



Universidad
Zaragoza

Trabajo Fin de Grado

Automatización de una vivienda unifamiliar

Automatization of a single-family house

Autor

Alba Continente Sisamón

Director

Dr. Diego Antolín Cañada

Escuela Universitaria Politécnica La Almunia
2017



**Escuela Universitaria
Politécnica - La Almunia**
Centro adscrito
Universidad Zaragoza

**ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

MEMORIA

Automatización de una vivienda
unifamiliar

Automatization of a single-family house

424.19.51

Autor: Alba Continente Sisamón

Director: Dr. Diego Antolín Cañada

Fecha: 5 de Febrero de 2020

INDICE DE CONTENIDO

1. RESUMEN	1
1.1. PALABRAS CLAVE	1
2. ABSTRACT	3
2.1. KEY WORDS	3
3. INTRODUCCIÓN	5
4. ANTECEDENTES	7
4.1. DOMÓTICA CABLEADA (KNX)	10
4.2. RADIODIFUSIÓN (ZIGBEE)	11
4.3. IoT	11
4.3.1. <i>Altavoces Inteligentes</i>	12
4.3.1.1. Google Home	12
4.3.1.2. Alexa	12
4.3.2. <i>Otros dispositivos IoT comerciales</i>	13
4.3.3. <i>IFTTT</i>	14
5. MARCO TEÓRICO	15
5.1. DEFINICIÓN DEL SISTEMA.	15
5.2. HARDWARE.	17
5.2.1. <i>Sensores.</i>	17
5.2.1.1. Sensor de Temperatura.	18
5.2.1.2. Sensores Fotoeléctricos.	19
5.2.1.3. Sensores de Presencia.	19
5.2.2. <i>Alimentación del circuito.</i>	21
5.2.3. <i>Dispositivos IoT.</i>	21
5.2.3.1. Microcontroladores IoT.	22
5.2.3.2. Placas de desarrollo.	24
5.2.3.2.1. Raspberry Pi.	24
5.2.3.2.2. Arduino.	24
5.2.3.2.3. NodeMCU.	25
5.2.3.2.4. Placas basadas en STM32.	25
5.2.4. <i>Actuadores</i>	26
5.2.4.1. Motores para persianas.	26

INDICES

5.2.4.2. Relés	27
5.3. SOFTWARE	28
5.3.1. <i>Protocolos de comunicación</i>	28
5.3.2. <i>Broker</i>	30
5.3.3. <i>Plataformas IoT</i>	30
5.3.3.1. Node-RED	31
5.3.4. <i>Lenguajes de programación</i>	31
5.3.4.1. Lenguaje C y C++	31
5.3.4.2. Lenguaje Arduino	32
5.3.4.3. Lenguaje JavaScript	32
6. DESARROLLO	35
6.1. ESTUDIO DE LA VIVIENDA A AUTOMATIZAR	35
6.2. ELECCIÓN DE COMPONENTES Y SOFTWARE	36
6.2.1. <i>Sensores</i>	37
6.2.1.1. Sensor de temperatura	37
6.2.1.2. Sensor de intensidad lumínica.	39
6.2.1.3. Sensor de presencia.	41
6.2.2. <i>Actuadores</i>	43
6.2.2.1. Motores de persianas	43
6.2.2.2. Relés	44
6.2.3. <i>Placa microcontroladora</i>	44
6.2.4. <i>Plataforma IoT</i>	46
6.3. PRUEBA INICIAL DE COMUNICACIÓN	48
6.3.1. <i>Configuración del Broker</i>	48
6.3.2. <i>Preparación de la plataforma</i>	49
6.3.3. <i>Programación en Arduino</i>	51
6.3.4. <i>Programación en LabVIEW.</i>	51
6.3.5. <i>Programación en Node-RED</i>	53
6.4. IMPLEMENTACIÓN DEL SISTEMA FINAL	55
6.4.1. <i>Conexión de la electrónica</i>	55
6.4.2. <i>Programación Arduino IDE</i>	56
6.4.3. <i>Programación Node-RED</i>	60
6.4.3.1. Dashboard	63
6.4.4. <i>Dashboard Android</i>	65
7. RESULTADOS	67
8. CONCLUSIONES	76
8.1. TRABAJOS FUTUROS	77

INDICE DE ILUSTRACIONES

Ilustración 1 UNIVAC I	7
Ilustración 2 ECHO IV.....	8
Ilustración 3 Módulos X10.....	9
Ilustración 4 Controladores X10.....	9
Ilustración 5 Dispositivos Google Home	12
Ilustración 6 Dispositivos Amazon Echo	13
Ilustración 7 Diagrama funcional IFTTT.....	14
Ilustración 8 Esquema global del sistema.....	16
Ilustración 9 Termistor NTC	18
Ilustración 10 Sensor Termopar.....	18
Ilustración 11 Sensor RTD	18
Ilustración 12 Tipos de sensores fotoeléctricos infrarrojos	20
Ilustración 13 Sensor de presencia por ultrasonidos	20
Ilustración 14 Funcionamiento del sensor PIR.....	20
Ilustración 15 Cargador universal con salida de bornes.....	21
Ilustración 16 Esquema básico de un microcontrolador.....	23
Ilustración 17 Diagrama de funcionamiento Pub/Sub.....	29
Ilustración 18 Información del broker CloudMQTT.	48
Ilustración 19 Visualización de los mensajes publicados en el broker CloudMQTT.	49
Ilustración 20 Inicialización de Node-RED de forma local en Windows.....	49
Ilustración 21 a) Menú de nodos. b) Información del flujo. c) Debugger del flujo. d) Configuración del dashboard global.....	50

INDICES

Ilustración 22 Visualización en LabVIEW del programa.	52
Ilustración 23 Flujo completo de la prueba en Node-RED.	53
Ilustración 24 Interfaz Node-RED con calefacción apagada (Izquierda) y encendida (derecha)	54
Ilustración 25 Montaje (Izquierda) y esquema eléctrico (Derecha) para el Node1.	55
Ilustración 26 Montaje (Izquierda) y esquema eléctrico (Derecha) para el Node2.	56
Ilustración 27 Diagrama de funcionamiento para el control de las persianas.....	57
Ilustración 28 Diagrama de funcionamiento del control de presencia.	58
Ilustración 29 Diagrama de funcionamiento del control de temperatura.	58
Ilustración 30 Curva y parámetros de calibración para el sensor MF-52.	59
Ilustración 31 Flujo principal de Node-RED.....	60
Ilustración 32 Subflujo "Temperatura".....	61
Ilustración 33 Subflujo "Luminosidad".	61
Ilustración 34 Subflujo "Entrada".	62
Ilustración 35 Interfaz web Node-RED.....	64
Ilustración 36 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función persianas.....	65
Ilustración 37 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función presencia.....	66
Ilustración 38 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función temperatura.....	66
Ilustración 39 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (calefacción apagada).	67
Ilustración 40 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (calefacción encendida).	68
Ilustración 41 Prototipo en el caso 2 (Calefacción encendida).	68
Ilustración 42 Interfaz web (Izquierda) y móvil (Derecha) para el caso 4 (manual OFF).	69

Ilustración 43 Interfaz web (Izquierda) y móvil (Derecha) para el caso 3 (manual ON).	69
Ilustración 44 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (automático, poca luz).	70
Ilustración 45 Prototipo en el caso 1 (automático, poca luz).	70
Ilustración 46 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (automático, mucha luz).	71
Ilustración 47 Prototipo en el caso 2 (automático, mucha luz).	71
Ilustración 48 Interfaz web (Izquierda) y móvil (Derecha) para el caso 4 (manual OFF).	72
Ilustración 49 Interfaz web (Izquierda) y móvil (Derecha) para el caso 3 (manual ON).	72
Ilustración 50 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (automático, poca luz).	73
Ilustración 51 Prototipo en el caso 1 (automático, sin presencia).	73
Ilustración 52 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (automático, presencia).	74
Ilustración 53 Prototipo en el caso 2 (automático, presencia).	74

INDICE DE TABLAS

Tabla 1 Tabla resumen de problemas y soluciones.	36
Tabla 2 Tabla comparativa de sensores de temperatura.....	38
Tabla 3 Tabla comparativa de sensores de intensidad lumínica.	40
Tabla 4 Tabla comparativa de sensores de presencia.....	42
Tabla 5 Tabla comparativa de placas microcontroladoras	45
Tabla 6 Tabla comparativa de plataformas IoT.	46

1. RESUMEN

Este proyecto se centra en el campo de la domótica, que busca el llevar las tecnologías automáticas al ámbito del hogar. Para ello, se adentra en las tecnologías IoT, utilizando comunicación WiFi y el protocolo MQTT.

El objetivo es el desarrollo de una aplicación que permita controlar las luces, persianas y climatización de una vivienda unifamiliar, por medio del uso sensores de presencia, de temperatura y de intensidad lumínica.

En primera instancia se plantea una prueba de laboratorio previa con el fin de conocer las funcionalidades de la plataforma y de la comunicación MQTT, la cual se realiza con un Arduino UNO y una programación en LabVIEW que permita comunicarse con la red.

Posteriormente se realiza la selección de componentes del sistema final por medio de tablas comparativas y se realiza el montaje de la electrónica.

La lectura de los sensores y la activación de los actuadores se diseña dentro de un entorno de programación Arduino, siendo los controladores dos módulos NodeMCU que se basan en el integrado ESP8266.

Toda la información recogida por los microcontroladores se publica en el broker MQTT, un servidor que aloja los datos bajo un determinado topic o tema, permitiendo a todos los dispositivos conectados acceder a estos.

El sistema global se gobierna a través de la plataforma Node-RED, la cual cuenta con una interfaz web que permite al usuario interactuar de manera remota con la aplicación. Por último, se presenta una interfaz adicional en un Smartphone Android con las mismas funciones.

El resultado es un prototipo completamente funcional en cuanto a envío y recepción de datos, que además cuenta con dos interfaces, una web y otra en una aplicación de móvil, lo cual hace que el sistema sea muy sencillo e intuitivo para el usuario.

1.1. PALABRAS CLAVE

Domótica, MQTT, WiFi, Node-RED, ESP8266



2. ABSTRACT

This project is focused on Domotics, seeking home automation through the Internet of Things (IoT), applying wireless networking technologies (WiFi) and the MQTT protocol.

The main objective is the development of an application for a smart house, allowing to control lights, shutters and air conditioning using presence, temperature and light intensity sensors.

Firstly, this project sets out a previous laboratory test in order to know about the functions of IoT platform and MQTT communication, which is carried out with an Arduino UNO and a LabVIEW program that enables network communication.

Subsequently, the components for the final system are selected through comparative tables and the electronic prototype is set up.

Both sensor reading and activation of actuators are designed within an Arduino programming environment, with two NodeMCU modules as controllers, which are based on the integrated circuit ESP8266.

All microcontrollers' collected data is shown in the MQTT broker, a server that hosts data under a specific topic, allowing access to all connected devices.

The global system is controlled via Node-RED platform that provides an integrated web interface, enabling user remote interaction with the software. Lastly, an additional Android interface is displayed on a Smartphone with the same functionalities.

To sum up, the result is a completely functional model regarding data transmission and reception with two interfaces, web-based and Android-based, which leads to an intuitive and user-friendly system.

2.1. KEY WORDS

Domotics, MQTT, WiFi, Node-RED, ESP8266



3. INTRODUCCIÓN

Durante los últimos años, hemos observado cambios muy rápidos en nuestro entorno. Nuestra generación, la generación Z, ha vivido diferentes etapas tecnológicas a raíz de la revolución informática que se inicia a mediados del siglo XX.

Este proceso ha afectado tanto a los teléfonos, popularizándose los móviles a principios de milenio y los Smartphone en la década del 2010; como a los ordenadores, que en los últimos 25 años han experimentado grandes cambios tanto visuales como funcionales; o a cualquier otro elemento cotidiano, incluyendo nuestros hogares.

Hoy en día no es extraño encontrar cada vez más dispositivos “inteligentes”, que permitan conectarse entre sí e incluso ser autónomos. La unión de todo este desarrollo se observa en la domótica, algo que se afianza cada día más en la vida de las personas.

La domótica es, además, una de las ideas más deseadas y extendidas, ya imaginada en la ciencia ficción desde mucho antes de concebir la posibilidad de que existiera realmente en el futuro. Esta consciencia de que existe dicha posibilidad resulta cada vez más atractiva y, a pesar de la sencillez que existe en los sistemas actuales como Alexa o Google Home y la profesionalidad que se puede ver en los sistemas cableados, se percibe una oportunidad en desarrollar un sistema mixto que permita la versatilidad y simpleza de lo inalámbrico sin dejar de lado la compatibilidad con elementos tradicionales sin conectividad.

Sumando a todo lo anterior los problemas de aislamiento y control de temperatura, además del control de algunas luces, que han existido siempre en mi casa, este proyecto se presenta también como la búsqueda de una solución óptima mediante la automatización a todos ellos.

El presente proyecto parte del estudio de sistemas domóticos ya existentes, a raíz de los cuales se plantea un sistema general. En este sistema mixto (WiFi y cableado), se definen los diferentes elementos, detallando separadamente el hardware y software de este. A continuación, se plantean los problemas a resolver, así como las acciones que tomar para arreglarlos. Posteriormente se presentan los diferentes componentes seleccionados para el prototipo. Con el fin de dar respuesta al objetivo del proyecto, se realiza una primera propuesta que permita conocer en detalle la comunicación MQTT y el funcionamiento de la plataforma Node-RED. Todo ello culmina en la implementación de un sistema constituido por los elementos seleccionados, la comunicación MQTT y la plataforma, dando como resultado un prototipo completo y dos interfaces de usuario, móvil y web.



4. ANTECEDENTES

Con el desarrollo de las diferentes tecnologías, siempre se han buscado múltiples campos de aplicación para éstas, tanto industrialmente como en la vida cotidiana. Por ello surge la inquietud de cómo sería el lugar donde vivimos si fuera "inteligente", si todo lo que hacemos los humanos lo hiciera de forma autónoma. De ahí viene la palabra domótica, proveniente del latín "domus" (casa) y del griego automático, "que se gobierna a sí mismo".

Esta inquietud se hizo mayor cuando a principios del siglo XX con la Revolución Industrial aparecieron los primeros electrodomésticos. Aunque no eran aparatos inteligentes, sí que cumplían con la primera parte de lo que provocaba dicha inquietud, empezaban a hacer cosas que los humanos hacíamos hasta ese momento, la aspiradora, la lavadora, la secadora o el lavavajillas son ejemplos de ello.

Durante los años 50 y 60 empiezan a aparecer los primeros ordenadores. Una de las primeras computadoras comerciales se llamó UNIVAC I (UNIVERSAL Automatic Computer I). No sería hasta los 60 que aparecerían los primeros monitores de computadora, ya que, hasta ese momento, como vemos en la Ilustración 1, se trataba de una computadora formada por 5000 tubos de vacío que era capaz de procesar unos 1000 cálculos por segundo, teniendo como interfaz de usuario una estación de control, que se puede apreciar en la ilustración.

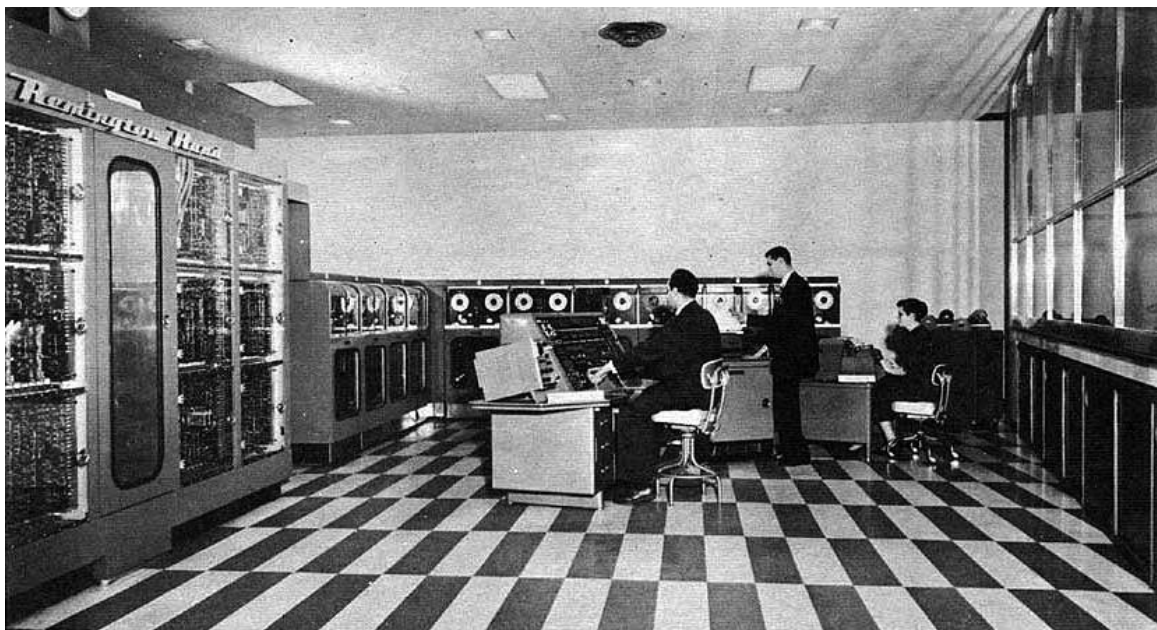


Ilustración 1 UNIVAC I

El considerado como primer producto domótico fue el ECHO IV (Electronic Computing Home Operator), que surgió en 1966, aunque nunca llegó a comercializarse. Se trataba de una computadora para el hogar que permitía controlar las cuentas y el inventario del hogar, además de los relojes digitales de la casa, el aire acondicionado y la antena de la televisión. También tenía un calendario, un reloj interno y un programa meteorológico que permitía procesar los datos de una estación meteorológica externa.

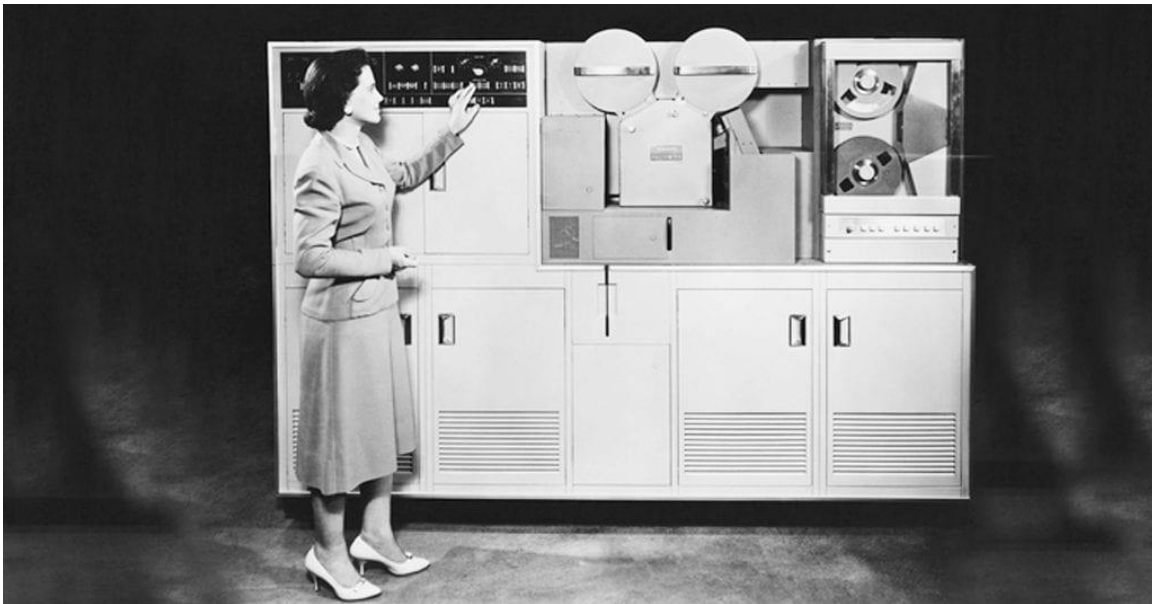


Ilustración 2 ECHO IV

En 1971 aparece el microprocesador, lo que hizo que el precio de la electrónica bajara y que se hicieran cada vez más pruebas con los nuevos elementos electrónicos que surgieron a raíz de estos avances.

A partir de los 80, tras muchas pruebas durante los 70, empiezan a ser comunes determinadas formas de automatización, por ejemplo, puertas de garaje, sistemas de seguridad, luces con sensor de movimiento o control de la temperatura por termostato. Esto se debe a la integración de los sistemas eléctrico y electrónico en busca de una comunicación integral de los diferentes elementos.

A raíz del aumento de cableado en los hogares, aparece el llamado Sistema de Cableado Estructurado (SCE), Structured Cabling System (SCS) en inglés. El SCE comenzó siendo una infraestructura de cables que integra múltiples sistemas. A día de hoy, se ha convertido en toda una metodología, basada en una serie de estándares, para realizar el diseño y la instalación de la infraestructura cableada de cualquier red, ya sea eléctrica, telefónica, de red de área local (LAN) o, más recientemente, de fibra óptica [1].

Este sistema de cableado conectaba, con diferentes cables y conectores estándar, todo tipo de dispositivos a una misma red, facilitando la comunicación entre estos.

En 1984 aparece en Estados Unidos un programa para la domótica llamado SAVE que, aprovechando el nacimiento en 1975 del protocolo X10 y la infraestructura SCE de la red eléctrica, permitía realizar acciones por control remoto. El X10 es un protocolo de comunicaciones que controla de forma remota diferentes dispositivos X10 a través de la línea eléctrica, tratándose estos elementos de módulos, por ejemplo, módulos para lámparas, timbres o de tomas de corriente; controlados por diferentes controladores X10.

Este protocolo se encuentra dentro de lo que llamamos Comunicaciones de la línea de potencia (PLC, Power Line Communications), que son comunicaciones a través de la red eléctrica. Estas comunicaciones funcionan enviando señales de una frecuencia mucho mayor que las de la señal eléctrica de 50 Hz que se utilizan para comunicar diferentes dispositivos entre sí. Concretamente, el protocolo X10 es parte de la tecnología PLC de control, cuyos estándares funcionan a una frecuencia por debajo de los 500 kHz [2].

El protocolo X10 funciona con un emisor y un receptor. El emisor manda señales de 2,5V a 120kHz de frecuencia en forma de ráfagas, de manera que cada transmisión de 1 milisegundo en la que existe una ráfaga corresponde a un bit con valor 1, mientras que la ausencia de dicha ráfaga corresponde a un bit de valor 0, transmitiendo una señal digital al receptor. Las ráfagas se mandan de forma que corresponda con el 0 en las tres fases de los sistemas trifásicos [2].



Ilustración 3 Módulos X10



Ilustración 4 Controladores X10

Con la llegada de las redes de comunicación surgen nuevas formas de integrar los diferentes elementos del hogar en una sola red conectada a internet al margen de la red eléctrica. Además, cada vez aparecen redes inalámbricas más veloces y protocolos más ligeros que permiten optimizarlas al máximo, como el WiFi, que aportan versatilidad

al sistema ya que no requieren una instalación compleja y permiten reubicar fácilmente los diferentes equipos.

En el panorama actual, los principales sistemas domóticos se diferencian en varias características, siendo la primera de ellas si la comunicación es cableada o inalámbrica. Los sistemas cableados ofrecen más robustez, aunque también una mayor inversión, son los que se suelen utilizar para grandes proyectos en los que es importante que el sistema sea robusto. Por otra parte, dentro de los sistemas inalámbricos encontramos por radiodifusión (Zigbee), radiofrecuencia, WiFi y Z-Wave (radio para domótica). Entre estas redes, ha cobrado una mayor importancia el WiFi cuya universalización en equipos y redes ha propiciado la conectividad de no sólo ordenadores sino también teléfonos, televisores o electrodomésticos. A raíz de esto se populariza el término Internet de las cosas (IoT, Internet of Things), que se refiere a la creciente interconexión digital de objetos cotidianos.

La otra gran diferenciación es entre sistemas estándar, con protocolos de comunicaciones estándar, por ejemplo, el X10 mencionado anteriormente, o el MQTT, desarrollado por IBM y utilizado en dispositivos como los altavoces Amazon Echo; o con sistemas propietarios, en los que cada fabricante utiliza su propio protocolo de comunicación, por ejemplo, Google Home. Pasaremos a explicar los principales sistemas de domotización actuales.

4.1. DOMÓTICA CABLEADA (KNX)

Dentro de la domótica cableada la marca más conocida actualmente es KNX, una asociación que, en España, agrupa a las empresas que trabajan con el estándar mundial KNX. Según la propia asociación, KNX es un estándar mundial y abierto que sucede y aglutina a los estándares previos de domótica cableada europeos (European Installation Bus, European Home Systems y BatiBUS). Los dispositivos KNX se pueden conectar y comunicar por medio de par trenzado, radio frecuencia, red eléctrica o IP/Ethernet. Estos dispositivos pueden ser tanto sensores como actuadores que se utilizan para el control del equipamiento de los edificios, por ejemplo, iluminación, persianas, sistemas de seguridad, gestión energética, calefacción, sistemas de ventilación y aire acondicionado, sistemas de supervisión y señalización, interfaces a servicios y sistemas de control de edificios... Todas ellas, funciones pueden ser controladas, supervisadas y señalizadas utilizando un sistema uniforme [3].

Un sistema KNX cuenta con 4 grupos de elementos físicos o hardware: actuadores, elementos que conectan físicamente con aquello que queremos controlar; sensores, encargados de recoger datos u órdenes, por lo que se incluyen interruptores y

pulsadores; pasarelas, que permiten conectar otros sistemas con distintos protocolos con nuestro sistema; y acopladores, que se encargan de separar físicamente el cableado de cada dispositivo del bus.

Por último, en cuanto a software tendremos por un lado el de gestión, que es el que se utiliza para poner en marcha el sistema y configurar los dispositivos; y por otro el de control, que permite controlar y visualizar la instalación. La herramienta de gestión es necesaria para configurar los dispositivos KNX y es esta asociación la que se encarga de suministrarla y regularla.

Actualmente encontramos este tipo de instalaciones en edificios de oficinas e industriales, ya que el precio es bastante elevado, por lo que no resulta rentable a pequeña escala.

4.2. RADIODIFUSIÓN (ZIGBEE)

ZigBee es un estándar que define una serie de protocolos de comunicación utilizados en redes inalámbricas de bajo rango y transferencia de datos. Los dispositivos pueden trabajar en frecuencias de 868 MHz, 915 MHz y 2.4 GHz, con una transmisión de datos máxima de 250000 bits por segundo. Por este motivo, ZigBee se utiliza principalmente para aplicaciones con requerimientos de baja transmisión de datos, bajo coste y maximización de la vida útil de las baterías, estando la mayor parte del tiempo en un modo de ahorro de energía de forma que pueden funcionar durante años antes de necesitar sustituir las baterías. Este estándar desarrollado por la ZigBee Alliance tiene como base el IEEE 802.15.4, con este protocolo tenemos un rango de entre 10 y 100 metros y una transmisión de datos entre 20 y 250 Kbps [4].

La red que utiliza ZigBee tiene una baja tasa de transferencia y compatibilidad en comparación a la red WiFi, que funciona de forma similar. No obstante, también permite conectar más dispositivos a una misma red, tiene un menor consumo de energía y puede trabajar en múltiples frecuencias.

4.3. IOT

Actualmente, la forma más accesible de domotización o automatización del hogar es el IoT. Esto es principalmente debido a la omnipresencia del WiFi en los elementos de la vida cotidiana, como móviles y electrodomésticos, y la comercialización de dispositivos que permiten, de forma muy sencilla, la configuración de este Internet de las cosas para cualquier vivienda, además, con muy bajo coste. Los dispositivos IoT comerciales más conocidos son los altavoces inteligentes.

4.3.1. *Altavoces Inteligentes*

En el panorama actual, la principal automatización comercial inalámbrica se hace por medio de unos altavoces inteligentes que comunican a su vez con un asistente virtual. Este asistente puede controlar diferentes dispositivos inteligentes (Smart Devices), compatibles con el sistema. La limitación que presentan estos sistemas de automatización es la necesidad de compatibilidad para todo lo que quieras controlar con ellos.

Hablaremos de los dos casos más conocidos de forma actual, Google Home y Amazon Echo, más conocidos por el nombre del asistente, Alexa.

4.3.1.1. *Google Home*

El sistema de automatización de Google Home se basa en un altavoz inteligente que permite utilizar comandos de voz, con los que se interactúa con Google Assistant. La integración de este altavoz y de Google Assistant con otros dispositivos es la que permite la automatización en el hogar. Este altavoz surgió con la idea de competir con Amazon y sus altavoces Amazon Echo.

En este caso, existe una dependencia de la compatibilidad de los dispositivos que quieras utilizar con el sistema de Google Home.



Ilustración 5 Dispositivos Google Home

4.3.1.2. *Alexa*

Alexa es la inteligencia artificial de Amazon, presente en diferentes dispositivos de la marca, principalmente los Amazon Echo. De forma similar a Google Home, se trata de un altavoz inteligente que se comunica con otros dispositivos compatibles.

El equivalente a Google Assistant o al asistente artificial de Apple, Siri, es una inteligencia artificial con reconocimiento de voz que te permite comunicarte con ella, teniendo en cuenta una palabra clave o de activación a partir del cual se recibe la comunicación, en este caso "Alexa".

Los dispositivos Echo, además de poder comunicarse con otros dispositivos compatibles conectados con la red, permiten el almacenamiento de varias aplicaciones propias o creadas por usuarios, llamadas "habilidades".

De nuevo, existe la misma dependencia o limitación a aquellos dispositivos compatibles.



Ilustración 6 Dispositivos Amazon Echo

4.3.2. Otros dispositivos IoT comerciales

A día de hoy, los dispositivos IoT o "conectados" existen dentro de todos los campos, desde electrodomésticos con aplicaciones para móvil propias y compatibles con los asistentes inteligentes hasta actuadores mucho más sencillos como relés, enchufes o bombillas. Hablaremos de algunos de los más conocidos comercialmente.

Enchufes, interruptores y otros actuadores inteligentes de diferentes marcas. Existe una gran variedad de actuadores inteligentes, por ejemplo, luces LED, enchufes e interruptores que se activan y desactivan a través del WiFi, sistemas de climatización, proyectores, difusores, cámaras... Lo más importante en el caso de estos elementos es que sean compatibles con el sistema IoT en el que queramos integrarlos.

Phillips Hue y otras bombillas inteligentes. En el caso de esta bombilla, compatible con muchos de los controles IoT, incluyendo Alexa; es necesario utilizar un puente que esté conectado al router de la casa gracias al cual se conectan todas las bombillas de la marca. Permite el control de color de la luz, así como la intensidad.

Home Connect y otros electrodomésticos inteligentes. Home Connect es el nombre que recibe el software que permite conectar los electrodomésticos pertenecientes a BSH, con marcas como Bosch, Siemens, Balay o NEFF; a otros dispositivos a través de la aplicación para Android e iOS. Es compatible también con otros softwares de control, destacando IFTTT, el cual permite interconectar no sólo dispositivos sino también cuentas de correo electrónico, Twitter o Uber, por ejemplo.

4.3.3. IFTTT

Para aliviar parte de esta limitación se integra en estos dispositivos IFTTT (IF This, Then That), un servicio web con el que crear y programar acciones, automatizando diferentes tareas en internet. Este servicio permite, por ejemplo, que Alexa se comuniquen con aplicaciones de Google, pero también que diferentes aplicaciones o redes sociales se comuniquen entre sí, así como con otros dispositivos inteligentes.



Ilustración 7 Diagrama funcional IFTTT

De esta forma, podremos integrar todo aquello que sea compatible con IFTTT en nuestro sistema controlado por un altavoz inteligente, sin embargo, esto sigue limitándonos e impidiéndonos utilizar sensores o actuadores tradicionales cableados sin algún tipo de "adaptador".

Con todas estas alternativas, podemos ver que el avance de la domótica ha sido muy grande en los últimos años. Encontramos un cierto vacío en sistemas a pequeña escala, de comunicación inalámbrica ya que es más barato para un sistema pequeño; pero que permitan interactuar con elementos no inteligentes o no compatibles con estos servicios.

Es en este contexto en el que surge la idea de una implementación de los protocolos inalámbricos utilizados en estos diferentes sistemas en módulos de microcontroladores que se encuentren cableados a los elementos físicos (sensores y actuadores).

5. MARCO TEÓRICO

Este proyecto se compone de dos partes fundamentales que desarrollar:

- Hardware o soporte físico del proyecto, en el que se incluyen sensores, microcontroladores, que estarán integrados en placas microcontroladoras; actuadores y otros dispositivos. Se va a hablar especialmente de lo que son los dispositivos compatibles con el Internet of Things o dispositivos IoT.
- Software o soporte lógico, en el que se tratará la programación del sistema, hablando de los lenguajes utilizados y de los protocolos, arquitecturas y diferentes elementos IoT.

5.1. DEFINICIÓN DEL SISTEMA.

El sistema domótico a implementar se fundamenta en el IoT, por lo que hablaremos de lo que es necesario para que este concepto se lleve a cabo en cualquier sistema.

El IoT es el concepto que se da a la interconexión de dispositivos y objetos a través de una red, que puede ser privada o por Internet, de forma que interactúen entre sí. Cualquier objeto o dispositivo es susceptible de convertirse en parte de la red IoT, ya sean dispositivos mecánicos y sensores u objetos cotidianos como el calzado, la ropa y los electrodomésticos; siempre y cuando puedan ser conectados a internet e interactuar de forma autónoma, sin intervención humana. El objetivo final es conseguir interacciones máquina a máquina o machine to machine (M2M) [5].

Así pues, una cosa dentro de este internet de las cosas, puede ser cualquier objeto natural o artificial al que se le asigne un dirección IP y que se capaz de transferir datos a través de una red [6].

Un sistema IoT se compone de las siguientes partes:

- Dispositivos IoT, que serán los encargados tanto de recopilar información para el sistema (sensores) como de realizar las acciones que requiera éste (actuadores). En este caso, los sensores y actuadores serán tradicionales y el dispositivo IoT que los comunique con la red será una placa microcontroladora.
- Protocolos de comunicación, que permiten que los diferentes dispositivos "hablen" entre sí en un mismo idioma. En este caso utilizaremos un protocolo llamado MQTT.

- Broker MQTT, dado que vamos a utilizar este tipo de comunicación, será necesario un Broker que se encargue de alojar todos los mensajes publicados y leídos por cada elemento. En nuestro sistema, hará la función de registro de comunicaciones
- Plataforma IoT, que se encarga de recopilar y procesar la información recibida, además de dar respuestas en función de dicha información. En nuestro caso, estaríamos hablando del principal cerebro del sistema.

Para poder desarrollar los diferentes elementos mencionados, haremos de forma previa un esquema que muestre cuáles van a ser y cómo se van a comunicar.

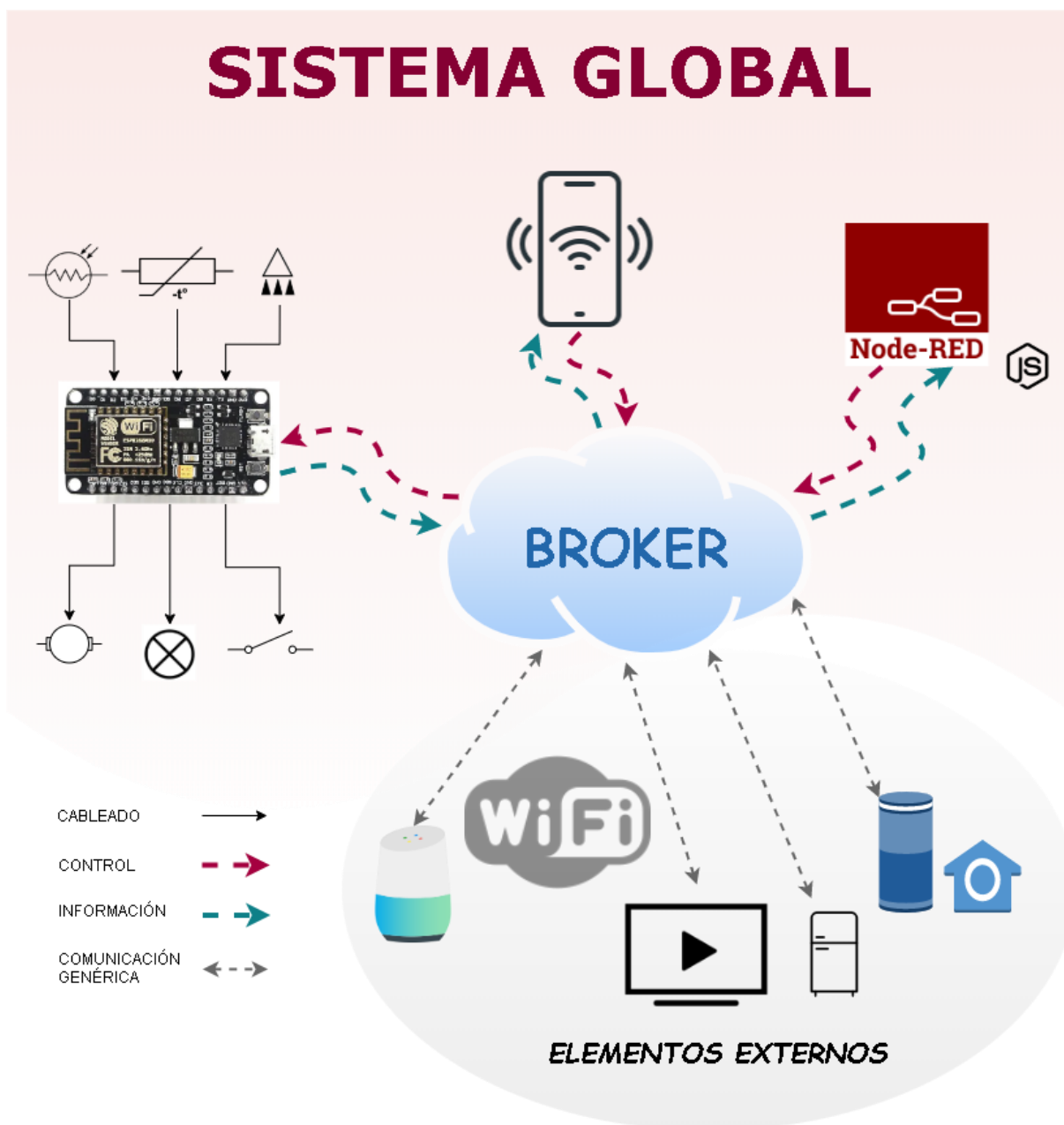


Ilustración 8 Esquema global del sistema

5.2. HARDWARE.

5.2.1. *Sensores.*

Los sensores son un elemento primordial e indispensable en cualquier tipo de automatización, ya que son los que recogen información del entorno para el sistema, permitiéndole conocer los valores que va adquiriendo la variable, zona o elemento a controlar. Sin sensores, el sistema no tendría forma de conocer las variaciones en el entorno y, por tanto, no sería capaz de autorregularse y autocontrolarse. Esto quiere decir que tendríamos que ser nosotros los que ajustáramos dichos parámetros que regulan el sistema, lo que dejaría de ser un sistema automático.

Como define la revista Nature: "Los sensores son dispositivos que detectan y cuantifican aspectos físicos de nuestro entorno. Las medidas pueden ser leídas directamente o convertidas a una señal para ser procesada por sistemas eléctricos, hidráulicos o neumáticos. Pueden cuantificar todo, desde temperatura, intensidad lumínica y deformaciones mecánicas hasta salinidad, ritmo cardíaco y presión sanguínea." [7].

Dentro de los sensores encontramos dos grupos, sensores digitales o de estado, que son aquellos cuya salida sólo consta de dos valores (HIGH/LOW, 1/0, TRUE/FALSE...) en función al valor de la entrada; y analógicos, que son aquellos con diferentes valores de salida para cada valor de entrada.

Los sensores analógicos pueden además dar la salida de forma analógica o de forma digital, utilizando un conversor interno. Además, dependiendo de la resolución que tengan estos sensores, requerirán una o varias entradas digitales para transmitir esta información. Para conseguir la misma información con una buena resolución en una sola entrada digital surgen protocolos como "Single-Wire" o "One-Wire", dos protocolos utilizados, por ejemplo, en algunos sensores de temperatura con resoluciones de entre 8 y 12 bits.

En nuestro caso, hablaremos de sensores analógicos de temperatura y fotoeléctricos para medir la iluminación en el exterior, y de sensores digitales de presencia para la detección de personas en los alrededores de la vivienda.

5.2.1.1. Sensor de Temperatura.

Dentro de los sensores, empezaremos hablando del de temperatura, que es el que varía sus características en función de la temperatura a la que se encuentra.

Tenemos tres tipos, sensores termopares, termistores y RTD, dependiendo de su funcionamiento.

Los termopares funcionan gracias a dos hilos de diferentes metales que se encuentran unidos en un extremo. A este extremo lo llamamos junta caliente o de medición, mientras que en otro extremo separado encontramos la junta fría. Es la diferencia de temperatura entre ambas la que genera una tensión diferencial que recibiremos en nuestro dispositivo. Existen diferentes tipos dependiendo de los materiales de los que están compuestos, que varían su rango de actuación, siendo el más permisivo el termopar tipo K, compuesto de una aleación entre cromo y níquel y otra de aluminio y níquel, con un rango de entre -200°C y 1250°C . Fuera de las variantes más comunes, encontramos termopares para altas temperaturas, como el tipo R, que tiene un margen entre 0°C y 1750°C .

Por otra parte, los sensores RTD funcionan gracias a la resistencia a la temperatura del material que los compone, ya que se trata de un alambre bien enrollado que a su alrededor tiene un núcleo de vidrio o cerámica. Son idóneos para la medición en entornos industriales ya que son inmunes al ruido eléctrico.

Por último, los termistores son sensores compuestos por semiconductores que varían su resistencia en función de la temperatura. De esta forma, en los casos anteriores era el propio sensor el que generaba una diferencia de tensión al cambiar la temperatura, pero en este caso lo que hace que varíe la tensión en bornes es la resistencia del sensor, que a su vez varía con la temperatura. Dentro de estos termistores hay dos tipos, NTC que tiene menor resistencia según aumenta la temperatura, o lo que es lo mismo, menor voltaje; y PTC, que tiene mayor resistencia según aumenta la temperatura, es decir que es directamente proporcional.



Ilustración 11 Sensor RTD



Ilustración 10 Sensor Termopar



Ilustración 9 Termistor NTC

5.2.1.2. *Sensores Fotoeléctricos.*

Llamamos fotoeléctricos a aquellos sensores que varían alguna característica eléctrica (resistencia, voltaje, intensidad...) debido a las variaciones de intensidad de la luz.

Dentro de los sensores fotoeléctricos, podemos encontrar diferentes tipos según sus usos, sensores de luz que permiten detectar el nivel de luz, variando éste los valores de la salida, o detectores de una luz a una longitud de onda determinada, normalmente infrarroja, que permiten conocer la presencia o ausencia de un elemento en un lugar concreto.

En nuestro caso, hablaremos de los sensores de luz, y dentro de éstos, el más común y utilizado es el *Light Dependant Resistor* (LDR) o resistor dependiente de la luz, que, como su propio nombre indica, varía su valor resistivo en función de la intensidad de la luz. A este dispositivo también se le puede llamar fotorresistor o fotorresistencia.

El funcionamiento de este sensor se basa en el efecto fotoeléctrico, por el cual determinados materiales emiten electrones al incidir sobre ellos una radiación electromagnética, en particular la luz, ya sea visible o ultravioleta. Normalmente la resistencia de un sensor LDR pueden variar entre $1M\Omega$ o más cuando no recibe luz y 100Ω cuando incide la mayor cantidad de luz sobre éste.

Por otra parte, existen sensores digitales que permiten medir los luxes de forma mucho más precisa, teniendo un mayor rango de intensidad lumínica. Hablamos en concreto de sensores como el MAX44009 o el BH1750FVI, sensores que se comunican por bus I2C, permitiendo un rango amplio con alta resolución.

5.2.1.3. *Sensores de Presencia.*

Los sensores de presencia, que también pueden considerarse detectores de movimiento, son dispositivos que detectan un movimiento físico y dan respuesta cuando se produce éste, es decir, son sensores digitales. Como detectores de movimiento se pueden utilizar muchos tipos de sensores, por ejemplo, los detectores de luz mencionados anteriormente permiten conocer la presencia o ausencia de un elemento y, por tanto, pueden detectar el movimiento de éste.

Encontramos varios tipos de sensores de presencia, empezando por los sensores activos, que son aquellos que emiten luz, microondas o sonido y detectan el cambio en dicha emisión. Aunque en nuestro caso los consideremos como sensores digitales, con respuesta 0 o 1, algunos de estos, como los de ultrasonido, permiten medir distancias gracias al tiempo que tarda en recibir la respuesta. En este punto estarían incluidos los

sensores fotoeléctricos anteriores, pero también otros como los sensores de ultrasonidos, que emiten y reciben una onda de ultrasonido y cambian de estado al no recibir esa onda. De forma similar funcionarían los fotoeléctricos, pero con luz normalmente infrarroja. En el caso de los fotoeléctricos existen tres tipos: de **barrera de luz**, compuesto por un emisor de luz y un receptor que detectaría si se deja de recibir esa luz, es decir, si se corta la barrera de luz que genera el emisor; de **reflexión sobre**

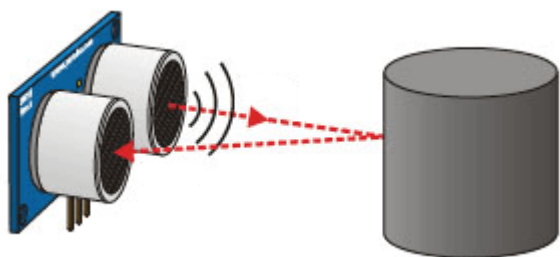


Ilustración 13 Sensor de presencia por ultrasonidos

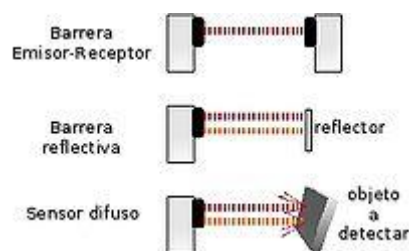


Ilustración 12 Tipos de sensores fotoeléctricos infrarrojos

espejo, en los que se utiliza un dispositivo que es a la vez emisor y receptor y un reflector catadióptrico, que refleja la luz emitida de vuelta al sensor, y, por último, de **reflexión sobre objeto**, que de nuevo, concentra el emisor y el receptor, emitiendo una luz infrarroja que al reflejarse en un objeto llega al receptor, momento en que se detecta dicho objeto.

Otro tipo de sensores de presencia son los sensores pasivos, que utilizan la detección de ondas infrarrojas, también llamados pasivos infrarrojos (PIR). Se llaman pasivos ya que, al contrario de los anteriores, no emiten ningún tipo de energía u onda para realizar la detección. Estos sensores miden la luz infrarroja que emiten los objetos dentro del campo de visión. De esta forma, no detectarían cualquier movimiento que pase por dicho campo, ya que trabajan a una longitud de onda determinada, normalmente ajustada al valor de la luz infrarroja que emite el cuerpo humano debido al calor que produce.



Ilustración 14 Funcionamiento del sensor PIR

Además, se pueden utilizar los acelerómetros como detectores de movimiento, ya que se basan en la fuerza aplicada para modificar las dimensiones de un material piezoeléctrico (material que genera un cierto voltaje al deformarse). De esta forma al apoyarse el objeto o persona a detectar sobre el sensor, se produciría una deformación detectable de forma eléctrica.

5.2.2. Alimentación del circuito.

Además de considerar los diferentes elementos que compondrán el circuito electrónico, deberemos encontrar una fuente de alimentación que permita convertir los 220 V de corriente alterna de la red eléctrica del hogar en 5 V para alimentar a las placas microcontroladoras. La fuente de alimentación cuenta con una transformación, un rectificado, un filtrado y una regulación del voltaje de entrada que permita dar una salida adecuada al circuito.

Para ello consideraremos una pequeña fuente de alimentación como pueden ser la HLK-PM01 o la TSP-05, que permitirán convertir el voltaje de forma sencilla, económica y ocupando un espacio mínimo.

Para el prototipo que se utiliza como demostración del funcionamiento, se dispone de un cargador universal con una salida de bornes de tensión, el cual nos permite tener una salida de 3 V, 4,5 V, 5 V, 7,5 V, 9 V o 12 V mediante un selector.



Ilustración 15 Cargador universal con salida de bornes.

5.2.3. Dispositivos IoT.

Llamamos Internet of Things (Internet de las cosas) a la conexión de diferentes elementos cotidianos a internet y entre ellos a través de esta red. Para que un dispositivo se considere IoT, debe contar con la capacidad de conectarse y comunicarse con internet. Existen dispositivos comerciales que son IoT, por ejemplo, los altavoces inteligentes mencionados anteriormente, pero también actuadores como bombillas,

interruptores o electrodomésticos, como hornos o lavavajillas que permiten un control por medio de una app. Además, existen otros dispositivos como algunos microcontroladores que, junto con la conexión a internet, permiten el control de los elementos citados anteriormente de forma individual.

Para el desarrollo de este proyecto se realiza un primer acercamiento al mundo IoT con un microcontrolador sin conectividad WiFi, utilizando el software Laboratory Virtual Instrument Engineering Workbench o LabVIEW como nexo con la red.

LabVIEW es un programa de National Instruments que permite diseñar y simular diferentes sistemas a través de bloques. Para esta aplicación se utilizan dos de las múltiples opciones que ofrece este software: la lectura del puerto serie del microcontrolador y la conexión con internet para el envío y recepción de datos. Esto nos permite la adaptación de dispositivos sin conexión WiFi, pero compatibles con LabVIEW, para que puedan funcionar como dispositivos IoT.

5.2.3.1. *Microcontroladores IoT.*

Una vez comprobada esta funcionalidad de LabVIEW, lo que haremos será elaborar un sistema independiente de la conexión a un PC, es decir, necesitaremos un microcontrolador que permita ser IoT.

Un microcontrolador es un microchip o circuito integrado que ejecuta los comandos que se graban en su memoria. Estos comandos se pueden modificar, es decir, es programable. Todos los microcontroladores constan de varios elementos básicos que cumplen funciones específicas:

- Unidad central de procesamiento (Central Processing Unit, CPU): La CPU es un circuito integrado que se encarga de efectuar las instrucciones del programa. A su vez, la CPU contiene una unidad aritmético lógica (ALU), que, como su propio nombre indica, realiza las operaciones aritméticas y lógicas; una unidad de control (CU) que interpreta y ejecuta las instrucciones de la memoria, y unos registros de todo lo que realiza.
- Memoria: Dentro de un microcontrolador encontramos dos tipos de memoria, la memoria RAM (Random Access Memory), en la que se guardan datos temporales que utiliza el procesador para realizar operaciones, y la memoria ROM (Read-Only Memory), donde se guardan los programas que se cargan en el microcontrolador y que el procesador lee para seguir las instrucciones. Dentro de estas memorias existen varios tipos, las memorias SRAM (Static Random Access Memory), que mantiene la información sin

necesidad de realimentarla, y DRAM (Dynamic Random Access Memory), que necesita un circuito dinámico de refresco; dentro de la RAM, y las memorias PROM (Programmable Read-Only Memory), que puede ser escrita y programada sólo una vez con un dispositivo especial; EPROM (Erasable Programmable Read-Only Memory), que puede borrarse exposición a luz ultravioleta, y la EEPROM (Electrically Erasable Programmable Read-Only Memory), que se puede borrar eléctricamente, especialmente la memoria flash, un tipo de EEPROM más novedosa que se puede borrar y reescribir de forma más rápida que la tradicional.

- Entradas/Salidas: Son aquellos pines que permiten al microcontrolador comunicarse con elementos externos.

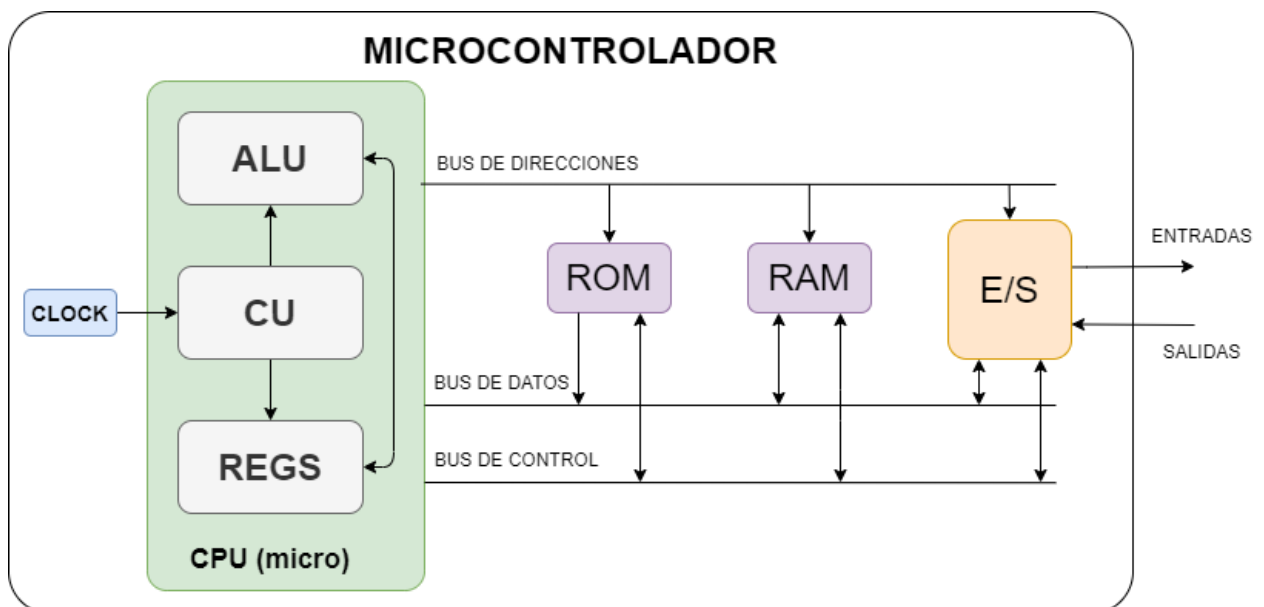


Ilustración 16 Esquema básico de un microcontrolador.

Los microcontroladores que permiten comunicación WiFi más conocidos y utilizados en la actualidad son los desarrollados por la empresa Espressif, concretamente se nombran como ESP8266 y, más reciente, ESP32.

En algunas placas de desarrollo, se utiliza sólo la CPU del microcontrolador y el resto de los elementos se externalizan, utilizando así la principal unidad de procesamiento en un ordenador, un microprocesador.

Los microprocesadores son circuitos integrados con una o varias CPU que, una vez integrados en un sistema con varios elementos (entradas, salidas, memoria...), son capaces de ejecutar todos los programas necesarios de un ordenador, incluyendo el sistema operativo de éste. Una de las principales placas de desarrollo del mercado actual

y de la que hablaremos más adelante, la Raspberry Pi, utiliza un microprocesador, el Broadcom BCM2711.

Todos estos microcontroladores, por sí solos, no serían capaces de llevar a cabo todas las funciones que se requieren de ellos, necesitan varios elementos con los que se comunican a través de las entradas y salidas. A las placas que integran estos microcontroladores y el resto de los elementos necesarios para las aplicaciones se les llama placas de desarrollo.

5.2.3.2. Placas de desarrollo.

Dentro de las diferentes placas de desarrollo existentes en el mercado, hay dos marcas que sobresalen entre las demás, Raspberry Pi y Arduino. Además, tenemos otras placas de desarrollo basadas en los chips de Espressif, en nuestro caso hablaremos de la placa de desarrollo NodeMCU, de la cual existen versiones tanto con el ESP8266 y como con el ESP32.

5.2.3.2.1. Raspberry Pi.

Las placas de desarrollo Raspberry Pi son desarrolladas y fabricadas por la Raspberry Pi Foundation. Se diferencian del resto de placas de las que hablaremos ya que es la única que se puede considerar un ordenador de placa simple (SBC) o reducida, esto quiere decir que puede funcionar como una computadora completa.

Actualmente el modelo más nuevo es la Raspberry Pi 4, aunque las versiones anteriores siguen a la venta en la página web. Este modelo cuenta con el microprocesador Broadcom BCM2711, motivo por el cual se considera un ordenador de placa simple, lo cual permite que ejecute un sistema operativo.

El sistema operativo oficial que utiliza Raspberry Pi es un software de código abierto basado en Linux, concretamente Linux Debian, llamado Raspbian. También funciona con otros sistemas, por ejemplo, Windows 10.

Otra de las principales diferencias de la SBC en contraposición a las placas microcontroladoras es el tipo de entradas y salidas que posee, ya que en este caso, las entradas y salidas no se encuentran dentro del chip principal, sino que son externas y por ello, pueden ser entradas y salidas de todo tipo, incluyendo USB, HDMI, Ethernet o puertos para cámara, además de 40 entradas/salidas de propósito general (General Purpose Input/Output, GPIO), puertos que también poseen las placas microcontroladoras.

5.2.3.2.2. Arduino.

Arduino se constituye como una plataforma electrónica de código abierto, que incluye software y hardware, Arduino Integrated Development Environment (IDE) y las

placas de Arduino. Estas placas son capaces de leer entradas y convertirlas en salidas, el funcionamiento de éstas depende de una serie de instrucciones que se mandan al microcontrolador de la placa, para lo que se utiliza el software ya mencionado, Arduino IDE, y el lenguaje de programación Arduino [8].

Dentro de esta marca existen multitud de placas en función del microcontrolador o los elementos que la componen, entre las más conocidas encontramos Arduino UNO, actualmente la versión R3, que utiliza el microcontrolador ATmega328P. También podemos encontrar ordenadores de placa simple en Arduino, concretamente la placa Arduino Yún es un ejemplo, aunque mucho más sencillo que Raspberry Pi.

Nos centraremos en las placas microcontroladoras que pueden funcionar como dispositivos IoT, es decir, aquellas con conectividad a internet, ya sea WiFi o Ethernet. Dentro de estas, destacaremos la placa Arduino MKR WiFi 1010, la cual tiene como principal microcontrolador un SAMD21 Cortex ARM M0, integrado en un módulo de Atmel, pero incluye un módulo de ESP32 preparado por U-BLOX, el NINA-W10.

5.2.3.2.3. NodeMCU.

NodeMCU es el nombre que reciben dos placas de desarrollo, la primera de ellas basada en el microcontrolador ESP8266 y la segunda en el ESP32. Una de las principales diferencias entre el ESP8266 y el ESP32 es la presencia de conectividad bluetooth en el caso del segundo, además de contar con más entradas y salidas.

De la mano de estas placas o kits de desarrollo NodeMCU, existía NodeMCU como firmware de código abierto basado en el lenguaje de programación Lua y desarrollado inicialmente para el ESP8266, precisamente para el kit de desarrollo NodeMCU. Posteriormente, gracias al apoyo de la comunidad, este firmware se puede utilizar en cualquier módulo o kit de ESP, incluido el ESP32 [9].

El lenguaje de programación Lua nace en 1993 en la Pontificia Universidad Católica de Rio de Janeiro, Brasil. Este lenguaje ha evolucionado y se utiliza en todo tipo de aplicaciones industriales, además de ser uno de los lenguajes líderes en desarrollo de videojuegos [10].

Gracias a todo este desarrollo en los microcontroladores ESP, los módulos NodeMCU son compatibles, a día de hoy, con el IDE de Arduino, por lo que se pueden programar como este.

5.2.3.2.4. Placas basadas en STM32.

Los STM32 conforman una familia de microcontroladores de 32 bit desarrollados por la empresa STMicroelectronics. Su bajo coste, a pesar de contar con la familia de

microprocesadores ARM Cortex-M, superior al AVR de Arduino, han provocado el crecimiento de la popularidad de estas placas de desarrollo [11].

Entre ellas nos centraremos en la placa dedicada a IoT que ha desarrollado STMicroelectronics, el kit de descubrimiento B-L475E-IOT01A el cual, además de contar con conectividad tanto WiFi como Bluetooth, integra otros componentes, como pueden ser un tag NFC con una antena NFC impresa, dos micrófonos, un sensor capacitivo de humedad relativa y temperatura o un barómetro, entre otros. Todo ello puede ser muy útil para aplicaciones IoT, además, estas placas se pueden programar a través de Arduino IDE y permiten conectividad también con la placa Arduino UNO como microcontrolador secundario [11].

5.2.4. Actuadores

Hablaremos de los actuadores que tendremos que utilizar a la hora de colocar el sistema en la vivienda. Dentro de esto hablaremos de los motores de persianas y de los relés que sirven para controlar los interruptores de la luz o de la climatización.

5.2.4.1. Motores para persianas.

El factor principal a considerar en estos motores, es si el control va a ser cableado o inalámbrico. Dado que el sistema es mixto, se considera un control cableado, siendo este más barato e igualmente compatible con esta aplicación.

Aunque se puede realizar el diseño con un motor tradicional para controlar las persianas, existen motores especializados para persianas domésticas, los motores tubulares, que se encuentran de forma comercial a un bajo coste. Otros motores comunes a la hora de controlar persianas son los motores de cinta, que dan uso a la correa ya presente para subir y bajar la persiana. Su instalación es más sencilla, ya que no es necesario desmontar el tambor de la persiana para introducir el motor. En contra, ocupan un mayor volumen si son exteriores, y dan menos opción a un control externo al módulo comercial, que ya lo incluye.

Los motores tubulares para persianas comerciales actuales incluyen: el cable (entre tres y cinco hilos, dependiendo del tipo de control), una rueda motriz para anclar al eje octogonal de la persiana y un anclaje a la pared en el extremo contrario a dicha rueda, así como los tornillos y pasadores necesarios para su montaje. Los finales de carrera ajustables como elemento de seguridad están integrados en el propio motor permitiendo limitar los valores máximo y mínimo del mismo.

El control de dichos motores se realiza a través de uno o dos interruptores. En el primer caso, se utiliza un selector de tres posiciones, parada, subida y bajada, utilizándose habitualmente un pulsador cuyo reposo sería la posición de parada. Al contar con dos interruptores, el primero controlaría la alimentación del motor y el segundo el sentido de giro. A la hora de automatizarlo, los interruptores se sustituyen por dos relés controlados por el sistema.

5.2.4.2. Relés

Un relé es un dispositivo electromecánico que relaciona dos circuitos diferentes entre sí, permitiendo al primer circuito, o circuito electromagnético, controlar uno o varios contactos del segundo circuito, normalmente de potencia. De esta forma, el circuito primario recibiría una señal de baja tensión por parte del controlador siendo el secundario el que activaría la carga de alta tensión [12].

Estos componentes resultan imprescindibles a la hora de realizar un proyecto domótico como el que se presenta, ya que nos permiten controlar con dispositivos IoT de baja tensión casi cualquier otro elemento del hogar, ya sean luces, motores o cualquier otro elemento controlado por la red eléctrica de la casa [12].

El funcionamiento de un relé se basa en utilizar un electroimán, es decir, una bobina enrollada alrededor de un núcleo metálico, y unos contactos eléctricos para el circuito secundario. Normalmente los contactos son tres, un contacto común que sería el móvil, un contacto NO o normalmente abierto y un contacto NC o normalmente cerrado, de forma que la posición de reposo sería con el contacto común o C conectado al contacto NC o normalmente cerrado. Al activar el electroimán y, por tanto, el relé, se conectaría el común C al NO gracias a una armadura que haría pivotar al contacto común [12].

5.3. SOFTWARE

5.3.1. *Protocolos de comunicación*

Llamamos protocolo de comunicación al conjunto de reglas que permite que dos elementos de una red se comuniquen entre ellos, transmitiendo información a través de alguna variación en una magnitud física. De esta forma, se definen la semántica, sintaxis y sincronización de esta comunicación, aunque también se suele establecer algún método que permita recuperar errores.

Los protocolos que se utilizan varían en función de la red de comunicación, de esta forma para la red telefónica tenemos por ejemplo el GSM o el 3G, para televisión el PAL o HDTV, y para redes de ordenadores se suelen utilizar el TCP, IP, UDP, SSL, entre otros [13].

Los protocolos pueden adoptarse como estándares si se controlan por organizaciones como IEEE o ISO, lo que implica que sus especificaciones serán públicas, pudiendo ser utilizados por cualquier empresa [13].

En nuestro caso nos centraremos en los protocolos de redes de ordenadores, y concretamente el protocolo más importante a la hora de utilizar Internet, el Protocolo de Control de Transmisiones (TCP) y el Protocolo de Internet (IP), siendo su combinación, el TCP/IP, un estándar de Internet [13].

Este protocolo está estructurado en cinco capas independientes, la capa física, la capa de acceso a la red, la capa de internet, que permite la comunicación entre dos redes diferentes y es en la que actúa el protocolo IP; la capa de transporte, que permite la seguridad del intercambio de datos y es en la que actúa el protocolo TCP, y, por último, la capa de Aplicación, en la que está la lógica que permite posibilitar las aplicaciones [13].

Es precisamente en esta última capa en la que se encuentran los protocolos específicos para el IoT. Los sistemas IoT requieren de protocolos específicos ya que el propio sistema necesita ciertas características, entre ellas encontramos el poder tener multitud de dispositivos conectados a la misma red, que dichos dispositivos puedan conectarse y desconectarse de forma sencilla, sin que sean dependientes entre sí y sin que esto modifique el comportamiento global del sistema; que permita comunicaciones simultáneas y con una respuesta rápida, una gran compatibilidad con todos los dispositivos, sistemas operativos y lenguajes posibles, entre otras [14].

Estos protocolos específicos son protocolos Máquina a máquina, Machine to Machine (M2M), entre los que destacaremos, el Constrained Application Protocol (CoAP) o el protocolo Message Queue Telemetry Transport (MQTT).

El protocolo CoAP está basado en el Protocolo de transferencia de hipertexto, Hypertext Transfer Protocol (HTTP), otro protocolo de la capa de Aplicación, utilizado principalmente para la arquitectura web cuya versión segura (HTTPS), es actualmente la forma más común de comunicarse con un servidor web, como podemos ver al conectarnos desde cualquier navegador. Este protocolo permite que unos dispositivos y redes más limitados puedan funcionar de la misma forma que con el protocolo HTTP, utilizando paquetes de datos más pequeños, pero manteniendo el funcionamiento del primero, que permite que el cliente pueda solicitar la información al servidor y el servidor sólo envíe dichos cambios cuando el cliente los pida [14].

El protocolo que se utiliza en este proyecto es el MQTT, un protocolo muy ligero desarrollado por IBM que permite que muchos dispositivos se conecten de forma simultánea a un mismo servidor central, llamado Broker, que recibe los mensajes de todos los dispositivos emisores y permite que todos los receptores lean dichos mensajes. Al contrario que en el caso del protocolo anterior, el protocolo MQTT utiliza una metodología llamada Publish/Subscribe o Pub/Sub, la cual permite que cada vez que aparece una información nueva en el Broker, llegue a todos los dispositivos que la necesitan. Para ello, el "Subscriber" informa al Broker de qué mensajes quiere recibir en función de un tema o "Topic" determinado, recibiendo todos los mensajes que los diferentes "Publisher" manden o publiquen en el Broker bajo ese tema. Además, el hecho de que este protocolo esté basado en la mensajería Message Queue implica que los mensajes recibidos por el Broker no se pierden al desconectarse el cliente o Subscriber, se crea una cola para cada cliente que se mantiene hasta que todos los mensajes son entregados a éste, garantizando que no habrá pérdida de comunicación en caso de micro caídas de la red WiFi [14].

De esta forma, la estructura sería la que podemos observar a continuación, de forma que, si el "Subscriber" sólo pide información del tema /cocina, sólo le llegará la información que se publique bajo ese "topic". En cualquier momento este elemento podría suscribirse al tema /sala también o dejar de estar suscrito al tema /cocina.



Ilustración 17 Diagrama de funcionamiento Pub/Sub.

5.3.2. *Broker*

Llamamos Broker al servidor que recoge y registra toda la información del sistema IoT. Para cubrir este servicio, existen Broker MQTT de pago y gratuitos y pueden funcionar de forma local o por internet, de forma que, si están alojados en un servidor externo, se puede necesitar un pago por ese servicio.

Entre los Broker MQTT, los más conocidos son los broker locales Mosquitto y Mosca, presentando mejores características y siendo más sencillo y adaptable Mosquitto. Normalmente los Broker online están mantenidos por un proveedor de dichos servicios, pero también están basados en alguno de estos dos programas básicos. Mosquitto tiene el beneficio de ser compatible con más sistemas operativos, incluido Linux y Raspbian, sin embargo, Mosca está basado en node.js y puede instalarse como un nodo de la plataforma IoT Node-RED [15].

Como ejemplo de Broker online podemos hablar de los servicios que presta Amazon bajo el nombre de Amazon Web Services (AWS), ya que entre ellos existe un Broker MQTT. Amazon Web Services es una plataforma en la nube que ofrece diferentes servicios, algunos gratuitos y otros de pago, de forma que no es necesario tener ni mantener un servidor propio para poder aprovecharse de dichos servicios. De esta forma, el Broker estaría alojado en la nube de AWS y habría que pagar por el mantenimiento y el uso del servicio. También hay otras empresas que ofrecen esta clase de servicios que permiten disponer de un Broker de forma online, incluso con una opción básica gratuita.

A este Broker de nuestro sistema estarán conectados todos los dispositivos IoT que envíen o reciban información del resto del sistema y la plataforma IoT, en este caso Node-RED, que es precisamente el siguiente punto a tratar.

5.3.3. *Plataformas IoT*

Una plataforma de procesamiento IoT recibe la información que se publica en el Broker por parte de todos los dispositivos, analizándola y dando a los dispositivos a través del Broker una respuesta que los dispositivos puedan interpretar, actuando en consecuencia. Esto implica que la mayor parte de la programación estará alojada en la plataforma, mientras que los dispositivos (microcontroladores) sólo recopilarán y enviarán los datos de los sensores al Broker y controlarán a los actuadores al recibir una información determinada de éste. Entre las plataformas, de forma similar a los Broker, encontramos algunas de pago y otras gratuitas, algunas en la nube y otras de forma local.

Existen por ejemplo la Samsung Artik Cloud, Arduino Cloud y ThingSpeak de MathWorks; que pertenecen a grandes empresas, pero también Thinger.io o Node-RED que son plataformas de código abierto y permiten ser instaladas de forma local, aunque Thinger.io también está disponible en un servicio web de pago.

5.3.3.1. Node-RED

En nuestro proyecto se utiliza la plataforma Node-RED gracias a la sencillez de instalación de forma local que tiene y la posibilidad del dashboard con la información e interacción con el sistema. Por otra parte, al ser de código abierto hay una gran comunidad y un gran número de funciones creadas por ésta, lo cual hace más sencilla la programación.

En Node-RED se trabaja con unidades llamadas nodos, los cuales se dedican a una tarea concreta, por ejemplo, hay nodos de entradas, ya sean entradas MQTT, HTTP, TCP, Serial...; nodos de salidas de los mismos tipos, nodos para generar un dashboard con botones o barras, por ejemplo; nodos de funciones que se pueden programar con código y muchos otros. Estos nodos se conectan entre sí, agrupándose en flujos o flows, lo que facilita una programación visual.

Esta plataforma trabaja con JavaScript Object Notation (JSON), es decir, utiliza programación JavaScript basada en objetos. Para entender mejor el funcionamiento de este tipo de programación, así como los demás utilizados en el proyecto, hablaremos a continuación de los lenguajes de programación utilizados.

5.3.4. Lenguajes de programación

Llamamos lenguaje de programación a la notación que se utiliza en las instrucciones con las que comunicarse con el hardware, es decir, con la que escribir las órdenes necesarias para la realización de un proceso determinado. De esta forma nace un programa, que, siguiendo las reglas que define el lenguaje, permite dar una determinada secuencia de acciones que pueda entender el dispositivo programado [16].

5.3.4.1. Lenguaje C y C++

Este lenguaje nace a principios de los 70 desarrollado como un lenguaje de implementación para el sistema operativo Unix, ambos desarrollados por los laboratorios Bell. Aunque hasta principios de los 80 el lenguaje estaba asociado exclusivamente con Unix, su uso se ha extendido hasta el punto de ser uno de los lenguajes más usados en la industria de la computación, siendo a día de hoy la base de muchos otros lenguajes [17].

Este lenguaje surge en los inicios de los sistemas de computación y, a la par del desarrollo de estos sistemas de diferentes formas, también surgen nuevos lenguajes.

Así, se realiza el diseño de C++, un lenguaje que, además de ser cercano al lenguaje de las máquinas, como ocurre con el lenguaje C; se busca que el lenguaje sea también cercano al problema que resolver, de forma que las soluciones puedan expresarse de forma directa y concisa. De esta forma, C++ aporta un mapeado directo de las operaciones integradas y los tipos de hardware, consiguiendo un uso eficiente de la memoria y de las operaciones de bajo nivel; además de mecanismos asequibles y flexibles para utilizar tipos definidos por el usuario, manteniendo los usos originales [18].

Como una mejora del lenguaje C, basados en el nuevo C++ surgen los dos lenguajes principales para este trabajo, el lenguaje Arduino y el lenguaje JavaScript.

5.3.4.2. Lenguaje Arduino

El lenguaje de programación Arduino es el que se utiliza en la plataforma Arduino, un lenguaje propio basado en C, utilizando algunas funciones de C++, y funciona de forma muy similar a estos.

Este lenguaje se constituye por tres partes principales, funciones, que permiten controlar la placa y realizar computaciones; variables, como se llama a los tipos y las constantes de datos, y estructuras, elementos del código de Arduino heredados de C++ [19].

5.3.4.3. Lenguaje JavaScript

El lenguaje JavaScript (JS) es ligero e interpretado, siendo un lenguaje utilizado en páginas web, así como en otros entornos sin navegador, por ejemplo node.js y Adobe Acrobat [20].

Este lenguaje desarrollado por Netscape y orientado a objetos se utiliza en multitud de páginas web y aplicaciones de servidor en el mundo. Los scripts que los componen son parte del estándar ECMAScript. Su sintaxis básica es similar a Java y C++ con el objetivo de reducir el tiempo de aprendizaje del lenguaje, evitando añadir nuevos conceptos [21].

Dentro de estos scripts del estándar ECMAScript que componen JavaScript, se forman subconjuntos. Basado en uno de dichos subconjuntos encontramos JavaScript Object Notation (JSON), o Notación de Objetos de JavaScript, un formato ligero de intercambio de datos, siendo simple de leer y escribir para humanos, así como fácil de interpretar y generar para las máquinas. Como todos los mencionados hasta ahora, es

parte de la familia de lenguajes C, compartiendo elementos con muchos de ellos, especialmente C++ y JavaScript [22].

De esta forma, en JSON encontramos dos clases de estructuras, lo que en muchos lenguajes se llama objeto o estructura, una colección de parejas de nombre/valor, y lo que se suele llamar arreglos, vectores o "arrays", una lista ordenada de valores. Cada valor puede ser una cadena de caracteres o "string", un número, un valor booleano (true/false), pero también un objeto o un "array", anidándose unas estructuras dentro de otras [22].



6. DESARROLLO

A la hora de desarrollar este proyecto, lo primero que deberemos tener en cuenta será el estudio de la casa a automatizar, estableciendo las necesidades que puede tener y eligiendo las funcionalidades que se ajusten mejor al que sería, en este caso, el usuario.

Posteriormente trataremos la elección de los diferentes elementos con los que implementaremos dicho sistema, hablando tanto de componentes, como pueden ser los sensores o las placas microcontroladoras, como de software y, específicamente, de la plataforma IoT y el broker MQTT.

Una vez realizado todo este previo a la implementación del sistema, se hablará de la primera prueba realizada en el laboratorio con LabVIEW como intermediario entre la placa Arduino UNO y la plataforma IoT, hablando de los desarrollos que se realizaron y cómo se relacionan con el sistema final.

Por último, trataremos las diferentes partes que constituyen el sistema final, hablando de su esquema general, del desarrollo en la placa microcontroladora y en físico, de la implementación de la programación en la plataforma y del dashboard, tanto en la plataforma como en el Smartphone, presentando finalmente un prototipo incluyendo todo lo mencionado.

6.1. ESTUDIO DE LA VIVIENDA A AUTOMATIZAR

Se trata de una vivienda con dos plantas con problemas de aislamiento e iluminación, los cuales trataremos de aliviar por medio de un **control de la iluminación y las persianas**, así como una **mejor gestión de la climatización**, especialmente aumentando el confort mediante la **activación remota o programada** de la misma. Este control inteligente es importante dadas las ausencias prolongadas de los usuarios, durante las cuales se haría una **mejor gestión de la energía**, desactivando este sistema y desajustándose la temperatura del hogar.

También se tendrá en cuenta la ausencia de **iluminación en la entrada principal**, teniendo que mantener una luz encendida durante toda la noche si alguien llega tarde a casa, problema que se solucionará mediante una luz inteligente que ahorraría energía.

Por otra parte, tendremos en cuenta la seguridad ya que, al ser una vivienda en un pueblo y no contar con ningún tipo de seguridad adicional, se puede beneficiar mucho

de una **simulación de habitabilidad**, evitando los robos que se pudieran producir al no estar nadie en casa.

Adicionalmente, se propone un **sistema de alarmas de despertador** que consistirán en la subida paulatina de las persianas haciendo que suene una alarma a la par.

Tabla 1 Tabla resumen de problemas y soluciones.

Problemas	Soluciones
Aislamiento pobre	Control de persianas para mejor aprovechamiento de horas de luz
Mala iluminación	Control de persianas y de la iluminación
Climatización poco eficiente Alto gasto en calefacción	Control de persianas y de calefacción Activación remota y programada de la climatización
Iluminación de la entrada principal	Activación automática de la luz de la entrada principal reduciendo consumo eléctrico
Baja seguridad	Simulación de habitabilidad que reduzca las posibilidades de robos
Confort	Mejoras de confort que permitan subir las persianas para despertar a los usuarios mientras suena música

6.2. ELECCIÓN DE COMPONENTES Y SOFTWARE

Los elementos que tendremos que elegir serán los **sensores**, contando con medir el nivel de luz, la temperatura y la presencia en la entrada; los **actuadores**, concretamente motores para persianas y relés; la **placa microcontroladora** que se encargará de recoger la información de los sensores y comunicarse con la plataforma para controlar los actuadores, la **plataforma** en la que se hará la gestión de toda esta información y el control del sistema y el **broker** en el que se alojará la información del sistema.

6.2.1. Sensores

Los sensores serán los encargados de medir las variables físicas que necesitamos dentro de nuestro sistema. Así pues, deberán ser adecuados para la aplicación que queremos, con lo que lo principal a tener en cuenta será el rango de medida del sensor, la forma en que le dará la salida a la placa microcontroladora y la precisión del sensor, además del precio.

6.2.1.1. Sensor de temperatura

Definiremos en primer lugar cuáles son los valores máximos y mínimos que deberá tener nuestro sensor de temperatura. Dado que la medida será en el interior de la casa, en cada una de las habitaciones, activando o desactivando individualmente los climatizadores de cada habitación, supondremos como temperatura mínima unos 0 grados centígrados para cubrir el caso de que la calefacción fallase, dando un margen bastante amplio, y como máximo unos 60 grados en el caso de que hiciera demasiado calor y pudiendo estar el sensor expuesto a la luz solar o sobrecalentarse por algún motivo. Los valores habituales máximos y mínimos en el interior de la casa son de aproximadamente 35°C sin ningún tipo de refrigeración y entre 5°C y 10°C sin ningún tipo de calefacción, por lo que es un margen más que suficiente. Además, idealmente tendríamos una precisión de menos de un grado de desviación, es decir, de $\pm 0,5^\circ\text{C}$ o, como máximo, de $\pm 1^\circ\text{C}$. Por último, buscaremos una salida del sensor analógica, principalmente para ocupar un solo pin del dispositivo con dicho sensor, el analógico, en vez de ocupar varios pines digitales y para evitar una programación muy pesada en la placa, ya que queremos que la programación de esta sea mínima, confiando el mayor peso del control a la plataforma.

Con estas características, consideramos varios sensores, como se puede observar en la siguiente tabla comparativa.

Tabla 2 Tabla comparativa de sensores de temperatura

	LM35	TMP36	DHT11	DS18B20	MF52
Rango	2 – 150 °C (V+, GND) -55 – 150 °C (V+, V-)	-40 – 150 °C	0 – 50 °C	-55 – 125 °C	-30 – 110 °C
Alimentación	4 – 20 V	2,7 – 5,5 V	3 – 5,5 V	3 – 5,5 V	---
Salida	Analógica -1-6 V	Analógica GND-Vin	Digital Single-Wire	Digital 1-Wire	Analógica GND-Vin
Precisión	±0,5 °C	±2 °C	±2 °C	±2 °C	<±0,5 °C
Resolución (digital)	N/A	N/A	8 bit	Programable 9-12 bit	N/A
Precio	1,55€/ud (15.5€/10ud)	2,4€/ud (12€/5ud)	2€/ud (10€/5ud)	1,1€/ud (11€/10ud)	0,03€/ud (3€/100ud)

Uno de los principales problemas que encontramos con la primera opción, el LM35, es que es necesaria una alimentación negativa para obtener grados bajo cero, algo que soluciona el TMP36. Además, en cuanto a salidas analógicas lo ideal sería que fueran por debajo de los 5 V, ya que es el voltaje máximo más común entre las placas microcontroladoras, e incluso que pudiera operar a 3,3 V, dado que hay varias placas que ofrecen una salida de alimentación de componentes de dicho voltaje.

Los sensores que nos dan salidas digitales, cada uno con su propio protocolo, podrían ser buenas opciones, sin embargo, la precisión que tienen no es demasiado buena. Podrían ser mejores opciones si se quisiera medir la humedad de forma adicional, ya que ambos incluyen dicha funcionalidad.

Por último, encontramos el MF52, un termistor NTC de 10kΩ que, además de ser sencillo de utilizar y tener un buen rango de temperaturas, se puede alimentar a diferentes voltajes, dando como salida máxima el valor del voltaje de alimentación. Por otra parte, es el más barato y su precisión depende únicamente de la tolerancia de la

resistencia, en este caso de $\pm 1\%$, y de la calibración que se haga mediante software, aplicando aproximaciones matemáticas.

Todos estos sensores tienen entre sus aplicaciones la medida de temperatura en sistemas de acondicionamiento del aire, termómetros y otros instrumentos de control de temperatura, motivo por el cual se han propuesto para esta aplicación.

6.2.1.2. *Sensor de intensidad lumínica.*

El sensor de intensidad lumínica se utilizará para medir la luz en el exterior con el objetivo de poder aprovechar al máximo las horas de luz en invierno y evitar que la casa se caliente demasiado en verano cuando se encuentre en modo automático. De esta forma necesitaremos medir la luz ambiental, teniendo en cuenta los niveles de luz diurna y el reloj de tiempo real integrado en la plataforma. Para ello podemos utilizar los valores de intensidad de la luz diurna en luxes, por lo que buscaremos un sensor que pueda medir entre 1 y algo más de 30000 o 40000 luxes, algo que nos da una garantía de poder cortar la iluminación para valores más altos en verano y más bajos en invierno. Como con el resto de los sensores, buscamos idealmente que se pueda alimentar a 3,3 V para garantizar que se pueda alimentar desde la placa microcontroladora.

Como referencia, podemos considerar varias condiciones, destacando que, a pleno sol, en verano y a mediodía, la intensidad superaría los 120000 luxes, mientras que por la mañana oscilaría entre 10000 y 100000 en la misma situación. A la sombra de un árbol sería de entre 6000 y 50000 luxes, mientras que la intensidad que entra en la casa en verano y con sol rondaría los 1000 o 1500 [23].

Podríamos excluir todos los rangos por encima de los 50000-60000 luxes ya que a partir de dichos valores la luz ya resulta demasiado intensa y consideraríamos puntos de corte críticos aquellos por debajo de los 1000 luxes que suponen el valor mínimo de luz en interior en verano, además del valor de iluminación de trabajo mínimo en el caso más restrictivo según la normativa española [24].

Seleccionaremos el sensor de temperatura por medio de la siguiente tabla comparativa:

Tabla 3 Tabla comparativa de sensores de intensidad lumínica.

	MAX44009 Módulo GY-49	BH1750FVI Módulo GY-302	Light Dependant Resistor (LDR GL55)
Rango de medida (Luxes)	0,045 – 188000	1 – 65535	1 – 1000
Alimentación	1,7 V – 3,6 V	3 V – 5 V	---
Salida	Digital I2C	Digital I2C	Analógica
Precio	3,04 €/ud (15,2€/5ud)	2,5 €/ud (12,5€/5ud)	0,18 €/ud (9€/50ud)

Descartamos el sensor LDR porque el rango no es suficiente para nuestra aplicación, aunque podría valernos para límites en interiores ya que por encima de los 1000 luxes en interiores la superficie se encuentra muy iluminada.

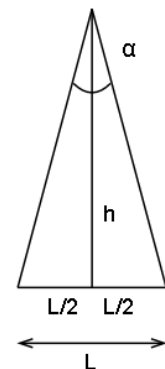
Aunque el módulo GY-49 supone una gran mejora con respecto al GY-302, para nuestra aplicación es excesivo, por lo que consideraríamos que con un rango de hasta 65535 luxes sería más que suficiente para los valores simples a los que se pretende limitar la intensidad lumínica que entra en casa.

6.2.1.3. Sensor de presencia.

En el caso del sensor de presencia a elegir, necesitaríamos que detecte que hay una persona en la entrada, a poder ser evitando que se active con elementos diferentes como la cortina de la entrada o algún otro objeto que pueda pasar por delante del sensor.

Teniendo en cuenta que la altura de la entrada es de 2,70 metros y la anchura es de 1,60 metros, necesitaremos que la detección, si se coloca en el centro de la entrada a la máxima altura, sea de al menos 3 metros.

Suponiendo el ángulo α como el ángulo de detección que necesitaríamos para detectar todo el ancho de la entrada, L como el ancho de la entrada, 1,60 metros, y h como la altura a la que colocaríamos el detector, los 2,70 metros de altura que tiene la entrada, podemos calcular el ángulo que necesitaríamos con la tangente de los dos triángulos que se forman con la altura.



De esta forma tendríamos:

$$\tan\left(\frac{\alpha}{2}\right) = \frac{L/2}{h} \rightarrow \alpha = 2 \cdot \operatorname{atan}\left(\frac{L}{2 \cdot h}\right)$$

$$\alpha = 2 \cdot \operatorname{atan}\left(\frac{160}{2 \cdot 270}\right) = 33,009 \rightarrow \alpha \approx 33^\circ$$

Por ello, nuestro sensor deberá tener un ángulo de detección mayor a 33° para poder detectar en todo el espacio de la entrada.

De la misma forma que con los sensores anteriores, buscamos que la alimentación sea inferior a 5 V y, preferiblemente, que pueda funcionar a 3,3 V. Además, la salida que buscamos es digital, es decir, que nos dé un valor '0' o 'LOW' y un valor '1' o 'HIGH', que, en caso de una salida analógica, sería un valor de 0 V para bajo y un valor de entre 3 V y 5 V para alto.

Hay múltiples tipos de sensores que se pueden utilizar como sensores de presencia, como ya se explicaba en el marco teórico, como pueden ser sensores de ultrasonidos, infrarrojos, cámaras, por ejemplo, térmicas...

Los principales factores que tendremos en cuenta en este apartado serán, además de los ya mencionados que son necesarios, el precio, la alimentación, el tipo de salida y qué elementos detecta (según el calor, según la forma...). Por otra parte, podría ser importante según la aplicación el tiempo de respuesta del sensor, pero en nuestro caso no resulta una característica determinante.

Una vez repasadas las características que necesitamos, realizaremos una tabla comparativa con las diferentes opciones de las que disponemos:

Tabla 4 Tabla comparativa de sensores de presencia

	PIR (KS052)	Sensor de Ultrasonidos (HC-SR04)	PIR (HC-SR501)	Sensor IR activo (IR FC-51)
Distancia de detección	7 m	2 cm – 4 m	7 m	2 cm – 30 cm
Ángulo de detección	100°	15°	<120°	35°
Qué detecta	Luz IR emitida por cuerpos	Distancia que recorre la onda hasta encontrar un obstáculo	Luz IR emitida por cuerpos Ajustable	Reflexión de la luz IR emitida por el sensor
Alimentación	3,3 – 5 V	5 V	5 – 20 V	3,3 – 5 V
Salida	HIGH 3 V LOW 0 V	Pin digital Echo cuando recibe la onda de vuelta	HIGH 3.3 V LOW 0 V	Digital
Precio	3 €/ud (9€/3ud)	1,8 €/ud (9€/5ud)	2,3 €/ud (7€/3ud)	1,95 €/ud

Dadas las características de detección de los sensores, sólo podemos dar por válidos los sensores PIR, que además son comúnmente utilizados como sensores de presencia para encender luces en baños públicos, patios, jardines, etc.

Dentro de estos, a pesar de la diferencia de precio, elegiremos el KS052 por dos motivos principalmente, el primero es la sencillez del modelo, contando sólo con el propio sensor y evitando que se puedan sufrir desajustes u otro tipo de errores debidos a la electrónica, como podrían darse en un módulo HC-SR501, y el segundo, la alimentación a 3,3 V, pudiendo alimentarse directamente desde cualquier controlador con una salida de 3,3 V.

6.2.2. Actuadores

Los actuadores serán principalmente los motores de las persianas que nos permitan automatizarlas y los relés que permitan automatizar el resto de los elementos.

6.2.2.1. Motores de persianas

Lo primero a tener en cuenta será el tipo de persianas que existen en la vivienda, en la casa a automatizar hay persianas tipo ventana y persianas tipo puerta, siendo cada tipo de diferentes tamaños y todas ellas de aluminio.

Las persianas tipo ventana serán 5 de 1,3 m de ancho y 1,1 m de alto, considerando 1.5 m por el exceso de persiana que se queda enrollado en el tambor, y las ventanas tipo puerta serán 3 de 1,3 m de ancho y 1,8 m de alto, considerando 2,2 m por lo ya mencionado. Dimensionaremos el motor utilizando el caso más exigente, es decir, con las persianas tipo puerta ya que tienen un mayor peso, porque conviene tanto a la hora de tener recambios si fallaran, como para poder tener más fácilmente ofertas al pedir más unidades de una misma referencia.

A la hora de calcular el motor a utilizar, lo primero es saber cuál es el peso de la persiana y el diámetro del eje octogonal al que se ancla la persiana. La media de peso por metro cuadrado de las persianas estándar de aluminio es de 4 Kg/m², por lo que podremos calcular el peso de la persiana.

$$\text{Peso Persiana} = \text{Peso Aluminio} \cdot \text{Superficie Persiana} = \frac{4\text{Kg}}{\text{m}^2} \cdot 1,3\text{m} \cdot 2,2\text{m} = 11,44\text{Kg} \approx 12\text{Kg}$$

En este caso tendríamos que usar un motor que soportara más de 12 Kg, teniendo en cuenta que los motores estándar suelen ser de más de 10 Nm, lo cual puede soportar sin problemas persianas de hasta 25 Kg, buscaremos el más barato dentro de esos 10 Nm, ya que los de pares menores, por ser más específicos, no salen tan económicos.

En este rango concreto, encontraremos los motores más baratos por entre 30 € y 40 € cada unidad.

6.2.2.2. Relés

A la hora de seleccionar los relés para este proyecto hay que tener en cuenta la salida, que tiene que ser apta para la red doméstica de 220 V, y la entrada, cuya alimentación debería ser de 5 V, para alimentarlo junto a la placa, o de 3.3 V si lo queremos activar desde el propio microcontrolador. Además, el pin de activación debe funcionar a 3.3 V para asegurar la conmutación con las salidas digitales de cualquier placa que utilicemos.

Por ello, podemos considerar varios tipos de relés, siendo interesantes los relés de estado sólido, ya que se pueden alimentar a 3 V, y, sobre todo, los módulos con relés alimentados a 5 V, pero con una entrada de señal que se puede activar a 3.3 V, ya que son más baratos que los relés de estado sólido y nos permiten realizar la misma función.

El otro factor a tener en cuenta será el requerimiento de amperaje que tenga esta entrada del relé, asegurando que la salida digital de la placa microcontroladora, que suele estar entre 10 y 20 mA, sea capaz de alimentarlo. Por este motivo, existe una limitación en el número de relés que una misma placa microcontroladora puede excitar, siendo una cuestión adicional a tener en cuenta a la hora de implementar el sistema definitivo. Esto apoya a su vez la postura de mantener un sistema descentralizado, ya que se podrá controlar un solo elemento, es decir uno o dos relés, por cada microcontrolador.

6.2.3. Placa microcontroladora

Dentro de las placas microcontroladoras tendremos que considerar únicamente aquellas que dispongan de conexión a internet, preferiblemente inalámbrica (Wi-Fi), además, dado que la programación que se va a desarrollar en la placa es mínima, no se precisa una alta capacidad de procesamiento, motivo por el cual priorizaremos un precio bajo a unas grandes prestaciones.

Por otra parte, dado que el sistema que queremos implementar es descentralizado, nos interesaría el tener varios elementos iguales que se comuniquen con la plataforma, controlando cada una de las placas una o dos de las funciones de cada habitación, motivo adicional para preocuparnos por el precio.

Antes de plantear las diferentes opciones, descartaremos la Raspberry Pi porque, para la aplicación que se va a utilizar esta placa, no se necesita un ordenador de placa simple, sino que resulta más adecuada una placa microcontroladora con

entradas/salidas de propósito general y tiene la capacidad suficiente para la programación que se va a desarrollar.

Veremos las diferentes opciones planteadas en una tabla comparativa:

Tabla 5 Tabla comparativa de placas microcontroladoras

	STM32 IoT B-L475E-IOT01A	Arduino Nano 33 IoT	NodeMCU ESP8266	NodeMCU ESP32
Comunicación	Wi-Fi Bluetooth NFC	Wi-Fi Bluetooth	Wi-Fi	Wi-Fi Bluetooth BLE
Nº Entradas	82 GPIO (5 ADC)	23 GPIO (8 ADC)	17 GPIO (1 ADC)	36 GPIO (16 ADC)
System-On-Chip (SoC)	STM32L4 Arm Cortex-M4	Arm Cortex-M0+ SAMD21 ESP32 module	ESP8266	ESP32
Alimentación (Vin)	5 V	3,3 V	5 V 20 V max	5 V
Alimentación de periféricos (Vout)	2 pin 5 V 3 pin 3,3 V	1 pin 3,3 V	3 pines 3,3 V	1 pin 3,3 V
Precio	49,43 €/ud	30,2 €/ud 18 €/ud	4,5 €/ud (22.5€/5ud)	7,2 €/ud (36€/5ud)

Escogeremos la placa más barata que cumpla con las funciones mínimas necesarias, es decir, tener al menos una entrada analógica para algunos de los sensores utilizados y comunicación Wi-Fi, además de algún pin de alimentación de 3,3 V.

Por ello, la mejor opción, dado que no necesitamos gran capacidad de procesamiento, es el NodeMCU con el ESP8266.

6.2.4. Plataforma IoT

La plataforma será el elemento más importante en programación y por ello, será primordial el lenguaje de programación que se utilice.

Partiremos de la base de que necesitaremos una plataforma gratuita e, idealmente, de código abierto, ya que tendrá un mayor desarrollo por parte de la comunidad, existiendo librerías creadas por ésta para diferentes funciones que nos puedan ayudar en el programa.

Otro elemento que buscamos es la existencia de un dashboard personalizable en la propia plataforma que nos permita visualizar los diferentes elementos que estemos programando y gestionando en la plataforma.

Tabla 6 Tabla comparativa de plataformas IoT.

	ThingSpeak	OpenHAB	IFTTT	Node-RED
Programación	Realización de gráficas con datos y visualización	Script por reglas Java	Por tareas Limitado	Visual (nodos) JSON
Código abierto	No, pertenece a MathWorks	Sí	No	Sí
Local/Online	Online	Local	Online	Local
Dashboard	Visualizaciones de MATLAB	HABPanel	Simple	Sí, totalmente personalizable
Broker integrado	Sí, propio	Moquette integrable	Sí, propio	Sí, mosca
Precio	Gratis limitado (3 suscripciones MQTT)	Gratis	Gratis	Gratis

Después de ver las diferentes opciones, descartamos de primeras ThingSpeak, ya que la versión gratuita es muy limitada, además de ser aplicado principalmente a gráficas de datos, algo que no se ajusta a lo que necesitamos.

Por otra parte, IFTTT está limitado a funciones muy simples, normalmente asociadas a redes sociales, además de no ser de código abierto y tener un dashboard muy simple.

Por último, las opciones serían OpenHAB, dedicado principalmente a automatizaciones de casas, y Node-RED, que es de uso más general. Entre estas elegiremos Node-RED ya que su programación es más intuitiva y visual, siendo el entorno de programación más agradable en comparación al de OpenHAB, que se desarrolla por completo a través de scripts y reglas.

6.3. PRUEBA INICIAL DE COMUNICACIÓN

Antes de proceder a hacer el montaje final con los elementos elegidos, se realiza una primera toma de contacto con la comunicación MQTT con un Arduino UNO conectado a través de un programa preparado para ello en LabVIEW.

El objetivo de esta prueba será realizar la simulación de un sistema de calefacción monitorizado a través de Arduino, que a su vez se comunicará por medio de LabVIEW con el broker. A este broker se conecta a su vez un Smartphone Android y la plataforma Node-RED que permiten la visualización y control del sistema. Además del sistema de calefacción, se harán otro tipo de pruebas generales con el dashboard y la comunicación.

En esta prueba se utilizan los siguientes elementos.

- Broker: CloudMQTT
- Plataforma: Node-RED para Windows.
- Dispositivos: Arduino UNO (LabVIEW) y Android.

6.3.1. Configuración del Broker

En este caso, utilizaremos un broker online común, ya que nos sirve para hacer algunas pruebas sin necesidad de instalar ni alojar de forma local el servicio. En concreto, se utiliza CloudMQTT, un broker que nos permite tener un servidor MQTT protegido por contraseña de forma gratuita para conectar hasta 5 dispositivos.

A continuación, se muestran los detalles de la conexión, con los que luego configuraremos el programa de LabVIEW y la plataforma para que la comunicación funcione correctamente.

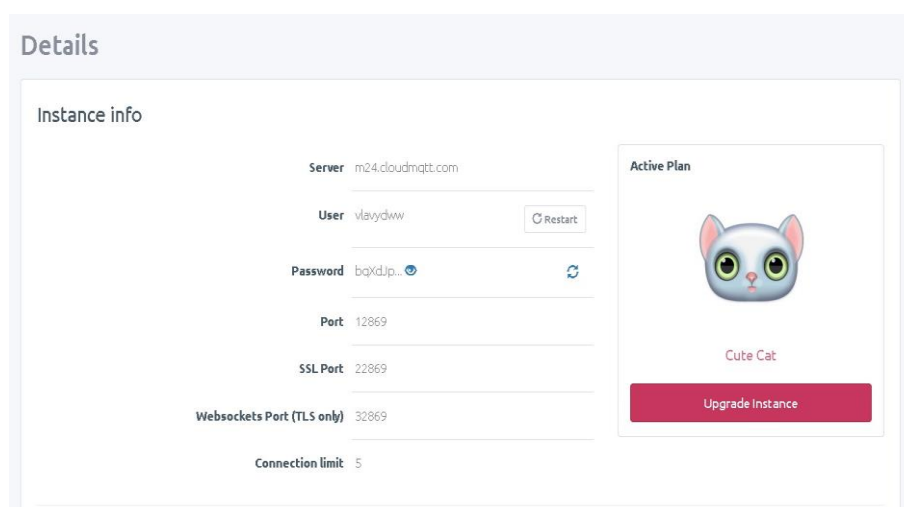


Ilustración 18 Información del broker CloudMQTT.

Lo más importante que encontraremos aquí será la pestaña llamada Websocket UI, donde podremos ver todos los mensajes que se publican con su topic en tiempo real, así como publicar mensajes en un topic específico.

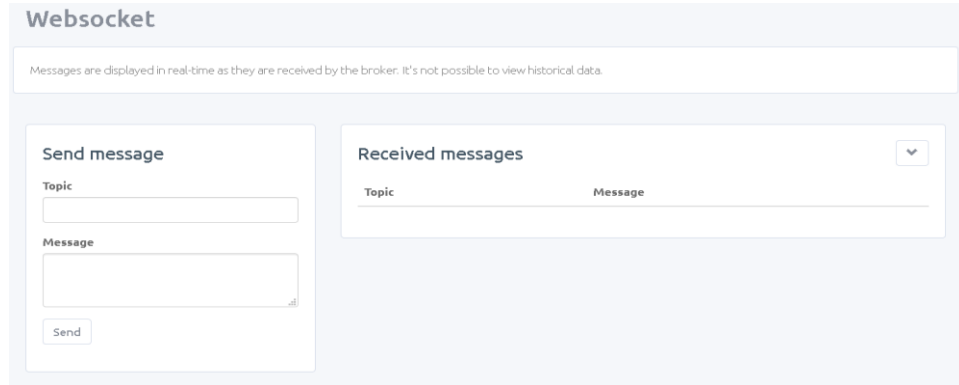


Ilustración 19 Visualización de los mensajes publicados en el broker CloudMQTT.

6.3.2. Preparación de la plataforma

La plataforma Node-RED se instala de forma local como parte de node.js en el dispositivo, iniciando el servicio desde el sistema operativo, ya sea Windows, Linux o Raspbian.

Para la ventana de Windows observamos la siguiente interfaz:

```

Selecionar node-red
C:\Users\albac>node-red
27 Jun 20:24:11 - [info]

Welcome to Node-RED
=====

27 Jun 20:24:11 - [info] Node-RED version: v0.20.5
27 Jun 20:24:11 - [info] Node.js version: v10.15.3
27 Jun 20:24:11 - [info] Windows_NT 10.0.17134 x64 LE
27 Jun 20:24:17 - [info] Loading palette nodes
27 Jun 20:24:27 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
27 Jun 20:24:28 - [info] Dashboard version 2.14.0 started at /ui
27 Jun 20:24:29 - [info] Settings file : \Users\albac\.node-red\settings.js
27 Jun 20:24:29 - [info] Context store : 'default' [module=memory]
27 Jun 20:24:29 - [info] User directory : \Users\albac\.node-red
27 Jun 20:24:29 - [warn] Projects disabled: editorTheme.projects.enabled=false
27 Jun 20:24:29 - [info] Flows file : \Users\albac\.node-red\flows_LAPTOP-R1KQ6TEU.json
27 Jun 20:24:30 - [info] Server now running at http://127.0.0.1:1880/
27 Jun 20:24:30 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

27 Jun 20:24:30 - [info] Starting flows
27 Jun 20:24:30 - [info] Started flows
27 Jun 20:24:30 - [info] [mqtt-broker:TFG prueba Alba] Connected to broker: mqtt://m24.cloudmqtt.com:12869

```

Ilustración 20 Inicialización de Node-RED de forma local en Windows.

Una vez iniciado el servicio, y mientras esta ventana se mantenga abierta, podremos acceder al servidor en el que se mantiene la plataforma en la dirección que aparece, en este caso <http://127.0.0.1:1880/>

Los mensajes del log de la plataforma se pueden mostrar tanto en esta ventana de comandos como en el debugger de la visualización web del servicio.

La programación en Node-RED es una combinación de programación por bloques y programación en JSON o JavaScript Object Notation. Aparecen ciertos bloques preinstalados y se pueden añadir diferentes nodos para funciones nuevas. En nuestro caso añadiremos una serie de nodos que permiten generar un dashboard a partir de la programación de Node-RED.

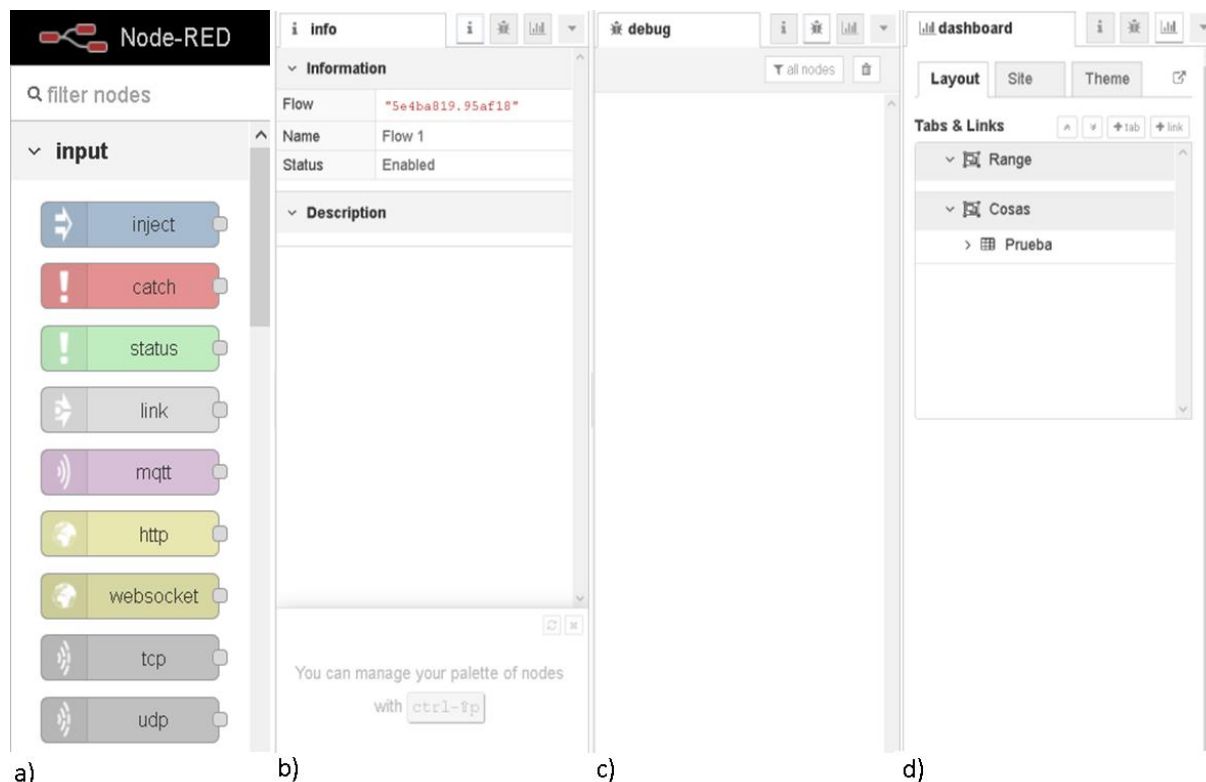


Ilustración 21 a) Menú de nodos. b) Información del flujo. c) Debugger del flujo. d) Configuración del dashboard global.

En el lateral izquierdo se observa el menú que permite seleccionar diferentes nodos, mientras que en el lado derecho se encuentra la información, el Debugger y la configuración del Dashboard.

La parte principal de Node-RED es el diseño de Flows. En este se van añadiendo los diferentes nodos y se enlazan entre sí, de forma que hay nodos de entrada, de salida y otros que tienen tanto entradas como salidas.

Para que le llegue la información a la plataforma utilizaremos LabVIEW, que leerá la información del Arduino UNO utilizando comunicación VISA de puerto serie y se encargará de publicarla en el servidor.

6.3.3. Programación en Arduino

El programa de Arduino se encarga de publicar valores en el puerto serie para que LabVIEW los vea, no tendremos un sensor de temperatura, sino que el Arduino simula el valor que nos proporcionaría uno, algo que conseguiremos mediante dos métodos.

Por una parte, se utiliza un cable alimentado a 5 voltios, 3,3 voltios y GND como valores de entrada analógicos, viendo los 3 valores, uno máximo, otro mínimo y uno intermedio; y, por otra parte, a través de la programación, enviando valores a través del puerto serie de Arduino, de LabVIEW, del broker o desde el móvil, entre -20º y 60º, los cuales nos permiten comprobar cómo se comporta nuestro sistema en diferentes casos. Esto nos permite verificar el correcto funcionamiento en el dashboard de Node-RED y en el teléfono, además de comprobar la velocidad de la conexión.

6.3.4. Programación en LabVIEW.

Para LabVIEW lo primero que necesitaremos será obtener los paquetes que contienen los SubVIs que nos permiten conectar el programa a un servidor por medio de MQTT, llamado LVMQTT master.

Para el desarrollo de nuestra aplicación combinaremos los dos ejemplos que nos proporciona la herramienta LabVIEW que son; Connect and Publish y Connect and Subscribe, además de los comandos del paquete VISA. Al unir los diferentes elementos antes mencionados se obtiene el siguiente interfaz en LabVIEW, que nos permite publicar directamente la información del Arduino extraída mediante el VISA, además de poder recoger la información de los topics que se requiera, en este caso, nos interesa leer la salida del valor TRUE/FALSE de la calefacción, que activa y desactiva el termostato programado en Node-RED.

Este sería el panel frontal del programa de LabVIEW. Aquí podemos ver los ajustes de publicación, el broker MQTT al que se publica, la información del usuario y el topic en el que se publica. Tenemos varios informes de error, un botón para dejar de publicar y otros cuadros que muestran diferentes parámetros sobre el ID del mensaje, los mensajes publicados y, también importante, el VISA resource name con el que seleccionaremos el puerto COM del que vamos a leer.

Otro de los elementos importantes es el LED de "Calef" que podemos ver en la parte interior derecha, al que le llegan los valores a los que está suscrito el programa, en este caso al topic que publica si la calefacción está encendida o no.

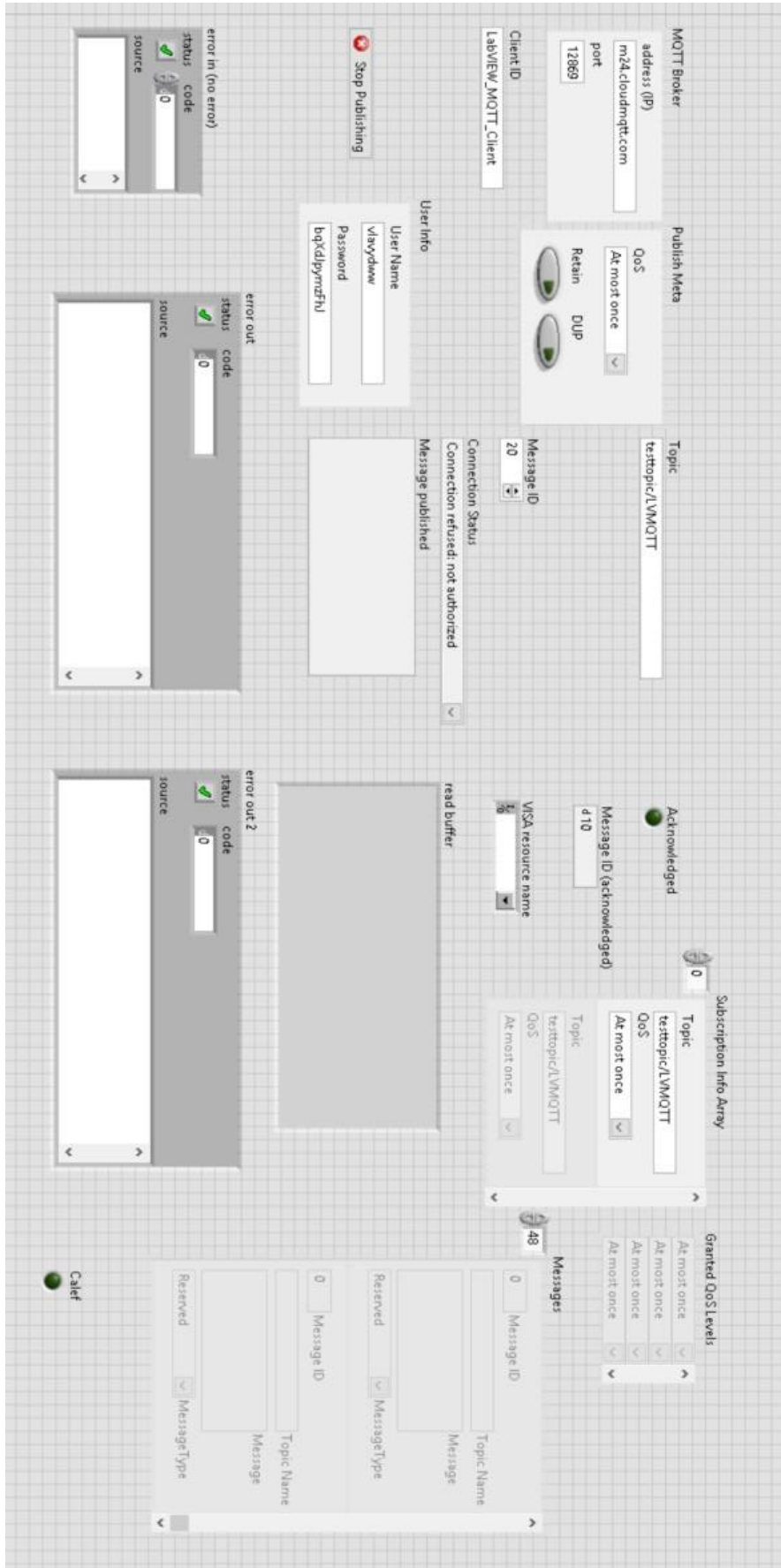


Ilustración 22 Visualización en LabVIEW del programa.

6.3.5. Programación en Node-RED

A la hora de realizar la programación en la plataforma, se realizará un flujo completo en el que, a través de varios bloques y la comunicación entre ellos, se pueda simular un termostato completo, incluyendo la medida de temperatura, el ajuste del valor de corte, así como de la histéresis, y el indicador que muestre si la calefacción se encuentra encendida o apagada.

Entre los bloques utilizados se pueden observar con color lila los que se tratan de conexión MQTT, de recepción, como testterm, y de envío como test/calef. Además, se utilizan los bloques verdes, bloques de debug, para ver la salida de otros bloques en el debugger de la plataforma. Los bloques de color azul pertenecen al dashboard y se corresponden con leds, interruptores, controles numéricos, barras y un indicador semicircular en el caso de "Temperatura". Por último, los bloques de color amarillo son funciones propias de la plataforma para modificar objetos. En este caso, se utilizan los bloques "Change" que permiten dar un valor determinado, normalmente el que reciben de la entrada, a un objeto diferente, para este flujo se utiliza el objeto flow, en el que se alojan variables del flujo activo, bajo las categorías hist, para el valor de histéresis, y term, para el valor del termostato. El flujo completo se organiza para una mejor visualización.

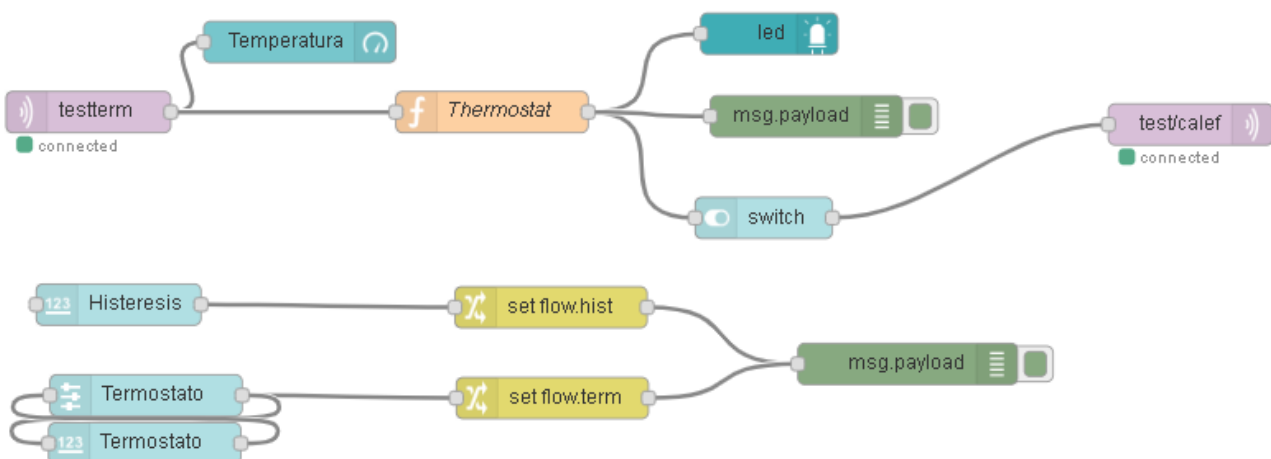


Ilustración 23 Flujo completo de la prueba en Node-RED.

En este caso podemos ver claramente cómo podemos editar el valor del termostato, la temperatura a la que queremos que se encuentre la habitación y el valor de histéresis. A su vez, tenemos el indicador en el que se publica el valor de temperatura de forma continua, en este caso de 25 grados ya que hemos pedido que el Arduino publique ese valor por ser un valor simple para ver como prueba.

Al poner el termostato por encima de esos 25 grados, podemos ver como el LED se enciende de color verde en lugar de rojo, indicando que la temperatura está baja y, por tanto, hay que encender la calefacción.

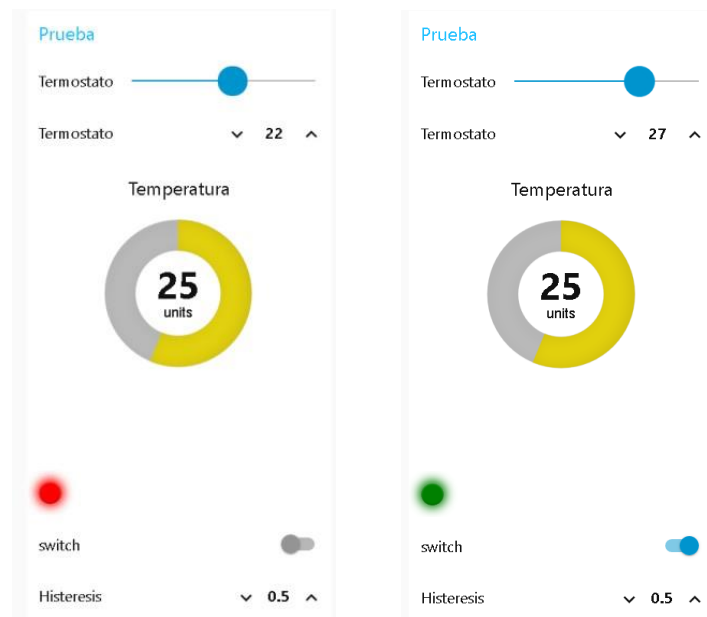


Ilustración 24 Interfaz Node-RED con calefacción apagada (Izquierda) y encendida (derecha)

Una vez ya se ha producido la primera toma de contacto con todas las funciones del sistema a desarrollar, se puede iniciar la implementación del sistema final a través de un prototipo.

6.4. IMPLEMENTACIÓN DEL SISTEMA FINAL

El desarrollo del sistema comienza con la conexión de la electrónica, incluyendo las placas; continúa con el desarrollo software en Arduino IDE y en Node-RED, y culmina en la configuración de una interfaz Android.

6.4.1. Conexión de la electrónica

Los elementos que tenemos que conectar son un termistor como sensor de temperatura, un sensor PIR para la detección de presencia en la entrada y un sensor BH1750 integrado en un módulo GY-302 como sensor de intensidad lumínica en luxes.

De estos sensores, sólo el termistor da una salida analógica, el sensor PIR da una salida digital de 1 bit y el sensor BH1750 tiene salida de bus I2C. Los pines del NodeMCU que utilizaremos serán el A0 para el termistor, el D1 y D2 para el bus I2C y otro de los pines digitales, el D0, para el sensor PIR. De esta forma, quedan disponibles el resto de los pines digitales, D3-D8, para utilizarlos como salidas a diferentes LED, en el caso del prototipo, y a relés en el caso de la futura implementación física. A la hora de conectar los sensores, la alimentación vendrá dada por el NodeMCU, a 3,3 V, así como la tierra, que será común. Debemos tener en cuenta la forma de conectarlos, utilizando resistencias cuando sea necesario. En este caso, se realiza un prototipo formada por dos microcontroladores, Node1, con el sensor BH1750, y Node2, que controlará el termistor y el PIR. Se realiza primero el montaje del Node1, conectando el sensor BH1750 con cuatro cables, rojo para Vcc a 3,3 V, negro para GND, verde para el pin SCL conectado a D1 y amarillo para el pin SDA que se conecta en D2. Además, tendremos un LED amarillo conectado al pin D0 como indicador de la respuesta que se reciba de la plataforma y con una resistencia de 100Ω a tierra.

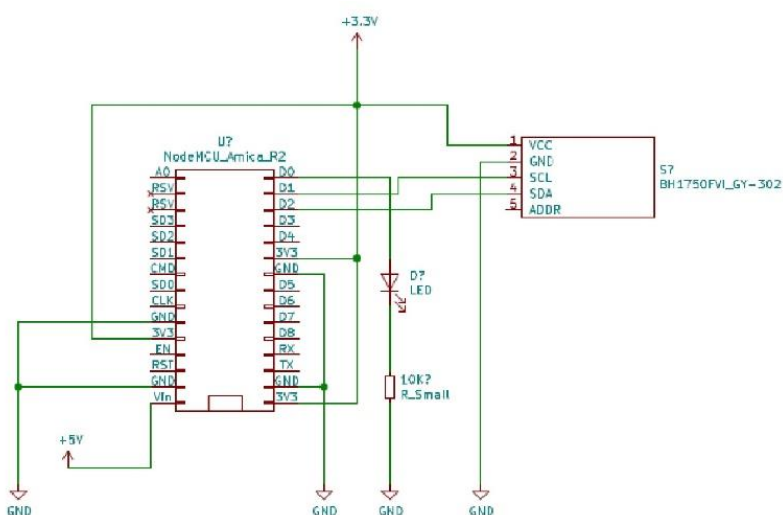
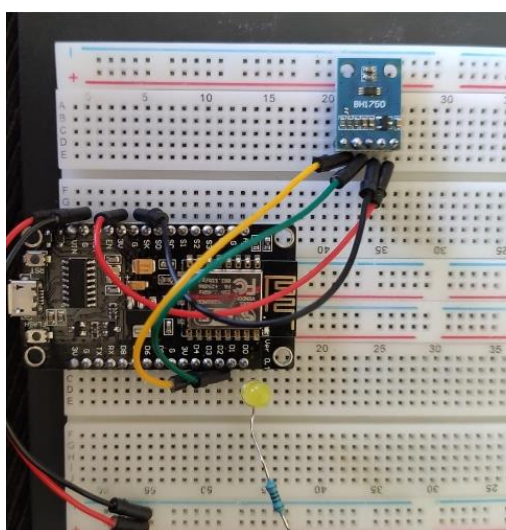


Ilustración 25 Montaje (Izquierda) y esquema eléctrico (Derecha) para el Node1.

El segundo microcontrolador o Node2 se conecta en la misma placa, pero de forma independiente al Node1, siendo el único punto común para ambos la alimentación por parte del cargador universal. Lo primero que se conecta en nuestro Node2 es el termistor, el cual conectaremos entre la tierra del microcontrolador y la entrada analógica A0, añadiendo también una resistencia de 10K Ω , ya que es el valor de calibración del MF52, entre la entrada A0 y la salida de 3,3 V, actuando como un divisor de tensión. Al otro lado del microcontrolador se añade el PIR, alimentado con 3,3 V al conector + (cable rojo) y GND al pin - (cable negro), y conectando la salida al pin digital D8 con un cable naranja. Por último, se añaden dos LED, uno rojo para la salida de la calefacción de la plataforma y uno azul para el PIR, ambos con una resistencia de 100 Ω , al igual que el amarillo anterior. En este caso ocuparán las salidas digitales D1 y D0, respectivamente.

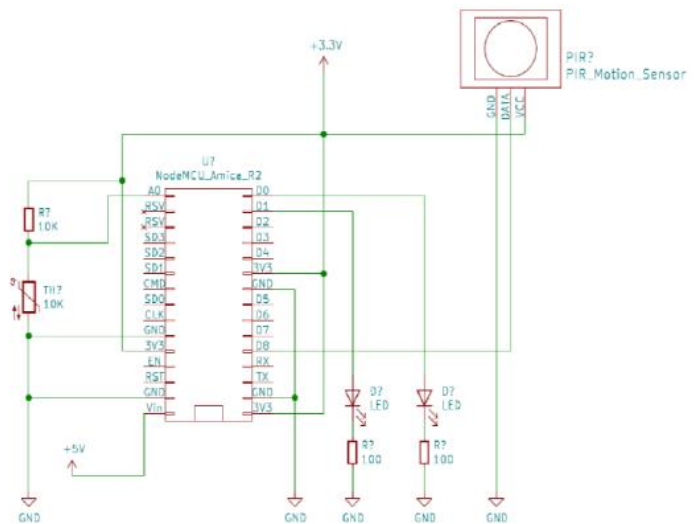
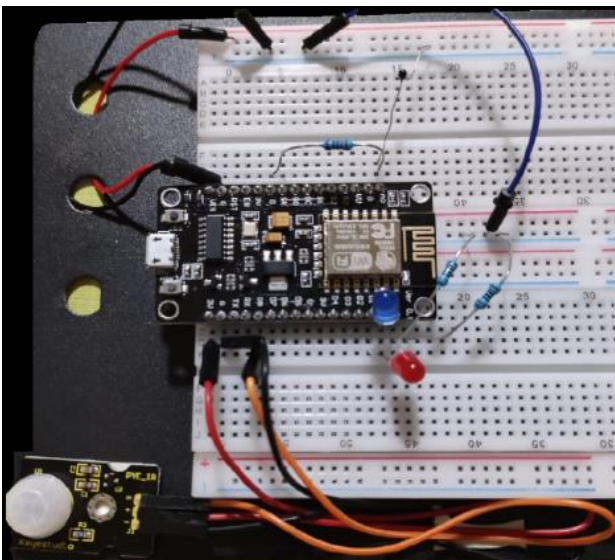


Ilustración 26 Montaje (Izquierda) y esquema eléctrico (Derecha) para el Node2.

6.4.2. Programación Arduino IDE

Lo primero a tener en cuenta en este apartado son las librerías que se necesitan para los diferentes componentes electrónicos, empezando por la librería propia de NodeMCU, además de la librería EspMQTTCient con funciones específicas para la comunicación MQTT. Para el sensor BH1750 también se necesita una librería que nos permita realizar la lectura de este.

Todas estas librerías se pueden instalar directamente en el propio IDE de Arduino o descargar desde el repositorio público de GitHub.

Una vez instaladas las librerías, el código a desarrollar debe incluir la conexión al WiFi y el broker MQTT. Al conectarse, el microcontrolador se suscribirá a los topics

necesarios, en nuestro caso "TempS", "PIRs" y "LuzS", y, siempre que esté conectado, iniciará un bucle de lectura de sensores y publicación de los datos leídos. Al recibir un mensaje en cualquiera de los topics a los que se encuentra suscrito, se iniciarán funciones específicas para cada uno de ellos en las que se ejecutará la lógica que requiera de dichas variables, en nuestro caso, activar y desactivar las diferentes salidas digitales.

Para las diferentes funciones programadas en Arduino, cada una de las cuales puede funcionar de forma independiente en un NodeMCU, existe un diagrama de funcionamiento. A continuación, se puede ver el diagrama del control de persianas, siendo las líneas punteadas comunicación MQTT con la plataforma. Así, el bucle que se realizaría dentro de este NodeMCU consistiría en conectarse a la red WiFi y al broker MQTT y, una vez comprobado que la conexión está bien realizada, se suscribe al topic "Node1/LuzS" y procede a realizar una lectura del sensor. Esa lectura se manda a través del WiFi a la plataforma bajo el topic "Node1/LuzE" y cada vez que se reciba una lectura de "Node1/LuzS", se utiliza dicho valor para dar una respuesta que haga que se suba o se baje la persiana.

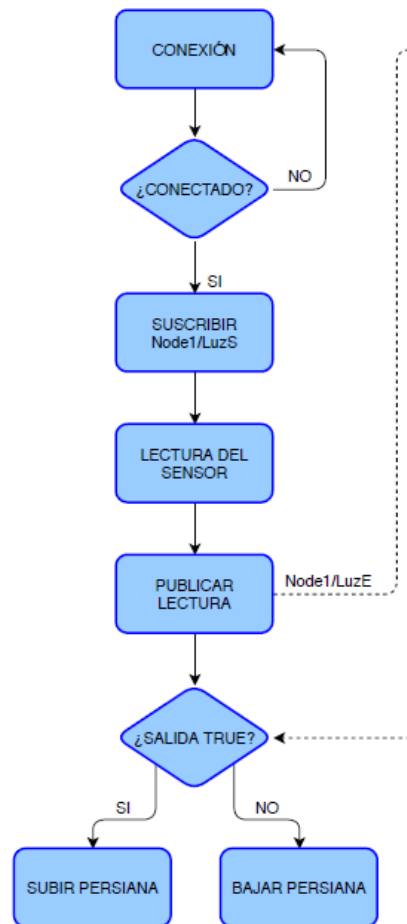


Ilustración 27 Diagrama de funcionamiento para el control de las persianas

De forma similar se desarrollan las funciones de control de temperatura y control de la luz de entrada, ya que el control de las variables se realiza en la plataforma, ocupándose los microcontroladores de únicamente mandar datos de lectura de sensores e interpretar las señales digitales recibidas desde esta.

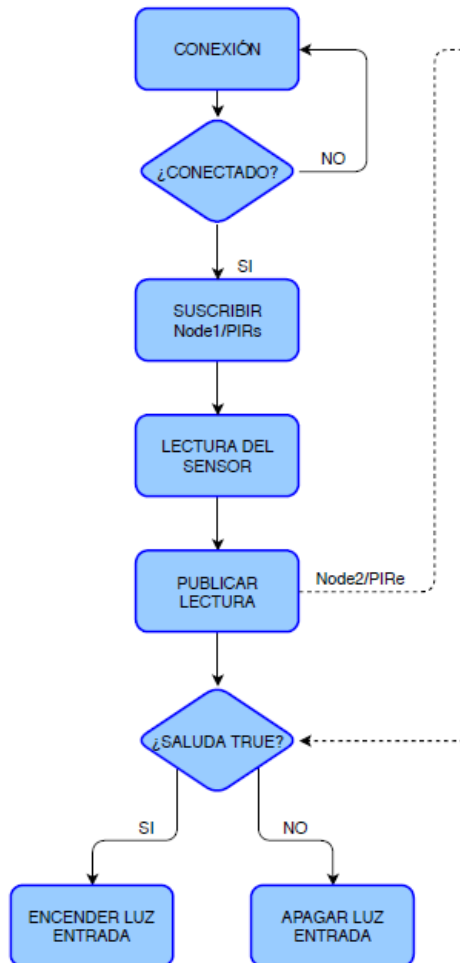


Ilustración 28 Diagrama de funcionamiento del control de presencia.

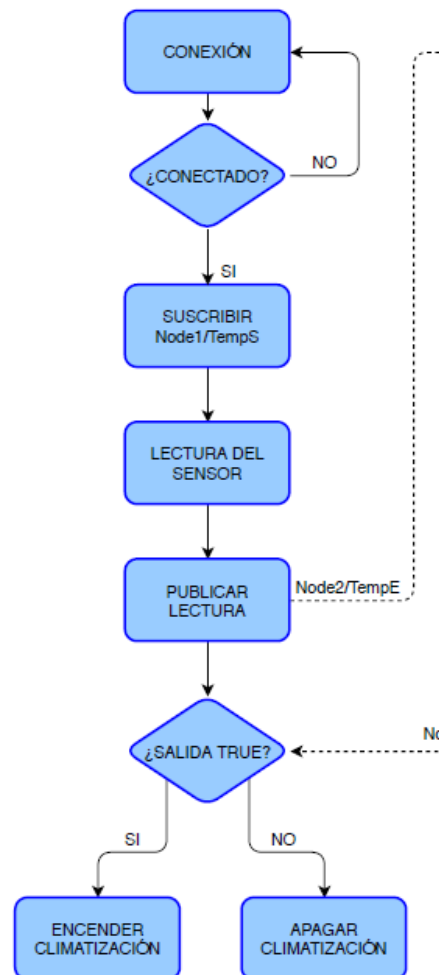


Ilustración 29 Diagrama de funcionamiento del control de temperatura.

Para la correcta calibración del termistor utilizado, con una resistencia nominal de 10 kΩ, dado que no es un sensor lineal, se realiza una aproximación matemática de tercer orden, según el modelo de Steinhart-Hart, a través de las siguientes ecuaciones y consiguiendo un error de aproximación inferior a 0,005 °C [25].

$$T_{sh} = \frac{1}{A + B \cdot \log(R) + C \cdot (\log(R))^3}$$

$$T = T_{sh} - \frac{V^2}{k \cdot R}$$

Donde k es el factor de disipación obtenido del datasheet, V el voltaje de alimentación y R el valor de resistencia nominal del termistor.

Se utilizan valores A, B y C calibrados para una resistencia MF52 de 10 k Ω , obtenidos a partir de los valores de resistencia del termistor para 5 $^{\circ}\text{C}$, 25 $^{\circ}\text{C}$ y 35 $^{\circ}\text{C}$. Para calcularlos, Stanford Research Systems dispone de una calculadora que dibuja la curva T-R y calcula los coeficientes A, B y C necesarios [26].

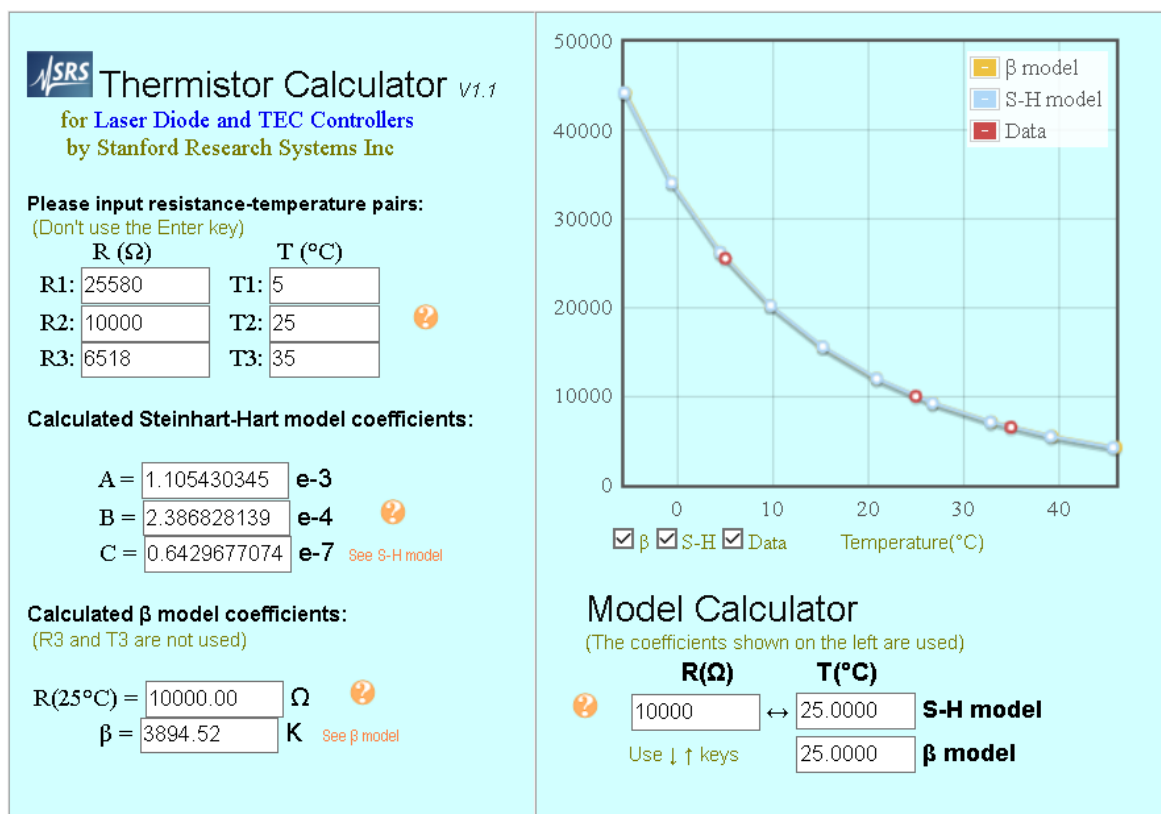


Ilustración 30 Curva y parámetros de calibración para el sensor MF-52.

Todo esto, junto a la librería "math" que permite utilizar logaritmos y la programación necesaria para incluir la ecuación, permite una calibración lo más exacta posible para el sensor utilizado.

Se puede observar el código resultante en los anexos.

6.4.3. Programación Node-RED

Para la programación de Node-RED tendremos que instalar algunos nodos que nos permitan mostrar las diferentes variables en una interfaz propia o dashboard, como se llama en la plataforma. Existen múltiples nodos creados y desarrollados por la comunidad de esta plataforma, algunos de los cuales podremos utilizar para esta aplicación.

El primer nodo que instalaremos será el broker MQTT llamado Mosca, que nos permitirá tener toda la comunicación MQTT integrada dentro de la plataforma. Además, tendremos que añadir nodos que nos permitan mostrar imágenes en nuestro dashboard, así como los elementos del dashboard de los que ya se habla en la primera prueba.

Finalmente se desarrollan un flujo principal y tres subflujos, uno para cada uno de los sensores utilizados, que se pueden observar en los anexos.

En el flujo principal podremos observar los subflujos, el nodo del broker MQTT, los nodos que nos permiten mostrar imágenes y la función de seguridad. Esta función de seguridad se desarrolla de forma que se puedan subir las persianas por la mañana, discriminando entre verano e invierno, y bajar por la noche de forma automática. Además, una vez las persianas se han bajado se iniciaría una secuencia de luces en la casa.

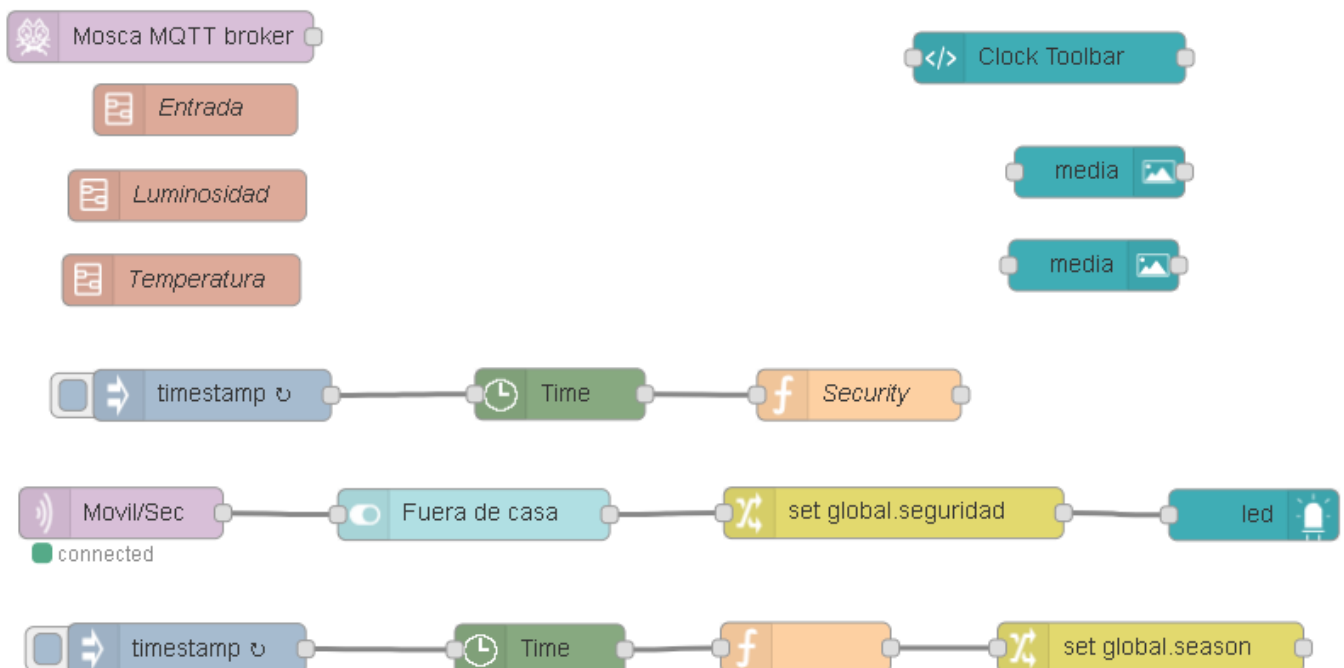


Ilustración 31 Flujo principal de Node-RED.

Una vez entramos en el subflujo "Temperatura", se desarrolla una función que permite seleccionar los valores de histéresis y temperatura del termostato desde la interfaz web. Estos valores serán los que activen la calefacción, en caso de ser el valor del termostato mayor que el medido por el sensor, o la apaguen, en el caso contrario. Además, se cuenta con un modo verano que hace que esto funcione de forma inversa, activando la climatización al ser la temperatura medida mayor que la deseada.

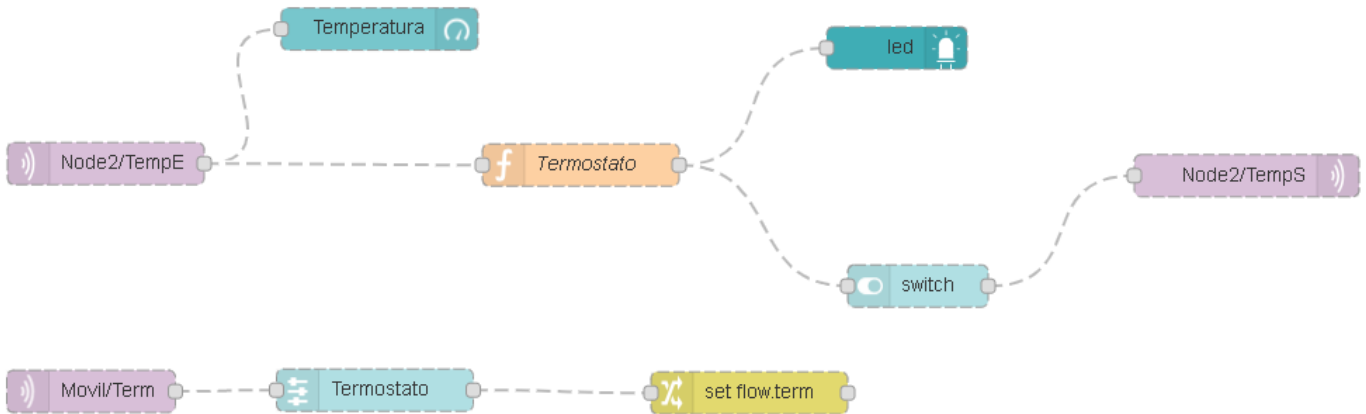


Ilustración 32 Subflujo "Temperatura".

El subflujo "Luminosidad" funciona de forma similar, pero cuenta con un interruptor que permite activar o desactivar el modo automático que, al estar desactivado, permite activar y desactivar la salida, en este caso las persianas, por medio de otro interruptor.

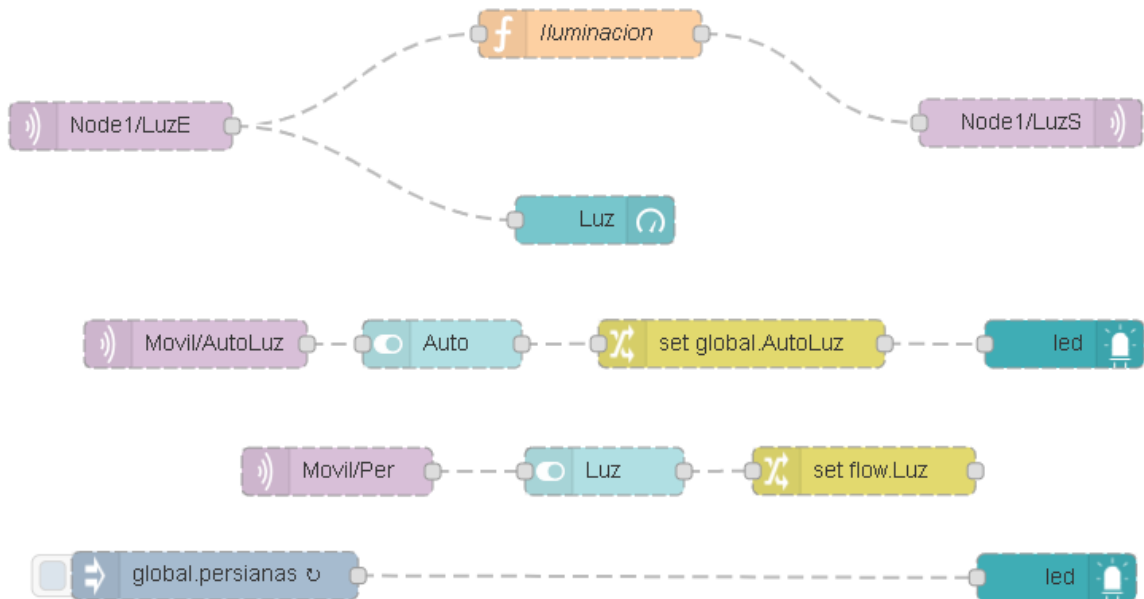


Ilustración 33 Subflujo "Luminosidad".

Por último, el subflujo "Entrada", permite activar o desactivar una salida en función de la lectura del sensor PIR que se situaría en la entrada de la casa. De la misma forma que en el caso de "Luminosidad", se incluye también un interruptor para activar y desactivar el modo automático. Adicionalmente, la salida sólo se activaría a ciertas horas, discriminando de nuevo entre verano e invierno.

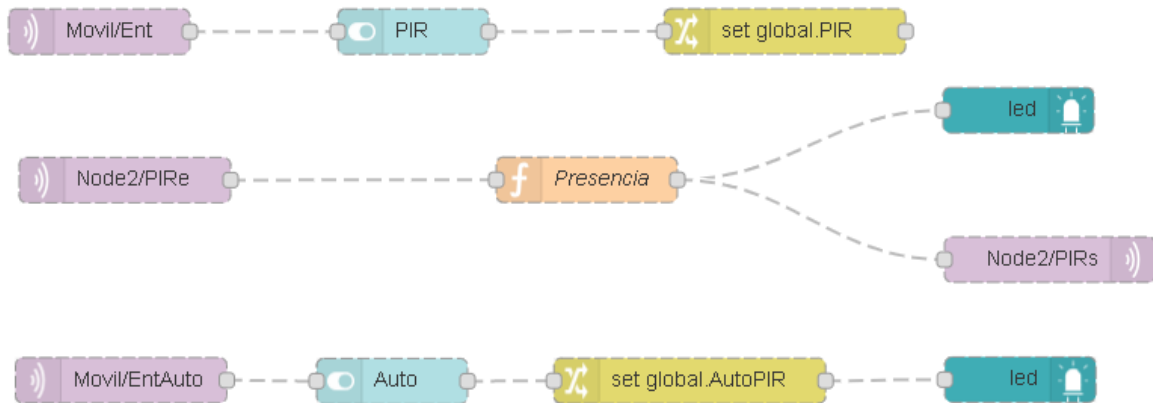


Ilustración 34 Subflujo "Entrada".

6.4.3.1. Dashboard

Todos los elementos que se presentan en los flujos anteriores en color azul forman parte del dashboard o interfaz web generado por la plataforma. Además de realizar las configuraciones adecuadas que permitan mostrar los elementos bajo el nombre y en la posición adecuadas, se configuran individualmente los valores posibles para cada nodo del dashboard.

Con ello, se consigue una interfaz web con cinco secciones, "Temperatura", "Luz", "Seguridad", "Casa" y "Entrada". Los controles del subflujo "Temperatura" se encontrarán en esta sección, en la que podemos ver un medidor de la temperatura de la habitación en grados centígrados, un selector del valor de temperatura del termostato, otro que permite seleccionar el valor de histéresis, un interruptor para el modo verano y un LED indicador del estado de la climatización.

La sección "Luz" cuenta con los controles del subflujo "Iluminación", mostrando el valor de intensidad lumínica en luxes, un indicador del modo automático y otro del estado de las persianas y dos interruptores, uno de modo automático y otro que permite activar y desactivar las persianas en modo manual.

La sección "Seguridad" cuenta con un interruptor para activar esta funcionalidad cuando la casa está vacía, y con un indicador que muestra el estado de la función. En la sección "Casa" se muestran imágenes de los dos pisos de la casa que varían según la hora del día.

Por último, la sección "Entrada" permite el control del sensor PIR y la luz de entrada asociada a este. Para ello hay dos interruptores, uno para el modo automático y otro para activarlo y desactivarlo cuando está activado el modo manual. Además, hay dos indicadores, uno del modo automático y otro del estado de la luz de la entrada.

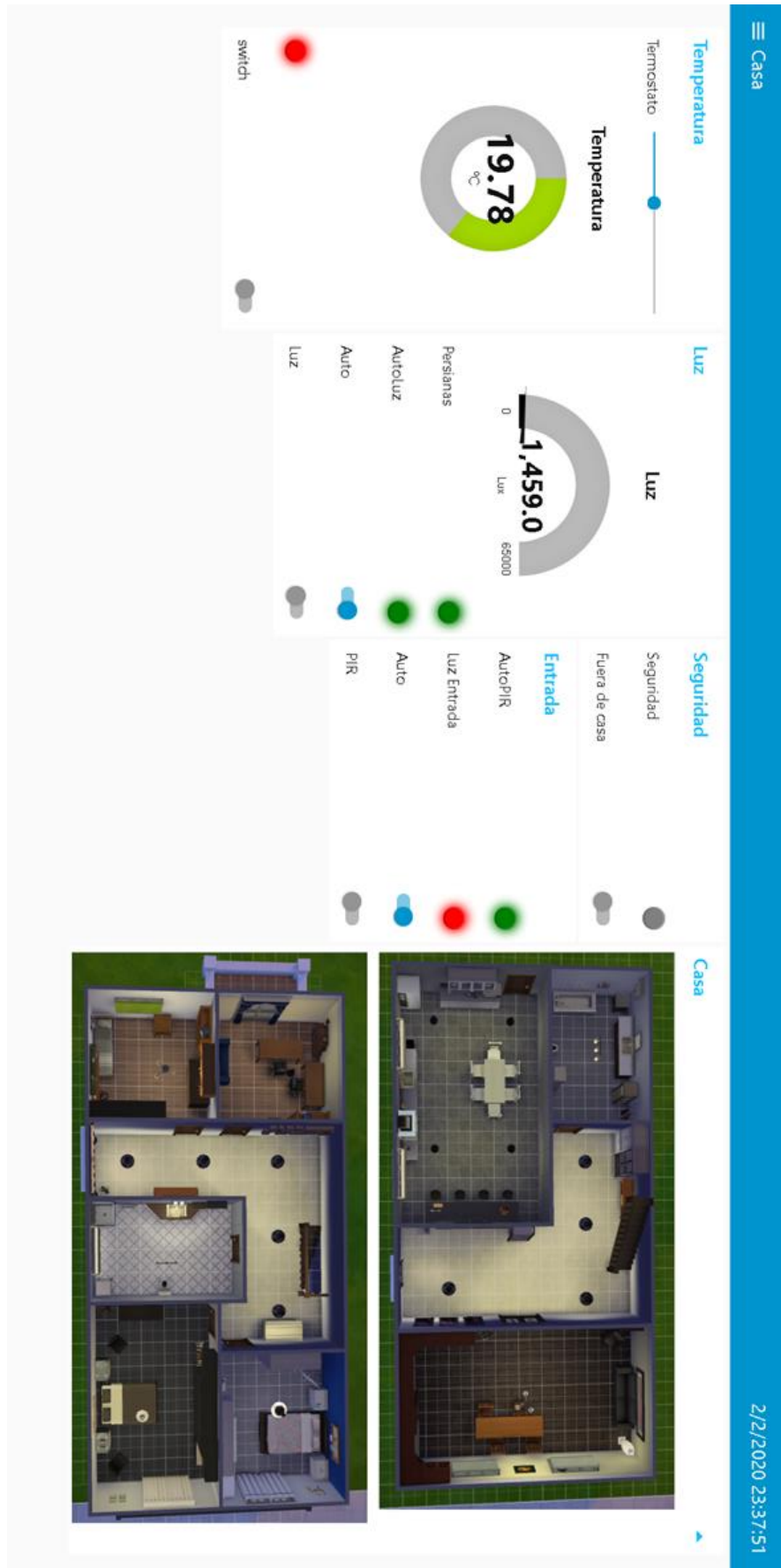


Ilustración 35 Interfaz web Node-RED.

6.4.4. Dashboard Android

A la hora de desarrollar el Dashboard en Android, se utiliza la aplicación mencionada en la prueba, añadiendo nuevos elementos provenientes de los datos de los sensores. Las funciones que tiene esta interfaz son por un lado informativas, mostrando los valores de medida y activación de los diferentes elementos que conforman el sistema, y por otro lado de control, pudiendo interactuar con la personalización existente en la interfaz web. Para realizar las comprobaciones de la lógica desarrollada, se habilita temporalmente también la modificación de algunos valores de entrada a la plataforma.

Se tendrá en cuenta el diagrama de funcionamiento al desarrollar la interfaz móvil, buscando controlar y visualizar los elementos del dashboard ya presente en Node-RED, por lo que en el caso de la función de persianas o iluminación se establecen una visualización de la información del sensor de intensidad lumínica en luxes y dos interruptores o botones, uno para activar el modo automático y otro para interactuar con las persianas en modo manual.

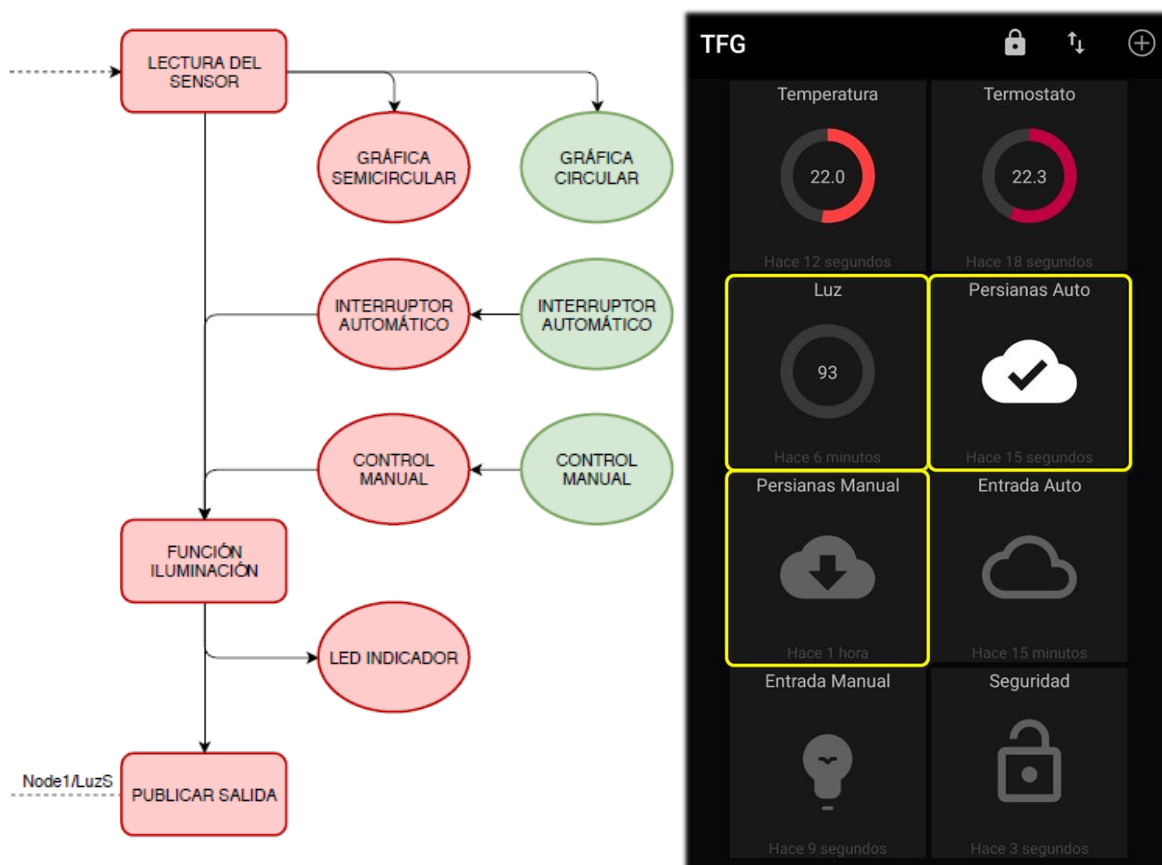


Ilustración 36 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función persianas.

En la función de presencia sólo tendremos dos botones, similares a los del caso anterior, que permitan activar y desactivar el modo automático y activar y desactivar la luz en la entrada al estar en modo manual. Además, podemos ver en la esquina inferior derecha el interruptor para la activación del modo de seguridad.

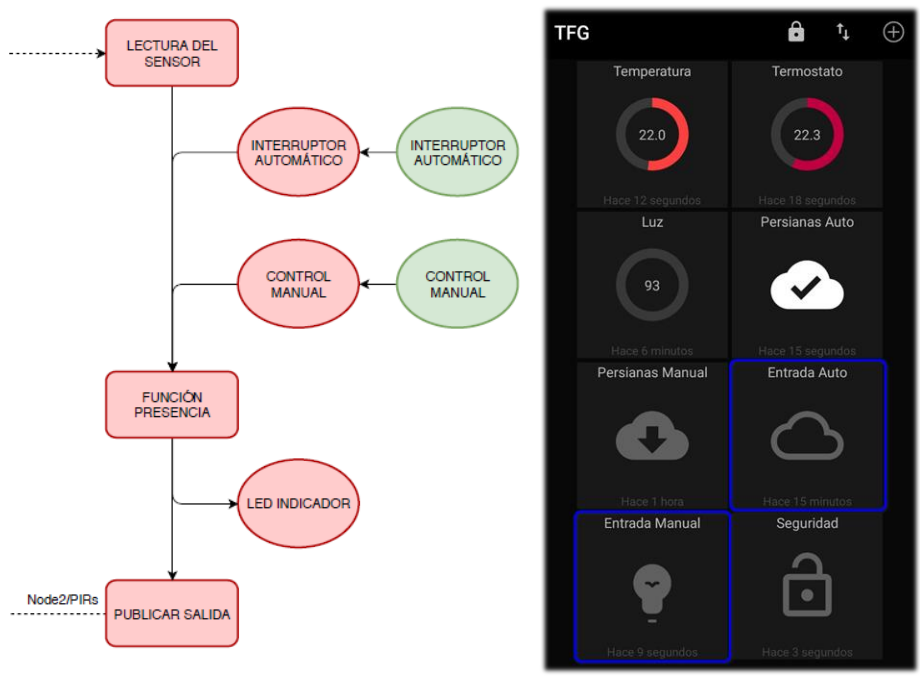


Ilustración 37 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función presencia.

De forma similar, se encuentra la visualización de las variables y el control en el caso de la temperatura, utilizando gráficas circulares.

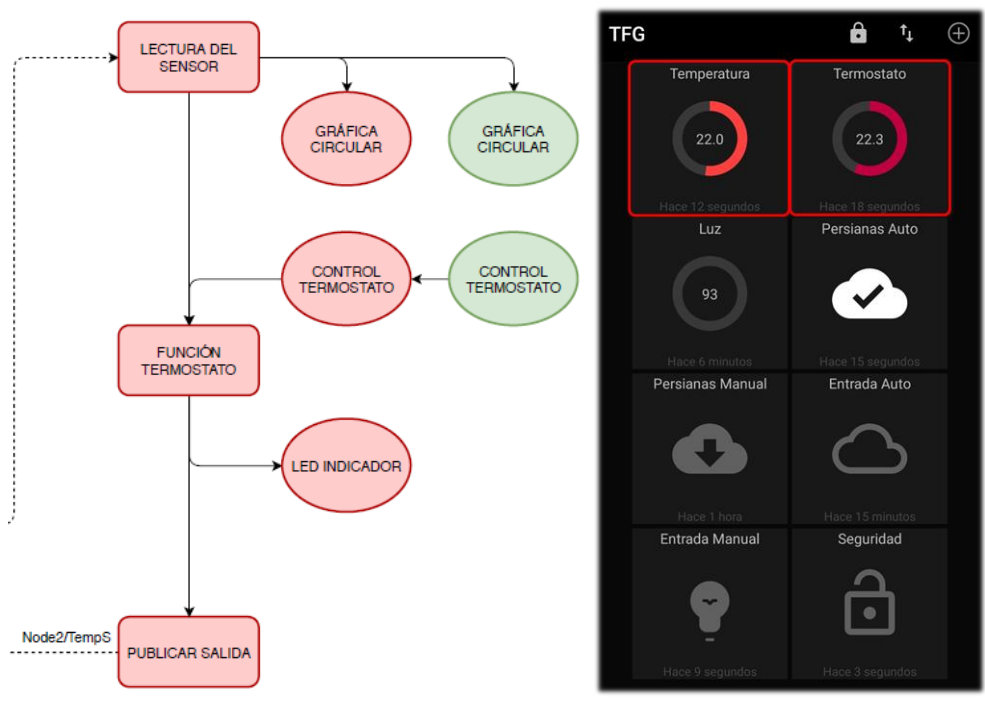


Ilustración 38 Diagrama de funcionamiento (Izquierda) e interfaz móvil (Derecha) para la función temperatura.

7. RESULTADOS

Tras la implementación completa del prototipo, la plataforma y la interfaz móvil, así como la conexión establecida entre ambos, se requiere un testeo del sistema completo, realizando una serie de pruebas y comprobando la respuesta del sistema ante diferentes supuestos.

En primer lugar, se comprueba la función temperatura, desde la conexión del sensor al microcontrolador y el envío de los datos hasta el control a través de las interfaces.

Así pues, se presentan los dos supuestos principales, modificando el valor del termostato desde la interfaz móvil, respuesta de la plataforma al cambio y del prototipo ante la activación y desactivación de la calefacción.

En el caso 1 se coloca el valor del termostato a 16,5 °C desde la interfaz móvil, el led de temperatura se mantiene apagado, ya que la temperatura medida es de 21,68 °C.

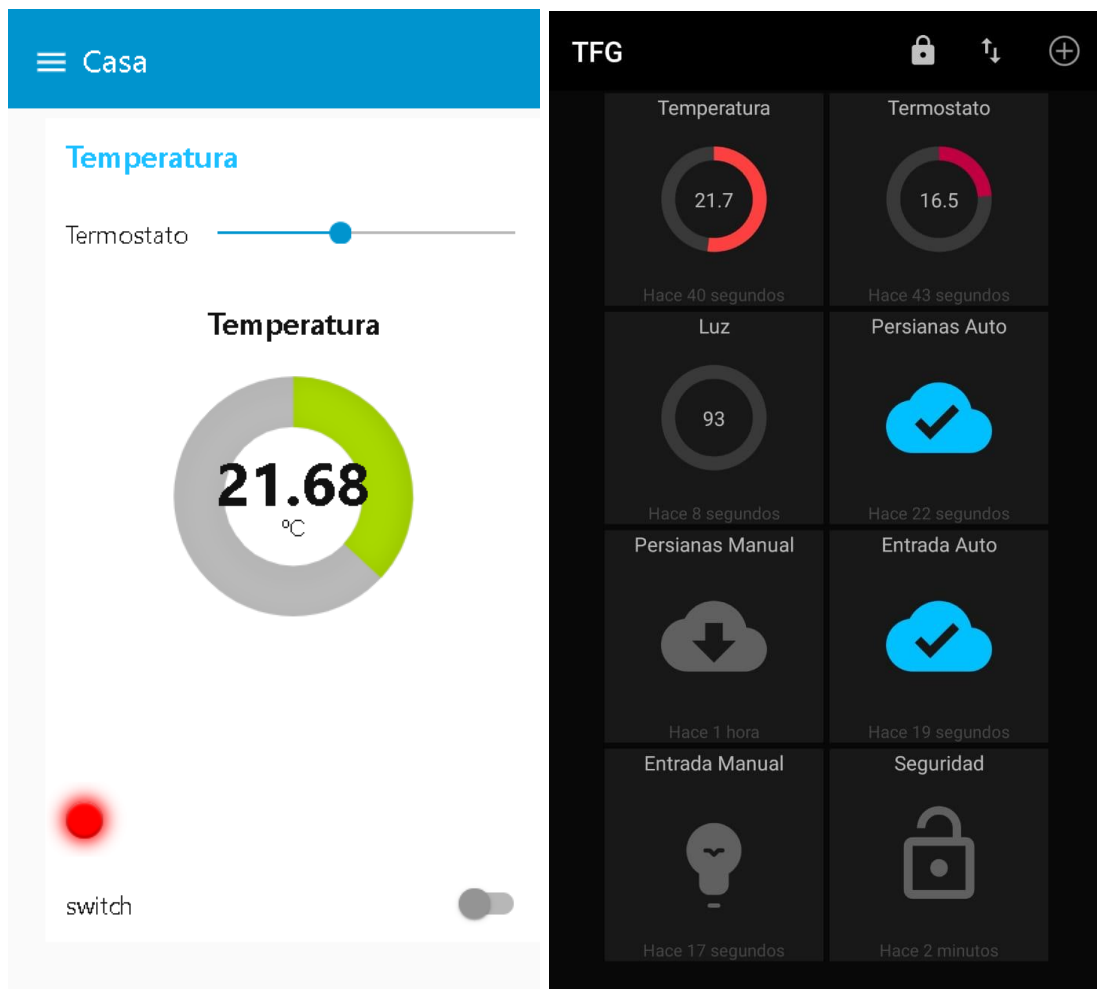


Ilustración 39 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (calefacción apagada).

Resultados

Para el caso 2, se aumenta el valor del termostato a 22,3 °C y, como el valor de histéresis sea de 0,5 °C, se activa cuando se encuentra en 21,68 °C y se mantiene encendida para el valor 22,03 °C mostrado. En el momento en que pasa de 22,7 °C, se apaga.

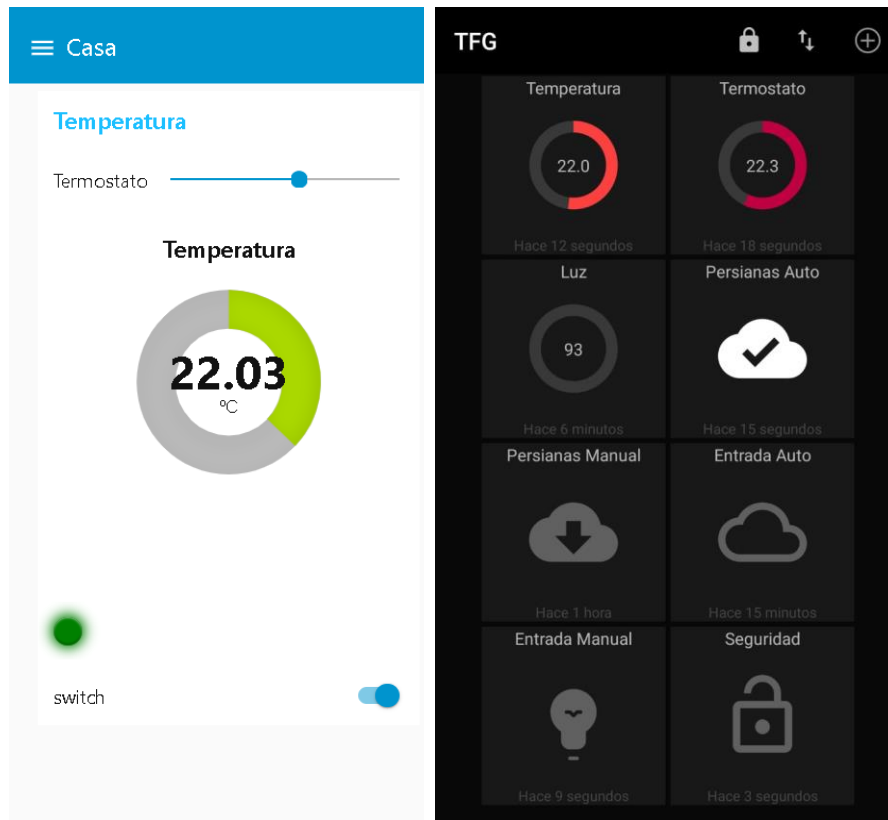


Ilustración 40 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (calefacción encendida).

Para este caso 2 podemos observar también que el led correspondiente a la calefacción (rojo) del prototipo se enciende a la par que el led de la interfaz.

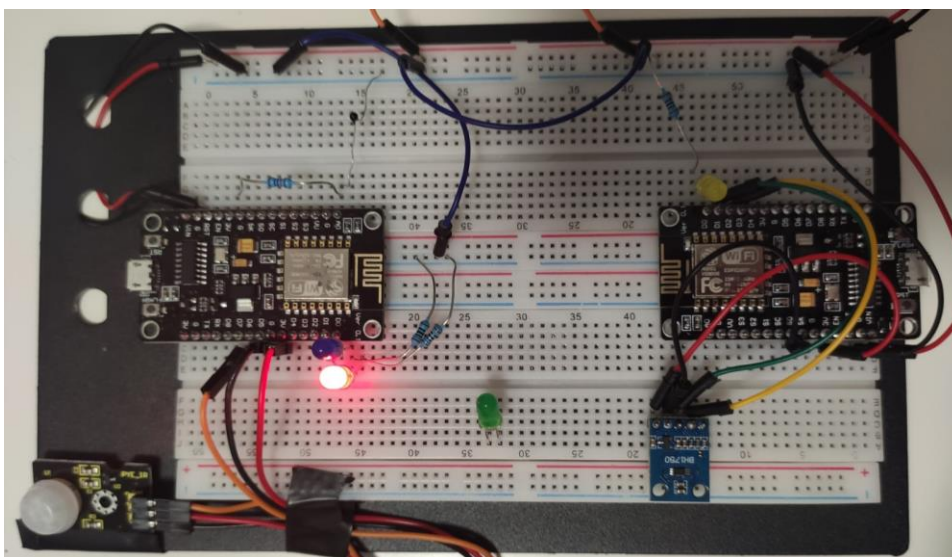


Ilustración 41 Prototipo en el caso 2 (Calefacción encendida).

De forma análoga, se realizan pruebas con el sensor de intensidad lumínica, en este caso existiendo cuatro casos posibles: modo automático con poca luz, modo automático con mucha luz, modo manual ON y modo manual OFF.

Para los dos últimos casos se puede observar claramente con las interfaces, según los elementos activos e inactivos:

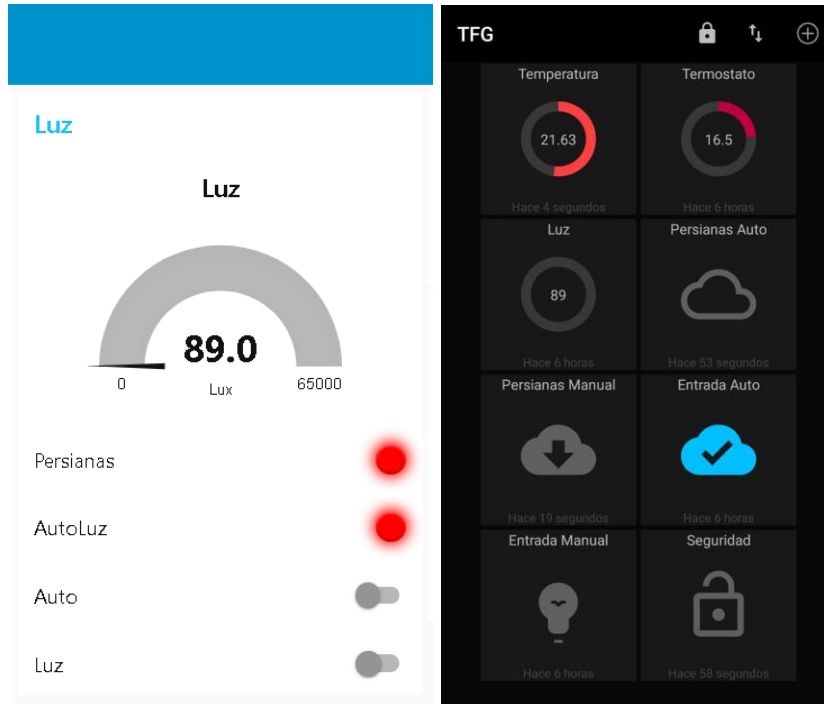


Ilustración 42 Interfaz web (Izquierda) y móvil (Derecha) para el caso 4 (manual OFF).

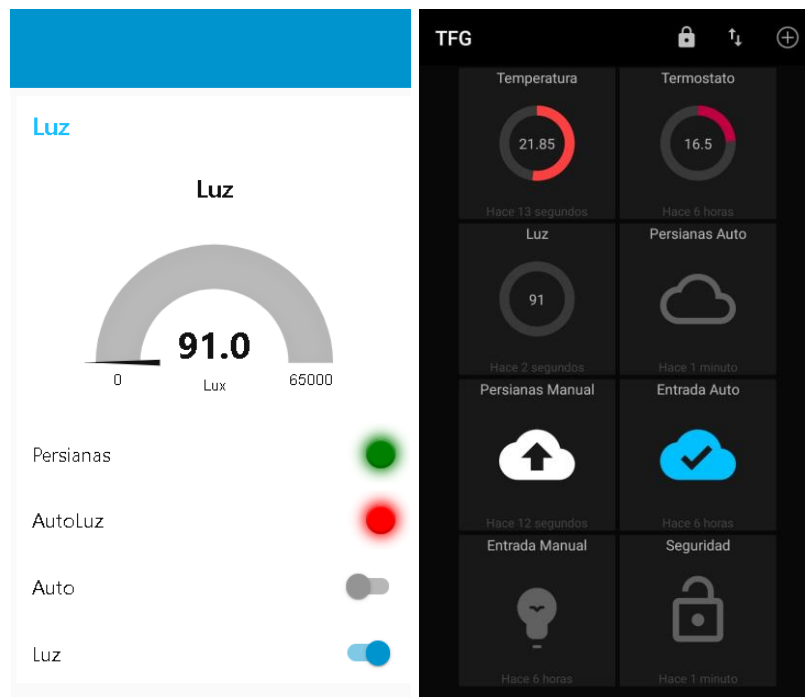


Ilustración 43 Interfaz web (Izquierda) y móvil (Derecha) para el caso 3 (manual ON).

En los casos 1 y 2, se comprueba con el modo automático y utilizando un foco de luz centrado en el sensor de forma que se superen los 1000 luxes de corte establecidos.

Así pues, para el caso 1, modo automático con poca luz se obtiene:

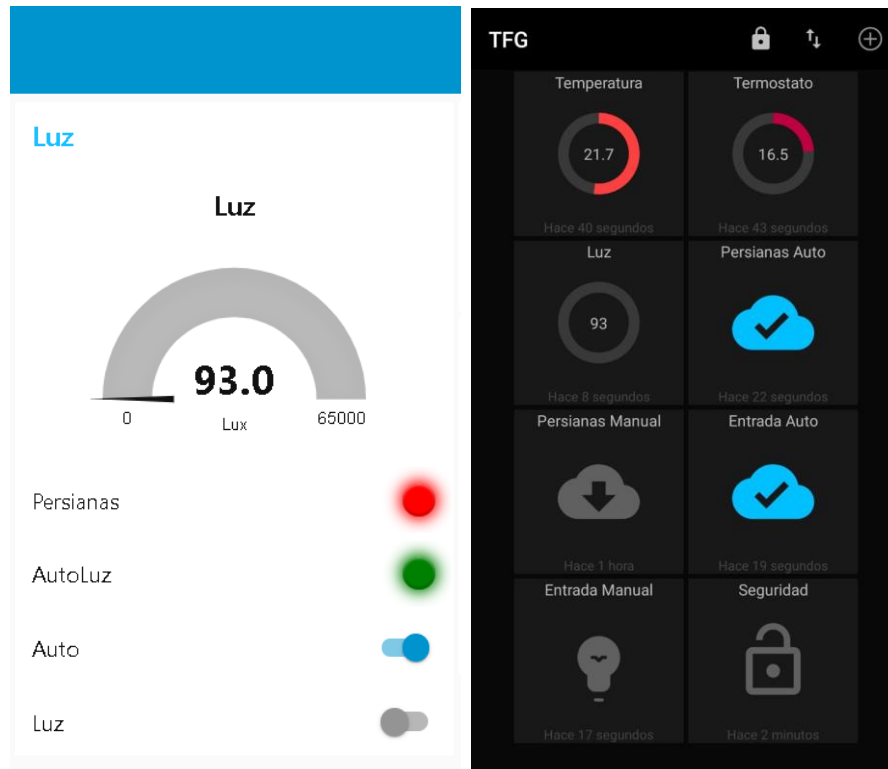


Ilustración 44 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (automático, poca luz).

Para este caso 1 podemos observar también que el led correspondiente a las persianas (amarillo) del prototipo se mantiene apagado, al no estar presente el foco de luz.

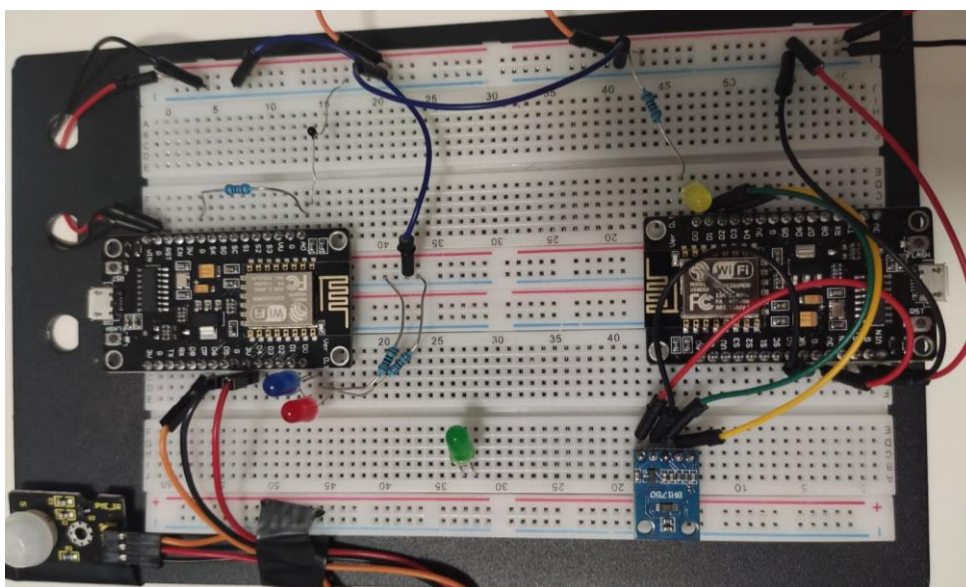


Ilustración 45 Prototipo en el caso 1 (automático, poca luz).

Por otro lado, para el caso 2, modo automático con mucha luz se obtiene:

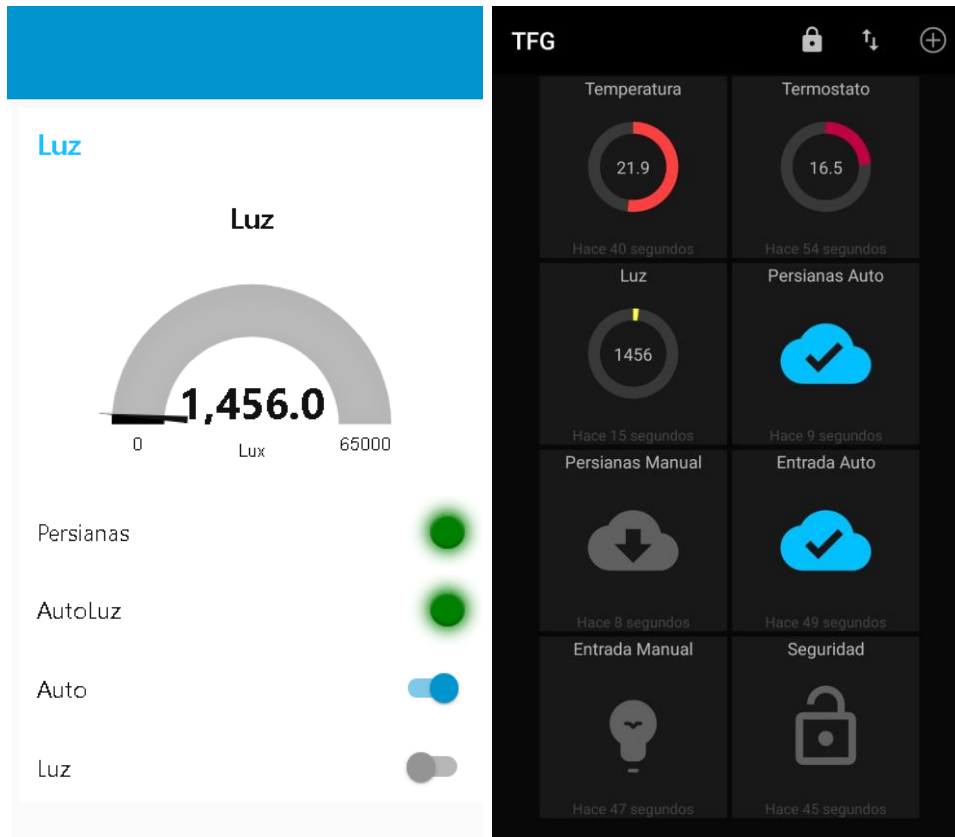


Ilustración 46 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (automático, mucha luz).

En este segundo caso se enciende el led amarillo del prototipo, ya que lo enfocamos directamente con luz artificial.

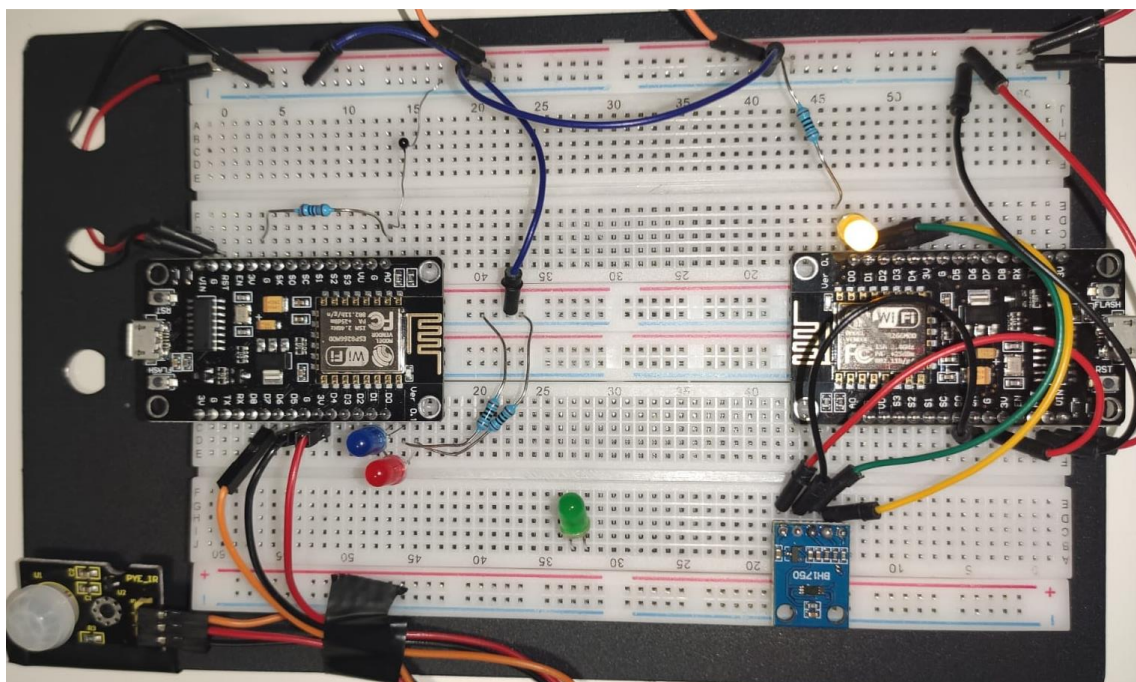


Ilustración 47 Prototipo en el caso 2 (automático, mucha luz).

Resultados

Finalmente, se prueba la función de presencia, comprobando de nuevo cuatro casos, dos automáticos y dos manuales.

Así pues, para los dos casos manuales se muestran las interfaces.

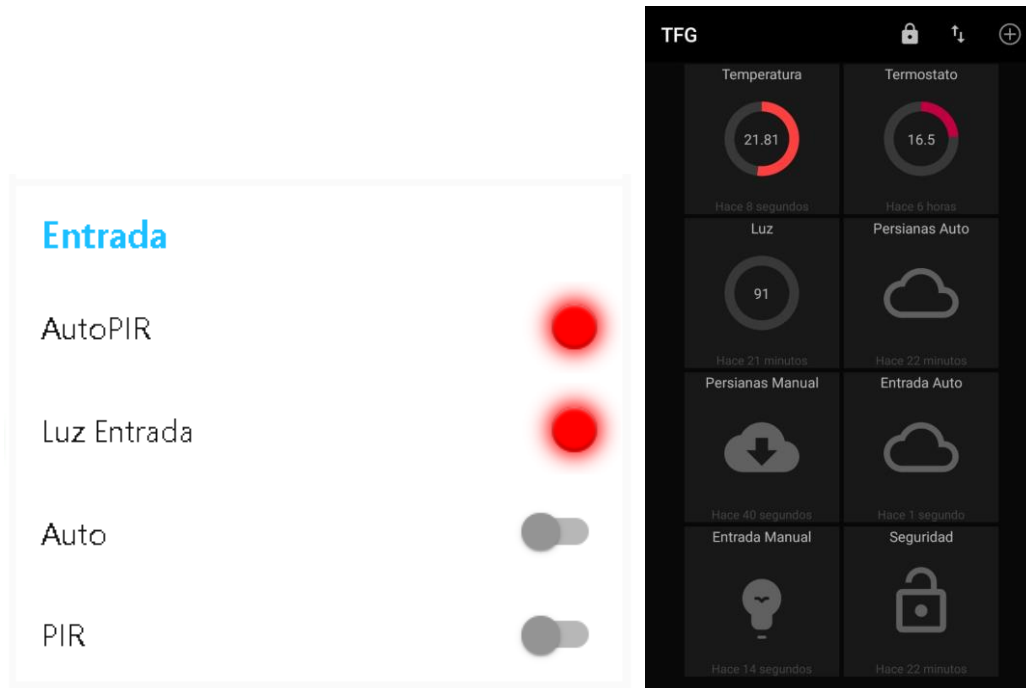


Ilustración 48 Interfaz web (Izquierda) y móvil (Derecha) para el caso 4 (manual OFF).

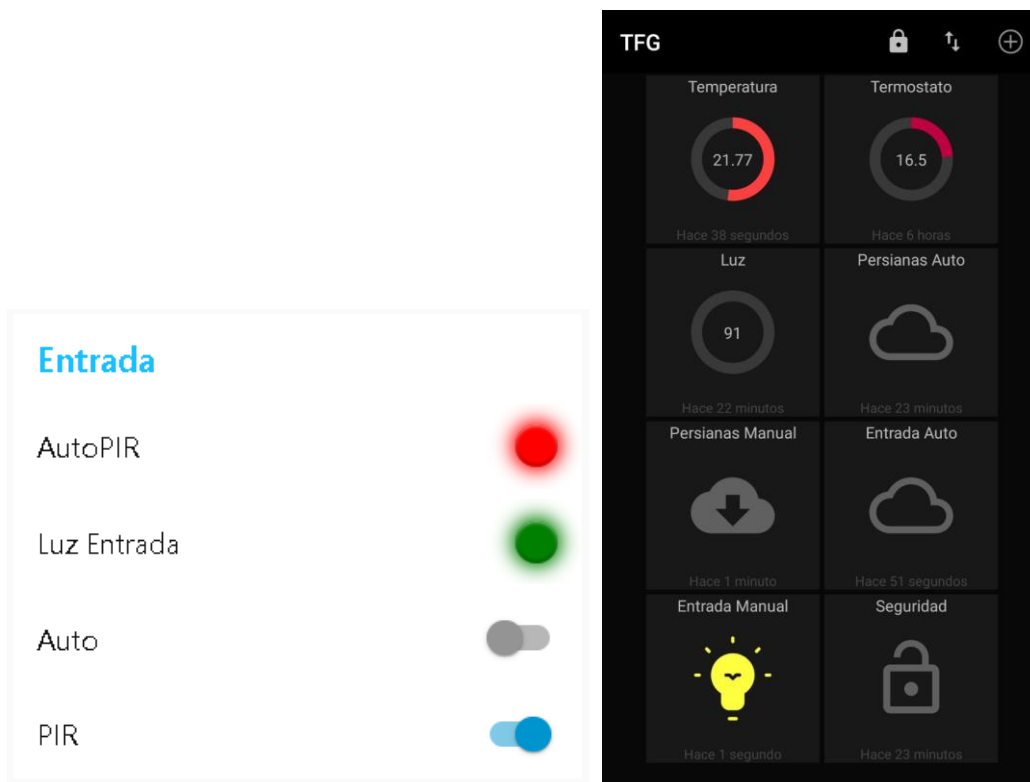


Ilustración 49 Interfaz web (Izquierda) y móvil (Derecha) para el caso 3 (manual ON).

En los casos 1 y 2, se comprueba con el modo automático, activando el sensor de presencia.

Así pues, para el caso 1, modo automático sin detección se obtiene:

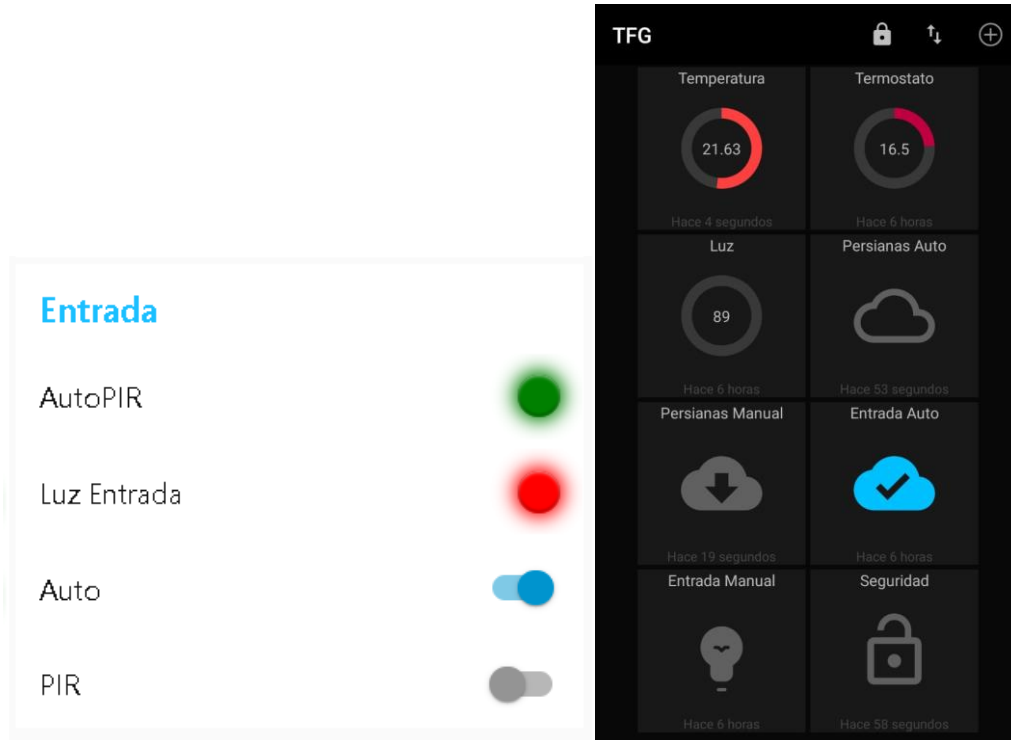


Ilustración 50 Interfaz web (Izquierda) y móvil (Derecha) para el caso 1 (automático, poca luz).

Para este caso 1 podemos observar también que el led activado por el sensor de presencia (azul) del prototipo se mantiene apagado, al no detectar ningún obstáculo.

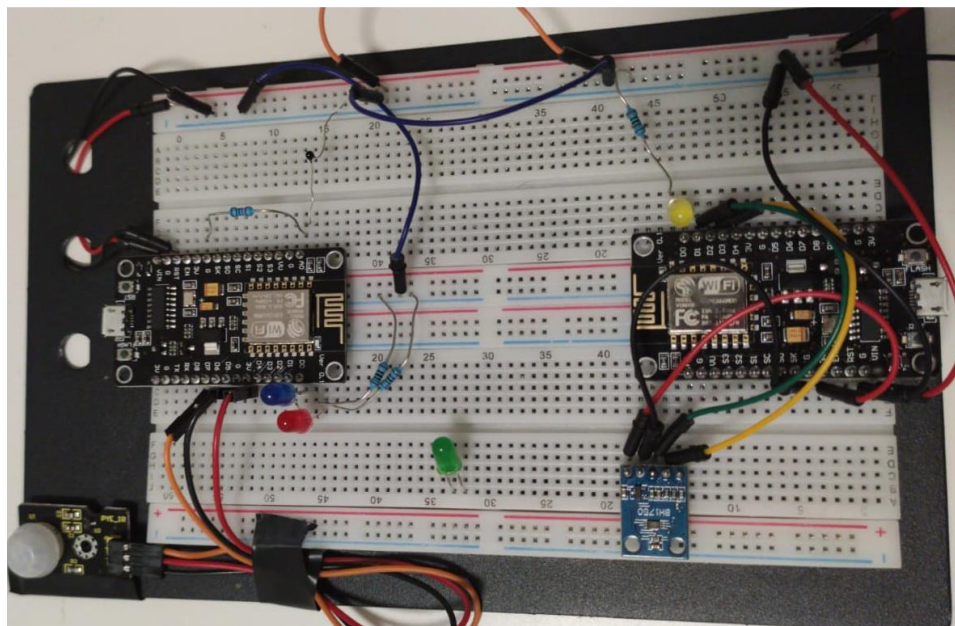


Ilustración 51 Prototipo en el caso 1 (automático, sin presencia).

Por otro lado, para el caso 2, modo automático con presencia se obtiene:

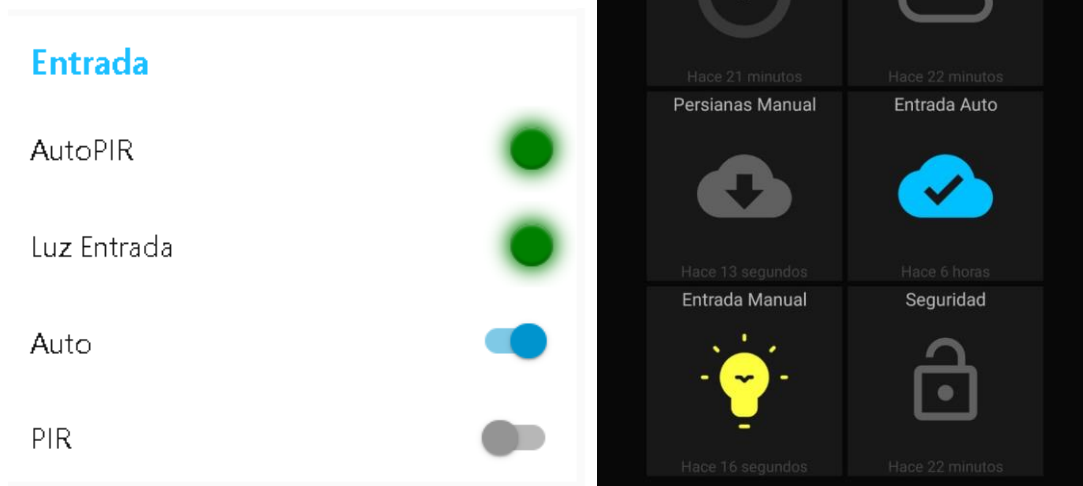


Ilustración 52 Interfaz web (Izquierda) y móvil (Derecha) para el caso 2 (automático, presencia).

En este segundo caso se enciende el led azul del prototipo, al poner un obstáculo y el sensor detectar presencia.

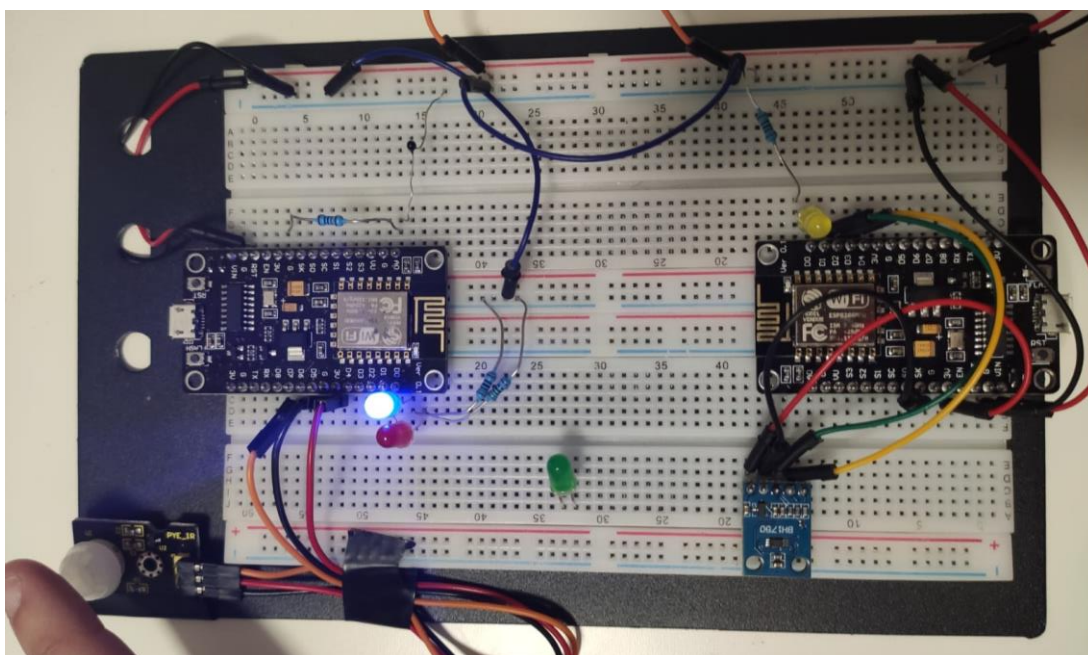


Ilustración 53 Prototipo en el caso 2 (automático, presencia).

Durante las pruebas se han observado bastantes problemas de conectividad debidos al broker utilizado, por lo que ha sido necesario sustituirlo en algunas ocasiones, e incluso utilizar uno para cada dispositivo.

Estos problemas se solventarían al optar por un broker local, con la desventaja de no poder hacer modificaciones fuera de la propia vivienda o el rango de la red local, o por un servicio de pago mensual que garantice un cierto mantenimiento y no limite el número de conexiones.

Por otra parte, los broker de código abierto de prueba han resultado más estables que el broker privado que nos genera la web de CloudMQTT, por lo que resultan más recomendables para este tipo de investigaciones.

8. CONCLUSIONES

Podemos concluir que los objetivos planteados al inicio de esta memoria se han alcanzado, ya que se ha realizado exitosamente el prototipado de la automatización de la vivienda, incluyendo las soluciones a los problemas descritos.

De esta forma, el control de la temperatura permite la gestión individualizada de cada habitación, tanto para invierno como para verano; el control a través del sensor de luz permite una mejor gestión de las persianas, sin dejar de lado un modo manual; el control de la luz de entrada permite la activación desde el exterior, no sólo al poderse controlar automáticamente sino también pudiendo activarse de forma remota; y el sistema que simula habitabilidad para añadir seguridad a la vivienda también se implementa en la plataforma.

En cuanto al prototipo resultante, tras los resultados de las diferentes pruebas efectuadas podemos considerar que es funcional en términos domésticos, ya que los sensores presentan un error asumible en este campo, pudiendo encontrar sustitutos más precisos en caso de que la aplicación lo requiera, aumentando el coste del sistema. El principal problema que se presenta en las pruebas viene debido a la comunicación, especialmente por parte del broker, que no es el idóneo al contar con limitaciones tanto de conexiones como de mensajes.

El sistema propuesto en la presente memoria destaca por su amplia compatibilidad, tanto con dispositivos inalámbricos o IoT como con dispositivos cableados, siendo además compatible con otros protocolos como el TCP/UDP o el HTTP.

Otra de las características más remarcables es la gran personalización y versatilidad de la que dispone, dado que la plataforma utilizada permite una programación libre, en contraste con otros sistemas existentes en el mercado basados en reglas o rutinas, lo cual permite su uso en otros ámbitos.

Todo ello se consigue con unos costes mínimos tanto de hardware, utilizando los componentes más económicos pero adecuados al sistema, como de software, dado que todos los programas empleados son de libre acceso y con código abierto.

8.1. TRABAJOS FUTUROS

El siguiente paso de este proyecto sería la implementación en la vivienda del sistema completo, proponiendo mejoras como el alojar la plataforma en un servidor que evite el mantener un ordenador encendido o, en su defecto, un SBC o Single-Board Computer, como puede ser una Raspberry Pi.

Por otra parte, al realizar la instalación del sistema se puede tener en cuenta que el añadir elementos IoT comerciales como un altavoz inteligente o una bombilla WiFi se puede realizar con pequeños cambios en el software.

Además, se han pensado funciones adicionales como un sistema de alarmas como despertador, permitiendo elegir varios tonos y haciendo que las persianas se suban a la hora de despertarse, o la adición de una cámara IP que permita una retransmisión en directo en la interfaz.

9. BIBLIOGRAFÍA

- [1] F. González, «Sistemas de Cableado Estructurado», 2016.
- [2] U. T. N. de A. Instituto Nacional Superior del Profesorado Técnico, «Powerline Communications (PLC)». jun-2004.
- [3] «¿Qué es KNX? - KNX Association [Official website]». [En línea]. Disponible en: <https://www2.knx.org/uy/knx/associacion/que-es-knx/index.php>. [Accedido: 10-nov-2019].
- [4] S. Farahani, *ZigBee wireless networks and transceivers*. Amsterdam ; Boston: Newnes/Elsevier, 2008.
- [5] «¿Qué es IoT (Internet Of Things)?», *Deloitte Spain*. [En línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html>. [Accedido: 11-nov-2019].
- [6] «¿Qué es Internet de las cosas (IoT)? - Definición en WhatIs.com», *SearchDataCenter en Español*. [En línea]. Disponible en: <https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT>. [Accedido: 11-nov-2019].
- [7] «Sensors - Latest research and news | Nature». [En línea]. Disponible en: <https://www.nature.com/subjects/sensors>. [Accedido: 02-nov-2019].
- [8] «Arduino - Introduction». [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>. [Accedido: 09-nov-2019].
- [9] «Overview - NodeMCU Documentation». [En línea]. Disponible en: <https://nodemcu.readthedocs.io/en/latest/>. [Accedido: 09-nov-2019].
- [10] R. Ierusalimschy, L. H. de Figueiredo, y W. Celes, «The evolution of Lua», en *Proceedings of the third ACM SIGPLAN conference on History of programming languages - HOPL III*, San Diego, California, 2007, pp. 2-1-2-26, doi: 10.1145/1238844.1238846.
- [11] «B-L475E-IOT01A - STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi - STMicroelectronics». [En línea]. Disponible en: https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-discovery-kits/b-l475e-iot01a.html. [Accedido: 31-ene-2020].
- [12] «Manejar cargas de más de 220V con Arduino y salida por relé», *Luis Llamas*. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-salida-rele/>. [Accedido: 19-ene-2020].
- [13] J. A. J. Toro, *UF1874 - Mantenimiento de la infraestructura de la red de comunicaciones*. Editorial Elearning, S.L., 2015.
- [14] «Protocolos de comunicación para IoT», *Luis Llamas*. [En línea]. Disponible en: <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>. [Accedido: 08-dic-2019].
- [15] «MQTT Brokers/Severs and Hosting Guide». [En línea]. Disponible en: <http://www.steves-internet-guide.com/mqtt-hosting-brokers-and-servers/>. [Accedido: 10-dic-2019].
- [16] *Técnicos de Soporte Informático de la Comunidad de Castilla Y Leon. Temario Volumen i Ebook*. MAD-Eduforma.
- [17] D. M. Ritchie, *The C Programming Language*. 1978.

- [18] B. Stroustrup, *The C++ programming language*, Fourth edition. Upper Saddle River, NJ: Addison-Wesley, 2013.
- [19] «Arduino Reference». [En línea]. Disponible en: <https://www.arduino.cc/reference/en/>. [Accedido: 17-dic-2019].
- [20] «JavaScript», *Documentación web de MDN*. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Accedido: 17-dic-2019].
- [21] «Acerca de JavaScript», *Documentación web de MDN*. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript. [Accedido: 17-dic-2019].
- [22] «JSON». [En línea]. Disponible en: <http://www.json.org/json-es.html>. [Accedido: 17-dic-2019].
- [23] F. M. Domínguez, *Instalaciones eléctricas de alumbrado e industriales*. Editorial Paraninfo, 2003.
- [24] T. Alvarez Bayona y Centro Nacional de Nuevas Tecnologías (INSHT), «Iluminación en el puesto de trabajo. Criterios para la evaluación y acondicionamiento de los puestos». Instituto Nacional de Seguridad e Higiene en el Trabajo (INSHT).
- [25] «Medir temperatura con Arduino y termistor (MF52)», *Luis Llamas*. [En línea]. Disponible en: <https://www.luisllamas.es/medir-temperatura-con-arduino-y-termistor-mf52/>. [Accedido: 04-feb-2020].
- [26] «SRS Thermistor Calculator». [En línea]. Disponible en: <https://www.thinksrs.com/downloads/programs/Therm%20Calc/NTCCalibrator/NTCCalculator.htm>. [Accedido: 04-feb-2020].



Relación de documentos

<input checked="" type="checkbox"/> Memoria	80	páginas
<input type="checkbox"/> Anexos	35	páginas

La Almunia, a 05 de 02 de 2020

Firmado: Alba Continente Sisamón



**Escuela Universitaria
Politécnica** - La Almunia
Centro adscrito
Universidad Zaragoza

Nº TFG:
424.19.51

Director:

Fdo:
Dr. Diego Antolín
Cañada

Título TFG:

Automatización de una vivienda unifamiliar

Autor:

Alba Continente Sisamón

5 de Febrero de 2020



**Escuela Universitaria
Politécnica** - La Almunia
Centro adscrito
Universidad Zaragoza

Nº TFG:
424.19.51

Director:

Fdo:
Dr. Diego Antolín
Cañada

Título TFG:

Automatización de una vivienda unifamiliar

Autor:

Alba Continente Sisamón

5 de Febrero de 2020



**Escuela Universitaria
Politécnica - La Almunia**
Centro adscrito
Universidad Zaragoza

**ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

Automatización de una vivienda
unifamiliar

Automatization of a single-family house

424.19.51

Autor: Alba Continente Sisamón
Director: Dr. Diego Antolín Cañada
Fecha: 5 de Febrero de 2020

