

Colegio Sant Josep
Sant Sadurní d'Anoia

Domótica open source

Curso 2018-19

Autor Roger Miranda Pérez
Tutor Adrià Segarra Torné
Curso 2do de Bachillerato B
Ámbito Ciencia y tecnología
Fecha 8 de enero de 2018

Índice

Agradecimientos	XI
Resumen	XII
I Introducción	
1 Objetivos del trabajo	2
2 Alcance del trabajo	3
II Información general	
3 Domótica	5
3.1 Servicios que ofrece.....	6
III Robótica	
4 Robótica	9
5 Arduino	10
6 LoLin NodeMCU v3	11
6.1 Ventajas y desventajas	12
7 Raspberry Pi	14
IV Interpretación.....	15
8 Diagrama de flujo	16
9 Tabla de verdad	18
9.1 Lógica binaria	18
V Diseño y montaje del sistema domótico	20
10 Introducción	21
11 Módulo botón.....	22
11.1 Sensores y actuadores.....	23

11.1.1	Botón pulsador.....	23
11.2	Conexión.....	23
11.3	Diagrama de flujo.....	24
11.4	Tabla de verdad.....	25
11.5	Montaje	25
12	Módulo movimiento	26
12.1	Sensores y actuadores.....	27
12.1.1	Sensor de movimiento	27
12.2	Conexión.....	28
12.3	Diagrama de flujo.....	29
12.4	Tabla de verdad.....	30
12.5	Montaje	30
13	Módulo apertura	32
13.1	Sensores y actuadores.....	33
13.1.1	Sensor de apertura magnético	33
13.2	Conexión.....	34
13.3	Diagrama de flujo.....	35
13.4	Tabla de verdad.....	35
13.5	Montaje	36
14	Módulo seguridad	37
14.1	Sensores y actuadores.....	38
14.1.1	Buzzer	38
14.1.2	Sensor de llama	38
14.1.3	Sensores de gas.....	39
14.2	Conexión.....	39
14.3	Diagrama de flujo.....	40
14.4	Tabla de verdad.....	41
14.5	Montaje	41
15	Módulo climático simple	42
15.1	Sensores y actuadores.....	43
15.1.1	Fotoresistor.....	43
15.1.2	Sensor de humedad y temperatura.....	43
15.2	Conexión.....	44

15.3	Diagrama de flujo.....	45
15.4	Tabla de verdad.....	45
16	Módulo climático avanzado	46
16.1	Sensores y actuadores.....	47
16.1.1	Fotoresistor.....	47
16.1.2	Sensor de humedad y temperatura.....	47
16.1.3	Pluviómetro	47
16.2	Conexión.....	50
16.3	Diagrama de flujo.....	50
16.4	Tabla de verdad.....	51
16.5	Montaje	51
17	Módulo alarma	53
17.1	Sensores y actuadores.....	54
17.1.1	Buzzer.....	54
17.1.2	4-Digit Display.....	54
17.1.3	Lector de huellas dactilares GT-521F52	55
17.1.4	4x4 Keypad	55
17.2	Conexión.....	56
17.3	Diagrama de flujo.....	57
17.4	Tabla de verdad.....	58
17.5	Montaje	59
18	Módulo enchufe	62
18.1	Sensores y actuadores.....	63
18.1.1	Relé	63
18.1.2	SCT-013	64
18.2	Conexión.....	65
18.3	Diagrama de flujo.....	66
18.4	Tabla de verdad.....	67
18.5	Montaje	67
19	Módulo riego	71
19.1	Sensores y actuadores.....	72
19.1.1	Sensor de humedad	72
19.1.2	Crowtail-smart pump shield	72

19.1.3	Válvula de cuatro vías	73
19.1.4	Bomba de agua	73
19.2	Conexión.....	74
19.3	Diagrama de flujo.....	75
19.4	Tabla de verdad.....	75
19.5	Montaje	75
20	Módulo estación meteorológica	77
20.1	Sensores y actuadores.....	78
20.1.1	Shield	78
20.1.2	Pluviómetro	78
20.1.3	Anemómetro	79
20.1.4	Veleta	79
20.2	Conexión.....	80
20.3	Diagrama de flujo.....	80
20.4	Tabla de verdad.....	80
20.5	Montaje	81
21	Módulo LED	82
21.1	Sensores y actuadores.....	83
21.1.1	LED Strip RGBWW.....	83
21.1.2	Transistores.....	83
21.2	Conexión.....	83
21.3	Diagrama de flujo.....	84
21.4	Tabla de verdad.....	85
21.5	Montaje	85
22	Módulo central	88
22.1	Domótica	89
22.2	WEB	91
22.3	Base de datos	95
22.4	Montaje	96
VI	Conclusiones	
23	Introducción	99
24	Seguridad	100

25 Actuadores	103
26 Sensores	104
27 Servicios	105
28 Comunicación	106
29 Precio	107
30 Instalación y montaje	111
31 Sostenibilidad medioambiental	112
32 Código abierto y hardware libre	114
33 Conclusiones generales	115
 VII Anexos	
A MQTT	117
B Código	118
B.1. Arduino.....	118
B.1.1 Módulo botón.....	119
B.1.2 Módulo movimiento	121
B.1.3 Módulo apertura	123
B.1.4 Módulo seguridad.....	125
B.1.5 Módulo climático simple	128
B.1.6 Módulo climático avanzado.....	131
B.1.7 Módulo alarma	134
B.1.8 Módulo enchufe	140
B.1.9 Módulo riego.....	142
B.1.10 Módulo estación meteorológica.....	145
B.1.11 Módulo LED	145
B.2. Python.....	148
B.3. WEB.....	156
B.3.1 /index.php.....	156
B.3.2 /login.php.....	172
B.3.3 /graph.php	172
B.3.4 /editar.php.....	176
B.3.5 /camara.php	180
B.3.6 /post.php.....	181

B.3.7	/files/php/GG.php	181
B.3.8	/files/php/index.php.....	182
B.3.9	/files/php/index.php.....	183
B.3.10	/files/css/barra.css	184
B.3.11	/files/css/misestilos.css	185
B.3.12	/files/form/login.php	185
B.3.13	/files/form/eliminarUser.php.....	186
B.3.14	/files/form/eliminarBtn.php	187
B.3.15	/files/form/cambiarSend.php	188
B.3.16	/files/form/cambiarRefresh.php	189
B.3.17	/files/form/cambiarNombre.php	190
B.3.18	/files/form/addUser.php	191
B.3.19	/files/form/addBtn.php	192
C	Precios actuales	118
Bibliografía		198

Índice de figuras

3.1: IDEA DE SISTEMA DOMÓTICO	5
3.2: PRECIO DEL KWH EN FUNCIÓN DE LA HORA DEL DÍA	6
4.1: EJEMPLO DE ROBOT HUMANOIDE.....	9
5.1: PLACA ARDUINO UNO R3.....	10
6.1: PLACA LOLIN NODEMCU.....	11
6.2: NODEMCU v3 (IZQUIERDA), Y NODEMCU v1 Y V2 (DERECHA)	12
6.3: COMPARATIVA DE LAS PLACAS NODEMCU.....	13
7.1: RASPBERRY PI.....	14
8.1: DIAGRAMA DE FLUJO PARA UN LOOP	16
8.2: SÍMBOLO DE DOCUMENTO IMPRESO.....	17
8.3: SÍMBOLO DE ENTRADA/SALIDA	17
8.4: SÍMBOLO DE BASE DE DATOS.....	17
10.1: ESQUEMA DEL SISTEMA DOMÓTICO	21
11.1: MÓDULO BOTÓN	22
11.2: ESQUEMA DE CONEXIÓN I DEL MÓDULO BOTÓN	23
11.3: ESQUEMA DE CONEXIÓN II DEL MÓDULO BOTÓN.....	24
11.4: DIAGRAMA DE FLUJO DEL MÓDULO BOTÓN.....	24
11.5: MÓDULO BOTÓN TERMINADO	25
12.1: MÓDULO MOVIMIENTO	26
12.2: SENSOR PIR.....	27
12.3: EJEMPLO DE SENSOR PIR	27
12.4: DETECCIÓN DEL PIR	28
12.5: CONEXIÓN DEL SENSOR PIR	28
12.6: ESQUEMA DE CONEXIÓN DEL MÓDULO MOVIMIENTO	28
12.7: DIAGRAMA DE FLUJO DEL MÓDULO MOVIMIENTO.....	29
12.8: MÓDULO MOVIMIENTO TERMINADO.....	31
12.9: MÓDULO MOVIMIENTO (CONEXIONES)	31
13.1: MÓDULO APERTURA.....	32
13.2: FUNCIONAMIENTO DEL MÓDULO APERTURA	33
13.3: INHIBICIÓN DEL MÓDULO APERTURA.....	33
13.4: ESQUEMA DE CONEXIÓN DEL MÓDULO APERTURA.....	34
13.5: DIAGRAMA DE FLUJO DEL MÓDULO APERTURA	35
13.6: PROTOTIPO DEL MÓDULO APERTURA	36
13.7: MÓDULO APERTURA TERMINADO	36
14.1: MÓDULO SEGURIDAD.....	37
14.2: BUZZER.....	38
14.3: SENSOR DE LLAMA	38
14.4: ESQUEMA DE CONEXIÓN DEL MÓDULO SEGURIDAD.....	40
14.5: DIAGRAMA DE FLUJO DEL MÓDULO SEGURIDAD	40
14.6: MÓDULO SEGURIDAD	41
14.7: CONEXIONES DEL MÓDULO SEGURIDAD	41

15.1: MÓDULO CLIMÁTICO SIMPLE	42
15.2: CONEXIÓN DE UN FOTORESISTOR	43
15.3: DHT11	44
15.4: CONVERSIÓN ANALÓGICO-DIGITAL DEL DHT11	44
15.5: ESQUEMA DE CONEXIÓN DEL MÓDULO CLIMÁTICO SIMPLE	44
15.6: DIAGRAMA DE FLUJO DEL MÓDULO CLIMÁTICO SIMPLE.....	45
16.1: MÓDULO CLIMÁTICO AVANZADO	46
16.2: PLUVIÓMETRO	47
16.3: SENSOR DE AGUA.....	48
16.4: ESQUEMA DEL SENSOR DE AGUA PARA ARDUINO.....	48
16.5: GRÁFICA DE VALORES EN FUNCIÓN DE LA ALTURA DEL AGUA.....	49
16.6: ESQUEMA DE CONEXIÓN DEL MÓDULO CLIMÁTICO AVANZADO	50
16.7: DIAGRAMA DE FLUJO DEL MÓDULO CLIMÁTICO AVANZADO	50
16.8: MONTAJE DE LA ENTRADA MICROSD	51
16.9: PLACA CON VARIOS SENSORES INSTALADOS	51
16.10: FOTORESISTOR	52
16.11: SENSOR DE ALTURA DEL AGUA	52
16.12: MÓDULO CLIMÁTICO AVANZADO TERMINADO	52
17.1: MÓDULO ALARMA.....	53
17.2: 4-DIGIT DISPLAY	54
17.3: PANTALLA LCD PARA ARDUINO.....	54
17.4: LECTOR DE HUELLAS DACTILARES.....	55
17.5: 4X4 KEYPAD	55
17.6: CONEXIÓN DE LOS BOTONES.....	56
17.7: ESQUEMA DE CONEXIÓN DEL MÓDULO ALARMA.....	56
17.8: DIAGRAMA DE FLUJO DEL MÓDULO ALARMA	57
17.9: 4X4 KEYPAD Y 4-DIGIT DISPLAY	59
17.10: PRUEBA DE CÓDIGO	59
17.11: PROTOTIPO DEL MÓDULO ALARMA.....	59
17.12: SHIELD CREADO PARA EL MÓDULO ALARMA.....	60
17.13: PLACA CONECTADA AL VOLTAJE, DENTRO DEL DISEÑO 3D	60
17.14: MÓDULO ALARMA (INTERIOR)	60
17.15: SENSOR DE HUELLAS (INTERIOR).....	61
17.16: MÓDULO ALARMA TERMINADO	61
18.1: MÓDULO ENCHUFE	62
18.2: RELÉ	63
18.3: PARTES DEL RELE.....	63
18.4: SCT-013	64
18.5: FUNCIONAMIENTO DEL SCT-013.....	64
18.6: CONEXIÓN DEL MÓDULO ENCHUFE.....	65
18.7: DIAGRAMA DE FLUJO DEL MÓDULO ENCHUFE.....	66
18.8: INTERIOR DEL ENCHUFE HEMBRA.....	67
18.9: INTERIOR DEL ENCHUFE MACHO.....	67
18.10: ENCHUFES CONECTADOS CON EL RELE.....	67
18.11: PRUEBA DEL MÓDULO ENCHUFE	68
18.12: INTENTO DE CONEXIÓN DEL SCT-013	68
18.13: PROTOTIPO DEL MÓDULO ENCHUFE	69

18.14: PARTE INFERIOR DEL MÓDULO ENCHUFE	69
18.15: INTERIOR DEL MÓDULO ENCHUFE	69
18.16: PARTE SUPERIOR DEL MÓDULO ENCHUFE	70
18.17: PARTE INFERIOR DEL MÓDULO ENCHUFE	70
19.1: MÓDULO RIEGO	71
19.2: SENSOR DE HUMEDAD	72
19.3: SHIELD DEL KIT DE RIEGO.....	72
19.4: VÁLVULA DE CUATRO VÍAS.....	73
19.5: BOMBA DE AGUA.....	73
19.6: CÁLCULO DEL CAUDAL DE LA BOMBA DE AGUA	74
19.7: DIAGRAMA DE FLUJO DEL MÓDULO RIEGO.....	75
19.8: PREPARACIÓN DEL MONTAJE DEL MÓDULO RIEGO	76
19.9: MÓDULO CONECTADO	76
20.1: ESTACIÓN METEOROLÓGICA	77
20.2: SHIELD DE LA ESTACIÓN METEOROLÓGICA.....	78
20.3: PLUVÍMETRO	78
20.4: ANEMÓMETRO	79
20.5: VELETA.....	79
20.6: DIAGRAMA DE FLUJO DEL MÓDULO ESTACIÓN METEOROLÓGICA	80
20.7: CAJA CON LAS PARTES DEL MÓDULO	81
20.8: PARTES DEL MÓDULO	81
21.1: MÓDULO LED	82
21.2: TIRA LED RGBWW	83
21.3: CONEXIÓN DEL MÓDULO LED	83
21.4: DIAGRAMA DE FLUJO DEL MÓDULO LED	84
21.5: MÓDULO LED POR DEBAJO	85
21.6: BRILLO DEL MÓDULO LED	86
21.7: MÓDULO LED FUNCIONANDO	86
21.8: TIRA RGBWW	86
21.9: MÓDULO LED	87
21.10: MÓDULO LED TERMINADO.....	87
22.1: DIAGRAMA DE FLUJO DEL MÓDULO CENTRAL.....	90
22.2: WEB > INFORMACIÓN	91
22.3: WEB > GRÁFICA	92
22.4: WEB > CÁMARA	93
22.5: WEB > EDITAR.....	94
22.6: BASE DE DATOS	95
22.7: INSTALACIÓN DE LA PANTALLA TÁCTIL	96
22.8: CÁMARA PARA RASPBERRY PI CON CARCASA.....	96
22.9: CÁMARA MONTADA.....	97
24.1: ERROR GENERADO POR UNA CONSULTA SQL.....	101
24.2: EJEMPLO DE SQL INJECTION.....	102
A.1: ESQUEMA DE UN SISTEMA CONECTADO CON MQTT.....	117
C.1: PRECIO MÍNIMO.....	195
C.2: PRECIO MÁXIMO.....	197

ÍNDICE DE TABLAS

9.1: EJEMPLO DE TABLA DE VERDAD	18
9.2: EJEMPLO DE XOR	19
12.1: TABLA DE VERDAD DEL MÓDULO MOVIMIENTO	30
13.1: TABLA DE VERDAD DEL MÓDULO APERTURA.....	35
20.1: TABLA DE VERDAD DEL MÓDULO SEGURIDAD.....	41
17.1: TABLA DE VERDAD DEL MÓDULO ALARMA.....	58
18.1: TABLA DE VERDAD DEL MÓDULO ENCHUFE	67
21.1: TABLA DE VERDAD DEL MÓDULO LED	85
23.1: PUNTOS A FAVOR Y EN CONTRA DEL SISTEMA DOMÓTICO DE ESTE TRABAJO	99
29.1: COSTE FIJO POR MÓDULO.....	107
29.2: COSTE DEL MÓDULO BOTÓN	107
29.3: COSTE DEL MÓDULO MOVIMIENTO.....	108
29.4: COSTE DEL MÓDULO APERTURA	108
29.5: COSTE DEL MÓDULO SEGURIDAD	108
29.6: COSTE DEL MÓDULO CLIMÁTICO SIMPLE.....	108
29.7: COSTE DEL MÓDULO CLIMÁTICO AVANZADO.....	109
29.8: COSTE DEL MÓDULO ALARMA	109
29.9: COSTE DEL MÓDULO ENCHUFE.....	109
29.10: COSTE DEL MÓDULO LED	109
29.11: COSTE DEL MÓDULO RIEGO	110
29.12: COSTE DEL MÓDULO ESTACIÓN METEOROLÓGICA	110
29.13: COSTE DE LA RASPBERRY	110

Agradecimientos

Quiero agradecer a mis padres, Ramón Miranda e Isabel Pérez, por financiarme la gran mayoría del proyecto. A mi tutor del TREC, Adrià Segarra, por sus recomendaciones y ánimos. A Ánder Hurtado, por ayudarme a configurar Linux (Apache y MySQL). A Indalecio13, por ayudarme con el esquema del sensor de agua, y a Larry Sanger y a Jimmy Wales, además de todos los creadores de los libros y WEBs usadas en este trabajo.

Resumen

La domótica es un sistema capaz de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación.

En este proyecto se creará un sistema domótico a partir de Arduino y Raspberry Pi. Cada placa Arduino tendrá ciertos sensores o actuadores, dependiendo de su función, que le proporcionan información y esta se enviará de forma inalámbrica mediante Wi-Fi (usando un protocolo MQTT) a Raspberry Pi. Una vez los datos son recibidos, estos se procesan y se realizan otras acciones, como mostrarlos o enviar otra señal.

El sistema domótico incluirá una página WEB para que el usuario interactúe con él de forma cómoda. Además, tiene una base de datos que almacena información para mostrar información de los últimos segundos, minutos, horas, o días y guardar preferencias.

Todo el proyecto será publicado para lograr un mayor impacto, y que más población pueda disfrutar de las ventajas de tener un sistema domótico.

El elevado coste de los sistemas domóticos actúa como barrera. Con este proyecto, se espera reducir considerablemente el precio total. Además, esta alternativa será más sostenible y adaptable para cada usuario (el consumidor podrá modificar el software y hardware de acuerdo con sus necesidades).

Parte I

Introducción

Capítulo 1

Objetivos del trabajo

El objetivo de este trabajo es la creación de un sistema domótico open source de bajo precio, que permita acercar la domótica a un mayor público. Con ello se pretende colaborar en la transición hacia un consumo de energía más sostenible y más inteligente, facilitando la integración de las fuentes de energía renovables.

Capítulo 2

Alcance del trabajo

- » Creación de módulos para el sistema domótico
- » Creación de un programa que procese toda la información
- » Creación de una página WEB para una interacción más sencilla para el usuario
- » Creación de una base de datos para el almacenamiento de información

Parte II

Información general

Capítulo 3

Domótica¹

Se llama domótica a los sistemas capaces de automatizar una vivienda o edificación, aportando servicios de gestión energética, seguridad, bienestar y comunicación (Figura 3.1), y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar.



Figura 3.1: Idea de sistema domótico

El término domótica viene de la unión de las palabras domus (que significa casa en latín) y tica (de automática, palabra en griego: Que funciona por sí sola).

La domótica y la robótica han experimentado un auge importante en los últimos cinco años. Cada vez más, las personas necesitamos poder controlar ciertos ambientes o aspectos de nuestros hogares, ya sea por comodidad, por falta de tiempo o ambas cosas a la vez.

La domótica no sólo nos libra de posibles tareas domésticas, sino que también puede ser empleada como una aliada para el gasto energético, con lo que miramos por el medio ambiente y por nuestra economía.

¹ (D'Addario, 2018, p. 13-20)

3.1 Servicios que ofrece

» *Ahorro energético*

Se puede llegar de muchas maneras, en muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos energía, sino una gestión eficiente de los mismos.

- Climatización y calderas:

Uso de termostatos. Se puede encender y apagar usando un control de enchufe, mediante telefonía móvil, fija, Wi-Fi o Ethernet.

- Control de toldos y persianas eléctricas:

Proteger automáticamente el toldo del viento, con un mismo sensor de viento que actúe sobre todos los toldos.

- Protección automática del sol:

Mediante un mismo sensor de sol que actúe sobre todos los toldos y persianas.

- Gestión eléctrica:

Desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado.

- Gestión de tarifas:

Deriva el funcionamiento de algunos aparatos a horas de tarifa reducida, estas pueden verse en la Figura 3.2.



Figura 3.2: Precio del kWh en función de la hora del día

» *Confort*

El confort conlleva todas las actualizaciones que se puedan llevar a cabo que mejoren la comodidad de una vivienda, dichas actualizaciones pueden ser de carácter tanto pasivo, como activo o mixtas.

- Iluminación:

Apagado general de todas las luces de la vivienda.

- Automatización del apagado/encendido en cada punto de luz.

- Regulación de la iluminación según el nivel de luminosidad ambiente.
 - Automatización de todos los distintos sistemas: Dotándolos de control eficiente y de fácil manejo.
 - Integración del portero al teléfono, o del videoportero al televisor.
- » *Control vía Internet*
Generación de macros (grupo de instrucciones programadas cuya tarea es la automatización de tareas repetitivas) y programas de forma sencilla para el usuario y automatización.
- » *Seguridad*
Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales como la seguridad personal.
- Alarms de intrusión:
Se utilizan para detectar o prevenir la presencia de personas extrañas en una vivienda o edificio.
 - Cierre de persianas puntual y seguro.
 - Detectores y alarmas de detección de incendios, detector de gas, escapes de agua e inundación, concentración de monóxido de carbono...
 - Alerta médica y teleasistencia.
 - Acceso a cámaras IP.
- » *Comunicaciones*
Sistemas o infraestructuras de comunicaciones que posee el hogar.
Control remoto desde Internet, PC, mandos inalámbricos, aparellaje eléctrico...
- Teleasistencia.
 - Telemantenimiento.
 - Informes de consumo y costes.
 - Transmisión de alarmas.
 - Intercomunicaciones.
 - Telefonillos y videoporteros.

Parte III

Robótica

Capítulo 4

Robótica²

La idea de “robot” surgió hace más de 3000 años. Podemos decir que, hace más de 3000 años aproximadamente que el ser humano piensa en cómo desarrollar máquinas autónomas que realicen ciertos trabajos repetitivos, duros y nocivos. A estas máquinas es a lo que podríamos llamar robots.

El origen de la palabra Robot podría venir de la palabra checa *roboata*, que significa trabajo forzado, obligación o esclavo.

Isaac Asimov acuñó la palabra robótica y estableció unas reglas o leyes que deberían tener programadas todos los robots en sus “cerebros”. Es lo que conocemos como las 3 leyes de la robótica.

- » 1^a ley: Un robot no debe dañar a un ser humano, ni por su inacción dejar que resulte dañado.
- » 2^a ley: Un robot obedecerá a un humano, excepto cuando dichas órdenes entren en conflicto con la 1^a ley.
- » 3^a ley: Un robot debe protegerse así mismo, excepto si esta ley entra en conflicto con la 1^a o 2^a ley.

Un robot es un dispositivo manipulador controlado por computador. Puede verse un ejemplo de robot humanoide³ en la Figura 5.1.

Hace más de 30 o 40 años que se empezaron a introducir robots en el entorno empresarial, sobre todo en las fábricas de construcción de vehículos, donde hoy en día, realizan operaciones repetitivas en una cadena de montaje.



Figura 4.1: Ejemplo de robot humanoide

² (Porcuna López, 2016, p. 25-26)

³ Los robots humanoides son robots diseñados para simular la forma y los movimientos de un ser humano.

Capítulo 5

Arduino⁴

Arduino se ha erigido como una de las plataformas escogidas para llevar a cabo proyectos tecnológicos. Esto es así debido a su versatilidad como instrumento para este cometido, por la gran cantidad de sensores que puede incorporar y por los precios que se están alcanzando, lo que permite hoy en día a los usuarios tenerlo al alcance de la mano y del bolsillo. Debido también a la facilidad con la que se puede programar e interactuar con el medio que nos rodea. Por todo esto, Arduino es idóneo para llevar a cabo proyectos sobre robótica, alcanzando un nivel nada despreciable.

Arduino (Figura 5.1) es una placa de circuito impreso. Junto a unos componentes electrónicos, un microcontrolador y una serie de pines de entrada y salida, podemos crear proyectos basados en sistemas electrónicos con ella; esto incluye materias como la robótica, la domótica u otros proyectos de carácter electrónico en los que podamos pensar.

Arduino es una plataforma *open-hardware*. Todos los esquemas de placa, diseños y componentes son accesibles para todo el mundo, es decir, son públicos.

Ahora mismo, podemos encontrar en el mercado versiones más potentes de la familia Arduino, así como competidores en este terreno como las placas BeagleBone o Raspberry Pi.



Figura 5.1: Placa Arduino UNO R3

⁴ (Porcuna López, 2016, p. 34-51)

Capítulo 6

LoLin NodeMCU v3⁵

El módulo NodeMCU v3 de LoLin (Figura 6.1) es una tarjeta de desarrollo basado en el ESP8266-12E. El módulo posee algunas características que les diferencia con el NodeMCU tradicional. Está diseñado para facilitar el desarrollo de proyectos con ESP8266⁶. El corazón del NodeMCU es el módulo ESP-12E, el cual pertenece a la familia ESP-XX de la compañía All-Thinker.



Figura 6.1: Placa LoLin NodeMCU

Su tamaño físico es superior a las versiones 1 y 2 (Figura 6.2), dispone de una salida de alimentación de 5v conectado directamente a la alimentación del puerto USB, y usa el chip CH340G como convertidor TTL a USB, a diferencia de la versión 1.0 que utilizaba el chip CP2102.

⁵ (Díaz Gómez, 2016)

⁶ ESP8266 es un chip de bajo coste para dotar a los proyectos de conexión Wi-Fi.



Figura 6.2: NodeMCU v3 (izquierda), y NodeMCU v1 y v2 (derecha)

6.1 Ventajas y desventajas

NodeMCU tiene la posibilidad de desarrollar proyectos con ESP8266, el cual es un chip con Wi-Fi integrado producido por la compañía china Espressif Systems, y dicho chip tiene un precio bajo. Al contar con Wi-Fi permite realizar proyectos de *IoT* (*Internet of Things*) a muy bajo precio, además de ser programable con el IDE de Arduino.

Las placas salieron a un precio aproximado de 2,61€/unidad. Ese es un precio extremadamente bajo si se tiene en cuenta que para hacerlo con Arduino UNO R3 (con el *shield* de Wi-Fi incluido) saldría por unos 45€/unidad. Cabe destacar que Arduino UNO R3 dispone de más memoria, pero calidad-precio no merece la pena.

No está adaptada para una *protoboard* (se explicará más a fondo en la sección 12.1), ya que no deja agujeros libres para los jumpers, como podemos observar en la Figura 6.3. Al ser el NodeMCU v3 más grande, no deja ningún punto de contacto (agujero en la “línea” vertical), evitando así la conexión de *jumpers*. En la derecha vemos el NodeMCU v2, al contar con un tamaño inferior nos deja un solo punto de contacto para cada pin, facilitando así la conexión para prototipos.

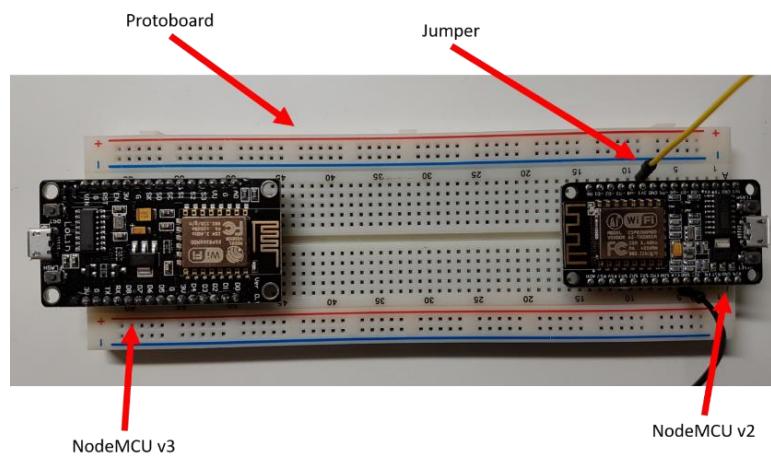


Figura 6.3: Comparativa de las placas NodeMCU

Capítulo 7

Raspberry Pi⁷

La idea básica del proyecto Raspberry Pi (Figura 7.1) es la de desarrollar una plataforma de computación pequeña y asequible que sirviera para estimular el interés de los niños por aprender los aspectos fundamentales de las tecnologías de la información y las telecomunicaciones. La rápida evolución de los dispositivos SoC (*System on a Chip*) de bajo coste para las aplicaciones móviles posibilitó una amplísima distribución de la plataforma asequible RPi a principios de 2012.



Figura 7.1: Raspberry Pi

Las placas Raspberry Pi propiamente dichas resultan demasiado complejas para que las utilice el público general. Es, sin embargo, su capacidad de ejecutar una versión empotrada (*embedded*) de Linux lo que las convierte en plataformas accesibles, adaptables y potentes. En conjunto, Linux y los sistemas empotrados hacen posible un desarrollo sencillo para este tipo de dispositivos, destinados a realizar funciones en edificios inteligentes, dispositivos IoT, robótica...

⁷ (Molloy, 2016, p. XIX)

Parte IV

Interpretación

Capítulo 8

Diagrama de flujo⁸

El diagrama de flujo (Figura 8.1) o flujograma o diagrama de actividades es la representación gráfica del algoritmo o proceso. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.

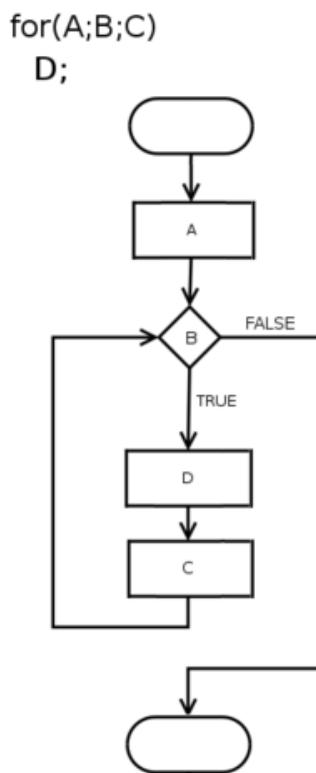


Figura 8.1: Diagrama de flujo para un loop⁹

Estos diagramas utilizan símbolos con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso.

La simbología y los significados son muy extensos, aquí hay algunos:

» *Óvalo o Elipse*

Inicio y final (abre y cierra el diagrama).

⁸ (Diagrama de flujo, 2018)

⁹ Bucle

» *Rectángulo*

Actividad (representa la ejecución de una o más actividades o procedimientos).

» *Rombo*

Decisión (formula una pregunta o cuestión).

» *Triángulo boca abajo*

Archivo definitivo (guarda un documento en forma permanente).

» *Triángulo boca arriba*

Archivo temporal (proporciona un tiempo para el almacenamiento del documento).

» *Flecha*

Indica el orden de la ejecución de las operaciones.

Además de estos, podemos encontrar el símbolo del documento impreso (Figura 14.2) (para imprimir un resultado, por ejemplo en la pantalla), el de entrada/salida (Figura 14.3) (representa la lectura de datos en la entrada y la impresión de datos en la salida), o el de la base de datos (Figura 14.4) (indica una lista de información con una estructura estándar que permite buscar y ordenar).



Figura 8.2: Símbolo de documento impreso

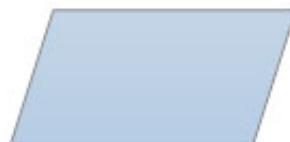


Figura 8.3: Símbolo de entrada/salida



Figura 8.4: Símbolo de base de datos

Capítulo 9

Tabla de verdad

Una tabla de verdad (Tabla 9.1) es una tabla que muestra un valor de una proposición compuesta, para cada combinación que se pueda asignar. Partiendo de un número de n variables, cada una de las cuales puede tomar el valor verdadero (1) o falso (0), podemos saber el número de combinaciones ($N_c = 2^n$).

Esto nos permite ver cómo reaccionará el circuito.

ENTRADAS		AND	NAND	OR	NOR	EXOR
A	B	0	1	0	1	0
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

funciones $A \cdot B$ $A \cdot \bar{B}$ $A+B$ $\bar{A}+\bar{B}$ $A \oplus B$

Tabla 9.1: Ejemplo de tabla de verdad

La salida procede de una expresión (la cual pueden darla, o dar los resultados y calcularla). En el caso de la Tabla 9.1 hay 2 entradas y 5 salidas. Las salidas tienen una forma de ser representadas, esta procede de operaciones binarias.

9.1 Lógica binaria¹⁰

La lógica binaria es la que trabaja con variables binarias y operaciones lógicas del Álgebra de Boole. Así, las variables sólo toman dos valores discretos: V (verdadero) y F (falso); aunque estos dos valores lógicos también se pueden denotar como sí y no, o como 1 y 0 respectivamente.

¹⁰ (Lógica binaria, 2018)

Las operaciones binarias fundamentales son las siguientes.

» *Negación lógica*

Invierte el valor lógico de su argumento, por lo que también se llama *NOT*. Se simboliza por una barra horizontal sobre su argumento (\bar{a}) o añadiendo un apóstrofo a continuación de él (a').

» *Unión o suma lógicas*

Función de varios argumentos que vale 0 sólo si todos sus argumentos valen 0, en el resto de casos vale 1. Equivale a la suma (en suma lógica $1+1=1$). Se representa como un operador binario entre sus argumentos, simbolizado por +, o bien *OR*.

» *Producto o multiplicación lógicos*

Función de varios argumentos que vale 1 sólo si todos sus argumentos valen 1, en el resto de los casos vale 0. Equivale a la multiplicación. Se representa como un operador binario entre sus argumentos, simbolizado por * o bien *AND*.

» *Operaciones lógicas compuestas*

Siguiendo el Álgebra de Boole, se pueden combinar operaciones, obteniendo así resultados más complejos. Un ejemplo sería $A^*(B+C)$.

También hay otras, menos usadas, como son el *XOR* y el *XNOR*. El *XOR* tiene una tabla de verdad como la Tabla 9.2, y el *XNOR* la misma, pero con la salida negada. Representadas con un símbolo de sumar, rodeado por un círculo, y en el caso del XNOR negándolo.

Entrada A	Entrada B	Salida
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 9.2: Ejemplo de *XOR*

Parte V

Diseño y montaje del sistema domótico

Capítulo 10

Introducción

Todo sistema domótico necesita sensores y actuadores, que en este trabajo se han decidido llamar “módulos”. Cada módulo tiene una función específica, que a menudo ya se especifica en su propio nombre.

Los módulos son los sentidos del sistema domótico. Con ellos el sistema obtendrá información como la temperatura, humedad, luz... para procesarla y realizar otras acciones, ya sea mostrar el valor en pantalla o enviar una señal de alarma.

En los módulos hay ciertas limitaciones. Uno de los objetivos del trabajo es que el sistema sea económico, así que una de las limitaciones es económica, aunque en algunos casos se ha optado por mejores sensores/actuadores para obtener mayor precisión, comodidad para el usuario, o posibilidad de realizar más acciones. Otra limitación es de conocimientos, ya que hay módulos extremadamente avanzados tanto en montaje como en programación que requieren de mayor experiencia.

Cada capítulo se divide en 5 secciones, estas están formadas por los sensores o actuadores que el módulo incluye, su conexión con la placa, el diagrama de flujo, la tabla de verdad y el montaje del módulo.

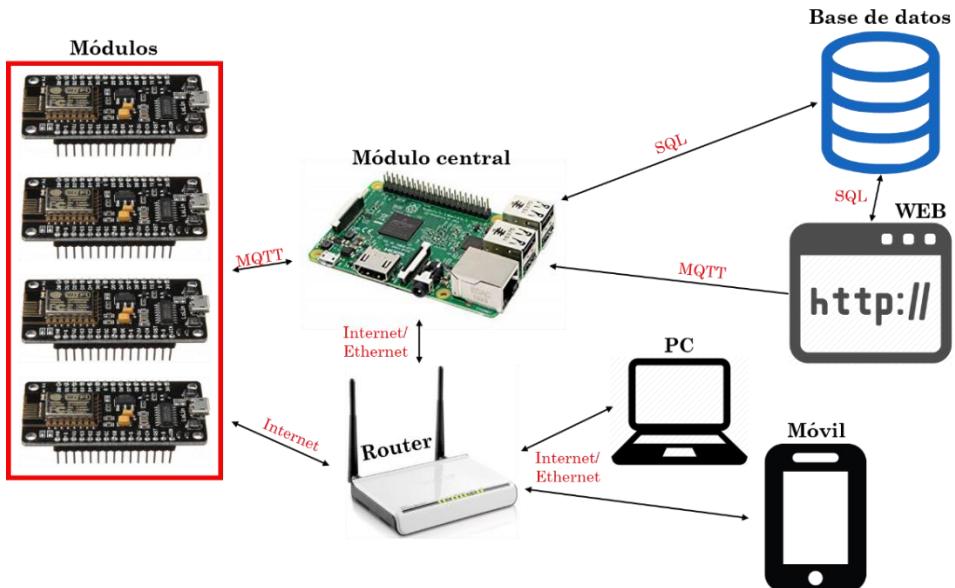


Figura 10.1: Esquema del sistema domótico

Capítulo 11

Módulo botón

El módulo botón (Figura 11.1) se usará para dar una señal al pulsar el botón que lo compone. Dicha señal será configurada des del ordenador central de forma que pueda realizar una acción, u otra.

Es el módulo más básico.

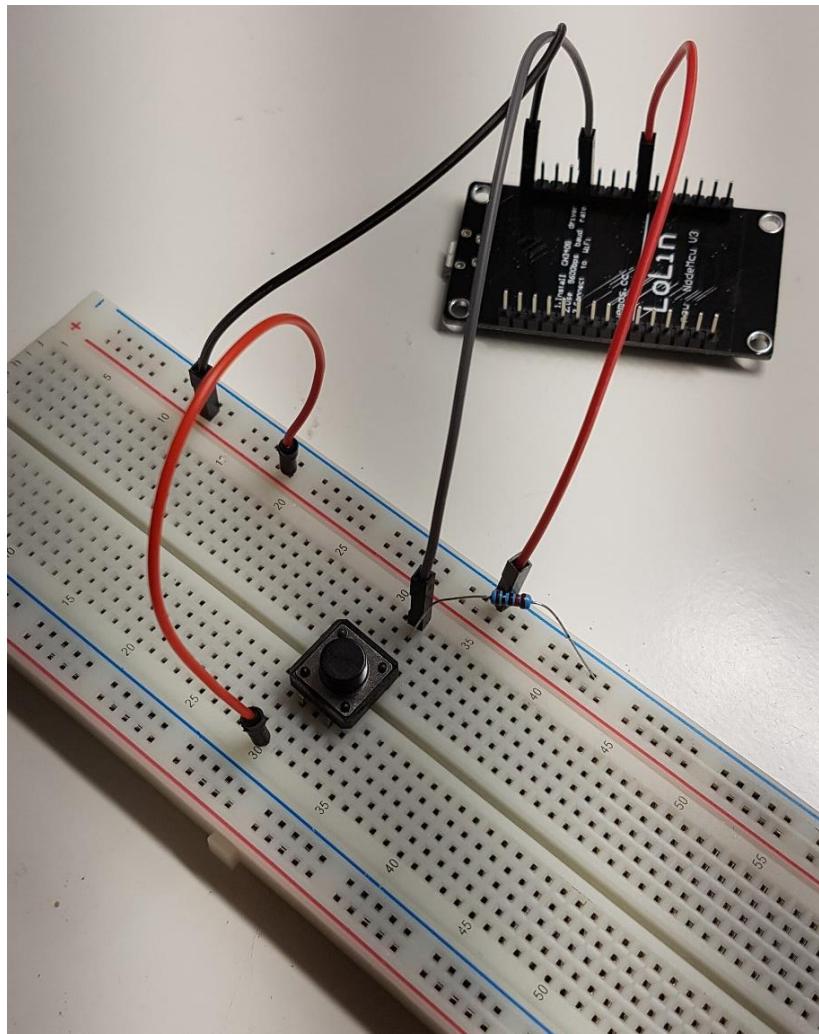


Figura 11.1: Módulo botón

11.1 Sensores y actuadores

11.1.1 Botón pulsador

Un botón pulsador (a veces denominado interruptor pulsador, o interruptor de contacto momentáneo) contiene al menos dos contactos, que se abren o se cierran cuando se pulsa el botón. Generalmente un resorte restablece el botón a su posición original cuando desaparece la presión externa (Platt, 2015, p. 33).

11.2 Conexión

La conexión (Figura 11.2 y 11.3) es muy fácil: un contacto del botón va a los 5v, y el otro a la entrada digital. Para no romperlo con la intensidad, es recomendable poner una resistencia.

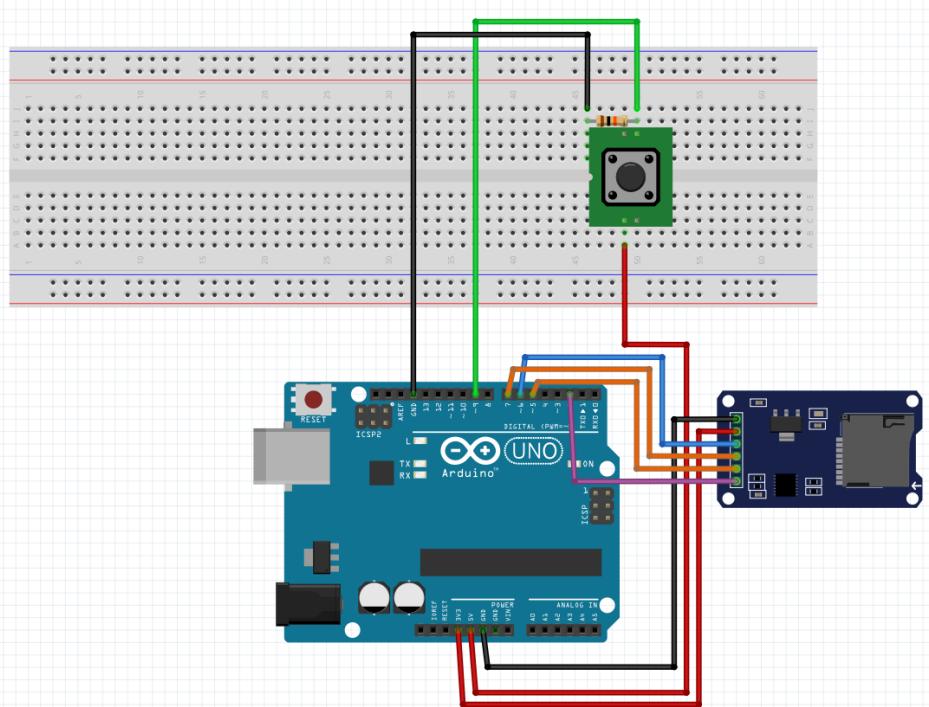


Figura 11.2: Esquema de conexión I del módulo botón

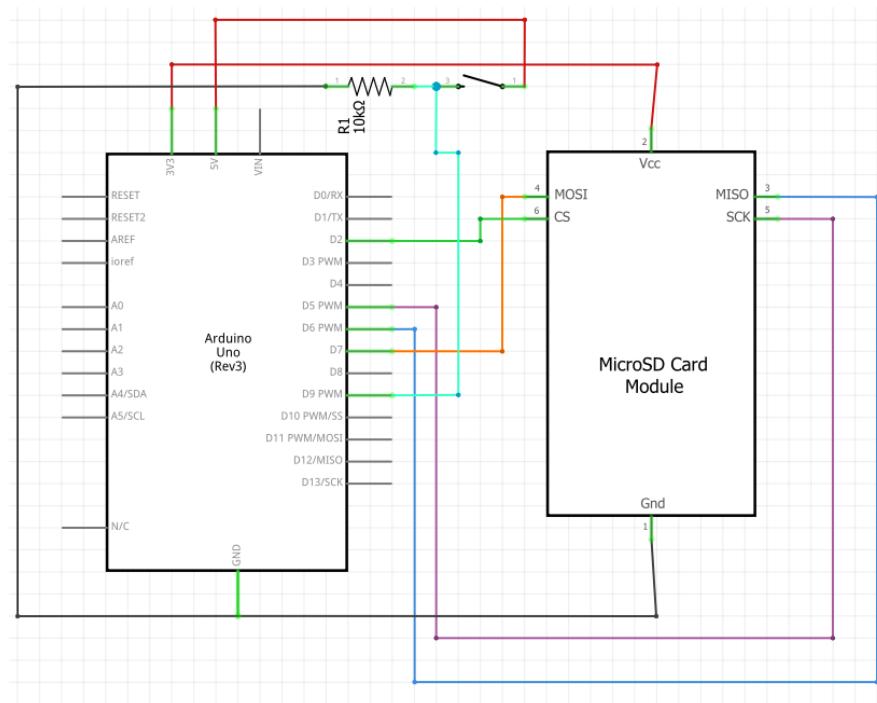


Figura 11.3: Esquema de conexión II del módulo botón

11.3 Diagrama de flujo

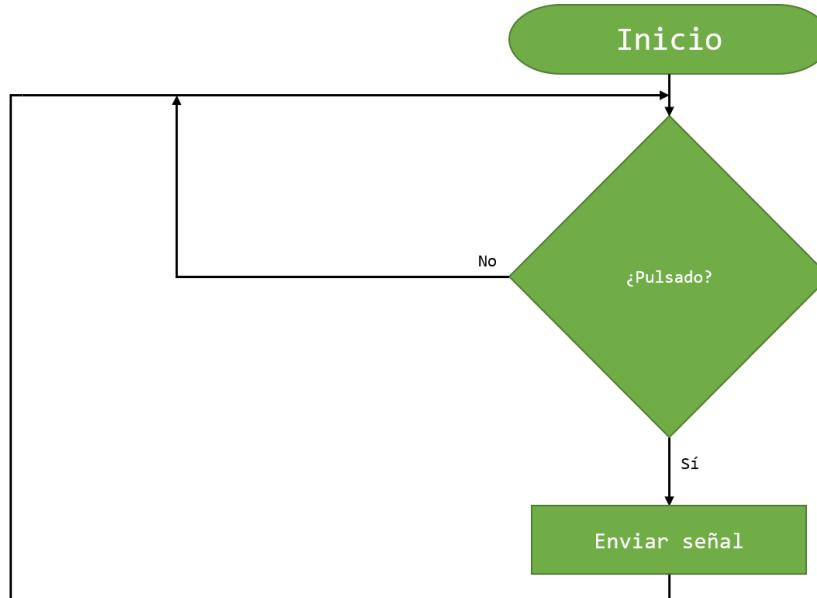


Figura 11.4: Diagrama de flujo del módulo botón

11.4 Tabla de verdad

$$a \begin{cases} 0 = \text{botón no pulsado} \\ 1 = \text{botón pulsado} \end{cases} \quad x \begin{cases} 0 = - \\ 1 = \text{enviar señal} \end{cases}$$

a	x=a
0	0
1	1

Tabla 11.1: Tabla de verdad del módulo botón

11.5 Montaje

Al ser un módulo tan básico no se ha montado definitivamente, solo se ha hecho en la protoboard (Figura 11.5).

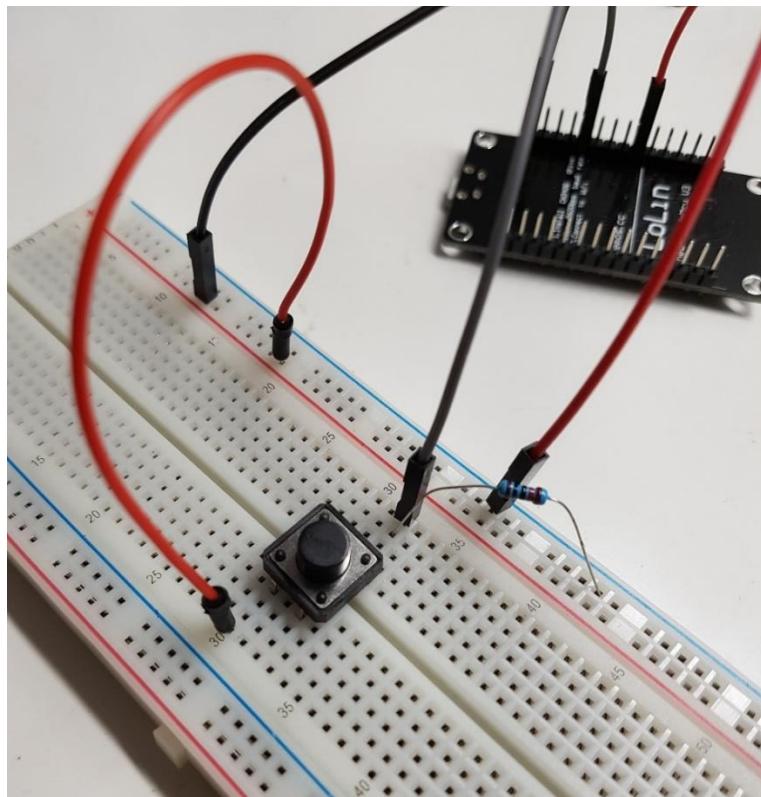


Figura 11.5: Módulo botón terminado

Capítulo 12

Módulo movimiento

El módulo movimiento tendrá como finalidad detectar el movimiento de la zona donde está, para avisar, si la alarma está activada, en el caso que haya un intruso.

Se compone de solo un sensor de movimiento (o sensor infrarrojo PIR de movimiento). Se ha añadido un LED (con su resistencia para no romperlo) para ver cuando detecta movimiento sin usar el monitor serial.

Es otro de los módulos más básicos.

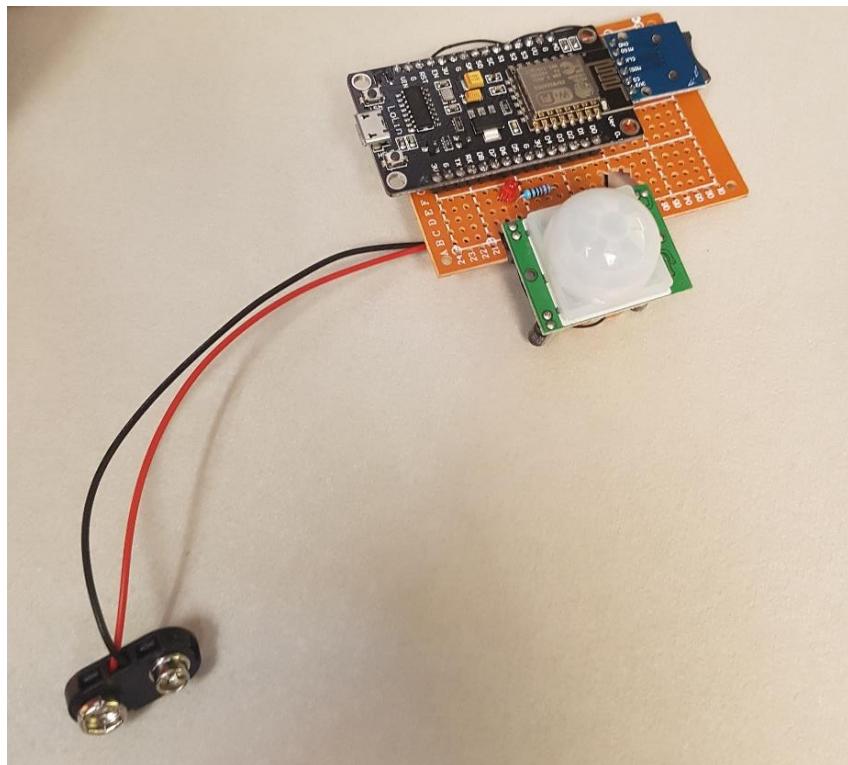


Figura 12.1: Módulo movimiento

12.1 Sensores y actuadores

12.1.1 Sensor de movimiento¹¹

Los sensores PIR (Passive InfraRed sensor) (Figura 12.2) son algo complicados porque existen múltiples variables que afectan a la entrada y salida. Para comenzar a explicar cómo funciona un sensor, se usará el ejemplo de la Figura 12.3.

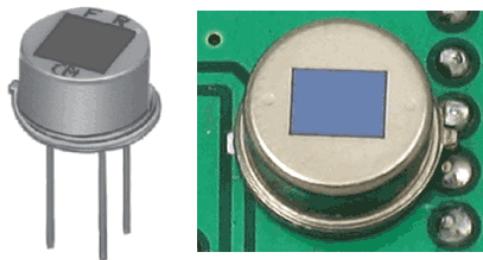


Figura 12.2: Sensor PIR

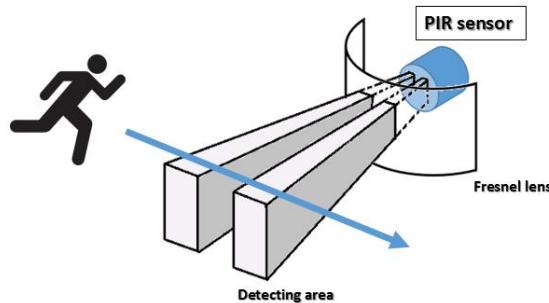


Figura 12.3: Ejemplo de sensor PIR

El sensor PIR tiene dos ranuras, cada una está hecha de un material especial que es sensible a la radiación infrarroja (IR o Infrared Radiation). Cuando el sensor está inactivo, ambas ranuras detectan la misma cantidad de IR. Cuando pasa un cuerpo caliente como un humano o un animal, primero lo intercepta la mitad del sensor PIR, lo que provoca un cambio diferencial positivo entre las dos mitades. Cuando el cuerpo caliente abandona el área de detección, ocurre lo contrario, por lo que el sensor genera un cambio diferencial negativo. Estos pulsos de cambio son lo que se detecta (Figura 12.4).

¹¹ (Sensor de movimiento para luz de escalera, 2018)

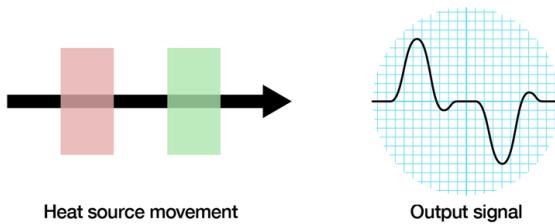


Figura 12.4: Detección del PIR

Los sensores PIR son bastante genéricos y en su mayor parte varían solo en precio y sensibilidad. La mayor parte de la magia real ocurre con la óptica, con ella se logra un área de detección más grande en una de más pequeña.

12.2 Conexión

El sensor PIR es muy fácil de conectar, como indica la Figura 12.5.

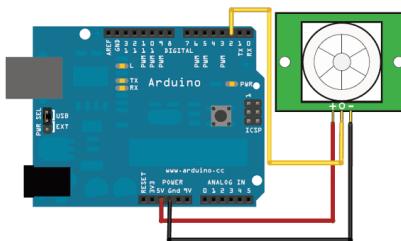


Figura 12.5: Conexión del sensor PIR

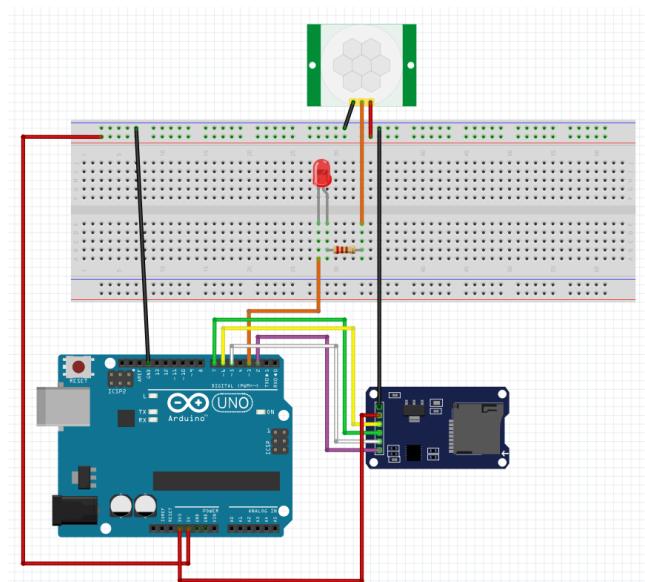


Figura 12.6: Esquema de conexión del módulo movimiento

12.3 Diagrama de flujo

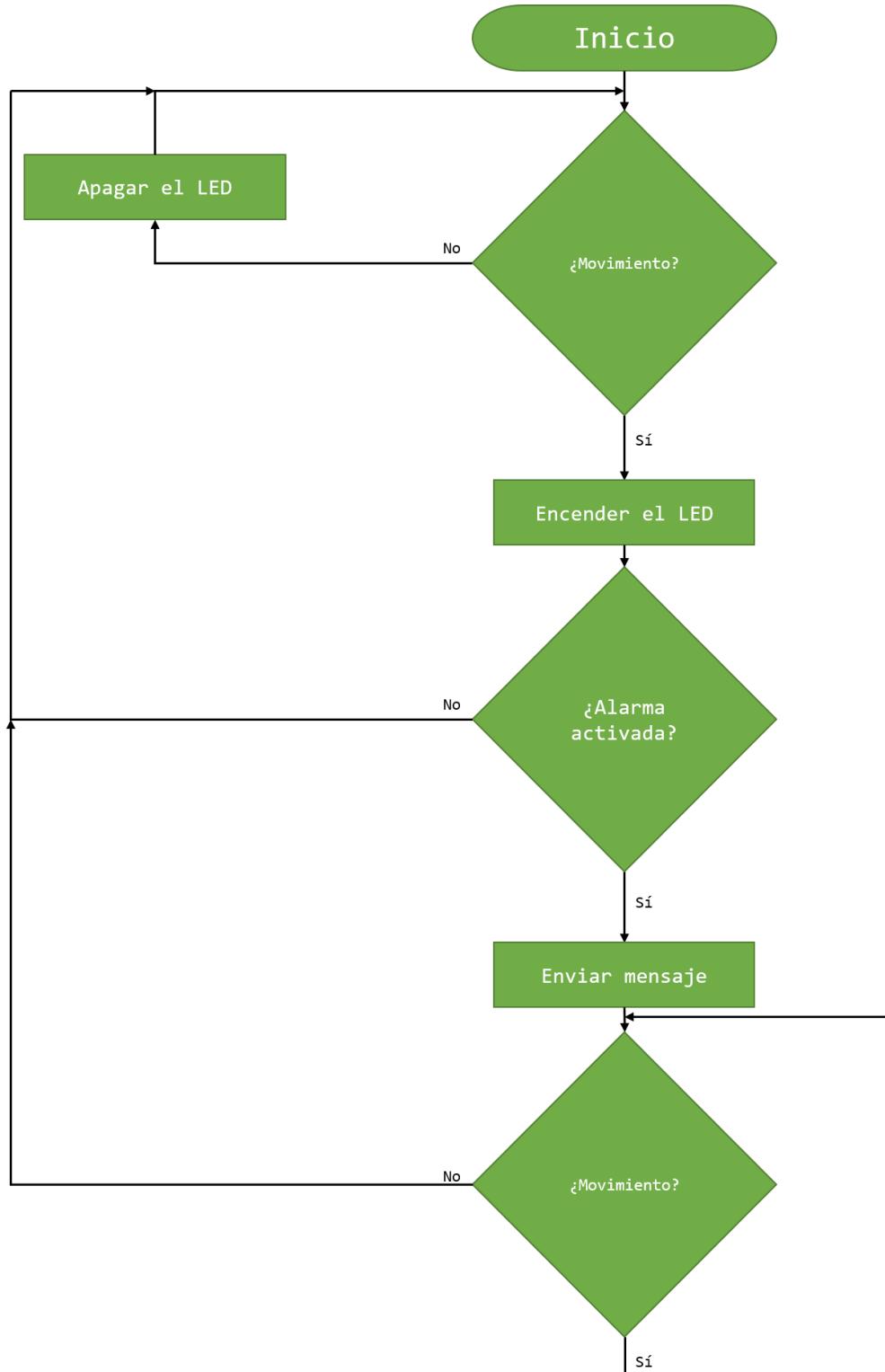


Figura 12.7: Diagrama de flujo del módulo movimiento

12.4 Tabla de verdad

$$\begin{array}{ll}
 a \left[\begin{array}{l} 0 = \text{sensor apagado} \\ 1 = \text{sensor encendido} \end{array} \right] & x \left[\begin{array}{l} 0 = \text{LED apagado} \\ 1 = \text{LED encendido} \end{array} \right] \\
 b \left[\begin{array}{l} 0 = \text{sin movimiento} \\ 1 = \text{movimiento} \end{array} \right] & y \left[\begin{array}{l} 0 = \text{no enviar datos} \\ 1 = \text{enviar datos} \end{array} \right] \\
 c \left[\begin{array}{l} 0 = \text{sin movimiento en el anterior ciclo} \\ 1 = \text{movimiento en el anterior ciclo} \end{array} \right] &
 \end{array}$$

a	b	c	x=b	y=a*b*c̄
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

Tabla 12.1: Tabla de verdad del módulo movimiento

12.5 Montaje

En las siguientes figuras se puede ver el módulo terminado (Figura 12.8), y sus conexiones (Figura 12.9).

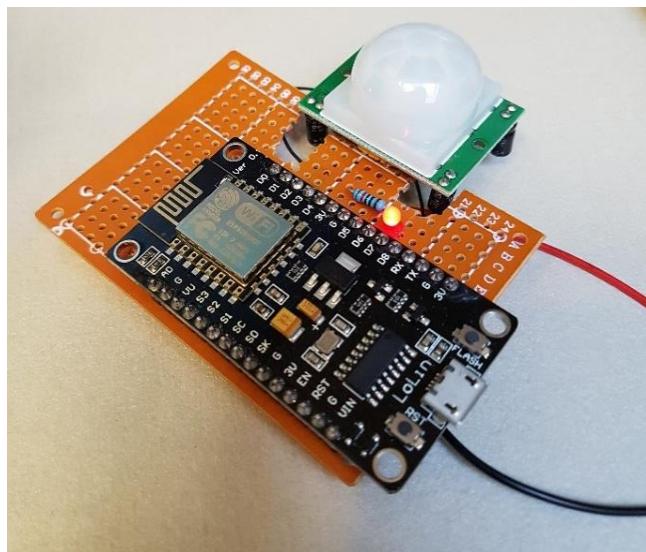


Figura 12.8: Módulo movimiento terminado

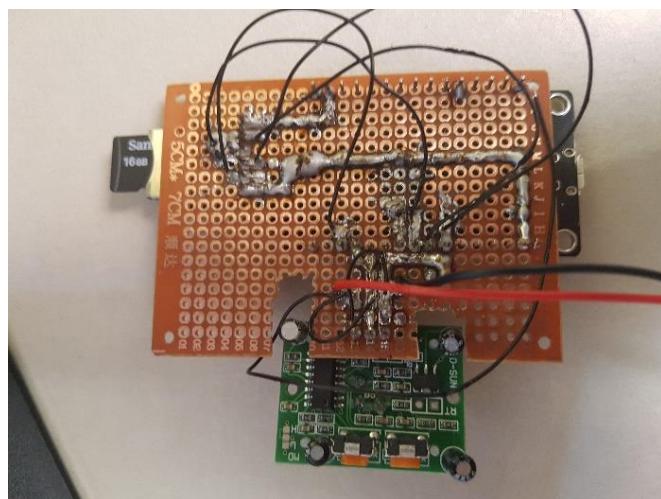


Figura 12.9: Módulo movimiento (conexiones)

Capítulo 13

Módulo apertura

El módulo apertura permitirá detectar intrusos. Lo hará mediante un sensor magnético que enviará una señal si la puerta o ventana donde está situado se abre.

Dispone de un sensor magnético y se ha añadido un LED (con su resistencia) para obtener visualmente el estado (encendido si la puerta/ventana está cerrada).

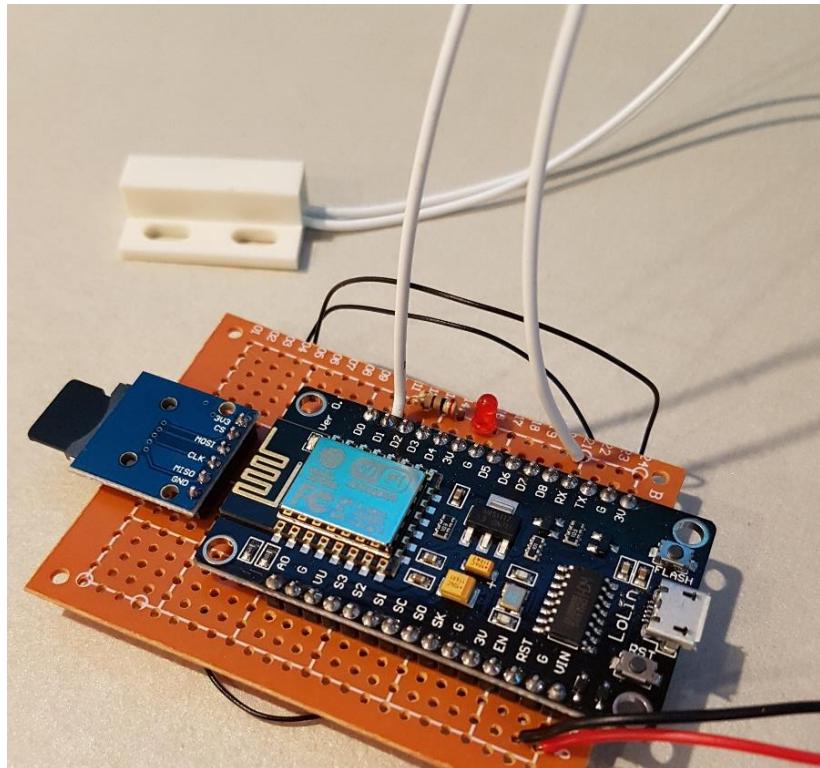


Figura 13.1: Módulo apertura

13.1 Sensores y actuadores

13.1.1 Sensor de apertura magnético

El sensor consta de dos partes (Figura 13.2): en una se encuentra un imán, y en la otra un receptor del campo magnético que genera, conectada a dos cables (+5V y entrada digital) en el que se obtiene corriente si el imán está dentro de su zona de alcance.

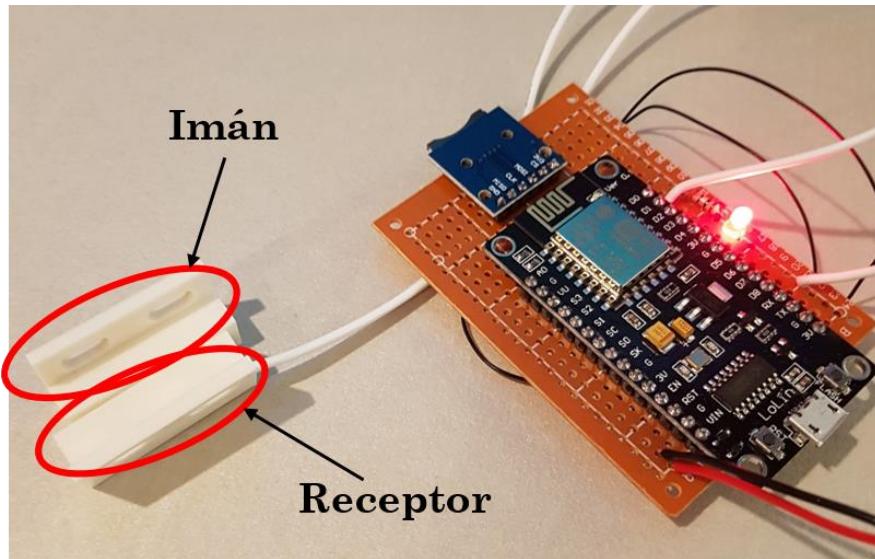


Figura 13.2: Funcionamiento del módulo apertura

Por su forma de funcionar, puede inhibirse con mucha facilidad con otro imán externo, como se muestra en la Figura 13.3 (se ha usado un imán de 90kg de fuerza a unos 8cm). No obstante, por su tamaño (puede ocultarse con gran facilidad), precio y facilidad de conexión y programación se ha decidido usar en este proyecto a pesar de esa desventaja.

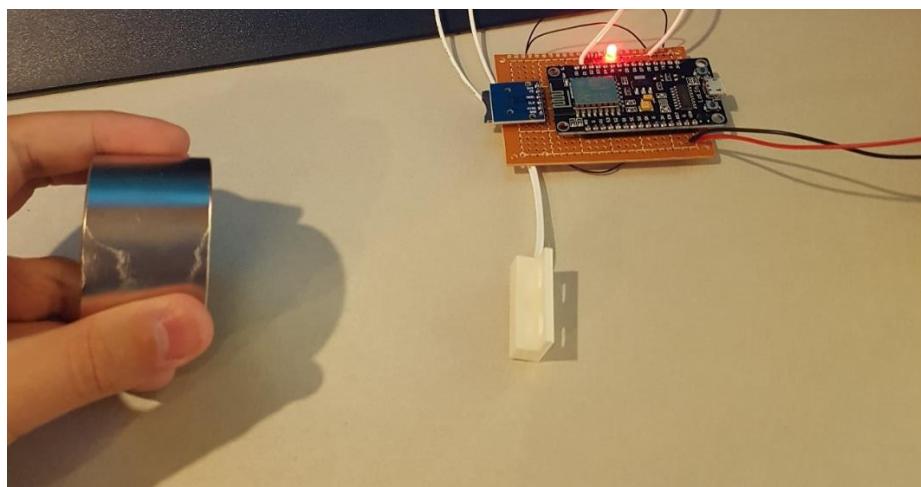


Figura 13.3: Inhibición del módulo apertura

13.2 Conexión

La conexión del sensor magnético es muy simple; un cable al positivo, y el otro al receptor. Es muy aconsejable no conectarlo directamente, ya sea conectando una resistencia de $10\text{k}\Omega$ en paralelo y a tierra, o en este caso, un LED que además produce una información visual.

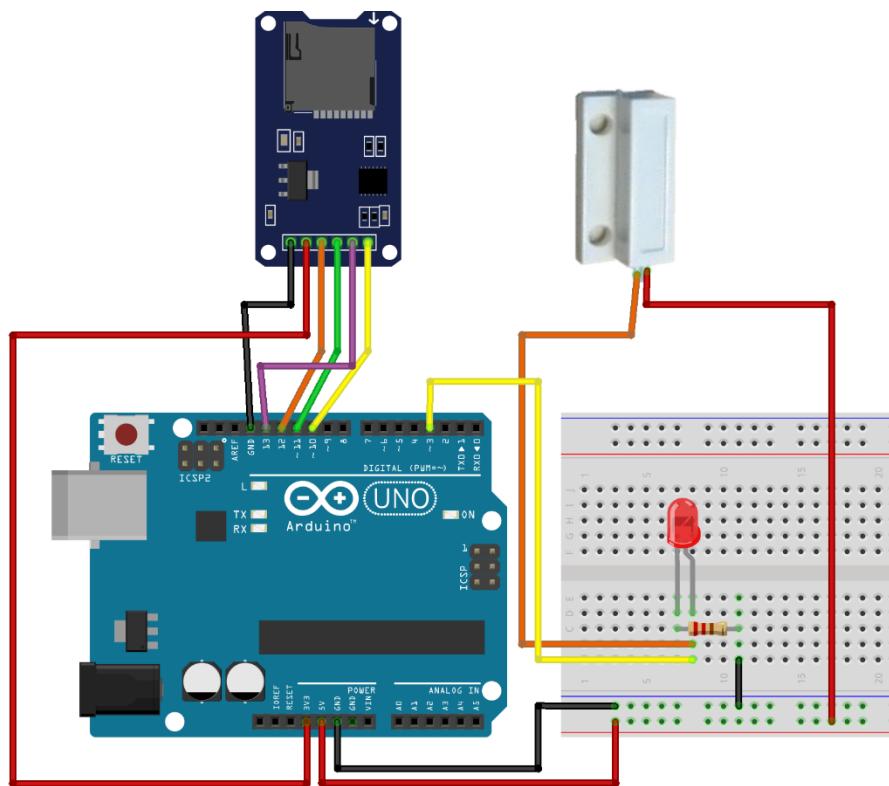


Figura 13.4: Esquema de conexión del módulo apertura

13.3 Diagrama de flujo

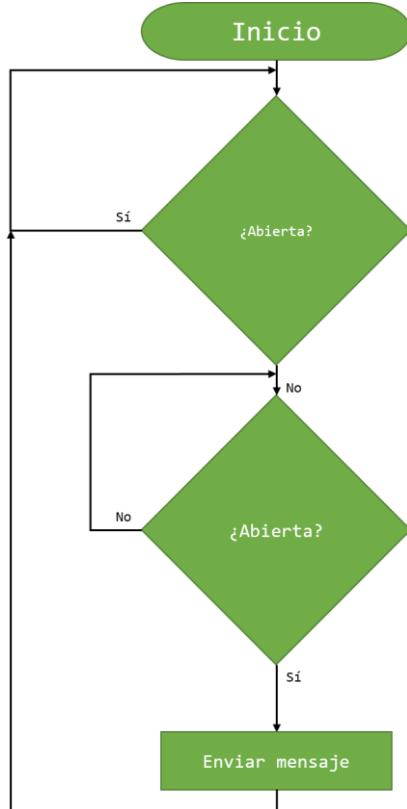


Figura 13.5: Diagrama de flujo del módulo apertura

13.4 Tabla de verdad

$$a \begin{cases} 0 = \text{cerrado} \\ 1 = \text{abierto} \end{cases}$$

$$x \begin{cases} 0 = - \\ 1 = \text{enviar alerta} \end{cases}$$

$$b \begin{cases} 0 = \text{antes, cerrado} \\ 1 = \text{antes, abierto} \end{cases}$$

a	b	x = $\bar{b} * a$
0	0	0
0	1	0
1	0	1
1	1	0

Tabla 13.1: Tabla de verdad del módulo apertura

13.5 Montaje

En las siguientes imágenes se puede ver el prototipo (Figura 13.6), y el módulo terminado (Figura 13.7).

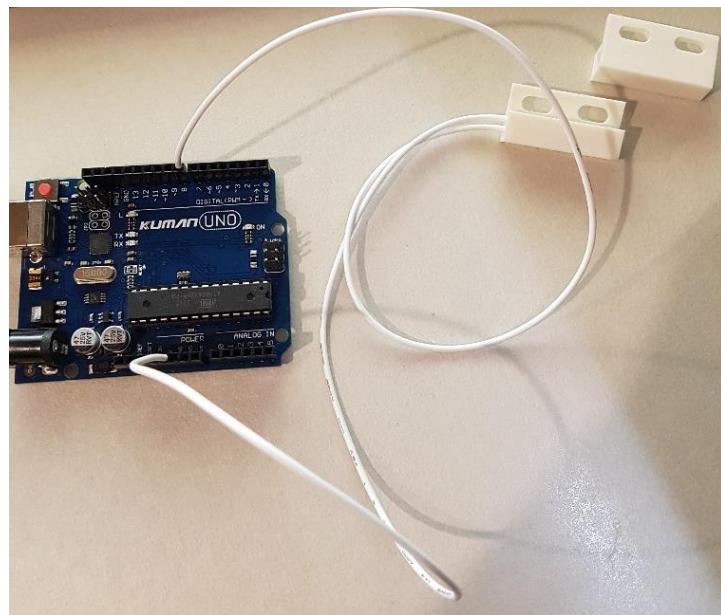


Figura 13.6: Prototipo del módulo apertura

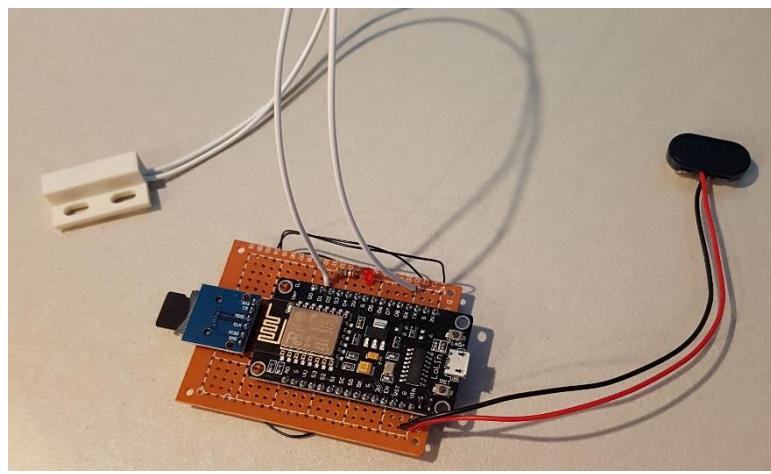


Figura 13.7: Módulo apertura terminado

Capítulo 14

Módulo seguridad

El módulo seguridad detectará algunos gases que son tóxicos en humanos, incluso mortales. También dispone de un buzzer para advertir si alguno de estos gases supera el límite establecido.

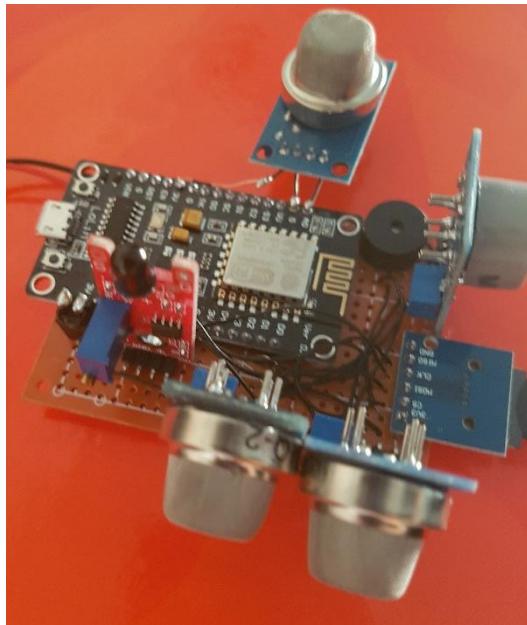


Figura 14.1: Módulo seguridad

14.1 Sensores y actuadores

14.1.1 Buzzer

Un buzzer convierte corriente eléctrica en sonido. Mediante un electroimán y una lámina metálica de acero logra crear un zumbido de un mismo tono (conectándolo a circuitos integrados especiales pueden crearse distintos tonos) (Zumbador, 2018).



Figura 14.2: Buzzer

14.1.2 Sensor de llama¹²

Un sensor de llama es un dispositivo que permite detectar la existencia de combustión por la luz emitida por la misma. Esta luz puede ser detectada por un sensor óptico, y ser capturado por las entradas digitales y las entradas analógicas de Arduino.

La llama es un fenómeno de emisión de luz asociado a los procesos de combustión. La combustión es un proceso que desprende grandes cantidades de energía en forma de calor. Durante la reacción se generan compuestos intermedios que liberan parte de su energía mediante la emisión de luz.

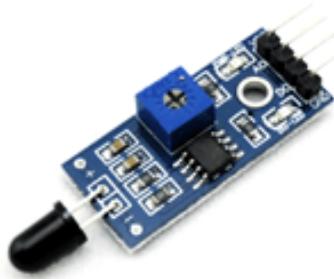


Figura 14.3: Sensor de llama

¹² (Llamas, 2016)

14.1.3 Sensores de gas

Funcionan mediante reacciones químicas. Se han usado 4 tipos distintos:

» *MQ2*

Puede detectar GLP¹³, propano, metano, alcohol, hidrógeno y humo. Siendo más sensible al GLP y propano.

» *MQ5*

Este es un sensor muy sencillo de usar para el GLP.

» *MQ7*

Este sensor es de alta sensibilidad al monóxido de carbono (CO), pero también es sensible al H₂.

» *MQ135*

Se utiliza en equipos de control de calidad del aire para edificios y oficinas, adecuado para la detección de NH₃, NO_x, alcohol, benceno, humo, CO₂...

14.2 Conexión

La conexión de este módulo es simple, pero al tener tantos sensores se complica un poco.

¹³ Gas licuado de petróleo

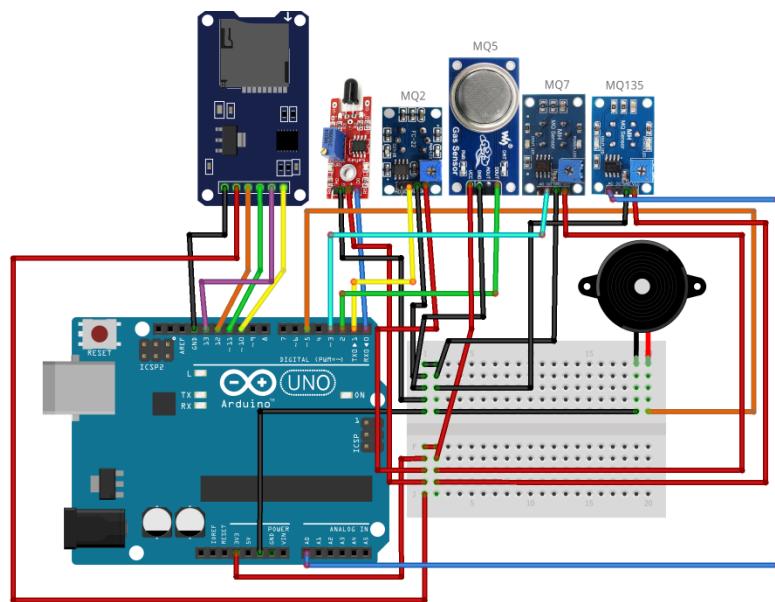


Figura 14.4: Esquema de conexión del módulo seguridad

14.3 Diagrama de flujo

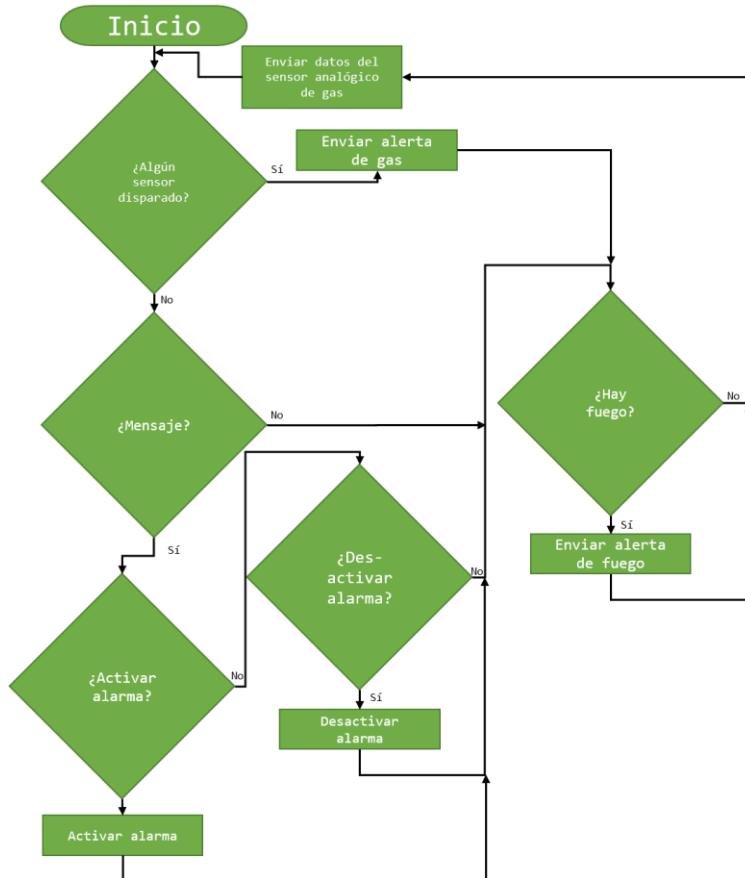


Figura 14.5: Diagrama de flujo del módulo seguridad

14.4 Tabla de verdad

$$\begin{aligned}
 a & \left[\begin{array}{l} 0 = \text{no hay fuego} \\ 1 = \text{hay fuego} \end{array} \right. & x & \left[\begin{array}{l} 0 = - \\ 1 = \text{enviar sensores de gas} \end{array} \right. \\
 b & \left[\begin{array}{l} 0 = - \\ 1 = \text{activar alarma} \end{array} \right. & y & \left[\begin{array}{l} 0 = - \\ 1 = \text{enviar alerta fuego} \end{array} \right. \\
 z & \left[\begin{array}{l} 0 = - \\ 1 = \text{activar alarma} \end{array} \right.
 \end{aligned}$$

a	b	x=1	y=a	z=b
0	0	1	0	0
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1

Tabla 20.1: Tabla de verdad del módulo seguridad

14.5 Montaje

La placa de pruebas ha quedado totalmente inundada de sensores.



Figura 14.6: Módulo seguridad

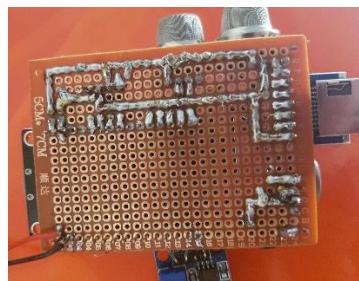


Figura 14.7: Conexiones del módulo seguridad

Capítulo 15

Módulo climático simple

El módulo climático simple enviará información sobre la temperatura, humedad y luz. Esto será posible gracias a los diversos sensores que tiene incorporado.

Los sensores son económicos y dan valores bastante precisos.

Este módulo está pensado para estar dentro de casa, ya que para el exterior se ha creado el módulo climático avanzado, con pluviómetro y una carcasa creada con impresora 3D.

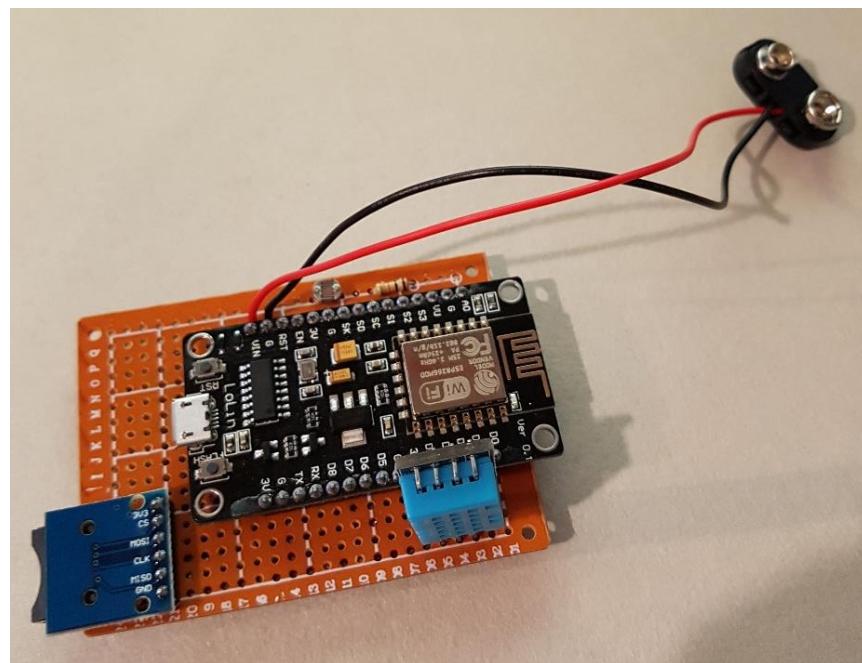


Figura 15.1: Módulo climático simple

15.1 Sensores y actuadores

15.1.1 Fotoresistor

Un fotoresistor es una resistencia de tipo variable que varía en función de la luz que le llega. Emitiendo un voltaje y leyendo el voltaje de salida, mediante una serie de cálculos, se puede obtener el % de luz.

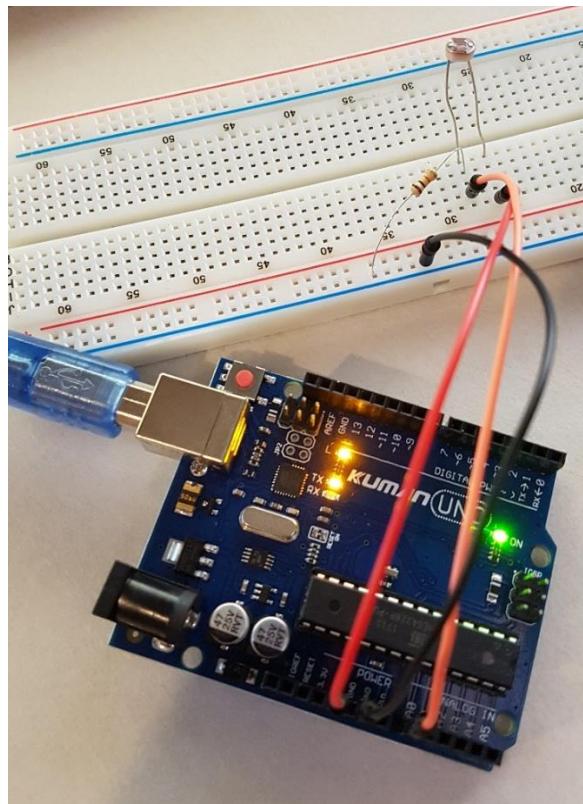


Figura 15.2: Conexión de un fotoresistor

15.1.2 Sensor de humedad y temperatura¹⁴

El DHT11 (Figura 15.3) puede obtener valores de un 20 a un 90% de humedad con una precisión del $\pm 5\%$ y de 0 a 50°C, con una precisión del $\pm 2\%$ °C.

¹⁴ (Hernández, 2018)

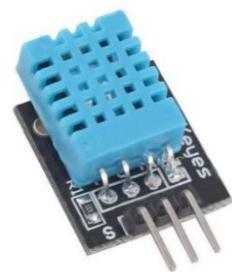


Figura 15.3: DHT11

El módulo realiza una conversión de analógico a digital. Como lo hace puede verse en la Figura 15.4.

0011 0101 0000 0000 0001 1000 0000 0000 0100 1001
8 bits humedad **8 bits humedad** **8 bits temperatura** **8 bits temperatura** **bits de paridad**

Figura 15.4: Conversión analógico-digital del DHT11

Así se obtienen los valores de temperatura y humedad (con decimales) mediante un pin digital.

Los bits de paridad se aseguran que la información sea correcta. Se obtiene sumando los 4 primeros grupos de 8 bits.

15.2 Conexión

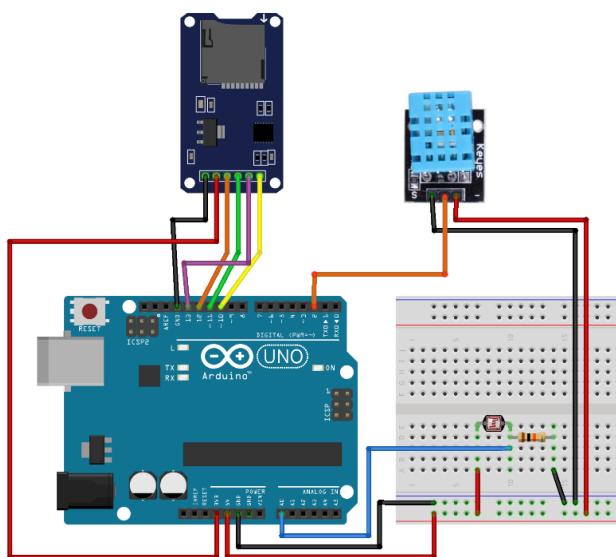


Figura 15.5: Esquema de conexión del módulo climático simple

15.3 Diagrama de flujo

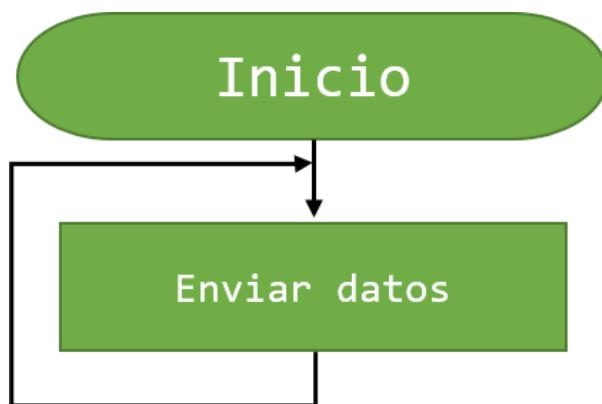


Figura 15.6: Diagrama de flujo del módulo climático simple

15.4 Tabla de verdad

No hace falta crear una tabla de verdad en esta situación, ya que no hay condición alguna, siempre se envían los datos.

Capítulo 16

Módulo climático avanzado

El módulo climático avanzado enviará información sobre la temperatura, humedad, nivel del agua y luz. Esto será posible gracias a los diversos sensores que tiene incorporado y del modelo 3D (en caso del pluviómetro).

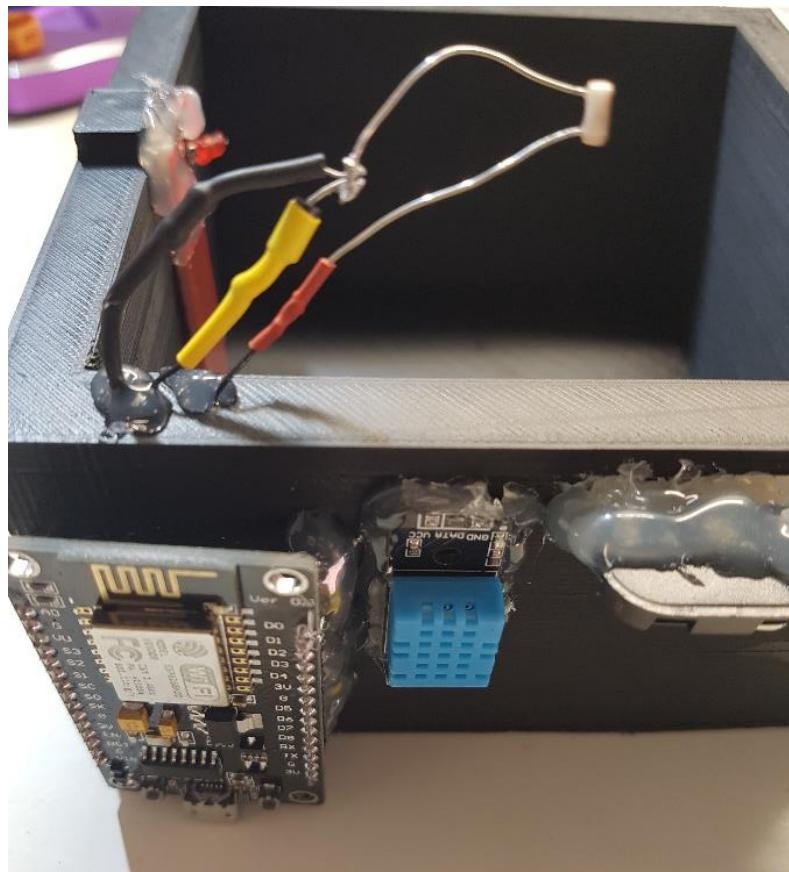


Figura 16.1: Módulo climático avanzado

16.1 Sensores y actuadores

16.1.1 Fotoresistor

El fotoresistor se explica en la subsección 16.1.1.

16.1.2 Sensor de humedad y temperatura

El sensor de humedad y temperatura se explica en la subsección 16.1.2.

16.1.3 Pluviómetro

El pluviómetro (Figura 16.2) es un instrumento que se emplea en las estaciones meteorológicas para la recogida y medición de la precipitación. Se usa para medir la cantidad de precipitaciones caídas en un lugar durante un tiempo determinado (Pluviómetro, 2018).



Figura 16.2: Pluviómetro

La cantidad de agua caída se expresa en milímetros de altura (o equivalentemente en litros por metro cuadrado).

Existen módulos de Arduino que miden la altura del agua (Figura 16.3). Estos sensores varían el voltaje de salida dependiendo de la altura del agua.



Figura 16.3: Sensor de agua

Este componente sería súper fácil de fabricar, el problema está en la resistividad del agua, al ser extremadamente elevada (aproximadamente de $200\Omega \cdot m$) el valor de salida sería demasiado bajo. La solución, está en conectar un transistor, funcionando como amplificador.

Se agradece a Indalecio13 por ayudar en Foros de Electrónica a corregir el esquema original, este sería el definitivo:

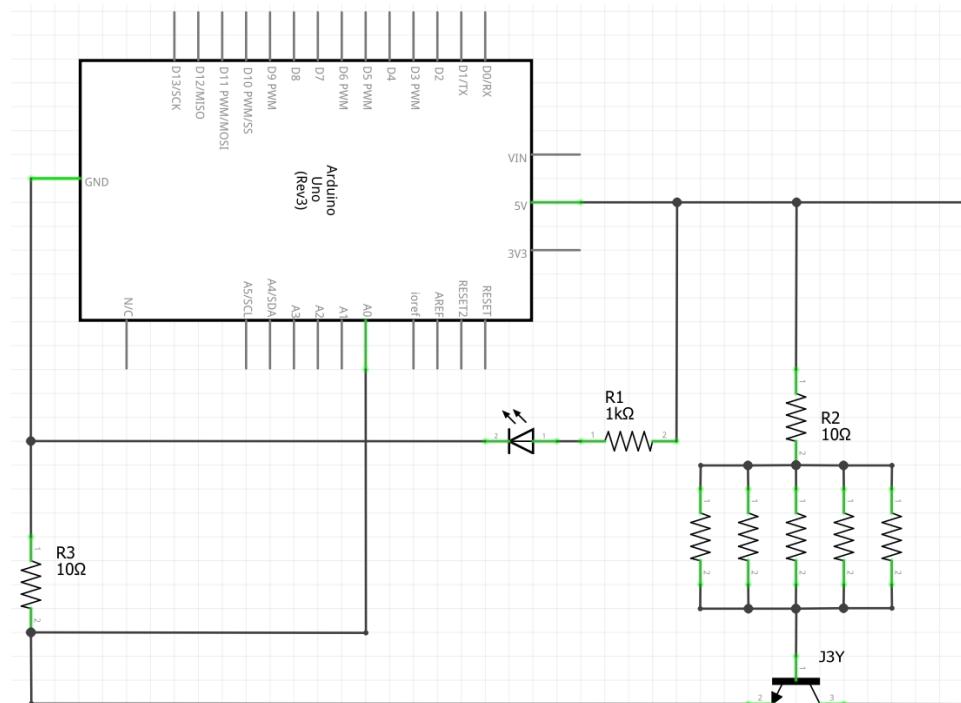


Figura 16.4: Esquema del sensor de agua para Arduino

Viendo el esquema, podemos entender por qué obtenemos voltaje cuando el agua pasa (se aplica voltaje a la base del transistor, dejando paso al colector, leyendo así el voltaje que pasa por R_3).

Las 5 resistencias colocadas en paralelo hacen que el componente haga lo que ha de hacer. Como ya se ha mencionado la resistencia del agua es muy elevada, si se conectaran tan solo dos tiras de bronce (una envía electrones, y la otra los recibe) la variación de voltaje sería mínima. Al conectar 5 en paralelo, la resistencia equivalente es igual a la resistencia que ofrece el agua dividida entre 5.

$$R_T = \frac{R_{Agua}}{5}$$

La R_T no es la resistencia total del circuito, sino del grupo de tiras de cobre.

Mediante valores tomados experimentalmente puede crearse una gráfica (Figura 16.5) que permita convertir de un valor a un rango de altura del agua.

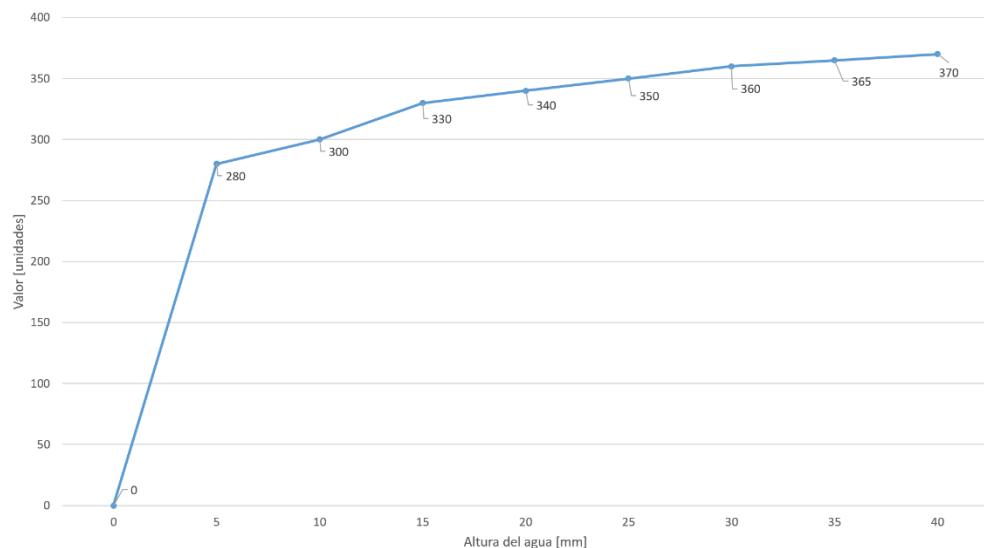


Figura 16.5: Gráfica de valores en función de la altura del agua

Con estos datos solo falta un recipiente donde almacenar el agua a medida que llueve, en el caso de este módulo tiene uno incluido en el modelo 3D. Con una dimensión de 0.6 dm^3 (0.4 dm^3 útiles) proporciona más de medio litro de almacenamiento. Mediante un sencillo cálculo pueden obtenerse los litros por metro cuadrado de agua precipitada.

16.2 Conexión

La conexión es sencilla, pero al disponer de solo una entrada analógica (la única que ofrece la placa NodeMCU) se ha tenido que crear un sistema para adaptarlo. Es muy sencillo, solo hay que suministrar corriente a uno de los dos sensores y leer la entrada analógica.

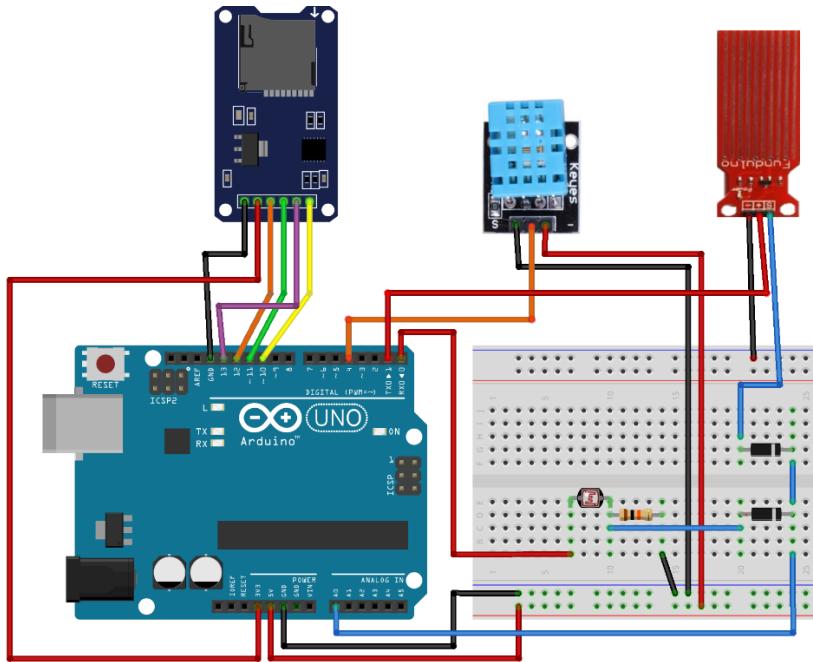


Figura 16.6: Esquema de conexión del módulo climático avanzado

16.3 Diagrama de flujo

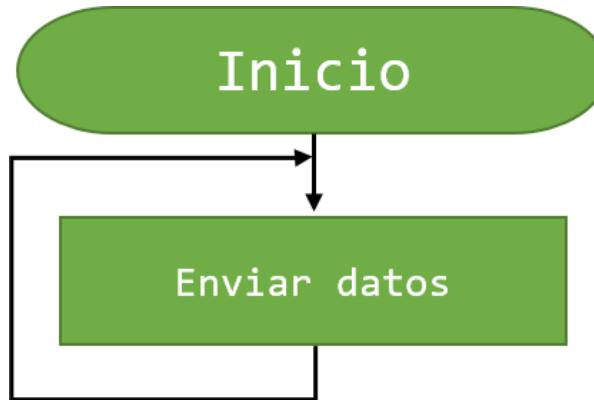


Figura 16.7: Diagrama de flujo del módulo climático avanzado

16.4 Tabla de verdad

Al ser un módulo que envía datos siempre no hace falta una tabla de verdad.

16.5 Montaje

Uno de los módulos que se ha empleado más tiempo en el montaje. Al ir todos los cables por el interior del modelo 3D ha complicado la tarea.

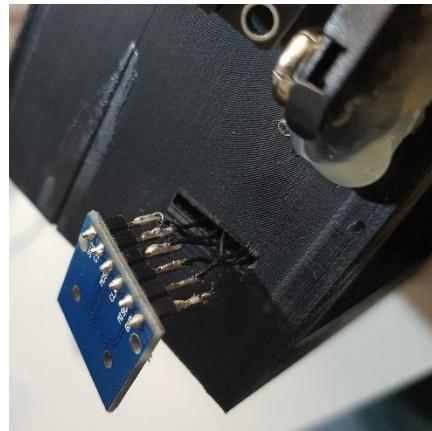


Figura 16.8: Montaje de la entrada MicroSD

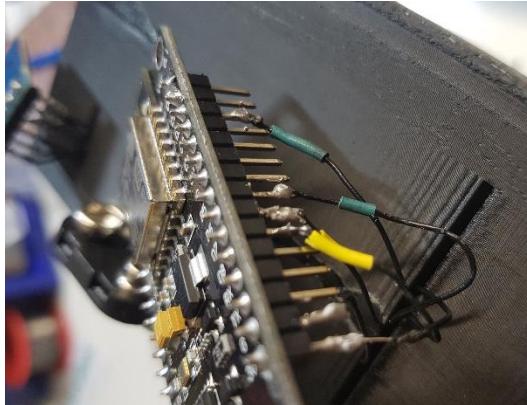


Figura 16.9: Placa con varios sensores instalados

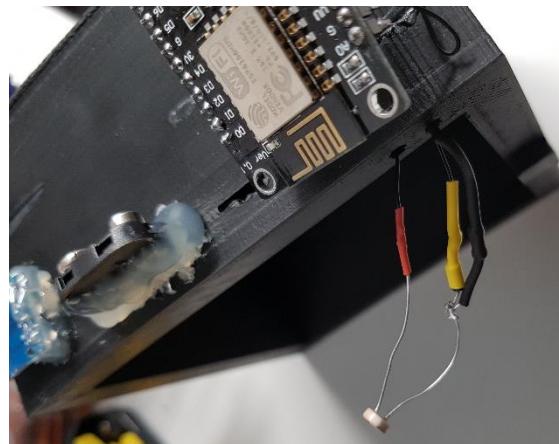


Figura 16.10: Fotoresistor



Figura 16.11: Sensor de altura del agua

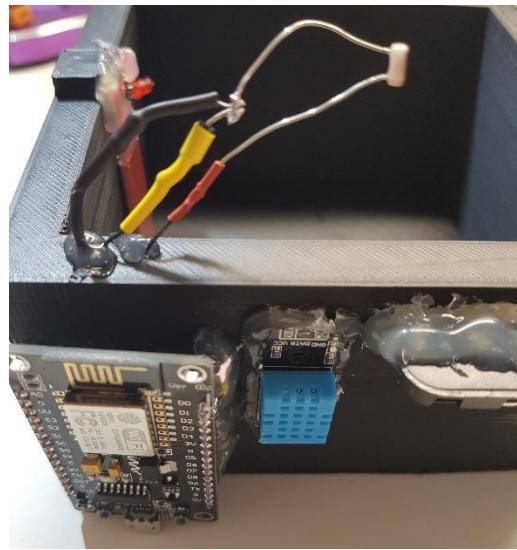


Figura 16.12: Módulo climático avanzado terminado

Capítulo 17

Módulo alarma

El módulo alarma tiene como finalidad activar y desactivar la alarma, ya sea mediante un código introducido en el keypad, o mediante el sensor dactilar. Se ha creado un recipiente con impresora 3D para que sea más elegante y práctico.



Figura 17.1: Módulo alarma

17.1 Sensores y actuadores

17.1.1 Buzzer

El buzzer se explica en la subsección 17.1.1.

17.1.2 4-Digit Display

Este actuador (Figura 17.2) puede mostrar un máximo de 4 dígitos, ideal para este módulo. Usa un microcontrolador (TM1637) para poder controlarse mediante tan solo 2 pines.

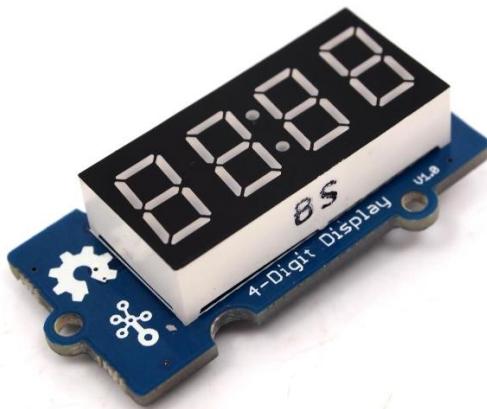


Figura 17.2: 4-Digit Display

Se podría haber usado una pantalla LCD¹⁵ (Figura 17.3) pero son un poco más caras.



Figura 17.3: Pantalla LCD para Arduino

¹⁵ Liquid Crystal Display

17.1.3 Lector de huellas dactilares GT-521F52

Como su nombre indica, es un lector de huellas dactilares (Figura 17.4). Permite almacenar hasta 200 huellas.



Figura 17.4: Lector de huellas dactilares

17.1.4 4x4 Keypad

El módulo 4x4 keypad para Arduino (Figura 17.5) consta de 16 botones puestos de forma estratégica para ahorrar entradas. Para implementar 16 botones en Arduino sería necesario 1 entrada de 5v, y 16 salidas. Obviamente esto no resulta práctico, así que idearon una forma para hacer lo mismo, empleando menos pines (Figura 17.6).



Figura 17.5: 4x4 Keypad

El 4x4 keypad tiene 8 salidas, una salida para cada fila, y una para cada columna. Conectar los pines de las filas a 5v y los pines de las columnas a GND (leyendo el resultado) sería un problema, ya que si se pulsa el botón 6, se leería 5v en el pin 2. Eso significaría que hay 1 (o más) botones pulsados en la fila del pin 2, pero no cuales. La solución está en alimentar fila por fila.

El código para hacer eso resulta fácil de programar y no habría problemas, pero existen librerías que ya lo tienen implementado.

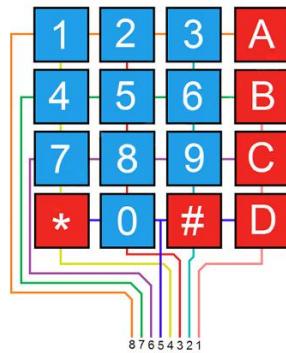


Figura 17.6: Conexión de los botones

17.2 Conexión

Este módulo usa casi todas las entradas de Arduino, evitando (como se pretendía des de un principio) la posibilidad de añadir una opción extra, un lector RFID¹⁶.

Además, el lector de huellas funciona con un voltaje de 3.3V (también en la comunicación), así que ha sido necesario construir un reductor de voltaje.

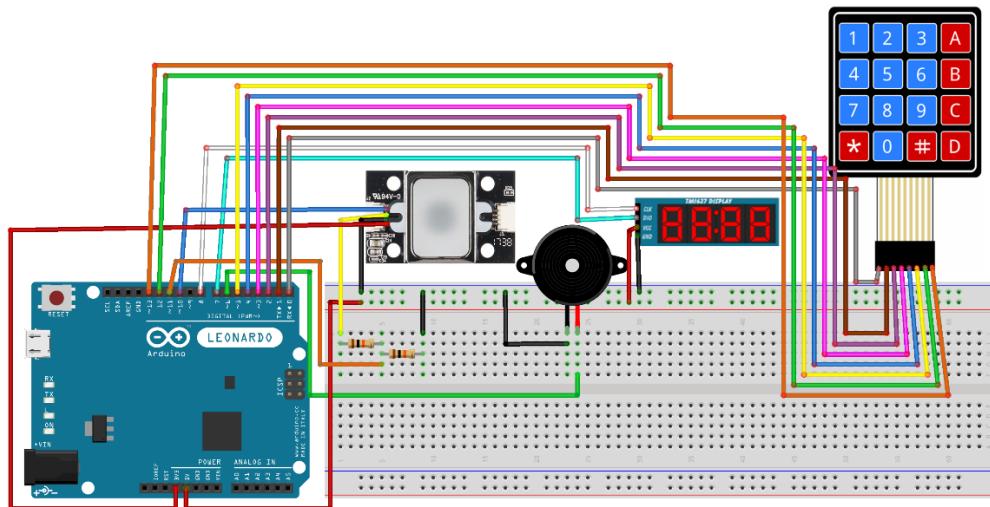


Figura 17.7: Esquema de conexión del módulo alarma

¹⁶ Identificación por radiofrecuencia. Hay componentes de Arduino que cuentan con esta posibilidad (normalmente mediante etiquetas pasivas, que no requieren corriente).

17.3 Diagrama de flujo

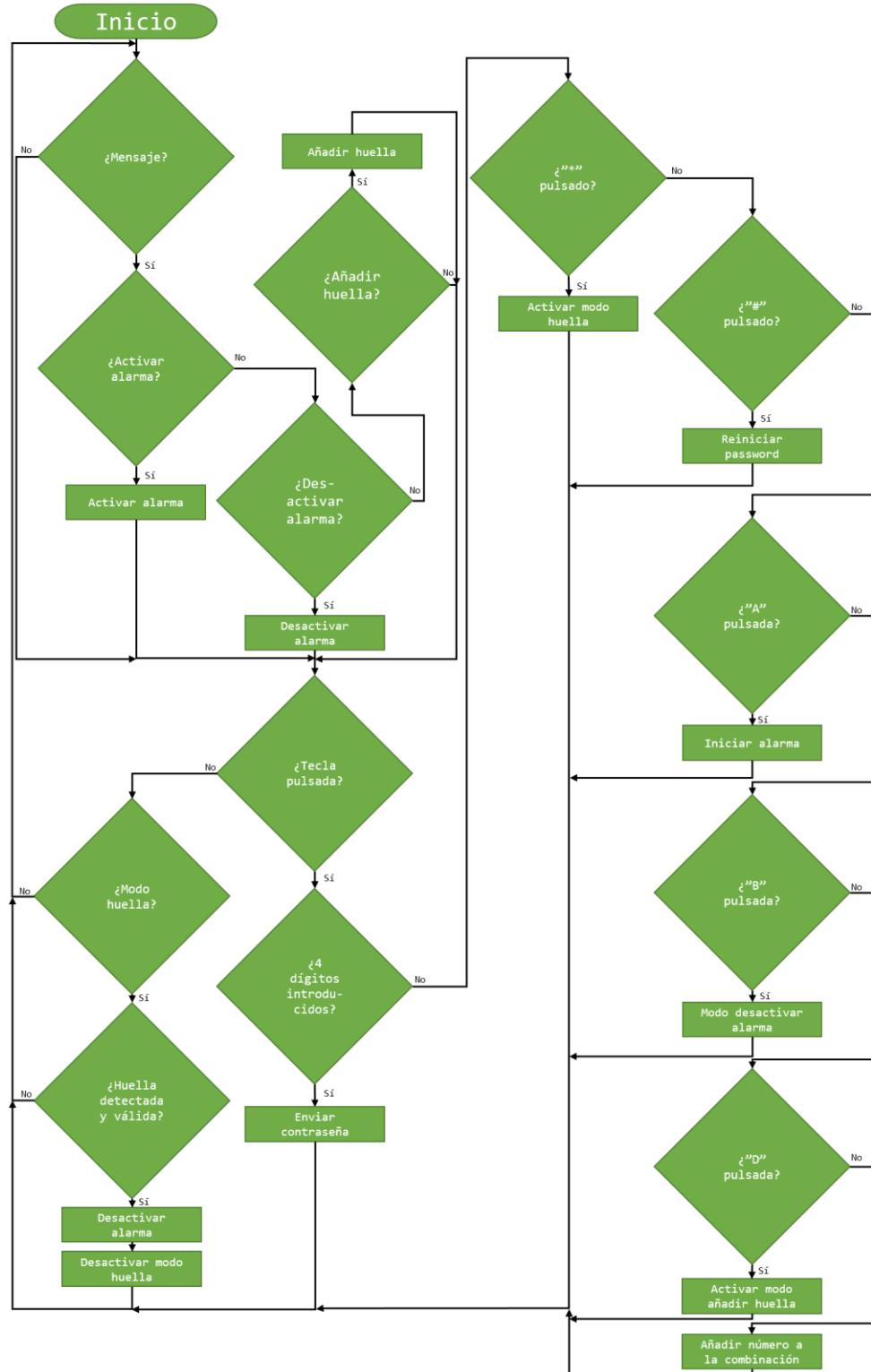


Figura 17.8: Diagrama de flujo del módulo alarma

17.4 Tabla de verdad

a	$\begin{cases} 0 = \text{menos de 4 dígitos} \\ 1 = 4 \text{ dígitos} \end{cases}$	x	$\begin{cases} 0 = - \\ 1 = \text{desactivar alarma} \end{cases}$
b	$\begin{cases} 0 = \text{contraseña incorrecta} \\ 1 = \text{contraseña correcta} \end{cases}$	y	$\begin{cases} 0 = - \\ 1 = \text{activar alarma} \end{cases}$
c	$\begin{cases} 0 = \text{sin huella} \\ 1 = \text{huella insertada} \end{cases}$		
d	$\begin{cases} 0 = \text{huella incorrecta} \\ 1 = \text{huella correcta} \end{cases}$		

a	b	c	d	$x=(c*d)+(a*b)$	$y=a*\bar{b}$
0	0	0	0	0	0
0	0	0	1	X	X
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	X	X
0	1	0	1	X	X
0	1	1	0	X	X
0	1	1	1	X	X
1	0	0	0	0	1
1	0	0	1	X	X
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	1	0
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

Tabla 17.1: Tabla de verdad del módulo alarma

17.5 Montaje

Uno de los mejores módulos, con un resultado muy aceptable.

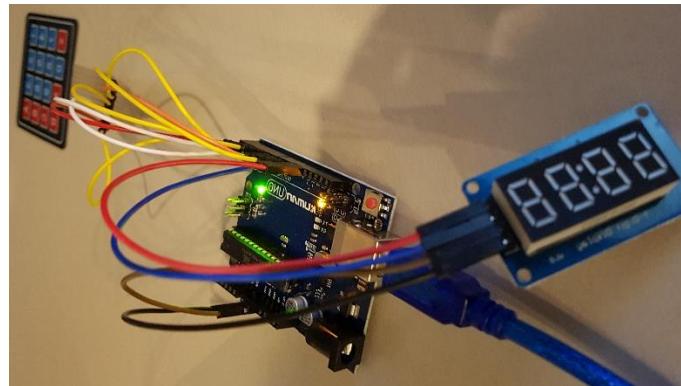


Figura 17.9: 4x4 Keypad y 4-Digit Display

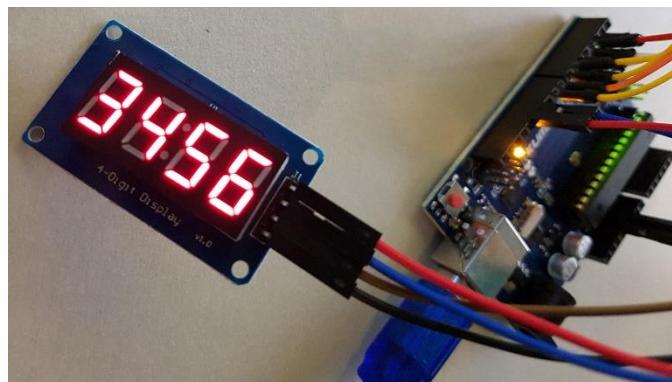


Figura 17.10: Prueba de código

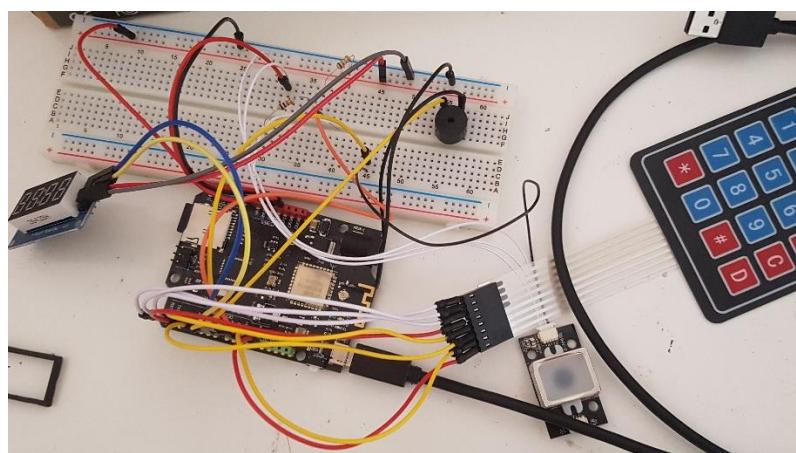


Figura 17.11: Prototipo del módulo alarma

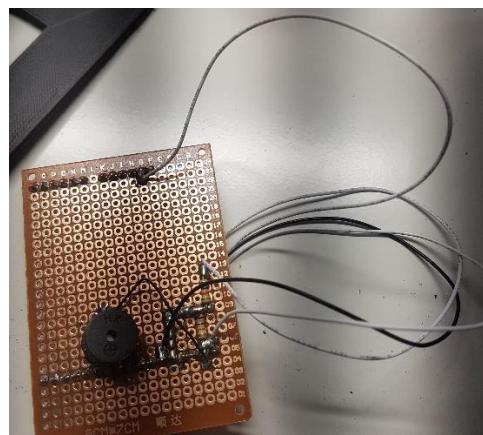


Figura 17.12: Shield creado para el módulo alarma



Figura 17.13: Placa conectada al voltaje, dentro del diseño 3D

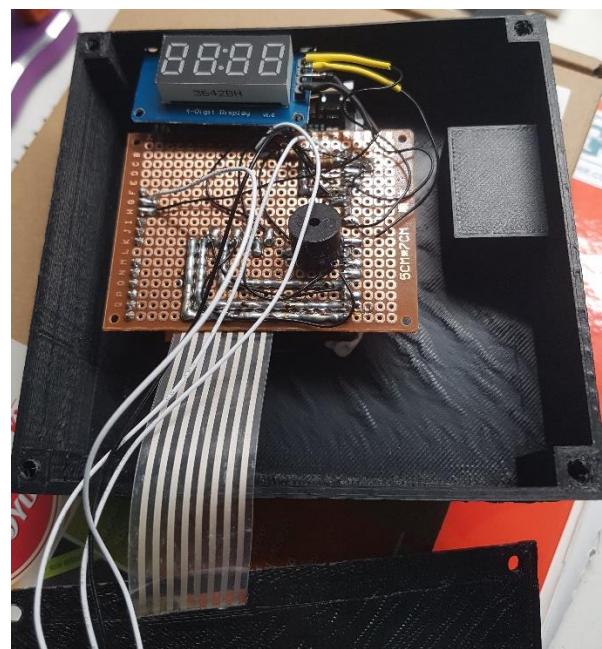


Figura 17.14: Módulo alarma (interior)

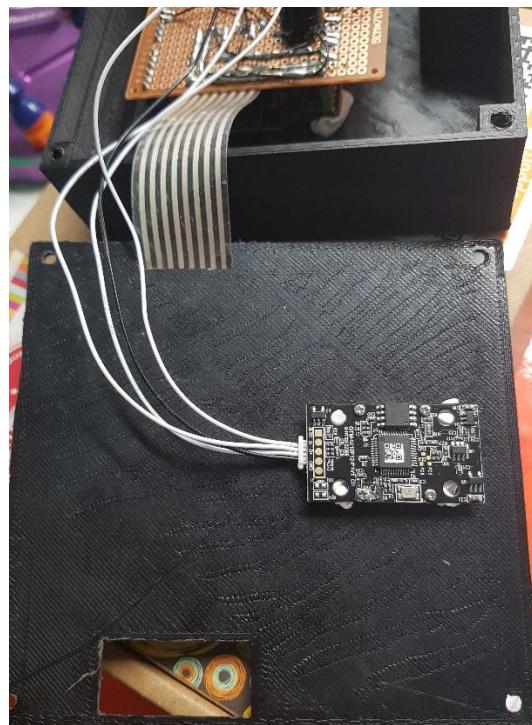


Figura 17.15: Sensor de huellas (interior)



Figura 17.16: Módulo alarma terminado

Capítulo 18

Módulo enchufe

Permite el paso de los electrones, o no, dependiendo de su estado.

Este es un módulo peligroso ya que trabaja con voltaje elevado (en Europa 230V, CA¹⁷) y un error podría ser grave. Por ese motivo se ha creado una carcasa protectora en impresora 3D, para minimizar el contacto con el cableado.

Inicialmente se tenía pensado añadir un sensor SCT-013, pero debido a su dificultad a la hora de montaje y programación no se ha podido instalar. Aun así, se explicará a la parte de sensores porqué sería una mejora de cara al futuro.

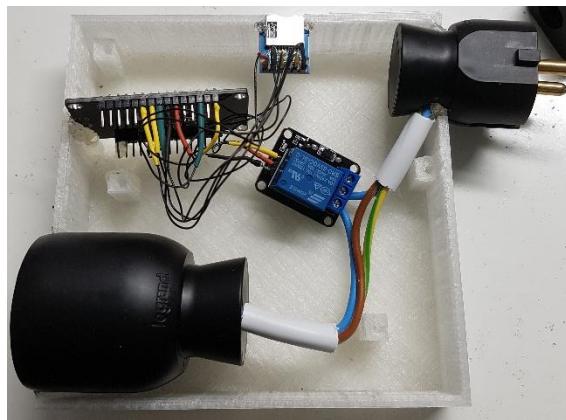


Figura 18.1: Módulo enchufe

¹⁷ Corriente Alterna, este tipo de corriente cambia de forma periódica la magnitud y el sentido, generando así una función senoidal.

18.1 Sensores y actuadores

18.1.1 Relé

(Relé, 2018)

El relé (Figura 18.2) es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes (Figura 18.3).

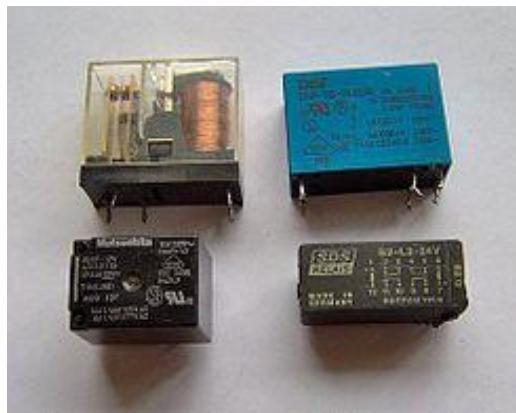


Figura 18.2: Relé

Dado que el relé es capaz de controlar un circuito de salida de mayor potencia que el de entrada, puede considerarse, en un amplio sentido, como un amplificador eléctrico.

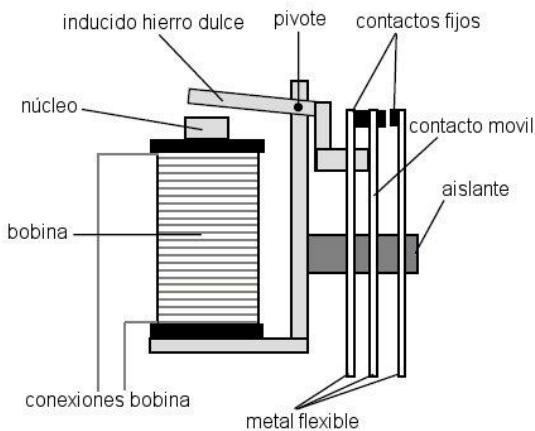


Figura 18.3: Partes del relé

18.1.2 SCT-013

(Tutorial sensor de corriente AC no invasivo SCT-013, 2018)



Figura 18.4: SCT-013

Los sensores de la serie SCT-013 (Figura 18.4) son sensores que trabajan como transformadores, la corriente que circula por el cable que deseamos medir actúa como el devanado primario (1 espira) e internamente tiene un devanado secundario que dependiendo del modelo pueden tener hasta más de 2000 espiras (Figura 18.5).

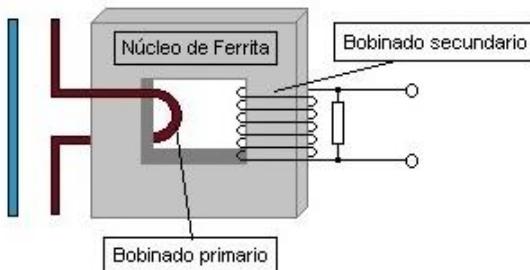


Figura 18.5: Funcionamiento del SCT-013

La cantidad de espiras representa la relación entre corriente que circula por el cable y la que el sensor nos entrega, esta relación o proporción es la que diferencia entre los diferentes modelos de sensores SCT-013, adicionalmente pueden tener una resistencia de carga en la salida de esta forma en lugar de corriente se trabaja con una salida voltaje.

Una ventaja de SCT-013 es que no necesitamos interrumpir (cortar o desempalmar) el cable que vamos a medir, esto porque al igual que una pinza amperimétrica tiene el núcleo partido.

18.2 Conexión

Descartando el SCT-013 la conexión es muy sencilla.

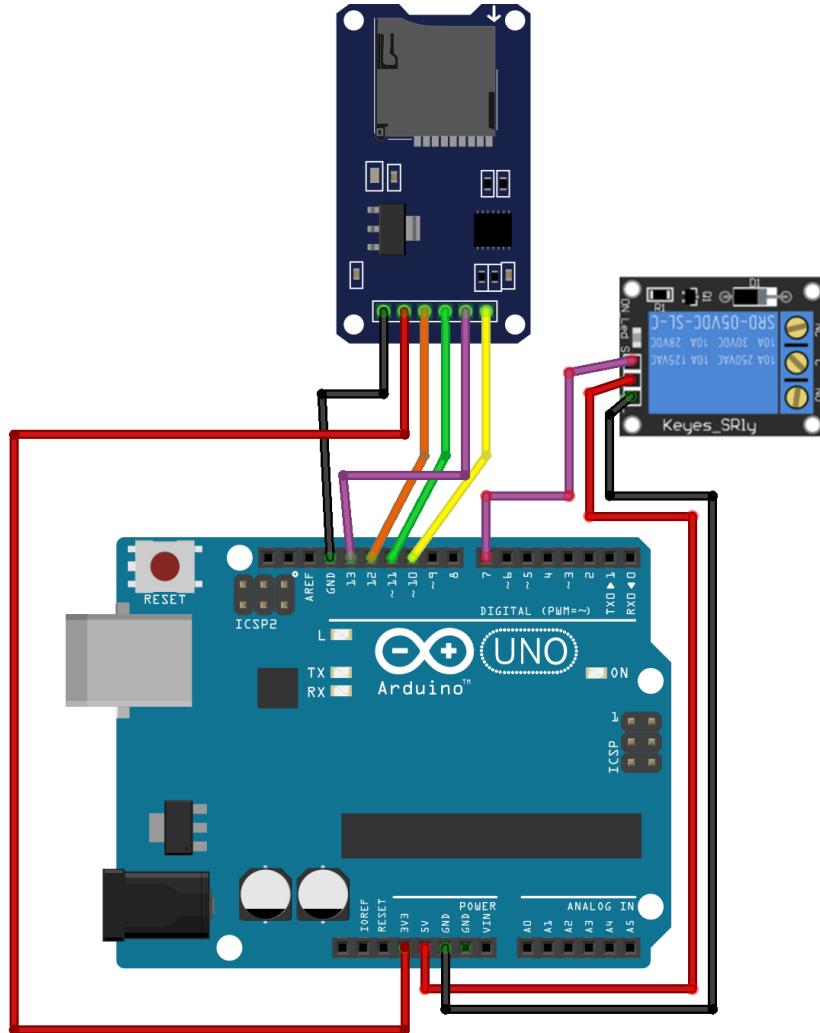


Figura 18.6: Conexión del módulo enchufe

18.3 Diagrama de flujo

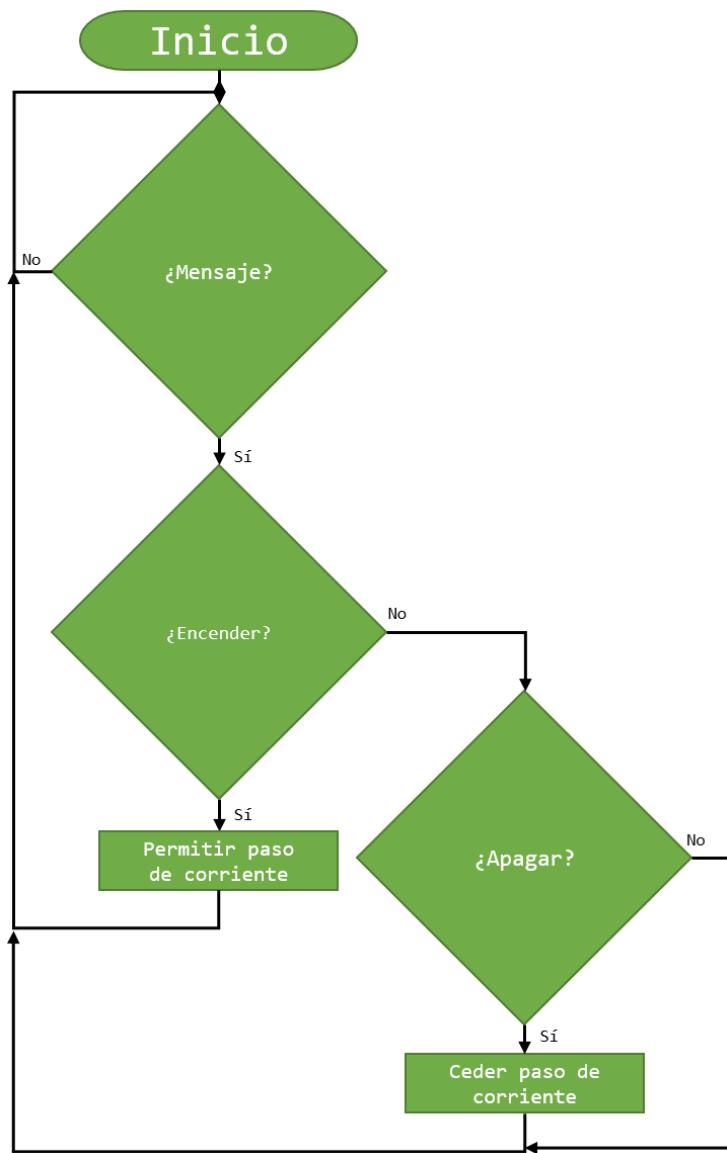


Figura 18.7: Diagrama de flujo del módulo enchufe

18.4 Tabla de verdad

$$\begin{aligned}
 a & \left\{ \begin{array}{l} 0 = - \\ 1 = \text{apagar} \end{array} \right. & x & \left\{ \begin{array}{l} 0 = \text{apagar} \\ 1 = \text{encender} \end{array} \right. \\
 b & \left\{ \begin{array}{l} 0 = - \\ 1 = \text{encender} \end{array} \right.
 \end{aligned}$$

a	b	x=b
0	0	X
0	1	1
1	0	0
1	1	X

Tabla 18.1: Tabla de verdad del módulo enchufe

18.5 Montaje



Figura 18.8: Interior del enchufe hembra



Figura 18.9: Interior del enchufe macho



Figura 18.10: Enchufes conectados con el relé



Figura 18.11: Prueba del módulo enchufe

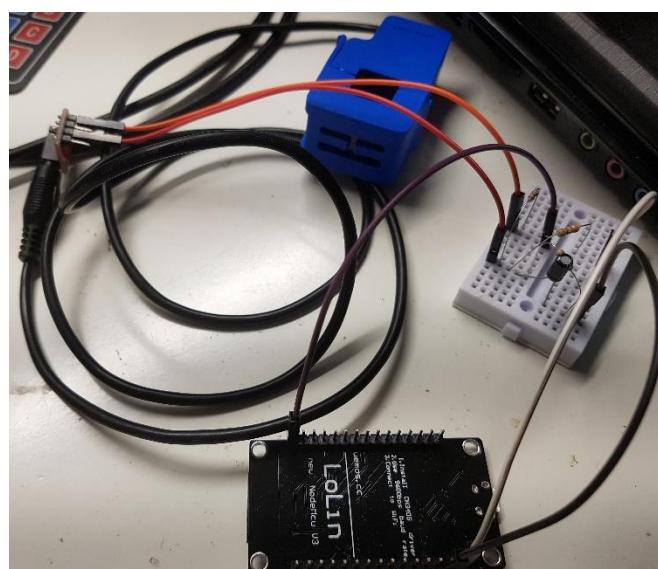


Figura 18.12: Intento de conexión del SCT-013



Figura 18.13: Prototipo del módulo enchufe

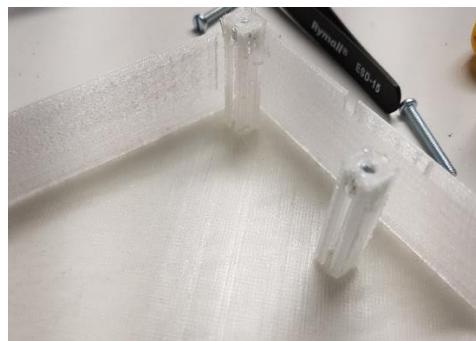


Figura 18.14: Parte inferior del módulo enchufe

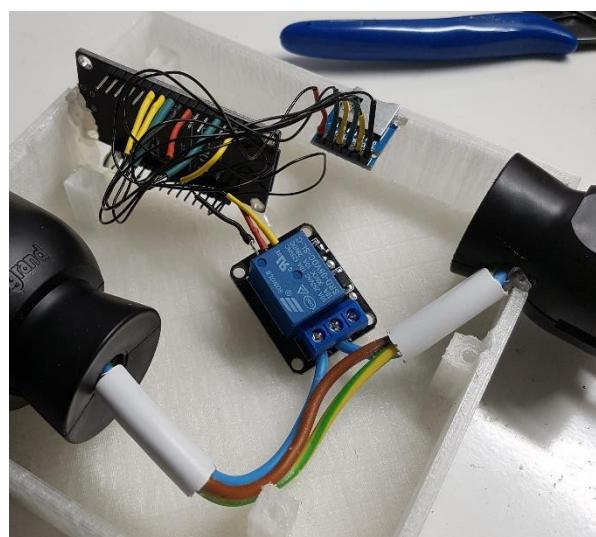


Figura 18.15: Interior del módulo enchufe

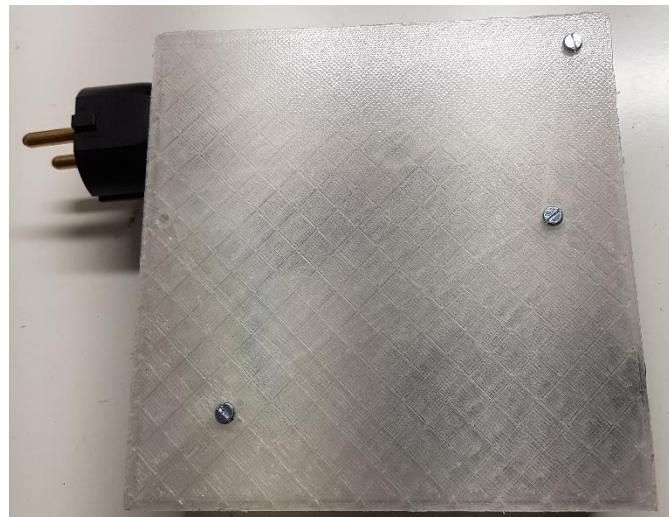


Figura 18.16: Parte superior del módulo enchufe

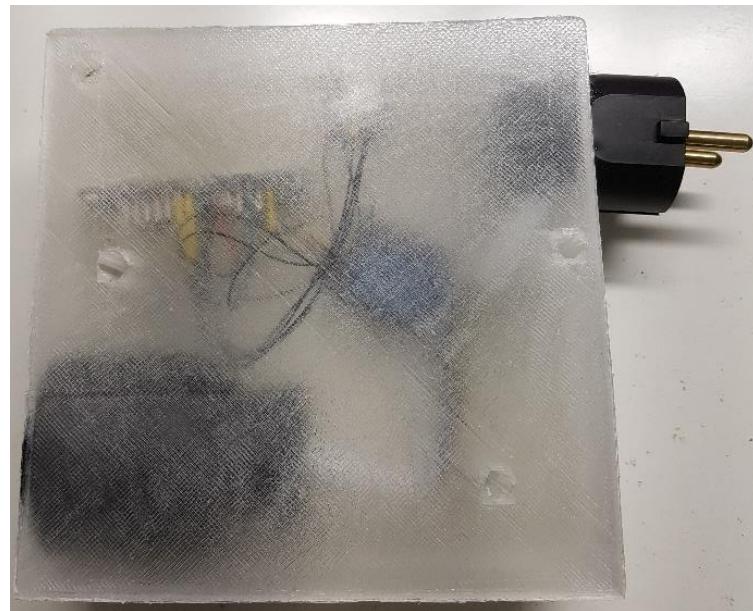


Figura 18.17: Parte inferior del módulo enchufe

Capítulo 19

Módulo riego

Este módulo permite regar las plantas automáticamente. Además de asignar la hora diaria de riego (o hacerlo manualmente), el programa central comprueba si hay altas probabilidades de lluvia al día siguiente y la humedad actual de las plantas para optimizar el agua.

Se ha construido a partir de un kit de Electrow (Arduino Automatic Smart Plant Watering Kit) bastante completo.

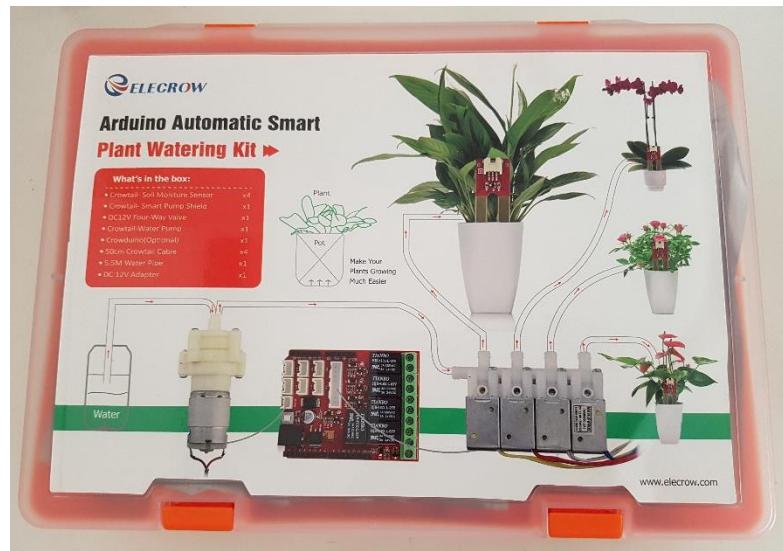


Figura 19.1: Módulo riego

19.1 Sensores y actuadores

19.1.1 Sensor de humedad



Figura 19.2: Sensor de humedad

Muy similar al sensor de altura del agua, retorna un valor en función de la humedad de la tierra.

Tiene una contra, y es que se oxida con el tiempo. Hay otros sensores tratados para que no se oxiden, pero cuestan un poco más (unos 15€/u), pero en este módulo se usarán los que vienen con el kit.

19.1.2 Crowtail-smart pump shield

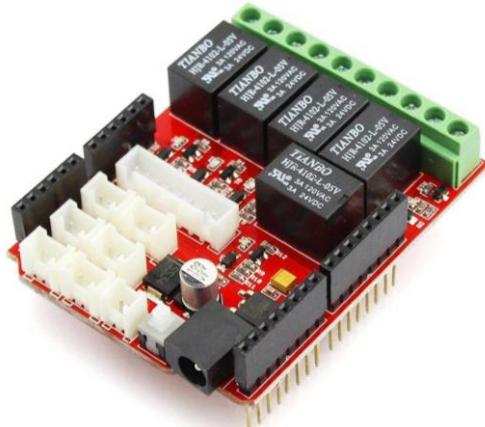


Figura 19.3: Shield del kit de riego

Un shield creado especialmente para este kit. Permite conectar los distintos sensores y actuadores de forma fácil a la placa Arduino.

Deja algunas entradas libres, haciendo posible añadir otra clase de sensores.

19.1.3 Válvula de cuatro vías

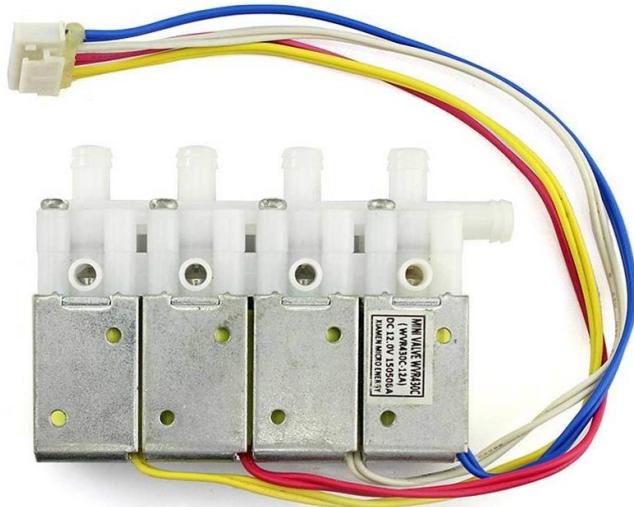


Figura 19.4: Válvula de cuatro vías

Imita a un relé, pero en vez de electrones, con agua. A partir de una entrada derivan cuatro salidas, que pueden abrirse o cerrarse mediante corriente eléctrica.

19.1.4 Bomba de agua



Figura 19.5: Bomba de agua

Absorbe el agua de un recipiente y la canaliza al sistema de tuberías.

Al tratarse de una bomba de bajo voltaje (12V) no es muy potente. No obstante, por el uso que se le da en este módulo (regar plantas) el caudal no está nada mal (aproximadamente 31.25ml/s - 1.88l/min - 112.5l/h, calculado cronometrando cuánto tiempo tarda en vaciar 250ml de agua, Figura 25.6).

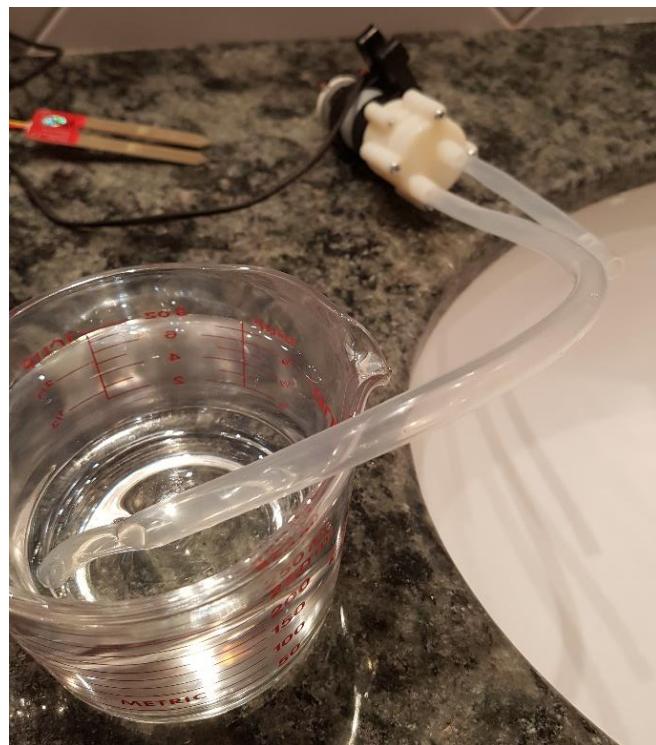


Figura 19.6: Cálculo del caudal de la bomba de agua

19.2 Conexión

La conexión es la indicada en el kit.

19.3 Diagrama de flujo

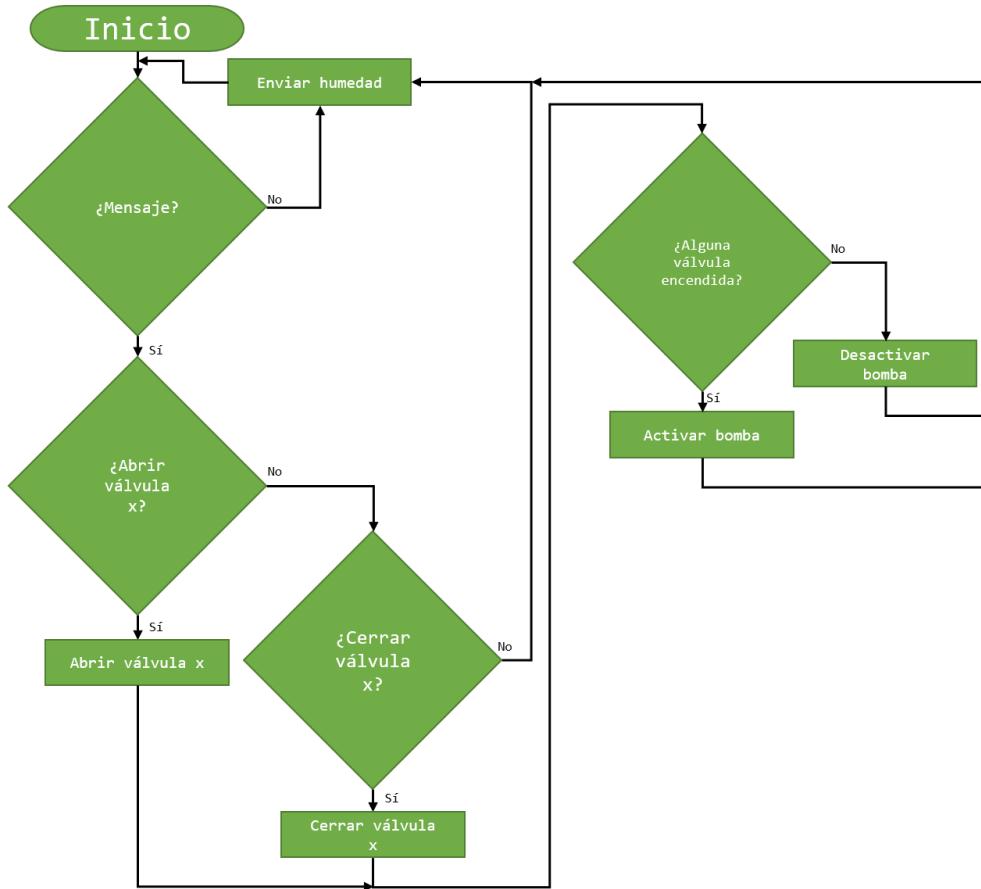


Figura 19.7: Diagrama de flujo del módulo riego

19.4 Tabla de verdad

La tabla de la verdad se complica demasiado debido a todas sus variables. En este caso es mucho mejor observar el diagrama de flujo.

19.5 Montaje

La parte más importante, ya que al tratarse de un kit su principal objetivo es construirlo. Además puedes mejorarla, ya sea añadiendo sensores, actuadores, o modificando el código.



Figura 19.8: Preparación del montaje del módulo riego

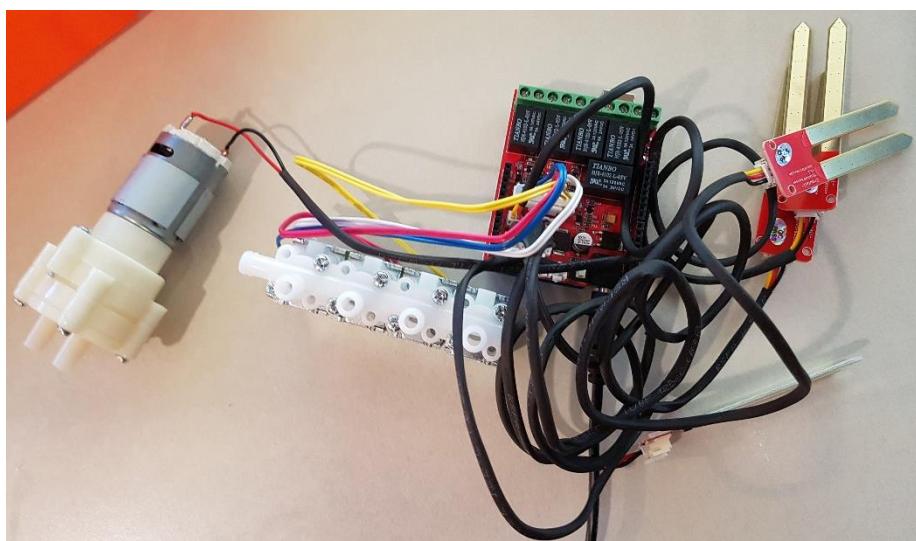


Figura 19.9: Módulo conectado

Capítulo 20

Módulo estación meteorológica

Es el módulo relacionado con el clima más avanzado y preciso. Los anteriores dos módulos se han creado para tener una buena relación calidad-precio, ya que el precio de este módulo es un tanto elevado (unos 120€).

Este módulo contiene un pluviómetro, un anemómetro y una veleta. Mediante los componentes mencionados anteriormente obtiene las precipitaciones, velocidad del viento y su dirección.

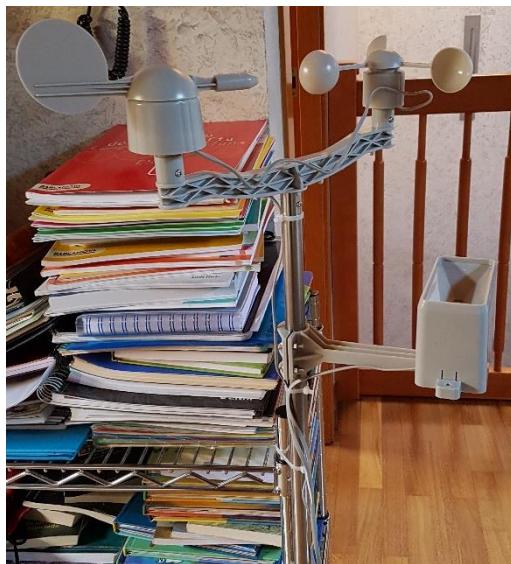


Figura 20.1: Estación meteorológica

Es muy recomendable tenerla fuera de casa.

20.1 Sensores y actuadores

20.1.1 Shield

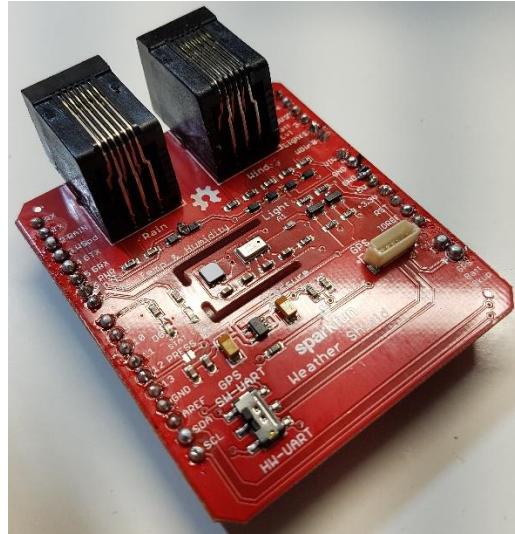


Figura 20.2: Shield de la estación meteorológica

Proporciona la cantidad de luz ambiente, humedad y temperatura, además de hacer de puente entre los sensores de viento y lluvia y Arduino, y permite añadir un GPS.

20.1.2 Pluviómetro



Figura 20.3: Pluviómetro

Mide la cantidad de lluvia caída.

20.1.3 Anemómetro



Figura 20.4: Anemómetro

Aparato meteorológico usado para medir la velocidad del viento.

20.1.4 Veleta



Figura 20.5: Veleta

Dispositivo giratorio que indica la dirección del viento.

20.2 Conexión

La conexión es la indicada en el kit.

20.3 Diagrama de flujo

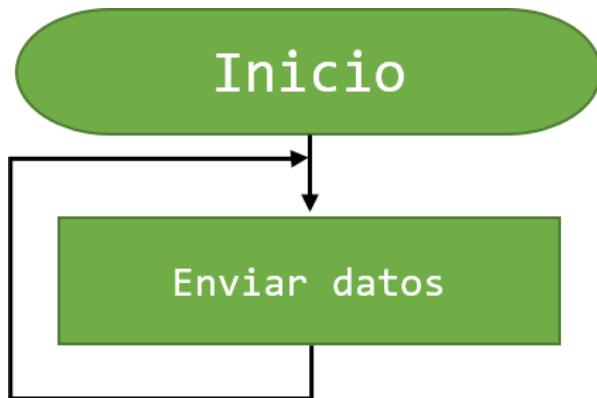


Figura 20.6: Diagrama de flujo del módulo estación meteorológica

20.4 Tabla de verdad

Al ser un módulo que envía datos siempre no hace falta una tabla de verdad.

20.5 Montaje



Figura 20.7: Caja con las partes del módulo



Figura 20.8: Partes del módulo

Capítulo 21

Módulo LED

El módulo LED, formado por un solo actuador (una tira LED), produce iluminación de distintos colores. Como desventaja, la luz que produce no es demasiada (Figura 21.6), así que tiene una función principalmente decorativa.

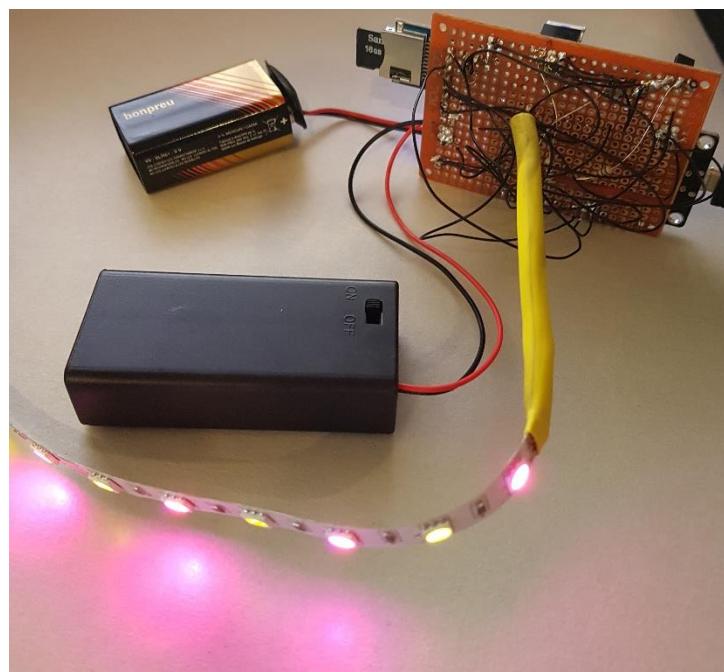


Figura 21.1: Módulo LED

21.1 Sensores y actuadores

21.1.1 LED Strip RGBWW

Se ha decidido comprar esta tira LED por su facilidad de ser modificada (las conexiones son accesibles, como puede comprobarse en la derecha de la Figura 21.2), y su bajo voltaje de operación (12V). Como ventaja adicional, cada LED trae su propia resistencia.

Además, al ser RGB¹⁸, se pueden reproducir distintos colores (Figura 21.2, componentes blancos). El “WW” del final hace referencia a blanco (white), en la Figura 21.2 serían los componentes de color amarillo.



Figura 21.2: Tira LED RGBWW

21.1.2 Transistores

Como ya se ha comentado, la tira de LEDs consume 12V (excesivo para los 5V que suministra Arduino). La solución a ese problema es el uso de transistores.

Para este módulo se han usado 3 transistores IRF530NPBF TO-220 (MOSFET), y 1 transistor BC548 TO-92 (NPN), que regularán el voltaje de salida en función del de entrada.

21.2 Conexión

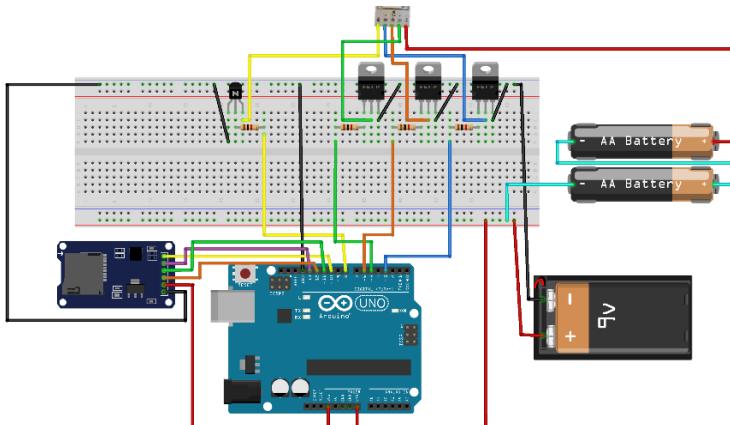


Figura 21.3: Conexión del módulo LED

¹⁸ Un LED RGB combina distintos niveles de rojo (red), verde (green) y azul (blue) para obtener cualquier color.

21.3 Diagrama de flujo

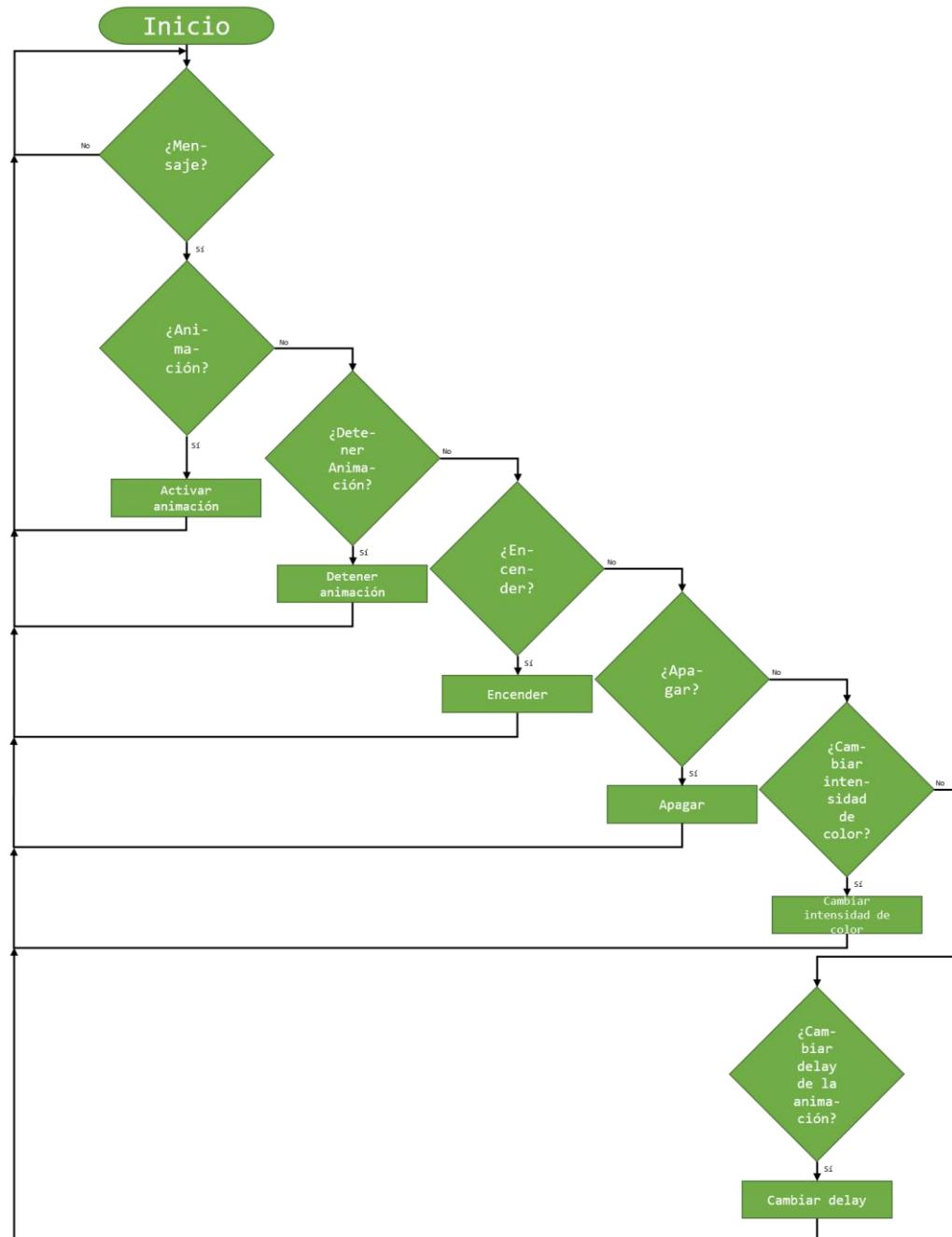


Figura 21.4: Diagrama de flujo del módulo LED

21.4 Tabla de verdad

$a \begin{cases} 0 = - \\ 1 = \text{iniciar animación} \end{cases}$	$x \begin{cases} 0 = - \\ 1 = \text{activar animación} \end{cases}$
$b \begin{cases} 0 = - \\ 1 = \text{detener animación} \end{cases}$	$y \begin{cases} 0 = - \\ 1 = \text{desactivar animación} \end{cases}$
$c \begin{cases} 0 = - \\ 1 = \text{cambiar delay animación} \end{cases}$	$z \begin{cases} 0 = - \\ 1 = \text{cambiar delay animación} \end{cases}$

a	B	c	x=a	y=b	c=z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	X	X	X
1	0	0	1	0	0
1	0	1	X	X	X
1	1	0	X	X	X
1	1	1	X	X	X

Tabla 21.1: Tabla de verdad del módulo LED

21.5 Montaje

Uno de los módulos más complicados de hacer, ya que se ha requerido el uso de transistores.

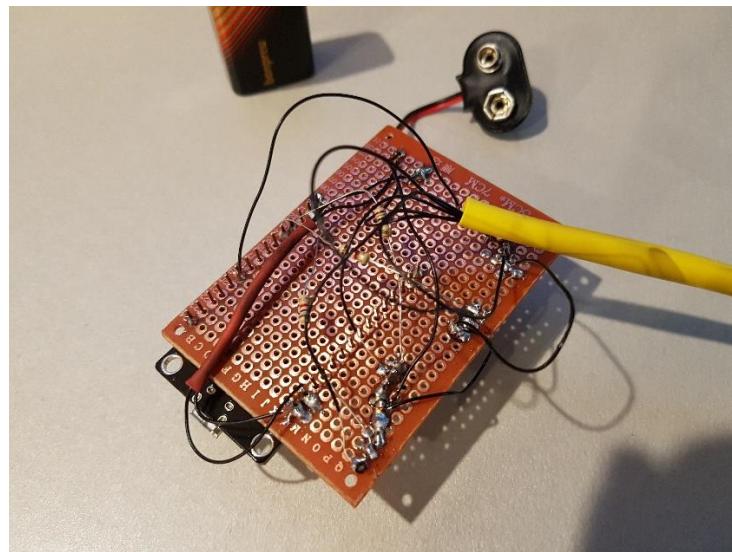


Figura 21.5: Módulo LED por debajo



Figura 21.6: Brillo del módulo LED

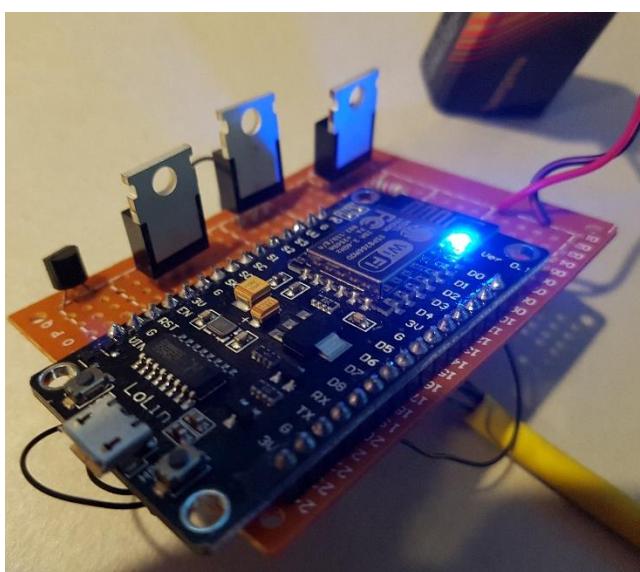


Figura 21.7: Módulo LED funcionando



Figura 21.8: Tira RGBWW

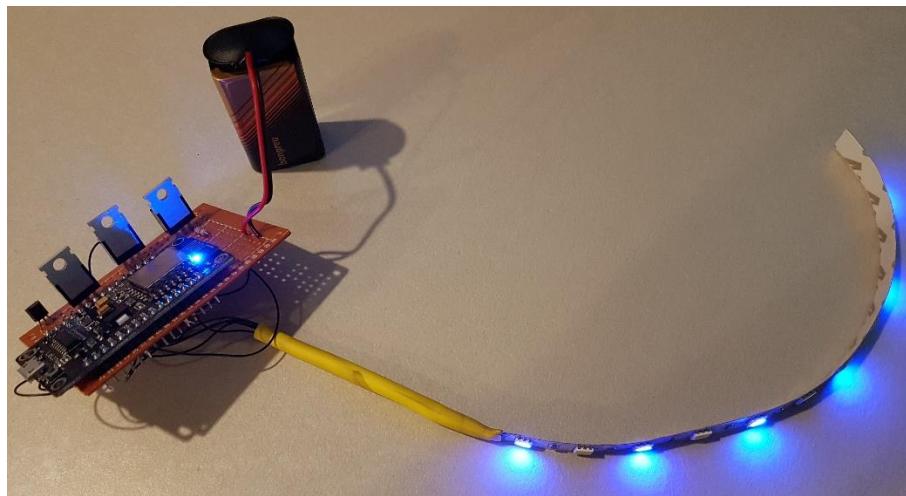


Figura 21.9: Módulo LED

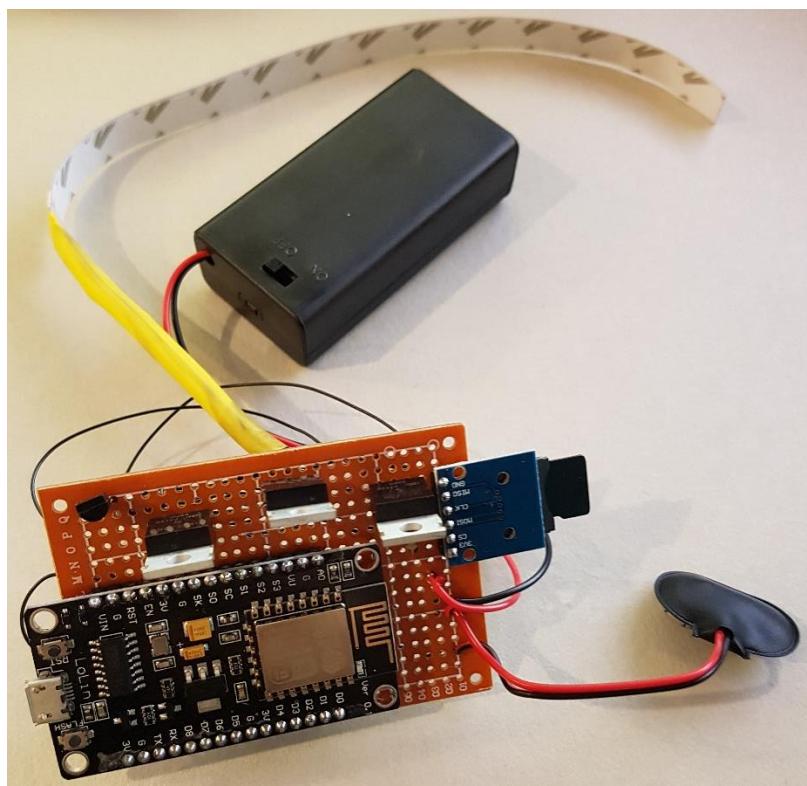


Figura 21.10: Módulo LED terminado

Capítulo 22

Módulo central

El módulo central es el “cerebro” de todo el sistema domótico, además de realizar otras tareas.

Puede dividirse en 3 partes:

- » *Domótica*

El encargado de procesar toda la información obtenida de los diversos módulos. Almacena datos, envía órdenes y realiza otras acciones como enviar SMS. Esta parte está programada en Python.

- » *WEB*

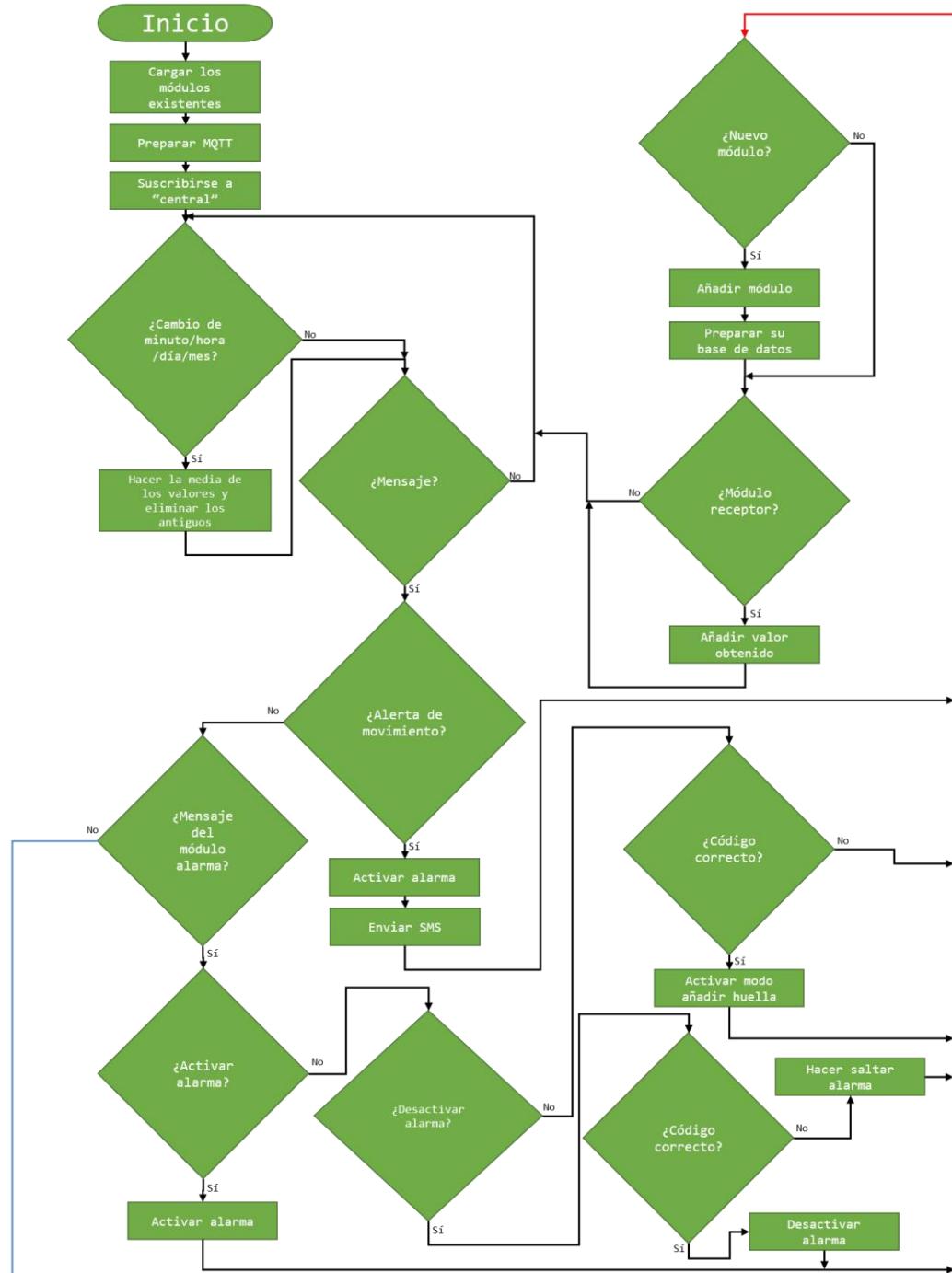
Proporciona al usuario una forma cómoda de interactuar con el sistema domótico. Al ser responsive puede accederse tanto en móviles como en tabletas u ordenadores. Ha sido programada con HTML, CSS, JavaScript y PHP.

- » *Base de datos*

En ella se almacenan datos para, más tarde, acceder a ellos para recuperar los valores. Se accede a la base de datos mediante SQL.

Puede encontrarse todo el código en el Anexo B, Código.

22.1 Domótica



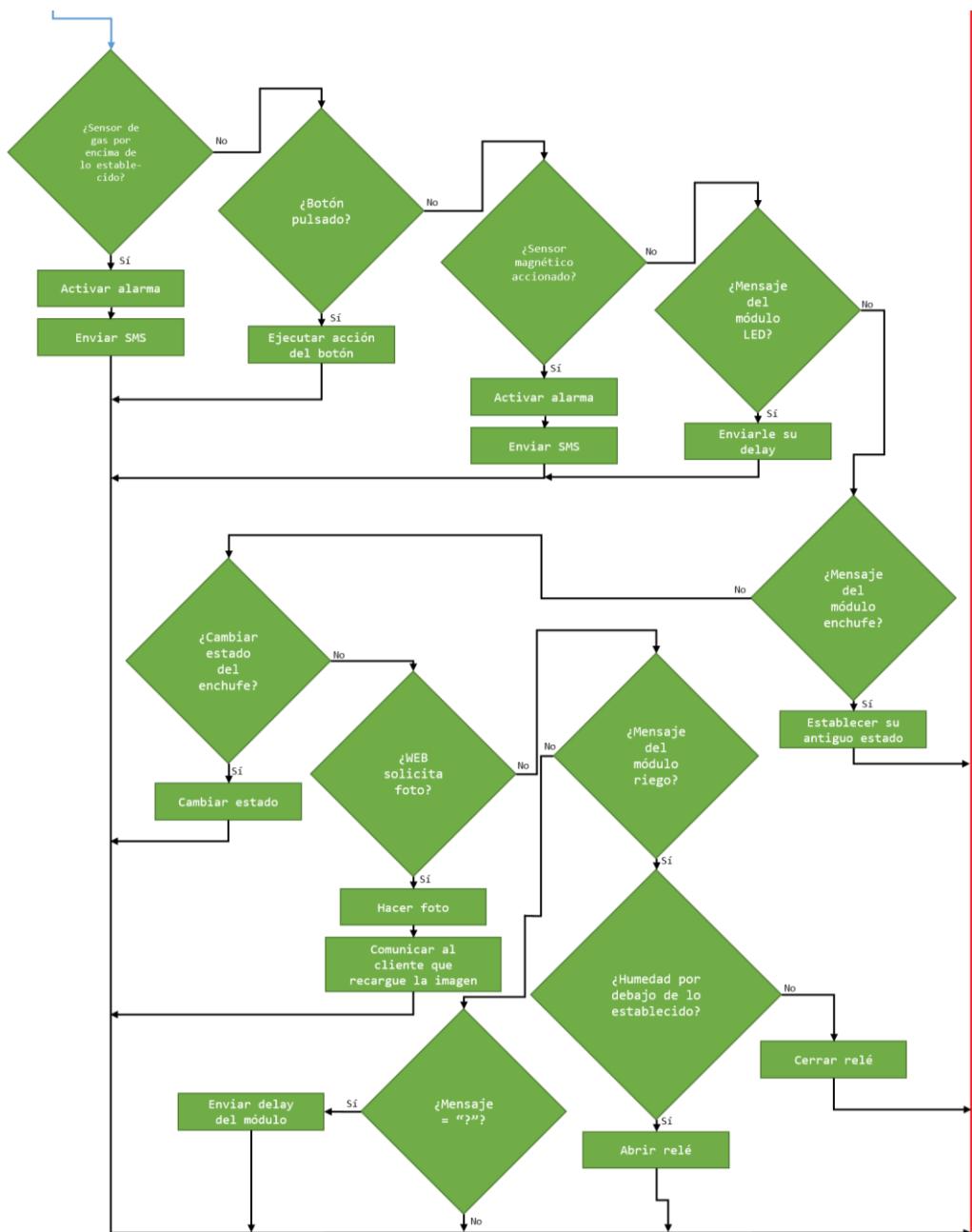


Figura 22.1: Diagrama de flujo del módulo central

22.2 WEB



Figura 22.2: WEB > Información

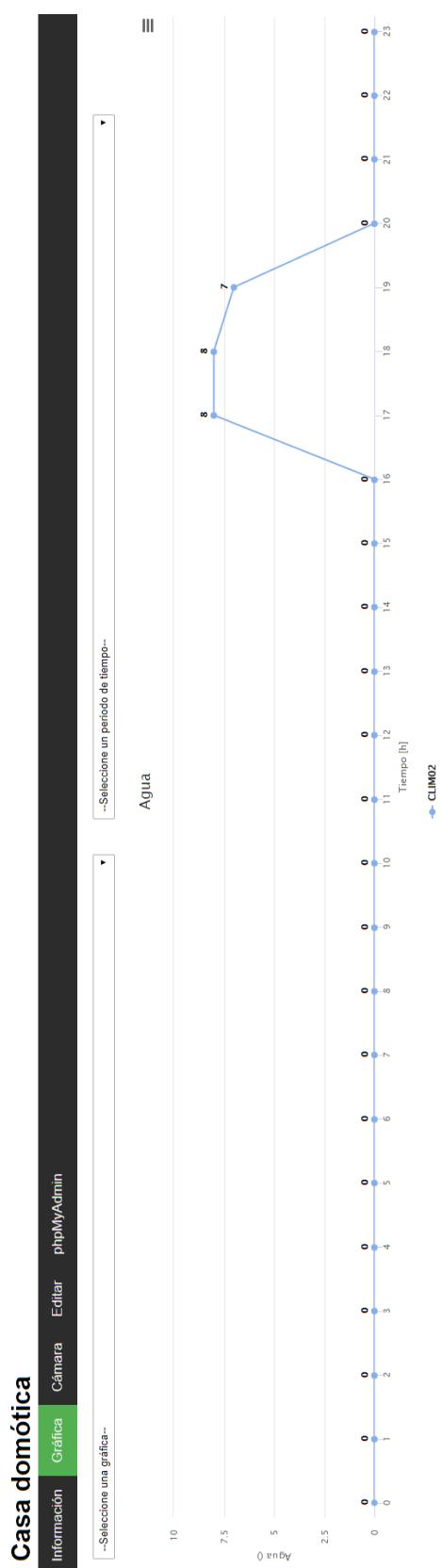


Figura 22.3: WEB > Gráfica

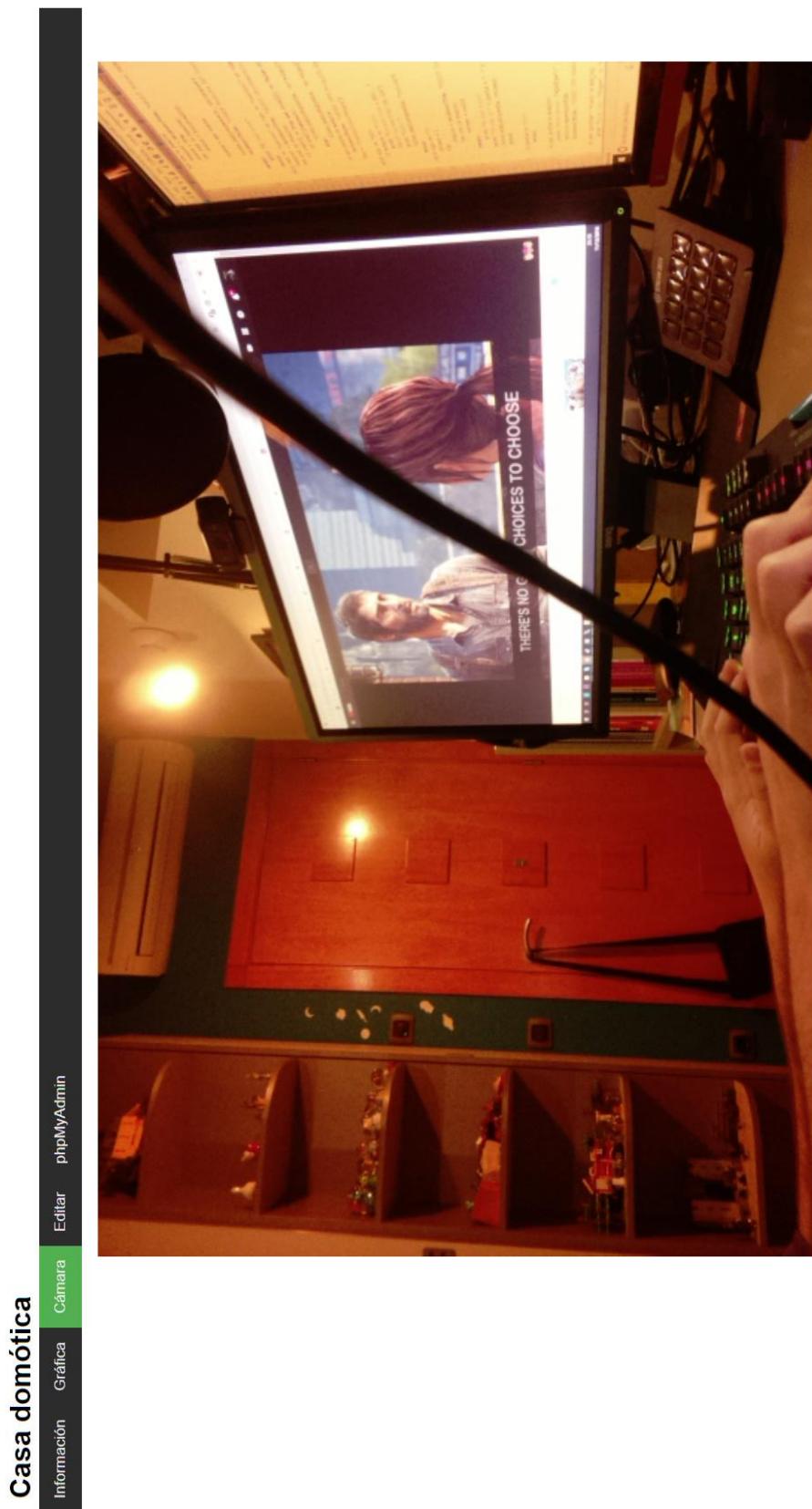


Figura 22.4: WEB > Cámara

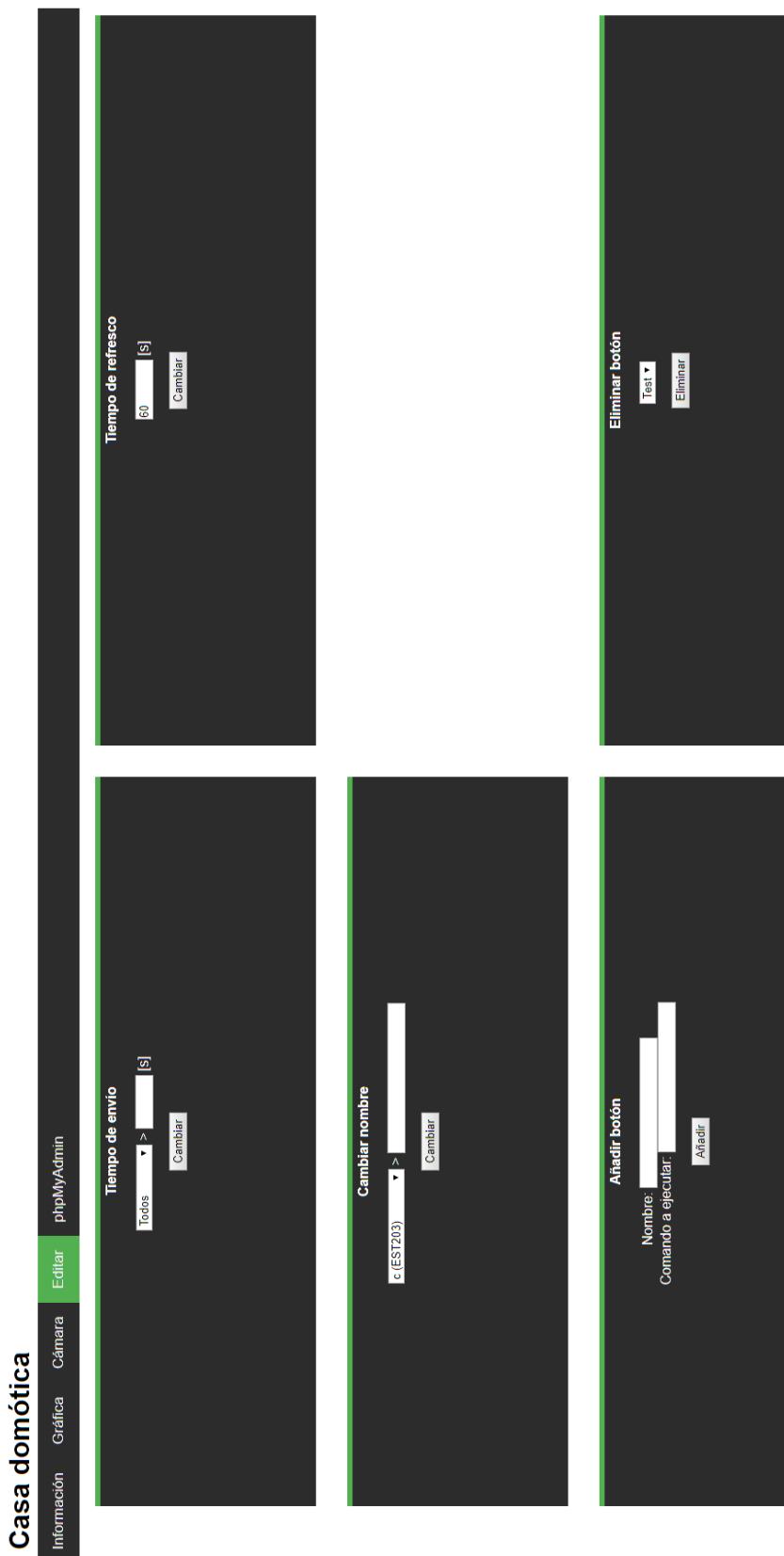


Figura 22.5: WEB > Editar

22.3 Base de datos

En la figura 22.6 se observan las tablas usadas en este sistema domótico. Hay información más técnica y detallado en el anexo B.4, Base de datos.

The screenshot shows the phpMyAdmin interface with the following details:

- Servidor:** localhost:3306
- Base de datos:** Domotica
- Panel de navegación:** Reciente, Favoritas
- Árbol de esquemas:** Domotica (contiene Nueva, Alarm, Alarma, Botones, config, Enchufes, LED, Nombres, Riego, RiegoHora, Tipos, Usuarios, Valor), information_schema, mysql, performance_schema, phpmyadmin.
- Barra superior:** Examinar, Estructura, SQL
- Mensaje de advertencia:** La selección actual no contiene una columna
- Resultado de la consulta:** Mostrando filas 0 - 17 (total de 18). La consulta es: `SELECT * FROM `Valor` ORDER BY `Time` DESC`
- Tabla de resultados:** Una tabla con 18 filas y 5 columnas: ind, Tiempo, Time, Val. Los datos son:

ind	Tiempo	Time	Val
7	m	30	433
10	m	30	423
9	m	30	429
8	m	30	425
9	m	29	440
8	m	29	437
7	m	29	445
10	m	29	435
5	h	19	7
2	h	19	790
1	h	19	1518
0	h	19	727
4	h	19	53
5	h	17	8
4	h	17	50
2	h	0	199
0	h	0	225
1	h	0	425

Figura 22.6: Base de datos

22.4 Montaje

Se ha usado una cámara para hacer fotos y una pantalla táctil para hacer la interacción más cómoda.



Figura 22.7: Instalación de la pantalla táctil

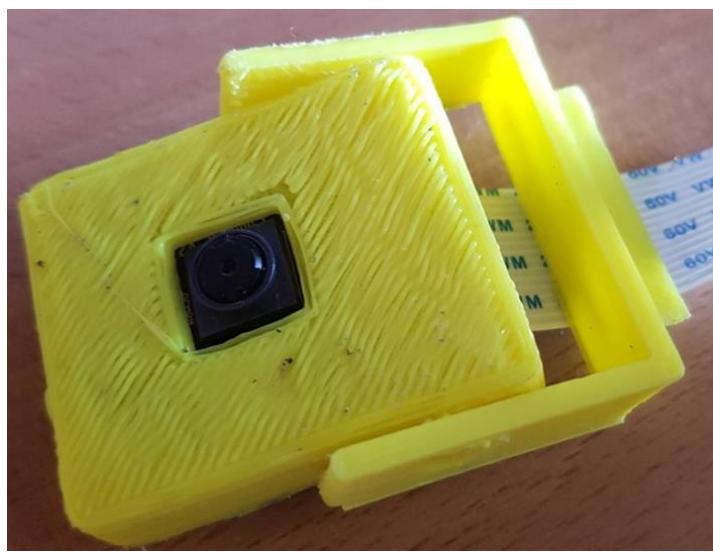


Figura 22.8: Cámara para Raspberry Pi con carcasa

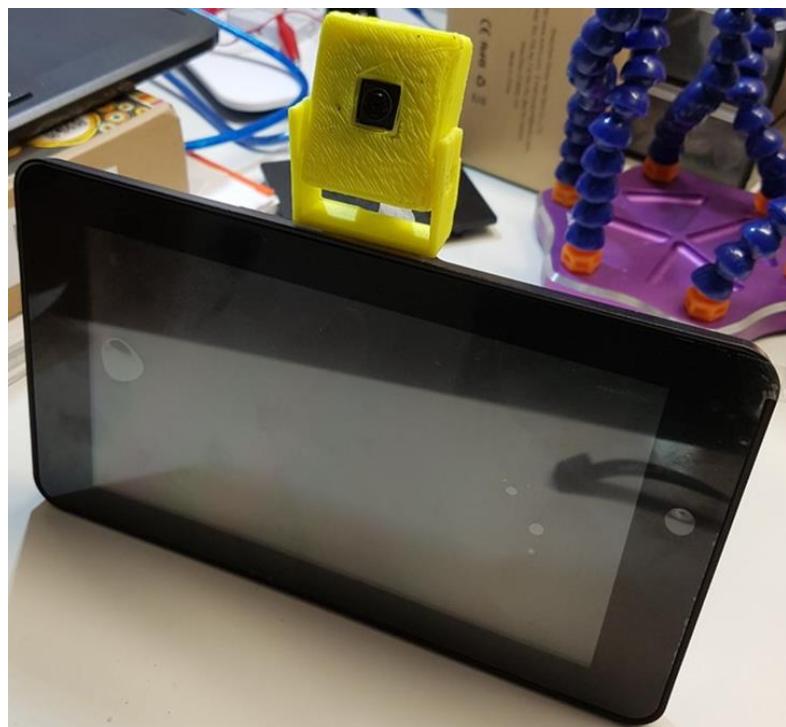


Figura 22.9: Cámara montada

Parte VI

Conclusiones

Capítulo 23

Introducción

Viendo los sistemas domóticos actuales se pueden comparar con el sistema que se quiere crear al final de este trabajo. Factores como los actuadores, la precisión, el montaje, los servicios... son las causas de que el precio se dispare.

Por esta razón, se han elegido 7 factores a comparar. Estos son los actuadores, los receptores, los servicios, la comunicación, el precio, el montaje y el código abierto y hardware libre.

En la Tabla 23.1 se pueden ver los puntos en los que los sistemas domóticos actuales son peores (↓), los puntos en los que el sistema domótico de este trabajo es mejor (↑) o mucho mejor (↑↑), y los puntos en los que ambos sistemas domóticos quedan en tablas (~).

Seguridad	↓
Actuadores	↓
Sensores y precisión	~
Servicios	~
Comunicación	~
Precio	↑
Instalación y montaje	↑
Sostenibilidad medioambiental	↑
Código abierto y hardware libre	↑↑

Tabla 23.1: Puntos a favor y en contra del sistema domótico de este trabajo

Capítulo 24

Seguridad

La seguridad es un factor muy importante, ya que si esta no es suficiente, un atacante puede obtener información confidencial, crearla, o eliminarla. A continuación se explicarán las vulnerabilidades halladas en la WEB y comunicaciones del trabajo, junto a la solución para dicha. Cabe destacar que los fallos de seguridad no han sido producidos a propósito, sino que por falta de conocimientos. Este punto hace que el sistema domótico sea muy malo, pero al tener diversas soluciones (que no se han implementado por falta de tiempo) se compensa.

El primer fallo de seguridad se produce en la comunicación Arduino-Raspberry, mediante MQTT. Esta comunicación requiere de una IP y un puerto, pero no requiere contraseña. Además, la información viaja plana (sin codificar). La solución sería añadir una contraseña (esta tendría que especificarse en las Micro SDs de los módulos Arduino) y algún tipo de codificación (existen muchos tipos, algunos más seguros que otros).

Los demás fallos se producen en la WEB, y se mencionarán todos los encontrados. Las técnicas empleadas se mencionan muy por encima, para más información se recomienda el libro “Hacking y seguridad de páginas Web”, publicado el 2014 por Ra-Ma.

El login, comparando con el resto de la WEB, está bien hecho. El password se codifica con md5, evitando así que alguien lo obtenga por *sniffing*¹⁹. Además, se obtiene el password del usuario introducido, y luego se compara el password obtenido con el introducido (revisando también que se obtenga al menos 1 password). Esto evita un tipo de SQL Injection²⁰ (' OR 1=1#), pero no todos. Después de algunos intentos y revisar el código fuente, se ha encontrado una vulnerabilidad. Esta consiste en poner un nombre de usuario concreto (' UNION SELECT md5('1')#), de forma que unas a la respuesta legítima una contraseña la cual se conozca (en el caso anterior, 1). Este problema se soluciona con lo que se denomina una declaración preparada.

¹⁹ Un atacante se pone en medio de la comunicación entre el cliente y servidor, y escucha toda la conversa

²⁰ Inyección SQL. Método de infiltración de código no deseado en SQL.

Otro problema relacionado con el login es la fuerza bruta, que consiste en intentar todas las posibles combinaciones. No hay una forma definitiva de solucionar este fallo, pero hay algunas que lo hacen más complicado de realizarlo con éxito. Entre ellas se destacan las siguientes:

- » Por parte del usuario
 - Siempre cambiar la contraseña, nunca dejar la que viene por defecto.
 - Usar contraseñas seguras (largas, con mayúsculas, minúsculas, números y caracteres especiales).
- » Por parte del servidor
 - Añadir *captcha*²¹
 - Añadir límite de intentos
 - Obtener la IP de posibles atacantes (para bloquearla)

Una vez dentro, las cosas empeoran. El constante uso de impresión de variables que el usuario introduce, sin filtro alguno, da la posibilidad a los atacantes de usar XSS²². Esto permitiría al atacante acceder a la WEB, ya logueado, sin saber ningún usuario ni contraseña (mediante la obtención de las cookies²³ del usuario víctima). Se soluciona fácilmente, con funciones implementadas por el propio PHP (*htmlspecialchars()*).

En esta parte de la WEB también se puede realizar SQL Injection, además de poder ver los errores que la página genera (Figura 24.1). Esto es peligroso porque el atacante puede obtener información sobre el tipo y versión de la base de datos, el sistema operativo del servidor, nombres y tipos de tablas... Se soluciona simplemente, ocultando el mensaje de error o creando algunos propios.



Figura 24.1: Error generado por una consulta SQL

Un ejemplo de SQL Injection sería insertar cualquier código que produzca un funcionamiento no deseado (' WHERE 1;# actualiza todas las tablas, como puede comprobarse en la Figura 24.2). La solución es la misma indicada en el SQL Injection del login.

²¹ Prueba realizada para distinguir humanos de ordenadores.

²² Cross-site scripting. Tipo de vulnerabilidad que ataca a la parte del usuario, no del servidor. Requiere Ingeniería social.

²³ Pequeña información enviada por un sitio WEB y almacenada en el navegador del usuario, permitiendo al sitio WEB consultar dicha información en todo momento.

ID	Nombre	Tiempo
EST203		4
TEMP01		4
LED01		4
CLIM02		4
INT01		4
RIEG01		4

Figura 24.2: Ejemplo de SQL Injection

Si se combinan las dos últimas vulnerabilidades se puede crear XSS Persistente, lo que significaría que todo usuario que entrara en la WEB ejecutaría el código malicioso insertado por el atacante. Tiene la misma solución que el ataque tipo XSS.

Así que, como se ha dicho, es problemático pero tiene solución.

Capítulo 25

Actuadores

Los actuadores convierten la energía eléctrica en energía mecánica, permitiendo así interactuar abriendo ventanas, bloqueando puertas... También pueden, mediante electroluminiscencia, crear luz.

La mayoría de actuadores requieren de motores potentes, que son caros y consumen una cantidad muy grande de energía. Esa es la razón principal de que Arduino no disponga de estos tipos de motores (con sus 5V no puede proveerle suficiente energía). No obstante, Arduino también tiene motores muy buenos que podrían usarse para estas tareas, solo que no lo harían tan bien.

El sistema domótico creado dispone de muy pocos actuadores (el módulo LED y el módulo riego) por las razones que ya mencionadas anteriormente. Se podrían añadir motores más potentes (para, por ejemplo, abrir la puerta del garaje) usando transformadores con mayor salida de voltaje (para suministrar el corriente necesario) y transistores o relés (para dar paso o cortar la corriente del transformador). Ya que no se han añadido motores potentes por falta de tiempo y material, este es un punto a favor de los sistemas domóticos actuales.

Capítulo 26

Sensores

Los sensores captan datos y los envían. Son los encargados de decir qué temperatura hace, la humedad, si un botón está presionado...

Arduino tiene una cantidad ingente de sensores, y se han usado muchos de ellos para crear los distintos módulos. Hay de más precisos, que ofrecen más funciones... y todo esto afecta al precio del sensor.

Los sensores incluidos en este trabajo son muchos, pero cabe destacar que los sensores que ponen en los sistemas domóticos actuales son los mismos, o otros que proporcionan aún más datos. Este es el motivo por el que este punto se queda en neutral.

Además, la precisión de la mayoría de sensores de Arduino no es muy buena, pero tampoco es mala. Por eso también el punto de precisión se queda en neutral.

Capítulo 27

Servicios

Los servicios que ofrece el sistema domótico de este trabajo son muchos: retransmisión de la cámara en vivo, control on-line, automatización de algunos sistemas... No obstante, los sistemas domóticos actuales también los tienen, así que este punto se queda en neutral.

Además, este sistema está limitado a una sola cámara para la Raspberry Pi, problema que los otros sistemas no tienen. Aunque es cierto que con una sola cámara hay suficiente para la mayoría de usuarios, algunos pueden necesitar más (para, por ejemplo, desconectar la alarma con reconocimiento facial y abrir la puerta del garaje con solo leer la matrícula del coche del dueño de la casa). Para estos usuarios, hay diversas soluciones (que no se han tenido en cuenta a la hora de hacer el trabajo, pero como se trata de un proyecto de código abierto, podrían hacérselo los propios usuarios):

- » Crear un *switcher*²⁴, permitiendo así (mediante un PIN auxiliar de la placa Raspberry) cambiar entre distintas cámaras.
- » Usar una cámara USB.
Raspberry permite cámaras mediante USB. Como punto en contra, estas consumen mucha CPU.

²⁴ Conmutador

Capítulo 28

Comunicación

Uno de los objetivos de este trabajo era hacer el sistema domótico sin necesidad de cableado, inalámbrico. Este factor ha conllevado bastantes problemas que se han ido solucionando poco a poco, hasta obtener el resultado deseado.

Para lograr el sistema inalámbrico, se usa como protocolo el MQTT. Esto es muy práctico, ya que da la posibilidad de mover módulos con facilidad e instalarlos sin instalar cableado por toda la casa.

Este método también puede ocasionar problemas: inhibiciones, lentitud de transporte si la red Wi-Fi no es rápida, e incluso necesidad de instalar repetidores, para que llegue conexión en sitios muy alejados del router.

El hecho que tenga puntos a favor y en contra hace que obtenga una puntuación neutral.

Capítulo 29

Precio

El precio es otro factor principal en los sistemas domóticos. Puede verse como un sistema domótico mínimamente bueno llega de sobra a los 1000€. En este trabajo uno de los objetivos es no sobrepasar esa cantidad.

Se ha creado un presupuesto para los módulos, y luego algunas mejoras que cuestan algo más de dinero pero ofrecen servicios adicionales.

Primero se ha de definir un coste fijo para cada módulo, ya que todos los módulos tienen ciertas partes en común (Tabla 29.1). Este coste es el de los componentes básicos para el funcionamiento y montaje, formado por la placa, el receptor de micro SD, la toma de 9V para baterías, la toma de 9V para el transformador, la SD y la placa de pruebas (se menosprecia el estaño, cableado y electricidad consumida por el soldador, además de los gastos de envío). Este coste sirve para la mayoría de los módulos, pero para los que no sirva se añadirá en el coste “suplemento”.

Coste fijo por módulo	
Placa NodeMCU	2.89€
Receptor micro SD	0.35€
9V (batería)	0.13€
9V (transformador)	0.10€
SD	1.67€
Placa de pruebas	0.13€
TOTAL	5.27€

Tabla 29.1: Coste fijo por módulo

A continuación se pueden observar las distintas tablas para cada módulo.

Módulo botón	
Coste fijo	5.27€
Botón	0.0580€
Resistencia de 10kΩ	0.0426€
TOTAL	5.37€

Tabla 29.2: Coste del módulo botón

Módulo movimiento	
Coste fijo	5.27€
LED	0.0045€
Resistencia de 220Ω	0.0052€
Sensor de movimiento	1.69€
TOTAL	6.97€

Tabla 29.3: Coste del módulo movimiento

Módulo apertura	
Coste fijo	5.27€
LED	0.0045€
Resistencia de 220Ω	0.0052€
Sensor magnético	3.60€
TOTAL	8.88€

Tabla 29.4: Coste del módulo apertura

Módulo seguridad	
Coste fijo	5.27€
Buzzer	0.65€
Sensor de llama	1.34€
MQ-2	3.29€
MQ-5	1€
MQ-7	3.49€
MQ-135	1.10€
TOTAL	16.14€

Tabla 29.5: Coste del módulo seguridad

Módulo sensor climático simple	
Coste fijo	5.27€
Sensor de humedad y temperatura	2.88€
Fotoresistor	0.1860€
Resistencia de 10kΩ	0.0426€
TOTAL	8.38€

Tabla 29.6: Coste del módulo climático simple

Módulo sensor climático avanzado	
Coste fijo	5.27€
Sensor de humedad, temperatura y presión	2.34€
Fotoresistor	0.1860€
Resistencia de 10kΩ	0.0426€
Diodo (x2)	0.0148€/u
Sensor de altura del agua	1.05€
Modelo 3D (108g)	1.84€
TOTAL	10.76€

Tabla 29.7: Coste del módulo climático avanzado

Módulo alarma	
Coste fijo	5.27€
Suplemento	22.46€
4x4 matrix keypad	2.61€
4 LED display	1€
Lector de huella dactilar	49.90€
Buzzer	0.65€
TOTAL	81.89€

Tabla 29.8: Coste del módulo alarma

Módulo enchufe	
Coste fijo	5.27€
Toma hembra	2.42€
Toma macho	1.31€
Relé	1.10€
Modelo 3D (40g)	0.7200€
Modelo 3D (77g)	1.39€
TOTAL	12.21€

Tabla 29.9: Coste del módulo enchufe

Módulo LED	
Coste fijo	5.27€
Tira RGBWW	5.31€
Entrada 3V (baterías)	0.45€
Transistor MOSFET (x3)	0.1630€/u
Transistor BJT	0.1120€
TOTAL	11.63€

Tabla 29.10: Coste del módulo LED

Módulo riego	
Coste fijo	5.27€
Suplemento	22.46€
Kit	42.26€
TOTAL	69.99€

Tabla 29.11: Coste del módulo riego

Módulo estación meteorológica	
Coste fijo	5.27€
Suplemento	22.46€
Estación	69.00€
Shield	42.00€
Conectores (x2)	1.10€/u
TOTAL	140.93€

Tabla 29.12: Coste del módulo estación meteorológica

Raspberry	
Raspberry Pi B+	39.99€
Pantalla táctil 7"	76.36€
Carcasa para pantalla táctil	14.20€
Refrigeración	10.99€
Cámara	25.40€
TOTAL	166,94€

Tabla 29.13: Coste de la Raspberry

En una casa media (2 habitaciones, 1 baño, 1 cocina, 1 comedor, un garaje, y 7 ventanas) el precio más económico sería de 399,07€ (1 módulo central, 2 módulos enchufes, 1 módulo climático avanzado, 1 módulo seguridad, 4 módulos movimiento, 8 módulos apertura, y 1 módulo alarma). La opción normal (opción económica, más 2 módulos LED, 2 módulos enchufes, 1 módulo riego y 1 módulo climático simple) saldría por 525,12€. Y la más cara (la opción normal, más 2 módulos enchufes, la estación meteorológica, y quitando el módulo climático avanzado), por 679,71€.

El resultado es un precio bastante aceptable, en comparación con los demás sistemas domóticos actuales (estimación con los mismos servicios: 727.20€-2,099.32€, puede consultarse en el anexo C, Precios actuales).

Capítulo 30

Instalación y montaje

El montaje para estos módulos es tan fácil que solo requiere unos conocimientos mínimos de soldadura. Dando unas instrucciones los compradores de este sistema domótico podrían ahorrarse algo de dinero.

Otra ventaja es la instalación, básicamente es cambiar el nombre y contraseña del Wi-Fi en las tarjetas micro SD y suministrar los módulos con energía (mediante pilas o transformadores).

Por lo mencionado, se considera un punto a favor.

Capítulo 31

Sostenibilidad medioambiental

A continuación se mencionarán las ventajas en cuanto a sostenibilidad y medioambiente que ofrece un sistema domótico open source:

» *Sistema flexible*

El sistema domótico es modular y adaptable a cada usuario, entonces, el usuario no adquiere servicios indeseados, además de durar más años ya que es ampliable, minimizando el coste y los residuos de producción.

» *Facil de reparar*

El sistema usa componentes accesibles para todos, y se ha realizado con soldadura no industrial. Esto permite que, en el caso de que se rompa, no se ha de comprar el módulo entero, permitiendo así minimizar residuos.

» *Energía*

Actualmente, los módulos usan pilas de 9V para funcionar. Esto no resulta viable, ya que genera muchos residuos y se ha de estar pendiente de la carga restante. Por esto, se puede (en un futuro) implementar baterías recargables o transformadores, evitando residuos.

» *Gestión energética*

La simple idea de ‘domótica’ implica gestión energética. No obstante, por la complejidad del sensor, no se ha podido implementar. Aún así, algunos módulos (como el módulo LED) se desconectan ahorrando energía (el ahorro de energía es mínimo, pero existente).

» *Integración con energías renovables*

Las energías renovables aún presentan varios inconvenientes. Uno de ellos es que los picos de generación de energía no siempre coinciden con los picos de consumo de energía de los usuarios. La domótica facilitaría al usuario derivar automáticamente el uso de aquellos sistemas más flexibles (como lavadoras, lavavajillas, secadora, etc) a los momentos pico de generación de energía, facilitando así aún más la integración de las energías renovables en nuestro día a día.

» *Plastico biodegradable*

Aunque no se ha empleado en este sistema, puede usarse (en vez de plástico ABS normal) plástico biodegradable para impresoras 3D, para que el impacto medioambiental sea menor.

Capítulo 32

Código abierto y hardware libre

Como el sistema domótico ha sido creado con elementos como Arduino y Raspberry, da la posibilidad de hacerlo de código abierto y de hardware libre. Esto significa que el usuario puede crear sus propios módulos e incluirlos sin problema alguno.

Todos los esquemas y código han sido publicados, y son accesibles por todo el mundo. Esto permite al supuesto comprador (si dispone de conocimientos de programación) modificar el código fuente dando así servicios que ni la empresa provee.

No sería necesario crear otro módulo, bastaría con mejorar los que ya están. Desoldar los componentes originales, añadir de mejores (más precisos, con más datos...) y modificar parte del código.

Con todos los esquemas y código, se puede montar des de cero; solo comprando los componentes. Permitiendo que cualquier persona pueda tener su sistema domótico, a bajo precio.

Estas son opciones que ningún sistema domótico actual ofrece, y esa es una gran ventaja para el sistema que este trabajo ofrece. Así pues, se ofrece una plataforma de sistema domótico sobre la cual el usuario puede hacer las mejoras que prefiera.

Puede consultarse el código y esquemas en la WEB
<https://github.com/rogermiranda1000/Dom-tica>.

Capítulo 33

Conclusiones generales

Después de todas las horas empleadas en el proyecto, se ha logrado crear el sistema domótico, la gestión del mismo, y la página WEB.

Aún así, hay factores que podrían mejorarse, los cuales se mencionarán a continuación.

» *Mejora de la seguridad*

Como ya se ha mencionado, el sistema domótico es vulnerable a ataques, pero puede solucionarse fácilmente.

» *Toma de transformador de 9V (corriente continua)*

Con esta mejora eliminas la obligación del usuario de comprobar las baterías de cada uno de los módulos, además de no generar residuos.

» *Eliminar las micro SDs*

Para esta mejora sería necesario configurar el router para que acepte los módulos sin necesidad de contraseña, pero una vez realizada la mejora no haría falta comprar las micro SDs ni su lector.

» *Más módulos*

Entre ellos se encontraría un módulo que interactuaría con aparatos infrarrojos (TV, aire acondicionado...), un módulo que indicaría las personas que hay en una habitación (dando la posibilidad de apagar las luces automáticamente cuando no haya nadie) y un módulo que ciega temporalmente a los ladrones (proporcionando así más seguridad).

A esto se le puede añadir una mejora del módulo enchufe: un sensor de corriente que permita ver el consumo.

» *Reconocimiento facial y de matrícula*

La cámara de Raspberry Pi permite realizar esta clase de mejoras. Ya se intentó hacer (algunas con éxito), pero faltó tiempo para implementarlo.

En definitiva, se han cumplido todos los objetivos generales de este trabajo, y espero que mi contribución sea útil y acerque la domótica a un mayor público, ya que los beneficios de la domótica son muchos (gestión energética, seguridad...) y su integración sería de gran ayuda.

Parte VII

Anexos

Anexo A

MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería publish-subscribe basado en el protocolo TCP/IP (MQTT, 2018). En otras palabras, hace posible enviar información inalámbricamente.

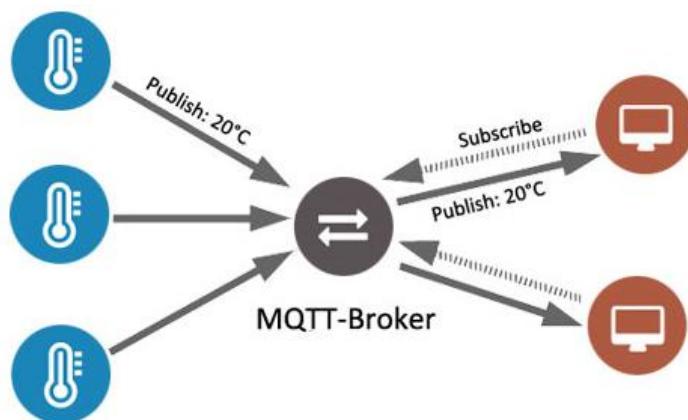


Figura A.1: Esquema de un sistema conectado con MQTT

Hay muchos *brokers*²⁵ MQTT disponibles, los principales son los siguientes:

- » *ActiveMQ*
- » *JoramMQ*
- » *Mosquitto* (*el usado en este proyecto*)
- » *RabbitMQ*

²⁵ “Intermediarios”

Anexo B

Código

Aquí se encuentra la parte de programación de este trabajo.

Para la creación de este sistema domótico han hecho falta conocimientos de varios lenguajes de programación. Arduino para las placas Arduino, Python para Raspberry Pi, y HTML, CSS y JavaScript para la WEB.

Para los que no tengan experiencia previa y estén interesados en entender el código (incluso mejorarlo si lo desean), puedo recomendarles algunos libros:

- » El gran libro de **Python**, de Marco Buttu, publicado el 2016 por Marcombo.
- » Robótica y domótica básica con **Arduino**, de Pedro Porcuna López, publicado el 2016 por Ra-Ma.
- » El gran libro de **HTML5, CSS3 y JavaScript**, de J.D Gauchat, publicado el 2017 por Marcombo.
- » Creación de un sitio web con **PHP y MySQL**, de Jacobo Pavón Puertas y Ezequiel Llarena Borges, publicado el 2015 por Ra-Ma.

Si ya conocen algún lenguaje de programación previo, con tutoriales WEB puede obtenerse un nivel bastante aceptable de programación con estos lenguajes.

B.1. Arduino

El código Arduino indicará a la placa (los módulos) lo que han de hacer, ya sea enviar, recibir o procesar información.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>

String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.10";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;

WiFiClient espClient;
PubSubClient client(espClient);

File myFile;

const int btn = D9;

void callback(char* topic, byte* payload, unsigned int length)
{ }

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect("btn01")) {
            Serial.println(" Conectado!");
            client.subscribe("btn");
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void setup() {
    pinMode(btn, INPUT);
    Serial.begin(9600);

    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
}

```

```
if (myFile) {
    Serial.println("SD:");
    int x = 0;
    while (myFile.available()) {
        String msgo = myFile.readStringUntil('\n');
        Serial.println(msgo);
        if(x==0) ssid = msgo;
        else if(x==1) password = msgo;

        x++;
    }
    myFile.close();
} else {
    Serial.println("Error en la lectura de la MicroSD.");
}

Serial.println("ssid: " + ssid);
Serial.println("pass: " + password);
cambiarDestino();
client.setCallback(callback);
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();

    if(digitalRead(btn) == HIGH) {
        Serial.println("Botón pulsado.");
        client.publish("central", "btn01");
        while(digitalRead(btn) != LOW) delay(1);
    }
}
```

B.1.2 Módulo movimiento

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>

String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.10";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;

WiFiClient espClient;
PubSubClient client(espClient);

File myFile;

const int movPin = D3;
bool alarma = false;

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    if(mensaje=="1") alarma = true;
    else if(mensaje=="0") alarma = false;
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect("MOV01")) {
            Serial.println(" Conectado!");
            client.subscribe("MOV");
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void setup() {
    pinMode(movPin, INPUT);
    Serial.begin(9600);
}

```

```
if (!SD.begin(4)) {
    Serial.println("Error en la lectura de la MicroSD.");
}

myFile = SD.open("send.txt");
if (myFile) {
    Serial.println("SD:");
    int x = 0;
    while (myFile.available()) {
        String msgo = myFile.readStringUntil('\n');
        Serial.println(msgo);
        if(x==0) ssid = msgo;
        else if(x==1) password = msgo;

        x++;
    }
    myFile.close();
} else {
    Serial.println("Error en la lectura de la MicroSD.");
}

Serial.println("ssid: " + ssid);
Serial.println("pass: " + password);
cambiarDestino();
client.setCallback(callback);
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();

    if(alarma) {
        if(digitalRead(movPin) == HIGH) {
            Serial.println("Movimiento");
            client.publish("central", "MOV01 movimiento MOV");
            while(digitalRead(movPin) != LOW) delay(1);
            Serial.println("Movimiento cesado");
        }
    }
}
```

B.1.3 Módulo apertura

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <DHT.h>

String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;

WiFiClient espClient;
PubSubClient client(espClient);

File myFile;

const int pinA = 5;

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect("MAG01")) {
            Serial.println(" Conectado!");
            client.subscribe("MAG01");
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void conectar() {
    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }
}

```

```

myFile = SD.open("send.txt");
if (myFile) {
    Serial.println("SD:");
    int x = 0;
    while (myFile.available()) {
        String msgo = myFile.readStringUntil('\n');
        Serial.println(msgo);
        if(x==0) ssid = msgo;
        else if(x==1) password = msgo;

        x++;
    }
    myFile.close();
} else {
    Serial.println("Error en la lectura de la MicroSD.");
}

ssid.remove(ssid.length()-1);
Serial.println("ssid: '" + ssid + "'");
Serial.println("pass: '" + password + "'");

Serial.print("Connecting to ");
Serial.print(ssid);
WiFi.mode(WIFI_STA);
WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

cambiarDestino();
client.setCallback(callback);
}

void setup() {
    Serial.begin(9600);
    pinMode(pinA, INPUT);

    conectar();
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();

    if(digitalRead(pinA)) {
        while(digitalRead(pinA)) delay(1);
        Serial.println("Abierto");
        client.publish("central", "MAG01 mag OPEN");
    }

    delay(10);
}

```

B.1.4 Módulo seguridad

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <DHT.h>
String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;
WiFiClient espClient;
PubSubClient client(espClient);
File myFile;
char* ID = "SEG01";

const int fuego = D0;
const int MQ2 = D1;
const int MQ5 = D2;
const int MQ7 = D3;
const int buzzer = D5;
const int MQ135 = A0;

bool alarma = false;
int tiempo = 0;

void setup() {
  Serial.begin(9600);
  pinMode(MQ2, OUTPUT);
  pinMode(MQ5, OUTPUT);
  pinMode(MQ7, OUTPUT);
  pinMode(MQ135, OUTPUT);

  conectar();
}

void loop() {
  if (!client.connected()) reconnect();
  client.loop();

  if(digitalRead(MQ2)==1 || digitalRead(MQ5)==1 ||
digitalRead(MQ7)==1) client.publish("central", (const char*)
(String(ID) + " gas 1").c_str());
  if(digitalRead(fuego)==1) client.publish("central", (const
char*) (String(ID) + " fuego").c_str());

  int gas = analogRead(MQ135);
  if(gas>=5) client.publish("central", (const char*)
(String(ID) + " gas " + String(gas)).c_str());

  if(alarma) {
    tiempo++;
    if((tiempo%100)<=50) tone(buzzer, 1000);
    else noTone(buzzer);
}

```

```

        }
    }

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    if(mensaje=="act") alarma = true;
    else if(mensaje=="deact") alarma = false;
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect(ID)) {
            Serial.println(" Conectado!");
            client.subscribe(ID);
            client.publish("central", (const char*) (String(ID) + "?").c_str());
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void conectar() {
    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
    if (myFile) {
        Serial.println("SD:");
        int x = 0;
        while (myFile.available()) {
            String msgo = myFile.readStringUntil('\n');
            Serial.println(msgo);
            if(x==0) ssid = msgo;
            else if(x==1) password = msgo;

            x++;
        }
    }
}

```

```
    myFile.close();
} else {
    Serial.println("Error en la lectura de la MicroSD.");
}

ssid.remove(ssid.length()-1);
Serial.println("ssid: '" + ssid + "'");
Serial.println("pass: '" + password + "'");

Serial.print("Connecting to ");
Serial.print(ssid);
WiFi.mode(WIFI_STA);
WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

cambiarDestino();
client.setCallback(callback);
}
```

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <DHT.h>

String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;

WiFiClient espClient;
PubSubClient client(espClient);

File myFile;

int retraso = 1000;
int pasado = 0;
char* ID = "CLIM01";

const int DHTPIN = 5;
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const int pResistor = A0;

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    Serial.println("Retraso de " + mensaje + " segundos.");
    retraso = mensaje.toInt();
    retraso *= 1000;
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect(ID)) {
            Serial.println(" Conectado!");
            client.subscribe(ID);
            client.publish("central", (const char*) (String(ID) + "?").c_str());
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

```

```

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void setup() {
    Serial.begin(9600);
    dht.begin();
    pinMode(pResistor, INPUT);

    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
    if (myFile) {
        Serial.println("SD:");
        int x = 0;
        while (myFile.available()) {
            String msgo = myFile.readStringUntil('\n');
            Serial.println(msgo);
            if(x==0) ssid = msgo;
            else if(x==1) password = msgo;

            x++;
        }
        myFile.close();
    } else {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    ssid.remove(ssid.length()-1);
    Serial.println("ssid: '" + ssid + "'");
    Serial.println("pass: '" + password + "'");

    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    cambiarDestino();
    client.setCallback(callback);
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();
}

```

```
// Leemos la humedad relativa
float h = dht.readHumidity();
// Leemos la temperatura en grados centígrados (por defecto)
float t = dht.readTemperature();
// Comprobamos si ha habido algún error en la lectura
if (isnan(h) || isnan(t)) {
    Serial.println("Error obteniendo los datos del sensor
DHT11");
    return;
}
Serial.println(String(h) + "%");
Serial.println(String(t) + "°C");

int value = analogRead(pResistor);
double per = (double) value/1024;
per *= 100;
Serial.print(String(value) + " - ");
Serial.print(per, 2);
Serial.println("% luz");
Serial.println("----");

pasado += 500;
if(pasado >= retraso) {
    client.publish("central", (const char*) (String(ID) + " luz
" + String(per)).c_str());
    client.publish("central", (const char*) (String(ID) + "
temperatura " + String(t)).c_str());
    client.publish("central", (const char*) (String(ID) + "
humedad " + String(h)).c_str());
    pasado = 0;
}

delay(500);
}
```

```

//El sensor de humedad y temperatura se rompió.
//#include <DHT.h>

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <DHT.h>
String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;
WiFiClient espClient;
PubSubClient client(espClient);
File myFile;
int retraso = 1000;
int pasado = 0;
char* ID = "CLIM02";

const int luz = D0;
const int agua = D1;
const int sensores = A0;

//#define DHTPIN D4
//#define DHTTYPE DHT11
//DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  pinMode(luz, OUTPUT);
  pinMode(agua, OUTPUT);
  pinMode(sensores, INPUT);
  //dht.begin();

  conectar();
}

void loop() {
  if (!client.connected()) reconnect();
  client.loop();

  /*float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.println("Humedad: " + String(h) + "%");
  Serial.println("Temperatura: " + String(t) + "°C"); */

  //lectura nivel del agua
  digitalWrite(agua, HIGH);
  delay(400);
  int a = analogRead(sensores);
  Serial.println("Agua: " + String(a));
  delay(100);
}

```

```

digitalWrite(agua, LOW);

//lectura luz
digitalWrite(luz, HIGH);
delay(400);
float l = analogRead(sensores);
l /= (float) 1023;
l *= (float) 100;
Serial.println("Luz: " + String(l) + "%");
delay(100);
digitalWrite(luz, LOW);

pasado += 1000;
if(pasado >= retraso) {
    client.publish("central", (const char*) (String(ID) + " luz
" + String(l)).c_str());
    client.publish("central", (const char*) (String(ID) + "
agua " + String(a)).c_str());
    pasado = 0;
}
}

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    Serial.println("Retraso de " + mensaje + " segundos.");
    retraso = mensaje.toInt();
    retraso *= 1000;
}

void reconnect() {
while (!client.connected()) {
    Serial.print("Conectando...");
    if (client.connect(ID)) {
        Serial.println(" Conectado!");
        client.subscribe(ID);
        client.publish("central", (const char*) (String(ID) + "?").c_str());
    } else {
        Serial.print("fallo, rc=");
        Serial.print(client.state());
        Serial.println(" intentado otra vez en 2 segundos...");
        cambiarDestino();
        delay(2000);
    }
}
}

void cambiarDestino() {
dest = !dest;
if(dest) client.setServer(mqtt_server, 1883);
}

```

```

    else client.setServer(mqtt_server2, 1883);
}

void conectar() {
    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
    if (myFile) {
        Serial.println("SD:");
        int x = 0;
        while (myFile.available()) {
            String msgo = myFile.readStringUntil('\n');
            Serial.println(msgo);
            if(x==0) ssid = msgo;
            else if(x==1) password = msgo;

            x++;
        }
        myFile.close();
    } else {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    ssid.remove(ssid.length()-1);
    Serial.println("ssid: '" + ssid + "'");
    Serial.println("pass: '" + password + "'");

    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    cambiarDestino();
    client.setCallback(callback);
}

```

B.1.7 Módulo alarma

```

#include <Keypad.h>
#include "FPS_GT511C3.h"
#include "TM1637.h"

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SD.h>String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;
WiFiClient espClient;
PubSubClient client(espClient);
File myFile;
char* ID = "ALAR01";

int pasado = 0;
bool activado = false;

const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {13, 12, 5, 4};
byte colPins[COLS] = {3, 2, 1, 0};
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins,
colPins, ROWS, COLS);

FPS_GT511C3 fps(10, 11);

#define CLK 8
#define DIO 7
TM1637 tm1637(CLK,DIO);

const int buzzer = D6;

char password[4] = { ' ', ' ', ' ', ' ' };
bool alarma = false;
bool anadiendo = false;
//a = alarma; d = desactivar alarma; h = huella; m = añadir
huella; n = añadir huella (password correcto)
char modo = ' ';
int tiempo = 0;

void setup(){
  Serial.begin(9600);
  //while(!Serial.available()) delay(1);

  fps.Open();
}

```

```

tm1637.init();
tm1637.set(BRIGHT_TYPICAL);

pinMode(buzzer, OUTPUT);

conectar();
}

void loop() {
  if (!client.connected()) reconnect();
  client.loop();

  if(activado) {
    pasado++;
    if(pasado%100<=50) tone(buzzer, 1000);
    else noTone(buzzer);
  }

  char customKey = customKeypad.getKey();

  if (customKey) {
    //Serial.println(customKey);
    if(customKey=='*') {
      reset();
      modo = 'h';
      fps.SetLED(true);
    }
    else if(customKey=='#') reset();
    else if(customKey=='A') {
      reset();
      modo = 'a';
      tiempo = 0;
      client.publish("central", (const char*) (String(ID) + "alarma act").c_str());
      for(int x = 0; x<4; x++) {
        tone(buzzer, 1000);
        delay(400);
        noTone(buzzer);
        delay(400);
      }
    }
    else if(customKey=='B') {
      reset();
      modo = 'd';
    }
    else if(customKey=='C');
    else if(customKey=='D') {
      reset();
      modo = 'm';
      fps.SetLED(true);
    }
    else {
      if(modo == 'd' || modo == 'm') {
        for(int x = 0; x!=3; x++) password[x] = password[x+1];
        password[3] = customKey;
      }
    }
  }
}

```

```

//Debug
Serial.print("Password: ");
String pass = "";
for(int x = 0; x!=4; x++) {
    pass+=String(password[x]);
    tm1637.display(x, (int8_t) password[x]);
}
Serial.println(pass);

if(password[x]!=' ') {
    String aci = "deact";
    if(modo=='m') aci = "enr";
    client.publish("central", (const char*) (String(ID) +
" alarma "+aci+" "+pass).c_str());
}
}

if(modo == 'a');
else if(modo == 'h') {
    if (fps.IsPressFinger())
    {
        fps.CaptureFinger(false);
        int id = fps.Identify1_N();
        if (id <200) {
            //Huella válida
            client.publish("central", (const char*) (String(ID) + " alarma deact").c_str());
            modo = ' ';
        }
    }
}
else if(modo == 'n') {
    anadiendo = false;
    Enroll();
}
}

void reset() {
for(int x = 0; x!=4; x++) {
    password[x] = ' ';
    tm1637.display(x, ' ');
}
if(modo=='m' || modo=='h') {
    fps.SetLED(false);
}
}

void Enroll()
{
    if(anadiendo == true) return;
    anadiendo = true;
    // find open enroll id
    int enrollid = 0;
}

```

```

bool usedid = true;
while (usedid == true)
{
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
}
fps.EnrollStart(enrollid);

// enroll
Serial.print("Press finger to Enroll #");
for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 1);
Serial.println(enrollid);
while(fps.IsPressFinger() == false) delay(100);
bool bret = fps.CaptureFinger(true);
int iret = 0;
if (bret != false)
{
    Serial.println("Remove finger");
    for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 0);
    fps.Enroll1();
    while(fps.IsPressFinger() == true) delay(100);
    Serial.println("Press same finger again");
    for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 2);
    while(fps.IsPressFinger() == false) delay(100);
    bret = fps.CaptureFinger(true);
    if (bret != false)
    {
        Serial.println("Remove finger");
        for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 0);
        fps.Enroll2();
        while(fps.IsPressFinger() == true) delay(100);
        Serial.println("Press same finger yet again");
        for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 3);
        while(fps.IsPressFinger() == false) delay(100);
        bret = fps.CaptureFinger(true);
        if (bret != false)
        {
            Serial.println("Remove finger");
            for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 0);
            iret = fps.Enroll3();
            if (iret == 0)
            {
                Serial.println("Enrolling Successful");
            }
            else
            {
                Serial.print("Enrolling Failed with error code:");
                Serial.println(iret);
                for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t)
9);
            }
        }
    }
    else Serial.println("Failed to capture third finger");
    for(int x = 0; x!=4; x++) tm1637.display(x,(int8_t) 9);
}
else Serial.println("Failed to capture second finger");

```

```

        for(int x = 0; x!=4; x++) tm1637.display(x, (int8_t) 9);
    }
    else Serial.println("Failed to capture first finger");
    for(int x = 0; x!=4; x++) tm1637.display(x, (int8_t) 9);
}

void activar() {
    activado = true;
}

void desactivar() {
    activado = false;
    for(int x = 0; x<3; x++) {
        tone(buzzer, 1000);
        delay(200);
        noTone(buzzer);
        delay(200);
    }
}

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    if(mensaje=="act") activar();
    else if(mensaje=="deact") desactivar();
    else if(mensaje=="enr") Enroll();
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect(ID)) {
            Serial.println(" Conectado!");
            client.subscribe(ID);
            client.publish("central", (const char*) (String(ID) + "?").c_str());
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

void conectar() {
    if (!SD.begin(4)) {

```

```
    Serial.println("Error en la lectura de la MicroSD.");
}

myFile = SD.open("send.txt");
if (myFile) {
    Serial.println("SD:");
    int x = 0;
    while (myFile.available()) {
        String msgo = myFile.readStringUntil('\n');
        Serial.println(msgo);
        if(x==0) ssid = msgo;
        else if(x==1) password = msgo;

        x++;
    }
    myFile.close();
} else {
    Serial.println("Error en la lectura de la MicroSD.");
}

ssid.remove(ssid.length()-1);
Serial.println("ssid: '" + ssid + "']");
Serial.println("pass: '" + password + "'");

Serial.print("Connecting to ");
Serial.print(ssid);
WiFi.mode(WIFI_STA);
WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

cambiarDestino();
client.setCallback(callback);
}
```

B.1.8 Módulo enchufe

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>

String ssid = "";
String password = "";
char* mqtt_server = "192.168.1.230";
char* mqtt_server2 = "192.168.1.229";
bool dest = false;
const char* cliente = "INT01";

const int rele = D0;

WiFiClient espClient;
PubSubClient client(espClient);

File myFile;

const int btn = D9;

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];

    if(mensaje=="1") digitalWrite(rele, HIGH);
    else if(mensaje=="0") digitalWrite(rele, LOW);
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Conectando...");
        if (client.connect(cliente)) {
            Serial.println(" Conectado!");
            client.subscribe(cliente);
            client.publish("central", (const char*) (String(cliente)
+ " enchufe").c_str());
        } else {
            Serial.print("fallo, rc=");
            Serial.print(client.state());
            Serial.println(" intentado otra vez en 2 segundos...");
            cambiarDestino();
            delay(2000);
        }
    }
}

void cambiarDestino() {
    dest = !dest;
    if(dest) client.setServer(mqtt_server, 1883);
    else client.setServer(mqtt_server2, 1883);
}

```

```

}

void setup() {
    pinMode(btn, INPUT);
    Serial.begin(9600);
    pinMode(rele, OUTPUT);
    digitalWrite(rele, LOW);

    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
    if (myFile) {
        Serial.println("SD:");
        int x = 0;
        while (myFile.available()) {
            String msgo = myFile.readStringUntil('\n');
            Serial.println(msgo);
            if(x==0) ssid = msgo;
            else if(x==1) password = msgo;

            x++;
        }
        myFile.close();
    } else {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    ssid.remove(ssid.length()-1);
    Serial.println("ssid: " + ssid);
    Serial.println("pass: " + password);

    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    cambiarDestino();
    client.setCallback(callback);
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();
}

```

B.1.9 Módulo riego

```

#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

int retraso = 1000;
char* ID = "RIEGO1";

//Pines
int moisture1 = A0;
int moisture2 = A1;
int moisture3 = A2;
int moisture4 = A3;
//Relays
int relay1 = 3;
int relay2 = 4;
int relay3 = 5;
int relay4 = 6;
bool r1 = false;
bool r2 = false;
bool r3 = false;
bool r4 = false;

// set water pump
int pump = 2;

#define DHTPIN 2
#define DHTTYPE DHT11
// Dirección MAC
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// IP del servidor
IPAddress mqtt_server(192, 168, 1, 229);
IPAddress mqtt_server2(192, 168, 1, 230);
bool dest = false;
EthernetClient ethClient;
PubSubClient client(ethClient);

void setup() {
  Serial.begin(9600);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);
  pinMode(relay4, OUTPUT);
  pinMode(pump, OUTPUT);

  if (Ethernet.begin(mac) == 0) Serial.println("Error al
configurar el Ethernet");
  cambiarDestino();
  client.setCallback(callback);
}

//Mensajes
void callback(char* topic, byte* payload, unsigned int length)
{

```

```

String mensaje = "";
for (int i=0;i<length;i++) mensaje += (char)payload[i];
if(mensaje[0]!='m') {
    Serial.println("Retraso de " + mensaje + " segundos.");
    retraso = mensaje.toInt();
    retraso *= 1000;
}
else {
    Serial.println("Mensaje: "+mensaje);
    //Mensaje = encender?
    if(mensaje=="me1") {
        digitalWrite(relay1, HIGH);
        r1=true;
    }
    else if(mensaje=="me2") {
        digitalWrite(relay2, HIGH);
        r2=true;
    }
    else if(mensaje=="me3") {
        digitalWrite(relay3, HIGH);
        r3=true;
    }
    else if(mensaje=="me4") {
        digitalWrite(relay4, HIGH);
        r4=true;
    }
    //Mensaje = apagar?
    else if(mensaje=="ma1") {
        digitalWrite(relay1, LOW);
        r1=false;
    }
    else if(mensaje=="ma2") {
        digitalWrite(relay2, LOW);
        r2=false;
    }
    else if(mensaje=="ma3") {
        digitalWrite(relay3, LOW);
        r3=false;
    }
    else if(mensaje=="ma4") {
        digitalWrite(relay4, LOW);
        r4=false;
    }
}

if(r1 || r2 || r3 || r4) digitalWrite(pump, HIGH);
else digitalWrite(pump, LOW);
}

void loop() {
    if (!client.connected()) reconnect();
    client.loop();

    int moisture1_value = analogRead(moisture1);
    int moisture2_value = analogRead(moisture2);
    int moisture3_value = analogRead(moisture3);
}

```

```

int moisture4_value = analogRead(moisture4);
Serial.println("Enviando datos...");
Serial.println("M1V: "+String(moisture1_value));
Serial.println("M2V: "+String(moisture2_value));
Serial.println("M3V: "+String(moisture3_value));
Serial.println("M4V: "+String(moisture4_value));
Serial.println("");

// Envio
client.publish("central", (const char*) (String(ID) + "humedad1 " + String(moisture1_value)).c_str());
client.publish("central", (const char*) (String(ID) + "humedad2 " + String(moisture2_value)).c_str());
client.publish("central", (const char*) (String(ID) + "humedad3 " + String(moisture3_value)).c_str());
client.publish("central", (const char*) (String(ID) + "humedad4 " + String(moisture4_value)).c_str());
delay(retraso);
}

void reconnect() {
while (!client.connected()) {
Serial.print("Conectando...");
if (client.connect(ID)) {
Serial.println(" Conectado!");
client.subscribe(ID);
client.publish("central", (const char*) (String(ID) + "?").c_str());
} else {
Serial.print("fallo, rc=");
Serial.print(client.state());
Serial.println(" intentado otra vez en 2 segundos...");
cambiarDestino();
delay(2000);
}
}
}

void cambiarDestino() {
dest = !dest;
if(dest) client.setServer(mqtt_server, 1883);
else client.setServer(mqtt_server2, 1883);
}

```

B.1.10 Módulo estación meteorológica

Este código es el mismo que el código de ejemplo, añadiendo la lectura de la MicroSD y la comunicación MQTT (consultar el código de los demás módulos).

B.1.11 Módulo LED

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <DHT.h>
String ssid = "";
String password = "";
const char* mqtt_server = "192.168.1.230";
const char* mqtt_server2 = "192.168.1.229";
bool dest = false;
WiFiClient espClient;
PubSubClient client(espClient);
File myFile;
const char* cliente = "LED01";

const int WHITEPIN = D1;
const int REDPIN = D3;
const int GREENPIN = D8;
const int BLUEPIN = D4;
int FADESPEED = 5; // make this higher to slow down
bool anim = false;
int proceso = 0;

void callback(char* topic, byte* payload, unsigned int length)
{
    String mensaje = "";
    for (int i=0;i<length;i++) mensaje += (char)payload[i];
    //Serial.println(mensaje);

    String msg[3] = {"", "", ""};
    int msg_val = 0;
    int pos;
    msg[0] = mensaje;
    while((pos = mensaje.indexOf (' ')) != -1) {
        msg[msg_val] = mensaje.substring(0, pos);
        msg_val++;
        mensaje = mensaje.substring (pos+1, mensaje.length());
    }
    msg[msg_val] = mensaje;
    Serial.println(msg[0] + ";" + msg[1] + ";" + msg[2]);

    if(msg[0] == "anim") {
        anim = true;
    }
}
```

```

proceso = 0;
digitalWrite(REDPIN, LOW);
digitalWrite(GREENPIN, LOW);
digitalWrite(BLUEPIN, HIGH);
}
else if(msg[0] == "animS") anim = false;
else if (msg[0] == "off") {
digitalWrite(WHITEPIN, LOW);
digitalWrite(REDPIN, LOW);
digitalWrite(GREENPIN, LOW);
digitalWrite(BLUEPIN, LOW);
}
else if (msg[0] == "on") {
digitalWrite(WHITEPIN, HIGH);
digitalWrite(REDPIN, HIGH);
digitalWrite(GREENPIN, HIGH);
digitalWrite(BLUEPIN, HIGH);
}
else if (msg[0] == "delay") {
FADESPEED = msg[1].toInt();
}
else if(msg[0] == "set") {
int num = msg[2].toInt();
int pin = 0;
if(msg[1]== "white") pin = WHITEPIN;
else if(msg[1]== "red") pin = REDPIN;
else if(msg[1]== "green") pin = GREENPIN;
else if(msg[1]== "blue") pin = BLUEPIN;
analogWrite(pin, num);
}
}

void reconnect() {
while (!client.connected()) {
Serial.print("Conectando...");
if (client.connect(cliente)) {
Serial.println(" Conectado!");
client.subscribe(cliente);
client.publish("central", (const char*) (String(cliente)
+ " led").c_str());
} else {
Serial.print("fallo, rc=");
Serial.print(client.state());
Serial.println(" intentado otra vez en 2 segundos...");
cambiarDestino();
delay(2000);
}
}
}

void cambiarDestino() {
dest = !dest;
if(dest) client.setServer(mqtt_server, 1883);
else client.setServer(mqtt_server2, 1883);
}

```

```

void conectar() {
    if (!SD.begin(4)) {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    myFile = SD.open("send.txt");
    if (myFile) {
        Serial.println("SD:");
        int x = 0;
        while (myFile.available()) {
            String msgo = myFile.readStringUntil('\n');
            Serial.println(msgo);
            if(x==0) ssid = msgo;
            else if(x==1) password = msgo;

            x++;
        }
        myFile.close();
    } else {
        Serial.println("Error en la lectura de la MicroSD.");
    }

    ssid.remove(ssid.length()-1);
    Serial.println("ssid: '" + ssid + "'");
    Serial.println("pass: '" + password + "'");

    Serial.print("Connecting to ");
    Serial.print(ssid);
    WiFi.mode(WIFI_STA);
    WiFi.begin((const char*) ssid.c_str(), (const char*)
password.c_str());
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    cambiarDestino();
    client.setCallback(callback);
}

void setup() {
    Serial.begin(9600);
    pinMode(WHITEPIN, OUTPUT);
    pinMode(REDPIN, OUTPUT);
    pinMode(GREENPIN, OUTPUT);
    pinMode(BLUEPIN, OUTPUT);

    digitalWrite(WHITEPIN, LOW);
    digitalWrite(REDPIN, LOW);
    digitalWrite(GREENPIN, LOW);
    digitalWrite(BLUEPIN, LOW);

    conectar();
}

```

```

void loop() {
    if (!client.connected()) reconnect();
    client.loop();

    if(anim) {
        if(++proceso==1) executeAnimationM(REDPIN);
        else if(proceso==2) executeAnimationL(BLUEPIN);
        else if(proceso==3) executeAnimationM(GREENPIN);
        else if(proceso==4) executeAnimationL(REDPIN);
        else if(proceso==5) executeAnimationM(BLUEPIN);
        else if(proceso==6) executeAnimationL(GREENPIN);
        else if(proceso==7) proceso = 0;
    }
}

void executeAnimationM(int PIN) {
    for (int c = 0; c < 256; c++) {
        analogWrite(PIN, c);
        delay(FADESPEED);
    }
}

void executeAnimationL(int PIN) {
    for (int c = 255; c > 0; c--) {
        analogWrite(PIN, c);
        delay(FADESPEED);
    }
}

```

B.2. Python

En este trabajo Python ha sido el “cerebro” del sistema domótico. Con una finalidad principal de procesar información y re-direccionar mensajes, es el código más extenso.

```

#!/usr/bin/python
import socket
import fcntl
import struct

import time
import paho.mqtt.client as mqtt
import paho.mqtt.publish as publish

import MySQLdb

import datetime

from twilio.rest import Client
account_sid = '...' # Found on Twilio Console Dashboard
auth_token = '...' # Found on Twilio Console Dashboard

```

```

myPhone = '+34...' # Phone number you used to verify your Twilio
account
TwilioNumber = '+34...' # Phone number given to you by Twilio
client = Client(account_sid, auth_token)

import picamera

from lxml import html
import requests
city = open("lluvia.txt", "r")

modulos = []
mod_tip = []

db =
MySQLdb.connect("localhost", "phpmyadmin", "pass", "Domotica")
cursor = db.cursor()

def database(obtener, sql):
    try:
        cursor.execute(sql)

        if obtener == True:
            return cursor.fetchall()
        else:
            db.commit()
    except (MySQLdb.Error, MySQLdb.Warning) as e:
        db.rollback()
        print "DB FAIL"
        print e

def tiempo(t):
    print "Tiempo limite, actualizando... [" + t + "]"
    try:
        results = database(True, "SELECT ind,Val FROM Valor
WHERE Tiempo=\\"+t+"\\"")
        total = {}
        num = {}
        for row in results:
            ind = int(row[0])
            if not ind in total:
                total[ind] = 0
                num[ind] = 0
            total[ind] += row[1]
            num[ind] += 1
        database(False, "DELETE FROM Valor WHERE
Tiempo=\\"+t+"\\"")
        for indicador in total.keys():
            tmp = ''
            hora = datetime.datetime.now().minute
            total[indicador] /= num[indicador]
            if t == 's':
                tmp = 'm'
            elif t == 'm':
                tmp = 'h'
                hora = datetime.datetime.now().hour

```

```

        elif t == 'h':
            tmp = 'd'
            hora = datetime.datetime.now().day
        elif t == 'd':
            return

        ind = total[indicador]
        print str(indicador) + ":" + str(ind)

        database(False, "INSERT INTO
Valor(ind,Tiempo,Time,Val) VALUES
(" + str(indicador) + ",\\" + str(tmp) + "\",\"" + str(hora) + "\",\"" + str(ind) + ""
")")
    except (MySQLdb.Error, MySQLdb.Warning) as e:
        print "DB load error"
        print e

def enviar(ID, tip, valu):
    try:
        results = database(True, "SELECT ind FROM Tipos
WHERE ID=\"" + ID + "\" AND Tipo=\"" + tip + "\" AND RoA='r';")
        ind = 0
        for row in results:
            ind = row[0]
        database(False, "INSERT INTO
Valor(ind,Tiempo,Time,Val) VALUES
(" + str(ind) + ",\"s\",\"" + str(datetime.datetime.now().second) + "\",\"" + str(valu) + "\")")
    except (MySQLdb.Error, MySQLdb.Warning) as e:
        print "DB load error"
        print e

def get_ip_adress(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(
        s.fileno(),
        0x8915,
        struct.pack('256s', ifname[:15])
    )[20:24])

def ip():
    try:
        return get_ip_adress('eth0')
    except:
        pass

    try:
        return get_ip_adress('wlan0')
    except:
        pass
    return ""

def on_connect(client, userdata, flags, rc):
    #print("Conectado. " + str(rc))
    client.subscribe("central")

```

```

def on_message(client, userdata, msg):
    message = str(msg.payload)
    #print(msg.topic + ": " + message)
    msg = message.split(' ')
    print msg

    prefix = msg[0]
    tipo = msg[1]
    mensaje = ""
    if len(msg) >= 3:
        mensaje = msg[2]
    RoA = 'r'
    if tipo == "movimiento":
        RoA = 'a'
        if mensaje == "MOV":
            alarma()
    elif tipo == "alarma":
        RoA = 'a'
        if mensaje == "deact" or mensaje=="enr":
            results = database(True, "SELECT code FROM
Alarm;")
            for row in results:
                if(msg[0]=="" or
(msg[0]==str(row[0]))):
                    if mensaje == "deact":
                        client.publish(prefix,
"deact")
                        database(False, "UPDATE
Alarm SET status=0 WHERE 1;")
                    else:
                        client.publish(prefix,
"enr")
                elif mensaje == "act":
                    database(False, "UPDATE Alarm SET status=1
WHERE 1;")
    elif tipo == "gas":
        results = database(True, "SELECT ind FROM Tipos
WHERE ID=\""+prefix+"\\";" )
        for row in results:
            if mensaje=="1":
                RoA = 'a'
                gas(row2[0])
            else:
                results2 = database(True, "SELECT
maxVal FROM Alarma WHERE ind=\""+str(row[0])+"\"")
                for row2 in results2:
                    if mensaje>=row2[0]:
                        gas(row2[0])
    elif tipo == "fuego":
        RoA = 'a'
        fuego()
    elif tipo == "btn":
        results = database(True, "SELECT Ejecutar FROM
Botones WHERE Nombre=\""+mensaje+"\\";" )
        for row in results:

```

```

        print "Publicando '{0}' a
central...".format(row[0])
            client.publish("central", "btn " +
str(row[0]))
            return
        elif tipo == "mag":
            RoA = 'a'
            if mensaje == "OPEN":
                alarma()
        elif tipo == "led":
            RoA = 'a'
            results = database(True, "SELECT Tiempo FROM
Nombres WHERE ID=\\""+prefix+"\\";")
            for row in results:
                client.publish(prefix, "delay " +
str(row[0]))
                print "Delay LED: {0}".format(row[0])
        elif tipo == "enchufe":
            RoA = 'a'
            client.publish("central", "btn int "+prefix)
            client.publish("central", "btn int "+prefix)
        elif tipo == "int":
            try:
                results = database(True, "SELECT ind FROM
Tipos WHERE ID='"+mensaje+"';")
                for row in results:
                    results2 = database(True, "SELECT
Status FROM Enchufes WHERE ind='"+str(row[0])+"'")
                    status = True
                    for row2 in results2:
                        status = bool(row2[0])

                    status = not status

                    client.publish(mensaje, int(status))
                    database(False, "UPDATE Enchufes SET
Status="+str(int(status))+"" WHERE ind='"+str(row[0])+"';")
            except:
                print "DB load error"
            return
        elif tipo == "humedad1" or tipo == "humedad2" or tipo ==
"humedad3" or tipo == "humedad4":
            results2 = database(True, "SELECT
checkHumidity, humedad, checkRain, rangoCheck, probCheck FROM Riego
WHERE ID='"+mensaje+"';")
            for row2 in results2:
                if row2[0] == True and (row2[2] == False or
clima(row2[3], row2[4]) == True):
                    if row2[1]>mensaje:
                        client.publish(prefix,
'me'+tipo[7])
                else:
                    client.publish(prefix,
'ma'+tipo[7])
            if not prefix in modulos:

```

```

        database(False, "INSERT INTO
Riego(ID,checkRain,rangoCheck,probCheck,checkHumidity,humedad,f
orzarRiego,vecesPorDia,duracionRiego) VALUES
(\\""+prefix+"\\",1,24,60,1,40,0,0,5);")
    tipo = "humedadPlanta"+tipo[7]

    elif tipo == "WEB":
        if mensaje == "foto":
            print "Enviando foto a WEB..."
            foto("/var/www/html/img.jpg")
            print "Listo!"
            client.publish(prefix, "refresh")
        return
    elif tipo == "?":
        retraso = 1
        try:
            results = database(True, "SELECT Tiempo FROM
Nombres WHERE ID='"+prefix+"'")
            for row in results:
                retraso = row[0]
        except:
            print "DB load error"
        print "Retraso de " + prefix + ": " + str(retraso)
    + "s"
        client.publish(prefix, retraso)
        return

    if prefix == "btn":
        return

    if not prefix in modulos:
        print "Nuevo modulo ("+prefix+")"
        modulos.append(prefix)
        database(False, "INSERT INTO
Nombres(ID,Nombre,Tiempo) VALUES
(\\""+prefix+"\\",\\\""+prefix+"\\",5);")

        if not (prefix+" "+tipo) in mod_tip:
            print "Nuevo tipo ("+tipot+", de "+prefix+")"
            mod_tip.append(prefix+" "+tipo)
            try:
                results = database(True, "SELECT MAX(ind)
FROM Tipos")
                val = results[0][0]
                if val is None:
                    val = -1
                val += 1
                #print val
                database(False, "INSERT INTO
Tipos(ind,ID,Tipo,RoA) VALUES
(\"+str(val)+\",\\\""+prefix+"\\",\\\""+tipot+"\\",\\\""+RoA+"\");")
                if(tipo == "enchufe"):
                    database(False, "INSERT INTO
Enchufes(ind>Status) VALUES ("+str(val)+",1);")
                elif(tipo == "led"):
```

```

                database(False, "INSERT INTO
LED(ind,Status,R,G,B,W) VALUES
("+str(val)+",\"simp\",0,0,0,0);")
            elif(tipo == "alarma"):
                database(False, "INSERT INTO
Alarma(ind,maxVal,status) VALUES (" +str(val)+ ",600,0;")
            except (MySQLdb.Error, MySQLdb.Warning) as e:
                print "DB load error"
                print e

        if RoA == 'r':
            enviar(prefix, tipo, mensaje)

def gas(ind):
    database(False, "UPDATE Alarma SET gas=1 WHERE
ind=" +str(ind)+";")
    client.messages.create(
        to=myPhone,
        from_=TwilioNumber,
        body='Hay mucho gas en su casa.')

def fuego():
    results = database(True, "SELECT ind FROM Tipos WHERE
ID=\"" +prefix+ "\";")
    for row in results:
        database(False, "UPDATE Alarma SET fuego=1 WHERE
ind=" +str(row[0])+";")
        client.messages.create(
            to=myPhone,
            from_=TwilioNumber,
            body='Su casa esta ardiendo.')

def clima(rango, prob):
    page = requests.get(city.read(), timeout = 15.0)
    if page.status_code == 200:
        tree = html.fromstring(page.content)
        num = []

        bucle("0")
        bucle("1")

        for x in range(rango-1):
            num[x] = num[x][:-1]
            if num[x][0]!='<' and int(num[x]) >= prob:
                return True

    return False

def bucle(ind):
    new = ""
    x = 1
    no = False

    while no == False:
        new = insertar(ind, x)
        if new != "":

```

```

        num.append(new)
    else:
        no = True
    x+=1

def on_publish(mosq, obj, mid):
    print("mid: " + str(mid))

def alarma():
    client.messages.create(
        to=myPhone,
        from_=TwilioNumber,
        body='Movimiento en su casa.')

def foto(path):
    with picamera.PiCamera() as picx:
        #picx.start_preview()
        picx.capture(path)
        #picx.stop_preview()
        picx.close()

while ip() == "":
    time.sleep(1)

print "Modulos encontrados:"
try:
    results = database(True, "SELECT * FROM Nombres")
    for row in results:
        pre = row[0]
        print "{0}: {1}".format(pre, row[1])
    modulos.append(pre)
except:
    print "DB load error"

print "Tipos:"
try:
    results = database(True, "SELECT ID,Tipo FROM Tipos")
    for row in results:
        mod_tip.append(row[0] + row[1])
        print "{0}: {1}".format(row[0], row[1])
except:
    print "DB load error"

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect(ip(), 1883, 60)
client.loop_start()

while True:
    #client.publish("led1", "switch")

    tiemp = datetime.datetime.now()
    if tiemp.second == 0:
        tiempo('s')
        if tiemp.minute == 0:

```

```

        tiempo('m')
    if tiemp.hour == 0:
        tiempo('h')
    if tiemp.day == 1:
        tiempo('d')

time.sleep(1)

```

B.3. WEB

La WEB permite al usuario interactuar de forma cómoda con el sistema domótico. Los ficheros de las WEBS se dividen en varias carpetas, se indicará el nombre del archivo junto con la carpeta.

B.3.1 /index.php

```

<!DOCTYPE html>
<html lang="es">
<head>
<?php include 'files/php/index.php'; ?>
<link rel="stylesheet" href="files/css/misestilos.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT Valor FROM config WHERE
Name='refreshTime';";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<META HTTP-EQUIV=Refresh CONTENT=". $row["Valor"] . " '>";
    }
}
$conn->close();
?>
</head>
<body>
<header>

```

```

<?php include 'files/php/arriba.php'; ?>
</header>

<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$x = 0;
$sql = "SELECT ID,Nombre FROM Nombres;";
$result = $conn->query($sql);
$lineas = $result->num_rows;
if ($lineas > 0) {
    while($row = $result->fetch_assoc()) {
        $sql2 = "SELECT ind,Tipo,RoA FROM Tipos WHERE
ID='".$row["ID"]."';";
        $result2 = $conn->query($sql2);
        if (($result2->num_rows > 0)) {
            if($x % 2 == 0) echo "<div id='exterior'>";
            echo "<div id='modulos'>";
            echo "<b><u>".$row["Nombre"]."</u></b>";
            echo "<br>";

            while($row2 = $result2->fetch_assoc()) {
                if($row2["RoA"]=="r" &&
($row2["Tipo"]!="humedadPlanta1" &&
$row2["Tipo"]!="humedadPlanta2" &&
$row2["Tipo"]!="humedadPlanta3" &&
$row2["Tipo"]!="humedadPlanta4" && $row2["Tipo"]!="alarma")) {
                    echo ucfirst($row2["Tipo"]).": ";

                    $sql3 = "SELECT Val, MAX(Time) FROM
Valor WHERE Tiempo='s' AND ind='".$row2['ind']."' ";
                    $result3 = $conn->query($sql3);
                    if ($result3->num_rows > 0) {
                        while($row3 = $result3-
>fetch_assoc()) {
                            echo $row3["Val"];
                            echo "<br>";
                        }
                    }
                }
            }
        }
    }
}
else {
    if($row2["Tipo"]=="enchufe") {
        $sql3 = "SELECT Status FROM
Enchufes WHERE ind='".$row2['ind']."' ";
        $result3 = $conn->query($sql3);
}

```

```

        $status = "";
        if ($result3->num_rows > 0) {
            while($row3 = $result3-
>fetch_assoc()) {
                if($row3["Status"]==1)
                    $status = "Apagar";
                else $status =
"Encender";
            }
            echo "<button
onclick='inte(\"".$row["ID"]."\")'>".$status."</button>";
        }
    }
    else if($row2["Tipo"]=="alarma") {
        $sql3 = "SELECT status FROM
Alarm WHERE 1;";
        $result3 = $conn->query($sql3);
        $status = "";
        if ($result3->num_rows > 0) {
            while($row3 = $result3-
>fetch_assoc()) {
                if($row3["status"]==1)
                    $status = "Apagar";
                else $status =
"Encender";
                echo "<button
onclick='alarm(\"".$row["ID"]."','".$row3["status"]."')'>".$stat
us." alarma</button>";
            }
        }
        else if($row2["Tipo"]=="gas") {
            $sql3 = "SELECT
status,fuego,gas FROM Alarma WHERE ind=".$row2['ind'].";";
            $result3 = $conn->query($sql3);
            $status = False;
            if ($result3->num_rows > 0) {
                while($row3 = $result3-
>fetch_assoc()) {
                    if(($row3["fuego"]==1
|| $row3["gas"]==1) && !($_GET["ID"]==$row["ID"] &&
$_GET["stat"]=="apagarAlarma")) $status = True;
                    if(!($status)) echo
'Niveles correctos.';
                    else {
                        echo "<button
onclick='alarma(\"".$row["ID"]."','".$row2['ind']."'')'>Apagar
alarma</button>";
                    }
                    echo "<br>";
                    $v = $row3["maxVal"];
                }
            }
            if($_GET["ID"]==$row["ID"] && $_GET["stat"]=="setVal") $v =
$_GET["val"];
        }
    }
}

```

```

echo "Limite: <input
type='range' id='lim".$row["ID"]."'" min='5' max='1023'
value='".$sv."'/>
onchange='alChange(\"".$row['ID']."\", ".$row2['ind']).">" ;
}
}
//Riego
else
if($row2["Tipo"]=="humedadPlanta1") {
    $sql2 = "SELECT * FROM Riego
WHERE ID=\"".$row["ID"]."\";";
    $result2 = $conn->query($sql2);
    if ($result2->num_rows > 0) {
        while($row2 = $result2-
>fetch_assoc()) {
            //Lluvia
            echo '<input
id=\"'.$row["ID"].'R" type="checkbox"
onchange="riego(\"'.$row["ID"].'\", \'checkRain\')"';

            if (($row2["checkRain"]==1 || ($_GET["stat"]=="checkRain" &&
$_GET["val"]=="true")) && (( $_GET["stat"]=="checkRain" &&
$_GET["val"]!="false") || $_GET["stat"]!="checkRain")) {
                //True
                echo ' checked';
                echo '>Considerar
la lluvia';
                echo '<br>';
            }
            echo 'En un rango
de ';
            $seleccionado = "";
            $sql3 = "SELECT
rangoCheck FROM Riego WHERE ID=\"".$row['ID']."'\"";
            $result3 = $conn-
>query($sql3);
            if ($result3-
>num_rows > 0) {
                while($row3 =
$result3->fetch_assoc()) {
                    $seleccionado = $row3["rangoCheck"];
                }
            }
            if($_GET["rangoCheck"]!="" && $_GET["ID"]==$row['ID'])
                $seleccionado = $_GET["rangoCheck"];
                echo '<select
id="rangoCheck'.$row['ID'].'"
onchange="riego(\"'.$row['ID'].'\", \'rangoCheck\')>"';
                echo '<option
value="" disabled="disabled">Rango:>';
                for($i=1; $i<=48;
$i++) {

```

```

echo '<option
value="'. $i .'>';

if($seleccionado==$i) echo ' selected="selected"';
echo '>' . $i;
}
echo '</select>';

echo ' horas,';
echo '<br>';
echo 'con una
probabilidad del ';

$seleccionado = "";
$sql3 = "SELECT
probCheck FROM Riego WHERE ID=\\"". $row['ID'] ."\\";
$result3 = $conn-
>query($sql3);
if ($result3-
>num_rows > 0) {
while($row3 =
$result3->fetch_assoc()) {

$seleccionado = $row3["probCheck"];
}

if($_GET["probCheck"] !="" && $_GET["ID"]==$row['ID'])
$seleccionado = $_GET["probCheck"];
echo '<select
id="probCheck' . $row['ID'] . '"'
onchange="riego(\\"'. $row['ID'] .'\\', \\'probCheck\\')>';
echo '<option
value="" disabled="disabled">%:>';
for($i=1; $i<=9;
$i++) {
$a=$i*10;
echo '<option
value="'. $a .'>';

if($seleccionado==$a) echo ' selected="selected"';
echo '>' . $a;
}
echo '</select>';

echo '%.' ;
}
else {
echo '>Considerar
la lluvia';
}
echo '<br>';

//Humedad

```

```

echo '<input
id="'. $row["ID"] . 'H" type="checkbox"
onchange="riego(\'' . $row["ID"] . '\', \'checkHumidity\')"'';

if( ($row2["checkHumidity"]==1 || 
($_GET["stat"]=="checkHumidity" &&
$_GET["val"]=="true")) && (($_GET["stat"]=="checkHumidity" && $_GET
["val"]!="false") || $_GET["stat"]!="checkHumidity")) {
    //True
    echo ' checked';
    echo '>Considerar
la humedad de la tierra';
    echo '<br>';
    echo 'Regar si la
humedad es inferior al ';

$seleccionado = "";
$sql3 = "SELECT
humedad FROM Riego WHERE ID=\\"". $row['ID'] . "\\";
$result3 = $conn-
>query($sql3);
if ($result3-
>num_rows > 0) {
    while($row3 =
$result3->fetch_assoc()) {
        $seleccionado = $row3["humedad"];
    }
}

if($_GET["humedad"]!="" && $_GET["ID"]==$row['ID'])
$seleccionado = $_GET["humedad"];
echo '<select
id="humedad".' . $row['ID'] . '" onchange="riego(\'' . $row['ID'] . '\',
\'humedad\')"'>';
echo '<option
value="" disabled="disabled">%:>';
for($i=1; $i<=9;
$i++) {
    $a=$i*10;
    echo '<option
value="'. $a . '"';
}

if($seleccionado==$a) echo ' selected="selected"';
    echo '>' . $a;
}
echo '</select>';

echo '%.>';
}
else {
    echo '>Considerar
la humedad de la tierra';
}
echo '<br>';

```

```

//Forzar riego (en
desarrollo)
/*echo '<input
id="'. $row["ID"] . 'W" type="checkbox"
onchange="riego(\'' . $row["ID"] . '\', \'checkWater\')"';

if(($row2["forzarRiego"]==1 || ($_GET["stat"]=="checkWater" &&
$_GET["val"]=="true"))&&($(_GET["stat"]=="checkWater"&&$_GET["v
al"]!="false") || $_GET["stat"]!="checkWater")) {/*
//True
/*echo ' checked';
echo '>Forzar
riego';
echo '<br>';

$seleccionado = "";
$sql3 = "SELECT
vecesPorDia FROM Riego WHERE ID=\'' . $row['ID'] . '\';";
$result3 = $conn-
>query($sql3);
if ($result3-
>num_rows > 0) {
while($row3 =
$result3->fetch_assoc()) {

$seleccionado = $row3["vecesPorDia"];
}

}

if($_GET["vecesPorDia"]!="" && $_GET["ID"]==$row['ID']){
$seleccionado = $_GET["vecesPorDia"];
echo '<select
id="vecesPorDia' . $row['ID'] . '"
onchange="riego(\'' . $row['ID'] . '\', \'vecesPorDia\')"'>';
echo '<option
value="" disabled="disabled">Veces/día:';
for($i=1; $i<=4;
$i++) {
echo '<option
value="'. $i . '"';
if($seleccionado==$i) echo ' selected="selected"';
echo '>' . $i;
}
echo '</select>';

echo ' veces al
día,';

echo '<br>';
echo 'durante ';

$seleccionado = "";
$sql3 = "SELECT
duracionRiego FROM Riego WHERE ID=\'' . $row['ID'] . '\';";
$result3 = $conn-
>query($sql3);
}

```

```

if ($result3-
>num_rows > 0) {
                                while($row3 =
$result3->fetch_assoc()) {

$seleccionado = $row3["duracionRiego"];
}
}

if($_GET["duracionRiego"]!="") && $_GET["ID"]==$row['ID'])
$seleccionado = $_GET["duracionRiego"];
echo '<select
id="duracionRiego'.$row['ID'].'"
onchange="riego(\''.$row['ID'].'\', \'duracionRiego\')">';
echo '<option
value="" disabled="disabled">Minutos:>';
for($i=1; $i<=6;
$i++) {
$o = $i*5;
echo '<option
value="'. $o .'>';

if($seleccionado==$o) echo ' selected="selected"';
echo '>'. $o;
}
echo '</select>';

echo ' minutos,>';
echo '<br>';
echo 'A las:>';
echo '<br>';

$seleccionado = "";
$sql3 = "SELECT
vecesPorDia FROM Riego WHERE ID=\\"". $row['ID']. "\\";
$result3 = $conn-
>query($sql3);
if ($result3-
>num_rows > 0) {
                                while($row3 =
$result3->fetch_assoc()) {

$seleccionado = $row3["vecesPorDia"];
}
}

if($_GET["vecesPorDia"]!="") && $_GET["ID"]==$row['ID'])
$seleccionado = $_GET["vecesPorDia"];
for($i=0;
$i<intval($seleccionado); $i++) {
echo "> ";
echo "<br>";
} */
/*}
else echo '>Forzar
riego';

```

```

                echo '<br>';*/
            }
        }
    }
    else if($row2["Tipo"]=="led") {
        echo '<select id="LEDStatus
'. $row['ID']."' onchange="led(\''.$row['ID'].'\",
\''. $row2['ind']. '\')">';
        $seleccionado = "";
        $sql3 = "SELECT Status FROM LED
WHERE ind=\"$row2['ind'];\"";
        $result3 = $conn->query($sql3);
        if ($result3->num_rows > 0) {
            while($row3 = $result3-
>fetch_assoc()) {
                $seleccionado =
$row3["Status"];
            }
        }
        if($_GET["modo"] != "" &&
$_GET["ind"]==$row2['ind']) $seleccionado = $_GET["modo"];
        echo '<option value=""'
disabled="disabled">Modo:';
        if($seleccionado=="simp") echo
'<option value="simp" selected="selected">Simple';
        else echo '<option
value="simp">Simple';
        if($seleccionado=="anim") echo
'<option value="anim" selected="selected">Animación';
        else echo '<option
value="anim">Animación';
        if($seleccionado=="avan") echo
'<option value="avan" selected="selected">Avanzado';
        else echo '<option
value="avan">Avanzado';
        echo '</select>';

        if($seleccionado=="simp") echo
'<div style="display: inline;" id="simp">';
        else echo '<div style="display:
none;" id="simp">';
        echo "<br><br>";
        echo "<button
onclick='ledB(\"".$row["ID"]."\", \"on\", \"\",
0)'>Encender</button>";
        echo "<br>";
        echo "<button
onclick='ledB(\"".$row["ID"]."\", \"off\", \"\",
0)'>Apagar</button>";
        echo "</div>";
        if($seleccionado=="anim") echo
'<div style="display: inline;" id="anim">';
    }
}

```

```

        else echo '<div style="display:
none;" id="anim">';
        echo "<br><br>";
        echo "<button
onclick='ledB(\"".$row["ID"]."\",\"anim\", \"\",
0)'>Iniciar</button>";
        echo "<br>";
        echo "<button
onclick='ledB(\"".$row["ID"]."\",\"animS\", \"\",
0)'>Detener</button>";
        echo "</div>";
        if($seleccionado=="avan") echo
'<div style="display: inline;" id="avan">';
        else echo '<div style="display:
none;" id="avan">';
        echo "<br>";

        $r=0;
        $g=0;
        $b=0;
        $w=0;
        $sql3 = "SELECT R,G,B,W FROM
LED WHERE ind='".$row2['ind']."' ";
        $result3 = $conn->query($sql3);
        if ($result3->num_rows > 0) {
            while($row3 = $result3-
>fetch_assoc()) {
                if($_GET["color"]!="R")
                    $r =
                else $r =
                if($_GET["color"]!="G")
                    $g =
                else $g =
                if($_GET["color"]!="B")
                    $b =
                else $b =
                if($_GET["color"]!="W")
                    $w =
                else $w =
            }
        }

        echo "R: <input type='range'
id='R' min='0' max='255' value='".$r."'
onchange='ledB(\"".$row['ID']."\",\"set\",\"red\",
".$row2['ind'].")'>";
        echo "<br>";
        echo "G: <input type='range'
id='G' min='0' max='255' value='".$g."'
onchange='ledB(\"".$row['ID']."\",\"set\",\"green\",
".$row2['in
d'].")'>";
        echo "<br>";
    
```

```

                echo "B: <input type='range'
id='B' min='0' max='255' value='".$b."'
onchange='ledB(\"".$row['ID']. "\",\"set\",\"blue\",".$row2['ind
'].")'>";
                echo "<br>";
                echo "W: <input type='range'
id='W' min='0' max='255' value='".$w."'
onchange='ledB(\"".$row['ID']. "\",\"set\",\"white\",".$row2['in
'].")'>";
                echo "</div>";
            }
        }

        echo "</div>";
        if($x % 2 == 1) echo "</div>";
        $x++;
    }
}

$sql2 = "SELECT * FROM Botones";
$result2 = $conn->query($sql2);
$lineas2 = $result2->num_rows;
if ($lineas2 > 0) {
    while($row2 = $result2->fetch_assoc()) {
        if($x % 2 == 0) echo "<div id='exterior'>";
        echo "<div id='modulos'>";
        echo "<b><u>".$row2["Nombre"]."</u></b>";
        echo "<br>";
        echo "<button
onclick='exe(\"".$row2["Nombre"]."\")'>Ejecutar
\"".$row2["Ejecutar"]."\"</button>";

        echo "</div>";
        if($x % 2 == 1) echo "</div>";
        $x++;
    }
    if($lineas+$lineas2 == $x) echo "</div>";
}
$conn->close();
?>

<script type="text/javascript">
    var clientID = "ID-" + Math.round(Math.random() *
1000);
    var client = new Paho.MQTT.Client(window.location.host,
9001, clientID);
    console.log(window.location.host);

    client.connect({onSuccess:onConnect});

    function onConnect() {
        console.log("connected");
        client.subscribe(clientID);
    }

```

```

        function exe(txt) {
            var message = new Paho.MQTT.Message(clientID+" btn
"+txt);
            message.destinationName = "central";
            client.send(message);
        }
        function inte(txt) {
            var message = new Paho.MQTT.Message(clientID+" int
"+txt);
            message.destinationName = "central";
            client.send(message);
        }

        location.reload();
    }

    function alarma(ID, ind) {
        var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ID=' + ID + '&stat=apagarAlarma&ind=' + ind;
        window.history.pushState({path:newurl}, '',newurl);

        <?php
            $servername = "localhost";
            $username = "phpmyadmin";
            $password = "pass";
            $dbname = "Domotica";

            // Create connection
            $conn = new mysqli($servername, $username,
$password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn-
>connect_error);
            }

            $stt = $_GET["stat"];
            if($stt=="checkWater") $stt =
"forzarRiego";
            $sql = "UPDATE Alarma SET fuego=0,gas=0
WHERE ind=".$_GET["ind"]."";
            if(mysqli_query($conn, $sql)) echo
"/*Sí*/";
            else echo "/*No*/";

            $conn->close();
        ?>

        var message = new Paho.MQTT.Message("deact");
        message.destinationName = ID;
        client.send(message);

        location.reload();
    }

    function alChange(ID, ind) {

```

```

        var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ID=' + ID + '&stat=setVal&val=' + document.getElementById("lim" + ID)
.value + '&ind=' + ind;
        window.history.pushState({path:newurl}, '', newurl);

        <?php
            $servername = "localhost";
            $username = "phpmyadmin";
            $password = "pass";
            $dbname = "Domotica";

            // Create connection
            $conn = new mysqli($servername, $username,
$password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn-
>connect_error);
            }

            $stt = $_GET["stat"];
            if($stt=="checkWater") $stt = "forzarRiego";
            $sql = "UPDATE Alarma SET
maxVal=".$_GET["val"]." WHERE ind=".$_GET["ind"]."";
            if(mysqli_query($conn, $sql)) echo /*Si*/";
            else echo /*No*/";

            $conn->close();
        ?>

        location.reload();
    }

    function riego(ID, btn) {
        if(btn=="checkRain" || btn=="checkHumidity" ||
btn=="checkWater") {
            var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ID=' + ID + '&stat=' + btn + '&val=' + document.getElementById(ID+btn[5])
].checked;

            window.history.pushState({path:newurl}, '', newurl);
            <?php
                $servername = "localhost";
                $username = "phpmyadmin";
                $password = "pass";
                $dbname = "Domotica";

                // Create connection
                $conn = new mysqli($servername, $username,
$password, $dbname);
                // Check connection
                if ($conn->connect_error) {
                    die("Connection failed: " . $conn-
>connect_error);

```

```

        }

        $stt = $_GET["stat"];
        if($stt=="checkWater") $stt =
"forzarRiego";
        $sql = "UPDATE Riego SET
".$stt."=".$_GET["val"]." WHERE ID=\"".$_GET["ID"]."\";";
        if(mysqli_query($conn, $sql)) echo
/*Sí*/;
        else echo /*No*/;

        $conn->close();
    ?>

        location.reload();
}
else if(btn=="probCheck" || btn=="rangoCheck" ||
btn=="humedad" || btn=="vecesPorDia" || btn=="duracionRiego") {
    var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ID=' + ID + '&btn=' + btn + '&btn=' + document.getElementById(btn+I
D).value;

window.history.pushState({path:newurl}, '', newurl);
<?php
    $servername = "localhost";
    $username = "phpmyadmin";
    $password = "pass";
    $dbname = "Domotica";

        // Create connection
        $conn = new mysqli($servername, $username,
$password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn-
>connect_error);
        }

        $sql = "UPDATE Riego SET
".$_GET["btn"]."=".$_GET[$_GET["btn"]]." WHERE
ID=\"".$_GET["ID"]."\";";
        if(mysqli_query($conn, $sql)) echo
/*Sí*/;
        else echo /*No*/;

        $conn->close();
    ?>

        location.reload();
}
}

function alarm(ID, stat) {
    var msg = "deact";
    if(stat==0) msg = "act";
}

```

```

        var message = new Paho.MQTT.Message(ID+" "+msg);
        message.destinationName = "central";
        client.send(message);

        location.reload();
    }

    function led(ID, ind) {
        var g = document.getElementById("LEDStatus"+ID).value;
        var d1 = document.getElementById('simp');
        var d2 = document.getElementById('anim');
        var d3 = document.getElementById('avan');
        ledB(ID, "animS", "", 0);

        if(g=="simp") {
            d1.style.display = 'inline';
            d2.style.display = 'none';
            d3.style.display = 'none';
        }
        else if(g=="anim") {
            d1.style.display = 'none';
            d2.style.display = 'inline';
            d3.style.display = 'none';
        }
        else if(g=="avan") {
            d1.style.display = 'none';
            d2.style.display = 'none';
            d3.style.display = 'inline';
        }
    }

    var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ind=' + ind + '&modo=' + g;
    window.history.pushState({path:newurl}, '',newurl);
<?php
    $servername = "localhost";
    $username = "phpmyadmin";
    $password = "pass";
    $dbname = "Domotica";

    // Create connection
    $conn = new mysqli($servername, $username,
$password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn-
>connect_error);
    }

    $sql = "UPDATE LED SET
Status='". $_GET["modo"] ."\" WHERE ind=". $_GET["ind"] ."";
    if(mysqli_query($conn, $sql)) echo /*Sí*/";
    else echo /*No*/";

```

```

        $conn->close();
    ?>
    location.reload();
}

function ledB(ID, sel, color, ind) {
    var msg = sel;
    if(sel=="set") {
        var valor = 0;
        if(color=="red") valor =
document.getElementById('R').value;
        else if(color=="green") valor =
document.getElementById('G').value;
        else if(color=="blue") valor =
document.getElementById('B').value;
        else if(color=="white") valor =
document.getElementById('W').value;

        msg += (" " + color + " " + valor);
    }

    var message = new Paho.MQTT.Message(msg);
    message.destinationName = ID;
    client.send(message);

    if(sel=="set") {
        var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?ind=' + ind + '&color=' + color[0].toUpperCase() + '&valor=' + valor;

        window.history.pushState({path:newurl}, '', newurl);
        <?php
            if($_GET["color"]!="") {
                $servername = "localhost";
                $username = "phpmyadmin";
                $password = "pass";
                $dbname = "Domotica";

                // Create connection
                $conn = new mysqli($servername,
$username, $password, $dbname);
                // Check connection
                if ($conn->connect_error) {
                    die("Connection failed: " . $conn-
>connect_error);
                }

                $sql = "UPDATE LED SET
Status=\"avan\", ".$_GET["color"]."=".$_GET["valor"]."\" WHERE
ind=".$_GET["ind"]."";
                if(mysqli_query($conn, $sql)) echo
"/*Sí*/";
                else echo "/*No*/";

                $conn->close();
            }
        
```

```

        ?>
        location.reload();
    }
}

</script>

<footer>
    Última actualización: <?php echo date("h:i:sa");?>
</footer>
</body>
</html>

```

B.3.2 /login.php

```

<?php
    session_start();
    $_SESSION["Nombre"] = $_POST["nombre"];
    $_SESSION["Pass"] = $_POST["pass"];
    echo "<script> history.go(-1); </script>";
?>

```

B.3.3 /graph.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <?php include 'files/php/index.php'; ?>
    <script
    src="https://code.highcharts.com/highcharts.js"></script>
    <script
    src="https://code.highcharts.com/modules/exporting.js"></script>
    <script src="https://code.highcharts.com/modules/export-data.js"></script>
    <style>
        .highcharts-data-table {
            display: none;
        }
        .highcharts-credits {
            display: none;
        }

        select {
            margin: auto;
            margin-left: auto;
            margin-right: auto;
            padding: 4px;
            width: 45%;
            margin: 1%;
        }
    </style>

```

```

        @media (max-width: 600px) {
            select {
                width: 95%;
                margin-top: 2%;
            }
        }
    </style>
</head>
<body>
    <header>
        <?php include 'files/php/arriba.php'; ?>
    </header>

    <select id="mySelect" onchange="myFunction()">
        <option value="" selected="selected"
disabled="disabled">--Seleccione una gráfica--</option>
        <?php
            $servername = "localhost";
            $username = "phpmyadmin";
            $password = "pass";
            $dbname = "Domotica";

            // Create connection
            $conn = new mysqli($servername, $username,
$password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn-
>connect_error);
            }
            $sql = "SELECT Tipo FROM Tipos WHERE RoA='r' group
by Tipo;";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<option
value='".$ucfirst($row["Tipo"])."'">".$ucfirst($row["Tipo"])."</op
tion>";
                    array_push($ind, $row["Tipo"]);
                }
            }
            $conn->close();
        ?>
    </select>

    <select id="mySelectT" onchange="myFunction()">
        <option value="" selected="selected"
disabled="disabled">--Seleccione un periodo de tiempo--</option>
        <option value="s">60 s
        <option value="m">60 min
        <option value="h">24 h
        <option value="d">30 d
    </select>

    <div id="container" style="min-width: 310px; height: 400px;
margin: 0 auto; display: none;"></div>

```

```

<?php include 'files/php/GG.php'; ?>
<script>
    <?php if($_GET['g']==" " && $_GET['t']==" ") {
echo "</script></body></html>"; die; } ?>
    document.getElementById("container").style.display =
"block";
    var unidades = "";
    if(<?php echo "'.$_GET['g'].'"; ?>=="Temperatura")
unidades = '°C';
    else if(<?php echo "'.$_GET['g'].'"; ?>=="Humedad")
unidades = '%';
    else if(<?php echo "'.$_GET['g'].'"; ?>=="Presión")
unidades = 'hPa';

<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "SELECT ind, ID FROM Tipos WHERE
Tipo=".strtolower($_GET['g'])."';";
$result = $conn->query($sql);
$ind = array();
$IDs = array();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        array_push($ind, $row["ind"]);
        array_push($IDs, $row["ID"]);
    }
}

$nombre = array();
foreach($IDs as $x) {
    $sql = "SELECT Nombre FROM Nombres WHERE
ID=".$.x."';";
    $result = $conn->query($sql);
    while($row = $result->fetch_assoc()) {
        array_push($nombre, $row["Nombre"]);
    }
}

$TEXTO = array();
$tmp = 0;
if($_GET['t']=="s" || $_GET['t']=="m") $tmp = 60;
else if($_GET['t']=="h") $tmp = 24;
else if($_GET['t']=="d") $tmp = 30;

```

```

        foreach($ind as $x) {
            $val = array();
            $index = array_search($x, $ind);
            $sql = "SELECT Time,Val FROM Valor WHERE
ind=\"".$x."\" AND Tiempo=\"".$_GET['t']."' ORDER BY
Valor.Time ASC;";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                $z = 0;
                while($row = $result->fetch_assoc()) {
                    while($row["Time"] > $z) {
                        if($val[$z-1]== "") array_push($val,
0);
                        else array_push($val, $val[$z-1]);
                        //array_push($val, 0);
                        $z++;
                    }
                    if($row["Time"] == $z) {
                        array_push($val, $row["Val"]);
                        $z++;
                    }
                }
                while($z<$tmp) {
                    array_push($val, 0);
                    $z++;
                }
            }
            $valores = "";
            for($y = 0; $y != $tmp; $y++) $valores .=
$val[$y].", ";
            $valores = substr($valores, 0, -2);
            array_push($TEXTO, "{ name:
'".$nombre[$index]."', data: [\".$valores.\" ] }");
        }
        $conn->close();
    ?>

Highcharts.chart('container', {
    chart: {
        type: 'line'
    },
    title: {
        text: <?php echo "'.$_GET['g'].'" ; ?>
    },
    yAxis: {
        title: {
            text: <?php echo "'.$_GET['g'].'" ; ?>+' ('+unidades+') '
        }
    },
    xAxis: {
        title: {
            text: 'Tiempo ['+<?php echo "'.$_GET['t'].'" ; ?>+' ] '
        }
    },
    plotOptions: {

```

```

        line: {
            dataLabels: {
                enabled: true
            },
            enableMouseTracking: true
        }
    },
    series: [
        <?php foreach($TEXTO as $t) {
            echo $t;
            if(array_search($t, $TEXTO) !=(count($TEXTO)-1)) echo ", ";
        } ?>
    ]
});
</script>
</body>
</html>

```

B.3.4 /editar.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <?php include 'files/php/index.php'; ?>
    <link rel="stylesheet" href="files/css/misestilos.css">
</head>
<body>
    <header>
        <?php include 'files/php/arriba.php'; ?>
    </header>

    <div id="exterior">
        <div id="modulos">
            <form action="files/form/cambiarSend.php"
method="post">
                <b>Tiempo de envío</b><br><br>
                <select name="modulo" required>
                    <option value="todos">Todos</option>
                    <?php
                        $servername = "localhost";
                        $username = "phpmyadmin";
                        $password = "pass";
                        $dbname = "Domotica";

                        // Create connection
                        $conn = new mysqli($servername,
$username, $password, $dbname);
                        // Check connection
                        if ($conn->connect_error) {
                            die("Connection failed: " . $conn-
>connect_error);
                        }
                        $sql = "SELECT * FROM Nombres;";
                        $result = $conn->query($sql);

```

```

        if ($result->num_rows > 0) {
            while($row = $result-
>fetch_assoc()) {
                WHERE ID=""". $row["ID"] ."";;
                $sql2 = "SELECT RoA FROM Tipos
                $result2 = $conn->query($sql2);
                if ($result2->num_rows > 0) {
                    while($row2 = $result2-
>fetch_assoc()) {
                        if($row2["RoA"] == 'r')
echo "<option value='". $row["ID"] ."'. '$row["Nombre"] ."
(".$row["Tiempo"].") . '</option>'";
                    }
                }
            }
        ?>
        </select> > <input type="number" name="tiempo"
min="1" max="50" required> [s]
<br><br>
<input type="submit" value="Cambiar">
</form>
</div>
<div id="modulos">
    <form action="files/form/cambiarRefresh.php"
method="post">
        <b>Tiempo de refresco</b><br><br>
        <?php
            $sql = "SELECT Valor FROM config WHERE
Name='refreshTime';";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {
                    echo "<input type='number'
name='RefreshTime' min='1' max='300' value='". $row["Valor"] ."'"
required>";
                }
            }
        ?>
        [s]<br><br>
        <input type="submit" value="Cambiar">
    </form>
</div>
</div>

<div id="exterior">
    <div id="modulos">
        <form action="files/form/cambiarNombre.php"
method="post">
            <b>Cambiar nombre</b><br><br>
            <select name="modulo" required>
                <?php
                    $sql = "SELECT ID, Nombre FROM
Nombres;";
                    $result = $conn->query($sql);
                    if ($result->num_rows > 0) {

```

```

        while ($row = $result-
>fetch_assoc()) {
            $sql2 = "SELECT Tipo,RoA FROM
Tipos WHERE ID='".$row["ID"]."';
            $result2 = $conn->query($sql2);
            if ($result2->num_rows > 0) {
                while ($row2 = $result2-
>fetch_assoc()) {
                    if ($row2["RoA"] == 'r' ||

$row2["Tipo"] == "led") echo "<option
value='".$row["ID"]."'>".$row["Nombre"]."(
".$row["ID"].")</option>";
                }
            }
        }
    ?>
    </select> > <input type="text" name="nombre"
maxlength="20" required>
<br><br>
<input type="submit" value="Cambiar">
</form>
</div>
</div>

<div id="exterior">
    <div id="modulos">
        <form action="files/form/addBtn.php" method="post">
            <b>Añadir botón</b><br><br>
            Nombre: <input type="text" name="nombre"
maxlength="20" required><br>
            Comando a ejecutar: <input type="text"
name="ejecutar" maxlength="80" required>
            <br><br>
            <input type="submit" value="Añadir">
        </form>
    </div>
    <div id="modulos">
        <form action="files/form/eliminarBtn.php"
method="post">
            <b>Eliminar botón</b><br><br>
            <select name="nombre" required>
                <?php
                    $sql = "SELECT Nombre FROM Botones";
                    $result = $conn->query($sql);
                    if ($result->num_rows > 0) {
                        while ($row = $result-
>fetch_assoc()) {
                            echo "<option
value='".$row["Nombre"]."'>".$row["Nombre"]."</option>";
                        }
                    }
                ?>
            </select>
            <br><br>
            <input type="submit" value="Eliminar">
        </form>
    </div>
</div>

```

```

        </form>
    </div>
</div>
<div id="exterior">
    <div id="modulos">
        <form action="files/form/addUser.php"
method="post">
            <b>Añadir usuario</b><br><br>
            Nombre: <input type="text" name="newNombre"
maxlength="20" required><br>
            Contraseña: <input type="password"
name="newPassword" maxlength="80" required>
            <br><br>
            <input type="submit" value="Añadir">
        </form>
    </div>
    <div id="modulos">
        <form action="files/form/eliminarUser.php"
method="post">
            <b>Eliminar usuario</b><br><br>
            <?php
                $sql = "SELECT Nombre FROM Usuarios;";
                $result = $conn->query($sql);
                if ($result->num_rows > 0) {
                    if ($result->num_rows > 1) {
                        echo "<select name='nombre'>";
                    }
                    while ($row = $result-
>fetch_assoc()) {
                        echo "<option
value='".$row["Nombre"]."'>".$row["Nombre"]."</option>";
                    }
                    echo "</select>";
                    echo "<br><br>";
                    echo "<input type='submit'
value='Eliminar'>";
                }
                else {
                    echo "Ha de existir mínimo 1
usuario.";
                }
            }
            $conn->close();
        ?>
        </form>
    </div>
</div>
</body>
</html>

```

B.3.5 /camara.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <?php include 'files/php/index.php'; ?>
    <link rel="stylesheet" href="files/css/misestilos.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
</head>
<body>
    <header>
        <?php include 'files/php/arriba.php'; ?>
    </header>

    <br>
    <div style="width: 70%; height: 100%; margin: 0 auto;">
        <img src='img.jpg' id='img1' style="width: 100%; height: auto;">
    </div>

    <script type="text/javascript">
        var clientId = "ID-" + Math.round(Math.random() * 1000);
        var client = new Paho.MQTT.Client(window.location.host, 9001, clientId);
        console.log(window.location.host);

        client.onMessageArrived = onMessageArrived
        client.connect({onSuccess:onConnect});

        function onConnect() {
            console.log("connected");
            client.subscribe(clientId);
        }
        function show_me() {
            var message = new Paho.MQTT.Message(clientId+" WEB foto");
            message.destinationName = "central";
            client.send(message);
        }
        function onMessageArrived(msg) {
            var mensaje = msg.payloadString
            console.log(mensaje);
            if(mensaje=="refresh")
                document.getElementById("img1").src =
                    "img.jpg?"+Math.ceil(Math.random()*1000000);
        }
        window.setInterval(function(){
            show_me()
        }, 1000);
    </script>
</body>
</html>

```

B.3.6 /post.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-
mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
</head>
<body>

    <script type="text/javascript">
        var clientID = "ID-" + Math.round(Math.random() *
1000);
        var client = new Paho.MQTT.Client(window.location.host,
9001, clientID);
        console.log(window.location.host);

        client.connect({onSuccess:onConnect});

        function onConnect() {
            console.log("connected");
            client.subscribe(clientID);

            <?php
                //if($_GET['info'] != "" && $_GET['clase'] !=
 "") {
                    echo "var message = new
Paho.MQTT.Message(clientID+'.'+
".$_POST["clase"]/*$_GET['clase']*./".
".$_POST["info"]/*$_GET['info']*./");";
                    echo 'message.destinationName = "prueba";';
                    echo "client.send(message);";
                    /*
                    else echo "console.log('Sin post.');"*/
                }
            ?>
        }
    </script>
</body>
</html>

```

B.3.7 /files/php/GG.php

```

<script>
function myFunction() {
    var g = document.getElementById("mySelect").value;
    var t = document.getElementById("mySelectT").value;

    if(g != "" && t != "") {
        var newurl = window.location.protocol + "//" +
window.location.host + window.location.pathname +
'?g=' + g + '&t=' + t;
        window.history.pushState({path:newurl}, '', newurl);
    }
}

```

```

        location.reload();
    }
}

</script>

```

B.3.8 /files/php/index.php

```

<title>Casa domótica</title>
<meta charset="utf-8">
<!--Header-->
<link rel="stylesheet" href="files/css/barra.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="icon" type="image/png" href="domotica-ico.png" />

<?php
    function formulario() {
        echo "<center><br>";
        echo "<form action='login.php' method='post'>";
        echo "<b>Usuario: </b> <input type='text' name='nombre'
maxlength='20' required><br>";
        echo "<b>Contraseña: </b> <input type='password'
name='pass' maxlength='20' required><br><br>";
        echo "<input type='submit' value='Entrar'>";
        echo "</form>";
        echo "</center>";
        die();
    }

    session_start();
    if(!isset($_SESSION["Nombre"])) formulario();

    $servername = "localhost";
    $username = "phpmyadmin";
    $password = "pass";
    $dbname = "Domotica";

    // Create connection
    $conn = new mysqli($servername, $username, $password,
$dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT Pass FROM Usuarios WHERE
Nombre='".$_SESSION["Nombre"]."';";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            if(md5($_SESSION["Pass"])!=$row["Pass"]) {
                echo "<center><br>Usuario o contraseña
errónea.</center>";
            }
        }
    }
}

```

```

        $conn->close();
        formulario();
    }
}
else {
    echo "<center><br>Usuario o contraseña
errónea.</center>";
    $conn->close();
    formulario();
}

$conn->close();
?>

```

B.3.9 /files/php/index.php

```

<h1 style="padding-left:16px; margin-top:2px; margin-
bottom:2px;">Casa domótica</h1>
<div class="topnav" id="myTopnav">
    <?php $url = $_SERVER['PHP_SELF']; ?>
    <a href="index.php" <?php if(($url=='/index.php') ||
$url=='') echo 'class="active"'; ?>>Información</a>
    <a href="graph.php" <?php if($url=='/graph.php') echo
'class="active"'; ?>>Gráfica</a>
    <a href="camara.php" <?php if($url=='/camara.php') echo
'class="active"'; ?>>Cámara</a>
    <a href="editar.php" <?php if($url=='/editar.php') echo
'class="active"'; ?>>Editar</a>
    <a href="phpmyadmin">phpMyAdmin</a>
    <a href="javascript:void(0);" class="icon"
onclick="mFunction()">
        <i class="fa fa-bars"></i>
    </a>
</div>

<script>
    function mFunction() {
        var x = document.getElementById("myTopnav");
        if (x.className === "topnav") {
            x.className += " responsive";
        } else {
            x.className = "topnav";
        }
    }
</script>

```

B.3.10 /files/css/barra.css

```
body {
    margin: 0;
    font-family: Arial, Helvetica, sans-serif;
}

.topnav {
    overflow: hidden;
    background-color: #333;
}

.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 17px;
}

.topnav a:hover {
    background-color: #ddd;
    color: black;
}

.active {
    background-color: #4CAF50;
    color: white;
}

.topnav .icon {
    display: none;
}
@media screen and (max-width: 600px) {
    .topnav a:not(:first-child) {display: none;}
    .topnav a.icon {
        float: right;
        display: block;
    }
}
@media screen and (max-width: 600px) {
    .topnav.responsive {position: relative;}
    .topnav.responsive .icon {
        position: absolute;
        right: 0;
        top: 0;
    }
    .topnav.responsive a {
        float: none;
        display: block;
        text-align: left;
    }
}
```

B.3.11 /files/css/misestilos.css

```
#exterior {
    margin: auto;
    width: 95%;
}

#modulos {
    background-color: #333;
    color: white;
    display: inline-block;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
    height: 240px;
    margin: 1%;
    width: 46%;
    border-top: 6px solid #4CAF50;
    padding: 4px;
}

@media (max-width: 600px) {
    #modulos {
        width: 95%;
        margin-top: 2%;
    }
}

@media (min-width: 600px) {
    #exterior {
        display: flex;
    }
}
```

B.3.12 /files/form/login.php

```
<?php
    session_start();
    $_SESSION["Nombre"] = $_POST["nombre"];
    $_SESSION["Pass"] = $_POST["pass"];
    echo "<script> history.go(-1); </script>";
?>
```

B.3.13 /files/form/eliminarUser.php

```

<html>
<head>
    <?php include '../php/index.php'; ?>
    <style>
        .button {
            background-color: #4CAF50;
            border: none;
            color: white;
            padding: 15px 32px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 16px;
            margin: 4px 2px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <center>
        <?php
            $servername = "localhost";
            $username = "phpmyadmin";
            $password = "pass";
            $dbname = "Domotica";

            // Create connection
            $conn = new mysqli($servername, $username,
$password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn-
>connect_error);
            }

            $sql = "DELETE FROM Usuarios WHERE
Nombre='".$_POST["nombre"]."';"

            if(mysqli_query($conn, $sql)){
                echo "Se ha eliminado el usuario
".$_POST["nombre"]."";
            } else {
                echo "Error: " . mysqli_error($conn);
            }

            $conn->close();
        ?>
        <br>
        <a href="#" class="button">OK</a>
    </center>
</body>
</html>

```

```

<html>
<head>
<?php include '../php/index.php'; ?>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<center>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "DELETE FROM Botones WHERE
Nombre='".$_POST["nombre"]."';"

if(mysqli_query($conn, $sql)){
    echo "Se ha eliminado el botón
".$_POST["nombre"]."";
} else {
    echo "Error: " . mysqli_error($conn);
}

$conn->close();
?>
<br>
<a href="#" class="button">OK</a>
</center>
</body>
</html>

```

B.3.15 /files/form/cambiarSend.php

```

<html>
<head>
<?php include '../php/index.php'; ?>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<center>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "UPDATE Nombres SET
Tiempo='".$_POST["tiempo"]."';
if($_POST["modulo"]!="todos") $sql .= " WHERE
ID='".$_POST["modulo"]."";
$sql .= ";";
if(mysqli_query($conn, $sql)){
    echo "Se ha cambiado el tiempo de envío de
".$_POST["modulo"]." a ".$_POST["tiempo"]." segundos.";
} else {
    echo "Error: " . mysqli_error($conn);
}
$conn->close();
?>
<br>
<a href="#" class="button">OK</a>
</center>
</body>
</html>

```

```

<html>
<head>
<?php include '../php/index.php'; ?>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<center>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "UPDATE config SET
Valor=".$_POST["RefreshTime"]." WHERE Name='refreshTime';";

if(mysqli_query($conn, $sql)){
    echo "Se han establecido
".$_POST["RefreshTime"]." segundos como delay.";
} else {
    echo "Error: " . mysqli_error($conn);
}

$conn->close();
?>
<br>
<a href='../editar.php' class="button">OK</a>
</center>
</body>
</html>

```

```

<html>
<head>
<?php include '../php/index.php'; ?>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<center>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "UPDATE Nombres SET
Nombre='".$_POST["nombre"]." WHERE
ID='".$_POST["modulo"]."';"

if(mysqli_query($conn, $sql)){
    echo "Se ha cambiado el nombre de
".$_POST["modulo"]." a ".$_POST["nombre"]."";
} else {
    echo "Error: " . mysqli_error($conn);
}

$conn->close();
?>
<br>
<a href="#" class="button">OK</a>
</center>
</body>
</html>

```

B.3.18 /files/form/addUser.php

```

<html>
<head>
    <?php include '../php/index.php'; ?>
    <style>
        .button {
            background-color: #4CAF50;
            border: none;
            color: white;
            padding: 15px 32px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 16px;
            margin: 4px 2px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <center>
        <?php
            $servername = "localhost";
            $username = "phpmyadmin";
            $password = "pass";
            $dbname = "Domotica";

            // Create connection
            $conn = new mysqli($servername, $username,
$password, $dbname);
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn-
>connect_error);
            }

            $sql = "SELECT Nombre FROM Usuarios WHERE
Nombre='".$_POST["newNombre"]."';";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                echo "El usuario ".$_POST["newNombre"]." ya
existe.";
            }
            else {
                $sql = "INSERT INTO Usuarios(Nombre,Pass)
VALUES
('".$_POST["newNombre"]."', '".$_md5($_POST["newPassword"])."')";

                if(mysqli_query($conn, $sql)){
                    echo "Se ha añadido el usuario
".$_POST["newNombre"].";
                } else {
                    echo "Error: " . mysqli_error($conn);
                }
            }
        </?php>
    </center>
</body>

```

```

        }

        $conn->close();
    ?>
    <br>
    <a href="#" ..../editar.php" class="button">OK</a>
    </center>
</body>
</html>

```

B.3.19 /files/form/addBtn.php

```

<html>
<head>
<?php include '../php/index.php'; ?>
<style>
.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<center>
<?php
$servername = "localhost";
$username = "phpmyadmin";
$password = "pass";
$dbname = "Domotica";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "SELECT Nombre FROM Botones WHERE
Nombre='".$._POST["nombre"]."';";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo "El botón ".$._POST["nombre"]." ya
existe.";
}

```

```

        }
    else {
        $sql = "INSERT INTO Botones(Nombre,Ejecutar)
VALUES ('".$_POST["nombre"]."', '".$_POST["ejecutar"]."');"

        if(mysqli_query($conn, $sql)){
            echo "Se ha añadido el botón
".$_POST["nombre"].", al ser pulsado ejecutará
".$_POST["ejecutar"]."";
        } else {
            echo "Error: " . mysqli_error($conn);
        }
    }

$conn->close();
?>
<br>
<a href="#" class="button">OK</a>
</center>
</body>
</html>

```

B.4. Base de datos

La base de datos permite almacenar y recuperar información de una forma muy cómoda.

El nombre de la base de datos es ‘Domotica’, y está formada por diversas tablas, y estas están formadas por diversas columnas. A continuación se dejará toda la información, además de cada tipo de columna (entre corchetes).

- » *Alarm*
 - *code [int(11)]*
 - *status [tinyint(1)]*
- » *Alarma*
 - *ind [int(11)]*
 - *maxVal [int(11)]*
 - *fuego [tinyint(1)]*
 - *gas [tinyint(1)]*
- » *Botones*
 - *Nombre [tinytext]*
 - *Ejecutar [tinytext]*
- » *config*
 - *Name [tinytext]*
 - *Valor [int(11)]*

- » *Enchufes*
 - *ind [int(11)]*
 - *Status [tinyint(1)]*
- » *LED*
 - *ind [int(11)]*
 - *Status [tinytext]*
 - *R [smallint(6)]*
 - *G [smallint(6)]*
 - *B [smallint(6)]*
 - *W [smallint(6)]*
- » *Nombres*
 - *ID [tinytext]*
 - *Nombre [tinytext]*
 - *Tiempo [tinyint(4)]*
- » *Riego*
 - *ID [tinytext]*
 - *checkRain [tinyint(1)]*
 - *rangoCheck [int(11)]*
 - *probCheck [int(11)]*
 - *checkHumidity [tinyint(1)]*
 - *humedad [int(11)]*
 - *forzarRiego [tinyint(1)]*
 - *vecesPorDia [int(11)]*
 - *duracionRiego [int(11)]*
- » *RiegoHora*
 - *ID [tinytext]*
 - *Hora [int(11)]*
- » *Tipos*
 - *ind [int(11)]*
 - *ID [tinytext]*
 - *Tipo [tinytext]*
 - *RoA [varchar(1)]*
- » *Usuarios*
 - *Nombre [tinytext]*
 - *Pass [text]*
- » *Valor*
 - *ind [int(11)]*
 - *Tiempo [varchar(1)]*
 - *Time [tinyint(4)]*
 - *Val [int(11)]*

Anexo C

Precios actuales

Aquí se encuentra el cálculo de precios mínimo y máximo.

The screenshot shows a website for alarm systems. At the top, there are navigation links: ALARMAS, DETECTORES, SIRENAS, ACCESORIOS, CÁMARAS, DOMÓTICA, NOS COMUNICADOR, OFERTAS-PACKS, and BLOG. On the right, it says "21 PRODUCTOS / 727,20€" and "IVA INCLUIDO EN TODOS LOS PRECIOS". Below this is a table of products:

PRODUCTO	PRECIO	CANTIDAD	TOTAL
Sensor / Detector magnético puertas-ventanas inalámbrico TOTAL ALARM TA-1	19,90 €	- 8 +	159,20 €
Alarma SIN cuotas Domótica TA-1 (WiFi / LAN / GSM) (SOLO panel)	169,00 €	- 1 +	169,00 €
Sensor / Detector de movimiento (PIR) inmune a mascotas	31,90 €	- 4 +	127,60 €
Teclado inalámbrico adicional TOTAL ALARM TA-1	39,90 €	- 1 +	39,90 €
mando a distancia inalámbrico inteligente/domótico compatible con alarmas TASC-TA1	34,90 €	- 2 +	69,80 €
Sensor / Detector de gas inalámbrico	37,90 €	- 1 +	37,90 €
Sensor / Detector de humo inalámbrico y con alarma interna	29,90 €	- 1 +	29,90 €
Sensor / Detector de inundación por agua inalámbrico	24,90 €	- 1 +	24,90 €
Servicio de Configuración Kit alarma SIN cuotas	39,00 €	- 1 +	39,00 €
Plan TRANQUILIDAD: Ayuda PERSONALIZADA en la instalación/configuración	30,00 €	- 1 +	30,00 €

At the bottom, there is a "DESGLOSE DEL PRECIO" table:

Total CON IVA	727,20€
Gastos de envío (Mercadona)	0,00€
Iva	116,21€
Incluye TOTAL (Iva y gastos de envío)	727,20€

A blue button at the bottom right says "REALIZAR PEDIDO".

Figura C.1: Precio mínimo

The screenshot shows the Z-Wave España website's shopping cart page. At the top, there's a header with the Z-Wave España logo, a language switch to Spanish (Español), a search icon, and a cart icon showing a total of € 2,099.32. Below the header, a navigation menu includes links for Sensors, Interruptors, Dimmers, Termostatos, Enchufes, Jardín, Persianas, Controles, Cámaras, Cursos, and Todos. A breadcrumb trail indicates the user is at 'Inicio > Carro de Compras'. The main content area is titled 'Carro de Compras' and contains a message: 'Productos marcados con *** no están disponibles en la cantidad deseada o se han agotado!'. Below this, a table lists the items in the cart:

Imagen	Nombre Producto	Modelo	Cantidad	Precio Unitario	Total
	eedomus Plus Connected Object	ES-71-COB-eedomusPlus	1	€ 297.66	€ 297.66
	Enchufe controlado (On / Off) TKB Home ***	ES-52-TKB-E_TZ68G	4	€ 45.00	€ 180.00
	Control LEDS RGBW Fibaro	ES-32-FIB_FGRGB-101	2	€ 59.99	€ 119.98
	Estación meteorológica personal para iPhone/ iPad y Android	ES-14-NET_NW501-WW	1	€ 169.00	€ 169.00
	Sensor de gas Z-Wave Plus- Zipato ***	ES-14-ZIP-HS1CG.Z	1	€ 49.00	€ 49.00
	Sensor de Monóxido de Carbono Z-Wave Plus - Fibaro	ES-14-FIB-FGCD-001	1	€ 99.99	€ 99.99
	Detector de movimiento con medida de T° - ZIPATO ***	ES-12-ZIP_ZP3102-EU	4	€ 46.46	€ 185.86
	Detector de Humo Z-Wave Plus – Zipato ***	ES-16-ZIP-HS1SA.Z	1	€ 39.00	€ 39.00
	Detector apertura ZD2102 Z-Wave Plus – Vision Security	ES-13-VIS_ZD2102	8	€ 44.99	€ 359.90

Anexo C. Precios actuales

	Detector de Inundación - Z-Wave Plus - Hank ***	ES-17-HAN-HKZW-FLD01	1	 	€ 34.99	€ 34.99
	Estación meteorológica Z-Wave Plus PoPP	ES-11-POP_005206	1	 	€ 199.95	€ 199.95
	Controlador de Riego Wi-Fi 16 zonas Gen 3 - GreenIQ	ES-69-GIQ-SYS-EU-16CI-FR	1	 	€ 299.00	€ 299.00
	Mando a distancia con soporte de Pared Z-Wave Plus Philio	ES-72-PHI_PSR04-1	1	 	€ 64.99	€ 64.99
<p>▶ USAR CODIGO DE CUPÓN</p> <p>▶ USAR VALE DE REGALO</p> <p>▶ INTRODUCE TU DESTINO PARA OBTENER EL CALCULO DE ENVIO</p>						
						Sub-Total
						€ 1,734.98
						21% IVA
						€ 364.35
						Total
						€2,099.32
CONTINUAR COMPRANDO				PAGAR		

Figura C.2: Precio máximo

Bibliografía

- Aliverti, P. (2017). Electrónica para makers. Marcombo.
- Conductor eléctrico. (2018, 19 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:49, septiembre 22, 2018 desde https://es.wikipedia.org/w/index.php?title=Conductor_el%C3%A9ctrico&oldid=110718807.
- D'Addario, M. (2018). Domótica. Createspace.
- Diagrama de flujo. (2018, 20 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:34, septiembre 22, 2018 desde https://es.wikipedia.org/w/index.php?title=Diagrama_de_flujo&oldid=110733247.
- Díaz Gómez, M. A. (2016, 7 de diciembre). NodeMCU dev kit v3 de LoLin. Recuperado de <http://arduinoamuite.blogspot.com/2016/12/nodemcu-dev-kit-v3-de-lolin.html>.
- Diodo. (2018, 3 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 18:51, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Diodo&oldid=110381348>.
- Domótica. (2018, 6 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:49, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Dom%C3%B3tica&oldid=110463131>.
- Gómez Gómez, M. (2006). Electrónica General. Ra-Ma.
- Hernández, L. (2018, 24 de enero). Cómo utilizar el sensor DHT11 para medir la temperatura y humedad con Arduino. Recuperado de <https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>.
- Llamas, L. (2016, 1 de julio). Detector de llama con Arduino y sensor infrarrojo. Recuperado de <https://www.luisllamas.es/detector-llama-arduino/>.
- Lógica binaria. (2018, 25 de julio). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 14:02, septiembre 22, 2018 desde https://es.wikipedia.org/w/index.php?title=L%C3%B3gica_binaria&oldid=109519629.

MQTT. (2018, 21 de septiembre). *Viquipèdia, l'Enclopèdia Lliure*. Fecha de consulta: 18:02, septiembre 21, 2018 desde <https://ca.wikipedia.org/w/index.php?title=MQTT&oldid=20344504>.

Multímetro. (2018, 11 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:47, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Mult%C3%A9metro&oldid=110564467>.

Platt, C. (2015). Enciclopedia de componentes electrónicos (volumen 1). Estripor.

Pluviómetro. (2018, 20 de abril). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 18:10, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Pluvi%C3%B3metro&oldid=107166760>.

Porcuna López, P. (2016). Robótica y domótica básica con Arduino. Ra-Ma.

Quiñónez, H. (2015, 16 de febrero). Protocolos de Domótica. Recuperado de <https://www.monografias.com/trabajos104/protocolos-domotica/protocolos-domotica.shtml>.

Raspberry Pi. (2018, 21 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 13:47, septiembre 22, 2018 desde https://es.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=110757509.

Relé. (2018, 18 de octubre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 10:18, octubre 21, 2018 desde <https://es.wikipedia.org/w/index.php?title=Rel%C3%A9&oldid=111356343>.

Rosas González, M. C. (2014). Los riesgos del Internet de las Cosas.

Semiconductor. (2018, 20 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 18:32, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Semiconductor&oldid=110732515>.

Sensor de movimiento para luz de escalera. (2018, 12 de junio). Recuperado de <https://microcontrollers.pro/sensor-de-movimiento-para-luz-de-escalera/>.

Transistor. (2018, 24 de septiembre). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 18:45, septiembre 29, 2018 desde <https://es.wikipedia.org/w/index.php?title=Transistor&oldid=110811449>.

Tutorial sensor de corriente AC no invasivo SCT-013. (2018, 21 de octubre). Recuperado de https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-s.html.

Zumbador. (2018, 26 de mayo). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 15:19, septiembre 22, 2018 desde <https://es.wikipedia.org/w/index.php?title=Zumbador&oldid=108152062>.