



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

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

ANEXOS

**Diseño y desarrollo de un sistema
domótico basado en IOT para vivienda
unifamiliar**

**Design and development of a home
automation system based on IOT for
single family housing**

424.17.55

Autor: Adrián Orte Ruiz

Director: Jesus Ponce De León Vázquez

Fecha: 06/02/2018

INDICE DE CONTENIDO

ANEXO 1. PLANOS	1
ANEXO 2. (PLANIFICACIÓN)	2
ANEXO 3. (PRESUPUESTO)	4
3.1. MATERIALES	4
3.2. CUADRO DE PRECIOS DESCOMPUESTOS	5
3.3. COSTES DE LA MANO DE OBRA	8
3.4. PRECIO TOTAL Y SUBTOTALES	9
ANEXO 4. (PLIEGO DE CONDICIONES)	12
4.1. DEFINICIÓN Y ALCANCE DEL PLIEGO	¡ERROR! MARCADOR NO DEFINIDO.
4.2. NORMAS, LEYES Y REGLAMENTOS	¡ERROR! MARCADOR NO DEFINIDO.
4.3. MATERIALES Y CONDICIONES TÉCNICAS	¡ERROR! MARCADOR NO DEFINIDO.
4.4. CONDICIONES ECONÓMICAS	¡ERROR! MARCADOR NO DEFINIDO.
4.5. PLAZOS DE ENTREGA	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 5. (DATOS DE LA MEMORIA)	12
5.1. MARCO TEÓRICO	17
5.1.1. PROTOCOLOS	17
5.1.1.1. Z-WAVE	17
5.1.1.2. ZIGBEE	17
5.1.1.3. LORAWAN	18
5.1.1.4. SIGFOX	18
5.1.1.5. WIFI	19
5.1.2. Sensores y detectores	19
5.1.2.1. Reed Switch	19
5.1.2.2. Sensor de luz LDR	21
5.1.2.3. Detectores de humo	21
5.1.2.3.1. De tipo de cámara de ionización	21
5.1.2.3.2. De tipo óptico	22
5.1.2.3.3. Detectores de humo por aspiración	23
5.1.2.4. Detectores de Monóxido de Carbono	23
5.1.2.5. Detectores de presencia	24
5.1.2.5.1. Detectores de presencia de infrarrojos	24

INDICES

5.1.2.5.2. Detectores por ultrasonidos	24
5.1.2.5.3. Detectores Diales	24
5.1.2.6. Termostatos Inteligentes	25
5.1.2.7. Sensores de gas	25
5.1.2.7.1. Sensores Semiconductores	25
5.1.2.7.2. Sensores Infrarrojos	26
5.1.2.7.3. Sensores Ultrasónicos	26
5.1.2.7.4. Sensores Electroquímicos	26
5.1.2.7.5. Sensores Catalíticos	26
5.1.2.8. Sensores de Inundación	27
5.1.2.9. Videovigilancia	27
5.1.2.9.1. Videovigilancia por CCTV.	27
5.1.2.9.2. Videovigilancia por internet	27
5.1.3. Microcontroladores	28
5.1.3.1. ARDUINO	28
5.1.3.2. ESP 8266	29
5.1.3.3. Raspberry Pi	29
5.2. ELECCIÓN DE COMPONENTES	30
5.2.1. Sistema de accionamiento de luces	30
5.2.2. Sistema de detección de humo	32
5.2.3. Sistema de detección de monóxido de carbono	32
5.2.4. Sistema de calefacción	33
5.2.5. Sistema de riego	35
5.2.6. Sistema de detección de fugas de agua	37
5.2.7. Sistema de control de persianas	38
5.2.8. Sistema de alarma	40
5.3. CONSUMOS ENERGÉTICOS	42
5.3.1. Sistema de accionamiento de luces	42
5.3.2. Sistema de detección de humo	42
5.3.3. Sistema de detección de CO	43
5.3.4. Sistema de calefacción	43
5.3.5. Sistema de riego	44
5.3.6. Sistema de detección de fugas de agua	44
5.3.7. Sistema de control de persianas	45
5.3.8. Sistema de alarma	45
5.3.9. Cálculo de sistema de alimentación	46
5.3.9.1. Pilas	47
5.3.9.2. Fuente de alimentación	48

INDICES

5.4. PROGRAMACIÓN	50
5.4.1. Configuración del microcontrolador	50
5.4.2. Base de datos	51
5.4.3. Sistema de accionamiento luces	53
5.4.4. Sistema de detección de humo	58
5.4.5. Sistema de detección de monóxido de carbono	61
5.4.6. Sistema de calefacción	62
5.4.7. Sistema de riego	67
5.4.8. Sistema fugas de agua	76
5.4.9. Sistema control de persianas	81
5.4.10. Sistema de alarma	86
5.4.10.1. Subsistema de detección de presencia	86
5.4.10.2. Subsistema de estado de puertas y ventanas	90
5.4.10.3. Videovigilancia	92
5.4.11. BIBLIOGRAFÍA	93

INDICE DE ILUSTRACIONES

Ilustración 1-Reed Switch comercial («¿Que es un Reed Switch?», 2010)	19
Ilustración 2-Reed Switch normalmente abierto (NO)(«¿Que es un Reed Switch?», 2010).....	20
Ilustración 3-Reed Switch doble («¿Que es un Reed Switch?», 2010)	20
Ilustración 4- LDR («LDR-fotoresistencia», 2010)	21
Ilustración 5- Cámara de ionización («Tipología de los detectores de incendios de incendios y gases», 2012)	22
Ilustración 6- Detector óptico («Tipología de los detectores de incendios de incendios y gases», 2012)	22
Ilustración 7- Detectores de humo por aspiración («Tipología de los detectores de incendios de incendios y gases», 2012)	23
Ilustración 8-Tipos de Arduino («¿Qué es Arduino? ¿Qué modelo comprar?», 2013)	28

INDICES

Ilustración 9-Raspberry PI	29
Ilustración 10-Nodemcu	30
Ilustración 11-PIR.....	30
Ilustración 12-LDR.....	31
Ilustración 13-Relé(«SLA-05VDC-SL-C_Datasheet.pdf» , 2015).....	31
Ilustración 14 -MQ-2	32
Ilustración 15- MQ-7	32
Ilustración 16-LM35(Dilmen, 2014).....	33
Ilustración 17- DHT11(«DHT11 Sensor – Tienda y Tutoriales Arduino», 2016)..	33
Ilustración 18- DS18B20 («DS18B20 Sensor de Temperatura», 2017)	34
Ilustración 19 - DHT11(«DHT11 Sensor – Tienda y Tutoriales Arduino», 2016).35	35
Ilustración 20- DHT22(«Medir temperatura y humedad con Arduino y sensor DHT11-DHT22», 2016).....	35
Ilustración 21- SHT75(«SHT75 Datasheet PDF - Sensirion - Pinout,Circuit - FindIC.us», 2007.).....	36
Ilustración 22- FC-28(eodos, 2015)	36
Ilustración 23- Sensor de agua.....	37
Ilustración 24-Tabla de motor según persiana(«Elige motor adecuado al peso y eje de la persiana», 2007).....	39
Ilustración 25- Motor paso a paso NEMA 17	39
Ilustración 26- PIR.....	40
Ilustración 27- Reed Switch	41
Ilustración 28-D-link930LB	41
Ilustración 29-Pilo Litio de la marca VARTA para aplicaciones electrónicas.(«Pila litio industria CR AA 3V 2Ah», 2016).....	47
Ilustración 30-Fuente de alimentación 50W MeanWell(«LED Fuente de alimentación 50W 5V 10A ; MeanWell, RS-50-5: Amazon.es: Bricolaje y herramientas», 2015)	48
Ilustración 31- configuración Wifi Nodemcu.....	50

Ilustración 32-orden de envío y lectura de datos.....	50
Ilustración 33- Orden para identificar a cliente conectado	51
Ilustración 34-Estructura base de datos.....	52
Ilustración 35- Formulario para envío de datos a la base de datos	52
Ilustración 36-Configuración LDR.....	53
Ilustración 37- Base de datos LDR	54
Ilustración 38-Forma de mandar a reposar al microcontrolador	54
Ilustración 39-Forma de guardar valor del sensor en archivo JSON	54
Ilustración 40-Código de modos de uso	55
Ilustración 41- Barra de direcciones al pulsar botón apagar.....	55
Ilustración 42-Variable con el valor de la barra de direcciones	55
Ilustración 43- Forma de guardar datos en archivo JSON	56
Ilustración 44- Variable que contiene los datos del archivo JSON	56
Ilustración 45- Formato archivo JSON	56
Ilustración 46- Condiciones programadas en el microcontrolador	56
Ilustración 47- Condiciones programadas para el modo automático.....	57
Ilustración 48- Configuración sensor PIR («Detector de movimiento con Arduino y sensor PIR», 2015)	57
Ilustración 49- Función para devolver datos de humo.....	58
Ilustración 50- Base de datos sistema de humo	58
Ilustración 51- Interfaz donde se muestran los datos	59
Ilustración 52- Obtención de los datos almacenados en la base de datos	59
Ilustración 53- Condiciones que se cumplen dependiendo del valor del sensor..	60
Ilustración 54- Archivo para envío de correo electrónico	60
Ilustración 55- Formato de correo electrónico.....	60
Ilustración 56-Base de datos monóxido de carbono.....	61
Ilustración 57-Función del sensor CO	61
Ilustración 58- Librerías usadas para configuración de sensor.....	62

INDICES

Ilustración 59- Función para obtener datos de temperatura	62
Ilustración 60- Base de datos sistema de calefacción	63
Ilustración 61- Interfaz con dato de temperatura en tiempo real.....	63
Ilustración 62-Obtención de valores de la base de datos.....	63
Ilustración 63- Órdenes para mostrar temperatura en la web	64
Ilustración 64- Forma de guardar dato temperatura en archivo JSON	64
Ilustración 65- Caja para enviar temperatura	64
Ilustración 66- Código de programación caja.....	64
Ilustración 67- Guardado del dato de temperatura en archivo JSON	65
Ilustración 68-Transformación del string en float	65
Ilustración 69- Condiciones para encender la caldera con histéresis programada	66
Ilustración 70- Web general de aplicación	67
Ilustración 71-Base de datos riego.....	67
Ilustración 72- Obtención de datos de la base de datos	67
Ilustración 73- Forma de mostrar los datos por pantalla	68
Ilustración 74- Interfaz modo manual	68
Ilustración 75-Páginas asociadas a o botones de interfaz	69
Ilustración 76-Opciones de zonas	69
Ilustración 77- Archivo JSON.....	70
Ilustración 78- Barra de direcciones web	70
Ilustración 79- Condiciones programas en PHP	70
Ilustración 80- Variable que guarda los datos del archivo JSON	71
Ilustración 81- Condicional if con substring correspondiente a zona 1.....	71
Ilustración 82- Interfaz modo automático	72
Ilustración 83- PHP con valores de desplegable y forma de envío por método "POST".....	72
Ilustración 84-Variable que contiene el valor del desplegable	73

Ilustración 85-Forma de guardar valor	73
Ilustración 86- Librería de reloj	73
Ilustración 87- Función para conectar con el servidor	74
Ilustración 88-Función para dar la zona horaria del país	74
Ilustración 89- Variables para guardar datos de reloj	74
Ilustración 90-Variabes con datos del archivo JSON.....	74
Ilustración 91- Condiciones para accionar y apagar el riego del modo automático	75
Ilustración 92-Base de datos fuga de agua.....	76
Ilustración 94- Petición para extraer datos de la base de datos y mostrarlos en pantalla.....	76
Ilustración 95- Condiciones programadas cuando se detecta agua	77
Ilustración 96-Condiciones programadas cuando no se detecta agua	78
Ilustración 97-Botón de Rearme	78
Ilustración 98-Dirección que aparece cuando se pulsa el botón rearme.....	78
Ilustración 99- Referencia asociada a botón rearme	78
Ilustración 100-Variable que toma el valor de la barra de direcciones	79
Ilustración 101- Condición que se cumple cuando se pulsa el botón rearme	79
Ilustración 102- Envío de mensaje de correo cuando se detecta agua.....	79
Ilustración 103-String con contenido del archivo JSON	80
Ilustración 104- Substring con dato importante	80
Ilustración 105- Condiciones programadas para apagar o encender el relé.....	80
Ilustración 106- Pines del motor	81
Ilustración 107-Array para usar las fases de forma alternada	81
Ilustración 108-Función para determinar que bobina ha de activarse.....	82
Ilustración 109-Funciones para mover el motor en ambos sentidos	82
Ilustración 110- Variable para contar pasos	83
Ilustración 111- Barra de direcciones Web con valor del botón pulsado.....	83

INDICES

Ilustración 112- Variable para tomar valor seleccionado	84
Ilustración 113-Condiciones programadas para cada valor	84
Ilustración 114- Valor guardado en archivo JSON	84
Ilustración 115-Variable del microcontrolador con valor archivo JSON	84
Ilustración 116- Ejemplo de una de las cinco posiciones	85
Ilustración 117-Declaración del PIR como entrada en el microcontrolador	86
Ilustración 118- Base de datos Seguridad	86
Ilustración 119- Petición de los datos de la base de datos.....	87
Ilustración 120- Array de datos	87
Ilustración 121-Modos seleccionables.....	87
Ilustración 122- Barra de direcciones al pulsar el botón “modo en casa”.....	87
Ilustración 123-variable con valor del botón pulsado	87
Ilustración 124-Método para guardar datos en un archivo JSON	88
Ilustración 125- Archivo JSON para registrar una incidencia.....	88
Ilustración 126-Condiciones cuando no se detecta intruso y el modo fuera de casa esta activado.	89
Ilustración 127-Condiciones cuando se detecta intruso y el modo fuera de casa esta activado.	89
Ilustración 128- Envío de correo electrónico.....	89
Ilustración 129-Correo electrónico.	89
Ilustración 130- Montaje de aplicación	90
Ilustración 131- Función del sensor Reed Switch	90
Ilustración 132- Base de datos para la aplicación de estado ventanas	91
Ilustración 133- Condiciones para mostrar imagen puerta abierta o puerta cerrada	91
Ilustración 134-Envío de mensaje de alarma	92
Ilustración 135-Cámara IP	92

INDICE DE TABLAS

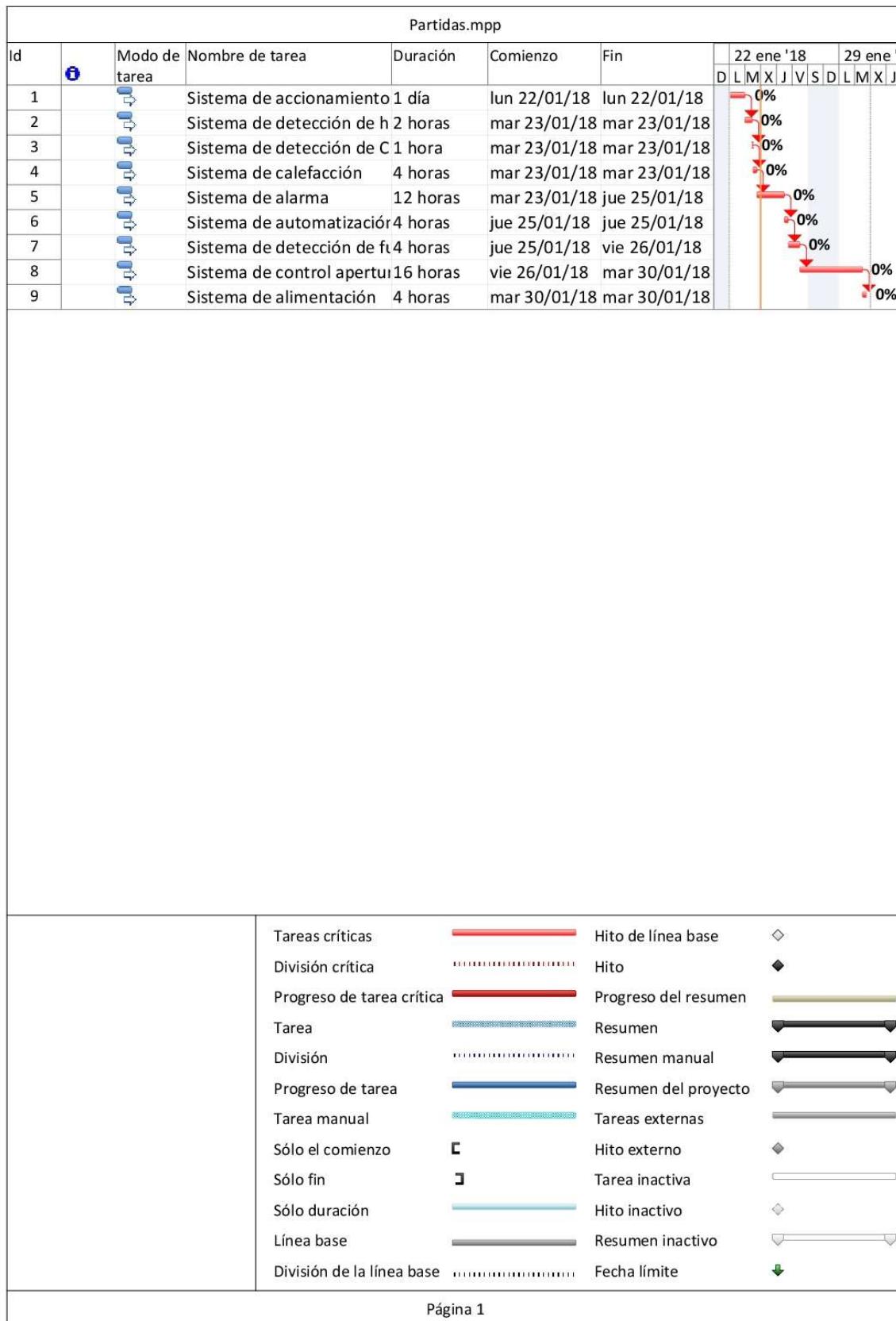
Tabla 1-Coeficiente K según material	38
Tabla 2-Consumo sistema de accionamiento luces	42
Tabla 3- Consumo sistema de detección de humo	42
Tabla 4-Precio sistema de detección de monóxido de carbono	43
Tabla 5-Consumo sistema de calefacción	43
Tabla 6-Consumo sistema de riego	44
Tabla 7-Consumo sistema de fugas de agua.....	44
Tabla 8-Consumos sistema control persianas	45
Tabla 9-Consumo subsistema de presencia	45
Tabla 10-Consumo subsistema estado de puertas y ventanas.....	45
Tabla 11-Resumen general de consumos eléctricos	46
Tabla 12-Precio fuente alimentación sistema	49
Tabla 13-Fuente de alimentación motores (<SODIAL(R)24W Conductor Fuente de alimentacion Transformador DC 12V 2A por Banda Luz LED Lampara: Amazon.es: Bricolaje y herramientas>, 2016).....	49



ANEXO 1. PLANOS

PLANIFICACIÓN

ANEXO 2. PLANIFICACIÓN





PRESUPUESTO

ANEXO 3. PRESUPUESTO

3.1. MATERIALES

CÓDIGO	CANTIDAD UD	RESUMEN	PRECIO	IMPORTE
010101	33,000 ud	rele	1,97	65,01
010102	41,000 ud	PIR	2,00	82,00
010102MO	17,000 ud	Camaras	50,00	850,00
010203	19,000 ud	Nodemcu v3	5,20	98,80
010204	2,000 ud	Sensor MQ_2	3,78	7,56
010204H2O	7,000 ud	Sensor humedad FC-28	1,99	13,93
010204HUMEDA	4,000 ud	Sensor humedad FC-28	1,80	7,20
010204M	1,000 ud	Sensor MQ_7	3,58	3,58
010204MCAL	4,000 ud	Sensor DS18B20	2,00	8,00
010205	1,000 ud	LDR	1,00	1,00
010504SWITCH	17,000 ud	Red switch	2,00	34,00
01080301MOT	16,000 ud	MOTOR NEMA 17	48,00	768,00
			Grupo 010	1.939,08
0110ASDF	1,000 ud	FUENTE ALIMENTACION 50W	14,47	14,47
			Grupo 011	14,47
		TOTAL.....		1.953,55

3.2. CUADRO DE PRECIOS DESCOMPUESTOS

CÓDIGO	CANTIDAD UD	RESUMEN	PRECIO	SUBTOTAL	IMPORTE
CAPÍTULO 01 SISTEMA DOMÓTICO IOT					
SUBCAPÍTULO 01.01 SISTEMA DE ACCIONAMIENTO DE LUCES					
01.01.01	ud	SISTEMA DE ACCIONAMIENTO LUCES Suministro y colocación de sistema de luces para 72 puntos de luz de una vivienda unifamiliar (considerando 3 luces por Relé), IOT luces, compuesto de 5 microcontroladores tipo Nodemcu v3, 24 PIR (sensor de presencia), 1 LDR y 24 relés tipo SLA-05vdc-SL-c, todo perfectamente ensamblado y programado para su funcionamiento (No incluye alimentación)			
O01OB200	16,000 h.	Oficial 1ª electricista	25,00	400,00	
010101	24,000 ud	relé	1,97	47,28	
010102	24,000 ud	PIR	2,00	48,00	
010203	5,000 ud	Nodemcu v3	5,20	26,00	
010205	1,000 ud	LDR	1,00	1,00	
%%GG	13,000	GASTOS GENERALES	522,30	67,90	
%%BI	6,000	BENEFICIO INDUSTRIAL	590,20	35,41	
TOTAL PARTIDA.....					625,59
Asciende el precio total de la partida a la mencionada cantidad de SEISCIENTOS VEINTICINCO EUROS con CINCUENTA Y NUEVE CÉNTIMOS					
SUBCAPÍTULO 01.02 SISTEMA DE DETECCIÓN DE HUMO					
01.02.01	ud	SISTEMA DE DETECCIÓN DE HUMO Suministro y colocación de sistema de detección humo para una vivienda unifamiliar de dos cocinas, IOT detección de humo, compuesto de 1 microcontrolador tipo Nodemcu v3, y dos sensores MQ-2 de humo (uno por cocina), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluido en este precio la alimentación			
O01OB200	2,000 h.	Oficial 1ª electricista	25,00	50,00	
010203	1,000 ud	Nodemcu v3	5,20	5,20	
010204	2,000 ud	Sensor MQ_2	3,78	7,56	
%%GG	13,000	GASTOS GENERALES	62,80	8,16	
%%BI	6,000	BENEFICIO INDUSTRIAL	70,90	4,25	
TOTAL PARTIDA.....					75,17
Asciende el precio total de la partida a la mencionada cantidad de SETENTA Y CINCO EUROS con DIECISIETE CÉNTIMOS					
SUBCAPÍTULO 01.03 SISTEMA DE DETECCIÓN DE CO					
01.03.01	ud	SISTEMA DE DETECCIÓN DE CO Suministro y colocación de sistema de detección de CO para una sala de calderas de vivienda unifamiliar, IOT detección de CO, compuesto de 1 microcontrolador tipo Nodemcu v3, y 1 sensor MQ-7 de monóxido de carbono (uno en la sala de calderas), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentación.			
O01OB200	1,000 h.	Oficial 1ª electricista	25,00	25,00	
010203	1,000 ud	Nodemcu v3	5,20	5,20	
010204M	1,000 ud	Sensor MQ_7	3,58	3,58	
%%GG	13,000	GASTOS GENERALES	33,80	4,39	
%%BI	6,000	BENEFICIO INDUSTRIAL	38,20	2,29	
TOTAL PARTIDA.....					40,46
Asciende el precio total de la partida a la mencionada cantidad de CUARENTA EUROS con CUARENTA Y SEIS CÉNTIMOS					

PRESUPUESTO

CÓDIGO	CANTIDAD UD	RESUMEN	PRECIO	SUBTOTAL	IMPORTE
--------	-------------	---------	--------	----------	---------

SUBCAPÍTULO 01.04 SISTEMA DE CALEFACCION

01.04.01	ud	SISTEMA DE CALEFACCION			
		Suministro y colocación de sistema regulación de temperatura de vivienda mediante el control de la calefacción para una vivienda unifamiliar (considerando 4 zona de regulación), IOT calefacción, compuesto de 2 microcontroladores tipo Nodemcu v3, y 4 relés tipo SLA-05vdc-SL-c, y 4 sensores de temperatura tipo DS18B20 todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye alimentación			
O01OB200	4,000 h.	Oficial 1º electricista	25,00	100,00	
010203	2,000 ud	Nodemcu v3	5,20	10,40	
010204MCAL	4,000 ud	Sensor DS18B20	2,00	8,00	
010101	4,000 ud	relé	1,97	7,88	
%%GG	13,000	GASTOS GENERALES	126,30	16,42	
%%BI	6,000	BENEFICIO INDUSTRIAL	142,70	8,56	

TOTAL PARTIDA..... 151,26

Asciende el precio total de la partida a la mencionada cantidad de CIENTO CINCUENTA Y UN EUROS con VEINTISEIS CÉNTIMOS

SUBCAPÍTULO 01.05 SISTEMA DE ALARMA

01.05.01	ud	SUBSISTEMA ESTADO APERTURA PUERTAS Y VENTANAS			
		Suministro y colocación de sistema para control del estado de apertura de puertas y ventanas para una vivienda unifamiliar (considerando 16 ventanas y 1 puerta principal), IOT Apertura puertas y ventanas, compuesto de 2 microcontroladores tipo Nodemcu v3, y 17 reed switch perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentación.			
O01OB200	4,000 h.	Oficial 1º electricista	25,00	100,00	
010203	2,000 ud	Nodemcu v3	5,20	10,40	
010504SWITCH	17,000 ud	Red switch	2,00	34,00	
%%GG	13,000	GASTOS GENERALES	144,40	18,77	
%%BI	6,000	BENEFICIO INDUSTRIAL	163,20	9,79	

TOTAL PARTIDA..... 172,96

Asciende el precio total de la partida a la mencionada cantidad de CIENTO SETENTA Y DOS EUROS con NOVENTA Y SEIS CÉNTIMOS

01.05.02 SUBSISTEMA DETECCION DE PRESENCIA

		Suministro y colocación de subsistema detección de presencia en vivienda unifamiliar (considerando colocación en todas las plantas), IOT DERECCION DE PRESENCIA, compuesto de 2 microcontrolador tipo Nodemcu v3, 17 PIR (se consideran 16 ventanas y 1 puerta) todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT,no incluye sistema de alimentación.			
O01OB200	4,000 h.	Oficial 1º electricista	25,00	100,00	
010203	2,000 ud	Nodemcu v3	5,20	10,40	
010102	17,000 ud	PIR	2,00	34,00	
%%GG	13,000	GASTOS GENERALES	144,40	18,77	
%%BI	6,000	BENEFICIO INDUSTRIAL	163,20	9,79	

TOTAL PARTIDA..... 172,96

Asciende el precio total de la partida a la mencionada cantidad de CIENTO SETENTA Y DOS EUROS con NOVENTA Y SEIS CÉNTIMOS

01.05.03 SUBSISTEMA VIDEO VIGILANCIA

		Suministro y colocación de subsistema de video vigilancia en vivienda unifamiliar (considerando colocación en todas las plantas), IOT video vigilancia, compuesto de 17 video-cámaras todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentación.			
O01OB200	4,000 h.	Oficial 1º electricista	25,00	100,00	
010102MO	17,000 ud	Cámaras	50,00	850,00	
%%GG	13,000	GASTOS GENERALES	950,00	123,50	
%%BI	6,000	BENEFICIO INDUSTRIAL	1.073,50	64,41	

TOTAL PARTIDA..... 1.137,91

Asciende el precio total de la partida a la mencionada cantidad de MIL CIENTO TREINTA Y SIETE EUROS con NOVENTA Y UN CÉNTIMOS

CÓDIGO	CANTIDAD UD	RESUMEN	PRECIO	SUBTOTAL	IMPORTE
SUBCAPÍTULO 01.06 SISTEMA DE RIEGO JARDIN					
01.06.01	ud	SISTEMA DE RIEGO JARDIN			
		Suministro y colocación de sistema para control del riego para jardín (considerando 4 zonas), IOT riego, compuesto de 1 microcontrolador tipo Nodemcu v3, 4 relés tipo SLA_05vdc_SL_c y 4 sensores de humedad tipo FC-28 (no incluye válvulas de accionamiento) perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentación.			
O01OB200	4,000 h.	Oficial 1ª electricista	25,00	100,00	
010203	2,000 ud	Nodemcu v3	5,20	10,40	
010101	4,000 ud	relé	1,97	7,88	
010204HUMEDA	4,000 ud	Sensor humedad FC-28	1,80	7,20	
%%GG	13,000	GASTOS GENERALES	125,50	16,32	
%%BI	6,000	BENEFICIO INDUSTRIAL	141,80	8,51	
TOTAL PARTIDA.....					150,31

Asciende el precio total de la partida a la mencionada cantidad de CIENTO CINCUENTA EUROS con TREINTA Y UN CÉNTIMOS

SUBCAPÍTULO 01.07 SISTEMA DETECCIÓN DE FUGAS DE AGUA

01.07.01	ud	SISTEMAS DETECCIÓN FUGAS DE AGUA			
		Suministro y colocación de sistema de detección de fugas de agua para una vivienda unifamiliar en 5 baños y 2 cocinas, IOT detección de H2O, compuesto de 1 microcontrolador tipo Nodemcu v3, 1 relé tipo SLA-05 vdc-sl-c y 7 sensores tipo S0023 de la empresa solectro shop de detección de H2O de (uno en cada cuarto húmedo de la vivienda), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentación. Incluye la conexión del relé a la acometida. (No incluye válvulas de accionamiento).			
O01OB200	4,000 h.	Oficial 1ª electricista	25,00	100,00	
010203	1,000 ud	Nodemcu v3	5,20	5,20	
010101	1,000 ud	relé	1,97	1,97	
010204H2O	7,000 ud	Sensor humedad FC-28	1,99	13,93	
%%GG	13,000	GASTOS GENERALES	121,10	15,74	
%%BI	6,000	BENEFICIO INDUSTRIAL	136,80	8,21	
TOTAL PARTIDA.....					145,05

Asciende el precio total de la partida a la mencionada cantidad de CIENTO CUARENTA Y CINCO EUROS con CINCO CÉNTIMOS

SUBCAPÍTULO 01.08 SISTEMA DE CONTROL DE APERTURA DE PERSIANAS

01.08.01	ud	SISTEMA CONTROL APERTURA PERSIANAS			
		Suministro y colocación de sistema control de apertura de persianas en vivienda unifamiliar (considerando 16 ventanas con persiana), IOT CONTROL APERTURA PERSIANAS, compuesto de 1 microcontrolador tipo Nodemcu v3, 16 motores paso a paso NEMA 17 y 16 drivers de motor DRV8825(uno para cada motor) todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentación.			
O01OB200	16,000 h.	Oficial 1ª electricista	25,00	400,00	
010203	3,000 ud	Nodemcu v3	5,20	15,60	
01080301MOT	16,000 ud	MOTOR NEMA 17	48,00	768,00	
01080401DRIVE	16,000 ud	DRIVERS DRV8825	4,60	73,60	
%%GG	13,000	GASTOS GENERALES	1.257,20	163,44	
%%BI	6,000	BENEFICIO INDUSTRIAL	1.420,60	85,24	
TOTAL PARTIDA.....					1.505,88

Asciende el precio total de la partida a la mencionada cantidad de MIL QUINIENTOS CINCO EUROS con OCHEENTA Y OCHO CÉNTIMOS

PRESUPUESTO

CÓDIGO	CANTIDAD UD	RESUMEN	PRECIO	SUBTOTAL	IMPORTE
SUBCAPÍTULO 01.09 ALIMENTACION					
01.09.01	ud	ALIMENTACION Suministro y colocación de fuentes de alimentación para sistemas domóticos considerando los incluidos en la vivienda unifamiliar del proyecto, incluye alimentación de sensores, microcontroladores y motores todo ello probado y en funcionamiento.			
001OB200	4,000 h.	Oficial 1ª electricista	25,00	100,00	
0110ASDF	1,000 ud	FUENTE ALIMENTACION 50W	14,47	14,47	
0111ASDJJD	16,000 ud	FUENTE DE ALIMENTACION 24W	14,47	231,52	
%%GG	13,000	GASTOS GENERALES	346,00	44,98	
%%BI	6,000	BENEFICIO INDUSTRIAL	391,00	23,46	
TOTAL PARTIDA.....					414,43

Asciende el precio total de la partida a la mencionada cantidad de CUATROCIENTOS CATORCE EUROS con CUARENTA Y TRES CÉNTIMOS

3.3. COSTES DE LA MANO DE OBRA

LISTADO DE MANO DE OBRA VALORADO

CÓDIGO	CANTIDADUD	RESUMEN	PRECIO	IMPORTE
001OB200	63,000 h.	Oficial 1ª electricista	25,00	1.575,00
		Grupo 001		1.575,00
		TOTAL		1.575,00

3.4. PRECIO TOTAL Y SUBTOTALES

CAPÍTULO 01 SISTEMA DOMÓTICO

SUBCAPÍTULO 01.01 SISTEMA DE REGULACION DE LUCES

01.01.01	ud	SISTEMA DE REGULACION LUces			
		Suministro y colocacion de sistema de luces para 72 puntos de luz de una vivienda unifamiliar(considerando 3 luces por Rele), IOT luces, compuesto de 5 microcontroladores tipo Nodemcu v3, 24 PIR (sensor de presencia),1 LDR y 24 relés tipo SLA-05vdc-SL-c, todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion.	1	1,00	
				1,00	625,59

TOTAL SUBCAPÍTULO 01.01 SISTEMA DE REGULACION DE 625,59

SUBCAPÍTULO 01.02 SISTEMA DE DETECCION DE HUMO

01.02.01	ud	SISTEMA DE DETECCION DE HUMO			
		Suministro y colocacion de sistema de deteccion humo para una vivienda unifamiliar de dos cocinas, IOT detección de humo, compuesto de 1 microcontrolador tipo Nodemcu v3, y dos sensores MQ-2 de humo(uno por cocina), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion.	1	1,00	
				1,00	75,17

TOTAL SUBCAPÍTULO 01.02 SISTEMA DE DETECCION DE 75,17

SUBCAPÍTULO 01.03 SISTEMA DE DETECCION DE CO

01.03.01	ud	SISTEMA DE DETECCION DE CO			
		Suministro y colocacion de sistema de detección de CO para una sala de calderas de vivienda unifamiliar, IOT detección de CO, compuesto de 1 microcontrolador tipo Nodemcu v3, y 1 sensor MQ-7 de monóxido de carbono(uno en la sala de calderas), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion.	1	1,00	
				1,00	40,46

TOTAL SUBCAPÍTULO 01.03 SISTEMA DE DETECCION DE CO 40,46

SUBCAPÍTULO 01.04 SISTEMA DE CALEFACCION

01.04.01	ud	SISTEMA DE CALEFACCION			
		Suministro y colocacion de sistema regulación de temperatura de vivienda mediante el control de la calefacción para una vivienda unifamiliar(considerando 4 zona de regulación), IOT calefacción, compuesto de 2 microcontroladores tipo Nodemcu v3, y 4 relés tipo SLA-05vdc-SL-c, y 4 sensores de temperatura tipo DS18B20 todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion.	1	1,00	
				1,00	151,26

TOTAL SUBCAPÍTULO 01.04 SISTEMA DE CALEFACCION. 151,26

PRESUPUESTO

SUBCAPÍTULO 01.05 SISTEMA DE ALARMA

01.05.01	ud	SUBSISTEMA ESTADO APERTURA PUERTAS Y VENTANAS		1	1,00			
		Suministro y colocacion de sistema para control del estado de apertura de puertas y ventanas para una vivienda unifamiliar(considerando 16 ventanas y 1 puerta principal), IOT Apertura puertas y ventanas, compuesto de 2 microcontroladores tipo Nodemcu v3, y 17 reed switch perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentacion.						
01.05.02	ud	SUBSISTEMA DETECCION DE PRESENCIA				1,00	172,96	172,96
		Suministro y colocacion de subsistema deteccion de presencia en vivienda unifamiliar (considerando colocacion en todas las plantas), IOT DERECCION DE PRESENCIA, compuesto de 2 micro-controlador tipo Nodemcu v3, 17 PIR (se consideran 16 ventanas y 1 puerta) todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT,no incluye sistema de alimentacion.						
01.05.03	ud	SUBSISTEMA VIDEO VIGILANCIA		1	1,00			
		Suministro y colocacion de subsistema de video vigilacia en vivienda unifamiliar (considerando colocacion en todas las plantas), IOT video vigilancia, compuesto de 17 video-camaras todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, no incluye sistema de alimentacion.				1,00	172,96	172,96

TOTAL SUBCAPÍTULO 01.05 SISTEMA DE ALARMA **1.483,83**

SUBCAPÍTULO 01.06 SISTEMA DE AUTOMATIZACION DE RIEGO JARDIN

01.06.01	ud	SISTEMA DE AUTOMATIZACION RIEGO JARDIN		1	1,00			
		Suministro y colocacion de sistema para control del riego para jardin (considerando 4 zonas), IOT riego, compuesto de 1 microcontroladores tipo Nodemcu v3, 4 relés tipo SLA_05vdc_SL_c y 4 sensores de humedad tipo FC-28 (no incluye valvulas de accionamiento) perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion.				1,00	150,31	150,31

TOTAL SUBCAPÍTULO 01.06 SISTEMA DE AUTOMATIZACION..... **150,31**

DETECCION DE FUGAS DE AGUA

01.07.01	ud	SISTEMAS DETECCION FUGAS DE AGUA			
		Suministro y colocacion de sistema de deteccion de fugas de agua para una vivienda unifamiliar en 5 baños y 2 cocinas, IOT detección de H2O, compuesto de 1 microcontrolador tipo Nodemcu v3,1 re-le tipo SLA-05 vdc-sl-c y 7 sensores tipo S0023 de la empresa solectro shop de detección de H2O de (uno en cada cuarto humedo de la vivienda), todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentacion. Incluye la conexión del relé a la acometida. (No incluye válvulas de accionamiento).	1	1,00	
				1,00	145,05

TOTAL SUBCAPÍTULO 01.07 SISTEMA DETECCION DE FUGAS..... **145,05**

SUBCAPÍTULO 01.08 SISTEMA DE CONTROL DE APERTURA DE PERSINAS

01.08.01	ud	SISTEMA CONTROL APERTURA PERSIANAS			
		Suministro y colocación de sistema control de apertura de persianas en vivienda unifamiliar (considerando 16 ventanas con persiana), IOT CONTROL APERTURA PERSIANAS, compuesto de 1 microcontrolador tipo Nodemcu v3, 16 motores paso a paso NEMA 17 y 16 drivers de motor DRV8825(uno para cada motor) todo perfectamente ensamblado y programado para su funcionamiento en sistema IOT, (no incluido en este precio) no incluye sistema de alimentación.	1	1,00	
				1,00	1.505,88

TOTAL SUBCAPÍTULO 01.08 SISTEMA DE CONTROL DE ... **1.505,88**

SUBCAPÍTULO 01.09 ALIMENTACION

01.09.01	ud	ALIMENTACION			
		Suministro y colocación de fuentes de alimentación para sistemas domóticos considerando los incluidos en la vivienda unifamiliar del proyecto, incluye alimentación de sensores y microcontroladores (no incluye la de motores) todo ello probado y en funcionamiento.	1	1,00	
				1,00	414,43

TOTAL SUBCAPÍTULO 01.09 ALIMENTACION **414,43**

TOTAL CAPÍTULO 01 SISTEMA DOMOTICO..... **4.591,98**

TOTAL..... **4.591,98**

PLIEGO DE CONDICIONES

ANEXO 4. PLIEGO DE CONDICIONES

4.1. DATOS BÁSICOS DE LA VIVIENDA

4.1.1. Superficie ocupada y distribución de estancias

Se desea llevar a cabo el diseño de un sistema domótico en una vivienda unifamiliar ya existente.

La vivienda se distribuye en cuatro plantas con el siguiente programa:

Planta Sótano:

Salón cocina, baño, sala diáfana y cuarto de instalaciones de una superficie aproximada de 100m².

Planta Baja:

Salón, cocina, baño, dormitorio, habitación, cuarto de lavandería, y vestíbulo de una superficie aproximada de 100m².

Planta Primera:

Cuatro dormitorios, dos baños, y distribuidor de superficie aproximada de unos 100 m²

Planta Buhardilla:

Habitación diáfana con baño de superficie aproximada de 65m².

Esta vivienda tiene en su evolvente 16 ventanas y una puerta, dato utilizado en este proyecto para el sistema de alarma.

La vivienda comprende una parcela de 500 m² y consta de cuatro plantas, con una superficie total de vivienda de 365m².

4.1.2. *Equipamiento domestico*

La implantación de un sistema domótico no tendrá sentido si no existe un grado mínimo de equipamiento en la vivienda o edificio.

El equipamiento susceptible de ser domotizado es el siguiente:

- Suministro agua, de forma que se puede actuar sobre las válvulas de acometida en caso de detección de fugas.
- Sistema de calefacción. Existe una sala con la caldera que podrá ser controlada desde internet.
- Persianas, que se podrán automatizar y controlar desde internet
- Bombillas que podrán ser accionadas mediante a relés y el uso de un dispositivo conectado a internet.
- Ventanas y puertas a las que se les colocara un sensor que indique su estado.

4.2. REQUERIMIENTO DE LOS USUARIOS

4.2.1. *Tipologías de usuarios*

El sistema domótico está orientado a mejorar las prestaciones de la vivienda para un público en general. Por lo tanto, el sistema se instala con el objetivo de mejorar el nivel de vida de cualquier persona que instale el sistema mejorando las prestaciones de su vivienda.

Los beneficios que aportará el sistema domótico implantado en la vivienda a los usuarios de la misma serán:

- Simplificación de tareas.
- Seguridad, tanto de bienes como personal.
- Confort

PLIEGO DE CONDICIONES

4.3. FUNCIONALIDAD REQUERIDA AL SISTEMA

El presente proyecto, tiene como objetivo el diseño de un sistema domótico en una vivienda unifamiliar ya existente.

Al tratarse de una vivienda ya construida, será necesaria la elección de una tecnología que no suponga reformas para tendido de nuevo cableado, sino que trate de provechar la infraestructura ya existente, como la red wifi que existe en la vivienda.

4.3.1. Especificaciones por funciones domóticas

Para que el diseño del sistema domótico resulte sencillo, lo más conveniente es agrupar las distintas aplicaciones o funcionalidades según el área de la domótica de que se trate. Así pues, comenzamos definiendo las funciones que se desean implementar en la vivienda:

4.3.1.1. *Confort*

- **Control de la iluminación:**

Se permitirá control de puntos de luz en la vivienda mediante la página web habilitada para esta aplicación. Esta web contendrá dos modos. Uno que permita el accionamiento del sistema pulsando un botón en la web y otro que hará que el sistema se active a través de sensores de presencia.

- **Control de persianas:**

Será posible el control de las persianas de la vivienda, al igual que en el caso anterior por medio de un dispositivo remoto a través de la red.

- **Control de calefacción:**

Para mayor comodidad, se controlará el sistema de calefacción por medio de un dispositivo conectado a internet de manera que la estancia permanezca a la temperatura requerida por el usuario.

- **Control de riego:**

Para automatización del riego del jardín se usará un sistema que accione el riego desde un dispositivo conectado a internet función de los horarios programados, además de contener un modo manual que permite conectar el sistema en cualquier momento.

4.3.1.2. Gestión de la Seguridad

- **Detección de fugas de agua, humo y monóxido de carbono:**

Mediante detectores de inundación en la cocina y baño y sensores de humo en las cocinas. En estos casos, se enviará un mensaje de peligro al correo electrónico del usuario.

En estos casos, el propio sistema efectuará las correspondientes acciones correctivas (cierre de válvulas, etc.).

- **Detección de incendios:**

Mediante detectores de humo en las cocinas. El objetivo de la instalación domótica en este caso es enviar un mensaje informando del problema e intentar solucionar este en el menor tiempo posible.

- **Detección de intrusos:**

Se instalará un completo sistema de seguridad. Este sistema incluirá sensores de puertas y ventanas abiertas, y de presencia. Éstos detectaran cualquier anomalía que pueda darse y se enviara un mensaje de correo electrónico informando de esto.

- **Videovigilancia:**

Como se ha ido viendo, se instalarán cámaras de seguridad en algunas de las estancias de la vivienda. El objetivo es que el usuario pueda controlar su vivienda de forma remota.

PLIEGO DE CONDICIONES

4.1. NORMATIVA

El reglamento que regula los sistemas domóticos es el nuevo Reglamento Electrotécnico para Baja Tensión [REBT], que entró en vigor el 18 de septiembre del 2003. En concreto, la instrucción ICT-BT-51: Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios. Según esta:

Todos los receptores, actuadores, sensores y controladores deben cumplir, una vez instalados, los requisitos de Seguridad y Compatibilidad Electromagnética que le sean de aplicación, conforme a lo establecido en la legislación nacional que desarrolla la Directiva de Baja Tensión (73/23/CEE) y la Directiva de Compatibilidad Electromagnética (89/336/CEE).

Emisiones EMC: EN 50081-1: Estándar de emisión genérica. Inmunidad EMC: EN 50082-1: Estándar de inmunidad genérica.

Todos los receptores, actuadores, sensores y controladores que se instalen en el sistema, deberán incorporar instrucciones o referencias a las condiciones de instalación y uso que deban cumplirse para garantizar la seguridad y compatibilidad electromagnética de la instalación, como, por ejemplo, tipos de cable a utilizar, aislamiento mínimo, apantallamientos, filtros y otras informaciones relevantes para realizar la instalación. En el caso de que no se requieran condiciones especiales de instalación, esta circunstancia deberá indicarse expresamente en las instrucciones.

Cuando el sistema domótico esté alimentado por muy baja tensión o la interconexión entre nodos y dispositivos de entrada este realizada en muy baja tensión, las instalaciones e interconexiones entre dichos elementos seguirán lo indicado en la ITC-BT-36.

Los dispositivos que inyectan en la instalación de baja tensión señales de 3 kHz hasta 148,5 kHz cumplirán lo establecido en la norma UNE-EN 50065-1.

(Señalización de baja frecuencia en instalaciones eléctricas) en lo relativo a compatibilidad electromagnética.

Los emisores de los sistemas que usan señales de radiofrecuencia o señales de telecomunicación deberán cumplir la legislación nacional vigente del "Cuadro Nacional de Atribución de Frecuencias de Ordenación de las Telecomunicaciones".

ANEXO 5. DATOS DE LA MEMORIA

5.1. MARCO TEÓRICO

5.1.1. PROTOCOLOS

5.1.1.1. Z-WAVE

Z-WAVE es un protocolo inalámbrico usado principalmente en domótica. Está diseñado para funcionar a 900MHz, lo que le permite evitar interferencias y permite que las ondas puedan superar barreras arquitectónicas, como paredes, y otros obstáculos al tener una mayor capacidad de penetración por su longitud de onda.(pentadom, 2014).

La topología de la red es de tipo malla. Cada nodo se puede comportar como emisor o receptor.

Para que la red funcione sin fallos el rango de operación no debe superar los 30 metros y su mayor desventaja es que, al estar la comunicación constantemente activa el consumo de energía es mayor.(Vega, 2014)

5.1.1.2. ZIGBEE

Zigbee es un protocolo de comunicación inalámbrico que utiliza la banda ISM. Funciona a través de 16 canales en el rango de frecuencias de 2.4GHz, con un ancho de banda de 5MHz. Para evitar colisiones en la transmisión de datos, utiliza el protocolo CSMA/CA. Sigue el estándar IEEE 802.15.4-2003.

Tipos de dispositivos Zigbee:

- El Coordinador ZigBee es el encargado del control de la red y de determinar los caminos de comunicación.
- Por debajo tenemos el router ZigBee que interconecta los nodos.
- El Dispositivo final ZigBee recibe información y se comunica el nodo padre.(«ZigBee, la tecnología barata para comunicación inalámbrica», 2015)

DATOS DE LA MEMORIA

5.1.1.3. LORAWAN

LoRaWAN es un protocolo de comunicación diseñado para IOT. Este protocolo ha sido desarrollado para la transmisión de pequeños paquetes de datos, a gran distancia, consumiendo una cantidad ínfima de energía, lo que permite maximizar las baterías. Utiliza la comunicación radio como tecnología para la transmisión de datos. Está basado en el estándar IEEE 802.15.4g y utiliza las bandas ISM 868 MHz y 915 MHz. («LoRaWAN – Long Range Wide Area Network», 2016)

La tipología usada en este protocolo se llama “estrella de estrellas”. Las puertas de enlace se encargan de enviar y recibir los mensajes entre los nodos y la red central o nube de datos.

Las puertas de enlace se comunican a través del protocolo LoRaWAN con los nodos. La mayor parte de la comunicación a los nodos es bidireccional, aunque LoRaWAN soporta transmisiones multidifusión para actualizar nodos o distribuir mensajes masivos de emergencia.(«LoRaWANTM | RS Components», 2016)

5.1.1.4. SIGFOX

Sigfox es una red radio pensada para aplicaciones IOT. Su funcionamiento está basado en el envío de mensajes de pequeño tamaño (12 bits), permitiendo el envío de hasta 140 mensajes al día. Al ser estos mensajes de un tamaño reducido, la vida de las baterías se puede prolongar en el tiempo hasta los 15 años.

Utiliza las frecuencias ISM, en Europa de la ISM 868MHZ, que blinda a la red frente a interferencias.

El control de los dispositivos se hace desde una interfaz API subida a la nube, desarrollada para Sigfox. Además, permite conectividad Ultra Narrow Band direccional.

Su alcance es de 10km en zonas urbanas y en zonas rurales, debido a un menor número de interferencias, de 30 a 50km. («Así es SIGFOX, la red alternativa para el Internet de las cosas», 2016)

5.1.1.5. WIFI

Wi-Fi es un estándar de conexión inalámbrica que surgió para interconectar los distintos dispositivos electrónicos sin necesidad de cables

Los estándares más extendidos comercialmente en la actualidad son el IEEE 802.11b, el IEEE 802.11g y el IEEE 802.11n, con unas velocidades respectivamente de 11Mbits/s, 54Mbits/s, 600Mbits/s. Usan la banda universal de 2.4GHz, aunque el IEEE 802.11n también es compatible con la banda de 5GHz.

En la actualidad se está implementando en el mercado el nuevo estándar IEEE 802.1ac, que permite velocidades de hasta 1000Mbps en la banda de los 5 GHz.(«¿Sabes qué es el estándar Wi-Fi y cuantos tipos hay?», 2014)

5.1.2. Sensores y detectores

5.1.2.1. Reed Switch

Reed Switch es un interruptor que se ve activado en presencia de un campo magnético. Este dispositivo es colocado en ventanas para la detección de intrusiones en viviendas.



Ilustración 1-Reed Switch comercial («¿Que es un Reed Switch?», 2010)

DATOS DE LA MEMORIA

Cuando es normalmente abierto, los contactos se cierran en presencia de un campo magnético. Cuando es normalmente cerrado, se abren en presencia de un campo magnético.

Su composición se basa en dos láminas compuestas de Ni y Fe selladas en una capsula de vidrio, que es llenado de nitrógeno u otro gas inerte. Entre las dos láminas se deja un pequeño espacio, que en presencia de un campo magnético desaparece (las láminas se tocan).

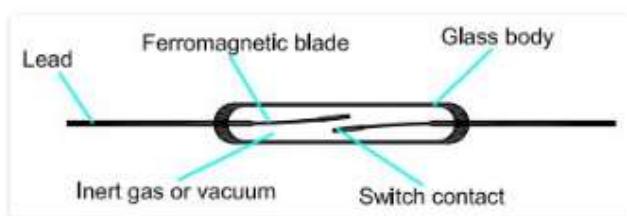


Ilustración 2-Reed Switch normalmente abierto (NO)(«¿Que es un Reed Switch?», 2010)

Existen también Reed Switch doble NO y NC. De forma que, en el estado inicial, el Reed switch se encuentra en estado normalmente cerrado y en presencia de un campo magnético el contacto es desplazado al normalmente abierto. .(«¿Que es un Reed Switch?», 2010)

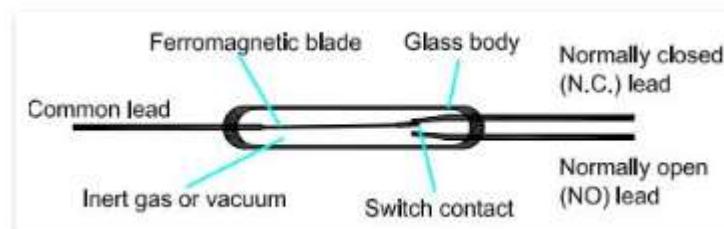


Ilustración 3-Reed Switch doble («¿Que es un Reed Switch?», 2010)

5.1.2.2. Sensor de luz LDR

Una resistencia LDR, es una resistencia que varía en función a la luz que incide sobre ella. Se utiliza para controlar sistemas que dependen de la luz. Una aplicación típica de esta resistencia es el control de una persiana en función de la luz que penetre en la resistencia.

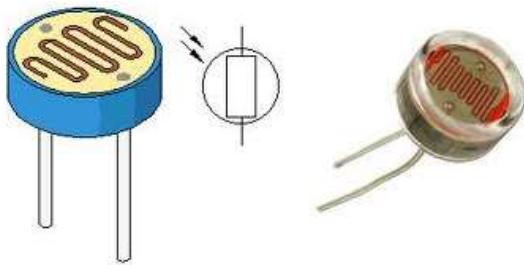


Ilustración 4- LDR («LDR-fotoresistencia», 2010)

Cuando un LDR no detecta radiación lumínica los electrones internos están firmemente unidos, pero cuando incide radiación lumínica la energía que incide sobre el sensor libera electrones variando la resistencia. Dependiendo del material con el que este fabrica, permite la detección de una longitud de onda determinada. El tiempo de respuesta del sensor es del orden de la décima de segundo.(«LDR-fotoresistencia», 2014)

5.1.2.3. Detectores de humo

5.1.2.3.1. De tipo de cámara de ionización

Estas cámaras perciben en el aire variaciones en la corriente de iones y activan la alarma cuando se produce un incendio. («Tipología de los detectores de incendios de incendios y gases», 2012).

DATOS DE LA MEMORIA



Ilustración 5- Cámara de ionización («Tipología de los detectores de incendios de incendios y gases», 2012)

5.1.2.3.2. De tipo óptico

Este tipo de detector está basado en células fotoeléctricas que al oscurecerse por el humo o iluminarse por reflexión eléctrica se activan.

Son adecuados para grandes superficies y cuando la altura del techo es superior a 12 metros.(«Tipología de los detectores de incendios de incendios y gases», 2012)



Ilustración 6- Detector óptico («Tipología de los detectores de incendios de incendios y gases», 2012)

5.1.2.3.3. Detectores de humo por aspiración

Una serie de tuberías toman muestras del aire y lo envían a un sensor interno que avisa de si se ha producido un incendio. («Tipología de los detectores de incendios de incendios y gases», 2012)



Ilustración 7- Detectores de humo por aspiración («Tipología de los detectores de incendios de incendios y gases», 2012)

5.1.2.4. Detectores de Monóxido de Carbono

En el mercado existen gran variedad de dispositivos, que detectan este gas mortal, invisible e inodoro.

La empresa RODA es una suministradora de estos sensores basados en un circuito electrónico, que mide la variación de resistencia entre electrodos de un sensor en estado sólido calefaccionado.

Los elementos que pueden generar este gas son estufas, hogares, salamandras, tiro balanceado y cocinas. Es conveniente colocar un sensor de este tipo para prevenir riesgos innecesarios.(«Detectores de monóxido de carbono», 2007)

DATOS DE LA MEMORIA

5.1.2.5. Detectores de presencia

Sensor que detecta movimiento. Su uso más comercializado es la automatización de luminaria. Son motivo un ahorro en energía.(«¿Cómo funciona un detector de presencia?», 2013)

5.1.2.5.1. Detectores de presencia de infrarrojos

Detectan la diferencia de calor emitido por el ser humano en movimiento con respecto al vacío. No atraviesan obstáculos y tienen una sensibilidad baja.(«¿Cómo funciona un detector de presencia?», 2013)

5.1.2.5.2. Detectores por ultrasonidos

En este caso se usan ondas que están fuera del espectro auditivo. Para detectar presencia de humanos se utiliza la diferencia entre la frecuencia de la onda emitida y la recibida. Atraviesa obstáculos.(«¿Cómo funciona un detector de presencia?», 2013)

5.1.2.5.3. Detectores Diales

Combina las dos tecnologías anteriores aprovechando la mejor de cada tecnología. Se utilizan cuando requiere un alto grado de detección.(«¿Cómo funciona un detector de presencia?», 2013)

5.1.2.6. Termostatos Inteligentes

Un termostato inteligente usa sensores (no solo de temperatura) y otras funciones que permiten al usuario un mayor control y automatización del sistema de calefacción de la vivienda.

Consta de dos elementos separados: relé y termostato. En un termostato convencional estos elementos están juntos debido a que no suele haber conexión a 220v cerca para conectar el relé donde se conecta el termostato

Se pueden conectar a dispositivos externos ya están provistos de conexión Wi-fi que permite el control desde cualquier lugar.(«Cinco razones para comprar un termostato inteligente», 2015)

5.1.2.7. Sensores de gas

Estos sensores sirven para indicar la presencia de algún gas en un lugar específico y miden la concentración de dicho gas.

Se pueden clasificar por:

Según su modo de operación: Sensores que funcionan por absorción, reacción química y contacto con el gas.

Según la tecnología: Sensores basados en infrarrojos o ultrasonidos.

Según el gas que detectan: Sensores de gases combustibles y sensores de gases tóxicos.(«Qué son los sensores de gas? – 330ohms», 2016)

5.1.2.7.1. Sensores Semiconductores

Sensores basados en semiconductores de óxido metal. Cuando el gas toca el sensor se produce una reacción química que reduce la resistencia del sensor.(«Qué son los sensores de gas? – 330ohms», 2016)

DATOS DE LA MEMORIA

5.1.2.7.2. Sensores Infrarrojos

Funcionan con emisiones y receptores de infrarrojos. Cuando un gas se encuentra entre el emisor y el receptor, la potencia de transmisión entre los elementos varía detectando el tipo de gas. Solo pueden detectar gases que contengan más de un tipo de átomo.(«Qué son los sensores de gas? – 330ohms», 2016)

5.1.2.7.3. Sensores Ultrasónicos

Estos sensores usan emisiones ultrasónicas para detectar cambios en el ruido de fondo del ambiente en donde se encuentren, principalmente para detectar fugas en tuberías. La fuga de un gas genera un sonido ultrasónico en un rango promedio entre los 25 kHz y los 10 MHz.(«Qué son los sensores de gas? – 330ohms», 2016)

5.1.2.7.4. Sensores Electroquímicos

Contienen dos electrodos divididos por una capa de electrolitos que puede ser sólido, líquido o gaseosa. Cuando una tensión de polarización se aplica a los electrodos se produce una reacción de reducción-oxidación que genera una corriente eléctrica directamente proporcional a la concentración de gas.(«Qué son los sensores de gas? – 330ohms», 2016)

5.1.2.7.5. Sensores Catalíticos

Su funcionamiento se basa en la oxidación del gas vía catalítica. Estos sensores se componen de dos bobinas de platino, una cubierta de un material catalizador que es la que varía su resistencia si detecta gas y otra que sirve como elemento de referencia. Por medio del calor generado, a partir de una corriente eléctrica en la superficie del elemento catalítico, se oxida el gas.(«Qué son los sensores de gas? – 330ohms», 2016)

5.1.2.8. Sensores de Inundación

Estos sensores detectan fugas de agua usando una sonda y contienen un sensor que analiza la señal que proviene de la sonda.

Mediante un sensor se puede mandar una orden desde el sistema de control que haga que el suministro de agua se corte ó otras ordenes programadas en caso de fuga.(«Sensores: Tipos y Funcionalidades - DomoPrac - Domótica práctica paso a paso», 2016)

5.1.2.9. Videovigilancia

5.1.2.9.1. Videovigilancia por CCTV.

CCTV o circuito cerrado de televisión, es una tecnología usada para vigilar por medio de cámaras, a través de monitor y algunos sistemas. Contienen detectores de movimiento que lo convierten en un sistema más sofisticado, ya que avisa del movimiento de elementos que se pueden escapar al ángulo de visión de la cámara, proporcionando mayor seguridad. Se le denomina sistema cerrado porque sus elementos están enlazados y está pensado para un número de espectadores limitados.(«VideoVigilancia», 2015)

5.1.2.9.2. Videovigilancia por internet

Consiste en la visualización del sistema de seguridad vía internet, con un ordenador u otro dispositivo con conexión a internet.

Consta del sistema de cámaras y un dispositivo de gestión de los mismos a través de internet para su visualización(«VideoVigilancia», 2015).

DATOS DE LA MEMORIA

5.1.3. Microcontroladores

5.1.3.1. ARDUINO

Arduino es una placa programable que contiene entradas y salidas tanto digitales como analógicas. Es una placa open hardware y su precio es muy bajo comparado con otras placas programables.

A un Arduino se le pueden conectar sensores que le permiten a la placa realizar determinadas acciones, una vez que se ha programado.

Existen diversidad de placas, cada una orientada al proyecto que se va a realizar. La más comercial es la placa Arduino uno R3. Para proyectos donde se requiere más memoria o entradas y salidas, se puede usar un Arduino mega entre otras.



Ilustración 8-Tipos de Arduino («¿Qué es Arduino? ¿Qué modelo comprar?», 2013)

La placa mini05 está orientado a proyectos donde se va a dejar permanente conectada la placa y no se tiene pensado reprogramarlo. Hay que soldar sus terminales y que comprar un adaptador USB.(«¿Qué es Arduino? ¿Qué modelo comprar?», 2013)

5.1.3.2. *ESP 8266*

Un ESP8266 es un chip que proporciona de conexión WIFI a un microcontrolador, aunque en sí mismo este chip es un microcontrolador que no necesita de otro para ser programado.

Este chip contiene de conexiones GPIO, que permiten ser configuradas para conectar sensores. Contiene un modo sleep, que permite que en este estado no consuma prácticamente batería y se despierte solo en el momento necesario. Esta característica hace que sea ideal para proyectos IOT.

Permite conectarse a un punto de acceso WIFI mediante comando AT. El módulo es capaz de enviar información a una dirección IP y un puerto una vez programado. Soporta los protocolos 802.11 b/g/n.(«Esp8266 Modulo Wifi | Blog de PatagoniaTec Electronica», 2015)

5.1.3.3. *Raspberry Pi*

Raspberry Pi es una placa de bajo coste que se diseñó para que fuera como un ordenador en miniatura, eliminando todos los accesorios que no afecten al funcionamiento de la placa.(«RASPBERRY PI – Historia de la Informática», 2013)

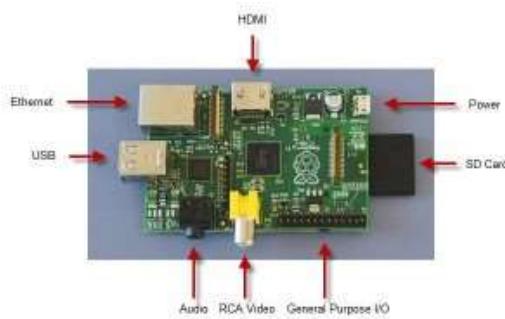


Ilustración 9-Raspberry PI

El modelo B es un ejemplo de raspberry PI. Contiene un chip integrado Broadcom BCM2835, que contiene un procesador ARM11 con varias frecuencias de funcionamiento y la posibilidad de subirla hasta 1 GHz, procesador gráfico y 512 de memoria RAM. Posee salidas para conectar accesorios, conexión Ethernet y puertos USB para conectar un adaptador WIFI. Se le puede instalar un sistema operativo, generalmente LINUX.

Existen también otros modelos con características superiores como la raspberry 2 y la raspberry 3.(«Raspberry Pi 3, 2, 1 y Zero en español», 2014)

DATOS DE LA MEMORIA

5.2. ELECCIÓN DE COMPONENTES

5.2.1. Sistema de accionamiento de luces

Como microcontrolador se ha utilizado la placa de desarrollo Nodemcu. La elección de esta placa viene dada porque es un microcontrolador programable que contiene un módulo Wifi integrado y se puede encontrar en el mercado por alrededor de 5 euros. Este precio y sus características lo hacen óptimo para esta aplicación. Este es el microcontrolador utilizado para todo el proyecto.



Ilustración 10-Nodemcu

Se ha usado un sensor PIR de presencia para encender las luces en el momento que detecte una persona. En concreto, se ha utilizado un PIR HC-SR501 que permite ajustar tanto el tiempo entre mediciones como la sensibilidad. Su precio es de 2 euros aproximadamente. La elección viene dada por su capacidad de ajuste y por su precio, siendo el más barato del mercado.



Ilustración 11-PIR

Para que las luces no se enciendan cuando la luz natural sea suficiente, se ha usado una resistencia LDR que permite saber la cantidad de luz que hay en un lugar.



Ilustración 12-LDR

El sistema de luces se activará gracias a un relé que accionará las luces cuando llegue la orden desde el microcontrolador. El relé utilizado es el SLA-05VDC-SL-C.



Ilustración 13-Relé([«SLA-05VDC-SL-C_Datasheet.pdf»](#), 2015)

Este relé se puede usar para activar dispositivos de hasta 30 Amperios. Una bombilla LED de 18W consume 1.5 A por lo que se podría llegar a encender hasta 20 bombillas de estas características, simultáneamente con un solo relé. Lo que quiere decir que este componente sirve para cubrir las necesidades de la aplicación (corte y suministro de iluminación) en la mayoría de las estancias que se utilizan comúnmente en edificación. Esto hace que el sistema sea muy versátil en este tipo de proyectos.

DATOS DE LA MEMORIA

5.2.2. Sistema de detección de humo

Este sistema cuenta con dos tipos de componentes. Sensores de humo mq-2 y un microcontrolador nodemcu. El sensor de humo ha sido elegido, por ser un sensor que se usa en equipos industriales y que se puede conseguir por un precio de aproximadamente 4 euros. Es el componente más barato usado en este tipo de aplicaciones. Además, este sensor permite ser conectado de manera analógica o digital.



Ilustración 14 -MQ-2

5.2.3. Sistema de detección de monóxido de carbono

Este sistema está compuesto por un microcontrolador nodemcu y un sensor de monóxido de carbono MQ-7. La elección de este componente viene dada por su alta utilización a nivel comercial y que además permite ser ajustado para lograr una detección óptima. Su precio es de unos 3.50 €.

Debido a su calidad y su precio este componente es óptimo para la aplicación a realizar.



Ilustración 15- MQ-7

5.2.4. Sistema de calefacción

Se han evaluado los sensores de temperatura que existen en el mercado dentro de los más utilizados, para elegir el que más se pueda adaptar al sistema. Se describen a continuación los que se han preseleccionado por su precio y características:

LM35

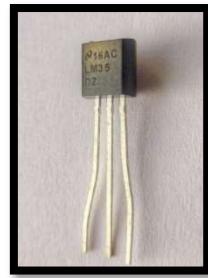


Ilustración 16-LM35(Dilmen, 2014)

- Precisión: $\pm 0,5^{\circ}\text{C}$
- Conversión: $10\text{mV}/^{\circ}\text{C}$
- Tiempo de respuesta: (100%):4 min.
- Offset: 0
- Precio: 2 €

Viene calibrado en grados Celsius y la salida es analógica. Tiene un rango de medición de entre 2° y 150°C . No lee valores negativos de temperatura a menos de que se introduzca un voltaje negativo. («Sensor de temperatura, escoge el mejor para tus proyectos con Arduino», 2016)

DHT11

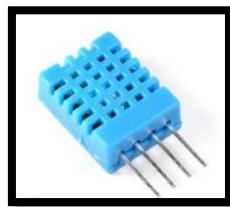


Ilustración 17- DHT11(«DHT11 Sensor – Tienda y Tutoriales Arduino», 2016)

- Voltaje de operación: de 3 V a 5,5 V

DATOS DE LA MEMORIA

- Rango de temperaturas: 0º C a 50º C
- Precisión: ±2º C
- Resolución: 8 bit
- Muestras/segundo: 0,5
- Precio: 2 €

Este sensor de temperatura viene con un sensor de humedad incorporado. Su rango de temperatura es de 0º a 50º y la salida es digital. Su mayor problema es la lentitud de las lecturas, una cada 2 segundos.(«Sensor de temperatura, escoge el mejor para tus proyectos con Arduino», 2016)

DS18B20



Ilustración 18- DS18B20 («DS18B20 Sensor de Temperatura», 2017)

- Voltaje de operación: de 3 V a 5,5 V
- Rango de temperaturas: -55º C a 125º C
- Precisión: ±0,5º C
- Resolución: 12 bit
- Precio: 2 €

Este sensor tiene un rango de temperatura de entre -55ºC y 125ºC. En hogares es raro tener temperaturas bajo cero, pero, no hay que dejar ninguna posibilidad. Su salida es digital y viene configurado por defecto con una resolución de 12 bits configurable.(«DS18B20 sensor de temperatura para líquidos con Arduino», 2017)

Puesto que todos los sensores tienen el mismo precio, se ha elegido el ds18b20 por tener un rango de temperaturas más acorde con la aplicación, además de tener mayor precisión que el resto de los sensores estudiados.

Como relé se ha elegido el SLA-05VDC-SL-C que permite activar electroválvulas de hasta 30 A.

5.2.5. Sistema de riego

Para la elección del sensor de humedad se han tenido en cuenta, gran variedad de sensores que hay en el mercado. Tras un estudio de los sensores más utilizados, se ha hecho una selección para este proyecto.

DHT 11

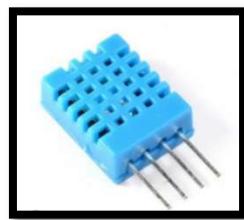


Ilustración 19 - DHT11(«DHT11 Sensor – Tienda y Tutoriales Arduino», 2016)

Este sensor sirve para medir tanto la humedad relativa como la temperatura ambiental. En este caso se estudiarán sus funciones como sensor de humedad puesto que va a ser la aplicación que reciba. Este sensor presenta una resolución del 1% por lo que no da valores decimales y su rango de valores esta entre el 20% al 95% de humedad. Se puede encontrar en el mercado por 2.55€(«Sensor DHT11 (Humedad y Temperatura) con Arduino |», 2012)

DHT22-AM2302

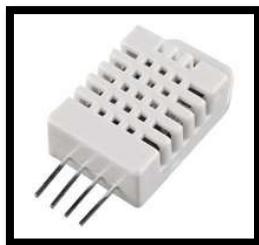


Ilustración 20- DHT22(«Medir temperatura y humedad con Arduino y sensor DHT11-DHT22», 2016)

DATOS DE LA MEMORIA

Este sensor surgió en el mercado para mejorar las características del sensor anterior. Su tiempo de respuesta está en torno a 5 segundos, por lo que, tarda alrededor de 5 segundos en reflejar un cambio en la toma de datos. Se puede encontrar en el mercado por 3.87€.(«Características DHT22 - Sensor de temperatura y humedad», 2014)

SHT75

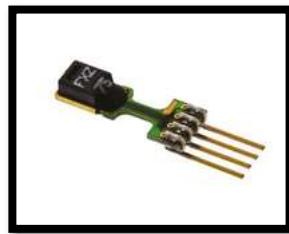


Ilustración 21- SHT75(«SHT75 Datasheet PDF - Sensirion - Pinout,Circuit - FindIC.us», 2007.)

Este sensor cuenta con una resolución del 0.05% y, por lo tanto, ofrece valores decimales. Su precisión es del ±1.8% sobre la humedad relativa. Su rango de medición es del 0% al 100% y su tiempo de respuesta es de 8 segundos. Se puede encontrar en el mercado por 36,41€.(«SHT75 - SENSIRION - Sensor de Humedad, 1.8 %, 3.3 V, 0% a 100% de Humedad Relativa, SIP, 4 Pines, 8 s | Newark element14 México», 2007)

FC-28

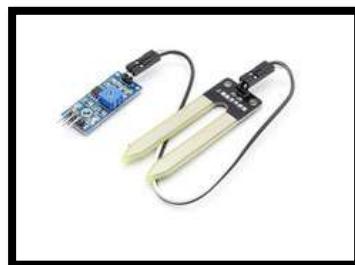


Ilustración 22- FC-28(eodos, 2015)

Sensor que mide la humedad en el suelo. No tiene precisión para medir la humedad absoluta en el suelo, pero para un sistema de riego no es necesario. Los valores obtenidos van desde 0 sumergido en agua a 1023 en aire seco. Usando una relación se puede obtener el porcentaje de humedad. Su precio en el mercado es de 1.80€.(«4. Lectura de sensores», 2015)

DATOS DE LA MEMORIA

Después de analizar varios de los sensores que se encuentran disponibles en el mercado, se ha elegido el sensor FC-28, puesto que es el más barato y además es fácilmente instalable en la tierra, sitio en el que irá para medir la humedad. Como microcontrolador se usará el mismo que para todas las aplicaciones anteriores por los motivos explicados.

Para accionar las válvulas de riego se usarán cuatro relés SLA-05VDC-SL-C de 30A, debido a que este relé puede accionar la mayoría de electroválvulas del mercado y el precio es de 2€.

5.2.6. Sistema de detección de fugas de agua

Para la elección del sensor de agua, se han estudiado las opciones del mercado para seleccionar el que más se adapte al proyecto. Por lo general se usan sensores de boyas que flotan cuando el agua entra en contacto con ellos. Un ejemplo de este tipo de dispositivo es el siguiente.

LB58

Este dispositivo de la empresa CableMatic, está basado en 3 boyas que, al ascender, activan la señal del sensor. Su precio es de 39.33€.

Los dispositivos de tipo boyas, son por lo general más caros y necesitan que el nivel de agua sea lo suficientemente alto para que empiecen a detectar, por lo que se ha desechado para este proyecto.

Por ello se ha optado por un dispositivo más económico que detecte el agua a un nivel cercano al suelo.

Sensor S0023



Ilustración 23- Sensor de agua

Este sensor, que comercializa la empresa Solectro Shop, puede ser usado tanto para detectar fugas de agua, como para medir el nivel del agua. Su precio es de 1.99€.

DATOS DE LA MEMORIA

5.2.7. Sistema de control de persianas

Como motor, se ha elegido un motor paso a paso con reductora. En el mercado hay un gran número de motores, pero se ha elegido el motor debido a dos factores, el par y al precio.

Para calcular el par motor se han usado las fórmulas que se emplean para el cálculo de persianas.

$$\text{Peso}(kg) = \text{ancho}(m) * \text{alto}(m) * k$$

Siendo k el coeficiente según el material:

MATERIAL	K
PVC	5 – 6 kg/m ²
ALUMINIO	5 – 6 kg/m ²
ACERO	13 – 14 kg/m ²
MADERA	12 – 13 kg/m ²

Tabla 1-Coeficiente K según material

Por ejemplo, para una persiana de aluminio o PVC, materiales más usados hoy en día, con unas dimensiones de 2 metros de ancho y 3 metros de largo y tiene un eje de 50 mm de diámetro se obtiene:

$$\text{Peso}(kg) = 2(m) * 3(m) * 5 = 30 Kg$$

Atendiendo al peso calculado y al diámetro del eje, se puede recurrir a una tabla que permite determinar el par en función del peso.

DATOS DE LA MEMORIA

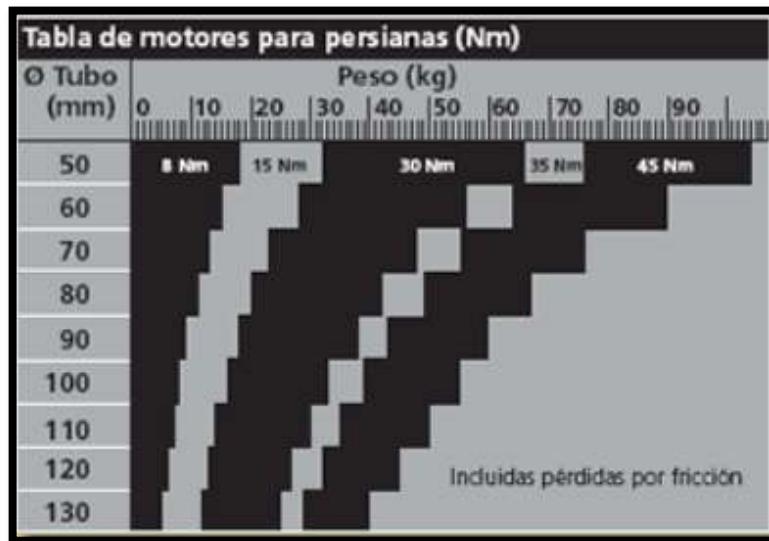


Ilustración 24-Tabla de motor según persiana(«Elige motor adecuado al peso y eje de la persiana», 2007)

Con el ejemplo calculado y viendo la tabla (Ilustración 122) se obtendría que con un motor de 15Nm el funcionamiento del sistema sería óptimo. Este ejemplo ha sido calculado con una persiana de gran longitud, para tener en cuenta un caso desfavorable.

El motor elegido ha sido un motor con reductora, NEMA 17 con reductora 100:1, que tiene un par de 25Nm.



Ilustración 25- Motor paso a paso NEMA 17

Este motor ha sido elegido porque dentro de la gran variedad de motores que existen en el mercado, ha sido el más barato que he encontrado con estas características. El precio es de 48 euros.

DATOS DE LA MEMORIA

Para controlar este motor se necesita un driver. Dentro de los drivers de bajo precio para motores paso a paso, los más famosos son el A4988 o el DRV8825. Los dos permiten el control de motores paso a paso de forma sencilla, pero el DRV8825 permite manejar motores de mayor potencia. En cuanto al precio, los dos drivers cuestan similar (el A4988 cuesta 3€ aproximadamente y el DRV8825 cuesta 4.60€ aproximadamente). Se ha elegido el DRV8825 debido al precio y a que permite una corriente de pico 2.5 A, suficiente para soportar la corriente del motor de 1.68A.

5.2.8. Sistema de alarma

Para detectar la presencia de un sujeto cuando se esté fuera de casa, se ha usado un detector PIR. En concreto se ha usado un sensor HC-501, debido a que es el sensor de presencia más barato del mercado, y permite el ajuste tanto de la sensibilidad como el tiempo entre mediciones.



Ilustración 26- PIR

Para detectar si las puertas y ventanas están abiertas o cerradas, se ha usado un Reed Switch. Después de haber realizado un estudio de mercado y viendo que la gran mayoría de estos dispositivos cuesta alrededor de 10 € (algunos incluso más). Se ha elegido el más barato encontrado y que cumple con su cometido a la perfección. El dispositivo elegido es el que se muestra en la imagen, que se puede conseguir por 2€. Si se hubiera elegido otro componente diferente a este, el precio rondaría los 10 €. Esto me parece una elección de peso, puesto que, a más elementos a detectar, mayor es la diferencia de precio.

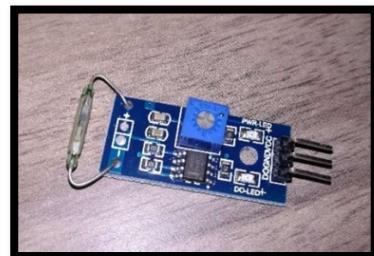


Ilustración 27- Reed Switch

También, se ha añadido una cámara que permitirá visualizar la vivienda desde la web. Por facilidad y porque integra conexión internet, se ha elegido una cámara IP. Como existen gran variedad de cámaras se elegido una que no suponga un gran incremento en el coste del sistema. Esta cámara es una D-link 930LB.



Ilustración 28-D-link930LB

DATOS DE LA MEMORIA

5.3. CONSUMOS ENERGÉTICOS

5.3.1. Sistema de accionamiento de luces

Componente	Unidades	Consumo Unidad(A)	Consumo Total (A)
Nodemcu V3	5	0.110A	0.550A
PIR	24	0.065A	1.56A
LDR	1	8.57×10^{-4} A	8.57×10^{-4} A
RELE SLA-05VDC-SL-C	24	0.089A	2.136A
TOTAL	4.246A		

Tabla 2-Consumo sistema de accionamiento luces

5.3.2. Sistema de detección de humo

Componente	Unidades	Consumo Unidad(A)	Consumo Total (A)
Nodemcu V3	1	0.110A	0.110A
MQ-2	2	0.180A	0.360A
TOTAL	0.470A		

Tabla 3- Consumo sistema de detección de humo

5.3.3. Sistema de detección de CO

Componente	Unidades	Precio unidad	Precio total
Nodemcu V3	1	5.20€	5.20€
MQ-7	1	3.58€	3.58€
TOTAL	8.78€		

Tabla 4-Precio sistema de detección de monóxido de carbono

5.3.4. Sistema de calefacción

Componente	Unidades	Consumo Unidad(A)	Consumo Total (A)
Nodemcu V3	2	0.110A	0.220A
RELÉ	4	0.089A	0.356A
DS18B20	4	1.5*10^-3A	6*10^-3A
TOTAL	0.582A		

Tabla 5-Consumo sistema de calefacción

DATOS DE LA MEMORIA

5.3.5. Sistema de riego

Componente	Unidades	Consumo	Consumo Total
		Unidad(A)	(A)
Nodemcu V3	1	0.110A	0.110A
Relé	4	0.08931A	0.357A
FC-28	4	0.100A	0.400A
TOTAL	0.867A		

Tabla 6-Consumo sistema de riego

5.3.6. Sistema de detección de fugas de agua

Componente	Unidades	Consumo	Consumo Total
		Unidad(A)	(A)
Nodemcu V3	1	0.110A	0.110A
Relé	1	0.089A	0.089A
Sensor agua	7	0.020A	0.140A
TOTAL	0.339A		

Tabla 7-Consumo sistema de fugas de agua

5.3.7. Sistema de control de persianas

Componente	Unidades	Consumo Unidad(A)	Consumo Total (A)
Nodemcu V3	3	0.110A	0.330A
DRV8825	16	0.008A	0.126A
TOTAL	0.456A		

Tabla 8-Consumos sistema control persianas

5.3.8. Sistema de alarma

Componente	Unidades	Precio unidad	Precio total
Nodemcu V3	1	0.110A	0.110A
PIR	6	0.065A	0.390A
TOTAL	0.500A		

Tabla 9-Consumo subsistema de presencia

Componente	Unidades	Consumo Unidad(A)	Consumo Total (A)
Nodemcu V3	2	0.110A	0.440A
Reed Switch	17	$1*10^{-9}$ A	$1.7*10^{-8}$
TOTAL	0.440A		

Tabla 10-Consumo subsistema estado de puertas y ventanas

DATOS DE LA MEMORIA

5.3.9. Cálculo de sistema de alimentación

Después de analizar los consumos de los sistemas de forma individual, se pueden englobar en una sola tabla, para saber el consumo total de todos los sistemas.

Sistema	Consumo
Sistema de accionamiento de luces	4.246A
Sistema de detección de humo	0.470A
Sistema de detección de CO	0.217A
Sistema de calefacción	0.582A
Sistema de riego	0.867A
Sistema fugas de agua	0.339A
Sistema control de persianas	0.456A
Subsistema de detección de presencia	0.500A
Subsistema estado ventanas y puertas	0.440A
TOTAL	8.117A

Tabla 11-Resumen general de consumos eléctricos

Una vez que se tiene conocimiento de estos datos, se puede analizar que alimentación es más conveniente para este sistema, atendiendo a determinados factores analizados.

5.3.9.1. Piles

Se va a realizar el cálculo del sistema usando pilas, para ello se van a tomar dos de las aplicaciones elaboradas, para determinación que duración tendrían las pilas. La primera aplicación será el sistema de luces que consume 4.246A, siendo la aplicación que más consume, aunque se han programado interrupciones del microcontrolador que alargarían la vida de las pilas. La segunda aplicación será el sistema de riego que, aunque es la segunda que más consume, su valor se encuentra cerca del rango de las demás aplicaciones.

Las pilas analizadas serán de litio, ya que su capacidad es muy superior al resto de pilas y estas aplicaciones tienen un gran consumo eléctrico. Si se conectan dos pilas de 3V en serie, se obtienen aproximadamente 6V, que quitando las pérdidas que se obtienen de la conexión, sirven para alimentar al microcontrolador. Además, de que el microcontrolador tiene un regulador que regula el voltaje que le llega, ajustándolo al nivel que necesita 5v. El resto de componentes también pueden ser alimentados con este voltaje.



Ilustración 29-Pilo Litio de la marca VARTA para aplicaciones electrónicas.(«Pila litio industria CR AA 3V 2Ah», 2016)

Por ejemplo, si se usa la pila que se muestra en la imagen de una capacidad de 2Ah (una capacidad que se encuentra dentro del rango de este tipo de pilas) para realizar los cálculos de duración de la pila, se obtiene:

$$\frac{\text{Capacidad de la batería}(mAh)}{\text{Consumo del dispositivo}} = \text{Duración}$$

DATOS DE LA MEMORIA

$$\text{Sistema de riego} \rightarrow \frac{2000}{867} = 2.30 \text{ horas}$$

$$\text{Sistema de luces} \rightarrow \frac{2000}{4246} = 0.471 \text{ horas}$$

Como se puede observar, conectar el sistema con pilas no tendría sentido ya que las pilas no durarían más de dos horas. Se podría colocar un mayor número de pilas en conexión serie-paralelo, para aumentar el tiempo de duración, pero un sistema de este tipo requiere que las pilas sean cambiadas cada varios años, cosa que no se consigue de ningún modo.

5.3.9.2. Fuente de alimentación

Debido a que el uso de pilas es inviable, se usaran fuentes de alimentación. El voltaje de la fuente de alimentación es de 5v y de corriente nominal de 10A para que pueda soportar la corriente del sistema de 8.117A. El precio de esta fuente de alimentación 14.47€ y permite la alimentación de todo el sistema con una sola fuente, exceptuando los motores.



Ilustración 30-Fuente de alimentación 50W MeanWell(«LED Fuente de alimentación 50W 5V 10A ; MeanWell, RS-50-5: Amazon.es: Bricolaje y herramientas», 2015)

Componente	Unidades	Precio unidad	Precio total
Fuente de alimentación 50W	1	14.47€	14.47€
TOTAL	14.47€		

Tabla 12-Precio fuente alimentación sistema

Para los motores se usarán fuentes de alimentación de 12V y 2A, puesto que el consumo del motor es 1.68A y son alimentados a 12V. La fuente de alimentación elegida es la que se muestra en la imagen, de la empresa SOLDIAL. La elección viene dada por su precio 4.38€.

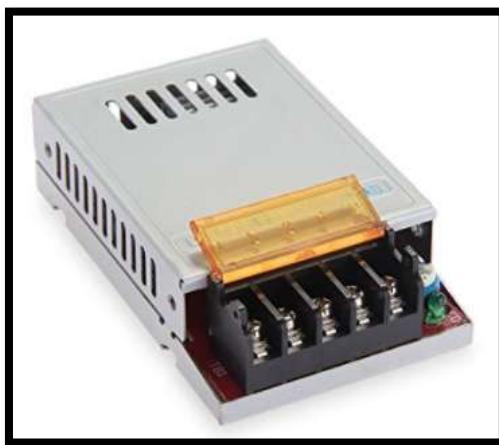


Tabla 13-Fuente de alimentación motores («SODIAL(R)24W Conducto Fuente de alimentacion Transformador DC 12V 2A por Banda Luz LED Lampara: Amazon.es: Bricolaje y herramientas», 2016)

Componente	Unidades	Precio unidad	Precio total
Fuente de alimentación 24W	16	14.47€	231.52€
TOTAL	231.52€		

DATOS DE LA MEMORIA

5.4. PROGRAMACIÓN

5.4.1. Configuración del microcontrolador

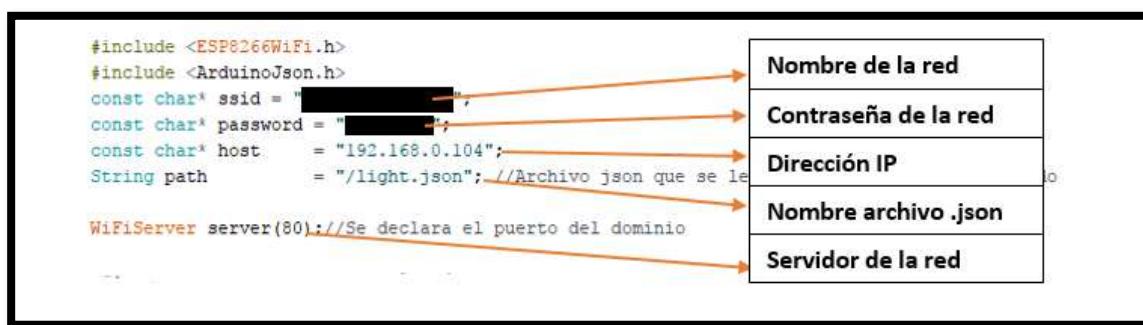


Ilustración 31-configuración Wifi Nodemcu

Para conectar con la red Wifi, se proporcionan los datos de la red (SSID, contraseña, IP y puerto). Cada placa tiene asignado el archivo JSON o la página web de la cual, lee o envía información. Al ejemplo de la imagen, se le asigna un archivo JSON que contiene la información que lee.

Para que la placa mande o reciba datos de la página asignada, debe introducirse una orden con un protocolo para que se puedan comunicar.

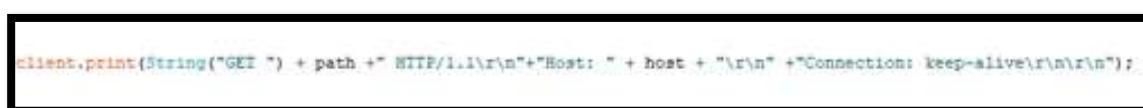


Ilustración 32-orden de envío y lectura de datos

La función “client.print”, sirve para imprimir o recibir datos de la página asignada. Dentro de esta, se introducen los datos para hacer una solicitud HTTP. Esta solicitud puede ser “GET” o “POST” en caso de recibir o de enviar datos según corresponda respectivamente.

DATOS DE LA MEMORIA

Para recibir la información que contienen los archivos JSON, se usa la orden “client.available()”. Esta orden sirve para especificar si hay un usuario usando el servidor y en este caso, se ejecutan las órdenes que contiene.

```
while(client.available()){

    String line = client.readStringUntil('\r');
    Serial.print(line);
    if (section=="header") {

        if (line=="\n") {
            section="json";
        }
    }
    else if (section=="json") {
        section="ignore";
        String result = line.substring(1);
    }
}
```

Ilustración 33- Orden para identificar a cliente conectado

Un string llamado “result”, guarda los datos que son demandados de la petición realizada por la orden “client.print”, para posteriormente, ser usados para diferentes órdenes que le son dadas al microcontrolador.

5.4.2. Base de datos

Para almacenar los datos de los sensores conectados al microcontrolador, se utiliza una base de datos. Apache es un servidor web HTTP que contiene el sistema de gestión de bases de datos MYSQL. Este sistema es el que se utiliza para almacenar los datos de los sensores conectados a la placa de desarrollo Nodemcu.

La estructura de la base datos que se utiliza en el proyecto es la que se explica en la siguiente imagen:

DATOS DE LA MEMORIA

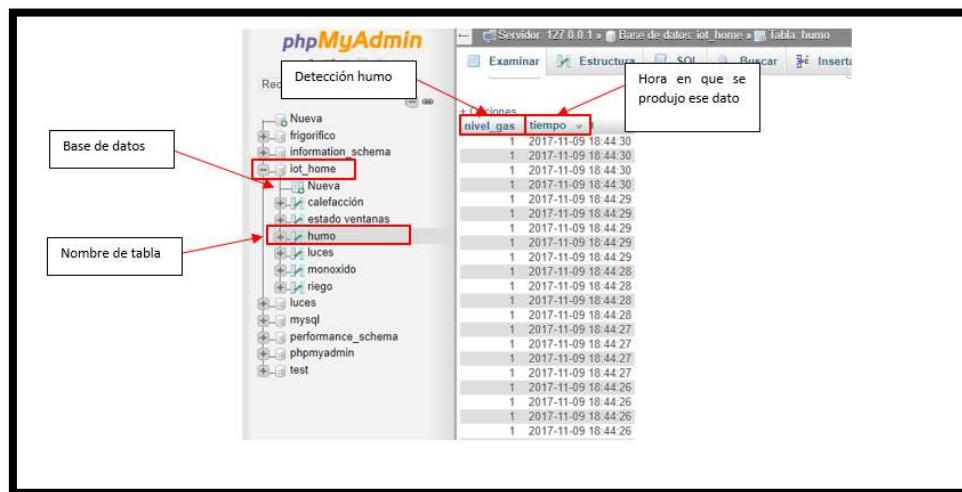


Ilustración 34-Estructura base de datos

La base de datos que contiene los datos de todas las aplicaciones realizadas se llama "iot_home". La base de datos está compuesta de tablas. Cada tabla corresponde a una aplicación IOT.

Una tabla contiene filas y columnas que corresponden a los datos de los sensores. Para la elaboración de este proyecto, además de la toma de datos de los sensores, se ha generado una columna para identificar la hora en la que se produjo el dato introducido.

Para enviar datos a la base de datos, se usa un formulario PHP que recoge los datos que vienen del microcontrolador y los introduce en la base de datos.

```

<?php
$hostDb="localhost";
$nombreDb="iot_home";
$usuario="root";
$clave="";
$conexion = mysqli_connect($hostDb, $usuario, $clave);
mysqli_select_db($conexion,"iot_home");
mysqli_query($conexion,"SET NAMES 'utf8'");
$humo = $_POST["nivel_gas"];
$mysql = "INSERT INTO `humo` (`nivel_gas`, `tiempo`) VALUES ('$humo', CURRENT_TIMESTAMP);";
mysqli_query ($conexion,$mysql);
mysqli_close($conexion);

```

Ilustración 35- Formulario para envío de datos a la base de datos

Para conectar el microcontrolador con la base de datos, se incluye la ubicación de la misma, el nombre de la tabla, además de los datos para acceder a ella. En este proyecto los datos de acceso no se han modificado, por lo que se usan los que vienen por defecto para conectar el microcontrolador con el servidor Apache, almacenado en el pc personal.

DATOS DE LA MEMORIA

Para conectar el microcontrolador con la base de datos, es necesario utilizar un formulario PHP y las funciones que este lenguaje proporciona. En este ejemplo, la variable “\$humo” contiene los datos que le llegan del microcontrolador. La forma de enviar los datos se ha explicado anteriormente y consiste en la orden “client.print” y el uso de “POST”.

La variable “\$mysqli”, que se muestra en la imagen anterior (ilustración 8), contiene la fórmula que da el gestor de base de datos, para introducir datos dentro de la tabla. La variable “\$humo”, mediante “POST”, envía los datos que recibe del microcontrolador a la columna de la base de datos “nivel_gas”. Finalmente, se cierra la conexión con el servidor.

5.4.3. Sistema de accionamiento luces

El sistema compuesto del microcontrolador y el LDR que se usa para detectar la luz natural fuera de la vivienda, es configurado para enviar los datos a una base de datos. El sensor está configurado de forma que cuando la luz sea máxima, el valor dado sea 5, y cuando no exista presencia de luz el valor sea 0. Después de realizar varias pruebas, se ha llegado a la conclusión de que cuando el valor del sensor sea menor que 1, se permita el encendido del sistema con el sensor PIR.

```
int sensorValue = analogRead(A0);
float LDR = sensorValue * (5.0 / 1023.0);
```

Ilustración 36-Configuración LDR

DATOS DE LA MEMORIA

Los datos de este sensor son enviados a una base datos tal y como explica en el punto 5.4.2

The screenshot shows the phpMyAdmin interface. On the left, there's a tree view of databases: 'iot_home' is expanded, showing tables like 'Nueva', 'frigorífico', 'information_schema', 'humo', 'inundación', 'luces', 'luz', 'monóxido', 'riegos', and 'seguridad'. The 'luz' table is selected. On the right, the main area has tabs for 'Examinar', 'Estructura', and 'SQL'. The 'Examinar' tab is active, displaying a warning message: 'La selección actual no contiene una columna'. Below it, a green checkmark says 'Mostrando filas 0 - 24 (total de 28567, La'. A SQL query is shown: 'SELECT * FROM `luz` ORDER BY `hora` DESC'. At the bottom, there's a table with three rows of data:

luz	hora
1.89	2018-01-16 22:03:53
1.87	2018-01-16 21:59:02
2.11	2018-01-16 21:54:02

Ilustración 37- Base de datos LDR

Como se muestra en la imagen anterior, se recibe un dato cada 5 minutos. Esto es debido, a que se ha configurado el microcontrolador para mandar un dato y ponerse en modo reposo. Cada 5 minutos el microcontrolador se conectará y mandará un dato. De esta forma se ahorra energía.

```
ESP.deepSleep(3e8, WAKE_RF_DEFAULT);
```

Ilustración 38-Forma de mandar a reposar al microcontrolador

Además, se guarda el valor del sensor en un archivo "luces.json" para poderlo utilizar en la segunda placa.

```
$str_datos = file_get_contents("luces.json");
$datos = json_decode($str_datos,true);

$datos["luces"][1] = $POST['luz'];

$fh = fopen("luces.json", 'w')
    or die("Error al abrir fichero de salida");
fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
fclose($fh);

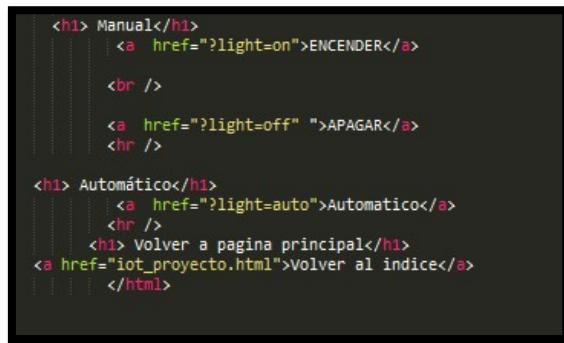
mysqli_query ($conexion,$mysql);
mysqli_close($conexion);
```

Ilustración 39-Forma de guardar valor del sensor en archivo JSON

Una vez que se ha explicado cómo se obtienen los valores de la luz, se va a pasar a explicar el funcionamiento de la aplicación;

DATOS DE LA MEMORIA

Para acceder a los modos de funcionamiento de la aplicación, se ha sido diseñado una web mediante código PHP combinado con HTML. El funcionamiento de esta web es el siguiente:



```
<h1> Manual</h1>
    <a href="?light=on">ENCENDER</a>
    <br />
    <a href="?light=off" ">APAGAR</a>
    <hr />

<h1> Automático</h1>
    <a href="?light=auto">Automatico</a>
    <hr />
    <h1> Volver a pagina principal</h1>
    <a href="iot_proyecto.html">Volver al indice</a>
</html>
```

Ilustración 40-Código de modos de uso

Cada botón tiene asignada una referencia “href”, al pulsar uno de los botones (encender, apagar o automático), la dirección contenida, en la barra de direcciones de la web, se completa con la referencia “href” que tiene asignado ese botón (“href=?light=on”, “href=?light=off” o “href=?light=auto””).

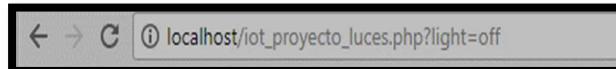


Ilustración 41- Barra de direcciones al pulsar botón apagar

Para poder tomar el valor de la barra de direcciones, se declara una variable “light” que mediante el uso de “GET”, guarda el valor de la referencia en esta variable.

```
$light =$_GET["light"];
```

Ilustración 42-Variable con el valor de la barra de direcciones

Dependiendo del valor que tome “light”, se cumple una de las condiciones “if” de la imagen que hace que se guarde el valor en un archivo JSON, para por leerlo con el microcontrolador. Este archivo se llama “luces.json”.

DATOS DE LA MEMORIA

```
else if ( $light == "off") {  
  
    $str_datos = file_get_contents("luces.json");  
    $datos = json_decode($str_datos,true);  
  
    $datos["luces"][0] =>$_GET["light"];  
  
    $fh = fopen("luces.json", 'w')  
        or die("Error al abrir fichero de salida");  
    fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));  
    fclose($fh);  
}
```

Ilustración 43- Forma de guardar datos en archivo JSON

En el microcontrolador, mediante el uso del método explicado en apartado 5.4.1, se leen los datos que llegan desde el formulario PHP y se guardan en el string “result”.

```
String result = line.substring(1);
```

Ilustración 44- Variable que contiene los datos del archivo JSON

Como el string es un array de dos datos (el dato del modo y el dato del sensor), se ha usado un substring para extraer el dato concreto que se necesita.

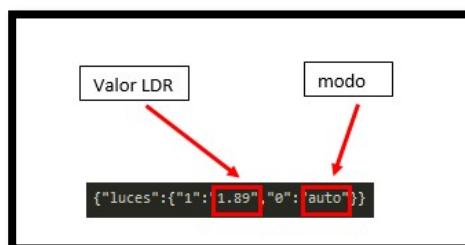


Ilustración 45- Formato archivo JSON

Se compara el resultado que llega desde la web con “on” ó “off” mediante un condicional “if” y en función de la condición que se cumple, se encienden o se apagan las luces.

```
if (result.substring(26, 28)=="on")  
{  
    digitalWrite(led, HIGH);  
    digitalWrite(Relay, HIGH);  
  
}  
if (result.substring(26,28)=="of")  
{  
    digitalWrite(led, LOW);  
    digitalWrite(Relay, LOW);  
  
}
```

Ilustración 46- Condiciones programadas en el microcontrolador

DATOS DE LA MEMORIA

En el modo automático, se usa el PIR para encender las luces y la información del segundo sistema colocado en el exterior mediante un sensor LDR. Cuando se activa el modo automático, si el sensor de luz detecta que hay insuficiencia de esta, y el PIR detecta a una persona, se activa el relé que enciende las luces y el Led de señalización. Al pasar unos segundos, si el PIR deja de detectar la presencia de una persona, el relé se desconecta y se apagan las luces y el led de señalización.

```
if (result.substring(26, 28)=="au" && luz1<1.00)
{
    if( val== HIGH )
    {
        Serial.print("apagado");
        digitalWrite(led, HIGH);
        digitalWrite(Relay, HIGH);
        delay(10);
    }
    if(val==LOW)
    {
        digitalWrite(led, LOW);
        digitalWrite(Relay, LOW);
    }
}
```

Ilustración 47- Condiciones programadas para el modo automático

Para regular la sensibilidad y el tiempo entre disparos del PIR, se utilizan los tornillos que contiene este dispositivo para ajustarlo de la manera que mejor convenga en cada espacio.

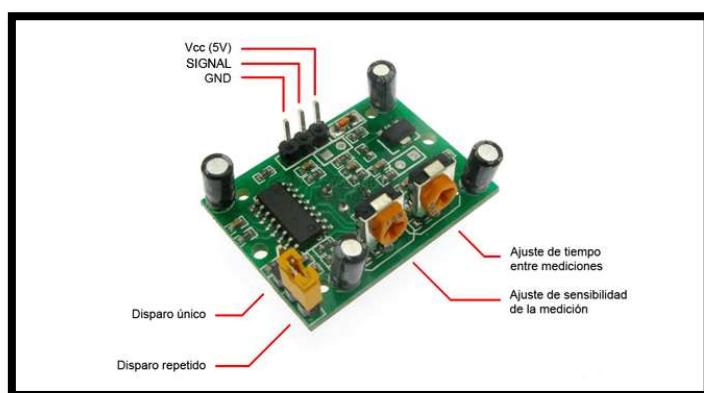


Ilustración 48- Configuración sensor PIR («Detector de movimiento con Arduino y sensor PIR», 2015)

DATOS DE LA MEMORIA

5.4.4. Sistema de detección de humo

El sensor de detección humo, MQ-2 tiene 4 pines; Los dos pines de alimentación, un pin para el envío de datos analógicos y un pin para enviar información de modo digital.

En este caso se ha usado el pin digital. Cuando el sensor manda el valor "1" al microcontrolador, significa, que no detecta humo en el ambiente. Cuando se envía el valor "0", detecta humo en el ambiente. Se puede calibrar la sensibilidad del sensor por medio de un potenciómetro y así, marcar el umbral de lo que se considera un incendio o por el contrario un nivel de humo despreciable para considerarlo alarma de incendio, como puede ser el humo de la cocina o de un cigarrillo.

Para leer los datos del sensor se ha creado una función en el microcontrolador, con la configuración del sensor.

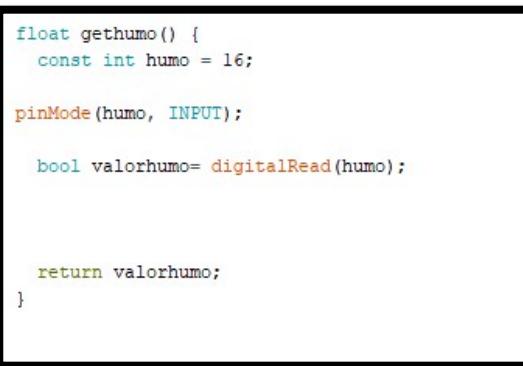


Ilustración 49- Función para devolver datos de humo.

Una vez configurado el sensor, los datos del mismo son enviados a la base de datos con el método explicado en el apartado 5.4.2.

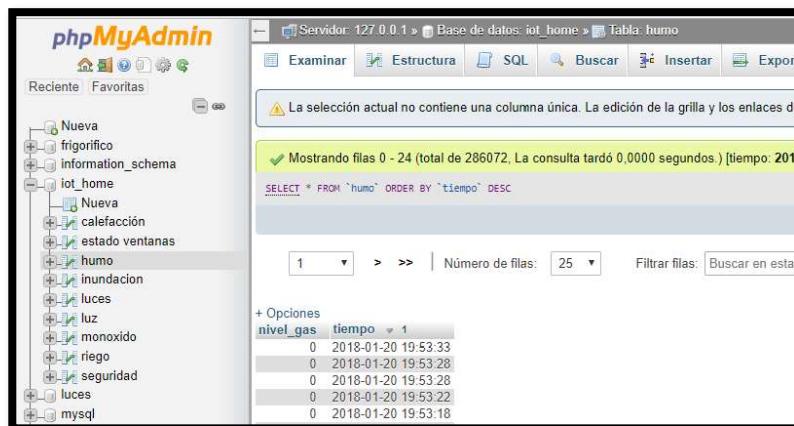


Ilustración 50- Base de datos sistema de humo

DATOS DE LA MEMORIA

El usuario tiene acceso a la web que se me muestra en la ilustración 32. En función de si el valor que llega del sensor es “1” ó “0”, el mensaje que aparece en la web cambiará.



Ilustración 51- Interfaz donde se muestran los datos

Si el dato es “1”, el mensaje que aparecerá será el mostrado en la imagen anterior “SIN RIESGO DE INCENDIO”. En caso de que el valor sea “0” el sensor estará detectando humo y desde la web se mostrará un mensaje de peligro incendio.

Para que la página web muestre los mensajes que se han descrito anteriormente, se ha configurado una página PHP como sigue:

```
$query = "SELECT * FROM `humo` ORDER BY `tiempo` DESC";
$result = mysqli_query($conexion, $query);
```

Ilustración 52- Obtención de los datos almacenados en la base de datos

Mediante el uso de la variable “query” se hace una petición para obtener los datos “humo” y “tiempo” de la base de datos y con la variable “result”, se guardan los datos que contiene la base de datos, conectándose con ella mediante la variable “\$conexión” tal y como se explica en el punto 5.4.2.

Mediante una condición “if” se compara el valor que llega de la base de datos tal y como se muestra en la siguiente imagen y dependiendo de este valor, se muestra por pantalla el mensaje que corresponde a cada estado.

- 1 – SIN RIESGO DE HUMO
- 0 - ALTA CONCENTRACIÓN DE HUMO

Además, de aparecer este mensaje. se guarda el valor “1” ó “0” dentro de un archivo JSON.

DATOS DE LA MEMORIA

```
if ($row["nivel_gas"]==1)
{
echo "SIN RIESGO DE INCENDIO";

$file = fopen("humo.json", "w") or die("can't open file");
fwrite($file, '1');
fclose($file);
}
if ($row["nivel_gas"]==0)
{
echo "ALTA CONCENTRACIÓN DE HUMO";

$file = fopen("humo.json", "w") or die("can't open file");
fwrite($file, '0');
}
```

Ilustración 53- Condiciones que se cumplen dependiendo del valor del sensor

Un segundo archivo PHP, se ha configurado para leer el estado del archivo JSON y, en caso de que el valor contenido en el mismo sea “0”, mande un mensaje de peligro al correo electrónico, para informar al usuario, sin que este tenga que estar pendiente de la página elaborada para el control del humo.

```
$jsondata = file_get_contents('humo.json');

$data = json_decode($jsondata, true);
header("Refresh: 1; URL='http://localhost/iot_proyecto_humo4.php'");
echo $jsondata;
if($data=="0")
{
$to = "    @hotmail.com";
$subject = "nivel_humo_alto ";
$message = "humo por encima de los niveles permitidos";

mail($to, $subject, $message);

echo ("mensaje enviado");

}

sleep(60); # un paro de 60 segundos antes de volver a iniciar las instrucciones
?>
```

Ilustración 54- Archivo para envío de correo electrónico

Como se muestra en la imagen, cuando el valor del sensor es “0”, se envía un mensaje al servicio de correos Hotmail, con el asunto “nivel_humo_alto” y con el mensaje “humo por encima de los niveles permitidos”.

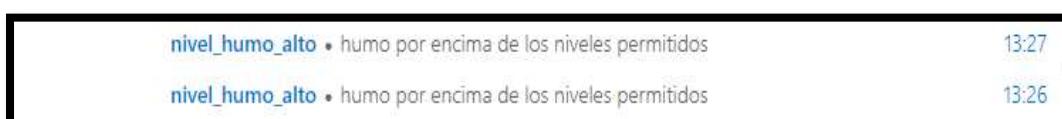
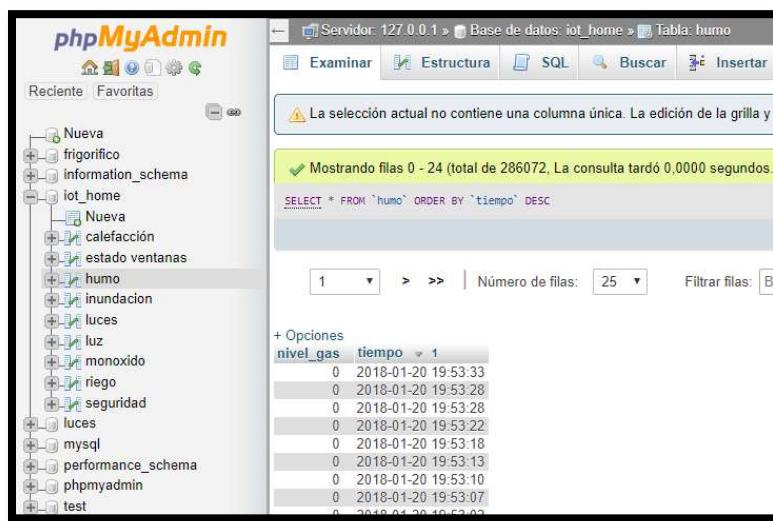


Ilustración 55- Formato de correo electrónico

5.4.5. Sistema de detección de monóxido de carbono

La configuración para que este sistema funcione, es idéntica a la que se explica en la aplicación sistema de detección de humo. La base de datos que se usa en este sistema es la creada para esta aplicación.



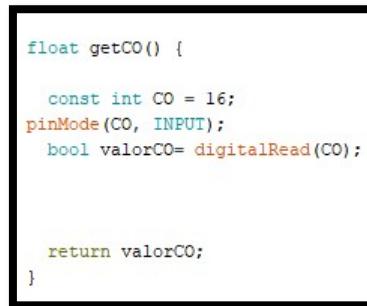
The screenshot shows the phpMyAdmin interface. On the left, the database structure is displayed under the 'iot_home' database, including tables like 'humo', 'luz', 'seguridad', etc. In the center, a query result for the 'humo' table is shown:

```
SELECT * FROM `humo` ORDER BY `tiempo` DESC
```

nivel_gas	tiempo
0	2018-01-20 19:53:33
0	2018-01-20 19:53:28
0	2018-01-20 19:53:28
0	2018-01-20 19:53:22
0	2018-01-20 19:53:18
0	2018-01-20 19:53:13
0	2018-01-20 19:53:10
0	2018-01-20 19:53:07
0	2018-01-20 19:53:02

Ilustración 56-Base de datos monóxido de carbono

El sensor MQ-7, al igual que el MQ-2, tienen un pin digital y un pin analógico para configurar el sensor. Por facilidad en la calibración, se ha elegido el pin digital que se ha regulado mediante el potenciómetro que contiene.



```
float getCO() {  
  
    const int CO = 16;  
    pinMode(CO, INPUT);  
    bool valorCO= digitalRead(CO);  
  
  
    return valorCO;  
}
```

Ilustración 57-Función del sensor CO

La forma en que se gestionan los datos en la página, y de enviar un mensaje en caso de alarma del sistema, para la detección de monóxido de carbono, es igual al utilizado en la aplicación de detección de humo, por lo que se omitirá volver a explicarlo nuevamente para no ser redundante.

DATOS DE LA MEMORIA

5.4.6. Sistema de calefacción

Para mostrar la temperatura en la web lo primero que se ha de hacer es configurar el sensor de temperatura para que proporcione datos y poder enviarlos a la base de datos. Para tomar los valores de temperatura se ha configurado el sensor dallas 18b20. Este sensor usa una entrada digital para mandar los datos. Esto es debido a la comunicación one-wire, basada en un complejo sistema de timings en la señal entre el dispositivo emisor y el receptor. Este dispositivo cuenta con una librería hecha por la compañía del fabricante para facilitar la configuración.

Como el fabricante proporciona la librería “DallasTemperature”, esta se usará para configurar el sensor, además de la librería “OneWire” que utiliza para la comunicación.

```
#include <OneWire.h>
#include <DallasTemperature.h>
```

Ilustración 58- Librerías usadas para configuración de sensor

Dentro de la librería “DallasTemperature” se encuentran las funciones que se muestran en la imagen inferior y que sirven para obtener la temperatura.

```
float getTemperature() {
    float temp;

    DS18B20.requestTemperatures();
    temp = DS18B20.getTempCByIndex(0);
    delay(100);

    return temp;
}
```

Ilustración 59- Función para obtener datos de temperatura

Una vez configurado el sensor para leer la temperatura, se ha configurado una base de datos como se explica en el punto 5.4.2 con los datos de temperatura y como se puede ver en la imagen siguiente (ilustración 60). Además, se ha configurado el microcontrolador para que mande un dato cada minuto y se ponga en modo reposo después, para el ahorro de energía del sistema.

DATOS DE LA MEMORIA

The screenshot shows the phpMyAdmin interface connected to a local server (127.0.0.1) and the database 'iot_home'. The left sidebar lists the database structure with the 'calefaccion' table selected. The main area displays a query results page for the 'calefaccion' table, ordered by 'tiempo' in descending order (DESC). The results show temperature values (e.g., 22.94, 23, 23.12) and dates/times (e.g., 2018-01-17 20:02:14, 2018-01-17 20:01:14, 2018-01-17 20:00:13, 2018-01-17 19:59:14, 2018-01-17 19:58:35).

Ilustración 60- Base de datos sistema de calefacción

Para mostrar los datos de temperatura al usuario, se ha configurado un web a la que este puede tener acceso, y observar la temperatura del lugar donde se ha colocado el sensor.

The screenshot shows a simple web page with a single line of text: "la temperatura es 12 a la hora (2017-11-30 18:06:52)". Below this text are two hyperlinks: "Introducir temperatura a alcanzar" and "Volver a menu".

Ilustración 61- Interfaz con dato de temperatura en tiempo real

Para mostrar los datos de la temperatura en la web, esta ha sido programada mediante código PHP, para mostrar los datos al usuario.

```
$query = "SELECT * FROM `calefaccion` ORDER BY `tiempo` DESC";
$result = mysqli_query($conexion, $query);
```

Ilustración 62-Obtención de valores de la base de datos

La variable "\$query" hace una petición a la base de datos para extraer los datos de temperatura("calefacción") y tiempo("tiempo").

La variable "\$result", sirve para recibir los datos de la base de datos haciendo uso de la variable anterior y de la conexión a la base de datos mediante el método explicado en el punto 5.4.2.

DATOS DE LA MEMORIA

Una vez hecho esto, se ha creado un array para mostrar los datos por pantalla. Además, la página se refresca cada dos segundos para mostrar los datos en tiempo real.

```
$row = mysqli_fetch_array($result, MYSQLI_BOTH);
printf ("la temperatura es: %s a la hora (%s)\n", $row["temperatura"], $row["tiempo"]);
header("Refresh: 2; URL='http://192.168.0.104/iot_proyecto_calefaccion2.php'");
```

Ilustración 63- Órdenes para mostrar temperatura en la web

Para que el dato de temperatura pueda ser leído desde el sistema que acciona la caldera, se ha guardado el dato de temperatura en un archivo JSON.

```
$str_datos = file_get_contents("cal.json");
$datos = json_decode($str_datos,true);

$datos["temperatura"][@] =$_POST['temperatura'];

$fh = fopen("cal.json", 'w')
or die("Error al abrir fichero de salida");
fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
fclose($fh);
```

Ilustración 64- Forma de guardar dato temperatura en archivo JSON

Por otra parte, el usuario puede acceder a una página web donde introducir la temperatura que quiere en la zona deseada, accionando el relé hasta alcanzar dicha temperatura. Para ello se ha proporcionado una caja de introducción de datos para anotar el valor de la temperatura.

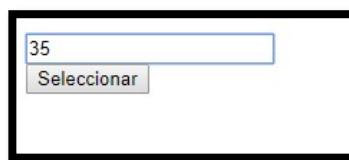


Ilustración 65- Caja para enviar temperatura

Esta caja se ha configurado para guardar el dato de la temperatura escrito dentro de un archivo JSON y posteriormente este dato, ser leído desde el microcontrolador para que realice una acción en función al dato enviado.

```
<form action="iot_proyecto_calefaccion2.php" method="post" name="caja">
<input type="text" name="caja_texto" id="valor1" value="" /> <br />
<input type="submit" value="Seleccionar" />
</form>
```

Ilustración 66- Código de programación caja

DATOS DE LA MEMORIA

Esta caja está configurada para que cuando se presione el botón enviar, la variable "caja" tome el valor que se ha escrito por pantalla, y lo envíe a una segunda página llamada "iot_proyecto_calefaccion2.php" mediante el uso de "POST". Esta segunda página tiene programado que, si le llega un dato de la caja, lo guarde en un archivo JSON, llamado "cal.json". Esto lo hace mediante la condición "if" que se muestra en la imagen siguiente.

```
if(isset($_POST['caja_texto'])){  
    $caja = $_POST['caja_texto'];  
  
    $dato_importante=print_r(json_encode($caja));  
  
    $str_datos = file_get_contents("cal.json");  
    $datos = json_decode($str_datos,true);  
  
    $datos["temperatura"][1] =$_POST['caja_texto'];  
  
    $fh = fopen("cal.json", 'w')  
        or die("Error al abrir fichero de salida");  
    fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));  
    fclose($fh);
```

Ilustración 67- Guardado del dato de temperatura en archivo JSON

Para que el microcontrolador que acciona la caldera pueda leer el archivo JSON, este ha sido programado para leer dicho archivo, tal y como se muestra en el punto del documento 5.4.1.

El string "result" usado para guardar los datos que contiene el archivo JSON, esta vez contiene el dato de la temperatura.

Para pasar el dato a un tipo de dato numérico, se ha realizado una conversión del string a coma flotante "float" (se usa este tipo de dato para obtener el valor con decimales).

```
float termostato=result.toFloat();
```

Ilustración 68-Transformación del string en float

Una vez se tiene la temperatura objetivo, introducida desde la web y la temperatura recibida desde el sistema dedicado a tomar datos de temperatura ambiental de las plantas de la vivienda, se pueden comparar estos dos datos para activar o desactivar el relé de la caldera en función de las condiciones programadas tal y como se describe a continuación.

DATOS DE LA MEMORIA

```
if(termostato >temperature+temperature*0.05){  
    digitalWrite(myRelee, HIGH);  
    digitalWrite(16,HIGH);  
  
}  
if (termostato < temperature-temperature*0.05 ){  
    digitalWrite(myRelee, LOW);  
    digitalWrite(16,HIGH);  
  
}
```

Ilustración 69- Condiciones para encender la caldera con histéresis programada

Si la temperatura a alcanzar es mayor a la temperatura ambiente sumándole un valor de un 5% a la misma, el relé se activará encendiendo la caldera.

Si la temperatura a alcanzar es menor a la temperatura ambiente restándole un valor de un 5% a la misma, el relé se desactivará apagando la caldera.

Este 5 % se corresponde con una histéresis programada para evitar continuos encendidos y apagados de la caldera. Se deja un margen por encima y por debajo de un 5% la temperatura ambiente que hace que se eviten estos problemas de continuos encendidos y apagados.

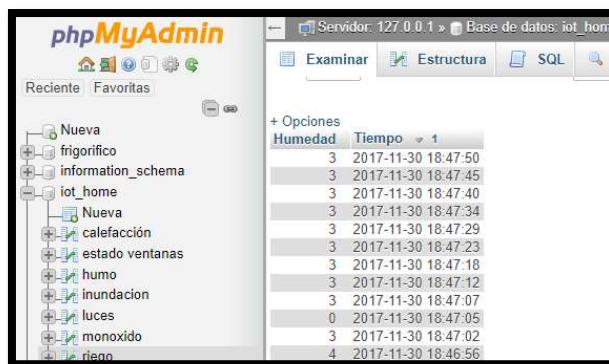
5.4.7. Sistema de riego

La interfaz general, se compone de una imagen, representativa de la aplicación, el dato de humedad que llega del sensor y las opciones para entrar en los modos.



Ilustración 70- Web general de aplicación

Para que el dato del sensor pueda ser visualizado desde la página, el primero paso es introducir los datos del sensor en la base de datos tal y como se explica en el apartado 5.4.2. A pesar de ser 4 zonas, como solo se dispone de un sensor, la tabla se compone de los datos de un solo sensor y la hora en que se introduce el dato. El dato de humedad se da en porcentaje.



Humedad	Tiempo
3	2017-11-30 18:47:50
3	2017-11-30 18:47:45
3	2017-11-30 18:47:40
3	2017-11-30 18:47:34
3	2017-11-30 18:47:29
3	2017-11-30 18:47:23
3	2017-11-30 18:47:18
3	2017-11-30 18:47:12
3	2017-11-30 18:47:07
0	2017-11-30 18:47:05
3	2017-11-30 18:47:02
4	2017-11-30 18:46:56

Ilustración 71-Base de datos riego

Para que el dato de humedad aparezca en pantalla, la interfaz ha sido programada con código PHP.

```
$query = "SELECT * FROM `riego` ORDER BY `tiempo` DESC";
$result = mysqli_query($conexion, $query);
```

Ilustración 72- Obtención de datos de la base de datos

DATOS DE LA MEMORIA

Como ocurre en algunas de las otras aplicaciones la variable “\$query” sirve para hacer una petición de los datos a base de datos y la variable “\$result” conecta con la base de datos para poder tomar los datos como se hace para introducir datos en la base de datos que se explica en el punto 5.4.2.

```
$row = mysqli_fetch_array($result, MYSQLI_BOTH);
printf (" la humedad es %s %s a la hora (%s)\n", $row["Humedad"], "%", $row["Tiempo"]);
header("Refresh: 2; URL='http://192.168.0.104/iot_proyecto_iego2.php'");
```

Ilustración 73- Forma de mostrar los datos por pantalla

Se ha creado un array, de nombre “\$row”, que sirve para almacenar los datos que llegan desde la base de datos para mostrarlos por pantalla. Finalmente se ha programado un refresco de 2 segundos para actualizar los datos.

A partir de esto, la aplicación se divide en modo manual y modo automático.

Modo manual

Al entrar en el modo manual aparecen las cuatro zonas de riego programadas.



Ilustración 74- Interfaz modo manual

Los botones son programados para que cada uno de ellos lleve a la página que contiene el código de esa zona, como se muestra en la imagen siguiente.



```
<hr/>
<h2>ENCENDER ZONA DE RIEGO</h2>
<hr />
<h3>zonas </h3>
<a href="iot_riego_manual_zon1.php">zona 1</a> <a href="iot_riego_manual_zona2.php">zona 2</a>
<a href="iot_riego_manual_zona4.php">zona 4</a>
<br/>
<hr />
<a href="iot_proyecto_riego2.php">Volver a acciones de riego</a>
<br/>
<hr />
<a href="iot_riego_manual.php">Volver a acciones de riego</a>
<br/>
<a href="iot_proyecto.html">Volver a menu</a>
</html>
```

Ilustración 75-Páginas asociadas a o botones de interfaz

Cada zona tiene asociada una página PHP. Además, se da la opción de volver a la interfaz principal de riego o a la interfaz del menú de las aplicaciones.

Cuando se pulsa el botón de una de las zonas, se dan opciones: accionar el riego o apagar el riego.

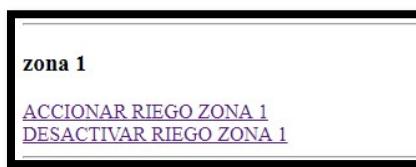


Ilustración 76-Opciones de zonas

Al pulsar uno de los botones, lo que se hace realmente es guardar un dato en un archivo JSON para que el microcontrolador pueda leer los datos.

Este archivo JSON es un array que contiene todos los datos correspondientes a la aplicación, es decir, el accionamiento de las cuatro zonas, la hora y el minuto de inicio y de final de las cuatro zonas. Finalmente, el último elemento sirve para diferenciar entre modo automático y modo manual. Cada zona tiene asignado un botón de manual o automático, aunque en la foto solo aparezca uno, por cuestiones de visibilidad de la foto en el documento.

DATOS DE LA MEMORIA

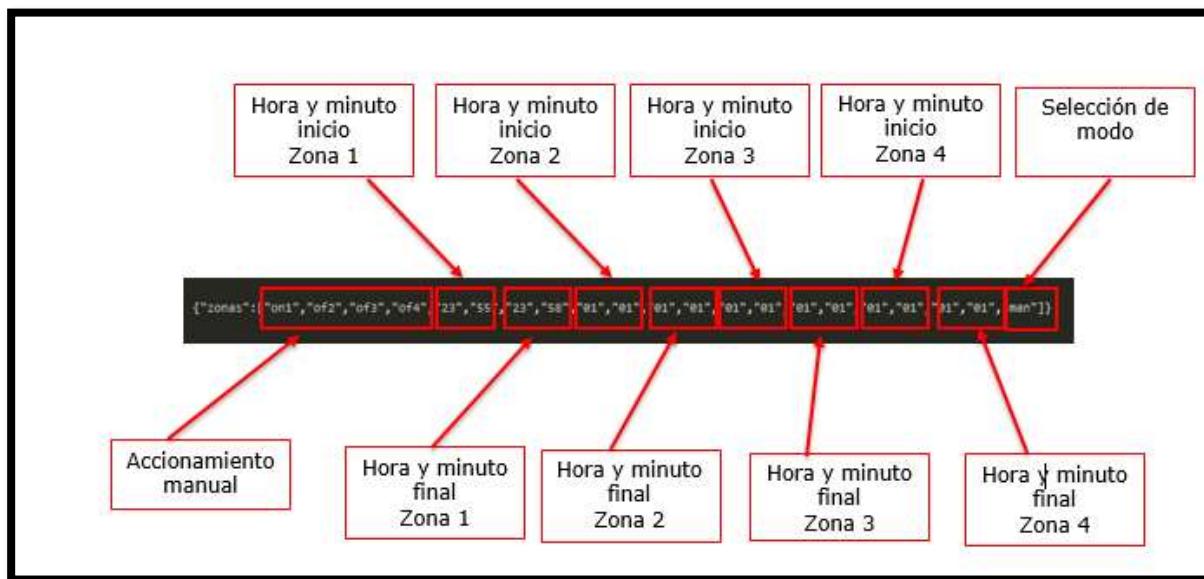


Ilustración 77- Archivo JSON

Cuando se presiona el botón de accionar el riego, la dirección de la barra de direcciones de la web se completa con el valor de la zona que se ha accionado y el valor "on" al igual que como sucede en el modo manual del sistema de luces.



Ilustración 78- Barra de direcciones web

La variable "\$zona1", guarda el valor de "on" y compara con un condicional "if" el contenido de la variable.

```
if($zonal=="on") {
    $str_datos = file_get_contents("zon1.json");
    $datos = json_decode($str_datos,true);

    // Modifica el valor, y escribe el fichero json de salida
    //
    $datos["zonas"][0] = "on1";

    $fh = fopen("zon1.json", 'w')
        or die("Error al abrir fichero de salida");
    fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
    fclose($fh);
    $datos["zonas"][20] = "man";

    $fh = fopen("zon1.json", 'w')
        or die("Error al abrir fichero de salida");
    fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
    fclose($fh);

}
```

Ilustración 79- Condiciones programas en PHP

DATOS DE LA MEMORIA

Tal como se muestra en la foto anterior, cuando se cumple la condición, se abre el contenido del archivo JSON y se guarda dentro del espacio “[0]” del array el valor “on1” y dentro del espacio [20] el valor manual. Esto es aplicable al caso de apagarse, en ese caso se guardaría el valor de “of1” dentro del espacio [0]. Si fuera la zona 2 se llenaría el espacio [1] del array con el valor de “on2” o “of2” y el elemento [21] con el valor manual y así sucesivamente con todas las zonas.

Para leer el valor del archivo desde el microcontrolador se configura para leer el archivo JSON tal y como se explica en el punto 5.4.1.

En este caso el string “result” contiene los datos del array.

```
String result = line.substring(1);
```

Ilustración 80- Variable que guarda los datos del archivo JSON

Para extraer del string los valores correspondientes a algún espacio específico del array, se usa un substring que contiene únicamente el valor de una zona concreta.

```
if (result.substring(11, 14) == "on1" && result.substring(115, 118) == "man" ) {  
    Serial.println("riegol");  
    digitalWrite(16, HIGH);  
}
```

Ilustración 81- Condicional if con substring correspondiente a zona 1

Dentro del espacio comprendido entre el 11 y el 14 del string, se encuentra el valor que corresponde al accionamiento de la zona 1 de forma manual y que será “on1” cuando se presione el botón encender o “of1” cuando se pulse el botón apagar. Además, en el espacio del array comprendido entre el elemento 115 y el 118, se encuentra el valor “man” o “aut”. Cuando desde la interfaz web se pulse el botón de accionamiento de riego, el archivo JSON contendrá los valores “on1” y “man”, cumpliéndose la función programada desde el microcontrolador y accionando el relé conectado al pin 16 y que hará que se accione la válvula de riego.

El valor de manual o automático se ha programado para que no se solapen los dos modos haciendo que se accione y apague el riego constantemente, por ser ordenado por el usuario, desde un modo accionar el riego y desde el otro apagar.

DATOS DE LA MEMORIA

Modo Automático

El modo automático sirve para programar una hora de inicio de riego y otra de final. Cada zona tiene programado un reloj para programar esa zona.



The screenshot shows a web page titled "zona1". It has two sections: "INICIO RIEGO" and "FINAL RIEGO". Both sections contain dropdown menus for "hora_inicio" and "min_inicio" or "hora_final" and "min_final". In the "FINAL RIEGO" section, the dropdown menu for "hora_final" is open, showing values from 30 to 37. A button labeled "AUTOMATICO" is visible above the dropdown. Below the dropdown is a link "Volver a menu".

Ilustración 82- Interfaz modo automático

Para realizar este formulario PHP, que permite configurar el riego de forma automática, se ha creado un desplegable para la hora y minuto de inicio y final de riego.



The screenshot shows a code editor with a title bar "Página de envío de datos". Below it, a red box highlights a portion of the code:

```
<form action="iot_proyecto_riegoAutomatico2_zona1.php" method="post">
    <h3>INICIO RIEGO</h3>
    hora_inicio:
    <select name="horaini">
        <option value="00">00</option>
        <option value="01" selected="selected">01</option>
        <option value="02">02</option>
        <option value="03">03</option>
        <option value="04">04</option>
        <option value="05">05</option>
        <option value="06">06</option>
        <option value="07">07</option>
        <option value="08">08</option>
        <option value="09">09</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>
        <option value="13">13</option>
        <option value="14">14</option>
        <option value="15">15</option>
        <option value="16">16</option>
        <option value="17">17</option>
        <option value="18">18</option>
```

Ilustración 83- PHP con valores de desplegable y forma de envío por método "POST"

DATOS DE LA MEMORIA

Cuando se selecciona un valor y se pulsa enviar, los datos son enviados a otra página (la página que se muestra en la imagen) por el método "POST" y almacenados por una variable que tiene asignado capturar el valor que le llega de la primera página.

```
$datos["zonas"][4] =$_POST['horario'];
```

Ilustración 84-Variable que contiene el valor del desplegable

Finalmente, el dato es guardado en el espacio del array que le corresponde dentro del archivo JSON, en este caso es el espacio "[4]" de dicho array, para ser leído desde el microcontrolador.

```
$fh = fopen("zon1.json", 'w')
      or die("Error al abrir fichero de salida");
fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
fclose($fh);
//$str_datos = file_get_contents("zon1.json");
//$datos = json_decode($str_datos,true);
```

Ilustración 85-Forma de guardar valor

Para leer el archivo JSON, el microcontrolador se ha configurado para leer este tipo de archivos, como se explica en el punto 5.4.1 y como se ha explicado en el modo manual de esta misma aplicación. En ese punto se explica que el string "result" contiene los datos del array y que para obtener el dato concreto que se necesita, de este string, se usa un substring.

Se ha creado un reloj que compare la hora actual con la hora programada y que pueda ejecutar la acción de riego cuando ambos valores sean iguales. Para crear el reloj se ha usado la librería NTPTIME, que es un protocolo que se ha ido extendiendo en la red para la sincronización de relojes. Para evitar derivas, utiliza datos de múltiples relojes de gran precisión en la red.

```
#include <NTPtimeESP.h>
```

Ilustración 86- Librería de reloj

Lo que hace esta librería, es mandar una petición de hora al servidor central, y devuelve la hora. Para devolver la hora de forma precisa, realiza cálculos teniendo en cuenta el tiempo que tarda en recibir y volver a mandar el mensaje con la hora y otros parámetros, para que el usuario final reciba la hora exacta.

Dentro de esta librería tiene una función para conectar con el servidor que proporciona la hora.

DATOS DE LA MEMORIA

```
NTPtime NTPch("ch.pool.ntp.org");
```

Ilustración 87- Función para conectar con el servidor

Este reloj es mundial, por lo que para dar la hora de la zona en la que te encuentras, se tiene que proporcionar la zona horaria del país. En este caso España.

```
dateTime = NTPch.getNTPtime(1.0, 1);
```

Ilustración 88-Función para dar la zona horaria del país

Una vez que se ha configurado el reloj, se puede extraer de la librería la hora, el minuto, el segundo, el año, el mes, el día, y la semana en tiempo real y almacenarlos en variables.

```
if(dateTime.valid){  
    NTPch.printDateTime(dateTime);  
  
    byte actualHour = dateTime.hour;  
    byte actualMinute = dateTime.minute;  
    byte actualSecond = dateTime.second;  
    int actualYear = dateTime.year;  
    byte actualMonth = dateTime.month;  
    byte actualDay = dateTime.day;  
    byte actualDayOfWeek = dateTime.dayOfWeek;
```

Ilustración 89- Variables para guardar datos de reloj

Con estos datos ya se puede comparar la hora enviada desde el formulario PHP con la hora actual y activar o desactivar el riego.

```
String horal= result.substring(35, 37);  
String minl= result.substring(40, 42);  
String horaf= result.substring(45, 47);  
String minf= result.substring(50, 52);  
String modo= result.substring(115, 118);
```

Ilustración 90-Variables con datos del archivo JSON

Los substring de la imagen anterior, contienen los datos de hora de inicio y de final de riego, que están dentro del array del archivo JSON, y son comparados con la hora real proporcionada por la librería "NTPtimeESP.h". El último substring contiene el dato del modo, para el modo automático es "aut". como se explica en el apartado del modo manual, este último dato sirve para que no se solapen ambos modos.

DATOS DE LA MEMORIA

Para activar el riego se ha configurado un condicional "if" cuyo cometido es que cuando la hora y minuto programado de inicio, sea igual a la hora y minuto del reloj, el riego se encienda (el relé se active) y que cuando la hora y minuto de final de riego sea igual a la hora del reloj, el riego se apague (el relé se desactive).

```
if(dateTime.hour== hourli && dateTime.minute==minuteli && result.substring(115, 118)=="aut")
{
    digitalWrite(16, HIGH);

}

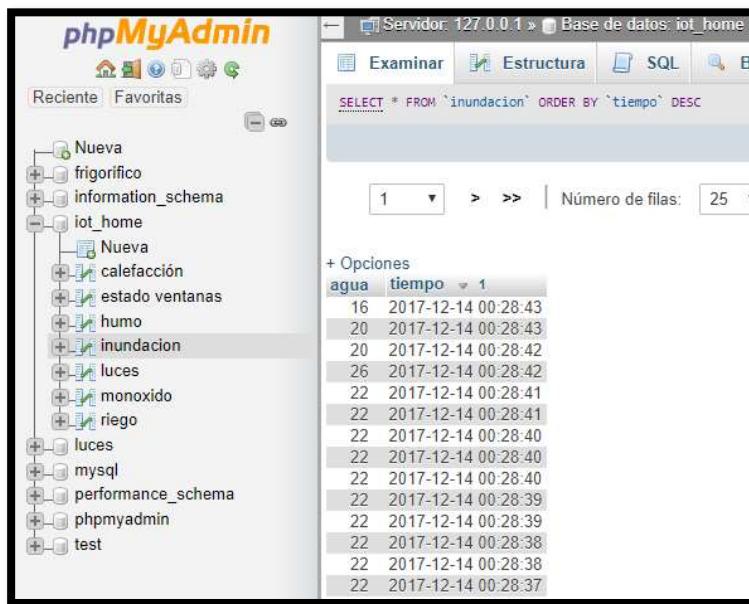
if(dateTime.hour== hourlf && dateTime.minute==minutelf && result.substring(115, 118)=="aut" )
{
    digitalWrite(16, LOW);
}
```

Ilustración 91- Condiciones para accionar y apagar el riego del modo automático

DATOS DE LA MEMORIA

5.4.8. Sistema fugas de agua

Se han introducido los datos del sensor de agua dentro de una base de datos para que, desde la web, se pueda ver el mensaje de “PELIGRO AGUA” o “sin incidencias”. Para ello se ha recurrido al método explicado en el punto 1.2.3, donde se explica cómo introducir los datos del sensor en una base de datos. El resultado se puede ver en la imagen siguiente.



agua	tiempo
16	2017-12-14 00:28:43
20	2017-12-14 00:28:43
20	2017-12-14 00:28:42
26	2017-12-14 00:28:42
22	2017-12-14 00:28:41
22	2017-12-14 00:28:41
22	2017-12-14 00:28:40
22	2017-12-14 00:28:40
22	2017-12-14 00:28:40
22	2017-12-14 00:28:39
22	2017-12-14 00:28:39
22	2017-12-14 00:28:38
22	2017-12-14 00:28:38
22	2017-12-14 00:28:37

Ilustración 92-Base de datos fuga de agua

Los datos del sensor son almacenados junto con la hora en la que se introducen.

Desde un formulario PHP se han manipulado los datos, que llegan desde la base de datos, para variar el mensaje que aparece en la web en función del valor del sensor.

Para leer los datos, que llegan de la base de datos, se hace una llamada a la tabla que contiene los datos requeridos.

```
$inundacion = $_GET["agua2"];
$query = "SELECT * FROM `inundacion` ORDER BY `tiempo` DESC";
$result = mysqli_query($conexion, $query);

/* array numérico */

$row = mysqli_fetch_array($result, MYSQLI_BOTH);
header("Refresh: 2; URL='http://192.168.0.104/iot_proyecto_agua2.php?agua2'");
```

Ilustración 93- Petición para extraer datos de la base de datos y mostrarlos en pantalla

DATOS DE LA MEMORIA

Mediante la variable “\$query”, se realiza una petición para conectar con los datos de la base de datos. La variable “\$result” recibe los datos de la base de datos, mediante la variable anterior y conectando con la base de datos, tal y como se explica en el punto 5.4.2 de conexión a una base de datos. Para almacenar los datos, ha creado un “array” que contiene los datos. Finalmente, se ordena refrescar la página cada dos segundos para mostrar los datos actuales.

Para mostrar el mensaje de “PELIGRO AGUA” o “sin incidencias” se ha estudiado como varían los valores del sensor cuando el mismo está en ausencia de agua y los valores que muestra cuando entra en contacto con el agua.

Cuando el sensor está en ausencia de agua, los valores que se muestran son inferiores al valor 100, cuando el mismo entra en contacto con el agua, el valor que se muestra es de alrededor de 400 y totalmente sumergido es de 600.

Para que, en caso de agua, se cierre rápidamente la acometida, se colocara el sensor a una altura cercana al suelo. En cuanto el sensor empiece a detectar agua, se cerrará la acometida.

En la web aparecerá un mensaje u otro dependiendo del valor del sensor, además de guardar un dato en un archivo JSON, que servirá para apagar o encender el relé que está conectado a la válvula de la acometida.

```
if ($row["agua"]>400)
{
echo "Estado:PELIGRO AGUA";

$str_datos = file_get_contents("agua.json");
$datos = json_decode($str_datos,true);

$datos["aguas"][0] = "of";
$fh = fopen("agua.json", 'w')
or die("Error al abrir fichero de salida");
fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));
fclose($fh);
}
```

Ilustración 94- Condiciones programadas cuando se detecta agua

Si el valor del sensor es mayor que 400, en pantalla aparecerá el mensaje “PELIGRO AGUA” debido al condicional “if” que se ha programado.

Además, se guardará el dato “of” en el archivo JSON que servirá para que el microcontrolador ordene desactivar el relé.

DATOS DE LA MEMORIA

Por el contrario, cuando el valor del sensor es menor de 400, ocurre lo siguiente:

```
if ($row["agua"]<400)
{
    echo "Estado:Sin incidencias";

}

printf (" a las (%s)\n", $row["tiempo"]);
```

Ilustración 95-Condiciones programadas cuando no se detecta agua

En la web se muestra el estado “sin incidencias”. En este caso no se guarda ningún valor dentro del archivo JSON, esto es debido a que para que se vuelva a activar el relé, y por lo tanto la acometida, se usa el botón rearme que se ha programado.

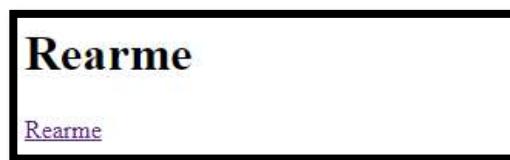


Ilustración 96-Botón de Rearme

Cuando se pulsa el botón rearme, de la web a la que tiene acceso el usuario ocurre lo siguiente. La barra de direcciones se completa con el valor "?agua2=on".



Ilustración 97-Dirección que aparece cuando se pulsa el botón rearne

Esto es debido a que el botón “rearne” tiene asociado la referencia “?agua2=on”.

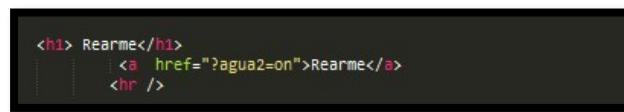


Ilustración 98- Referencia asociada a botón rearne

Se ha creado una variable que guarda el valor “on” de la barra de direcciones cuando se pulsa el botón “Rearne”. Esta variable tiene como objetivo capturar el valor de

DATOS DE LA MEMORIA

"agua2", asociado el botón "Rearme" y que aparece en la barra de direcciones cuando se pulsa dicho botón.

```
$inundacion = $_GET["agua2"];
```

Ilustración 99-Variable que toma el valor de la barra de direcciones

Cuando la variable "\$inundación" contiene el valor "on", se cumple una condición "if" programada.

```
if( $inundacion == "on") {  
  
    $str_datos = file_get_contents("agua.json");  
    $datos = json_decode($str_datos,true);  
  
    // Modifica el valor, y escribe el fichero json de salida  
    //  
    $datos["aguas"][0] = "on";  
  
    $fh = fopen("agua.json", 'w')  
    or die("Error al abrir fichero de salida");  
    fwrite($fh, json_encode($datos,JSON_UNESCAPED_UNICODE));  
    fclose($fh);
```

Ilustración 100- Condición que se cumple cuando se pulsa el botón rearne

Cuando se cumple esta condición, se guarda el valor "on" dentro del archivo JSON antes citado y esto hace que desde el microcontrolador se encienda el relé.

Además, se ha programado otra página PHP para que en caso de superar el valor 400 del sensor, se envié un mensaje de correo al usuario informando de la incidencia.

```
if ($row["agua"]>400)  
{  
  
    $to = "orte90@hotmail.com";  
    $subject = "anomalia ";  
    $message = "humo por encima de los niveles permitidos";  
  
    mail($to, $subject, $message);  
  
    echo ("mensaje enviado");  
  
    sleep(60); # un paro de 60 segundos antes de volver a iniciar las instrucciones
```

Ilustración 101- Envío de mensaje de correo cuando se detecta agua

Los datos escritos en el archivo JSON, son leído desde el microcontrolador, para encender el relé, cuando le llegue la orden desde la página web a la que tiene acceso el usuario.

DATOS DE LA MEMORIA

El microcontrolador es configurado tal y como se explica en el punto 5.4.1, donde se explica cómo extraer la información de un archivo JSON en el microcontrolador.

En este caso el String que se encarga de recolectar esa información ("result") contiene el dato de rearme del relé.

```
String result = line.substring(1);
```

Ilustración 102-String con contenido del archivo JSON

Para recopilar únicamente el dato del string, que interesa para comparar y obtener una respuesta en función del resultado, se usa un substring que permite extraer datos del string.

```
Serial.println(result.substring(25,27));
```

Ilustración 103- Substring con dato importante

Una vez extraído el dato, este se usa para comparar, mediante una condición "if", y en función del resultado, se cumple una de las condiciones de encendiendo o apagado del relé. Tal y como ya se ha explicado, el relé se encenderá cuando se pulse el botón rearne y se apagará cuando el nivel de agua sea suficiente, como para considerarlo un peligro, cerrando la válvula que tenga conectada.

```
if( result.substring(25,27)=="on")
{
digitalWrite(Relay,HIGH);
}

if( result.substring(25,27)=="of")
{
digitalWrite(Relay,LOW);
}
```

Ilustración 104- Condiciones programadas para apagar o encender el relé.

5.4.9. Sistema control de persianas

Para señalizar el sentido de giro, se han añadido dos leds. Un led señala el movimiento del motor en un sentido y el otro en el opuesto.

El microcontrolador ha sido programado para controlar el motor y hacerlo girar un determinado número de pasos, en función de la orden que se reciba desde la web. Existen varias de formas de programar este motor, excitando una a una las bobinas del motor, excitando las bobinas de dos en dos o alternando, primero una y luego dos.

- Al excitar las bobinas una a una, se obtiene una mayor velocidad, pero el par de torque es inferior.
- Al excitar las bobinas de dos en dos, se obtiene un mayor par de torque reduciendo la velocidad.
- Al excitar las bobinas de forma alternada se obtienen unas características intermedias.

Para la realización de esta aplicación y como modo de prueba, se ha usado la configuración alternada. Cambiar de una forma a otra no supondría un cambio de más de dos líneas de código.

Como el motor empleado tiene cuatro fases y se va a usar la forma alternada, se tienen 8 combinaciones. Para controlar las cuatro fases se han declarado cuatro variables que corresponden a los pines de las fases y se ha creado un array que contiene las combinaciones del modo alternado.

```
int motorPin1 = 16;  
int motorPin2 = 5;  
int motorPin3 = 4;  
int motorPin4 = 0;
```

Ilustración 105- Pines del motor

```
int lookup[8] = {B01000, B01100, B00100, B00110, B00010, B00011, B00001, B01001};
```

Ilustración 106-Array para usar las fases de forma alternada

DATOS DE LA MEMORIA

Para activar el motor y que gire, se ha creado una función que establece que bobina ha de activarse.

```
void setOutput(int out)
{
    digitalWrite(motorPin1, bitRead(lookup[out], 0));
    digitalWrite(motorPin2, bitRead(lookup[out], 1));
    digitalWrite(motorPin3, bitRead(lookup[out], 2));
    digitalWrite(motorPin4, bitRead(lookup[out], 3));
}
```

Ilustración 107-Función para determinar que bobina ha de activarse

Para diferenciar el sentido de giro, se han creado dos funciones que recorren la variable "lookup" en ambos sentidos mediante de un bucle for.

```
void anticlockwise()
{
    for(int i = 0; i < 8; i++)
    {
        setOutput(i);
        delayMicroseconds(motorSpeed);
    }
}

void clockwise()
{
    for(int i = 7; i >= 0; i--)
    {

        setOutput(i);
        delayMicroseconds(motorSpeed);
    }
}
```

Ilustración 108-Funciones para mover el motor en ambos sentidos

- La función "anticlockwise" sirve para girar el motor en el sentido contrario a las agujas del reloj, incrementando una variable "i" dentro de la función "setOutput".
- La función "clockwise" sirve para girar el motor en el sentido a las agujas del reloj, decrementando una variable "i" dentro de la función "setOutput".

Los pasos son contados mediante una variable "count" que será incrementara o decrementará según el sentido de giro y la orden que sea enviada desde la web.

```
static int count=0;
```

Ilustración 109- Variable para contar pasos

Como se desconoce el número de pasos que corresponde a las diferentes posiciones programadas de la ventana, debido a que dependiendo de la longitud de la misma varía el número de pasos, se han tomado unos valores para simular el movimiento real.

- En el 0% de apertura de la persiana la variable “count” será igual a 0.
- En el 25% de apertura de la persiana la variable “count” será igual a 500.
- En el 50% de apertura de la persiana la variable “count” será igual a 1000.
- En el 75% de apertura de la persiana la variable “count” será igual a 1500.
- En el 100% de apertura de la persiana la variable “count” será igual a 2000.

Desde la web se puede dar la orden de accionamiento de la persiana mediante la interfaz creada para esta aplicación.

Para que al microcontrolador reciba las órdenes dadas desde la web, esta se ha programado de la siguiente forma:

Al pulsar una de las opciones de la web, la barra de direcciones cambia tomando el valor de la opción pulsada.

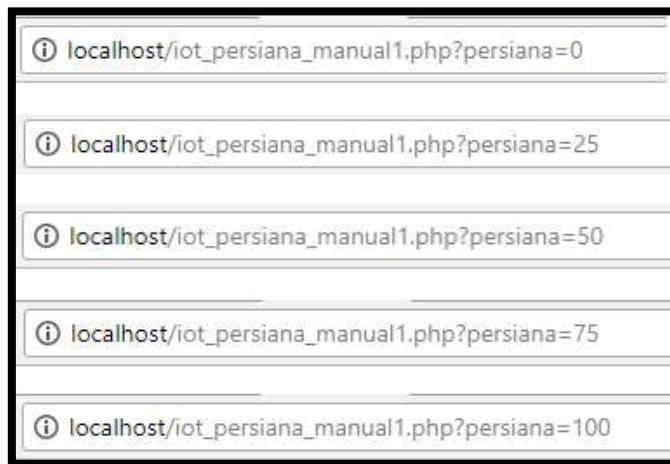


Ilustración 110- Barra de direcciones Web con valor del botón pulsado

Dentro de un formulario PHP, este valor es guardado por una variable “\$persiana” que ha sido creada para leer el valor de “persiana” dentro de la barra de direcciones.

DATOS DE LA MEMORIA

```
$persiana = $_GET["persiana"];
```

Ilustración 111- Variable para tomar valor seleccionado

Según el valor que tome la variable, se cumple una de las condiciones “if” programadas.

```
if( $persiana == "0") {
    $file = fopen("per.json", "w") or die("can't open file");
    fwrite($file, '{"per": "0"}');
    fclose($file);
}
else if ( $persiana == "25") {
    $file = fopen("per.json", "w") or die("can't open file");
    fwrite($file, '{"per": "25"}');

    fclose($file);
}
else if ( $persiana == "50") {
    $file = fopen("per.json", "w") or die("can't open file");
    fwrite($file, '{"per": "50"}');

    fclose($file);
}
else if ( $persiana == "75") {
    $file = fopen("per.json", "w") or die("can't open file");
    fwrite($file, '{"per": "75"}');

    fclose($file);
}
else if ( $persiana == "100") {
    $file = fopen("per.json", "w") or die("can't open file");
    fwrite($file, '{"per": "100"}');

    fclose($file);
}
```

Ilustración 112-Condiciones programadas para cada valor

Esto hace que se guarde, el valor que contiene la variable “\$persiana”, dentro de un archivo JSON para leerlo desde el microcontrolador.

```
{"per": "100"}
```

Ilustración 113- Valor guardado en archivo JSON

Para leer el valor en el microcontrolador, este ha sido configurado tal y como se explica en el punto 5.4.1.

La variable “result” en esta aplicación contiene el valor de la posición de la persiana.

```
String result = line.substring(1);
```

Ilustración 114-Variable del microcontrolador con valor archivo JSON

Para extraer el valor numérico que contiene el string “result”, se ha usado un substring.

DATOS DE LA MEMORIA

Cuando este substring sea igual a una de las condiciones if programadas, la persiana se moverá a una de las posiciones que contiene el sistema.

```
if(result.substring(9, 11)=="25")
{
    while (count>=0 && count<500)
    {
        count++ ;
        clockwise();
        digitalWrite(Subir,HIGH);
        Serial.println(count);
        delay(1);
    }
    while ( count>500)
    {
        count-- ;
        anticlockwise();
        digitalWrite(Bajar,HIGH);
        Serial.println(count);
        delay(1);
    }
}
```

Ilustración 115- Ejemplo de una de las cinco posiciones

Cuando el motor recibe una orden desde la web, puede ocurrir que el motor se encuentre por encima o por debajo de la posición ordenada. Es por esto por lo que, se han programado dos condiciones para cada posición. Al ser un motor paso a paso, la posición del motor es conocida gracias a los pasos recorridos por el motor.

- Cuando el motor se encuentre por encima de la posición ordenada, este bajara.
- Por el contrario, cuando el motor se encuentre por debajo de la posición ordenada, este subirá.

DATOS DE LA MEMORIA

5.4.10. Sistema de alarma

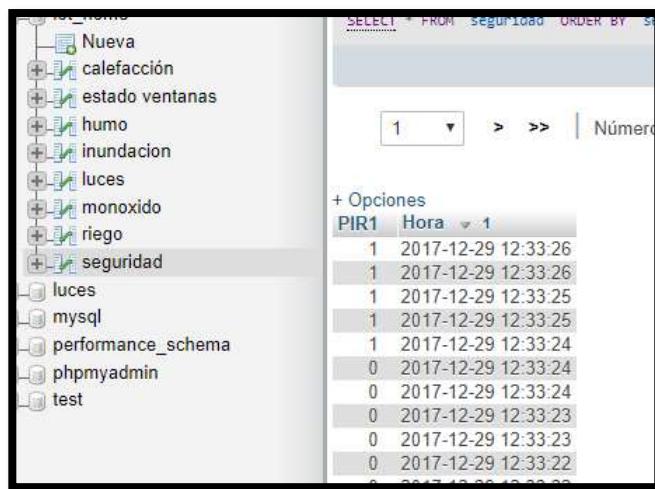
5.4.10.1. Subsistema de detección de presencia

Para realizar esta aplicación el sensor PIR se ha colocado en una entrada digital del microcontrolador para recibir los datos que este proporcione.

```
pinMode(PIR1_Input, INPUT);
```

Ilustración 116-Declaración del PIR como entrada en el microcontrolador

Cuando el sensor detecte presencia, el valor que entrega este sensor es “1”, por el contrario, cuando no detecta presencia entrega un “0”. Estos datos son enviados a una base de datos para poder usarlos desde la web a la que tiene acceso el usuario. Para configurar esta base de datos, se ha usado el método explicado en el punto 5.4.2.



The screenshot shows the phpMyAdmin interface. On the left, there's a tree view of databases and tables, with 'seguridad' selected. On the right, a query results table is displayed for the SQL command 'SELECT * FROM seguridad ORDER BY id'. The table has two columns: 'PIR1' and 'Hora'. The data shows alternating rows of 1 and 0, representing presence and absence detections over time.

PIR1	Hora
1	2017-12-29 12:33:26
1	2017-12-29 12:33:26
1	2017-12-29 12:33:25
1	2017-12-29 12:33:25
1	2017-12-29 12:33:24
0	2017-12-29 12:33:24
0	2017-12-29 12:33:24
0	2017-12-29 12:33:23
0	2017-12-29 12:33:23
0	2017-12-29 12:33:22
0	2017-12-29 12:33:22

Ilustración 117- Base de datos Seguridad

Para mostrar estos datos en la web, esta ha sido programada mediante código PHP y se usado el método que ofrece la base de datos para extraer los datos que contiene. La variable "\$query" está configurada para hacer una petición de datos de la tabla "seguridad". La variable "\$result" toma los datos de la tabla realizada para esta aplicación, haciendo uso de la petición anterior e indicándole los datos de conexión como se explicaba en el punto 5.4.2.

```
$query = "SELECT * FROM `seguridad` ORDER BY `Hora` DESC";
$result = mysqli_query($conexion, $query);
```

Ilustración 118- Petición de los datos de la base de datos

Estos datos son guardados en un array para posteriormente usarlos en ciertas condiciones programadas.

```
$row = mysqli_fetch_array($result, MYSQLI_BOTH);
```

Ilustración 119- Array de datos

Una de las condiciones para que se muestren los datos del sensor, es que se active el modo fuera de casa. Cada modo tiene un botón asociado, y este a su vez, una referencia.

```
<h1> MODOS</h1>
      <a href="?modo=on">MODO FUERA DE CASA </a>
      <hr />
      <a href="?modo=off">MODO EN CASA </a>
      <hr />
<hr/>
```

Ilustración 120-Modos seleccionables

Cuando se pulsa uno de los modos, la dirección contenida en la barra de direcciones web se completa con el valor de la referencia que tiene asociada.

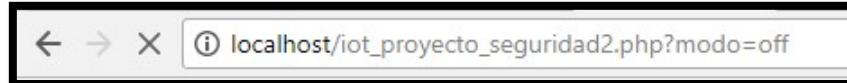


Ilustración 121- Barra de direcciones al pulsar el botón "modo en casa".

Este valor, es tomado por una variable que tiene por objeto conseguir el valor de "modo".

```
$modo = $_GET["modo"];
```

Ilustración 122-variable con valor del botón pulsado

Una vez explicados de donde llegan todos los datos, se van a explicar las condiciones programadas.

Para no perder el valor de "\$modo", debido a que se ha programado un refresco de dos segundos que borra los datos del valor seleccionados, el valor de la variable "\$modo" es guardado en un archivo JSON".

DATOS DE LA MEMORIA

La página lee el contenido del archivo JSON que es independiente al refresco de la página. El archivo JSON se llama "PIRMODO.json".

```
if($modo=="on")
{
    $file = fopen("PIRMODO.json", "w") or die("can't open file");
    fwrite($file, '{"pirmodo": "on"}');

    fclose($file);
}

if($modo=="off")
{
    $file = fopen("PIRMODO.json", "w") or die("can't open file");
    fwrite($file, '{"pirmodo": "off"}');

    fclose($file);

    $file = fopen("PIR1.json", "w") or die("can't open file");
    fwrite($file, '{"pir1": "0"}');

    fclose($file);
```

Ilustración 123-Método para guardar datos en un archivo JSON

Para mostrar el mensaje de incidencia, se ha usado el valor del archivo "PIRMODO.json" y los valores del sensor contenidos en el array explicado antes. Cuando el sensor esta detectado un intruso y el modo fuera de casa esta activado, se guarda el valor "1" para reflejar que existe una incidencia dentro de un segundo archivo PIR1.json. En caso de no detectar presencia, el valor contenido en el archivo será "0".

```
if( $row["PIR1"] == 1 && $PIRMODO['pirmodo']=="on") {

    $file = fopen("PIR1.json", "w") or die("can't open file");
    fwrite($file, '{"pir1": "1"}');

    fclose($file);

}
```

Ilustración 124- Archivo JSON para registrar una incidencia.

De esta forma, aunque el valor del sensor cambie, debido a que en algún momento dejara de detectar presencia, quedara constancia de que ha habido una incidencia.

Para mostrar los mensajes de "SIN INCIDENCIAS" o "INCIDENCIA" y "VER EN CORREO ELECTRONICO HORA DE INCIDENCIA", se han usado los valores del archivo PIR1.JSON y del archivo "PIRMODO.json".

DATOS DE LA MEMORIA

Cuando este activado el modo fuera de casa, pero no se haya detectado un intruso, aparecerá el mensaje "SIN INCIDENCIAS", debido a que el archivo "\$PIR1.json" no contendrá el valor "1".

```
if( $PIR1['pir1'] == '0' && $PIRMODO['pirmodo']=="on")
{
printf ("SIN INCIDENCIAS");
}
```

Ilustración 125-Condiciones cuando no se detecta intruso y el modo fuera de casa esta activado.

Cuando este activado el modo fuera de casa y se detecte un intruso, el archivo "\$PIR1.json" contendrá el valor "1" y aparecerán los mensajes "INCIDENCIA" y "VER EN CORREO ELECTRONICO HORA DE INCIDENCIA".

```
if( $PIR1['pir1'] == '1' && $PIRMODO['pirmodo']=="on")
{
printf ("VER EN CORREO ELECTRONICO HORA DE INCIDENCIA");
}

echo nl2br("\n ");

if( $PIR1['pir1'] == '1' && $PIRMODO['pirmodo']=="on') {
printf ("INCIDENCIA");
}
```

Ilustración 126-Condiciones cuando se detecta intruso y el modo fuera de casa esta activado.

Para que el usuario obtenga rápidamente el aviso de una incidencia, en caso de intrusión, se enviara un mensaje de correo electrónico aviso de este problema.

```
if( $row["PIR2"] == 1 && $PIRMODO['pirmodo']=="on") {

$to = "orteo@hotmail.com";
$subject = "alarma ventana";
$message = "OBJETO DETECTADO";

mail($to, $subject, $message);

echo ("mensaje enviado");
Sleep(60);

}
```

Ilustración 127- Envío de correo electrónico

En el correo electrónico aparece el mensaje que se muestra a continuación informando del problema.

aorcs90@gmail.com

alarma ventana • OBJETO DETECTADO

Ilustración 128-Correo electrónico.

DATOS DE LA MEMORIA

5.4.10.2. Subsistema de estado de puertas y ventanas

El montaje del sistema es el mostrado en la imagen. Una de las imágenes corresponde al estado sin imán y la otra con imán.

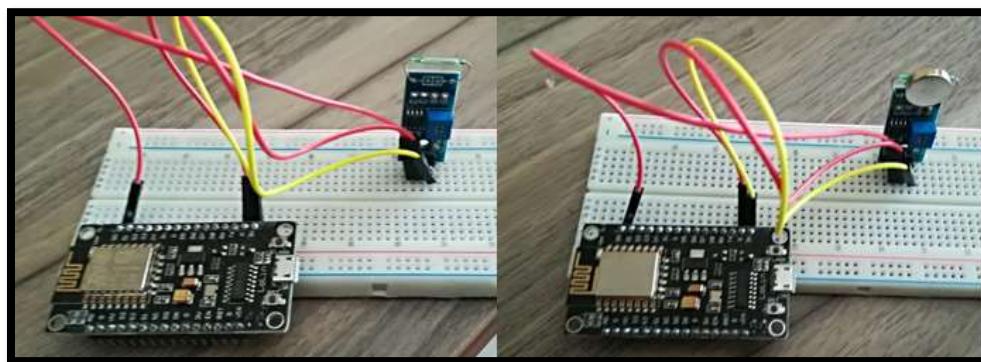


Ilustración 129- Montaje de aplicación

El Reed Switch es conectado al microcontrolador mediante sus tres pines. Estos son Vcc, GND y D0. Cuando entra en contacto con un campo magnético, manda una "0" digital al microcontrolador. En ausencia de campo magnético, manda un "1". Estos datos son enviados a una base de datos e interpretados por una página web.

Desde el microcontrolador se ha creado una función para configurar el sensor. La configuración de este sensor se compone, de la declaración de un pin digital como una entrada para recibir los datos del sensor.

```
int estadoven(){
    const int pinSensor = 16;
    pinMode(pinSensor, INPUT_PULLUP);
    int value = digitalRead(pinSensor);

    return value;

}
```

Ilustración 130- Función del sensor Reed Switch

Los datos de la función son mandados a la base de datos haciendo uso de la configuración explicada en el punto 5.4.2.

La estructura de la base de datos una vez configurada es la siguiente:

The screenshot shows the phpMyAdmin interface. On the left, the database structure is displayed under the 'iot_home' database, including tables like 'Nueva', 'frigorifico', 'information_schema', 'iot_home', 'Nueva', 'calefaccion', 'humo', 'inundacion', 'luz', 'monoxido', and 'riego'. The 'estado_ventanas' table is selected. The main area shows the table structure with columns: puerta, ventana1, ventana2, ventana3, ventana4, and tiempo. Below the table, a SQL query is visible: 'SELECT * FROM `estado_ventanas` ORDER BY `estado_ventanas`.`tiempo` DESC'. The results show several rows of data, each representing a state reading at a specific timestamp.

puerta	ventana1	ventana2	ventana3	ventana4	tiempo
1	1	1	1	1	2018-01-21 12:06:25
1	1	1	1	1	2018-01-21 12:06:18
1	1	1	1	1	2018-01-21 12:06:11
1	1	1	1	1	2018-01-21 12:06:05
1	1	1	1	1	2018-01-21 12:05:58
1	1	1	1	1	2018-01-21 12:05:51
1	1	1	1	1	2018-01-21 12:05:45

Ilustración 131- Base de datos para la aplicación de estado ventanas

En la base de dato se muestran los datos del Reed Switch, así como el momento en el que se introdujo ese dato. Se puede observar que los datos se introducen en la tabla cada cinco segundos, debido a que se ha programado el microcontrolador para entrar en modo reposo cada cinco segundos después de enviar un dato.

Para llevar a cabo la web a la que el usuario tiene acceso, se ha usado código PHP. Mediante un condicional "if" se compara el valor que está proporcionando el sensor con el valor "1" o "0" y en función al condicional que se cumpla aparece un mensaje de puerta abierta o puerta cerrada con su correspondiente imagen.

```
if($row["puerta"]==1)
{
    printf ("La puerta de la calle esta [abierta] <br>");
    print "<img src=\\"puertaabierta.jpg\\" >";
}

if($row["puerta"]==0)
{
    printf ("La puerta de la calle esta [cerrada] <br>");
    print "<img src=\\"puertacerrada.jpg\\" >";
}
```

Ilustración 132- Condiciones para mostrar imagen puerta abierta o puerta cerrada

Además, se enviará un mensaje de correo electrónico al usuario cuando el modo fuera de casa, explicado en el subsistema anterior, sea accionado y una puerta o ventana sea abierta.

DATOS DE LA MEMORIA

```
if( $row["PIR2"] == 1 && $PIRMODO['pirmodo']=="on") {  
  
    $to = "orte90@hotmail.com";  
    $subject = "alarma ventana";  
    $message = "OBJETO DETECTADO";  
  
    mail($to, $subject, $message);  
  
    echo ("mensaje enviado");  
    Sleep(60);  
  
}
```

Ilustración 133-Envío de mensaje de alarma

5.4.10.3. Videovigilancia

Para poder visualizar el hogar desde la cámara, se ha hecho uso de la dirección IP que toma la cámara, y del uso del protocolo para verla desde el navegador.

```

```

Ilustración 134-Cámara IP

5.4.11. BIBLIOGRAFÍA

4. Lectura de sensores. (2015). Recuperado 9 de diciembre de 2017, a partir de <https://es.libre.university/lesson/HJT0PhLRx/4.%20Lectura%20de%20sensores>

Características DHT22 - Sensor de temperatura y humedad. (2014). Recuperado 9 de diciembre de 2017, a partir de <http://rduinostar.com/documentacion/datasheets/dht22-caracteristicas-am2302/>

Cinco razones para comprar un termostato inteligente. (2015). Recuperado 29 de marzo de 2017, a partir de <https://domboo.es/por-que-comprar-un-termostato-inteligente/>

¿Cómo funciona un detector de presencia? - Twenergy. (2013, julio 29). Recuperado 29 de marzo de 2017, a partir de <http://twenergy.com/a/como-funciona-un-detector-de-presencia-912>

Detector de movimiento con Arduino y sensor PIR. (2015). Recuperado 11 de diciembre de 2017, a partir de <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Detectores de monóxido de carbono. (2007). Recuperado 28 de marzo de 2017, a partir de http://www.supercontrols.com.ar/productos/detector_monoxido.html

DHT11 Sensor – Tienda y Tutoriales Arduino. (2016). Recuperado 26 de diciembre de 2017, a partir de <https://www.prometec.net/producto/dht11-sensor-temperatura-y-humedad/>

Dilmen, N. (2014). *English: LM35 temperature sensor, semiconductor thermometer.* Measures 0-100 degrees C. Recuperado a partir de https://commons.wikimedia.org/wiki/File:LM35_temperature_sensor_semiconductor_thermometer_1480374_5_6_HDR_enhancer.jpg

DATOS DE LA MEMORIA

DS18B20 Sensor de Temperatura. (2017). Recuperado 26 de diciembre de 2017, a partir de <http://www.geekbotelectronics.com/producto/ds18b20-sensor-de-temperatura/>

DS18B20 sensor de temperatura para líquidos con Arduino. (2017). Recuperado 6 de diciembre de 2017, a partir de <https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>

Elige motor adecuado al peso y eje de la persiana. (2007). Recuperado 31 de diciembre de 2017, a partir de <http://www.motorizatuspersianas.com/1motor.html>

eodos. (2015). PFG – Sensor de humedad FC-28. Recuperado 26 de diciembre de 2017, a partir de <https://eodos.net/proyectos/sensor-de-humedad>

Esp8266 Modulo Wifi | Blog de PatagoniaTec Electronica. (2015). Recuperado 29 de marzo de 2017, a partir de <http://saber.patagoniatec.com/esp8266-modulo-wifi/>

LDR-fotoresistencia. (2014). Recuperado 28 de marzo de 2017, a partir de <https://electronica-electronics.com/info/LDR-fotoresistencia.html>

LED Fuente de alimentación 50W 5V 10A ; MeanWell, RS-50-5: Amazon.es: Bricolaje y herramientas. (2015). Recuperado 31 de diciembre de 2017, a partir de <https://www.amazon.es/LED-Fuente-alimentaci%C3%B3n-MeanWell-RS-50-5/dp/B00MWQDH00>

LoRaWAN – Long Range Wide Area Network. (2016). Recuperado 27 de marzo de 2017, a partir de <https://blogs.deusto.es/aplicaciones-tic/lorawan-long-range-wide-area-network/>

LoRaWAN™ | RS Components. (2016). Recuperado 27 de marzo de 2017, a partir de <http://es.rs-online.com/web/generalDisplay.html?id=i/lora>

Medir temperatura y humedad con Arduino y sensor DHT11-DHT22. (2016). Recuperado 26 de diciembre de 2017, a partir de <https://www.luisllamas.es/arduino-dht11-dht22/>

DATOS DE LA MEMORIA

pentadom. (2014, marzo 19). La domótica inalámbrica con Z-Wave, una realidad para viviendas existentes – PENTADOM Edificios Inteligentes. Recuperado 27 de marzo de 2017, a partir de <http://pentadom.com/2014/03/la-domotica-inalambrica-con-z-wave-una-realidad-para-viviendas-ya-construidas/>

Pila litio industria CR AA 3V 2Ah. (2016). Recuperado 31 de diciembre de 2017, a partir de <http://www.all-batteries.es/pila-litio-industria-cr-aa-3v-2ah-pcl8805.html>

¿Qué es Arduino? ¿Qué modelo comprar? (2013). Recuperado 29 de marzo de 2017, a partir de <https://www.luisllamas.es/que-es-arduino-que-modelo-comprar/>

¿Que es un Reed Switch? (2010). Recuperado 28 de marzo de 2017, a partir de <http://ayudaelectronica.com/que-es-un-reed-switch/>

Qué son los sensores de gas? – 330ohms. (2016). Recuperado 29 de marzo de 2017, a partir de <https://www.330ohms.com/blogs/blog/sensores-de-gas-como-funcionan>

RASPBERRY PI – Historia de la Informática. (2013). Recuperado 29 de marzo de 2017, a partir de <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

Raspberry Pi 3, 2, 1 y Zero en español. (2014). Recuperado 29 de marzo de 2017, a partir de <https://www.raspberryshop.es/>

¿Sabes qué es el estándar Wi-Fi y cuantos tipos hay? (2014, noviembre 19). Recuperado 28 de marzo de 2017, a partir de <http://blogthinkbig.com/estandar-wi-fi/>

Sensor de temperatura, escoge el mejor para tus proyectos con Arduino. (2016). Recuperado 6 de diciembre de 2017, a partir de <https://programarfácil.com/podcast/82-escoger-mejor-sensor-temperatura-arduino/>

Sensor DHT11 (Humedad y Temperatura) con Arduino |. (2012). Recuperado 9 de diciembre de 2017, a partir de <https://tallerarduino.com/2012/12/24/sensor-dht11-humedad-y-temperatura-con-arduino/>

DATOS DE LA MEMORIA

Sensores: Tipos y Funcionalidades - DomoPrac - Domotica practica paso a paso. (2015).

Recuperado 29 de marzo de 2017, a partir de <http://www.domoprac.com/hardware-y-productos-domoticos/sensores-tipos-y-funcionalidades.html>

SHT75 - SENSIRION - Sensor de Humedad, 1.8 %, 3.3 V, 0% a 100% de Humedad Relativa, SIP, 4 Pines, 8 s | Newark element14 México. (2007). Recuperado 9 de diciembre de 2017, a partir de <http://mexico.newark.com/sensirion/sht75/humidity-temperature-sensor/dp/18M2988>

SHT75 Datasheet PDF - Sensirion - Pinout,Circuit - FindIC.us. (2007). Recuperado 26 de diciembre de 2017, a partir de <http://www.findic.us/sht75-datasheet-pdf-en-4zp7ERr5e.html>

SLA-05VDC-SL-C_Datasheet.pdf. (2015). Recuperado a partir de https://arduino-info.wikispaces.com/file/view/SLA-05VDC-SL-C_Datasheet.pdf/577719703/SLA-05VDC-SL-C_Datasheet.pdf

SODIAL(R)24W Conductor Fuente de alimentacion Transformador DC 12V 2A por Banda Luz LED Lampara: Amazon.es: Bricolaje y herramientas. (2016). Recuperado 1 de enero de 2018, a partir de https://www.amazon.es/SODIAL-Conductor-alimentacion-Transformador-Lampara/dp/B016U4R6I8/ref=sr_1_9?ie=UTF8&qid=1514808728&sr=8-9&keywords=fuente+alimentaci%C3%B3n+12v+2A

Tipología de los detectores de incendios de incendios y gases. (2012). Recuperado 28 de marzo de 2017, a partir de <http://www.prefire.es/proteccion-contraincendios/deteccion-incendio-gases-4.php>

Vega, R. (2014, septiembre 3). Z-Wave: un protocolo inalámbrico para la domótica | Ricardo Vega. Recuperado 27 de marzo de 2017, a partir de <https://ricveal.com/blog/z-wave/>

DATOS DE LA MEMORIA

VideoVigilancia. (2015). Recuperado 29 de marzo de 2017, a partir de <http://www.hsoft.es/index.php/Servicios/sistemas-de-videovigilancia-alarma-y-domotica.html>

ZigBee, la tecnología barata para comunicación inalámbrica. (2015). Recuperado 27 de marzo de 2017, a partir de <https://www.elandroidelibre.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>



Relación de documentos

- (_) Memoria 63 páginas
(X) Anexos 97 páginas

La Almunia, a 06 de febrero de 2018

Firmado: Adrián Orte Ruiz