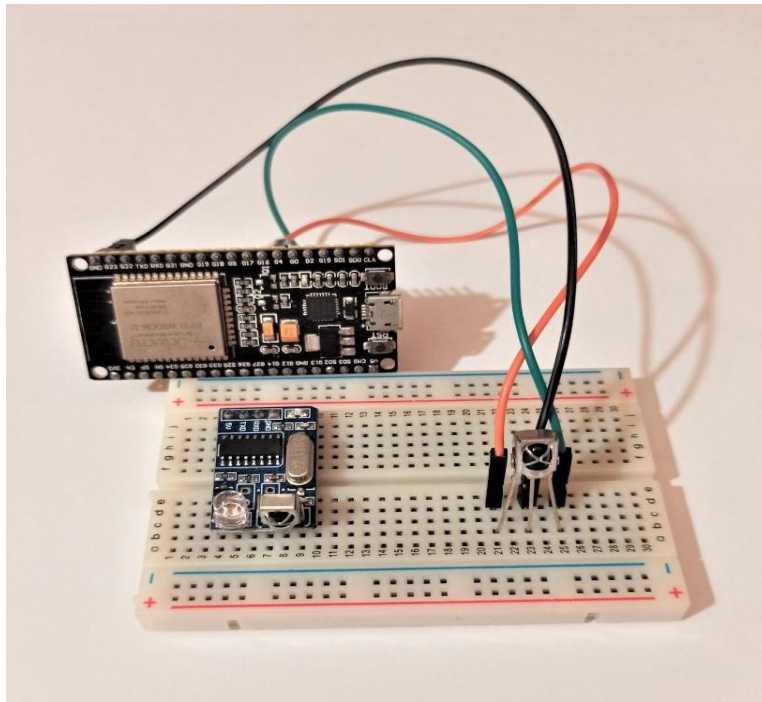


Figura 4.3. Sensor receptor IR.

4.2. Pruebas realizadas

Se comenzará con las pruebas realizadas a los módulos compuestos con los distintos sensores. Las pruebas realizadas sobre estos dispositivos fueron realizadas a través de la aplicación de Arduino procurando ver en el monitor serie si los valores que eran leídos por los sensores eran adecuados y eran acordes con lo que sucedía en el entorno, de igual modo también se probó que estas placas ESP32 enviaran los datos al módulo central. Se puede observar en la Figura 4.4 cómo se consigue leer los datos recogidos por los sensores de temperatura y humedad y por el sensor de movimiento, mientras que en la Figura 4.5 se muestra el estado del sensor de proximidad. En ambos casos se observa que los datos son acordes al estado en el que se encontraban los sensores y los parámetros externos que se encontraban a su alrededor.



Figura 4.4. Lecturas del sensor de temperatura y humedad y del sensor de movimiento.

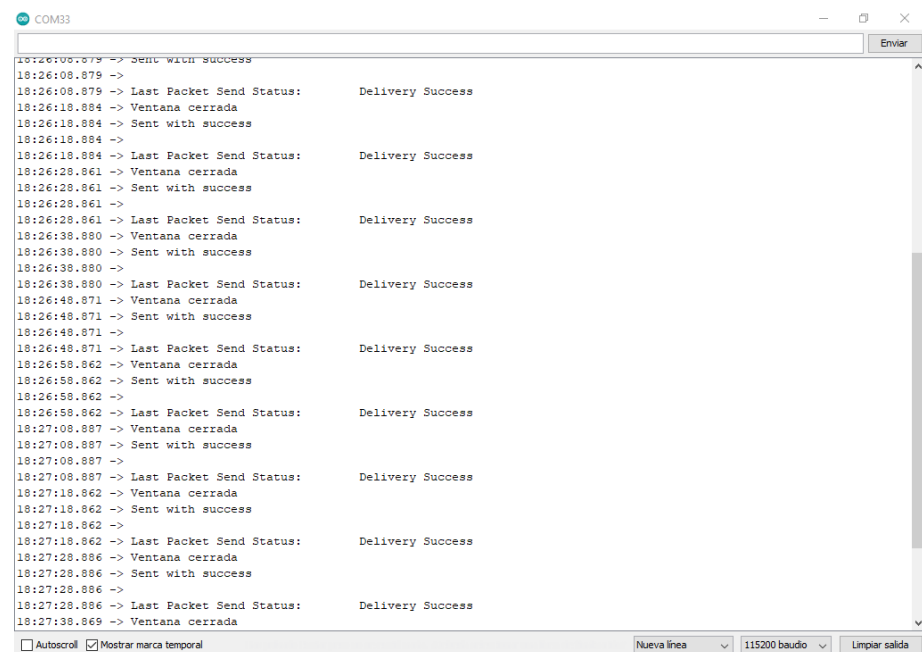


Figura 4.5 Lecturas del sensor de proximidad.

A continuación, se mostrarán las pruebas realizadas sobre el módulo principal para poder determinar que el envío de las señales IR transmitidas por el led IR ubicado en la placa pretaladrada son los mismos datos que va a recibir el *split* del aire acondicionado. Esto se comprobó con el receptor IR que se muestra en

Figura 4.3. Se realizó el envío de varias señales IR procedente de diferentes marcas y con diferentes acciones, de manera que se visualizara el funcionamiento de todas las marcas, ya que hay diferencias de funcionamiento entre ellas. En la Figura 4.6 se puede visualizar cuales son los códigos IR transmitidos por el módulo principal, enviándose mensajes en primer lugar, de la marca Panasonic, en segundo lugar, de la marca Mundo Clima y por último, de la marca Fujitsu. Mientras que en la Figura 4.7 se observa cuáles son los datos recibidos por el receptor, si se fijan en las últimas señales que se muestran en cada uno, se puede ver cómo son las mismas, dando por verificada esta prueba.

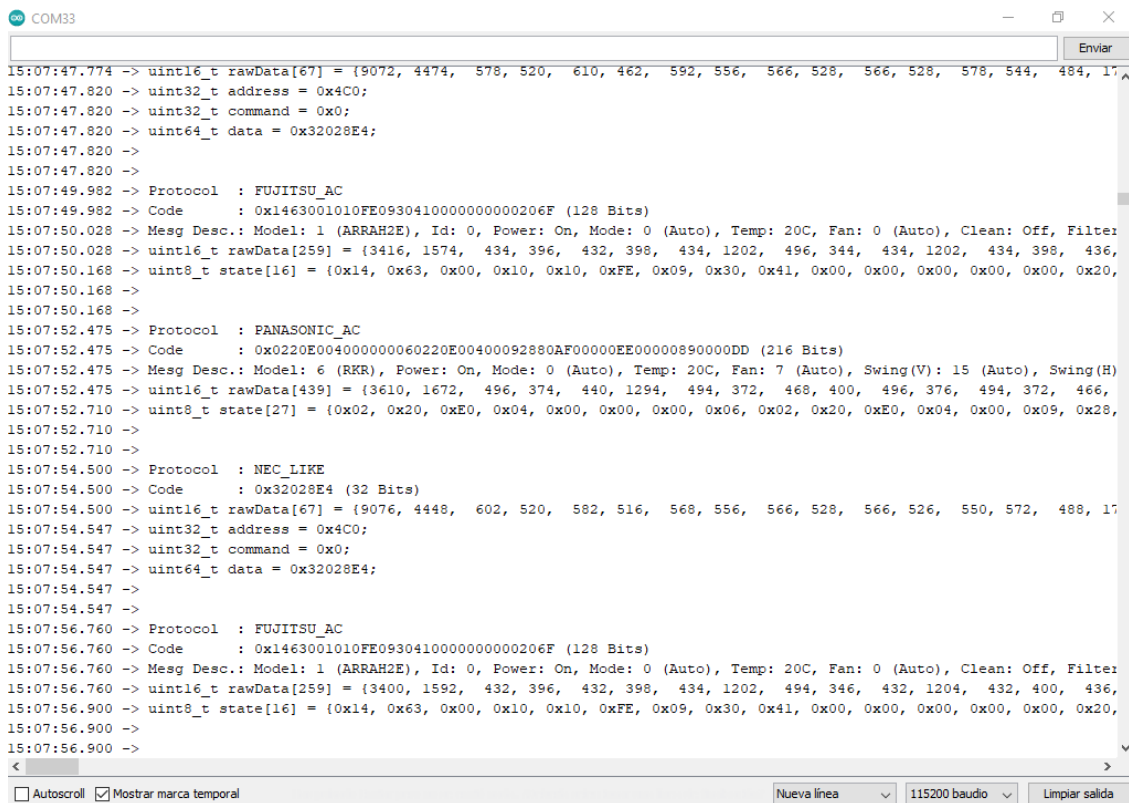
COM5

```

15:07:18.352 ->
15:07:20.605 -> Este es el code que se deberia de enviar (MC):9046, 4470, 534, 560, 566, 528, 568, 552, 566, 526, 5
15:07:20.605 ->
15:07:22.677 -> Este es el code que se deberia de enviar(FUJ): 3348, 1610, 436, 392, 434, 394, 434, 1200, 434, 396,
15:07:22.772 ->
15:07:24.936 -> Este es el code que se deberia de enviar (PAN): 3530, 1722, 444, 422, 444, 1288, 494, 372, 444, 422
15:07:25.123 ->
15:07:27.337 -> Este es el code que se deberia de enviar (MC):9046, 4470, 534, 560, 566, 528, 568, 552, 566, 526, 5
15:07:27.384 ->
15:07:29.446 -> Este es el code que se deberia de enviar(FUJ): 3348, 1610, 436, 392, 434, 394, 434, 1200, 434, 396,
15:07:29.539 ->
15:07:31.654 -> Este es el code que se deberia de enviar (PAN): 3530, 1722, 444, 422, 444, 1288, 494, 372, 444, 422
15:07:31.888 ->
15:07:34.094 -> Este es el code que se deberia de enviar (MC):9046, 4470, 534, 560, 566, 528, 568, 552, 566, 526, 5
15:07:34.141 ->
15:07:36.210 -> Este es el code que se deberia de enviar(FUJ): 3348, 1610, 436, 392, 434, 394, 434, 1200, 434, 396,
15:07:36.304 ->
15:07:38.421 -> Este es el code que se deberia de enviar (PAN): 3530, 1722, 444, 422, 444, 1288, 494, 372, 444, 422
15:07:38.608 ->
15:07:40.870 -> Este es el code que se deberia de enviar (MC):9046, 4470, 534, 560, 566, 528, 568, 552, 566, 526, 5
15:07:40.916 ->
15:07:42.938 -> Este es el code que se deberia de enviar(FUJ): 3348, 1610, 436, 392, 434, 394, 434, 1200, 434, 396,

```

Figura 4.6. Valores transmitidos por el módulo principal.



```

15:07:47.774 -> uint16_t rawData[67] = {9072, 4474, 578, 520, 610, 462, 592, 556, 566, 528, 566, 528, 578, 544, 484, 11...
15:07:47.820 -> uint32_t address = 0x4C0;
15:07:47.820 -> uint32_t command = 0x0;
15:07:47.820 -> uint64_t data = 0x32028E4;
15:07:47.820 ->
15:07:49.982 -> Protocol : FUJITSU_AC
15:07:49.982 -> Code : 0x1463001010FE0930410000000000206F (128 Bits)
15:07:50.028 -> Mesg Desc.: Model: 1 (ARRAH2E), Id: 0, Power: On, Mode: 0 (Auto), Temp: 20C, Fan: 0 (Auto), Clean: Off, Filter
15:07:50.028 -> uint16_t rawData[259] = {3416, 1574, 434, 396, 432, 398, 434, 1202, 496, 344, 434, 1202, 434, 398, 436,
15:07:50.168 -> uint8_t state[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20,
15:07:50.168 ->
15:07:50.168 ->
15:07:52.475 -> Protocol : PANASONIC_AC
15:07:52.475 -> Code : 0x0220E0040000000060220E00400092880AF00000EE00000890000DD (216 Bits)
15:07:52.475 -> Mesg Desc.: Model: 6 (RKR), Power: On, Mode: 0 (Auto), Temp: 20C, Fan: 7 (Auto), Swing(V): 15 (Auto), Swing(H)
15:07:52.475 -> uint16_t rawData[439] = {3610, 1672, 496, 374, 440, 1294, 494, 372, 468, 400, 496, 376, 494, 372, 466,
15:07:52.710 -> uint8_t state[27] = {0x02, 0x20, 0xE0, 0x04, 0x00, 0x00, 0x00, 0x06, 0x02, 0x20, 0xE0, 0x04, 0x00, 0x09, 0x28,
15:07:52.710 ->
15:07:52.710 ->
15:07:54.500 -> Protocol : NEC LIKE
15:07:54.500 -> Code : 0x32028E4 (32 Bits)
15:07:54.500 -> uint16_t rawData[67] = {9076, 4448, 602, 520, 582, 516, 568, 556, 566, 528, 566, 526, 550, 572, 488, 11...
15:07:54.547 -> uint32_t address = 0x4C0;
15:07:54.547 -> uint32_t command = 0x0;
15:07:54.547 -> uint64_t data = 0x32028E4;
15:07:54.547 ->
15:07:54.547 ->
15:07:56.760 -> Protocol : FUJITSU_AC
15:07:56.760 -> Code : 0x1463001010FE0930410000000000206F (128 Bits)
15:07:56.760 -> Mesg Desc.: Model: 1 (ARRAH2E), Id: 0, Power: On, Mode: 0 (Auto), Temp: 20C, Fan: 0 (Auto), Clean: Off, Filter
15:07:56.760 -> uint16_t rawData[259] = {3400, 1592, 432, 396, 432, 398, 434, 1202, 494, 346, 432, 1204, 432, 400, 436,
15:07:56.900 -> uint8_t state[16] = {0x14, 0x63, 0x00, 0x10, 0x10, 0xFE, 0x09, 0x30, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20,
15:07:56.900 ->
15:07:56.900 ->

```

Figura 4.7. Valores recibidos por el receptor IR.

La siguiente prueba se volverá a realizar sobre el módulo principal, aunque esta vez será sobre su configuración y recepción de mensajes del *broker* que se encuentra en el servidor de MQTT. Para ello, será necesario abrir varias aplicaciones, por un lado, estará el programa de MQTTfx suscrito a uno de los *topics*, para observar que los mensajes que se envían son los mismos que se reciben en el microcontrolador, por otro lado, se encontrará la interfaz de usuario, que será la encargada de realizar la transmisión de los mensajes. A esto, se incluirá la recepción de datos de los otros módulos, como se comentó con anterioridad, se pueden producir problemas si no se configuran bien el sistema de Wifi y el protocolo ESP-NOW para el intercambio de datos entre los módulos.

Para conocer si llegan todos los mensajes de la manera que se desea se realizará la pruebas varias veces y con diferentes envíos de mensaje, de modo que se pueda observar cómo son los mensajes que le llegan al módulo principal tras la pulsación de cada uno de los botones dispuestos en la interfaz de usuario, junto a los mensajes recibidos del *broker* se seguirán recibiendo y mostrando los

mensajes procedentes de los demás módulos compuestos por los distintos sensores.

Se comenzará con la configuración manual de la interfaz de usuario, procediendo a pulsar en primer lugar el botón de Encendido del sistema, en la Figura 4.8 cómo se pone a 1 el valor del Encendido Manual y se adquiere un valor de temperatura, según se encuentre en la interfaz. Este valor de la temperatura podrá ser modificado durante la ejecución siempre que se encuentre el dispositivo en encendido, si este se encuentra en apagado no se realizará ninguna acción.

```
17:09:56.277 -> Message arrived [/TFG/ConectarAA]
17:09:56.277 -> Este es el code que se deberia de enviar (PAN): 3530, 1720, 444, 424, 442, 1288, 444, 422,
17:09:56.510 ->
17:09:56.699 -> Marca Aire: 1
17:09:56.745 ->
17:09:56.745 -> Valor Encendido Manual: 1
17:09:56.745 ->
17:09:56.745 -> Valor Apagado Manual: 0
17:09:56.745 ->
17:09:56.745 -> Valor Temperatura Manual: 18
17:09:56.745 -> 0
17:09:56.745 ->
17:09:56.745 -> Valores Encendido Programado
17:09:56.745 -> Fecha:
17:09:56.745 -> 0/0/0
17:09:56.745 -> Tiempo:
17:09:56.745 -> 0:0
17:09:56.745 -> Valor de Temperatura: 0
17:09:56.745 -> Valor de Ejecucion: 0
17:09:56.745 ->
17:09:56.745 -> Valores Apagado Programado
17:09:56.745 -> Fecha:
17:09:56.745 -> 0/0/0
17:09:56.745 -> Tiempo:
17:09:56.745 -> 0:0
17:09:56.745 -> Valor de Ejecucion: 0
17:09:56.745 -> Fin Recogido Callback
17:09:56.745 ->
17:09:58.383 -> Packet received from: c8:c9:a3:c5:dc:ec
17:09:58.383 -> Board ID 2: 12 bytes
17:09:58.383 -> Ventana cerrada
17:09:58.383 ->
17:09:58.709 -> Packet received from: c8:c9:a3:c7:5b:90
17:09:58.709 -> Board ID 1: 20 bytes
```

Figura 4.8. Recepción de mensajes del *broker* y mediante ESP-NOW. Configuración de Encendido Manual.

En el caso de que se pulse el botón de Apagado, realizará un cambio en los datos transmitidos en el mensaje, ya que los valores de Encendido Manual y Temperatura Manual pasarán a ser 0, mientras que Apagado Manual valdrá 1, con lo cual, se producirá un apagado instantáneo del *split*. Este cambio que se produce en los mensajes se puede visualizar en la Figura 4.9.

```

17:12:48.789 -> Packet received from: c8:c9:a3:c7:5b:90
17:12:48.789 -> Board ID 1: 20 bytes
17:12:48.789 -> Temperatura: 28°C
17:12:48.789 -> Humedad: 40
17:12:48.789 -> No se ha detectado movimiento!
17:12:48.789 ->
17:12:56.754 -> Message arrived [/TFG/ConectarAA]
17:12:56.754 -> Este es el code que se debería de enviar (PAN): 3528, 1722, 492, 374, 444, 1288, 444, 422, 444,
17:12:56.940 ->
17:12:57.174 -> Marca Aire: 1
17:12:57.174 ->
17:12:57.174 -> Valor Encendido Manual: 0
17:12:57.174 ->
17:12:57.174 -> Valor Apagado Manual: 1
17:12:57.174 ->
17:12:57.174 -> Valor Temperatura Manual: 0
17:12:57.174 -> 0
17:12:57.174 ->
17:12:57.174 -> Valores Encendido Programado
17:12:57.174 -> Fecha:
17:12:57.174 -> 0/0/0
17:12:57.174 -> Tiempo:
17:12:57.174 -> 0:0
17:12:57.174 -> Valor de Temperatura: 0
17:12:57.221 -> Valor de Ejecucion: 0
17:12:57.221 ->
17:12:57.221 -> Valores Apagado Programado
17:12:57.221 -> Fecha:
17:12:57.221 -> 0/0/0
17:12:57.221 -> Tiempo:
17:12:57.221 -> 0:0
17:12:57.221 -> Valor de Ejecucion: 0
17:12:57.221 -> Fin Recogido Callback
17:12:57.221 ->

```

Figura 4.9. Recepción de mensajes del *broker* y mediante ESP-NOW. Configuración de Apagado Manual.

Con estas pruebas se terminaría la configuración manual y pasaríamos a la programable, la cual posee también los modos de encendido y apagado, con la diferencia de que en estos casos se puede elegir una fecha y un momento del día para que realicen la acción indicada. En la Figura 4.10, se puede observar cómo sería el mensaje transmitido una vez se han seleccionado todos los parámetros y se ha pulsado únicamente el botón de Encendido Programable.

```

17:16:08.889 -> Packet received from: c8:c9:a3:c7:5b:90
17:16:08.889 -> Board ID 1: 20 bytes
17:16:08.889 -> Temperatura: 28°C
17:16:08.889 -> Humedad: 40
17:16:08.889 -> Movimiento detectado!
17:16:08.889 ->
17:16:10.562 -> Message arrived [/TFG/ConectarAA]
17:16:10.562 -> Este es el code que se debería de enviar (PAN): 3530, 1722, 494, 346, 518, 1238, 494, 372,
17:16:10.795 ->
17:16:11.028 -> Marca Aire: 1
17:16:11.028 ->
17:16:11.028 -> Valor Encendido Manual: 1
17:16:11.028 ->
17:16:11.028 -> Valor Apagado Manual: 0
17:16:11.028 ->
17:16:11.028 -> Valor Temperatura Manual: 23
17:16:11.028 -> 0
17:16:11.028 ->
17:16:11.028 -> Valores Encendido Programado
17:16:11.028 -> Fecha:
17:16:11.028 -> 25/9/2022
17:16:11.028 -> Tiempo:
17:16:11.028 -> 5:9
17:16:11.028 -> Valor de Temperatura: 30
17:16:11.028 -> Valor de Ejecucion: 1
17:16:11.028 ->
17:16:11.028 -> Valores Apagado Programado
17:16:11.028 -> Fecha:
17:16:11.028 -> 0/0/0
17:16:11.028 -> Tiempo:
17:16:11.028 -> 0:0
17:16:11.028 -> Valor de Ejecucion: 0
17:16:11.028 -> Fin Recogido Callback
17:16:11.028 ->

```

Figura 4.10. Recepción de mensajes del *broker* y mediante ESP-NOW. Configuración de Encendido Programado.

Una vez se ha visualizado el Encendido Programado, se procederá a eliminar los valores enviados con el botón de Cancelar Programado y se realizará la configuración de los parámetros que intervienen en el Apagado Programado, que vuelven a ser la fecha y la hora a la que se desea efectuar la acción en el *split*, el mensaje transmitido en este caso se puede observar en la Figura 4.11.


```

17:18:48.993 -> Packet received from: c8:c9:a3:c7:5b:90
17:18:48.993 -> Board ID 1: 20 bytes
17:18:48.993 -> Temperatura: 28°C
17:18:48.993 -> Humedad: 40
17:18:48.993 -> No se ha detectado movimiento!
17:18:48.993 ->
17:18:55.799 -> Message arrived [/TFG/ConectarAA]
17:18:55.799 -> Este es el code que se deberia de enviar (PAN): 3528, 1722, 492, 374, 444, 1288, 444, 422,
17:18:55.987 ->
17:18:56.219 -> Marca Aire: 1
17:18:56.219 ->
17:18:56.219 -> Valor Encendido Manual: 0
17:18:56.219 ->
17:18:56.219 -> Valor Apagado Manual: 1
17:18:56.219 ->
17:18:56.219 -> Valor Temperatura Manual: 0
17:18:56.219 -> 0
17:18:56.219 ->
17:18:56.219 -> Valores Encendido Programado
17:18:56.219 -> Fecha:
17:18:56.219 -> 0/0/0
17:18:56.219 -> Tiempo:
17:18:56.219 -> 0:0
17:18:56.219 -> Valor de Temperatura: 0
17:18:56.265 -> Valor de Ejecucion: 0
17:18:56.265 ->
17:18:56.265 -> Valores Apagado Programado
17:18:56.265 -> Fecha:
17:18:56.265 -> 1/12/2022
17:18:56.265 -> Tiempo:
17:18:56.265 -> 15:0
17:18:56.265 -> Valor de Ejecucion: 1
17:18:56.265 -> Fin Recogido Callback

```

Figura 4.11. Recepción de mensajes del *broker* y mediante ESP-NOW. Configuración de Apagado Programado.

Por último, se va tratar de realizar el envío simultáneo de los parámetros de Encendido Programado y Apagado Programado, para ello, se realizará la configuración de todos los valores que necesitan estas dos funciones y una vez que esto se haya ejecutado, se procederá a pulsar los botones de Encendido Programado y Apagado Programado en el orden que se desee, de manera que al final se transmita un mensaje mostrando unos resultados cómo aparecen en la Figura 4.12, pero con los valores determinados por el usuario en el caso correspondiente.

```
17:21:14.742 -> Message arrived [/TFG/ConectarAA]
17:21:14.742 -> Este es el code que se deberia de enviar (PAN): 3528, 1722, 492, 374, 444, 1288, 444, 422,
17:21:14.926 ->
17:21:15.161 -> Marca Aire: 1
17:21:15.161 ->
17:21:15.161 -> Valor Encendido Manual: 0
17:21:15.161 ->
17:21:15.161 -> Valor Apagado Manual: 1
17:21:15.161 ->
17:21:15.161 -> Valor Temperatura Manual: 0
17:21:15.161 -> 0
17:21:15.161 ->
17:21:15.161 -> Valores Encendido Programado
17:21:15.161 -> Fecha:
17:21:15.161 -> 15/9/2022
17:21:15.161 -> Tiempo:
17:21:15.161 -> 10:23
17:21:15.161 -> Valor de Temperatura: 22
17:21:15.161 -> Valor de Ejecucion: 1
17:21:15.161 ->
17:21:15.161 -> Valores Apagado Programado
17:21:15.161 -> Fecha:
17:21:15.161 -> 1/12/2022
17:21:15.161 -> Tiempo:
17:21:15.161 -> 15:0
17:21:15.161 -> Valor de Ejecucion: 1
17:21:15.161 -> Fin Recogido Callback
17:21:15.161 ->
17:21:18.376 -> Packet received from: c8:c9:a3:c5:dc:ec
17:21:18.376 -> Board ID 2: 12 bytes
17:21:18.376 -> Ventana cerrada
17:21:18.376 ->
```

Figura 4.12. Recepción de mensajes del *broker* y mediante ESP-NOW. Configuración de Encendido Programado y Apagado Programado.

Por último, aunque no menos importante se realizó una prueba sobre la interfaz de usuario que se ha generado, en la cual se pretendía que esta recibiera un mensaje del *broker* mediante el cual se va a provocar que salte una pestaña emergente del programa en el caso de que el sensor de proximidad de la ventana se encuentre abierto, esta ventana emergente se puede observar en la Figura 4.13. Esta transmisión del estado de la ventana se producirá desde el microcontrolador principal a través del segundo *topic*, que se ha generado exclusivamente para esta transmisión, hasta la aplicación que contiene la interfaz de usuario.

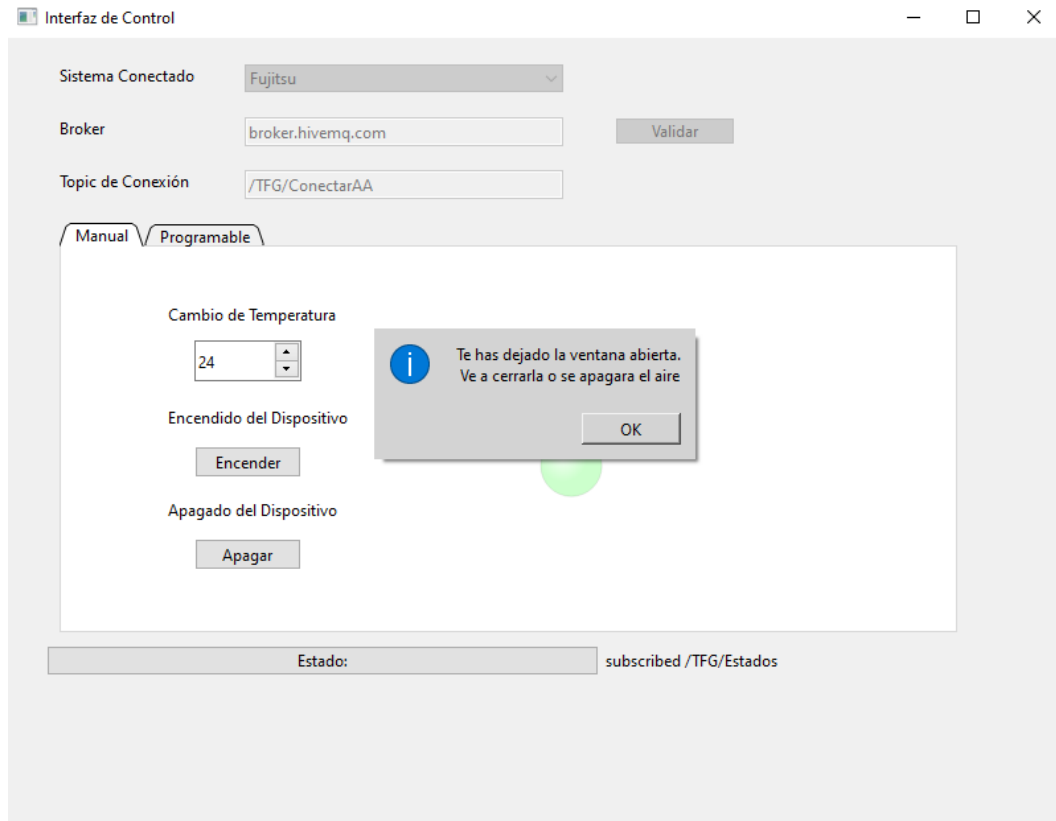


Figura 4.13. Ventana emergente por el estado de la ventana.

Tras concluir la realización de las pruebas, se puede verificar que se ha cumplido con todos los requisitos propuestos en la Tabla 2.1, y por lo tanto el sistema desarrollado puede considerarse un prototipo funcional.

Capítulo 5. Manual de instalación y uso

En este proyecto será necesario el desarrollo de dos manuales con los cuales el usuario sea capaz de poder realizar la instalación del programa que usará para el control del dispositivo y otro que le ayudará a comprender cómo es el programa que está utilizando y cómo ha de usarlo para que su funcionamiento sea óptimo.

5.1. Manual de instalación

En primer lugar, se encuentra el manual de instalación, el cual se dividirá en dos partes. Una primera parte, que será la instalación de los distintos microcontroladores junto con sus sensores, y una segunda que será la instalación del programa que usará el usuario para poder controlar el sistema de aire acondicionado.

Para la primera instalación, se deberá conocer cómo es la estancia de la vivienda donde se encontrará el dispositivo que se desea controlar. Una vez se tengan estos datos, se decidirá cuáles son los mejores lugares para colocar los distintos microcontroladores.

Empezaremos por el módulo principal, que está constituido por el microcontrolador ESP32 junto con el sensor IR, el cual, debe encontrarse situado a una distancia máxima de 2m, con visión directa, con respecto al receptor IR que se encuentra en el *split*, esto se puede observar en la Figura 5.1.



Figura 5.1. Módulo sensor de IR.

El segundo módulo estará formado por el segundo microcontrolador, además de los sensores de temperatura y movimiento, Figura 5.2. Este subsistema sería recomendable instalarlo en el techo de la habitación en la que se encuentre el dispositivo, de esta manera se conseguirán unas mejores lecturas de la temperatura y del movimiento en esa estancia, así el sensor de movimiento tendrá un mayor ángulo de visión.



Figura 5.2. Módulo sensor de movimiento y temperatura.

Por último, se instalará el tercer módulo, Figura 5.3, que está compuesto por el tercer microcontrolador y un sensor de proximidad. Este conjunto se deberá de colocar en una ventana o puerta, de manera que el sensor de proximidad pueda medir la apertura de dicha ventana o puerta según la distancia entre sus partes.



Figura 5.3. Módulo sensor de proximidad.

Para la segunda parte de la instalación, se encuentra la interfaz de usuario, que es una aplicación, la cual, solo es compatible en un ordenador con sistema operativo Windows.

Cuando iniciamos el programa, la primera acción que se deberá realizar, será la apertura del ejecutable, un archivo en formato '.exe'. Una vez el programa se haya abierto y se le haya dado permisos de administrador para poder ejecutarse, aparecerá el recuadro de la Figura 5.4, donde se deberá seleccionar el idioma en el cual se ejecutará el programa.

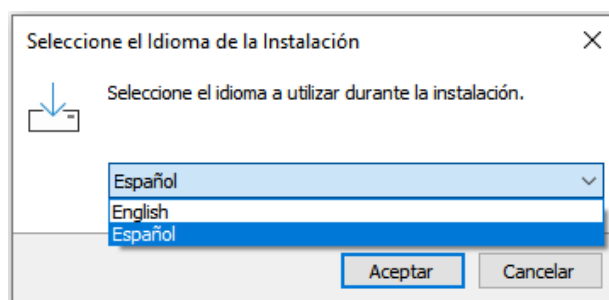


Figura 5.4. Selección de idioma.

A continuación, se encontrará con la ventana que se muestra en la Figura 5.5, en la cual, podrá seleccionar la ubicación en la que se realizará la instalación del programa, junto con todos los archivos que le corresponden para su correcto funcionamiento.

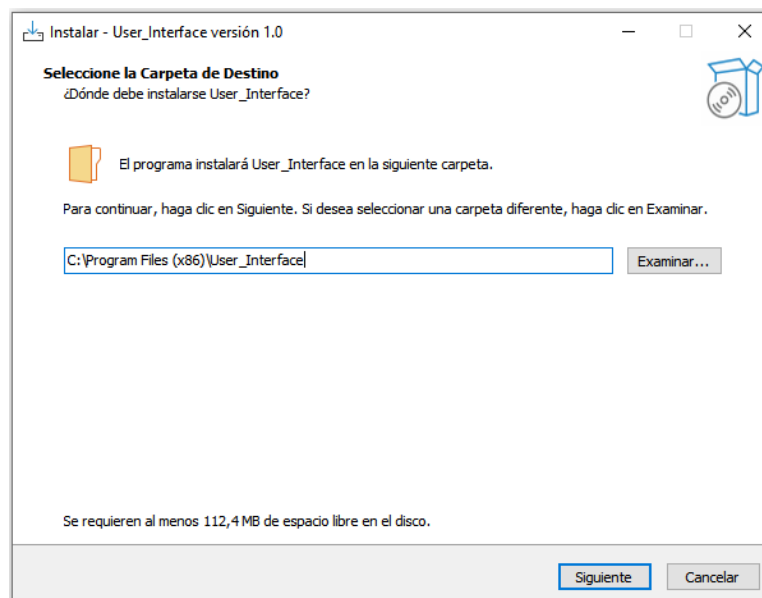


Figura 5.5. Selección de la ubicación de instalación.

A continuación, aparecerá la siguiente ventana, Figura 5 6, donde el usuario tendrá la opción si desea crear o no un acceso directo en el escritorio del ordenador para tener más facilidad a la hora de acceder al programa.

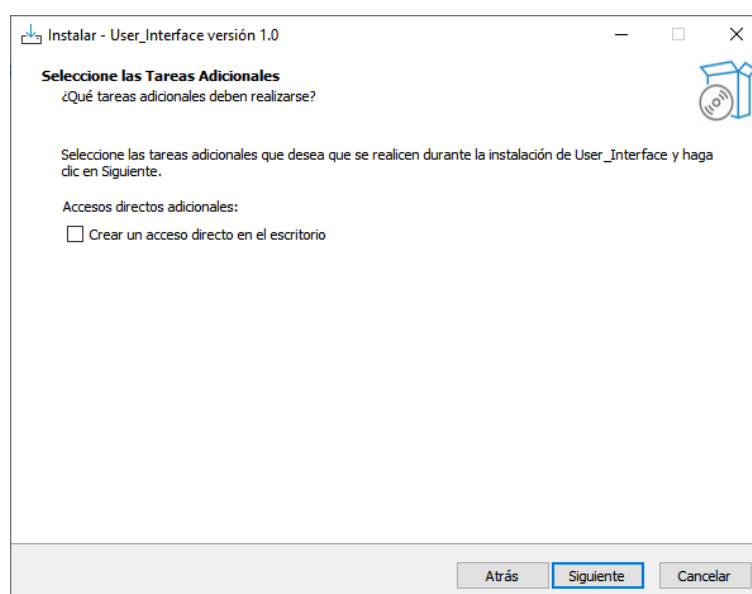


Figura 5 6. Creación del acceso directo en el escritorio.

Más adelante, se dejará ver la Figura 5.7, en la cual se pide la confirmación para realizar la instalación de la aplicación. Tras pulsar el botón de instalar, se cambiará la imagen de la ventana y comenzará la instalación, donde se podrá visualizar el avance mediante una barra de color verde.

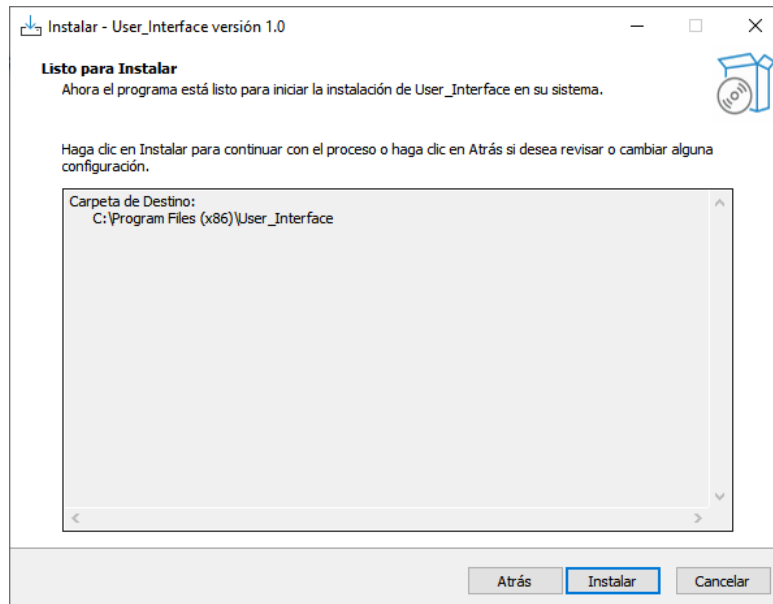


Figura 5.7. Confirmación de la instalación.

Por último, aparecerá la ventana, Figura 5.8, en la que se confirma que se ha realizado de manera correcta la instalación del programa, con la opción de ejecutarlo una vez finalizada la instalación.

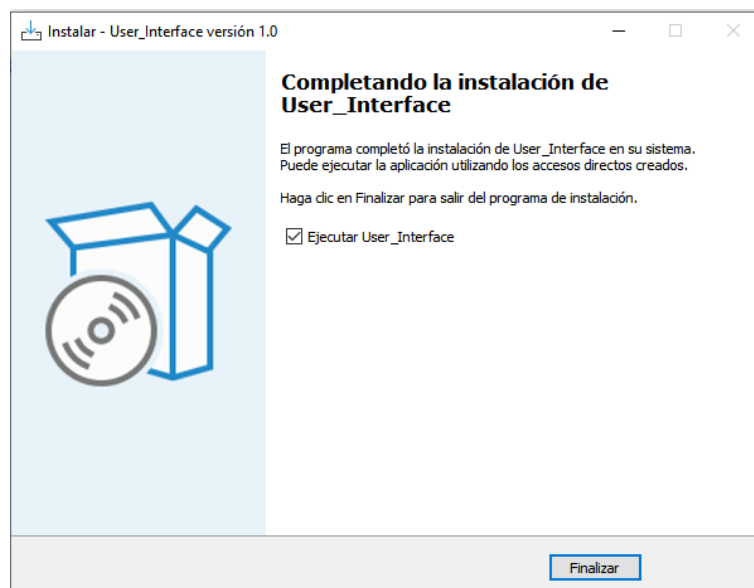


Figura 5.8. Fin de la instalación.

Una vez finalizada la instalación, debería aparecer una imagen similar a la que se muestra en la Figura 5.9, donde podremos comenzar a elegir los parámetros en los cuales se va a ejecutar el dispositivo de aire acondicionado. Tras lo cual, se daría por concluida la parte de instalación de la aplicación, que sería la interfaz del usuario con el dispositivo.

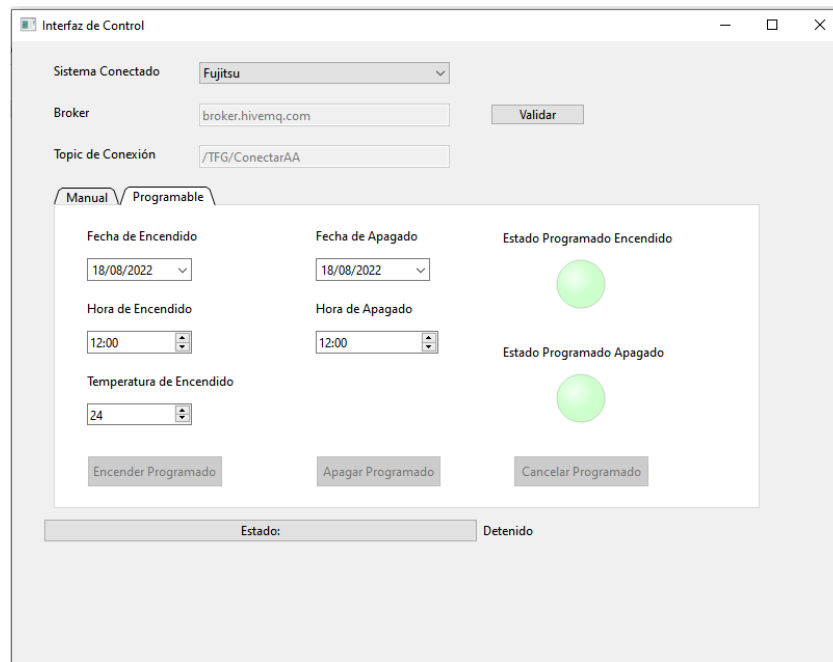


Figura 5.9. Interfaz de Usuario.

5.2. Manual de uso

En este apartado se procederá a realizar la explicación del funcionamiento de la aplicación que se ha desarrollado para conseguir controlar el *split* de un sistema de aire acondicionado.

Esta comenzará con la aparición de una ventana emergente como la que se muestra en la Figura 5.10. En esta ventana emergente se pueden observar distintas cosas, entre ellas se realiza un muestreo del nombre del *broker* y del *topic* de conexión que se utiliza para enviar la información necesaria. Justo encima de estos indicadores, se encuentra un desplegable, el cual, permite la selección de la marca del dispositivo que se desea usar. Dentro de estas marcas nos encontramos con: Mundo Clima, Panasonic y Fujitsu, esto se puede observar en la Figura 5.11.

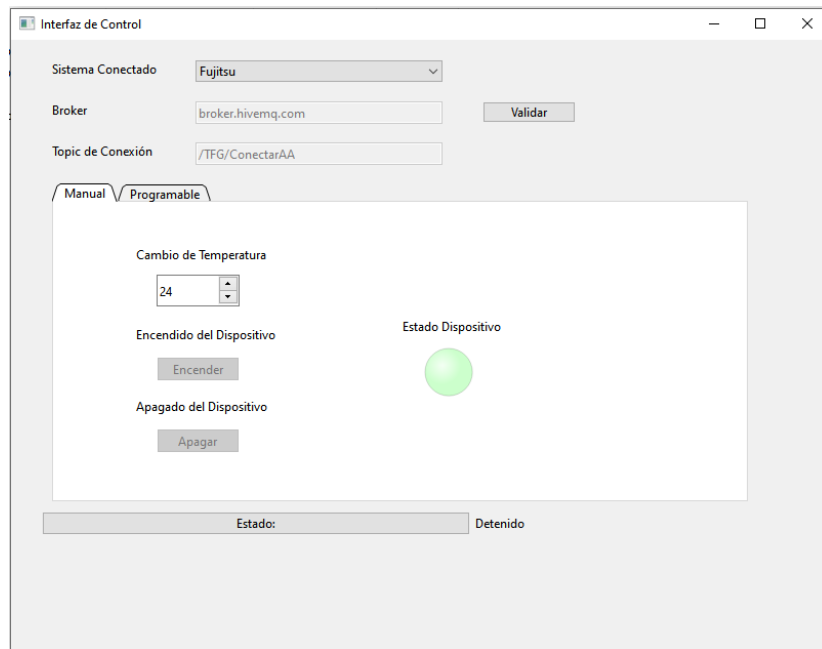


Figura 5.10. Interfaz de usuario para el control manual.

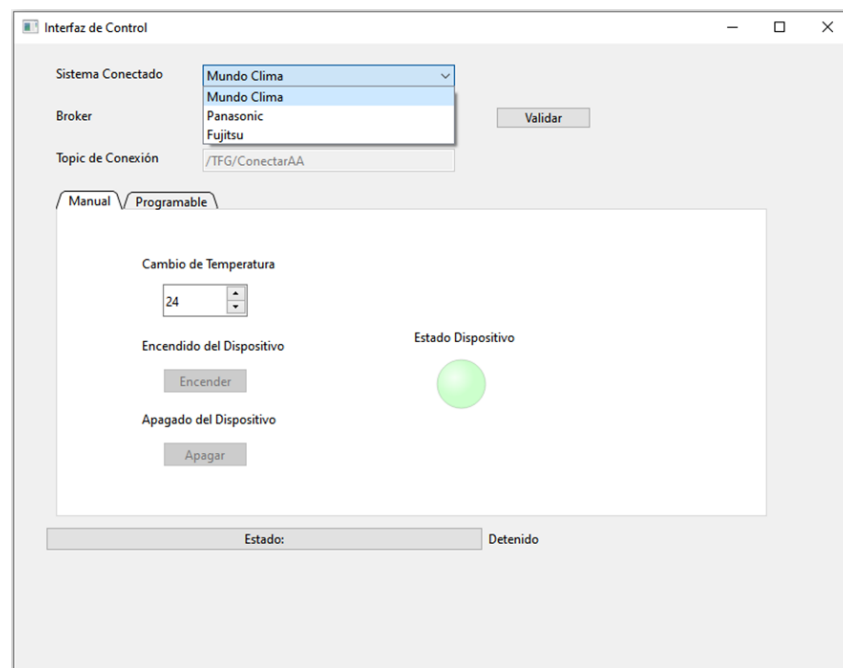


Figura 5.11. Desplegable de marcas.

Una vez que se ha seleccionado la marca del *split* de la que dispone el usuario, se procederá a pulsar el botón de validar, este permitirá el desbloqueo de los botones que se encuentran en la pestaña de control manual y de los que se encuentran en la pestaña de control programable, tal y como se muestran en las Figura 5.12 y Figura 5.13 respectivamente.

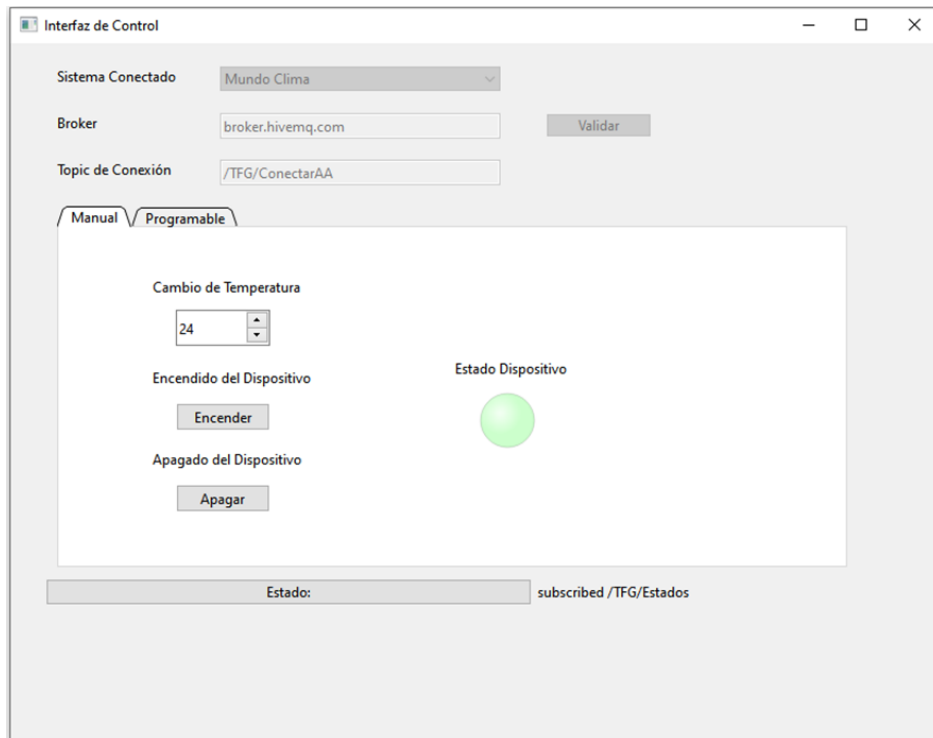


Figura 5.12. Pestaña de control manual.

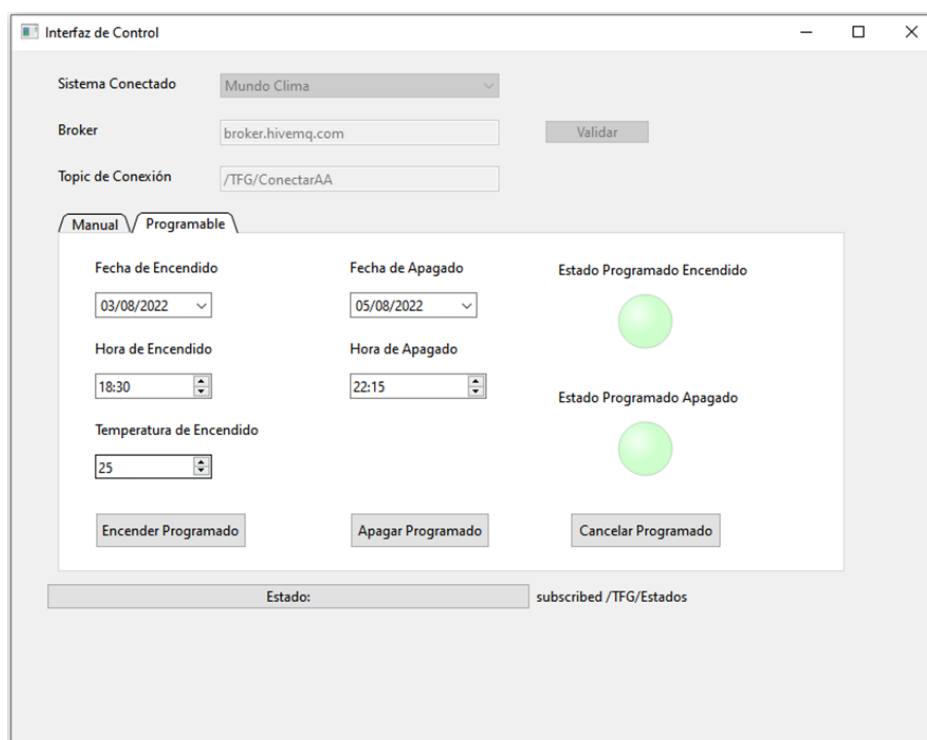


Figura 5.13. Pestaña de control programable.

A continuación, comenzaremos con la configuración y funcionamiento del control manual del sistema que permite el cambio de temperatura, encendido y apagado instantáneo del dispositivo, esto se puede realizar mediante los botones que se encuentran en la ventana. El cambio de la temperatura se podrá realizar de dos maneras distintas, una de ellas será pulsando directamente sobre el número que se encuentre e indicando cual es la nueva temperatura, esta deberá oscilar entre los valores de los 18°C y los 30°C. Una segunda opción sería pulsando sobre las flechas que aparecen en el lateral derecho del recuadro que contiene el valor de la temperatura, de manera que con estas flechas podrá indicar si se desea aumentar o decrementar la temperatura del aire.

Más adelante, se pueden observar dos botones, con los cuales se realizará el encendido y apagado del dispositivo, según el botón que se pulse. Además, si se realiza el encendido del *split* se encenderá el led de color verde que se encuentra a la derecha de la pestaña, como se puede observar en la Figura 5.14, si en el caso contrario se pulsara el botón de apagado el led se apagaría y volvería al estado de la Figura 5.15.

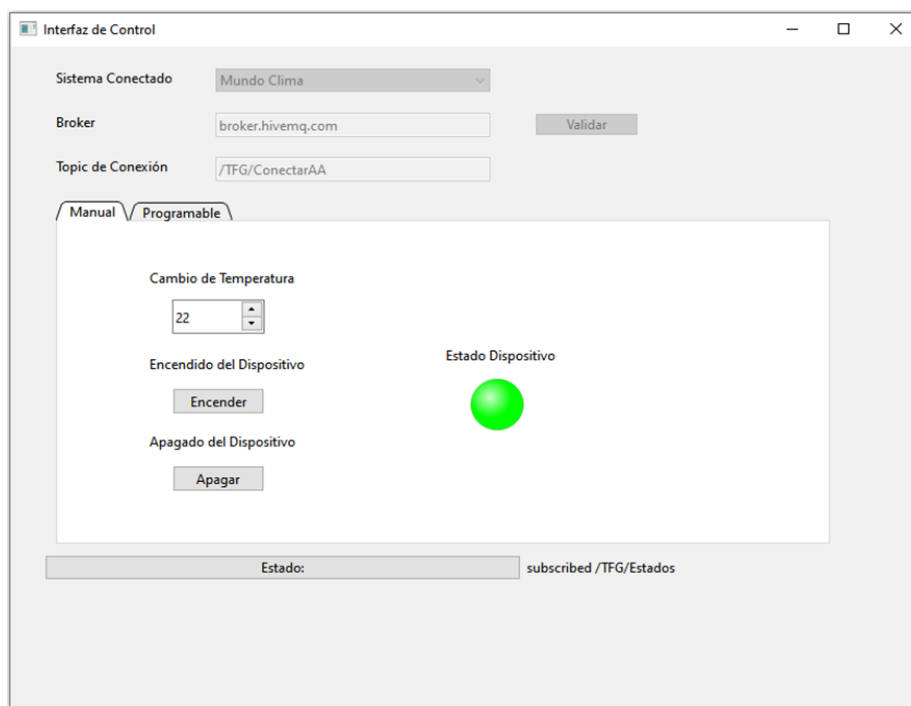


Figura 5.14. Activado el encendido manual.

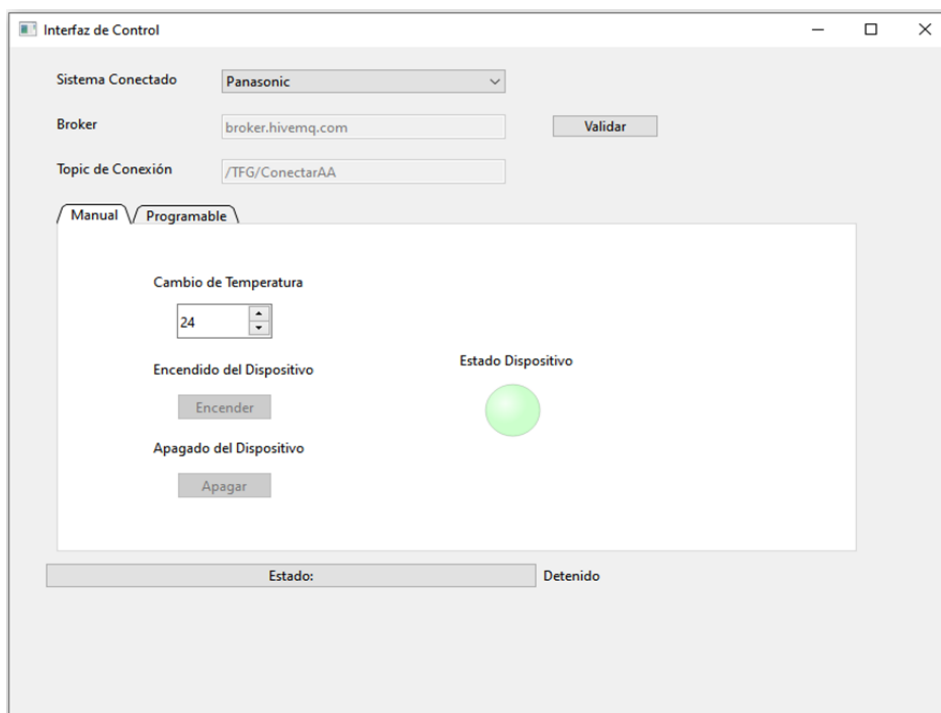


Figura 5.15. Activado el apagado manual.

Por último, se encuentra el control programable del sistema, al cual se puede acceder pulsando sobre la pestaña de programable que aparece en la ventana del ejecutable, sobre la que se vería una interfaz similar a la que se observa en la Figura 5.16.

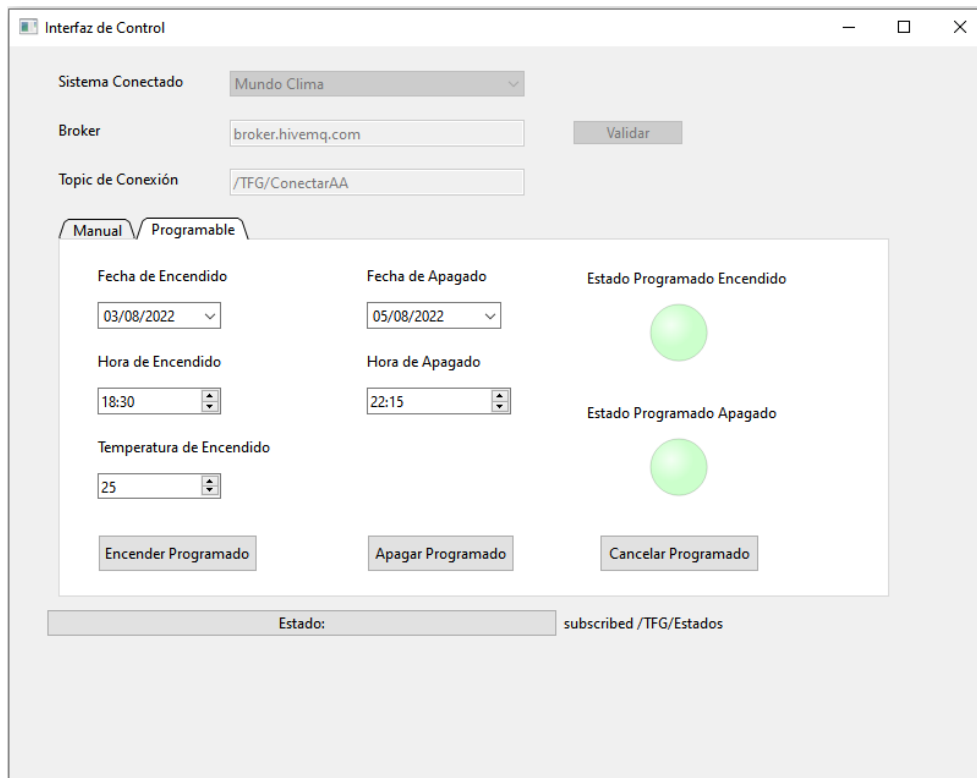


Figura 5.16. Interfaz de usuario para el control programable.

Dentro de este parte, se puede observar la configuración del encendido programado, es decir, el usuario establece el momento de encendido del *split*, y del apagado programable, el cual se encarga de apagar el dispositivo en el momento indicado, esto sucederá siempre y cuando este se encuentre en el estado inverso al seleccionado. Tanto la parte de control de encendido como de apagado son muy similares, ya que ambos cuentan con un desplegable sobre el que seleccionar la fecha, o introducirla mediante escritura manual sin necesidad de buscarla en el desplegable. Además de la fecha, aparece un recuadro sobre el que se modificará la hora, esto se podrá realizar de dos maneras distintas, mediante la introducción directa por teclado o tras pulsar en las flechas que aparecen a la derecha del recuadro, estas harán aumentar la parte de las horas o de los minutos dependiendo donde se haya pulsado con anterioridad. Se concluirá el encendido programado tras realizar la configuración de la

temperatura a la que se encenderá el dispositivo, esto se realizará de la misma manera que en el caso manual, mediante la introducción manual o mediante las flechas ubicadas a la derecha del recuadro. Una vez seleccionado todos los parámetros, se deberá pulsar sobre el botón de “Encender Programado”, el cual provocará que se encienda el led superior ubicado a la derecha de la ventana, de la misma manera que se muestra en la Figura 5.17.

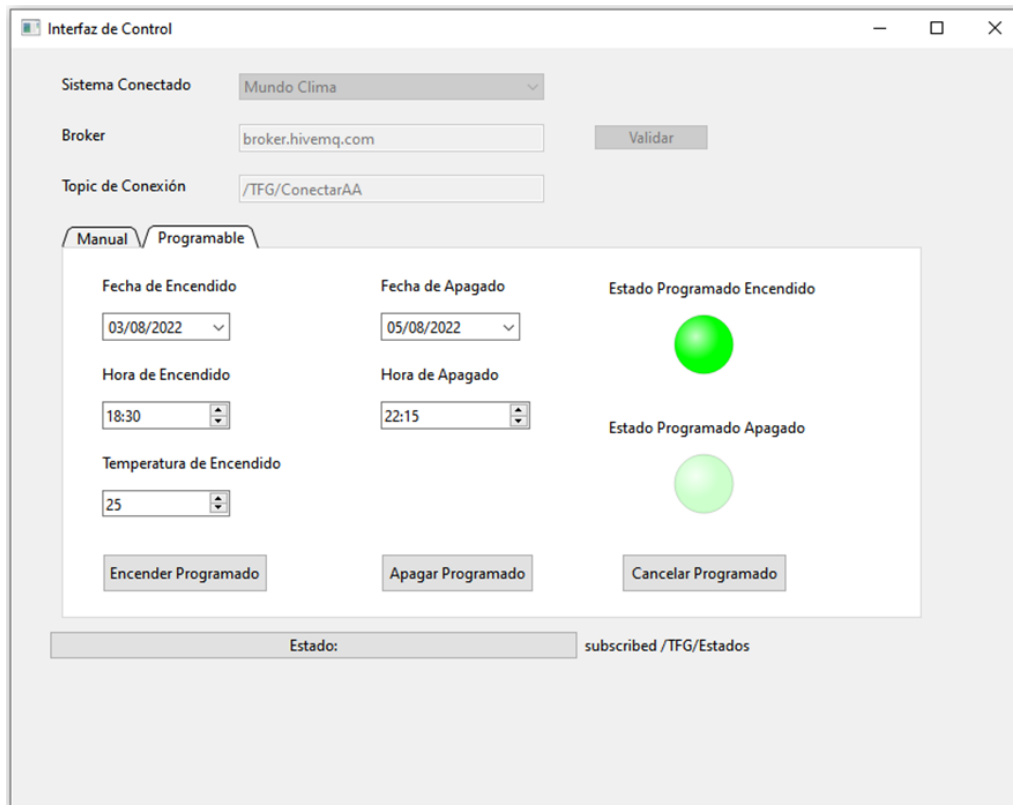


Figura 5.17. Activado encendido programable.

Para poder activar el apagado programable se deben configurar los parámetros de igual manera que se realiza para el encendido programable, con la única diferencia es que no será necesario seleccionar la temperatura. Una vez que se ha realizado la selección de las variables, se debe pulsar el botón inferior de “Apagar Programable”, el cual encenderá el led inferior de la parte derecha, tal y como se puede observar en la Figura 5.18.

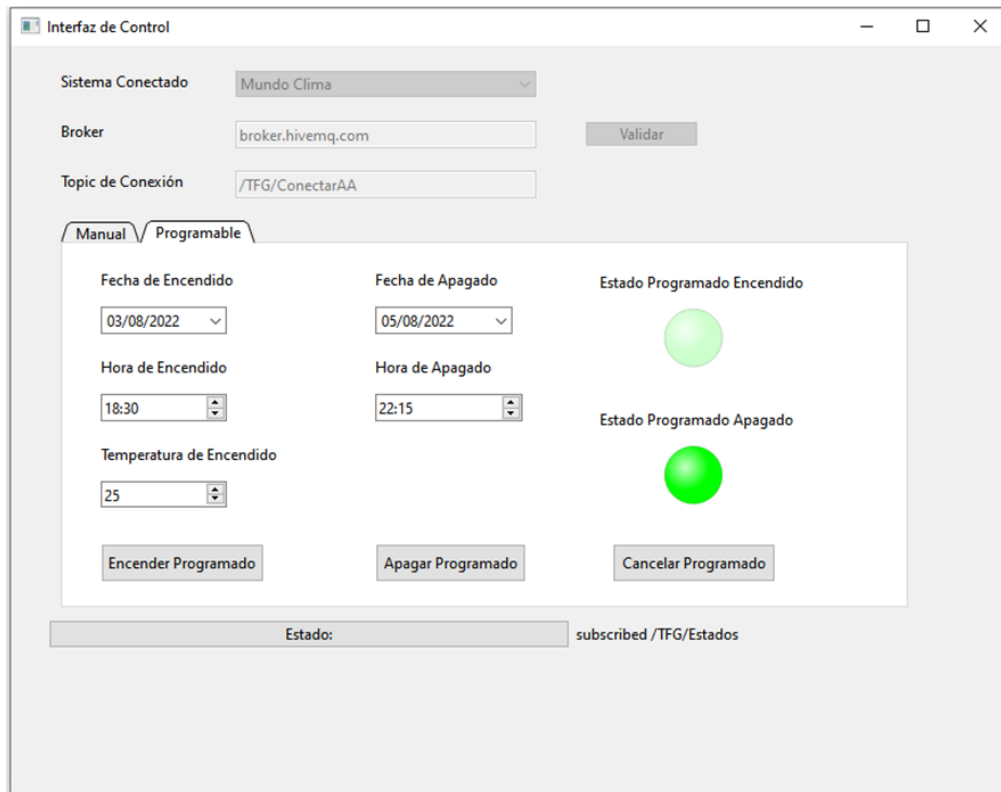


Figura 5.18. Activado el apagado programable.

Otro caso de funcionamiento sería activando simultáneamente el encendido y apagado programado, pudiendo tener ambos la misma fecha, pero una hora de activación distinta entre ellos, si no el programa no se ejecutaría correctamente. Esta doble activación se visualizará con el encendido de los dos leds situados a la derecha de la pantalla como se observa en la Figura 5.19.

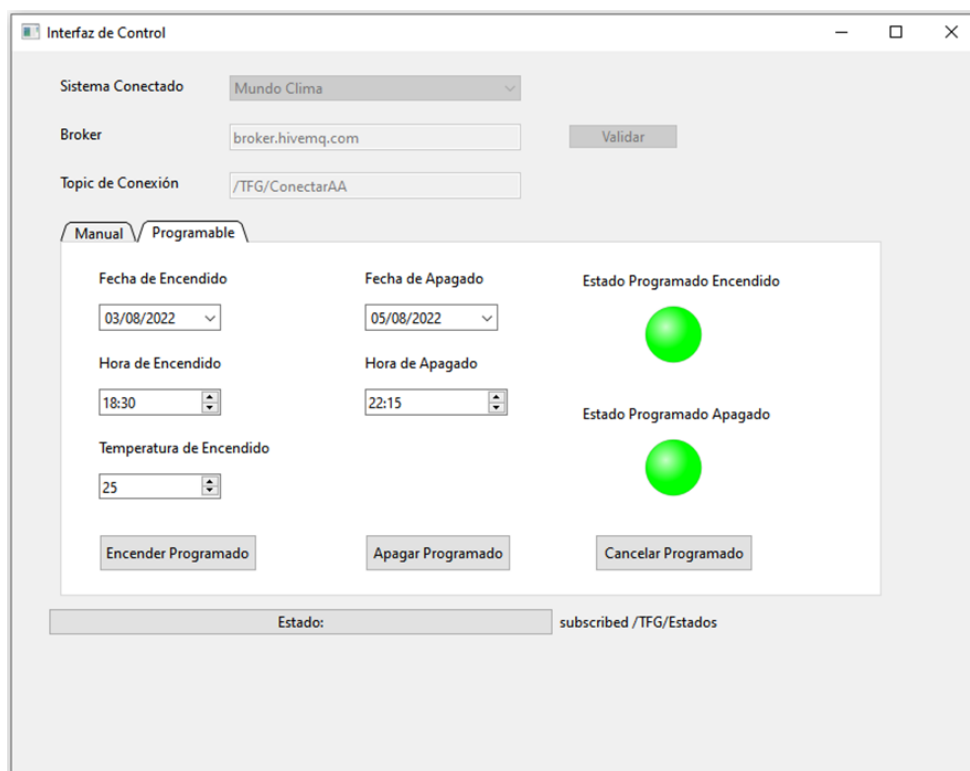


Figura 5.19. Activación conjunta de encendido y apagado programado.

El último botón que se podrá utilizar es el de “Cancelar Programado”, el cual se encarga de realizar la anulación de los parámetros que se han configurado tanto en el encendido como en el apagado programable, además de apagar los dos leds, de manera que se confirma la desactivación de las ejecuciones enviadas con anterioridad. Tras pulsar este botón volveríamos a la configuración inicial como la que se muestra en la Figura 5.16.

Capítulo 6. Conclusiones y trabajo futuro

6.1. Conclusiones

El prototipo final que se ha conseguido desarrollar ha permitido cumplir con todos los objetivos propuestos en la memoria, el cual, pretendía que un usuario pudiera controlar un sistema de aire acondicionado sin la necesidad de que este se encuentre en la vivienda donde esté ubicado el dispositivo.

Además, este proyecto puede servir como base para futuros trabajos, en los cuales se consiga que el prototipo llegue a ser un dispositivo completamente funcional y con ello se permita contribuir a la sociedad ayudando a los usuarios a no tener un gasto excesivo de la energía eléctrica a causa de un uso inadecuado del aire acondicionado, consiguiendo un mayor control sobre este y haciendo que su funcionamiento sea más efectivo. Junto a esto, se encuentra que no será necesario realizar un cambio en el sistema de aire acondicionado por falta de tecnología en él, ya que esto se podrá subsanar con nuestro dispositivo y su aplicación, de esta manera también se evita la generación de residuos tecnológicos innecesarios, siempre y cuando el sistema no se encuentre con algún problema el cual no se pueda resolver o se haya impuesto por normativa la ilegalidad de alguno de sus componentes a causa de la toxicidad que puedan producir.

Para finalizar, este proyecto ha pretendido ser el culmen de una etapa de estudios durante la que se ha llegado a obtener un alto nivel de conocimientos que han sido aplicados a lo largo de todo el trabajo. Se ha realizado una amplia aplicación de las competencias transversales que se han estado explicando durante los últimos años. Se ha pretendido aplicar al máximo las habilidades

adquiridas, entre ellas las de programación de distintos modelos de microcontroladores, mediante el uso de las plataformas más idóneas para el aprovechamiento de estos. También se han aplicado los conocimientos de diseño de PCBs, teniendo en cuenta todos los factores que se deben aplicar en estos casos, junto con el montaje de componentes y soldadura de estos en las placas.

6.2. Líneas futuras

Las líneas futuras posibles para este proyecto podrían ser muy extensas, ya que no se ha llegado a maximizar el rendimiento que poseen los elementos que componen el sistema completo. A esto habría que añadirle que las tecnologías que poseemos en nuestras viviendas siguen avanzando muy rápido y van apareciendo nuevos modos de control de los distintos dispositivos, entre ellos nos encontramos con el sistema de reconocimiento de voz, el cual, permite el control de los dispositivos mediante el habla de una persona.

También sería bueno tener en consideración que no se han llegado a implementar la funcionalidad completa de un mando de control remoto de un sistema de aire acondicionado, lo cual habría que realizar para poder seleccionar su modo de funcionamiento o la velocidad del ventilador con la que expulsa el aire.

Además, habría que tener en cuenta que solo se ha desarrollado la aplicación para ser instalada únicamente para ordenadores, ya sean portátiles o de sobremesa, compuestos por un sistema operativo de Windows, por lo que esto se podría extender para ser usado en otros dispositivos electrónicos, tales como un smartphone o una tablet.

Finalmente, y en el caso de que el sistema fuera empleado para el uso final que se desea tener de él y no ser solo un prototipo de pruebas, habría que mejorar la forma de alimentar los dispositivos, consiguiendo que se encontrará conectado a la red eléctrica, también, habría que añadirle un encapsulado robusto para almacenarlos dentro de la estancia del usuario.

Apéndice A. Presupuesto de elaboración

En este apartado procederemos a realizar un desglose del coste final del prototipo que se ha desarrollado a lo largo de toda la memoria. Para ello, comenzaremos dando el presupuesto utilizado en los componentes que se han incorporado, como se puede observar en la Tabla A.1.

Componente	Coste unitario (€)	Unidades empleadas	Coste total (€)
Microcontrolador ESP32	9.99€	3	29.97€
Sensor DHT22	8.99€	1	8.99€
Sensor PIR HC-SR501	6.99€	1	6.99€
Sensor de proximidad	2.16€	1	2.16€
Placas pretaladradas	7.99€	1	7.99€
Flux	5.95€	1	5.95€
Hilo de estaño	9.99€	1	9.99€
Total			72.04€

Tabla A.1. Coste de componentes y material fungible.

En la Tabla A.2 se realizará un listado de aquellas aplicaciones software que se han utilizado y los costes que estas tendrían si se decidiera utilizarlas de manera profesional junto con su periodo de uso.

Equipo/software	Coste unitario (€)	Periodo de amortización	Periodo de uso	Coste total (€)
DesignSpark	1.218,07€	12 meses	6 meses	604€
Eclipse	0€	0 meses	6 meses	0€
QT	3950€	12 meses	6 meses	1975€
MQTT Server HiveMQ	300€	1 mes	6 meses	1800€
Inno Setup Compiler	0€	0 meses	1 mes	0€
Total				4379€

Tabla A.2. Amortización de equipos y licencias de software.

Por último, se mostrarán los costes finales que tendría la realización de un proyecto juntando todos los componentes hardware y software que serían necesarios para su desarrollo, esto se detallará en la Tabla A.3.

Concepto	Coste (€)
Coste de componentes y material fungible	72.04€
Amortización de equipos y licencias de software	4379€
Mano de obra	200€
Coste total	4651.04€

Tabla A.3. Presupuesto total.

Apéndice B. Esquemáticos de las PCBs

Se ha decidido incorporar tanto los ficheros que contienen los códigos de funcionamiento cómo los diseños de los esquemáticos de las distintas placas PCB en un directorio de *Github*, el cual se encuentra en la dirección: https://github.com/Lebron1897/Air_Conditioner_TFG.

En este apéndice se mostrarán los esquemas de las diferentes placas PCB que se han diseñado para este proyecto, con el formato mediante el que se enviarían para realizar la impresión de cada una de ellas.

Se comenzará con el esquema de la placa diseñada para el módulo principal, el cual, contiene el interfaz IR. Este se muestra en la Figura B.1.

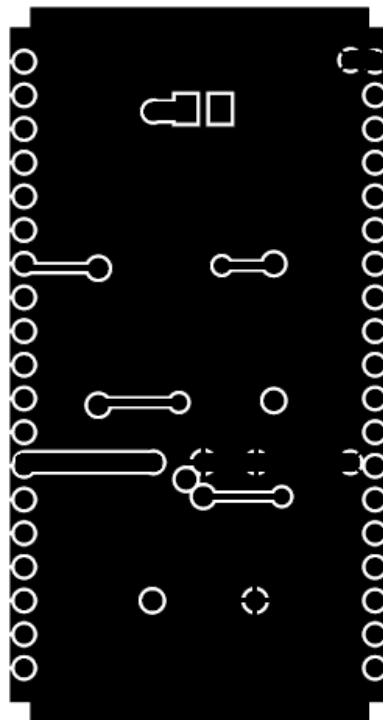


Figura B.1. Esquema del módulo principal.

El siguiente esquema, pertenecerá al módulo que contiene el sensor de temperatura y humedad y el sensor de movimiento. En la Figura X se observa la cara *top* del diseño, mientras que en la Figura Y se encuentra la cara *bottom*.

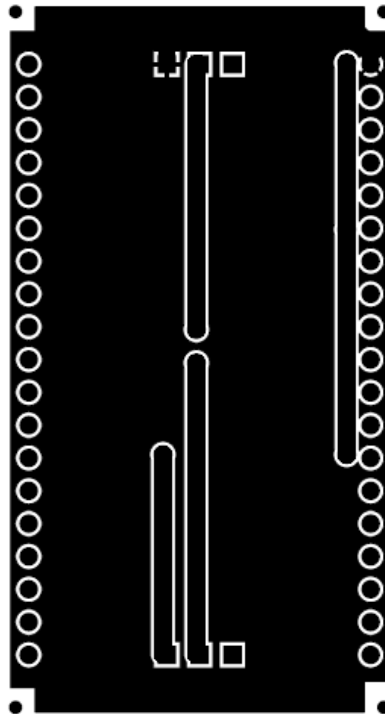


Figura B.2. Esquema del módulo con el sensor de temperatura y de movimiento por la cara *top*.

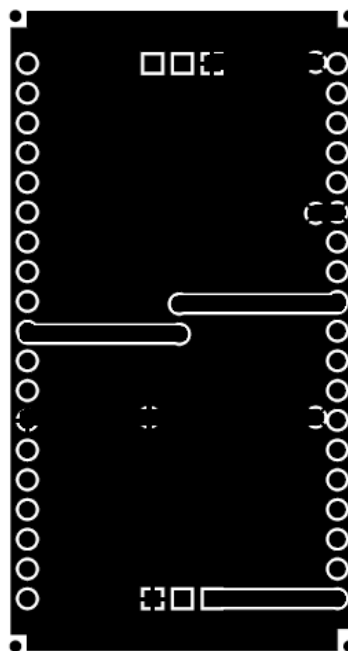


Figura B.3. Esquema del módulo con el sensor de temperatura y de movimiento por la cara *bottom*.

Por último, se observa en la Figura X y la Figura Y el esquema del diseño del módulo compuesto por el sensor de proximidad por las caras *top* s *top* y *bottom* respectivamente.

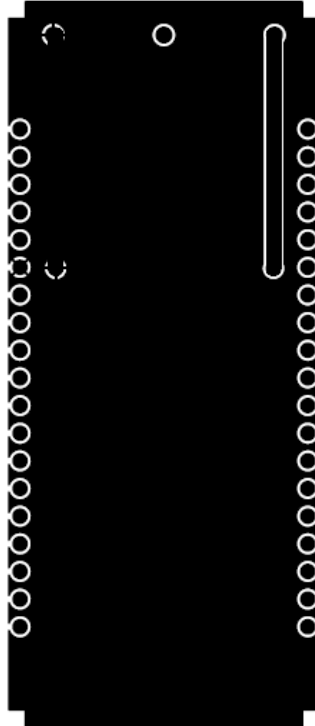


Figura B.4. Esquema del módulo con el sensor de proximidad por la cara *top*.

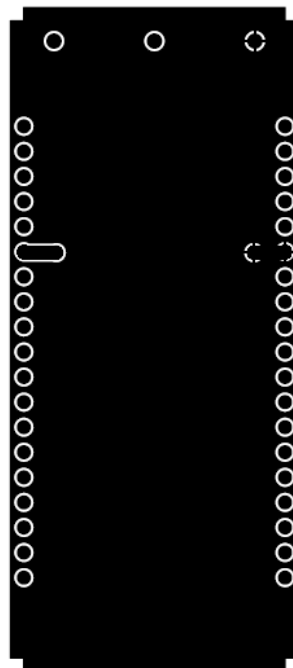


Figura B.5. Esquema del módulo con el sensor de proximidad por la cara *bottom*.

Referencias

- [1] C. González Báez, «Historia del aire acondicionado,» ElAireAcondicionado.com, 2021. [En línea]. Available: <https://www.elaireacondicionado.com/articulos/historia-del-aire-acondicionado>. [Último acceso: 29 Diciembre 2021].
- [2] I. Arnabat, «Los gases refrigerantes sustitutos - Infografía: evolución y futuro,» Caloryfrio.com, 8 Enero 2021. [En línea]. Available: <https://www.caloryfrio.com/refrigeracion-frio/evolucion-futuro-refrigeracion-gases-refrigerantes.html>. [Último acceso: 29 Diciembre 2021].
- [3] E. C. Zapata, «Aires acondicionados tipo Split: Información adicional,» Frigolutions.com, 7 Junio 2020. [En línea]. Available: <https://www.frigolutions.com/2020/06/aires-acondicionados-tipo-split.html>. [Último acceso: 29 Diciembre 2021].
- [4] Á. Ortega, «Aire acondicionado Split, Como funciona, Sus partes, Ciclo de refrigeración del aire acondicionado, Tipos, Diferencias entre aire acondicionado de ventana y SPLIT,» Ingenieriapotencial.blogspot.com, 1 Febrero 2019. [En línea]. Available: <http://ingenieriapotencial.blogspot.com/2019/02/aire-acondicionado-split-como-functiona.html>. [Último acceso: 29 Diciembre 2021].
- [5] S. De la Cruz, «El precio de la luz pulveriza todos los récords y alcanzará el martes su máximo histórico con 545 euros/MWh,» La Razón, 7 Marzo 2022. [En línea]. Available: <https://www.larazon.es/economia/20220307/ghf3igkl2bc5dkuuwa7fk3z6ve.html>. [Último acceso: 15 Junio 2022].
- [6] CNMC, «La nueva factura de la luz,» CNMC.es, 2021. [En línea]. Available: <https://www.cnmc.es/la-nueva-factura-de-la-luz>. [Último acceso: 29 Diciembre 2021].
- [7] J. García, «En 2019 generamos 53,6 millones de toneladas de residuos electrónicos en todo el mundo, según Naciones Unidas,» Xataka.com, 3 Julio 2020. [En línea]. Available: <https://www.xataka.com/otros->

- dispositivos/2019-generamos-53-6-millones-toneladas-residuos-electronicos-todo-mundo-naciones-unidas. [Último acceso: 29 Diciembre 2021].
- [8] Toshiba, «Adaptador Wifi,» Toshiba.com, 2021. [En línea]. Available: <https://www.toshiba-aire.es/adaptador-wifi>. [Último acceso: 29 Diciembre 2021].
- [9] Daikin, «Online Controller de Daikin,» Daikin.com, 2021. [En línea]. Available: https://www.daikin.es/es_es/area-de-profesionales/climatizacion-para-tu-negocio/sistemas-de-control/daikin-online-controller.html. [Último acceso: 29 Diciembre 2021].
- [10] Mitsubishi, «Software y Apps,» Mitsubishi.es, 2021. [En línea]. Available: <https://www.mitsubishielectric.es/aire-acondicionado/series/software-y-apps-2/>. [Último acceso: 29 Diciembre 2021].
- [11] E. Rodríguez De Luis, «Ocho kits para convertir un aire acondicionado en inteligente: guía de compra y recomendaciones,» Xataka.com, 14 Junio 2021. [En línea]. Available: <https://www.xataka.com/seleccion/6-kits-para-convertir-aire-acondicionado-inteligente-1>. [Último acceso: 29 Diciembre 2021].
- [12] Espressif Systems, «Espressif,» Marzo 2022. [En línea]. Available: https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Último acceso: 5 Julio 2022].
- [13] T. Liu, «Adafruit Shop,» Marzo 2016. [En línea]. Available: <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>. [Último acceso: 5 Julio 2022].
- [14] Punto Flotante SA, «Sensor infrarrojo de movimiento PIR,» Enero 2017. [En línea]. Available: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>. [Último acceso: 15 Julio 2022].
- [15] A. Pini, «DigiKey,» 21 Abril 2021. [En línea]. Available: <https://www.digikey.es/es/articles/the-fundamentals-of-proximity-sensors-selection-and-use-industrial-automation>. [Último acceso: 5 Julio 2022].
- [16] S. Santos, «Random Nerd Tutorials,» 21 Agosto 2020. [En línea]. Available: <https://randomnerdtutorials.com/esp32-esp-now-wi-fi-web-server/>. [Último acceso: 15 Julio 2022].
- [17] ABC Economía, «El precio de la luz este miercoles: las horas más baratas y las que tienen que evitar para ahorrar,» abc.es, 25 Noviembre 2021. [En línea]. Available: https://www.abc.es/economia/abci-precio-manana-espanoles-obligados-madrugar-para-ahorrar-factura-202111232039_noticia.html. [Último acceso: 29 Diciembre 2021].