



**UNIVERSIDAD POLITÉCNICA SALESIANA**

SEDE GUAYAQUIL

**CARRERA DE INGENIERÍA ELECTRÓNICA CON MENCIÓN EN  
TELECOMUNICACIONES.**

**PROYECTO DE TITULACIÓN**

Previa la obtención del Título de:

**INGENIERO ELECTRÓNICO CON MENCIÓN EN  
TELECOMUNICACIONES.**

**TEMA**

“DISEÑO E IMPLEMENTACIÓN DE UNA CASA DOMÓTICA PARA PERROS  
DOMÉSTICOS USANDO TECNOLOGÍA RASPBERRY PI Y ARDUINO CON  
CONTROL A TRAVÉS DE INTERFAZ WEB.”

**AUTORES**

Bastidas Martínez Leonel Andrés

Rodríguez Sares Kerly Andrea

DIRECTOR: MET. Gino Alvarado

GUAYAQUIL

2018

## **CERTIFICADOS DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

La responsabilidad del contenido de este Proyecto de Titulación, Los conceptos plasmados en este texto, ya sea en el desarrollo, análisis realizados y las conclusiones del presente trabajo son de exclusiva responsabilidad: Bastidas Martínez Leonel Andrés y Rodríguez Sares Kerly Andrea, y el patrimonio intelectual del mismo a la UNIVERSIDAD POLITÉCNICA SALESIANA”.

Guayaquil, marzo 15 2018

Firma) \_\_\_\_\_

Autor: Kerly Rodríguez Sares

Cédula: 0930410048

Firma) \_\_\_\_\_

Autor: Leonel Bastidas Martínez

Cédula: 0930061197

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO  
DE TITULACIÓN A LA UPS**

Yo, **BASTIDAS MARTÍNEZ LEONEL ANDRÉS**, con documento de identificación N° **0930061197**, manifiesto mi voluntad y cedo a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado “**DISEÑO E IMPLEMENTACIÓN DE UNA CASA DOMÓTICA PARA PERROS DOMÉSTICOS USANDO TECNOLOGÍA RASPBERRY PI Y ARDUINO CON CONTROL A TRAVÉS DE INTERFAZ WEB.**” mismo que ha sido desarrollado para optar por el título de **INGENIERO ELECTRÓNICO CON MENCIÓN EN TELECOMUNICACIONES** , en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscrito este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, marzo 15 2018

f) \_\_\_\_\_

Autor: Leonel Bastidas Martínez

Cédula: 0930061197

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO  
DE TITULACIÓN A LA UPS**

Yo, **RODRÍGUEZ SARES KERLY ANDREA**, con documento de identificación N° **0930410048**, manifiesto mi voluntad y cedo a la **UNIVERSIDAD POLITÉCNICA SALESIANA** la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de grado titulado “**DISEÑO E IMPLEMENTACIÓN DE UNA CASA DOMÓTICA PARA PERROS DOMÉSTICOS USANDO TECNOLOGÍA RASPBERRY PI Y ARDUINO CON CONTROL A TRAVÉS DE INTERFAZ WEB.**” mismo que ha sido desarrollado para optar por el título de **INGENIERO ELECTRÓNICO CON MENCIÓN EN TELECOMUNICACIONES**, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscrito este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, marzo 15 2018

f) \_\_\_\_\_

Autor: Kerly Rodríguez Sares

Cédula: 0930061197

**CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN  
SUSCRITO POR EL TUTOR**

Yo, **GINO ADRIÁN ALVARADO MEJÍA**, director del proyecto de Titulación denominado “**DISEÑO E IMPLEMENTACIÓN DE UNA CASA DOMÓTICA PARA PERROS DOMÉSTICOS USANDO TECNOLOGÍA RASPBERRY PI Y ARDUINO CON CONTROL A TRAVÉS DE INTERFAZ WEB.**” realizado por los estudiantes: Kerly Rodríguez Sares y Leonel Bastidas Martínez, certifico que ha sido orientado y revisado durante su desarrollo, por cuanto se aprueba la presentación de este ante las autoridades pertinentes.

Guayaquil, 2018

f) \_\_\_\_\_

MET. Gino Alvarado Mejía.

## **DEDICATORIA**

Dedico este proyecto de titulación en primer lugar a Dios por darme salud y permitirme llegar a culminar esta etapa importante en mi formación profesional, a mis padres por ser el pilar fundamental, que con paciencia y mucho amor supieron guiarme y orientarme toda mi etapa estudiantil y de vida, forjándome principios, enseñándome a no desfallecer ni rendirme ante ninguna circunstancia y siempre perseverar gracias a sus sabios consejos, a mi hermano que siempre ha estado junto a mí apoyándome y al amor de mi querida hija Juliet por ser mi mayor inspiración, a mis docentes a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza, a mis queridos compañeros que con el pasar de los años se convirtieron en grandes amigos y a mi amiga fiel e incondicional Brandy, por ser motivo de inspiración para realizar este proyecto de titulación.

### **Kerly Rodríguez Sares**

Dedico este proyecto de titulación a Dios, a mis padres, tías y hermanas. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mi familia quiénes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mis capacidades. Y a mí leal amigo y compañero Lucky, quién fue partícipe del inicio de este proyecto.

### **Leonel Bastidas Martínez**

## **AGRADECIMIENTOS**

Este proyecto de tesis es el resultado del esfuerzo en conjunto por varios años. Por esto agradecemos en primer lugar a Dios por permitirnos culminar con éxito nuestra carrera profesional, a nuestros padres por depositar toda su confianza en nosotros, a nuestro tutor Met. Gino Alvarado, por su aporte con sus conocimientos y a todas las personas que, de alguna forma, son parte de su culminación. Nuestros sinceros agradecimientos.

**Kerly Rodríguez Sares y Leonel Bastidas Martínez**

## ÍNDICE DE CONTENIDO

<b>CERTIFICADOS DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN</b> .....	ii
<b>CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UPS</b> .....	iii
<b>CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UPS</b> .....	iv
<b>CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN SUSCRITO POR EL TUTOR</b> .....	v
<b>DEDICATORIA</b> .....	vi
<b>AGRADECIMIENTOS</b> .....	vii
<b>ÍNDICE DE CONTENIDO</b> .....	viii
<b>ÍNDICE DE IMÁGENES</b> .....	xvi
<b>RESUMEN</b> .....	24
<b>ABSTRACT</b> .....	25
<b>INTRODUCCIÓN</b> .....	26
1.1 ANTECEDENTES.....	27
1.2 IMPORTANCIA Y ALCANCE.....	27
1.3 DELIMITACIÓN DEL PROBLEMA.....	28
1.3.1 ESPACIAL.....	28
1.3.2 TEMPORAL.....	29
1.3.3 ACADÉMICA.....	30
1.4 OBJETIVOS.....	30
1.4.1 OBJETIVO GENERAL.....	30
1.4.2 OBJETIVOS ESPECÍFICOS.....	30
1.5 MARCO METODOLÓGICO.....	31
1.5.1 MÉTODO EXPERIMENTAL.....	31



1.5.2	MÉTODO EXPERIMENTAL.....	31
2.1.	¿QUÉ ES ARDUINO?.....	32
2.1.1	ARDUINO MEGA 2560.....	32
2.1.2	ESPECIFICACIONES TÉCNICAS ARDUINO MEGA 2560.....	33
2.2.	¿QUÉ ES RASPBERRY PI?.....	34
2.2.1	DISEÑO DE RASPBERRY PI.....	34
2.2.2	MODELOS DE RASPBERRY.....	35
2.2.2.1	RASPBERRY PI 3 MODELO B.....	35
2.2.3	PUERTOS GPIO.....	36
2.3.	PYTHON.....	36
2.4.	SOFTWARE DE ACCESO REMOTO.....	37
2.4.1	VNC.....	37
2.4.2	SSH.....	37
2.4.3	SERVIDOR WEB.....	38
2.4.3.1	SERVIDOR APACHE.....	38
2.4.3.2	NGINX.....	38
2.5.	INTERFAZ WEB.....	39
2.5.1	COMPONENTES DE UNA INTERFAZ WEB.....	39
2.5.1.1	ELEMENTOS DE IDENTIFICACIÓN.....	39
2.5.1.2	ELEMENTOS DE NAVEGACIÓN.....	39
2.5.1.3	ELEMENTOS DE CONTENIDOS.....	39
2.5.1.4	ELEMENTOS DE INTERACCIÓN.....	39
2.5.2	LENGUAJE DE INTERFAZ WEB.....	40
2.5.2.1	CSS.....	40
2.5.2.2	JQUERY.....	40
2.5.2.3	PHP.....	41
2.6.	PUSHETTA.....	42

2.7.	MOTOR ELÉCTRICO.....	43
2.7.1	USOS .....	43
2.7.1.1	CAMBIO DE SENTIDO DE GIRO .....	43
2.7.2	CAJA REDUCTORA .....	44
2.8.	PUERTA AUTOMÁTICA.....	44
2.8.1	TIPOS DE PUERTAS .....	44
2.9.	LÁMPARA LED.....	44
2.9.1	CARACTERÍSTICAS DE ILUMINACIÓN CON LÁMPARAS LEDS. ....	45
2.9.2	TIPOS DE LÁMPARAS LEDS .....	46
2.10.	VENTILADOR.....	46
2.10.1	TIPOS DE VENTILADORES .....	47
2.11.	PLYWOOD .....	47
2.11.1	ELABORACIÓN .....	47
2.11.2	PLYWOOD MARINO.....	48
2.11.2.1	USOS .....	48
2.11.2.2	CARACTERÍSTICAS .....	48
2.11.2.3	TIPOS DE PLYWOOD MARINO .....	49
2.12.	SENSORES .....	49
2.12.1	SENSOR DE TEMPERATURA .....	49
2.12.1.1	LM35.....	50
	ESPECIFICACIONES LM35 .....	50
2.12.2	SENSOR DE LUZ .....	51
2.12.2.1	MÓDULO LDR .....	51
	ESPECIFICACIONES TÉCNICAS .....	52
2.12.3	SENSOR DE FINAL DE CARRERA O DE CONTACTO .....	52
2.12.4	SENSOR PIR .....	52
2.12.4.1	MÓDULO HC-SR501 .....	53

CARACTERÍSTICAS .....	53
2.12.4.1.1 ESQUEMA ELÉCTRICO .....	54
2.12.5 PUENTE H .....	54
2.12.5.1 DRIVER L298 .....	54
2.12.5.1.1 CARACTERÍSTICAS .....	55
2.12.5.1.2 ESQUEMA DE MONTAJE .....	55
2.12.5.1.3 ESQUEMA DE CONEXIÓN .....	56
2.12.6 SENSOR ULTRASÓNICO .....	57
2.12.6.1 FUNCIONAMIENTO .....	57
2.12.6.2 SENSOR DE DISTANCIA DE ULTRASONIDO HC-SR04.....	57
2.12.6.2.1 CARACTERÍSTICAS .....	58
2.13. RELÉ .....	58
2.13.1 MÓDULO RELÉ DE 8 CANALES OPTOACOPLADOR .....	58
2.14. GRABADOR DE VOZ .....	59
2.14.1 MÓDULO GRABACIÓN DE VOZ ISD1820C.....	59
2.14.1.1 CARACTERÍSTICAS .....	59
2.15. DOSIFICADORES.....	60
2.15.1 TIPOS DE DOSIFICADORES.....	60
2.16. TOLVA.....	60
2.17. CASA PARA PERROS.....	61
2.17.1 CARACTERÍSTICAS IDÓNEAS PARA ELECCIÓN DE CASA PARA PERROS.....	62
2.17.2 TIPOS DE CASAS PARA PERROS.....	62
2.18. CUIDADO DE LOS PERROS.....	64
2.18.1 RACIONES DIARIAS DE COMIDA PARA PERRO .....	66
2.18.1.1 CANTIDAD DE COMIDA RECOMENDADA PARA UN PERRO ADULTO .....	66

2.18.1.2 CANTIDAD DE COMIDA RECOMENDADA PARA UN CACHORRO .....	68
2.19 ACERO INOXIDABLE .....	69
2.19.1 TIPOS DE ACERO INOXIDABLE .....	69
2.20 UPS .....	69
2.21 ROUTER .....	70
2.21.1 FUNCIONAMIENTO .....	70
2.22 SOFTWARE RPI – CAM – WEB - INTERFACE .....	71
2.23 PI NOIR CAMERA V2 .....	71
3.1 DISEÑO Y ENSAMBLAJE DE LA ESTRUCTURA DE LA CASA PARA MASCOTAS .....	72
3.1.1 MATERIALES Y METODOLOGÍA .....	72
3.1.2 DIMENSIONALIDAD DE ELEMENTOS .....	73
3.1.3 ESTRUCTURA DE LA CASA .....	73
3.1.4 DISEÑO ESTRUCTURAL DE LA CASA .....	74
3.1.5 CONSTRUCCIÓN Y ENSAMBLAJE DE LA CASA .....	76
3.2 HABILITACIÓN Y CONFIGURACIÓN INICIAL DE RASPBERRY PI ...	80
3.2.1 INSTALACIÓN DEL SISTEMA OPERATIVO .....	80
3.2.2 CONFIGURACIONES INICIALES .....	82
3.3 DESARROLLO DE LA INTERFAZ WEB .....	88
3.3.1 CONSIDERACIONES DEL DISEÑO .....	88
3.3.2 INSTALACIÓN DE SERVIDOR APACHE Y PHP .....	89
3.3.3 DISEÑO DEL ENTORNO GRÁFICO EN LA INTERFAZ WEB .....	91
3.3.4 ESTRUCTURA DEL DIRECTORIO WEB DEL SISTEMA .....	92
3.3.5 SEGURIDAD DE ACCESO A INTERFAZ WEB .....	94
3.4 INSTALACIÓN DE SOFTWARE DE SEGUNDO PLANO, UTILIDADES Y ACCESORIOS .....	96

3.5 CONFIGURACIÓN DE ACCESO Y GESTIÓN REMOTA A TRAVÉS DE RED.....	106
3.6 DESARROLLO DE APLICACIONES DEL MENÚ PRINCIPAL.....	111
3.6.1 SISTEMA DE ALIMENTACIÓN.....	111
3.6.1.1 DISEÑO DEL DOSIFICADOR .....	112
3.6.1.2 CONTROL DEL SISTEMA.....	115
PROCESO PRINCIPAL .....	115
PROCESOS SECUNDARIOS .....	118
3.6.1.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ALIMENTACION .....	119
3.6.1.4 CIRCUITO DE CONTROL.....	120
3.6.1.5 VARIABLES DEL PROCESO .....	121
3.6.1.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA.....	124
3.6.2 SISTEMA DE ILUMINACIÓN .....	124
3.6.2.1 DISEÑO DEL SISTEMA DE ILUMINACIÓN.....	125
3.6.2.2 CONTROL DEL SISTEMA.....	126
3.6.2.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ILUMINACIÓN .....	130
3.6.2.4 CIRCUITO DE CONTROL DEL SISTEMA.....	130
3.6.2.5 VARIABLES DEL SISTEMA DE ILUMINACIÓN .....	131
3.6.2.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA.....	134
3.6.3 SISTEMA DE ACCESO .....	136
3.6.3.1 DISEÑO DE PUERTA AUTOMÁTICA .....	136
3.6.3.2 CONTROL DEL SISTEMA.....	138
3.6.3.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ACCESO .....	140
3.6.3.4 CIRCUITO DE CONTROL DEL SISTEMA.....	141
3.6.3.5 VARIABLES DEL SISTEMA DE ACCESO .....	142
3.6.3.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA.....	143

3.6.4 SISTEMA DE VENTILACIÓN .....	144
3.6.4.1 DISEÑO DEL SISTEMA DE VENTILACIÓN.....	144
3.6.4.2 CONTROL DEL SISTEMA.....	146
3.6.4.3 PUERTOS DE ARDUINO ASIGNADOS AL SISTEMA DE VENTILACIÓN.....	149
3.6.4.4 CIRCUITO DE CONTROL DEL SISTEMA.....	150
3.6.4.5 ARGUMENTOS DEL SISTEMA DE VENTILACIÓN.....	151
3.6.4.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA.....	152
3.6.5 SISTEMA DE VIDEOVIGILANCIA .....	152
3.6.5.1 DISEÑO DEL SISTEMA DE VIDEOVIGILANCIA.....	153
3.6.5.3 UBICACIÓN DE ELEMENTOS DEL SISTEMA.....	156
3.7 DISEÑO Y ELABORACIÓN DEL CENTRO DE CONTROL .....	157
3.7.1 ELEMENTOS DEL TABLERO DE CONTROL .....	157
3.7.2 ELABORACIÓN DEL CENTRO DE CONTROL .....	157
3.7.3 DISEÑO DE CIRCUITOS.....	160
3.7.4 CABLEADO DEL CENTRO DE CONTROL.....	163
3.8 DISEÑO DE FUENTE DE ALIMENTACIÓN DEL SISTEMA.....	165
3.8.1 DIAGRAMA DE ALIMENTACIÓN GENERAL DEL SISTEMA .....	166
3.8.2 CONEXIÓN DE FUENTE DE ALIMENTACIÓN GENERAL DEL SISTEMA.....	166
3.9 CONEXIÓN Y CABLEADO GENERAL DEL SISTEMA.....	168
4.1 SISTEMA DE ALIMENTACIÓN.....	173
4.2 SISTEMA DE ILUMINACIÓN .....	176
4.3 SISTEMA DE ACCESO .....	183
4.4 SISTEMA DE VENTILACIÓN .....	187
4.5 SISTEMA DE VIDEOVIGILANCIA .....	190
4.6 PRUEBAS DE CONTROL POR INTERFAZ Y PANTALLA TÁCTIL .....	191
.....	191

4.7 GRADO DE SATISFACCIÓN E INTEGRACIÓN DE LA CASA DOMÓTICA CON PERROS .....	192
4.8 ANÁLISIS DE RESULTADOS .....	194
5. CONCLUSIONES .....	196
6. RECOMENDACIONES .....	198
7. BIBLIOGRAFÍA. ....	201

## ÍNDICE DE IMÁGENES

Figura 1 Limitación del Hostal "El Bucanero" vista satelital .....	29
Figura 2 Limitación del Hostal "El Bucanero" street view .....	29
Figura 3 Placa de arduino mega 256 .....	33
Figura 4 Diseño de raspberry pi .....	35
Figura 5 Puertos GPIO .....	36
Figura 6 VNC .....	37
Figura 7 Interfaz PuTTY .....	38
Figura 8 Esquema de CSS .....	40
Figura 9 Lenguaje de programación JQuery .....	41
Figura 10 Lenguaje PHP .....	42
Figura 11 Pushetta .....	42
Figura 12 Motor eléctrico .....	43
Figura 13 Características de luminaria interior .....	45
Figura 14 Ventilador .....	46
Figura 15 Plywood .....	48
Figura 16 Sensor de temperatura LM35 .....	50
Figura 17 Módulo LDR .....	51
Figura 18 Funciones del sensor PIR .....	53
Figura 19 Módulo HC-SR501 .....	53
Figura 20 Esquema eléctrico del Módulo HC-SR501 .....	54
Figura 21 Driver L298 .....	55
Figura 22 Esquema eléctrico driver L298 .....	56
Figura 23 Esquema de conexión .....	56
Figura 24 Sensor de distancia de ultrasonido HC-SR04 .....	57
Figura 25 Relé de 8 canales .....	59
Figura 26 Módulo de grabación de voz ISD1820 .....	59
Figura 27 Tolva .....	61
Figura 28 Casa para perros .....	61
Figura 29 Casa para perro de madera .....	62
Figura 30 Casa de perro de plástico .....	63
Figura 31 Casa para perros para interiores .....	63



Figura 32 Nutrición del perro.....	64
Figura 33 Acicalamiento del perro.....	64
Figura 34 Ayuda profesional.....	65
Figura 35 Ejercicio.....	65
Figura 36 Educación de perros.....	66
Figura 37 Perro adulto.....	67
Figura 38 Cachorros.....	68
Figura 39 Ups.....	70
Figura 40 Router.....	70
Figura 41 PI Noir camera V2.....	71
Figura 42 Prototipo a escala del diseño estructural de la casa.....	74
Figura 43 Vista frontal del diseño estructural final de la casa.....	75
Figura 44 Vista lateral izquierda del diseño estructural final de la casa.....	75
Figura 45 Vista lateral derecho del diseño estructural final de la casa.....	75
Figura 46 Vista posterior del diseño estructural final de la casa.....	76
Figura 47 Vista superior de la casa.....	76
Figura 48 Resultado final de ensamble de estructura de casa domótica.....	77
Figura 49 Estructura en plywood marino y pruebas de dimensión.....	77
Figura 50 Estructura en plywood marino y pruebas de dimensión.....	78
Figura 51 Casa lijada, tratada y con base de color.....	78
Figura 52 Proceso de barnizado.....	79
Figura 53 Corte del linóleo plástico para la base.....	79
Figura 54 Pegado del linóleo a base de la casa.....	80
Figura 55 Resultado final del tapizado.....	80
Figura 56 Versión del sistema operativo.....	81
Figura 57 Instalación de Ios en SD CARD.....	81
Figura 58 Menú principal de herramienta de configuración de software del sistema.....	82
Figura 59 Submenú de la opción Preferences.....	84
Figura 60 Menú de configuración de interfaces en la ventana de configuración.....	85
Figura 61 Icono de red en escritorio Raspbian.....	85
Figura 62 Ejecución de comandos en ventana terminal raspberry pi.....	86
Figura 63 Cuadro de inicio de sesión.....	87
Figura 64 Página de inicio de interfaz web.....	91
Figura 65 Panel deslizante.....	91

Figura 66 Menú superior .....	91
Figura 67 Menú principal.....	92
Figura 68 Submenú de acceso.....	92
Figura 69 Directorio del servidor en explorador de archivos de raspbian .....	93
Figura 70 Ventana terminal raspberry pi .....	94
Figura 71 Sección de archivo Apache 2.....	95
Figura 72 Cuadro de autenticación en Mozilla Firefox .....	96
Figura 73 Pestaña de creación de nuevo canal en interfaz web PUSHETTA.....	96
Figura 74 Código API Key de la APP Pushetta.....	97
Figura 75 App Pushetta en Google Play .....	97
Figura 76 Canal suscrito en APP Pushetta.....	98
Figura 77 Opciones de canal en Pushetta.....	98
Figura 78 Notificación recibida en dispositivo Android.....	99
Figura 79 Descarga de software TeamViewer desde raspbian .....	100
Figura 80 Inicio se sesión de TeamViewer para raspbian.....	100
Figura 81 Ubicación de Tareas programadas en menú Raspbian .....	101
Figura 82 Cuadro de creación de tarea programada.....	101
Figura 83 Tareas programadas en CRON .....	102
Figura 84 Resultados del comando psaux en ventana terminal .....	102
Figura 85 Test de conexión de pantalla táctil.....	103
Figura 86 Diagrama de conexión de pantalla táctil a raspberry pi.....	103
Figura 87 Habilitación de cámara en menú interfaces de la configuración de raspberry pi.....	104
Figura 88 Test de cámara .....	105
Figura 89 Imagen capturada por cámara en el sistema .....	105
Figura 90 Diagrama de conexión de cámara a raspberry pi.....	105
Figura 91 Configuración DHCP de router Huawei propiedad de CNT.....	106
Figura 92 Menú de configuraciones de red.....	106
Figura 93 Configuraciones de red .....	107
Figura 94 Creación de dominio para servidor NO-IP .....	108
Figura 95 Configuración Dmz en router Huawei de proveedor Cnt.....	110
Figura 96 Ventana CMD realizando ping hacia dominio creado.....	110
Figura 97 Visualización de interfaz web iCAN en Chrome desde host de red externa. .....	111

Figura 98 Acceso a interfaz web ICAN desde teléfono Android con red móvil CLARO .....	111
Figura 99 Vista frontal del dosificador .....	112
Figura 100 Vista lateral del dosificador .....	112
Figura 101 Menú de alimentación .....	115
Figura 102 Selección de tipo de perro por raza o edad .....	116
Figura 103 Botón Servir porción del menú alimentación .....	116
Figura 104 Selección de tiempo de dosificación automática .....	117
Figura 105 Circuito del sistema de alimentación .....	120
Figura 106 Diagrama de conexiones del sistema de alimentación .....	120
Figura 107 Ubicación referencial de elementos del sistema de alimentación .....	124
Figura 108 Ubicación del dosificador al interior de la casa .....	124
Figura 109 Menú del sistema de iluminación .....	127
Figura 110 Submenú de modo automático .....	127
Figura 111 Selección de modo automático sensor .....	128
Figura 112 Selección de modo automático horario .....	129
Figura 113 Submenú de modo manual iluminación .....	129
Figura 114 Circuito del sistema de iluminación .....	130
Figura 115 Diagrama de conexiones del sistema de iluminación .....	131
Figura 116 Ubicación referencial de luminarias del sistema .....	134
Figura 117 Luminarias instaladas en zona de alimentación y descanso .....	135
Figura 118 Ubicación del sensor LDR .....	135
Figura 119 Sensor LDR conectado al sistema .....	135
Figura 120 Menú del sistema de acceso .....	139
Figura 121 Circuito del sistema de acceso .....	141
Figura 122 Diagrama de conexiones del sistema de acceso .....	141
Figura 123 Ubicación referencial de puerta y motor del sistema de acceso .....	143
Figura 124 Engrane, cremallera, puerta y FC abre del sistema de acceso .....	143
Figura 125 FC Cierra, sensor PIR interno y externo del sistema de acceso .....	144
Figura 126 Menú del sistema de ventilación .....	147
Figura 127 Submenú de modo automático .....	147
Figura 128 Submenú de modo manual .....	148
Figura 129 Circuito del sistema de ventilación .....	150
Figura 130 Diagrama de conexiones del sistema de ventilación .....	151

Figura 131 Ubicación referencial de extractores y sensor de temperatura .....	152
Figura 132 Extractores, caja contenedora y sensor de temperatura en zona de descanso .....	152
Figura 133 Configuración de parámetros de RPi_Cam_Web_Interface.....	155
Figura 134 Interfaz web del servidor RPi_Cam_Web_Interface.....	155
Figura 135 Iframe de transmisión de video en videovigilancia.html.....	156
Figura 136 Ubicación referencial de cámara PI NOIR V2 .....	156
Figura 137 Ubicación de cámara PI NOIR V2 en zona de descanso.....	157
Figura 138 Prototipo del centro de control .....	158
Figura 139 Plantilla vectorial para el corte laser de la plancha de acrílico.....	158
Figura 140 Corte laser de las piezas del centro de control y contenedor de sensores .....	159
Figura 141 Ubicación de elementos en tablero de acrílico .....	159
Figura 142 Resultado final del armado de las cajas de acrílico .....	159
Figura 143 Diseño final de la placa .....	160
Figura 144 Vista 3D del panel de conexión frontal .....	160
Figura 145 Diseño final del circuito de interconexión.....	161
Figura 146 Vista 3D del circuito de interconexión .....	161
Figura 147 Circuito impreso en papel transfer.....	162
Figura 148 Transferencia de impresión a baquelita de cobre .....	162
Figura 149 Circuito impreso en Baquelita .....	162
Figura 150 Circuito de interconexión terminado .....	163
Figura 151 Conexión de relés hacia bornera.....	164
Figura 152 Cable puente conectado en puertos GPIO del raspberry pi .....	164
Figura 153 Tablero de control cableado y energizado .....	164
Figura 154 Diagrama de alimentación general del sistema .....	166
Figura 155 UPS con cables soldados directo al interruptor de encendido del UPS. 167	
Figura 156 Conexión de fuente de alimentación general del sistema.....	167
Figura 157 Tomacorrientes con protección UPS .....	167
Figura 158 Ponchado de cables tipo puente para sensores .....	168
Figura 159 Conexión y pruebas de extractores del sistema de ventilación.....	168
Figura 160 Conexión del sensor de Temperatura LM35 .....	168
Figura 161 Pruebas de campo de detección del sensor PIR.....	169
Figura 162 Conexión del sensor PIR externo .....	169

Figura 163 Conexión del Dosificador de alimentos.....	169
Figura 164 Elaboración de la caja de resonancia para circuito reproductor de voz.	170
Figura 165 Conexión del circuito reproductor de voz y led en caja de resonancia .	170
Figura 166 Conexión del sensor de distancia del sistema de Alimentación .....	170
Figura 167 Conexión de pantalla táctil y cable de red .....	171
Figura 168 Conexión de sensores en panel de conexiones frontal.....	171
Figura 169 Integración del cableado a centro de control .....	171
Figura 170 Cableado final del centro de control.....	172
Figura 171 Vaciado de alimento en contenedor del dosificador.....	173
Figura 172 Alimento dentro del dosificador .....	173
Figura 173 Selección de opción “Servir Porción” para raza pequeña .....	174
Figura 174 Sistema de dosificación .....	174
Figura 175 Notificación en Pushetta de porción servida .....	174
Figura 176 Selección en modo automático del dispensador de comida.....	175
Figura 177 Proción de alimento servida por el dosificador para raza grande .....	175
Figura 178 Led de recarga de alimento encendido .....	176
Figura 179 Notificación en Pushetta de recarga de alimento.....	176
Figura 180 Luminaria L2 apagada .....	177
Figura 181 Luminaria de L1 apagada .....	177
Figura 182 Relés de iluminación desactivados .....	177
Figura 183 Selección On de “Luces Ambiente 1” .....	178
Figura 184 Relé activo de L1 .....	178
Figura 185 Encendido de L1 .....	179
Figura 186 Selección Off de L1 y On de L2.....	179
Figura 187 Relé de L1 apagado de y L2 encendido.....	179
Figura 188 Luminaria L2 encendida.....	180
Figura 189 Notificación Push de las acciones ejecutadas .....	180
Figura 190 Se selecciona el modo horario automático .....	180
Figura 191 Depuración de LUZ-DOSI.py para observar acciones en tiempo real ..	181
Figura 192 Se activa ambos relés de L1 y L2 .....	181
Figura 193 Depuración de LUZ-DOSI.py para observar acciones en tiempo real ..	181
Figura 194 Notificación Push del programa ejecutado .....	182
Figura 195 Selección del modo automático sensor.....	182
Figura 196 Depuración del proceso principal LUZ-DOSI.py.....	182

Figura 197 Selección de abierto en el menú de acceso .....	183
Figura 198 Depuración de ACCESO.py para observar en tiempo real el proceso ..	183
Figura 199 Puerta abierta de la casa domótica.....	184
Figura 200 Notificación Push que informa sobre la apertura de la puerta.....	184
Figura 201 Selección de cierre de la puerta .....	184
Figura 202 Depuración de ACCESO.py para observar en tiempo real el proceso ..	185
Figura 203 Notificación Push que informa sobre el cierre de la puerta.....	185
Figura 204 Puerta cerrada .....	185
Figura 205 Depuración de ACCESO.py con ejecución de seguridad para el cierre	186
Figura 206 La puerta intenta volver a cerrarse y esta vez es exitoso.....	186
Figura 207 Depuración de ACCESO.py para validar los procesos en tiempo real..	187
Figura 208 Notificación Push de corte de energía .....	187
Figura 209 Estado inicial de relés del sistema de ventilación.....	188
Figura 210 Selección de modo manual del sistema de ventilación.....	188
Figura 211 Estado activo de relés del sistema de ventilación .....	189
Figura 212 Extractores encendidos dentro de la zona de descanso. ....	189
Figura 213 Fijado 30 grados el valor de encendido del sistema .....	189
Figura 214 Fijado 30 grados el valor de encendido del sistema .....	190
Figura 215 Interfaz web de videovigilancia.....	191
Figura 216 Pantalla táctil con interfaz web.....	191
Figura 217 Perro asomado en el balcón de la casa.....	192
Figura 218 Perros descansando al interior de la casa.....	192
Figura 219 Perro en la zona de descanso de la casa domótica.....	193
Figura 220 Perro recibiendo aire del sistema de ventilación .....	193
Figura 221 Perro descansando luego de haberse alimentado en la zona de alimentación .....	193
Figura 222 Plano general de la casa .....	194
Figura 223 Vista lateral .....	194
Figura 224 Medidas de la cara frontal de la estructura .....	209
Figura 225 Medidas de la cara frontal de la estructura .....	209
Figura 226 Medidas de la cara lateral derecha de la estructura .....	210
Figura 227 Medidas de la estructura visto de cara superior.....	210

## ÍNDICE DE TABLAS.

Tabla 1 Especificaciones técnicas arduino mega 2560 .....	33
Tabla 2 Tipos de plywood marino .....	49
Tabla 3 Raciones de comida para perros adultos .....	67
Tabla 4 Raciones de comida para cachorros .....	68
Tabla 5 Puertos GPIO del sistema de alimentación .....	119
Tabla 6 Variables de HDOSIFICADOR.txt .....	123
Tabla 7 Puertos GPIO del sistema de iluminación.....	130
Tabla 8 Variables de LUCES.txt.....	134
Tabla 9 Puertos GPIO del sistema de acceso .....	140
Tabla 100 Variables de ACCESO.txt.....	142
Tabla 11 Puertos del arduino para el sistema de ventilación .....	150
Tabla 12 Parámetros que envía el raspberry pi al arduino vía serial .....	151

## RESUMEN

AÑO	TÍTULO	ALUMNOS	DIRECTOR DE TESIS	TEMA DE TITULACIÓN
2018	Ingeniería Electrónica	Bastidas Martínez Leonel Andrés.  Rodríguez Sares Kerly Andrea.	Ing. Ricardo Cajo. MET. Gino Alvarado	Diseño e implementación de una casa domótica para perros domésticos usando tecnología raspberry pi y arduino con control a través de interfaz web.

El presente proyecto de titulación “Diseño e implementación de una casa domótica para perros domésticos usando tecnología raspberry pi y arduino con control a través de interfaz web.” Tiene como objetivo brindar un cuidado ideal para los perros, facilitando la labor a los dueños, debido a que por sus arduas labores cotidianas se les imposibilita darle la atención que ellos necesitan.

Para el desarrollo del mismo contaremos con un sistema de acceso para el ingreso del can, junto con el dispensador de comida anexo a un grabador de voz que facilita la integración del perro con su nuevo hogar, haciendo del mismo un lugar placentero y acogedor, también posee un sistema de ventilación para mantener un clima idóneo, un sistema de iluminación y un sistema de vigilancia para que el dueño pueda monitorear el estado y las necesidades del can, todo ello será controlado mediante una tarjeta Raspberry Pi 3 junto a una tarjeta Arduino Mega, las cuales adquieren los datos obtenidos de los sensores ubicados en la casa y proceden a ejecutar acciones específicas que el usuario podrá seleccionar mediante la interfaz web.

Teniendo como resultado final una herramienta útil para ayudar al cuidado del perro, gracias a la domótica empleada en ello.



## ABSTRACT

YEAR	DEGREE	STUDENTS	TESIS DIRECTOR	TITLE TOPIC
2018	Electronic Engineering	Bastidas Martínez Leonel Andrés.  Rodríguez Sares Kerly Andrea.	Ing. Ricardo Cajo. MET. Gino Alvarado	Diseño e implementación de una casa domótica para perros domésticos usando tecnología raspberry pi y arduino con control a través de interfaz web.

The present project titled "Design and Implementation of a Home Automation for Domestic Dogs using Raspberry Pi and Arduino Technology with Control Through Web Interface." Its objective is to provide an ideal care for the dogs, facilitating the work to the owners, due that for their arduous daily tasks they are unable to give the attention they need.

For the development of the same we will have an access system for the entrance of the dog, along with the food dispenser attached to a voice recorder that facilitates the integration of the dog with his new home, making it a pleasant and welcoming place, too has a ventilation system to maintain a suitable climate, a lighting system and a monitoring system so that the owner can monitor the state and the needs of the dog, all this will be controlled by a card Raspberry Pi 3 next to an Arduino Mega card, which acquire the data obtained from the sensors located in the house and proceed to execute specific actions that the user can select through the web interface.

Having as final result a useful tool to help the dog care, thanks to the automation used in it.

## INTRODUCCIÓN

En el presente proyecto se encuentra el diseño e implementación de un sistema de casa domótica capaz de ayudar al cuidado de los canes, ya que en la actualidad son cada vez más las familias que adoptan al perro como mascotas, sin embargo de igual manera son muchas las que no poseen disponibilidad de tiempo para brindarle la atención y cuidado que necesitan ya sea debido a su estilo de vida, viajes largos o problemas puntuales, ellos tienden a sufrir de estrés o desórdenes alimenticios por falta de alimentación y cuidado. Esta problemática se viene acompañada de que en el mercado actual no existe una solución completa que sea capaz de brindar los cuidados esenciales que necesitan las mascotas y mucho menos que permitan llevar un canal de interacción y control con los mismos.

Para el entendimiento y total comprensión del tema se revisaron libros guía, y fuentes web, con el fin de poder consolidar los conocimientos referentes al caso de estudio:

Automatización, casa domótica, sistema de vigilancia y cuidado de canes en general, este último fue sustentado con veterinarios que ayudaron a despejar las dudas en base a todo lo prioritario para los canes, para así plasmar un escenario idóneo para los mismos.

Se obtiene como resultado final una herramienta favorable para los dueños de mascotas, centros veterinarios, refugios y ofertando al mercado hotelero una oportunidad de negocio innovadora, brindando un nuevo servicio a sus huéspedes.

## **1. EL PROBLEMA**

### **1.1 ANTECEDENTES**

Según estudios realizados por instituciones como: “Amigos con cola”, (FADA) Fundación amigos de los animales, (PAE) Protección animal Ecuador, entre otros, etc. Que fueron expuestos a través del diario METRO, [1] expresan que sólo en las ciudades de Guayaquil, Durán y Samborondón existe un alto índice de canes abandonados originándose esta problemática a causa que las personas no cumplen con la ordenanza del Reglamento de Tenencia y Manejo Responsable de Perros según el art. 3 [2] que “Todo propietario, tenedor y guía de perros estará obligado a mantener su mascota dentro de su domicilio, con las debidas seguridades, a fin de evitar situaciones de peligro para personas como para el animal”, siendo muchas veces el factor tiempo debido a las ocupaciones diarias que no permiten que se cumpla esta ordenanza , generando esta problemática que afecta mucho a los canes.

### **1.2 IMPORTANCIA Y ALCANCE**

Este proyecto de titulación tiene como finalidad el desarrollo de la casa domótica para perros, en que los mismos tengan un adecuado cuidado y que sus dueños mantengan control de su estado por medio de vigilancia, así como con servicios que permitirán que su alimentación sea automatizada, permanezcan en un ambiente seguro y con temperatura e iluminación correcta para su buen estado.

Ya que en la actualidad uno de los problemas que afrontamos es el abandono de los canes por motivos como:

- Alergias de familiares
- Inexperiencia
- Fallecimiento del propietario
- Sólo un antojo de los niños
- Nacimiento de un hijo
- Casas pequeñas o cambios de domicilio

- Comportamiento del animal
- Mantenimiento
- Animales viejos y enfermos
- Periodos vacacionales
- Ingreso de propietarios a hospitales
- Camadas inesperadas
- Falta de interés para cuidarlo

Todo esto viene dado por la irresponsabilidad, falta de conciencia, de educación y de ética que no se dan cuenta que están causando un gran mal a estos animales, ya que ellos también sienten y sufren.

La solución al ser basada en hardware y software libre puede ser referenciada y tomada como ejemplo para fundaciones o empresas privadas que quieran implementarla en beneficio de perros abandonados, centros de cuidado caninos, hostales especializados para mascotas o a su vez como solución para los hogares que necesitan tener un sistema que pueda brindarle los cuidados y la seguridad que sus mascotas merecen.

### **1.3 DELIMITACIÓN DEL PROBLEMA.**

#### **1.3.1 ESPACIAL**

Este proyecto será instalado y puesto en funcionamiento dentro del Hostal “EL BUCANERO”, propiedad de la Sra. Esperanza Roca, ubicado en la provincia de Santa Elena, ciudad Salinas, barrio Chipipe en la dirección Eloy Alfaro y Jaime Santander, después de haber sustentado en la ciudad de Guayaquil.



Figura 1 Limitación del Hostal "El Bucanero" vista satelital

Fuente: <https://www.google.com.ec/maps>



Figura 2 Limitación del Hostal "El Bucanero" street view

Recuperado de: <https://www.google.com.ec/maps>

### **1.3.2 TEMPORAL**

La elaboración de este proyecto de titulación está estimada en un tiempo mínimo de 6 meses, a partir de su aprobación, culminando y realizando su entrega en el periodo 2017 - 2018.

### **1.3.3 ACADÉMICA**

En este proyecto se pondrá en práctica todos los conocimientos ya adquiridos durante el periodo de formación como Ingeniero Electrónico con mención en Telecomunicaciones en la UPS-G, por partes de sus autores, dando énfasis a temas relacionados con: automatización industrial, sistemas micro procesados, redes, cableado estructurado, entre otras, para esto se sustentará bibliografías, textos y estudios que proporcionarán conceptos y teorías sobre el diseño del proyecto.

## **1.4 OBJETIVOS.**

### **1.4.1 OBJETIVO GENERAL**

Diseñar e implementar una casa domótica para perros domésticos usando tecnología Raspberry Pi y arduino con control a través de interfaz web.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- Diseñar un modelo de casa para perros con las prestaciones necesarias para la implementación del sistema domótico.
- Implementar un sistema automatizado de comida para mascotas.
- Implementar un sistema de ventilación y control de temperatura para el interior de la casa del perro.
- Implementar un sistema de iluminación automatizada para el interior de la casa de la mascota.
- Configurar un sistema de captura de video utilizando Webcam dentro del interior de la casa de la mascota.
- Diseñar un sistema de acceso de la mascota al interior de la casa domótica con control de bloqueo, apertura o modo automático a través de sensores.
- Establecer comunicación entre Raspberry y Arduino.

- Configurar un servidor para interfaz web usando Raspberry Pi con transmisión de datos.
- Programar en Python las instrucciones e información a ser utilizada en la interfaz web.
- Configurar la tarjeta Arduino Mega de manera que controle los actuadores y recoja la información de los sensores, además que envíe dichos datos a la Raspberry.
- Visualizar información de los sensores y establecer un control de los actuadores a través de la interfaz web del servidor Raspberry.

## **1.5 MARCO METODOLÓGICO**

Para el correcto funcionamiento e implementación de la casa domótica para perros domésticos, se maneja dos métodos que son:

### **1.5.1 MÉTODO EXPERIMENTAL**

Se realizó pruebas junto con los canes durante la culminación de cada fase, con el fin de observar la reacción que estostendrían. Se implementó distintas variables y condiciones con la finalidad de encontrar los parámetros idóneos para la automatización de la casa y confort de los perros.

### **1.5.2 MÉTODO EXPERIMENTAL**

Se ejecutó investigaciones técnicas y teóricas, inclusive se consultó a veterinarios con el objetivo de determinar las condiciones idóneas para los perros. Una vez obtenido los resultados se procedió a realizar pruebas con los materiales para la construcción y desarrollo del proyecto, teniendo como prioridad emplear un material no tóxico para los perros.

## **2. MARCO TEÓRICO**

### **2.1.¿QUÉ ES ARDUINO?**

Según [3], se puede decir que Arduino es comúnmente conocido como la plataforma de creación de códigos abiertos, de fácil manejo para los usuarios, basados en hardware y software.

Las placas de Arduino son muy versátiles ya que permiten leer entradas ya sea luz de sensores, un objeto sobre algún botón, leer mensajes y convertirlas en salidas, todo esto lo realiza gracias al conjunto de instrucciones que llegan al microcontrolador que se encuentra en el módulo, empleando un lenguaje programación basado en el cableado y en el software propio del Arduino.

Arduino es la herramienta más conocida a nivel mundial, siendo un gran aporte para la creación de varios proyectos tanto para estudiantes, aficionados, programadores, profesionales.

Naciendo en el instituto de Diseño de Interacción IBREA, siendo una herramienta de muy fácil uso para realizar prototipos de manera rápida y eficaz, va dirigido en especial a estudiantes de electrónica y programación. A raíz de extenderse de manera mundial tuvo que adaptarse a las nuevas necesidades y retos de los usuarios. Este software en la actualidad sigue creciendo gracias a los aportes de los miles de usuarios en todo el mundo.

#### **2.1.1 ARDUINO MEGA 2560**

El Arduino Mega 2560 es una placa electrónica, la cual consta de 54 pines digitales para las entradas y salidas, cabe recalcar que 15 de estas pueden ser utilizadas como salidas PWM, 16 entradas analógicas, 4 puertos de serie de hardware, un oscilador de cristal 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. [4].





Figura 3 Placa de arduino mega 256

Fuente: <http://arduino.cl/arduino-mega-2560/>

## 2.1.2 ESPECIFICACIONES TÉCNICAS ARDUINO MEGA 2560

Tabla 1 Especificaciones técnicas arduino mega 2560

Microcontrolador	Atmega2560
Tensión de funcionamiento	5 V
Voltaje de entrada	7-12 V
Voltaje DE entrada (limite)	6-20 V
Digital Pines I / O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente DC por pin I / O	20 mA
Corriente CC para Pin 3.3V	50 mA
Memoria flash	256 KB de los cuales 8 KB usado por cargador arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101.52 mm
Anchura	53.3 mm
Peso	37 g

Fuente: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

## 2.2. ¿QUÉ ES RASPBERRY PI?

Cuando nos referimos a Raspberry Pi sin duda alguna podemos dar el concepto de ser un computador, sólo que, con la placa reducida o simple, misma que fue desarrollado en Reino Unido por la Fundación Raspberry Pi, emplea sistemas operativos basados en Linux y su uso es muy frecuente en proyectos de electrónica y robótica. [5].

### 2.2.1 DISEÑO DE RASPBERRY PI

El diseño de la Raspberry Pi incluye:

- Un Chipset Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía),
- Un procesador gráfico (GPU) VideoCore IV.
- Un módulo de 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB).
- Un conector de RJ45 conectado a un integrado lan9512 -jzx de SMSC que nos proporciona conectividad a 10/100 Mbps.
- 2 buses USB 2.0.
- Una Salida analógica de audio estéreo por Jack de 3.5 mm.
- Salida digital de video + audio HDMI.
- Salida analógica de video RCA.
- Pines de entrada y salida de propósito general.
- Conector de alimentación microUSB.
- Lector de tarjetas SD. [6].

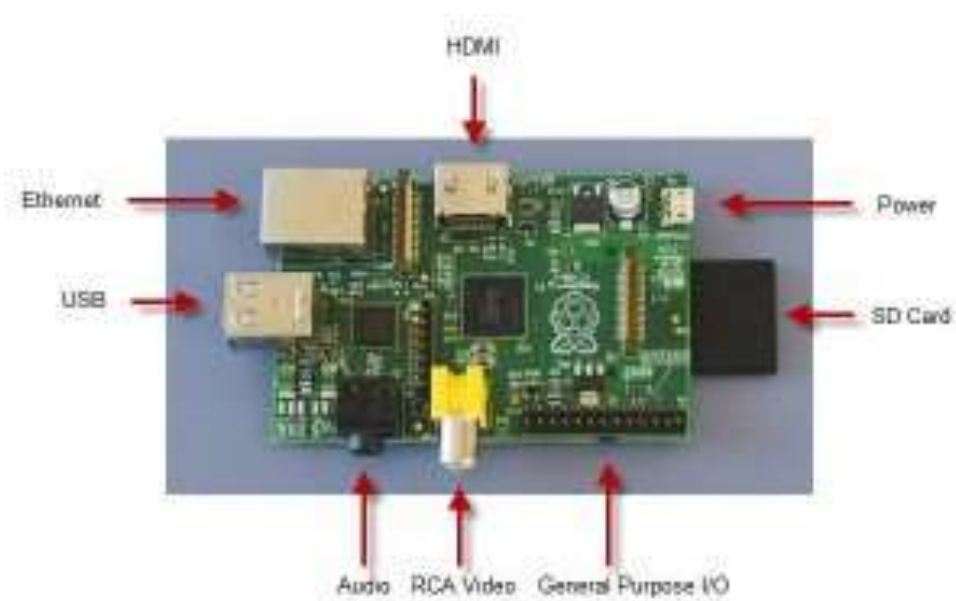


Figura 4 Diseño de raspberry pi

Fuente: <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

## 2.2.2 MODELOS DE RASPBERRY

Los esquemas del modelo A y el modelo B fueron lanzados el 20 de abril de 2012 por la fundación.

### 2.2.2.1 RASPBERRY PI 3 MODELO B

Sacada a la luz en el año 2016, renueva procesador, una vez más de la compañía Broadcom, una vez más un Quad-Core, pero pasa de 900MHz a 1.20GHz. Mantiene la RAM en 1GB. Su mayor novedad fue la inclusión de Wi-Fi 802.11n y Bluetooth 4.1 (baja energía) sin necesidad de adaptadores. [7]

### 2.2.3 PUERTOS GPIO

GPIO es el puerto el cual permite hacer la comunicación de dispositivos externos con la Raspberry Pi, ya que es un sistema entrada/salida. Consta de 26 pines de tipo unbuffered, los cuales pueden ser configurados según los requerimientos como entradas o salidas digitales.

Al poseer pines de masa y alimentación de 5V y 3,3V, comunicaciones seriales, I2C y SPI se recomienda trabajar con precaución cuando se conectan los dispositivos a los pines, debido a que no posee ninguna protección de circuitería, a diferencia del Arduino.

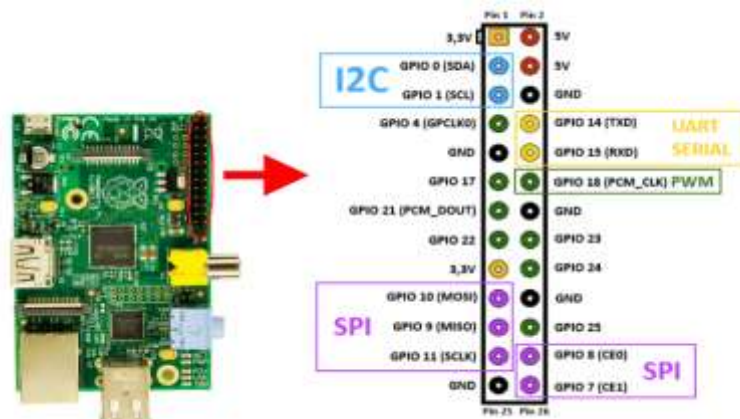


Figura 5 Puertos GPIO

Fuente: <http://diymakrs.es/usando-el-puerto-gpio/>

### 2.3.PYTHON

Cuando se habla de este lenguaje, se puede decir que [8]:

“Python es un lenguaje de programación que permite trabajar más rápido e integrar sus sistemas de manera más efectiva”.

Posee una gran biblioteca para soportar varias tareas comunes, otra opción que nos da es una interfaz programable a través de un aplicativo, y para ejecutar los programas de Python no se requiere un sistema operativo único, sino que es capaz de ejecutarse en

Mac, Windows, Linux y Unix sin ningún inconveniente y al ser un software libre no hay problemas al descargarlo y utilizarlo. [9].

## **2.4.SOFTWARE DE ACCESO REMOTO**

### **2.4.1 VNC**

Considerado como un software en cuya interfaz gráfica se puede visualizar la compartición de escritorio remoto de una computadora a la cual deseamos controlar, la misma ejecuta VNC en modo servidor y la conexión se realiza desde otro host el cual debe ejecutar VNC Viewer.

Este software permite interactuar con el host remoto a través de nuestro mouse y teclado local.



Figura 6 VNC

Fuente: <https://www.raspberrypi.org/documentation/remote-access/vnc/>

### **2.4.2 SSH**

Es un protocolo que permite acceder de forma remota a la línea de comando de la Raspberry Pi, desde otro computador o dispositivo en la misma red.

Uno de los softwares más usados para la conexión SSH es PuTTY, además es de conocimiento que dicho protocolo usa el puerto 22.



Figura 7 Interfaz PuTTY

### 2.4.3 SERVIDOR WEB

En la Raspberry Pi se puede usar el servidor web para albergar un sitio web completo sea este en la red local o global, o para visualizar la información seleccionada que se desee compartir a la red. [10].

Tenemos varios servidores web disponibles cada uno consta de ventajas enfocadas en su uso:

- Apache
- NGINX

#### 2.4.3.1 SERVIDOR APACHE

En la Raspberry Pi, el Apache es considerado como una aplicación de servidor web que permite servir páginas web.

Apache puede servir archivos HTML a través de HTTP y se le puede anexar módulos para servir páginas dinámicas. [11].

#### 2.4.3.2 NGINX

Conocido como motor x, también es considerada como una aplicación de servidor web en la Raspberry. [12].

## **2.5. INTERFAZ WEB**

Es un arreglo formado por una serie elementos gráficos, para su correcto diseño hay que tener en cuenta la interfaz web, ya que debe ser fácil, eficaz y grata para permitir una navegación satisfactoria a los usuarios en el sitio web. [13].

### **2.5.1 COMPONENTES DE UNA INTERFAZ WEB**

#### **2.5.1.1 ELEMENTOS DE IDENTIFICACIÓN**

Son aquellos que permiten al usuario identificar al sitio web al que están ingresando, ya sean estos: logos, título de la página, dominio, iconos, etc. [13].

#### **2.5.1.2 ELEMENTOS DE NAVEGACIÓN**

Son indispensables en todo diseño web, ya que son los encargados de permitir a los usuarios navegar por todas las páginas webs, debido a que se encuentran en todas las páginas web del sitio. [13].

#### **2.5.1.3 ELEMENTOS DE CONTENIDOS**

Se los encuentra en cada página web ya que es la zona donde se puede visualizar toda la información más importante de dicha página. [13].

#### **2.5.1.4 ELEMENTOS DE INTERACCIÓN**

Son aquellas cuya función es permitir la interacción del usuario con la página web, por ejemplo: una caja de búsqueda. [13].

## 2.5.2 LENGUAJE DE INTERFAZ WEB

### 2.5.2.1 CSS

Traducido al español las siglas CSS (Cascading Style Sheets) es Hoja de estilo en cascadas, llamado así ya que su aplicación es de arriba a abajo, cuya función es aplicar estilos ya sean estos colores, formas, etc., a varios documentos de una forma masiva.

La finalidad de CSS es poder separar la presentación del contenido, es decir los documentos HTML, solo tengan información y datos en este caso nos referimos al contenido y por otro lado los colores, diseño, formas, etc., son los relacionados a la presentación, obteniendo que los documentos CSS se encuentren independientes.

Trayendo consigo algunas ventajas, ya que, al querer realizar alguna modificación en la presentación, se podrá realizar en un solo sitio. [14]



Figura 8 Esquema de CSS

Fuente: <https://lenguajecss.com/p/css/introduccion/que-es-css>

### 2.5.2.2 JQUERY

Es la librería abierta, que permite reducir la programación en JavaScript, posee una gran cantidad de plugins ya sean unos gratis y otros pagados, cada uno de ellos tiene un sitio web que permiten descargar sus licencias, demos, archivos, etc. [15]





Figura 9 Lenguaje de programación JQuery

Fuente: <http://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>

### 2.5.2.3 PHP

Es aquel que emplea CSS para separar los diseños de los contenidos de la página web, posee un lenguaje gratuito y multiplataforma, lo cual permite crear páginas dinámicas.

Cuando hablamos de independencia de la base de datos y servidor web sin duda alguna PHP es la solución. Tiene como finalidad integrar las paginas HTML

Posee varias características como:

Velocidad: PHP es rápido a la hora de la ejecución y se integra bien con otras aplicaciones en especial en el sistema operativo Unix

Estabilidad: cuenta con su propio sistema de administración

Seguridad: los niveles de seguridad manejados por PHP pueden ser configurados desde .ini

Simplicidad: debido a que posee una gama de librerías, se le puede agregar extensiones, es de fácil entendimiento, más aún si ya se posee conocimientos de C Y C++. [16]

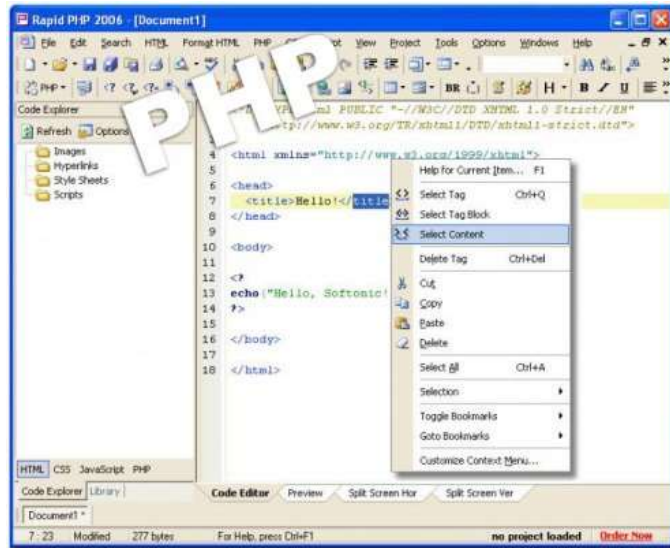


Figura 10 Lenguaje PHP

Fuente: <http://redgrafica.com/El-lenguaje-de-programacion-PHP>

## 2.6. PUSHETTA

Según lo indican [17]:

“Pushetta está hecho para que sea sencillo enviar comunicaciones de difusión a grupos de suscriptores”.

Consiste en crear un grupo con algún tema en particular, en el cual cada usuario que ingrese al mismo podrá recibir una notificación. [17].

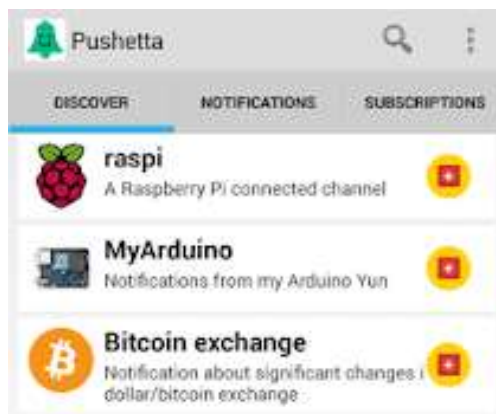


Figura 11 Pushetta

Fuente: <https://play.google.com/store/apps/details?id=com.gumino.pushetta>

## **2.7. MOTOR ELÉCTRICO**

Se puede decir que [18]:

“El motor eléctrico es un artefacto que transforma la energía eléctrica en energía mecánica, de manera que puede impulsar el funcionamiento de una máquina, esto ocurre por acción de los campos magnéticos que se generan gracias a las bobinas”.



Figura 12 Motor eléctrico

Fuente: <http://www.ermec.com/mellor/notas.htm>

### **2.7.1 USOS**

#### **2.7.1.1 CAMBIO DE SENTIDO DE GIRO**

En corriente alterna el cambio de sentido de giro se puede realizar en motores monofásicos, con invertir las terminales del devanado de arranque y para motores trifásicos se debe cambiar dos de las conexiones de alimentación, mientras que en la corriente continua se invierte los contactos del par de arranque. [19].

## **2.7.2 CAJA REDUCTORA**

Se conoce como aquella que permite mantener la velocidad de salida para el funcionamiento deseado, este mecanismo se basa en el conjunto de engranajes, por lo general se emplea un tornillo sin fin que ayuda a reducir la velocidad considerablemente. [20].

## **2.8.PUERTA AUTOMÁTICA**

Según los glosarios se define que [21]:

“Puerta constituida por hojas acristaladas o rellenas, móviles puestas en movimiento por un sistema de apertura automática.”

### **2.8.1 TIPOS DE PUERTAS**

Existen diversidad de tipos de puertas según el requerimiento que se desee darle [22]:

- Puertas correderas
- Puertas batientes
- Puertas basculantes
- Particiones peatonales

## **2.9. LÁMPARA LED**

En la actualidad las lamparas led tienen un gran uso comercial y decorativo, si se define el concepto se puede decir que [23]:

“Viene del inglés L.E.D traducido diodo emisor de luz. Se trata de un cuerpo semiconductor sólido de gran resistencia que, al recibir una corriente eléctrica de muy baja intensidad, emite luz de forma eficiente y con alto rendimiento”.


 <p><b>Downled Basic Uniled 18W Gama Basic</b></p> <p>Luminaria led Uniled Gama Basic. Gama de luminarias para iluminación interior.</p>			
<b>Descripción</b>			
Equipo para la sustitución de downlight convencionales.			
Esta luminaria led consigue un ahorro de más del 70 % con una mayor calidad de luz.			
<b>Características</b>			
Nombre:	Downled Basic Uniled 18W	Lámpara:	PLACA LEDS UNILED
Referencia:	Gama Basic	Unidades:	1
Categoría:	Iluminación Interior	Tipo de Lámpara:	Placa Uniled
Ubicación:	Interior	Tipo de Casquillo:	Placa Uniled
Instalación:	Empotrar	Tipo de Luz:	Fría-Neutra-Cálida
Medidas (m.m.):	205 Ø hueco, 225 Ø exterior, 1	Tensión:	110-220V
Peso (Kg.):	0.85	Potencia (w):	Consultar ficha técn
Grado Protección IP:	20	Número de Leds:	0
Color:	Blanco	Lúmenes:	Consultar ficha técn
Proyección:	Directa	Ángulo de Haz:	--
Potencia total(w):	0		

Figura 13 Características de luminaria interior

Fuente: [http://www.unile0d.es/luminarias\\_leds\\_interior/iluminacion\\_interior/170\\_leds.html](http://www.unile0d.es/luminarias_leds_interior/iluminacion_interior/170_leds.html)

### 2.9.1 CARACTERÍSTICAS DE ILUMINACIÓN CON LÁMPARAS LEDS

Las lámparas Leds tienen algunas ventajas como:

- Lentes ópticos: son empleados para lograr captar un mejor ángulo de iluminación en sus interiores.
- Lámparas con LEDs de alta intensidad: son aquellas en las cuales se emplean 0,5 Watts, gracias a este cambio en la actualidad se genera un 50% más de brillo. [24].

## 2.9.2 TIPOS DE LÁMPARAS LEDS

Los tipos de lamparas leds son [25]:

- LED industrial
- LED exterior
- LED comercial
- LED luz de emergencia
- Tubos LED
- LED Elite ´

## 2.10. VENTILADOR

Cuando se denota la palabra ventilador se dice que [26]:

“El ventilador es básicamente una máquina que tiene como objetivo principal la generación de una corriente de aire permanente que, sin embargo, puede variar su intensidad o dirección de acuerdo con el interés de quien lo maneje”.



Figura 14 Ventilador

Fuente: <http://conceptodefinicion.de/ventilador/>

## **2.10.1 TIPOS DE VENTILADORES**

Pueden ser clasificados en [27]:

“Ventiladores para el hogar:

- De mesa
- De pared
- De techo
- De piso

Ventiladores industriales:

- Centrífugos
- Helicoidales
- Axiales”

## **2.11. PLYWOOD**

Se define el contrachapado como [28]:

“El contrachapado también conocido como multilaminado, plywood, triplay o madera terciada, es un tablero elaborado con finas chapas de madera pegadas con las fibras transversalmente una sobre la otra con resinas sintéticas mediante fuerte presión y calor”.

### **2.11.1 ELABORACIÓN**

Para realizar la elaboración del contrachapado se colocan los troncos en una máquina, haciéndolos girar para realizar el corte, con la finalidad de obtener una hoja de chapa, luego pasa a una estufa, se procede a corregir sus imperfecciones y para culminar se pega a presión y a una temperatura de 140 °C. [29].

### **2.11.2 PLYWOOD MARINO**

Se define como [30]:

“PLYWOOD MARINO es un tablero fabricado con resina fenólica resistente al agua”.



Figura 15 Plywood

Fuente: [http://www.tablecom.com.ec/index.php?option=com\\_virtuemart/](http://www.tablecom.com.ec/index.php?option=com_virtuemart/)

#### **2.11.2.1 USOS**

Por su gran tenacidad a la humedad puede ser utilizado en [31] :

- Construcciones navales
- Cubiertas
- Pisos
- Campamentos
- Encofrados

#### **2.11.2.2 CARACTERÍSTICAS**

El plywood marino consta de [31]:

- Perfecta inmovilidad estructural en todo el tablero
- Disponible en calibres para satisfacer todas las necesidades de uso



### 2.11.2.3 TIPOS DE PLYWOOD MARINO

Tabla 2 Tipos de plywood marino

TIPOS DE PLYWOOD MARINO				
USO EXTERIORES				
CHAPA	CALIBRE (mm)	DIMENSIONES	TOLERANCIA A	
			EN ESPESOR	EN CUADRATURA
SANDE	4-5	1.22 X 2.44mts 4x8 pies	+0.2mm/-0.7mm	+/- 2.5mm
VIROLA	9-12-15	0.91 x 2.14mts 3x7 pies		+/- 2.5mm
ANIME	18			+/- 2.5mm

Fuente: <http://www.emfalu.com/images/plywood-marino.pdf>

## 2.12. SENSORES

Al hablar de sensores se dice que [32]:

“Un sensor o captador, como prefiera llamársele, no es más que un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular”.

### 2.12.1 SENSOR DE TEMPERATURA

Cuando se trata de temperatura, es algo usual que la mayoría de las personas conocen, pero ¿Cómo se produce la medición de esta?, gracias a que el sensor transforma las



## 2.12.2 SENSOR DE LUZ

Su función es reconocer las variantes en la intensidad de la luz, para su correcto funcionamiento constan de dos componentes: un emisor y receptor, el cargo del emisor es producir la luz y del receptor es percibir la luz producida.

Se lo emplea para la detección, posicionamiento y clasificación de objetos.

Los sensores de luz realizan la conversión de luz a la señal eléctrica, gracias a que poseen un transductor fotoeléctrico. Existen tres tipos de sensores fotoeléctricos, los sensores por barrera de luz, reflexión sobre espejo o reflexión sobre objetos. [35]

### 2.12.2.1 MÓDULO LDR

Cuando se requiere controlar la iluminación, o hacer seguidores solares, etc., el módulo LDR es el indicado ya que es aquel que cambia basado a la cantidad de luz suministrada.

El módulo consta de dos salidas, una analógica y otra digital, la analógica debe ser conectada a una entrada analógica y emplear un conversor ADC, mientras que la salida digital se la encuentra sólo ya sea encendido o apagado, dependiendo del indicio fijado en el potenciómetro. [36].



Figura 17 Módulo LDR

Fuente: [http://www.naylampmechatronics.com/661-thickbox\\_default/modulo-ldr.jpg](http://www.naylampmechatronics.com/661-thickbox_default/modulo-ldr.jpg)

## **ESPECIFICACIONES TÉCNICAS**

Las especificaciones técnicas del módulo LDR son [37]:

- “Voltaje de Operación: 5V DC
- Conexión de 4 cables: VCC, GND, DO, AO
- Salida analógica y digital(comparador)
- Opamp en modo comparador: LM393
- Potenciómetro para ajuste de comparador
- Led rojo de encendido y verde de salida digital”.

### **2.12.3 SENSOR DE FINAL DE CARRERA O DE CONTACTO**

El interruptor de límite es aquel encargado de enviar señales capaces de cambiar el estado de un circuito.

En su estructura interna pueden poseer tres tipos de interruptores, como:

- Normalmente abierto
- Normalmente cerrado
- Conmutados. [38].

### **2.12.4 SENSOR PIR**

Son aquellos capaces de detectar las variantes en la radiación infrarroja que receptan, provocando una alarma.

los sensores de movimientos son los más usuales, estos están separados en dos mitades para así detectar el cambio de radiación IR que receptan ambos lados, provocando el accionamiento de la alarma al producir dicha variación. [39].

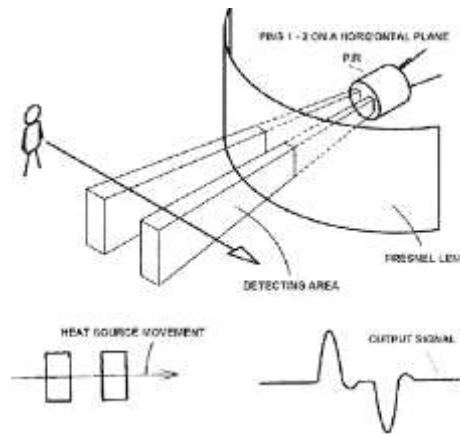


Figura 18 Funciones del sensor PIR

Fuente: <https://www.promotec.net/sensor-pir/#>

#### 2.12.4.1 MÓDULO HC-SR501



Figura 19 Módulo HC-SR501

Fuente: <https://electroni1ab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>

### CARACTERÍSTICAS

- *Sensor piro eléctrico (Pasivo) infrarrojo (También llamado PIR)*
- *El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización*
- *Rango de detección: 3 m a 7 m, ajustable mediante trimmer (Sx)*
- *Lente fresnel de 19 zonas, ángulo < 100°*
- *Salida activa alta a 3.3 V*

- *Tiempo en estado activo de la salida configurable mediante trimmer (Tx)*
- *Redisparo configurable mediante jumper de soldadura*
- *Voltaje de alimentación: 4.5 VDC a 20 VDC. [40].*

### 2.12.4.1.1 ESQUEMA ELÉCTRICO

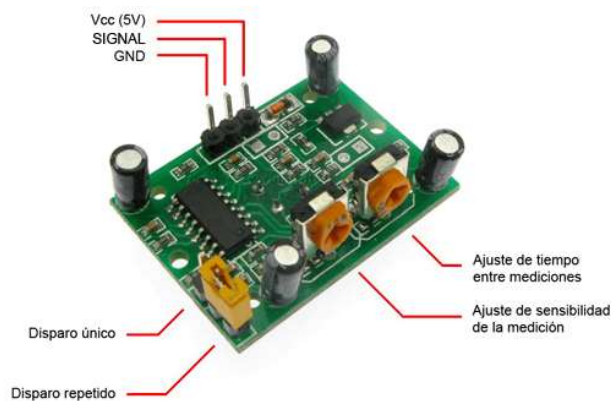


Figura 20 Esquema eléctrico del Módulo HC-SR501

Fuente: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

### 2.12.5 PUENTE H

Se define como [41]:

“Es una configuración circuital de diodos que nos permite controlar el giro de un motor, básicamente los diodos actúan como controladores de la polaridad que se le aplica al motor”.

#### 2.12.5.1 DRIVER L298

El módulo del driver L298 permite dominar dos motores tanto de corriente continua como paso a paso máximo hasta 2 amperes, en la actualidad se puede encontrar a disposición el driver soldado y con pines, a lo cual lo llamamos módulo. [41].

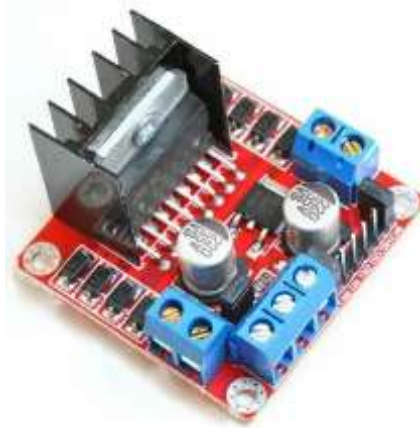


Figura 21 Driver L298

Fuente: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

#### 2.12.5.1.1 CARACTERÍSTICAS

- Regulador de voltaje LM7805
- Conectores **Output A** y **Output B** que son las salidas de los motores
- Control inputs, son los **terminales de control**, dos de ellos son los pines de habilitación de cada motor. [41].

#### 2.12.5.1.2 ESQUEMA DE MONTAJE

Para el esquema de montaje se tiene conocimiento que [42]:

“La placa de conexión del L298N incorpora una entrada de voltaje, una serie de jumpers para configurar el módulo, dos salidas A y B, y los pines de entrada que regulan la velocidad y el sentido de giro”.

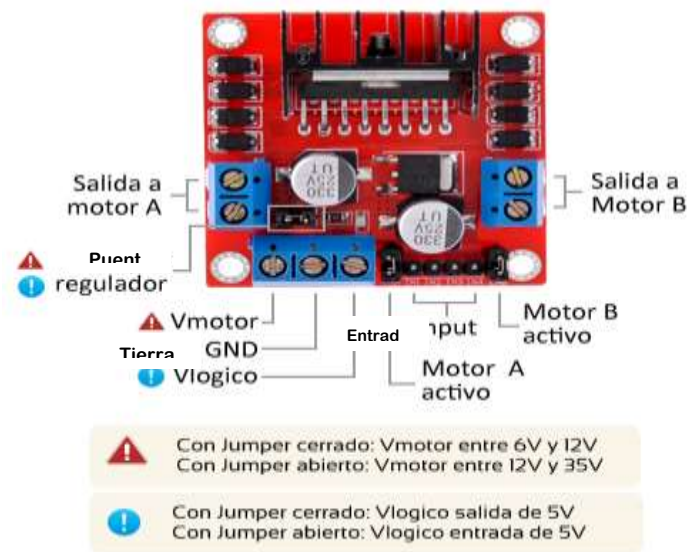


Figura 22 Esquema eléctrico driver L298

Fuente: <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>

### 2.12.5.1.3 ESQUEMA DE CONEXIÓN

En el caso de querer usar ambas fases, y poder elegir tanto el sentido de giro como la velocidad, el esquema de conexión sería el siguiente [42]:

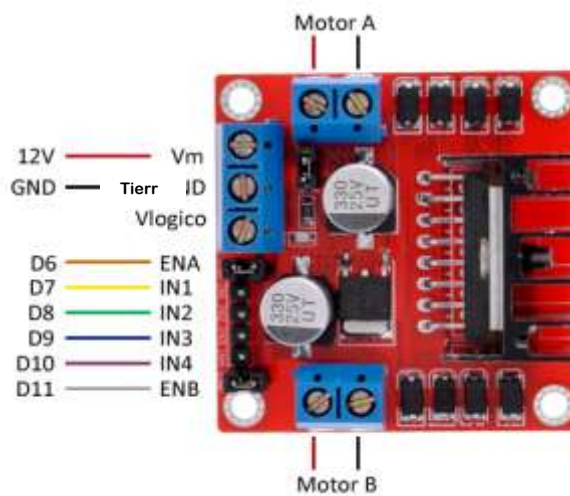


Figura 23 Esquema de conexión

Fuente: <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>



## 2.12.6 SENSOR ULTRASÓNICO

Se define como [43]:

“Son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros”.

### 2.12.6.1 FUNCIONAMIENTO

Para realizar el correcto funcionamiento el sensor trabaja de la siguiente manera [43]:

“Emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración”.

### 2.12.6.2 SENSOR DE DISTANCIA DE ULTRASONIDO HC-SR04

Se define como [44]:

“Sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos, contiene toda la electrónica encargada de hacer la medición”.



Figura 24 Sensor de distancia de ultrasonido HC-SR04

Fuente: <https://www.luisllamas.es/arduino-motor-corriente-continua-1298n/>

### **2.12.6.2.1 CARACTERÍSTICAS**

Pines de conexión:

- VCC
- Trig (Disparo del ultrasonido)
- Echo (Recepción del ultrasonido)
- GND
- Rango mínimo: 1.7 cm. [44].

### **2.13. RELÉ**

Se define al relé como [45]:

“Es un aparato eléctrico que funciona como un interruptor, abrir y cerrar el paso de la corriente eléctrica, pero accionado eléctricamente. El relé permite abrir o cerrar contactos mediante un electroimán, por eso también se llaman relés electromagnéticos o relevador.”

#### **2.13.1 MÓDULO RELÉ DE 8 CANALES OPTOACOPLADOR**

Este módulo de relevadores (relés) para conmutación de cargas de potencia. Los contactos de los relevadores están diseñados para conmutar cargas de hasta 10A y 250VAC (o 30VDC), aunque se recomienda usar niveles de tensión por debajo de estos límites. [46].



Figura 25 Relé de 8 canales

Fuente: <http://www.ecopech.net/modulo-relay-optoacoplado-de-8-canales.html>

## 2.14. GRABADOR DE VOZ

### 2.14.1 MÓDULO GRABACIÓN DE VOZ ISD1820C



Figura 26 Módulo de grabación de voz ISD1820

Fuente: <https://www.prometec.net/producto/modulo-de-grabacion-de-voz/>

#### 2.14.1.1 CARACTERÍSTICAS

Alguna de las características son [47]:

- “chip principal: ISD1820.
- Tamaño: 38mm \* 42.5mm.
- voltaje de funcionamiento: DC 3 ~ 5 V.

- Altavoz: 8R, 0.5 W.
- Sencillo de uso. Hasta 10 segundos de grabación de voz.
- Buena calidad, Permite modo reproducción continua.”

## **2.15. DOSIFICADORES**

Al referirse de dosificadores se puede decir que [48]:

“El dosificador en una envasadora es el encargado de suministrar la dosis de producto justa en cada paquete que se realiza. Se selecciona de acuerdo con las características del producto a envasar”.

Para la alta producción estos pueden ser acoplados a máquinas envasadoras automáticas, mientras que en baja producción será en semiautomáticas. [48].

### **2.15.1 TIPOS DE DOSIFICADORES**

A continuación, se detalla los tipos de dosificadores existentes:

Dosificador VOLUMÉTRICO

Dosificador a tornillo SIN FÍN

Dosificador GRAVIMÉTRICO (a Balanzas)

Dosificador a PISTÓN

Dosificador ISOBÁRICO [49].

## **2.16. TOLVA**

Se denomina tolva a [50]:

“Caja en forma de tronco de pirámide o de cono invertido y abierta por debajo, dentro de la cual se echan granos u otros cuerpos para que caigan poco a poco.”



Figura 27 Tolva

## 2.17. CASA PARA PERROS

Hoy en día contamos con una infinita variedad de modelos de casas para los perros, todas estas se adaptan a sus necesidades no sólo de ellos sino también de sus dueños, cabe recalcar que los canes necesitan tener su propio espacio puesto que les da la seguridad para cualquier circunstancia que se les presente, aportando con calidez en días de frío y frescura en días calurosos.



Figura 28 Casa para perros

Fuente: <https://www.muyperruno.com/casas-para-perros/>

### **2.17.1 CARACTERÍSTICAS IDÓNEAS PARA ELECCIÓN DE CASA PARA PERROS**

En el amplio mundo de casas para perros hay que tener en cuenta ciertas características estándares que necesitan poseer. [51]:

Durabilidad y resistencia al agua

Piso Elevado

Fácil Ensamblaje

Fresca en el verano / caliente en el invierno

Fácil de limpiar.

### **2.17.2 TIPOS DE CASAS PARA PERROS**

#### **1) CASAS DE PERRO DE MADERA**



Figura 29 Casa para perro de madera

Fuente: <https://www.muyperruno.com/casas-para-perros/>

Son las casas más conocidas en el medio asignadas para perros, se suele escoger el material madera debido a que posee las ventajas de soportar calor y a su vez humedad,

gracias a que es un aislante natural. Es recomendable emplear sellador no tóxico y ningún otro producto químico que pueda perjudicar a los perros. [51].

## 2) CASAS DE PERRO DE PLÁSTICO



Figura 30 Casa de perro de plástico

Fuente: <https://www.muyperruno.com/casas-para-perros/>

Una opción más moderna que la madera, es el plástico. Aunque no tiene la belleza natural de la madera, tiene muchas otras ventajas. El plástico es muy durable y a diferencia de la madera es naturalmente resistente al agua. [51].

## 3) CASAS PARA PERROS PARA INTERIORES



Figura 31 Casa para perros para interiores

Fuente: <https://www.muyperruno.com/casas-para-perros/>

Suelen ser hechas en material tela, debido a que no necesitan soportar diversas situaciones climáticas que puedan deteriorarla. [51].

## 2.18. CUIDADO DE LOS PERROS

Es un campo super extenso cuando se trata de hablar del cuidado de estos animales, más sin embargo existen ciertos parámetros importantes de considerar los cuales son:

### NUTRICIÓN



Figura 32 Nutrición del perro

Fuente: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>

Como todo ser vivo es una parte fundamental para el correcto crecimiento de los perros, ya que una correcta nutrición se verá reflejada en su aspecto físico, en su motricidad y salud. [52].

### ACICALAMIENTO



Figura 33 Acicalamiento del perro

Fuente: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>



Existen factores importantes como corte de pelo, recorte de uñas, cuidado de los oídos y limpieza de la cara son indispensable para los cuidados de cada raza. [52].

#### AYUDA PROFESIONAL



Figura 34 Ayuda profesional

Fuente: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>

Se considera que los expertos e indicados en dar el asesoramiento son los profesionales tales como veterinarios o entrenadores profesionales que llevan años en esta labor. [52].

#### EJERCICIO



Figura 35 Ejercicio

Fuente: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>

Uno de los ejercicios esenciales para ejercitar a los perros son las caminatas. En la actualidad existen parques exclusivos para realizarlas.

Otros ejercicios considerables son natación, jugar al escondite o tira y afloja. [52].

## EDUCACIÓN



Figura 36 Educación de perros

Fuente: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>

Según [52] se considera que:

“Todo perro la necesita o no podrá convivir en esta sociedad. La educación incluye socialización, obediencia básica, poner reglas y utilizar reforzamiento positivo. No incluye castigos o utilizar gritos. Ten autoridad, pero con benevolencia y amor”.

### **2.18.1 RACIONES DIARIAS DE COMIDA PARA PERRO**

#### **2.18.1.1 CANTIDAD DE COMIDA RECOMENDADA PARA UN PERRO ADULTO**

Para saber la porción ideal de comida que debe ser suministrada para los canes nos basaremos en [53].

Tabla 3 Raciones de comida para perros adultos

<b>¿CUÁNTO DEBE COMER UN PERRO?</b>				
<b>Porción diaria de alimento seco adultos (gramos)</b>				
<b>TAMAÑO</b>	<b>PESO (Kg)</b>	<b>ACTIVIDAD ALTA</b>	<b>ACTIVIDAD NORMAL</b>	<b>BAJA ACTIVIDAD</b>
MINIATURA	2-5	60-115	55-100	45-85
PEQUEÑO	5-10	115-190	100-170	85-145
MEDIANO	10-15	190-225	170-225	145-195
GRANDE	15-25	255-380	225-330	195-285
	25-40	380-535	330-475	285-410
	40-55	535-680	475-600	410-520
	55-70	680-820	600-720	520-620
	70-90	820-985	720-870	620-750

Fuente: <http://www.elcomercio.com/narices-frias/cantidad-alimentacion-perros-mascotas-salud.html>



Figura 37 Perro adulto

Fuente: <https://www.expertoanimal.com/cantidad-de-comida-diaria-para-perros-20020.html>

## 2.18.1.2 CANTIDAD DE COMIDA RECOMENDADA PARA UN CACHORRO

Tabla 4 Raciones de comida para cachorros

¿CUÁNTO DEBE COMER UN PERRO?						
Porción diaria de alimento seco cachorros(gramos)						
TAMAÑO	PESO (Kg)	2 MESES	3 MESES	4 MESES	5 MESES	6-12 MESES
MINIATURA	2	50	60	60	60	56
PEQUEÑO	5	95	110	115	115	110
	10	155	185	195	190	185
MEDIANO	17	215	265	285	285	280
GRANDE	25	270	350	375	375	370
	32	300	400	445	450	450
	40	355	475	525	530	530
	50	405	545	610	625	

Fuente: <http://www.elcomercio.com/narices-frias/cantidad-alimentacion-perros-mascotas-salud.html>



Figura 38 Cachorros

Fuente: <https://www.expertoanimal.com/cantidad-de-comida-diaria-para-perros-20020.html>

## **2.19 ACERO INOXIDABLE**

Se lo conoce como [53]:

“El acero inoxidable es una aleación (combinación o mezcla) de hierro (Fe) y carbono (C) siempre que el porcentaje de carbono sea inferior al 2%. Este porcentaje de carbono suele variar entre el 0,05% y el 2% como máximo”.

### **2.19.1 TIPOS DE ACERO INOXIDABLE**

Hay 4 tipos de aceros inoxidables, aunque el 65% del acero inoxidable utilizado es el austenítico, los cuales son [53]:

*Acero Inoxidable Austenítico*

*Acero Inoxidable Ferrítico*

*Acero Inoxidable Martensítico*

*Acero Inoxidable Dúplex.*

## **2.20 UPS**

Se define al UPS como [54]:

“Sistema de alimentación interrumpida, su función se basa en la protección eléctrica para los equipos corrigiendo automáticamente las variaciones de voltaje”.



Figura 39 Ups

Fuente: <http://www.forzaups.com/ec/forza-smart-ups-sl-800ul110v#support>

## 2.21 ROUTER

Se define como [55]:

“Un router es un dispositivo de red que se encarga de llevar por la ruta adecuada el tráfico”.



Figura 40 Router

Fuente <https://www.definicionabc.com/tecnologia/router.php>

### 2.21.1 FUNCIONAMIENTO

Para su correcto funcionamiento [55].

“Utiliza direcciones IP para saber a donde tienen que ir los paquetes de datos no como ocurre en los switches. Gracias a estas direcciones, que son únicas para cada máquina, este dispositivo puede conocer por donde debe enviar el paquete”.

## **2.22 SOFTWARE RPI – CAM – WEB - INTERFACE**

Se dice de este software que [56] :

“Es una interfaz web para el módulo Raspberry Pi Camera. Se puede utilizar para una amplia variedad de aplicaciones, incluida la vigilancia, grabación dvr y fotografía de largo de tiempo”.

## **2.23 PI NOIR CAMERA V2**

Se define como [57]:

“Es un panel de alta calidad de 8 megapíxeles con sensor de imagen IMX219 de Sony diseñado para Raspberry Pi, con lente de enfoque fijo. Es capaz de imágenes estáticas de 3280 x 2464 píxeles, y también admite video”.



Figura 41 PI Noir camera V2

Fuente: <https://www.raspberrypi.org/products/pi-noir-camera-v2/>

### **3. DESARROLLO INTEGRAL DE LA ESTRUCTURA Y SISTEMA DOMÓTICO DE LA CASA PARA MASCOTAS.**

#### **3.1 DISEÑO Y ENSAMBLAJE DE LA ESTRUCTURA DE LA CASA PARA MASCOTAS**

##### **3.1.1 MATERIALES Y METODOLOGÍA**

Para el diseño de la casa, se investigó diferentes materiales tomando en consideración su durabilidad, resistencia, maleabilidad, dimensión, grosor, peso y facilidad de manejo. El resultado de estas investigaciones determina que el material idóneo para la construcción es el Plywood marino el cual es resistente ante la humedad, el agua y a los rayos UV, adicional permite mantener una rentabilidad durante la elaboración del proyecto. El mismo es tratado para impedir la proliferación de bacterias u hongos que afecten su durabilidad.

El piso del interior de la casa está cubierto por un linóleo plástico para facilitar su limpieza e impermeabilidad, además puede ser remplazado fácilmente.

El techo de la casa es de material plástico resistente al agua y a los rayos UV, su función es cubrir la casa durante las lluvias y evitar que caiga la luz del sol directamente sobre la casa lo que genera calor.

El diseño estructural de la casa es único y desarrollado por los autores del presente proyecto, basados en distintas estructuras comerciales y hospitalarias para canes, se toma en consideración las necesidades del animal, se analizó el comportamiento y uso real que prestarían las mascotas al proyecto implementado.

Para la construcción de los distintos sistemas domóticos se empleó materiales que no causen afectación al can y sean resistentes, de la misma manera todo el cableado y disposición de los sensores se encuentran protegidos antes males de la intemperie, daño intencional o no intencional del can.

Tanto los materiales, diseño y dimensionamiento de la casa fueron expuestos ante veterinarios profesionales quienes dan su aprobación y certificación para la elaboración de la casa automatizada. [30], [58].



### **3.1.2 DIMENSIONALIDAD DE ELEMENTOS**

El grosor de las paredes de la casa es de 8 mm en la parte frontal, izquierda, derecha y superior. La parte posterior estará cubierta por una plancha de 5 mm.

La escalera se encuentra construida con grosor de 8mm lo cual es suficiente para soportar el peso de uno o varios canes grandes.

Los vidrios de las paredes tienen un grosor de 5 mm garantizando su resistencia a vibraciones y leves golpes.

La altura del interior de la casa es de 80 cm, el perro promedio tiene una altura de 70 cm en posición erguida con lo que se satisface las necesidades promedio de las mascotas.

### **3.1.3 ESTRUCTURA DE LA CASA**

Consiste en 3 comparticiones divididas en:

#### **ZONA DE DESCANSO**

Se refiere a la zona en la cual está ubicado el mueble de descanso para la mascota. Esta contiene el sistema de ventilación, la cámara de video vigilancia, una luminaria y la puerta de acceso principal automática.

#### **ZONA DE ALIMENTACIÓN**

Contiene el dosificador automatizado de mascotas, una luminaria, una pantalla LCD y una puerta de acceso manual la cual puede ser usada por el dueño en caso de alguna emergencia, mantenimiento, para recargar alimento dentro del contenedor o a su vez para facilitar la limpieza y acceso al interior de la casa.

Esta zona se encuentra separada de la zona de descanso por una pequeña pared para su distinción.

### **ALTILLO**

Sobre la zona de descanso se encuentra un altillo al cual se accede por medio de unas escaleras ubicadas en la zona exterior, pegadas a la pared lateral izquierda de la casa, la misma consiste en un espacio abierto sobre el cual puede subir la mascota para descansar, tomar aire o refrescarse. Posee un balcón para impedir que la mascota caiga, así como un techo que cubre del sol toda la estructura.

#### **3.1.4 DISEÑO ESTRUCTURAL DE LA CASA**

Para determinar el diseño final estructural de la casa se elaboró una maqueta a escala con la finalidad de tomar en consideración las necesidades y requerimientos de los objetivos planteados, así como el confort del perro.



Figura 42 Prototipo a escala del diseño estructural de la casa

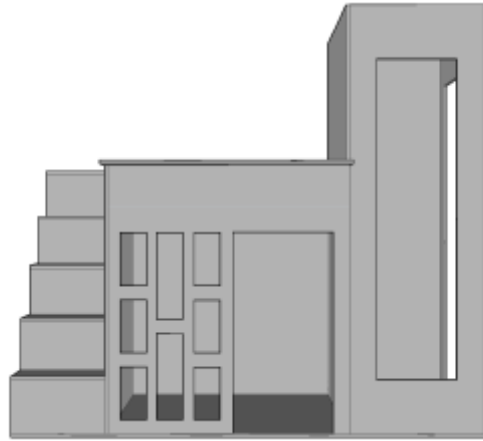


Figura 43 Vista frontal del diseño estructural final de la casa.

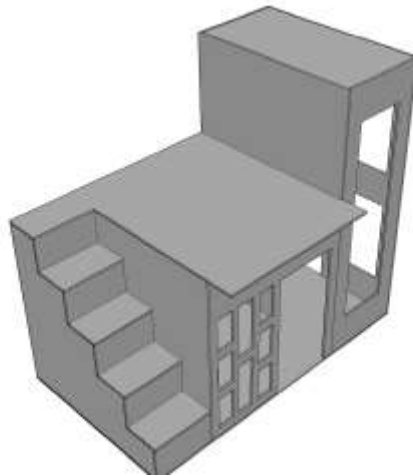


Figura 44 Vista lateral izquierda del diseño estructural final de la casa

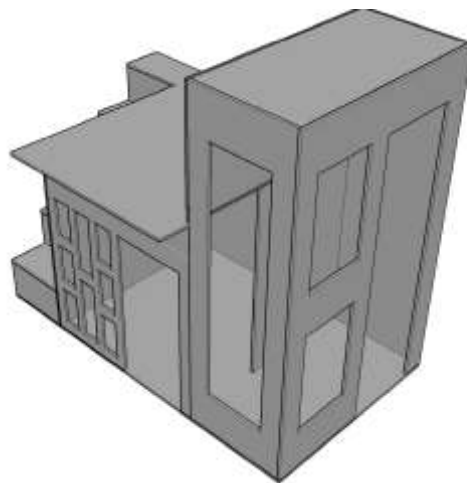


Figura 45 Vista lateral derecho del diseño estructural final de la casa

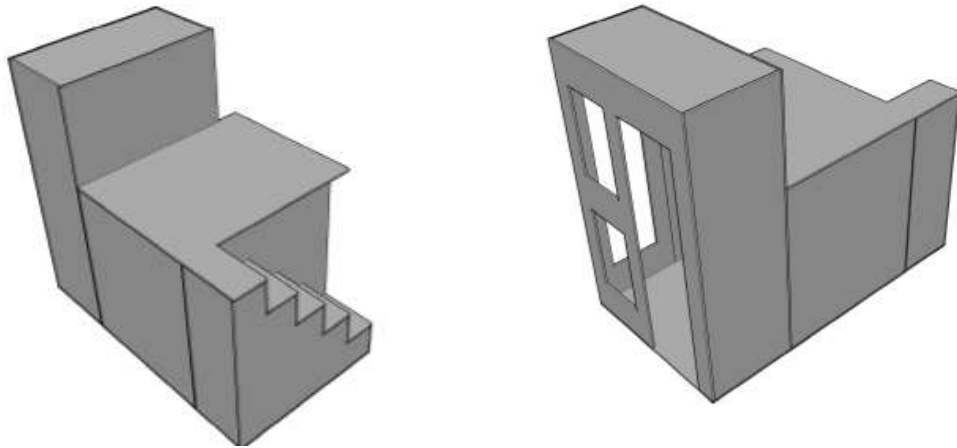


Figura 46 Vista posterior del diseño estructural final de la casa

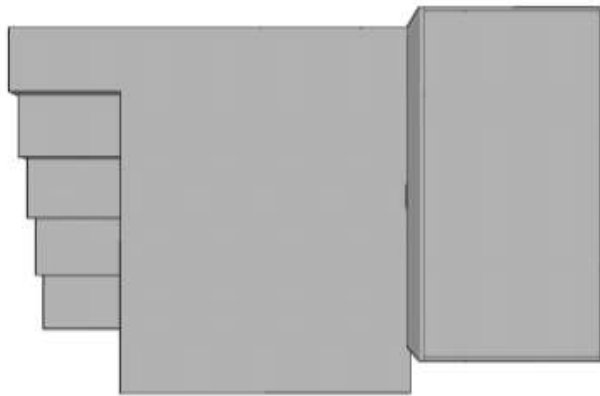


Figura 47 Vista superior de la casa

\* Las gráficas con medidas se encuentran ubicadas en el anexo del presente documento.

\* Las gráficas fueron elaboradas en AUTOCAD 2016.

### **3.1.5 CONSTRUCCIÓN Y ENSAMBLAJE DE LA CASA**

En base al diseño se hizo la compra y corte del Plywood Marino con dimensiones específicas.

Bajo la asesoría de un especialista en carpintería se realizó el ensamble y armado de la estructura tomando en cuenta la ubicación final de los sensores y actuadores que posteriormente serán colocados en la estructura.



Figura 48 Resultado final de ensamble de estructura de casa domótica.



Figura 49 Estructura en plywood marino y pruebas de dimensión



Figura 50 Estructura en plywood marino y pruebas de dimensión

### **CURACIÓN Y LAQUEADO DE MADERA:**

Se adquirió los materiales necesarios, se recibió la asesoría y recomendaciones necesarias para el proceso de curado y laqueado de la casa.



Figura 51 Casa lijada, tratada y con base de color



Figura 52 Proceso de barnizado

### **TAPIZADO DE BASE:**

Se tapizó la base de la casa con un linóleo plástico impermeable para facilitar la limpieza y dar más protección.



Figura 53 Corte del linóleo plástico para la base



Figura 54 Pegado del linóleo a base de la casa



Figura 55 Resultado final del tapizado

## **3.2 HABILITACIÓN Y CONFIGURACIÓN INICIAL DE RASPBERRY PI**

### **3.2.1 INSTALACIÓN DEL SISTEMA OPERATIVO**

El sistema operativo usado para la implementación del sistema es RASPBIAN, el cual es el IOS nativo de Raspberry. La descarga de la imagen se realiza desde su página oficial:





Figura 56 Versión del sistema operativo

Fuente: <https://www.raspberrypi.org/downloads/raspbian/>

Para este proyecto se elige la versión RASPBIAN STRETCH WITH DESKTOP debido a que su interfaz gráfica facilita la interacción con el usuario. [59].

Se hace uso del programa para Windows win32 Disk Imager. En la interfaz del software se selecciona la imagen del IOS previamente descargado, se elige la unidad de la tarjeta de memoria y se selecciona la opción “Write”, con esto se procederá a instalar el sistema operativo en la SD CARD, este proceso toma varios minutos.

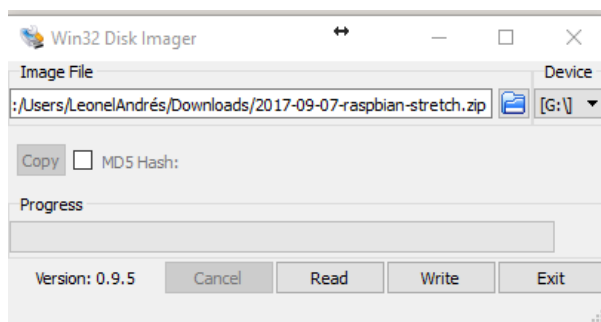


Figura 57 Instalación de Ios en SD CARD

La tarjeta es colocada en el Raspberry, al encenderlo se valida el arranque del sistema operativo y de tener conector un monitor nos mostrará el escritorio de Raspbian.

### 3.2.2 CONFIGURACIONES INICIALES

En el primer encendido de la tarjeta Raspberry PI es necesario conectar un monitor, teclado y mouse al dispositivo para realizar y habilitar las configuraciones iniciales.

Se hace uso del comando “sudo raspi-config” desde una ventana terminal para acceder a la herramienta de configuración de Software del sistema desde donde se realizarán las siguientes configuraciones como parte de las buenas prácticas y recomendaciones en seguridad y estabilidad.

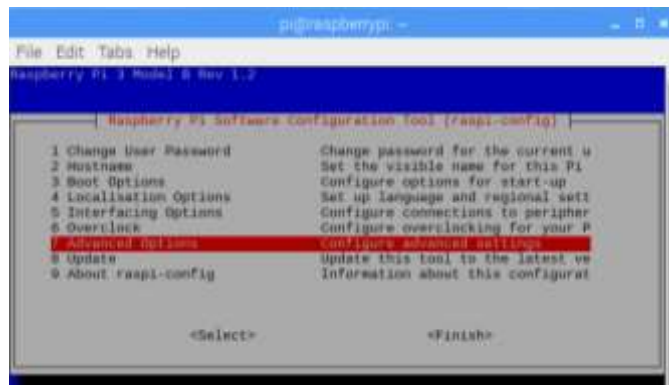


Figura 58 Menú principal de herramienta de configuración de software del sistema

### EXPANSIÓN DEL TAMAÑO DE PARTICIÓN:

Se expande el tamaño de la partición de manera que ocupe la capacidad total de memoria disponible en la SD CARD sobre la cual se encuentra instalado el sistema.

Para su habilitación se elige en orden las siguientes opciones:

1. Advance option
2. Expand Filesystem

## **CAMBIO DE HOSTNAME:**

Con la finalidad de identificar fácilmente el sistema dentro de una red local realizamos el cambio de Hostname, esto se realiza con fines de seguridad para que no se pueda identificar fácilmente la plataforma sobre la cual se tiene implementado los servicios.

Seleccionamos la opción 2 del menú principal “Hostname” y a continuación se ingresa el nombre elegido. Se nombra al sistema como **UPSiCAN**.

## **CAMBIO DE ZONA HORARIA:**

Se elige la zona horaria para que el sistema pueda actualizar la hora correcta según la ubicación geográfica.

Se sigue el siguiente procedimiento:

1. Se selecciona Localisation Options
2. Cambiar Timezone
3. Seleccionamos América
4. Se elige Lima debido a que comparte la misma zona horaria que Ecuador

Este cambio se verá reflejado al reiniciar la Raspberry.

## **CONFIGURACIONES DE TECLADO:**

La configuración del teclado se realiza desde la opción Mouse and Keyboard Setting del menú Preferences.



Figura 59 Submenú de la opción Preferences

Se realiza el siguiente procedimiento:

En la pestaña Keyboard seleccionamos Keyboard Layout.

En Country Seleccionamos España y en Variant español.

Para validar que la selección se realizó correctamente tipamos en el cuadro de texto donde se debe constatar que coincida con la tecla presionada.

### **SELECCIÓN DE IDIOMA Y PAIS:**

Se selecciona Raspberry Pi configuration en la pestaña del Menu Preferences.

En Localization, se selecciona Set Locale y se elige las siguientes opciones:

Language: es(Spanish)

Country: EC (Ecuador)

Character set: UTF – 8

Culminadas estas configuraciones es necesario reiniciar la Raspberry para que el idioma de la interfaz cambie a español.

## HABILITACION DE ACCESOS REMOTOS (VNC Y SSH)

En la opción Raspberry Pi Configuration del Menu Preferences, seleccionamos la pestaña interfaces y se habilita SSH y VNC.

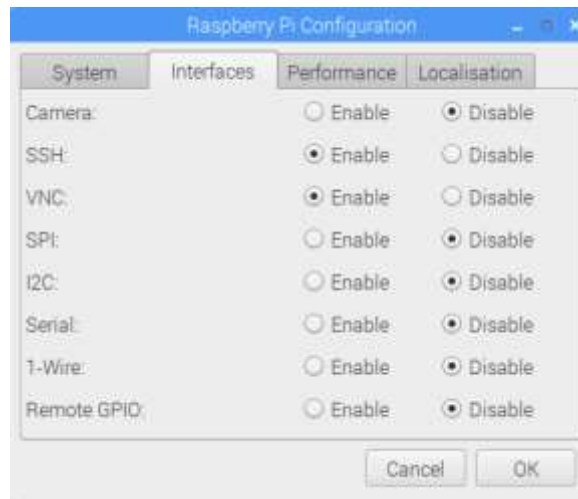


Figura 60 Menú de configuración de interfaces en la ventana de configuración

## CONFIGURACIONES DE RED:

Se conecta la Raspberry PI a un puerto Ethernet de manera que el servidor DHCP del router le asigne una IP, de no tener activado el servidor DHCP en nuestro router se configura la misma estáticamente.

De manera fácil Para conocer la IP que tiene nuestra Raspberry ubicamos el mouse sobre el siguiente icono.



Figura 61 Icono de red en escritorio Raspbian

De realizar clic sobre ese ícono se puede realizar la conexión a través de WIFI.

## CAMBIO DE NOMBRE AL USUARIO PI Y CONTRASEÑA:

Por cuestión de seguridad es sumamente importante realizar un cambio de contraseña y de usuario a PI, por defecto la clave del usuario es “raspberrypi” razón por la cual cualquier usuario podría tomar gestión del sistema y efectuar cambios.

Para realizar este cambio es necesario habilitar temporalmente el usuario root ya que viene deshabilitado por defecto. Para esto seguiremos los siguientes pasos:

### HABILITACIÓN DE USUARIO ROOT:

Con ingresar el comando: “sudo passwd root” asignamos una contraseña al usuario root con lo cual se habilita la misma.



```
pi@raspberrypi:~ $ sudo passwd root
sudo: unable to resolve host raspberrypi
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
pi@raspberrypi:~ $
```

Figura 62 Ejecución de comandos en ventana terminal raspberry pi

Se ingresa la siguiente configuración:

USER: root

PASS: UPS\_salesiana\_2017

Para efectuar los cambios se cierra la sesión actual y se inicia sesión con el usuario root, se coloca la clave antes ingresada y se presiona enter. Se abre una pantalla terminal y se ingresa los siguientes comandos:

```
Usermod -l iCAN pi -md /home/iCAN
groupmod -n iCAN pi
mv /home/iCAN /home pi
```

Con estos comandos se registra al usuario iCAN y se le otorga los privilegios y directorios de pi.

Por último, se asigna una nueva contraseña al usuario iCAN con el comando:

Sudo passwd iCAN

Se asigna los siguientes usuarios:

USER: iCAN

PASS: Kleo\_2017

Se cierra sesión y se ingresa con el usuario creado:

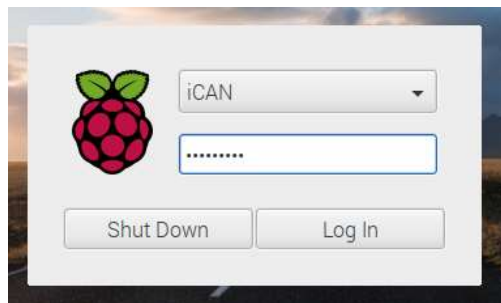


Figura 63 Cuadro de inicio de sesión

Iniciada la sesión por cuestiones de seguridad se procede a deshabilitar el usuario root con el comando “sudo passwd –l root”, el sistema solicita la clave y se da enter. En caso de requerir nuevamente privilegios de superusuario se puede proceder con la activación de este siguiendo los pasos antes mencionados.

## **LIBERACIÓN DE ESPACIO**

Para liberar espacio en la memoria y optimizar el sistema se desinstala Software innecesario y preinstalado en el firmware con el siguiente comando:

```
sudo apt-get --purge remove wolfram-engine bluej greenfoot nodered nuscratch scratch sonic-pi libreoffice* claws-mail claws-mail-i18n minecraft-pi python-pygame
```

Se reinicia la Raspberry y se ejecuta el siguiente comando:

```
sudo apt-get autoremove –purge
```

## **ACTUALIZACIÓN DEL SISTEMA**

El proceso de actualización del sistema consta de tres simples comandos:

```
sudo apt-get -y update
```

Con este comando ingresado en el terminal se refresca y actualiza la lista de repositorios de los cuales el sistema obtiene las últimas actualizaciones para los paquetes instalados. El `-y` es para evitar que solicite confirmación.

```
sudo apt-get -y upgrade
```

Por medio de este comando el sistema descarga las últimas actualizaciones de Software de los programas instalados en el sistema.

```
sudo apt-get clean
```

Finalmente digitamos el comando `apt-get clean` para eliminar cualquier paquete de instalación innecesario.

Con los procedimientos antes mencionados se culmina la configuración inicial y se habilita seguridad en el Raspberry PI.

El desarrollo de cada uno de los servicios y aplicaciones se ira detallando a continuación:

### **3.3 DESARROLLO DE LA INTERFAZ WEB**

#### **3.3.1 CONSIDERACIONES DEL DISEÑO**

Para el diseño de la interfaz web se hizo uso de las librerías de código abierto Bootstrap. Estas librerías dan facilidad para crear páginas web dinámicas que se adapten al cambio de pantalla e interactúen con las selecciones realizadas por parte del usuario.

El contenido de la página web está codificado en HTML5 y su diseño en CSS, se hace uso de librerías de jQuery correspondiente a JavaScript con lo cual se logra obtener



una página web responsiva en la que se pueda ir seleccionando las distintas opciones de una manera dinámica y que da fácil acceso a cada una de las opciones a elegir. De igual manera se hace uso de código PHP, se crearon distintos scripts con código PHP los cuales hacen de puente y conexión entre la interfaz web y los programas codificados en Python ejecutados a nivel de servidor, por medio de estos programas PHP se realiza la actualización de archivos .txt guardados en el directorio files, los mismos que son luego interpretados por los programas Python. Estos programas PHP son llamadas a través de funciones JQuery las cuales están definidas para cada elección tomada en la página web. Es necesario realizar la ejecución de los aplicativos de esta manera ya que JavaScript es ejecutado a nivel de cliente, en este caso los navegadores; en cambio Python se ejecuta a nivel de servidor, en este caso PHP es el intérprete para que ambos servicios se enlacen.

### **3.3.2 INSTALACIÓN DE SERVIDOR APACHE Y PHP**

Para que la casa domótica para perros pueda ser controlada a través de una interfaz web es necesario levantar un servidor apache en el Raspberry PI de manera que por medio de la dirección local de este dispositivo desde cualquier navegador se puede acceder al contenido de la interfaz.

Para realizar la instalación del servidor APACHE en el sistema ejecutamos el siguiente comando el cual descargará los ficheros desde un repositorio:

```
#sudo apt install apache2 apache2-mod-php7.0
```

Luego realizamos la instalación de PHP, así como de sus librerías:

```
#sudo apt-get install php7.0
```

Finalmente realizamos un reboot en la Raspberry con lo que se concluye la instalación del servidor.

```
#sudo reboot
```

## PERMISOS DE USUARIO Y DIRECTORIO APACHE:

Para tener administración del contenido en la carpeta /var/www y poder ejecutar instrucciones desde la web ejecutamos los siguientes comandos los cuales dan permisos administrativos al usuario iCAN sobre la carpeta en mención:

```
sudo adduser iCAN www-data ( Añade el usuario iCAN al grupo www-data)
sudo chown iCAN:www-data -R /var/www /html
sudo chmod 775 -R /var/www/html
sudo chmod g+s -R /var/www/html
```

Para que el usuario www-data tenga permisos sobre el puerto serial se ejecuta el siguiente comando:

```
usermod -a -G dialout www-data
```

Finalmente, para que la interfaz tenga privilegios de ejecución por terminal se realiza el siguiente cambio en las configuraciones del sistema:

En terminal se digita:

```
Sudo visudo
```

Al final del documento se agrega el siguiente código:

```
www-data ALL=(root) NOPASSWD:ALL
```

Se realiza un reinicio del servidor con lo que concluimos las configuraciones del servidor Apache para poder tener control del sistema desde la interfaz web.

```
sudo /etc/init.d/apache2 restart
```

Este cambio es necesario ya que de lo contrario las instrucciones que ingresemos por medio de la interfaz web no causarían efecto en el servidor.

### 3.3.3 DISEÑO DEL ENTORNO GRÁFICO EN LA INTERFAZ WEB

#### PÁGINA DE INICIO



Figura 64 Página de inicio de interfaz web

Consta de un panel deslizable donde se muestran gráficas acordes al proyecto y una pequeña descripción del producto



Figura 65 Panel deslizable

Un menú de cuatro opciones ubicada en la esquina superior derecha donde se puede seleccionar y obtener información más detallada del producto, así como reiniciar y apagar el sistema.



Figura 66 Menú superior

En su parte inferior central contiene 5 botones a través de los cuales se accede a ventanas emergentes que contienen las distintas selecciones a tomar por parte del usuario.



Figura 67 Menú principal

### VENTANA EMERGENTE

Son cuadros emergentes que se habilitan al seleccionar las opciones del menú principal. En estas se realizan las configuraciones y se ingresan los parámetros con los cuales queremos que el sistema trabaje.



Figura 68 Submenú de acceso

### 3.3.4 ESTRUCTURA DEL DIRECTORIO WEB DEL SISTEMA

La ubicación del directorio de la interfaz web y aplicativo es:

`/var/www/`

En este directorio se encuentran los distintos aplicativos y contenidos distribuidos en las siguientes carpetas:

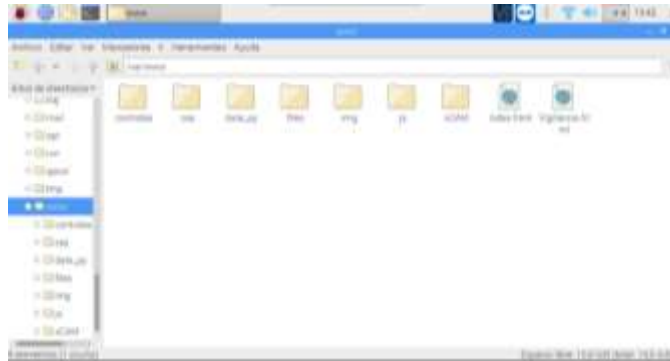


Figura 69 Directorio del servidor en explorador de archivos de raspbian

### **CSS, Js e img**

Contienen información de configuración, estructura y diseño de la interfaz web.

### **INDEX.HTML:**

Es la página principal que se muestra al ingresar la IP local del sistema.

### **VIGILANCIA.HTML**

Muestra la página a través de la cual monitoreamos el sistema de vigilancia.

### **CONTROLES:**

Contiene los scripts de las funciones PHP que son llamadas desde la interfaz web de index, las mismas ejecutan instrucciones y a su vez hacen el llamado de las funciones Python.

### **Data\_py**

Contiene los scripts Python encargados de activar y desactivar los puertos GPIO del Raspberry.

### **Files:**

Contiene los documentos txt en donde se guardan los valores y parámetros seleccionados en la interfaz web.

### **XCAM:**

Directorio del Software RPi-Cam-Web-Interface con el cual convertimos la Raspberry en un servidor para visualizar de manera remota las imágenes obtenidas por la Camera.

### 3.3.5 SEGURIDAD DE ACCESO A INTERFAZ WEB

Con la finalidad de proteger el acceso a la interfaz gráfica y de otorgar seguridad al servidor se restringe el acceso por medio de un usuario y contraseña, los mismos que son aplicados través de las directivas de autenticación `.htaccess`, `.htpasswd` y que son solicitados al instante en que se intenta acceder a la página desde cualquier host. Estas instrucciones son propias del servidor web Apache.

Primero se realiza la instalación de la directiva **htpasswd**, la misma se encuentra incluida en el paquete **apache2-utils**.

Desde una terminal de Raspbian digitamos lo siguiente:

```
sudo apt-get install apache2-utils
```

Con la ayuda de este programa se procede a generar los pares de usuarios – contraseñas que tendrán acceso al directorio del sistema sobre el cual se aplique la restricción. Es recomendable que las credenciales no se encuentren ubicadas en el mismo directorio del sistema, razón por la cual son ubicadas en una carpeta contenido en home. Para generar las credencias se ejecuta el siguiente código:

```
htpasswd -c /home/iCAN/credenciales/.htpasswd admin
```

Luego nos solicitará clave, para esta ocasión hemos creado la siguiente credencial:

Usuario: admin

Password: UPSiCAN



```
iCAN@UPSiCAN:~$ htpasswd -c /home/iCAN/USERS/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
```

Figura 70 Ventana terminal raspberry pi

De querer agregar más usuario se repite el mismo paso, pero sin el atributo `-c`. Cabe recalcar que las credenciales guardadas en esta localidad son encriptadas por el programa y están ocultas.

Creadas las credenciales se genera un txt de nombre .htaccess en el directorio del sistema que se protege, de esta manera le indicamos al servidor APACHE que use las credenciales generadas para autenticar el acceso al directorio y sus subcarpetas.

En el archivo .htaccess se ingresa el siguiente código y se guarda:

```
AuthUserFile /home/iCAN/credenciales/.htpasswd
/home/iCAN/credenciales/.htpasswd AuthType Basic
AuthName "SITIO RESTRINGIDO!"
Require valid-user
```

Finalmente es necesario editar un archivo de configuración de Apache para que se habilite la autenticación:

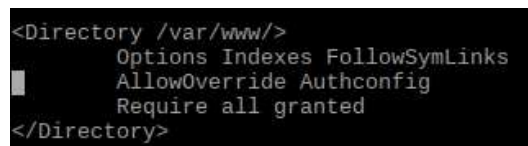
Ingresamos el comando:

```
sudo nano /etc/apache2/apache2.conf
```

Se cambia luego de: <Directory /var/www/>, la línea:

AllowOverride None por AllowOverride Authconfig

Se presiona control x, S y enter para guardar y salir.



```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride Authconfig
    Require all granted
</Directory>
```

Figura 71 Sección de archivo Apache 2

Se valida que se ha realizado correctamente la restricción de acceso reiniciando el servicio Apache y recargando la página en cualquier navegador web, lo primero que se nota es que aparece un cuadro de dialogo solicitando que se ingrese usuario y contraseña.

Ingresamos correctamente las credenciales para tener acceso a la página solicitada.

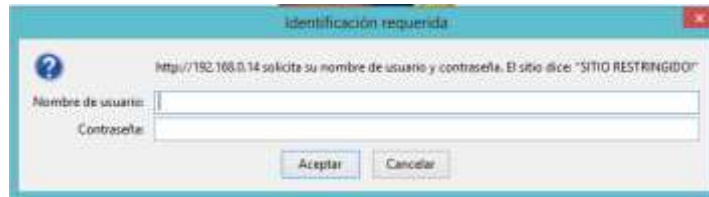


Figura 72 Cuadro de autenticación en Mozilla Firefox

### 3.4 INSTALACIÓN DE SOFTWARE DE SEGUNDO PLANO, UTILIDADES Y ACCESORIOS.

#### PUSHETTA

##### REGISTRÓ DE USUARIO EN PUSHETTA

Se ingresa al siguiente URL: <http://www.pushetta.com/accounts/signup/> y se crea una cuenta.

En Channel se elige la opción “Add a Channel” y se llena los datos acordes al proyecto considerando que esta información no puede ser editada una vez creado el canal.

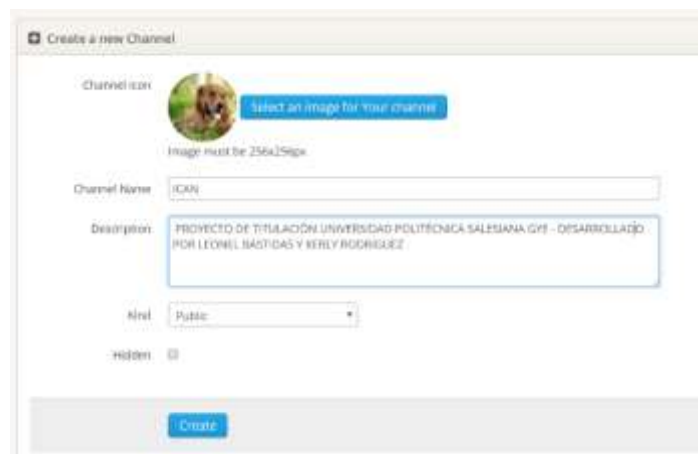


Figura 73 Pestaña de creación de nuevo canal en interfaz web PUSHETTA.

Se toma nota del nombre del canal, así como del API Key el cual es obtenido en la interfaz Dashboard de PUSHETTA.



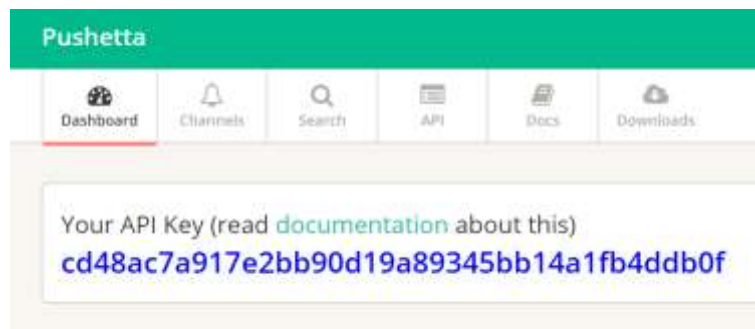


Figura 74 Código API Key de la APP Pushetta.

Fuente: <http://www.pushetta.com/my/dashboard/>

Las notificaciones PUSH pueden ser configuradas en dispositivos Android, iOS, Windows phone o como extensión del navegador Google Chrome.

<http://www.pushetta.com/pushetta-downloads/>

## HABILITACIÓN Y CONFIGURACIÓN DE APP PUSHETTA

En Google Play se busca “Pushetta” y se procede a instalar.



Figura 75 App Pushetta en Google Play

Fuente: <https://play.google.com/store/apps/details?id=com.gumino.pushetta>

Seleccionamos la APP, en “SUBSCRIPTIONS” se busca el nombre del canal previamente creado a través de la página web.

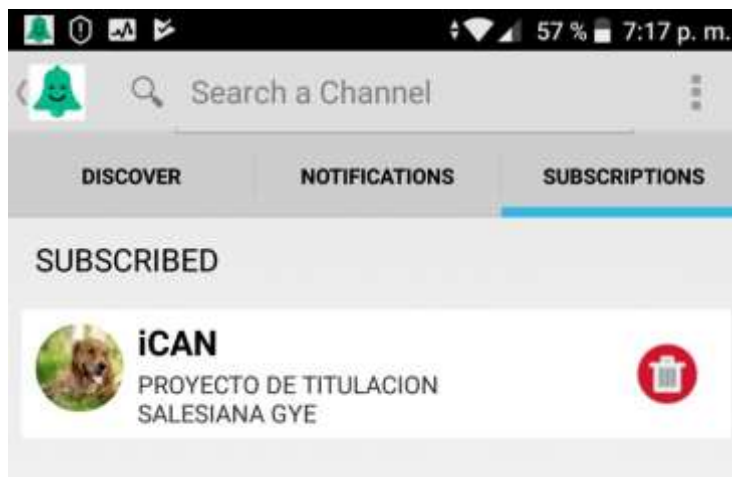


Figura 76 Canal suscrito en APP Pushetta

Con esta habremos vinculado nuestro servicio PUSH al móvil Android, esto puede ser verificado a través de las opciones del Channel por medio de la interfaz WEB, además se envía por medio de la casilla “Push a message” un mensaje inicial para constatar que se reciba la notificación en el móvil Android.

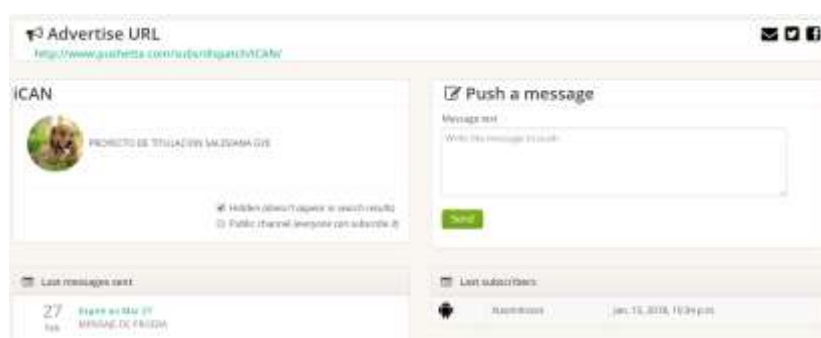


Figura 77 Opciones de canal en Pushetta

Fuente: <http://www.pushetta.com/my/channels/iCAN/#>

En el móvil Android registrado aparece la siguiente notificación:

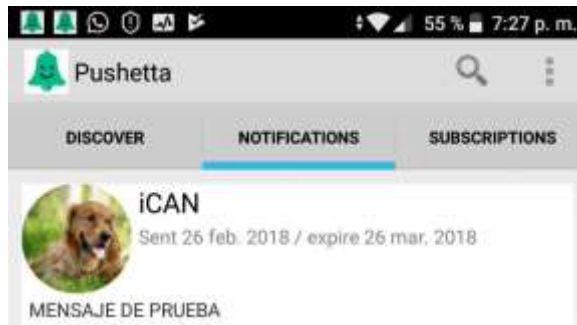


Figura 78 Notificación recibida en dispositivo Android

## INSTALACIÓN DE PUSHETTA EN RASPBERRY PI

En la siguiente URL se encuentra la documentación oficial de PUSHETTA para habilitarlo en distintas plataformas:

<http://www.pushetta.com/pushetta-docs/>

Para Raspberry PI inicialmente se instala la librería pushetta con el siguiente comando:

```
pip install pushetta
```

Una vez instalado solo es necesario crear un script .py con el siguiente contenido para empezar a recibir notificaciones.

```
from pushetta import Pushetta
API_KEY=" cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f"
CHANNEL_NAME="iCAN"
p=Pushetta(API_KEY)
p.pushMessage(CHANNEL_NAME, "Hello World")
```

A partir de este script se puede modificar acorde a los requerimientos del sistema.

## TEAMVIEWER

TeamViewer lanzó una versión soportada por el sistema operativo Raspbian, la misma puede ser descargada directamente desde el siguiente URL del sitio oficial:

<https://www.teamviewer.com/en/download/linux/#downloadAdditionalDownloads>



Figura 79 Descarga de software TeamViewer desde raspbian

## INSTALACIÓN

Luego de descargado se inicia sesión con una cuenta previamente creada para tener control y acceso de este:



Figura 80 Inicio de sesión de TeamViewer para raspbian.

Con esta herramienta se puede brindar soporte y asegurar el acceso remoto al equipo ya sea para gestionar actualizaciones o realizar algún cambio en programación si el usuario lo desea, el único requerimiento es tener una conexión activa hacia Internet.

## CRON

Cron es una utilidad de Raspbian que permite ejecutar acciones en el sistema acorde a horarios o eventos especificados.



Comando: `python /var/www/data_py/ACCESOS.py`

Se presiona Añadir y se verifica que se muestre de la siguiente manera:

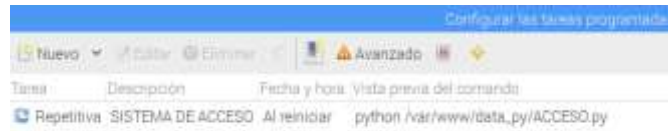


Figura 83 Tareas programadas en CRON

EL mismo procedimiento se realiza con el programa LUZ-DOSI.py.

AL finalizar cada vez que se reinicie la Raspberry se ejecutarán los dos programas que controlan la activación y ajuste de parámetros del sistema.

Al reiniciar se verifica que los programas estén activos a través del comando `ps aux | grep NAME`.



Figura 84 Resultados del comando psaux en ventana terminal

## PANTALLA TÁCTIL

Se integró una pantalla táctil de 7" Oficial de Raspberry PI al proyecto con la finalidad de disponer control embebido de todo el sistema.

En esta se visualizará la interfaz web por medio del navegador propio de Raspbian.

La pantalla contiene una tarjeta que facilita su conexión hacia el Raspberry PI por medio del puerto DSI, la alimentación se realiza por medio de una fuente de 5V independiente. Solo basta conectarla y encender el Raspberry para empezar a visualizar la imagen.



Figura 85 Test de conexión de pantalla táctil

## CONEXIÓN DE PANTALLA TÁCTIL



Figura 86 Diagrama de conexión de pantalla táctil a raspberry pi

## HABILITACIÓN DE INTERFAZ WEB EN MODO KIOSKO.

Para gestionar el sistema desde el Display ubicado al interior de la casa domótica es prescindible que la página local se abra en modo kiosco al arrancar el sistema.

Se configura la Raspberry Pi en modo kiosco de la siguiente manera:

Creamos un Fichero en el directorio LXDE-pi del usuario local:

```
sudo nano /home/iCAN/.config/lxsession/LXDE-pi/autostart
```

Se copia el siguiente código de manera que se ejecute automáticamente la aplicación chromium en modo kiosco con el url especificado.

```
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
@xscreensaver -no-splash
@point-rpi
@sed -i 's/"exited_cleanly": false/"exited_cleanly": true/' ~/.config/chromium/$
@chromium-browser --noerrdialogs --kiosk file:///var/www/html/index.html
```

## CÁMARA

Se hace uso de una cámara RASPI NOIR V2.1 instalada al interior de la casa en un encapsulado que la protege. Está conectada directamente al Raspberry PI por medio del puerto DSI designado en la placa.

La ventaja de este tipo de cámaras contra las USB es que no exigen mucho procesamiento por parte del GPU al estar conectadas directamente al mainboard.

## HABILITACIÓN CÁMARA

Para habilitar la Cámara en el sistema y validar su funcionamiento realizamos el siguiente paso:

En la pestaña Interfaces de la ventana de Configuración de Raspberry PI se activa la opción “Cámara”.

Se apaga el sistema, se procede a conectar la cámara de manera correcta y se enciende el sistema.

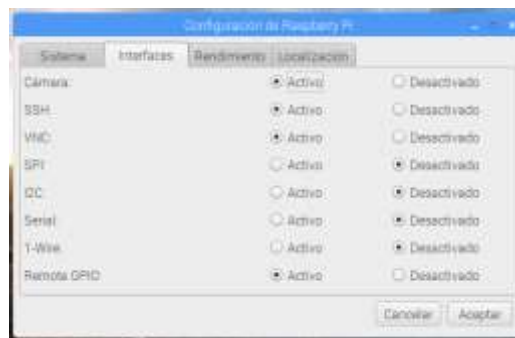


Figura 87 Habilitación de cámara en menú interfaces de la configuración de raspberry pi



En una ventana terminal se ejecuta el siguiente comando para constatar que se ha realizado exitosamente la instalación de la cámara.

```
raspistill -o test.jpg
```

El resultado será la captura fotográfica de una imagen de nombre test en el directorio actual. Con esta imagen se verifica el buen funcionamiento de la cámara.



Figura 88 Test de cámara



Figura 89 Imagen capturada por cámara en el sistema

## CONEXIÓN DE CÁMARA AL SISTEMA

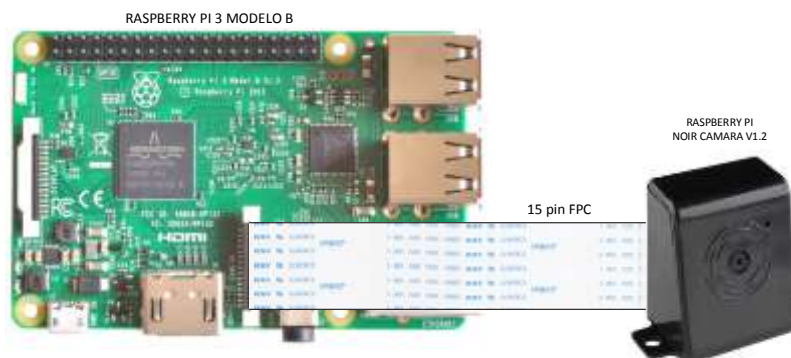


Figura 90 Diagrama de conexión de cámara a raspberry pi

### 3.5 CONFIGURACIÓN DE ACCESO Y GESTIÓN REMOTA A TRAVÉS DE RED

#### CONFIGURACIÓN ESTÁTICA DE IP

Se valida en la configuración DHCP de la red WIFI el Pool de asignación de IP, de esta manera se configura la Raspberry Pi con una IP estática fuera de este rango para evitar errores de duplicidad de IP.

Para el desarrollo de esta tesis se valida que el Pool de direcciones IP va de:

192.168.1.2 – 192.168.1.200



Figura 91 Configuración DHCP de router Huawei propiedad de CNT

Como estándar procederemos a configurar nuestra Raspberry con la última dirección disponible de la red, en este caso: 192.168.1.254

En el símbolo Wifi de la esquina superior derecha del escritorio Raspbian se selecciona la opción Wireless & Wired Network Settings.



Figura 92 Menú de configuraciones de red

En Configure se selecciona Interface – wlan0. Luego se llena los espacios vacíos con el direccionamiento IP definido y se da clic en Aplicar. Se cierra y se reinicia el sistema para cargar la configuración actual.



Figura 93 Configuraciones de red

## CLIENTE NO-IP

Debido a que la mayoría de los proveedores ISP provee una dirección IP pública dinámica es necesario registrarnos en un dominio público que puede ser gratis o pagado con la finalidad de que dinámicamente este dominio aprenda la IP pública asignada por parte del ISP a través de un cliente que debe estar dentro de nuestra red LAN local. Otra opción es contratar a través del ISP una dirección IP pública estática sin embargo esto representa un costo.

El desarrollo del presente proyecto hizo uso del software y dominio de NO-IP del siguiente URL: <https://www.noip.com/>

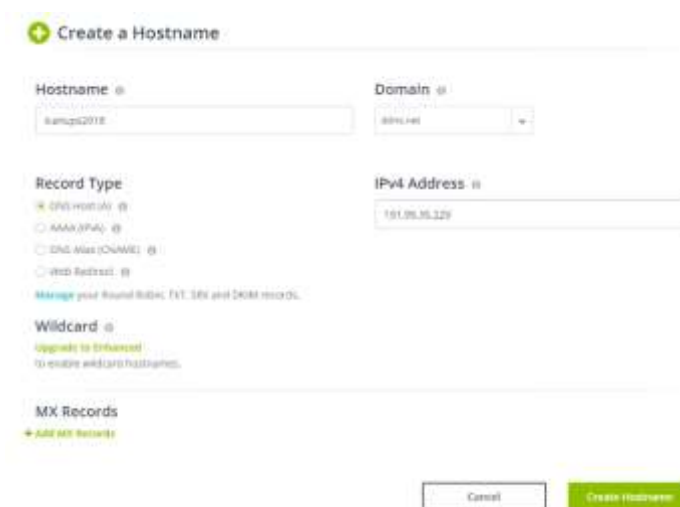
El cliente conectado a nuestra red local que envía las actualizaciones es el Raspberry PI.

## CREACIÓN DE CUENTA Y REGISTRO DE DOMINIO

Se crea una cuenta en my.noip.com llenando los campos especificados en la pestaña de inicio.

En el Menu principal se selecciona “Hostnames” y luego “Crear Hostname”. En la ventana emergente se llenan los campos solicitados.

Para el desarrollo del presente proyecto se registró el dominio: icanups2018.ddns.net



The screenshot shows a web form titled "Create a Hostname". It has several input fields and a dropdown menu. The "Hostname" field contains "icanups2018". The "Domain" dropdown menu is set to "ddns.net". Under "Record Type", the "DNS Hostname" option is selected. The "IPv4 Address" field contains "191.25.25.229". There are also sections for "Wildcard" and "MX Records". At the bottom right, there are two buttons: "Cancel" and "Create Hostname".

Figura 94 Creación de dominio para servidor NO-IP

## INSTALACION DE CLIENTE NO-IP EN RASPBERRY PY

Vía SSH o desde la ventana de comandos se ingresa los siguientes comandos:

```
mkdir no-ip
cd no-ip
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxvf noip-duc-linux.tar.gz
cd noip-2.1.9-1/
make
sudo make install
```

Durante la instalación se pedirá ingresar la contraseña root, así como el usuario y password antes configurados por medio de la URL: <https://www.noip.com/>

Para que la Raspberry funcione modo cliente NO-IP es necesario que el programa se ejecute cada vez que se reinicie el sistema. Con este fin se crea el siguiente fichero:

```
sudo nano /etc/init.d/noip2
```

En él se agrega el siguiente contenido:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: Servicio No-IP
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: arranque automatico para no-ip
# Description:
#
### END INIT INFO
sudo /usr/local/bin/noip2
```

Finalmente se guarda el archivo, se otorga permisos de ejecución y se coloca en cola de arranque.

```
sudo chmod +x /etc/init.d/noip2
sudo update-rc.d noip2 defaults
```

## **APERTURA DE PUERTOS EN ROUTER**

Para tener acceso remoto hacia la interfaz web instalada en nuestra Raspberry PI se requiere abrir puertos en nuestro Router de borde final.

Para tener total gestión de nuestra Raspberry PI a través de los puertos necesarios se coloca la IP previamente configurada 192.168.1.254 en la zona DMZ del router.

Si no se desea abrir todos los puertos sino solamente los requeridos, se puede realizar port forwarding de los puertos 80 (HTTP), 22 (SSH) y 5900-5910 (VNC) hacia la Ip 192.168.1.254 que previamente hemos habilitado en la Raspberry Pi 3.



Figura 95 Configuración Dmz en router Huawei de proveedor Cnt.

Bajo estas configuraciones se permite acceder vía WEB al servicio HTTP levantado en el Raspberry PI, además se permite la conexión SSH y VNC remotamente a través del dominio creado.

## PRUEBAS DE ACCESO REMOTO

```
C:\Users\LeonelAndrés>ping icanups2018.ddns.net

Haciendo ping a icanups2018.ddns.net [191.99.35.229] con 32 bytes de datos:
Respuesta desde 191.99.35.229: bytes=32 tiempo<1m TTL=64
Respuesta desde 191.99.35.229: bytes=32 tiempo<1m TTL=64
Respuesta desde 191.99.35.229: bytes=32 tiempo<1m TTL=64
Respuesta desde 191.99.35.229: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 191.99.35.229:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Figura 96 Ventana CMD realizando ping hacia dominio creado.



Figura 97 Visualización de interfaz web iCAN en Chrome desde host de red externa.

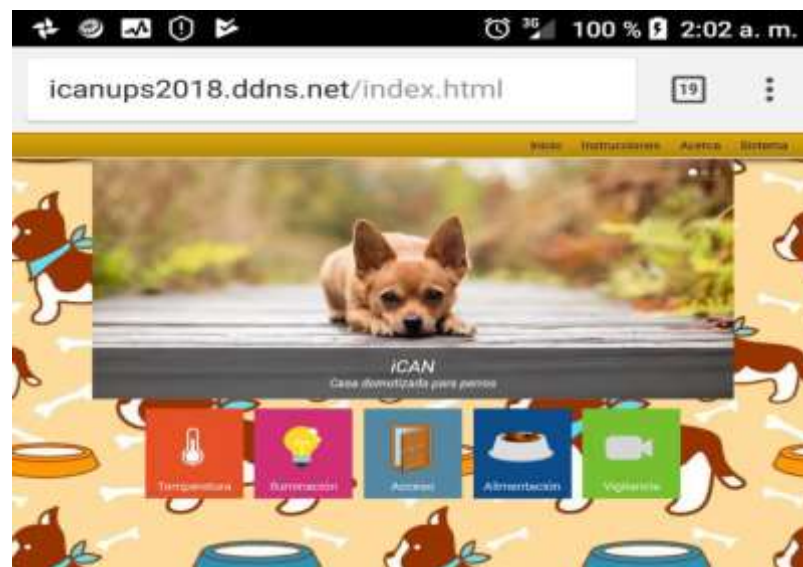


Figura 98 Acceso a interfaz web ICAN desde teléfono Android con red móvil CLARO

## 3.6 DESARROLLO DE APLICACIONES DEL MENÚ PRINCIPAL

### 3.6.1 SISTEMA DE ALIMENTACIÓN

Para asegurar la correcta alimentación de los canes en un horario establecido y en una proporción adecuada se desarrolló un sistema de alimentación automatizado el cual sirve una porción de alimento al can en hasta tres intervalos de tiempo predefinidos. La porción de alimento a dispensar es regulada y personalizada dependiendo de la raza o dieta que se requiera en hasta 3 proporciones.

### 3.6.1.1 DISEÑO DEL DOSIFICADOR

#### ESTRUCTURA

El diseño de la estructura se realizó tomando en consideración el área donde será ubicado el dosificador, el tipo de alimento a contener, asegurando facilidades de mantenimiento y reposición del alimento.

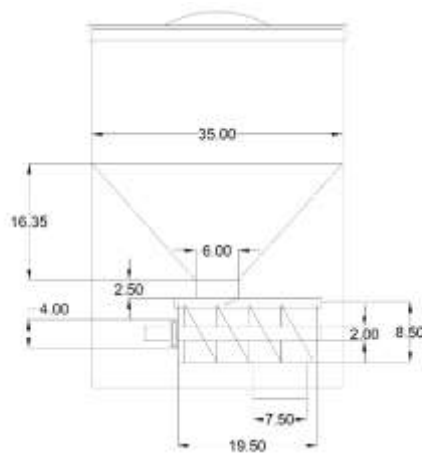


Figura 99 Vista frontal del dosificador

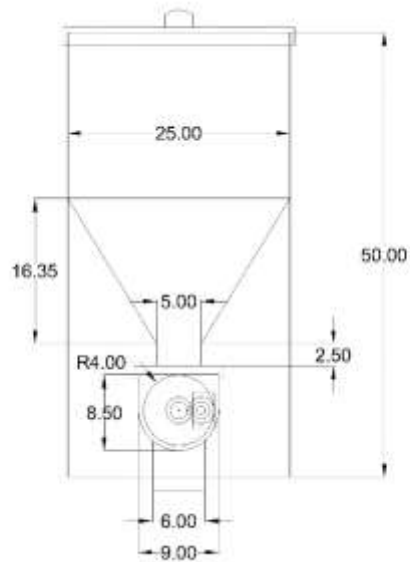


Figura 100 Vista lateral del dosificador



## **MATERIAL DEL DOSIFICADOR**

La construcción del dosificador se realizó con acero Inoxidable, debido a su resistencia a la corrosión, su resistencia al fuego, su estética, fácil limpieza y mantenimiento. Estas propiedades considerando que se trata de un envase que contendrá alimento lo convierten en el material ideal.

## **CAPACIDAD**

El dosificador está diseñado para contener un volumen aproximado de 20000 cm<sup>3</sup> de comida para mascotas. Lo suficiente para asegurar su alimentación por al menos 3 días, dependiendo de la cantidad de comida que el perro requiera.

## **TIPO DE ALIMENTO**

El diseño fue pensado para que contenga alimentos sólidos.

Se realizó pruebas exitosas usando croquetas de alimentos para perros de marcas comerciales tanto para raza adulta, pequeña o mediana sin restricción de marca alguna.

## **ELEMENTOS DE ACCIONAMIENTO DEL DOSIFICADOR**

### **TOLVA**

Se hace uso de una tolva para contener el alimento y que direcciona el mismo hacia el dosificador de tornillo.

## **DOSIFICADOR DE TORNILLO**

Un eje giratorio tipo tornillo realiza el desplazamiento horizontal del alimento hasta la apertura de expulsión.

Modificando la velocidad de giro y el tiempo de rotación del eje se determina la cantidad de alimento que es dosificado.

## **MOTOR DC**

Un motor DC de 24 V de giro continuo y con inversión de giro es el encargado de realizar el movimiento del eje giratorio. Para el diseño planteado el giro se efectuará solo en sentido horario.

## **FUENTE DE PODER**

Se hace uso de una fuente de alimentación de 24 V a 2.5 A para alimentar el motor. En las pruebas realizadas se valida que el amperaje utilizado por los motores llega a 1 Amperio.

## **ELEMENTOS DEL SISTEMA**

### **MÓDULO L298**

Por medio del driver L298 se realiza el control PWM del motor, esto con la intención de ajustar la velocidad de giro y cantidad de alimento que el dosificador sirve.

## RASPBERRY PI 3

Recibe las instrucciones ingresadas por medio de la interfaz web, las procesa, recibe información de los sensores y ejecuta los procesos principales que controlan el sistema para interactuar con los elementos de salida.

### 3.6.1.2 CONTROL DEL SISTEMA

El control del sistema de alimentación se lleva a cabo a través de un proceso principal y dos procesos secundarios detallados a continuación:

#### PROCESO PRINCIPAL

Las opciones escogidas del submenú ALIMENTACIÓN en la interfaz web son registradas en el archivo HDOSIFICADOR.txt a través de una función PHP que interactúa entre la pantalla del cliente y el servidor.



Figura 101 Menú de alimentación

El submenú del botón ALIMENTAR se divide en tres opciones:

## SELECCIÓN DE PORCIÓN DE ALIMENTO

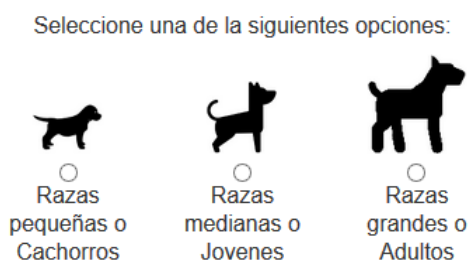


Figura 102 Selección de tipo de perro por raza o edad

## CONTROL DEL SISTEMA DE SELECCIÓN DE PORCIÓN

Desde el submenú Alimentación en la interfaz web se selecciona el tipo de perro a alimentar el cual está catalogado en tres diferentes porciones: Razas pequeñas o cachorros, Razas medianas o Jóvenes y Razas grandes o Adultos. Esta selección determinará qué cantidad de alimento se va a dosificar. La clasificación correcta de las porciones a dosificar se consultó y se recomendó por parte de un médico veterinario profesional.

La opción tomada es registrada en el archivo HSOSIFICADOR.txt y es leída por los programas Python LUZ-DOSI.py y DOSIMANUAL.py al ser ejecutados. Estos programas tienen predeterminados un tiempo y porcentaje de PWM para cada una de las tres opciones las cuales son cargadas en la ejecución dependiendo de la opción escogida.

## SERVIR PORCIÓN

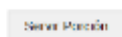


Figura 103 Botón Servir porción del menú alimentación

## CONTROL DE LA OPCIÓN “SERVIR PORCIÓN”

La selección de este botón en el submenú alimentación de la interfaz web ejecuta el script Python DOSI-MANUAL.py, este programa es el encargado de realizar la activación del puerto GPIO que va hacia el Driver L298 y que controla el giro del dosificador de tornillo. La elección tomada en la opción de selección de porción es leída y cargada durante la ejecución de este programa.

## DOSIFICACIÓN AUTOMÁTICA POR INTERVALOS DE TIEMPO



Figura 104 Selección de tiempo de dosificación automática

## CONTROL DE SISTEMA DE DOSIFICACIÓN AUTOMÁTICO

Esta es la función principal del sistema alimentación. En este menú ubicamos la selección de tres diferentes intervalos de tiempo como lo son Mañana, tarde y noche, cada intervalo es activado a través de un check y depende de su activación si se realiza o no la dosificación en el horario elegido. El valor de los tres check, así como el tiempo y minuto ingresado es guardado en el archivo HDOSIFICA.txt.

La activación de los puertos GPIO es llevado a cabo por parte del programa LUZ-DOSI.py

LUZ-DOSI.py es el programa raíz el cual valida constantemente si se ha efectuado un cambio en los parámetros del txt, Este mismo programa controla el sistema de Alimentación y de Luces automáticas. De detectar el cambio del primer valor en el documento HDOSIFICADOR.txt realizará una lectura de todas las variables del archivo y las reasignará a un proceso Python tipo thread que permite mantener en

ejecución un hilo del proceso totalmente independiente al programa principal, es decir en segundo plano, lo cual es conocido como multiprocesamiento. Este programa detecta si se ha habilitado la opción de servir en uno de los tres intervalos, de estar activado y coincidir la hora registrada con la hora local activa el respectivo puerto GPIO.

LUZ-DOSI.py está previamente configurado en el sistema a través de la herramienta CRON para que se ejecute al arranque del sistema.

## **PROCESOS SECUNDARIOS**

### **REPRODUCCIÓN DE SONIDO DURANTE ACTIVACIÓN DEL SISTEMA**

Con la finalidad de dar un aviso a la mascota de que se ha servido el alimento y de que el perro se familiarice con el sistema, en el instante en que se realiza la dosificación de alimento se reproduce un sonido previamente grabado en la memoria del circuito.

El sonido registrado recomendamos que sea un llamado con la voz del dueño del can indicando que se acerque a comer.

El circuito reproductor de sonido es un módulo ISD1820 ubicado sobre el dosificador, el mismo ha sido resguardado en una caja de resonancia de madera que contiene el parlante y que está ubicado en una zona de fácil acceso al usuario para permitir de manera sencilla grabar el sonido a reproducirse durante la ejecución del programa de dosificación.

La reproducción se lleva a cabo al recibir una señal de excitación por parte del Raspberry en el pin Play, la codificación que interactúa con el puerto GPIO se encuentra incluido en los scripts LUZ-DOSI.py y DOSI-MANUAL.py

### **NOTIFICACIÓN DE RECARGA DE ALIMENTO**

En el interior del dosificador se incluyó un sensor de distancia HC-SR04 con la finalidad de que mida el nivel de alimento que se encuentra al interior del envase dosificador. Este sensor está configurado para que luego de cada dosificación realice la medición y de un aviso mediante notificación PUSH al usuario cuando esté por debajo del rango especificado. De la misma manera enciende un LED sobre el contenedor para que visualmente podamos conocer que es necesario realizar una recarga de alimento en el contenedor.

La notificación PUSH se realiza a través del script Python push.py de PUSHETTA el cual envía un mensaje previamente establecido a todos los equipos registrados.

El control de este sistema se encuentra incluido en los scripts LUZ-DOSI.py y DOSI-MANUAL.py.

EL rango fijado en las pruebas realizadas determina que por debajo de los 20 cm se deberá activar la alarma.

### 3.6.1.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ALIMENTACION

Tabla 5 Puertos GPIO del sistema de alimentación

<b>SISTEMA:</b>	<b>ELEMENTOS DE ENTRADA</b>	<b>PUERTO ENTRADA</b>	<b>FUNCIÓN</b>
ALIMENTACIÓN	Sensor de distancia (HC-SR04)	TRIG – GPIO 9 ECHO – GPIO 11	Detecta nivel de alimento en tolva
	<b>ELEMENTOS DE SALIDA</b>	<b>PUERTO SALIDA:</b>	<b>FUNCIÓN</b>
	L298	GPIO 24	Activa motor dosificador de tornillo
	REPRODUCTOR SONIDO	GPIO 23	Reproduce sonido al dosificar
	LED	GPIO 25	Enciende led en caso de poco alimento

Fuente: Propia

### 3.6.1.4 CIRCUITO DE CONTROL

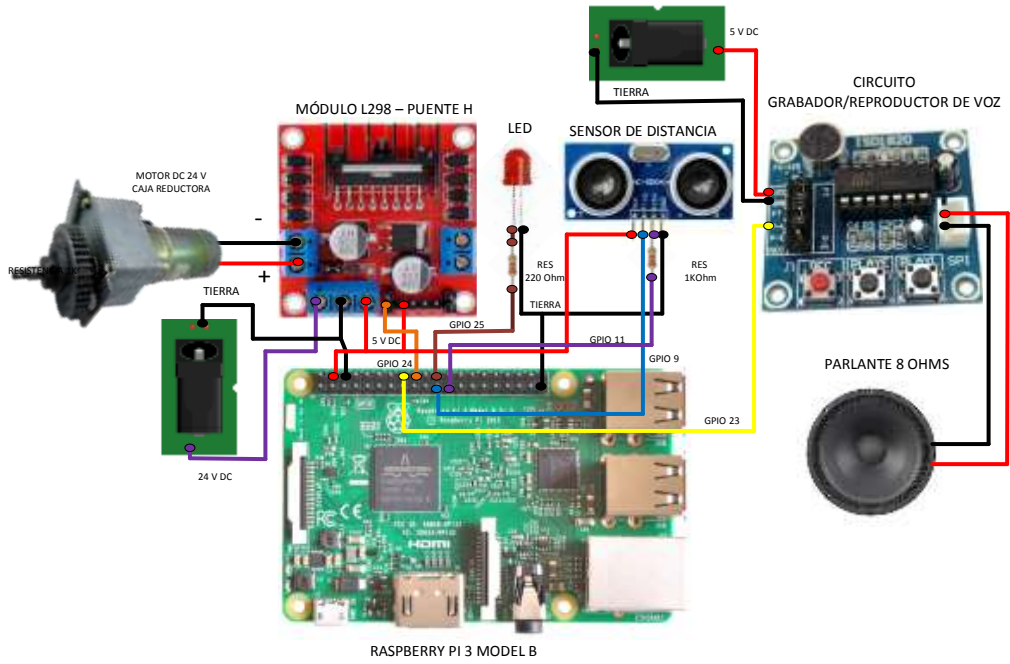


Figura 105 Circuito del sistema de alimentación

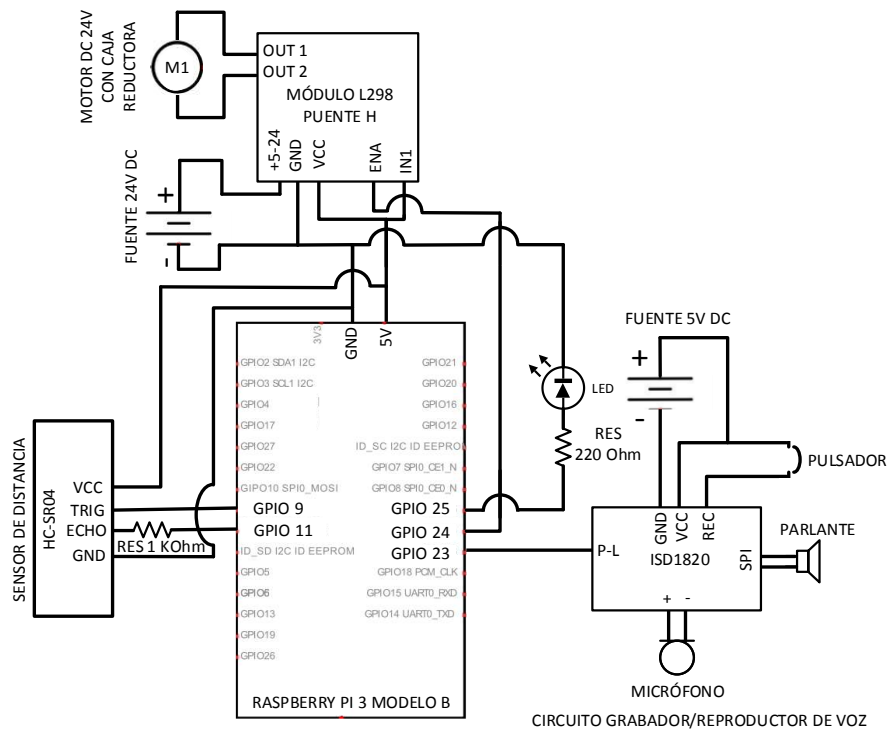


Figura 106 Diagrama de conexiones del sistema de alimentación



### 3.6.1.5 VARIABLES DEL PROCESO

Las variables que interactúan y registran los parámetros tomados de la interfaz web son las siguientes:

#### VARIABLES DE SELECCIÓN DE PORCIÓN

La selección de una de ellas asigna los siguientes valores:

Radius: p\_pequeño:

PWM: 50

Time: 2

Radius: p\_mediano:

PWM: 70

Time: 3

Radius: p\_grande:

PWM: 80

Time: 3

#### VARIABLES DE DOSIFICACIÓN AUTOMÁTICA

##### MAÑANA

Cuadro de selección: habilita que se sirva en la mañana, al seleccionarlo deberá cambiar el valor de 0 por 1.

**Seleccionado: asigna a variable  $A\_M = 1$**

**No seleccionado: asigna a variable  $A\_M=0$**

Hora: recoge la hora en que se servirá por la mañana, si está habilitado mañana actualiza este valor en el valor fijado.

**H\_M: valor de la hora**

**M\_M: valor del minuto**

## **TARDE**

Cuadro de selección: habilita que se sirva en la tarde, al presionar aceptar deberá cambiar el valor de 0 por 1.

**Seleccionado: asigna a variable A\_T = 1**

**No seleccionado: asigna a variable A\_T=0**

Hora: recoge la hora en que se servirá por la tarde, si está habilitado tarde actualiza este valor en el valor fijado.

**H\_T: valor de la hora**

**M\_T: valor del minuto**

## **NOCHE**

Cuadro de selección: habilita que se sirva en la noche, al presionar aceptar deberá cambiar el valor de 0 por 1.

**Seleccionado: asigna a variable A\_N= 1**

**No seleccionado: asigna a variable A\_N=0**

Hora: recoge la hora en que se servirá por la noche, si está habilitado noche actualiza este valor en el valor fijado.

**H\_N: valor de la hora**

**M\_N: valor del minuto**

## ACEPTAR

Recoge los valores de las variables ingresadas y las envía a **alimentacion.php**

**Alimentacion.php** escribe los valores en **hdosificador.txt**

Cada vez que se ejecuta **alimentacion.php** se reinicia el subprograma de control **cambiando el valor de AR = 1**

## DETALLE VARIABLES EN HDOSIFICADOR.txt

Este orden debe ser tal cual se describe en la tabla:

Tabla 6 Variables de HDOSIFICADOR.txt

0	AR	Cada vez que cambia a 1 se reinicia el programa de control
3	TIME	Registra el tiempo que permanecerá activado el dosificador
90	PWM	Se refiere al porcentaje de velocidad con la cual girará el dosificador
1	A_M	Si es 1 habilita leer la variable y servir en ese horario
18	H_M	Valor por guardarse para la hora en la mañana
16	M_M	Valor por guardarse para el minuto en la mañana
0	A_T	Si es 1 habilita leer la variable y servir en ese horario
01	H_T	Valor por guardarse para la hora en la tarde
29	M_T	Valor por guardarse para el minuto en la tarde
0	A_N	Si es 1 habilita leer la variable y servir en ese horario
02	H_N	Valor por guardarse para la hora en la noche
53	M_N	Valor por guardarse para el minuto en la noche

Fuente: Propia

### 3.6.1.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA

La ubicación del dosificador, caja de reproducción de sonido y diodo led se encuentra ubicado en la zona de alimentación. Esta ubicación fue estratégicamente diseñada para que el usuario tenga fácil acceso a través de la puerta lateral y pueda llenar el depósito, así como proceder con la grabación del sonido a reproducir.

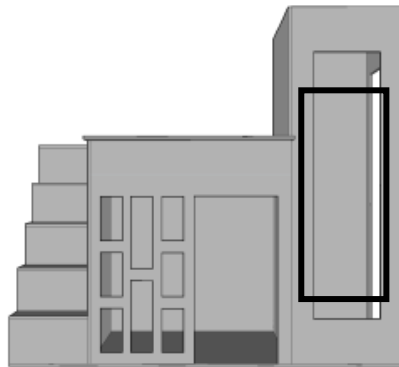


Figura 107 Ubicación referencial de elementos del sistema de alimentación



Figura 108 Ubicación del dosificador al interior de la casa

### 3.6.2 SISTEMA DE ILUMINACIÓN

El sistema de iluminación está compuesto por dos luces led de 18 W cada una las cuales se encienden en base a los parámetros elegidos por parte del usuario en la interfaz web.

Las luces pueden ser encendidas a través de un sensor de luz LDR, manualmente o por intervalo de tiempo.

### **3.6.2.1 DISEÑO DEL SISTEMA DE ILUMINACIÓN**

El sistema se desarrolló bajo las siguientes condiciones:

- Tener bajo consumo energético
- No deben generar calor.
- Fácil limpieza y mantenimiento
- Fácil reemplazo

## **ELEMENTOS DEL SISTEMA**

### **LUMINARIAS**

Cumpliendo las condiciones antes especificadas se eligió luz led para el sistema de iluminación. Se hace uso de 2 luminarias de 18W cada una. Ambas proporcionan suficiente luz al interior de la casa. Su activación es individual razón por la cual el usuario en modo manual puede elegir si encender una o ambas.

Las dos luminarias cuentan con su respectiva fuente de poder de fábrica cuya alimentación es de 110 V.

### **MÓDULO SENSOR DE LUZ LDR**

EL sensor de LUZ LDR es el encargado en modo automático de dar aviso al sistema que debe encender las luces. El sensor usado para la implementación del proyecto está

integrado a un módulo, el mismo que facilita su conexión y a través del cual se puede regular su sensibilidad.

La alimentación del sensor es realizada por parte de la fuente del Raspberry.

### **MÓDULO RELÉ DE 8 CANALES.**

Para activar o desactivar el encendido de las luces se hace uso de relés. Un módulo de 8 relés facilita la conexión de estos hacia el Raspberry. Es necesario hacer uso de relés ya que las luminarias trabajan a 110V.

La alimentación del módulo es realizada por medio de una fuente de 5V a 2.5 Amperios.

### **RASPBERRY PI 3**

Recibe las instrucciones ingresadas por medio de la interfaz web, las procesa, recibe información de los sensores y ejecuta los procesos principales que controlan el sistema para interactuar con los elementos de salida.

#### **3.6.2.2 CONTROL DEL SISTEMA**

El control del sistema de iluminación se realiza en base a las selecciones que realicemos en la interfaz web. Nuestro Raspberry PI será el encargado de recoger los valores del sensor, procesar las variables ingresadas en la interfaz web y activar o desactivar los puertos GPIO correspondientes según sea el caso.

## MENÚ DE ILUMINACIÓN: SELECCIÓN DE MODO

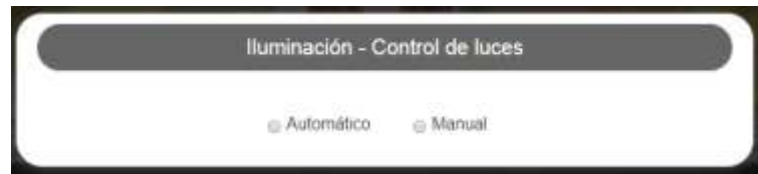


Figura 109 Menú del sistema de iluminación

En la primera ventana se debe elegir el modo de control de Iluminación, este puede ser Automático o Manual.

## MODO AUTOMÁTICO



Figura 110 Submenú de modo automático

Al seleccionar modo automático se desplegarán dos opciones: Sensor y Horario

## SENSOR

Este control se lleva a cabo por medio del programa python LUZ-DOSI.py el cual a través de las variables guardadas en un archivo txt habilita las interrupciones del puerto de entrada GPIO 17 donde se conecta el sensor. Cada cambio de estado en este puerto

llama a la clase Automático el cual cambia de estado los puertos GPIO de salida que controlan los relés que encienden y apagan las luces.



Figura 111 Selección de modo automático sensor

## **HORARIO**

Al seleccionar Horario se despliegan dos opciones:

### **DESDE**

Usando las flechas de desplazamiento se ingresa la hora en que se desea encender las luces.

### **HASTA**

Usando las flechas de desplazamiento se ingresa la hora en que se desea apagar las luces.

Al presionar OK estos valores son guardados en el archivo LUCES.txt.



El programa raíz LUZ-DOSI.py detecta la selección realizada y activa la clase thread Iluminación. Esta clase compara minuto a minuto el rango de tiempo seleccionado con la hora actual para encender y apagar las luces acordes a lo programado.



Figura 112 Selección de modo automático horario

## MODO MANUAL

En el modo Manual podemos realizar el encendido o apagado de manera independiente de las dos luminarias.

Ambiente 1 hace referencia a la zona de descanso, Ambiente 2 en cambio se refiere a la zona de alimentación.

Al seleccionar ON / OFF en esta interface llamaremos al programa LUZ\_MANUAL.py el cual se encarga de activar o desactivar los puertos GPIO acorde a lo seleccionado, además desactiva la ejecución de la clase thread Iluminación. El detalle del funcionamiento se encuentra ubicado en: variables del sistema de iluminación.



Figura 113 Submenú de modo manual iluminación

### 3.6.2.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ILUMINACIÓN

Tabla 7 Puertos GPIO del sistema de iluminación

SISTEMA	ELEMENTO DE ENTRADA	PUERTO ENTRADA	FUNCIÓN
ILUMINACIÓN	MÓDULO SENSOR LDR	GPIO 17	DETECTAR LUZ / OSCURIDAD
	ELEMENTOS DE SALIDA	PUERTO SALIDA:	FUNCIÓN
	RELE	GPIO 27	ACTIVA LAMP 1
	RELE	GPIO 22	ACTIVA LAMP 2

Fuente: Propia

### 3.6.2.4 CIRCUITO DE CONTROL DEL SISTEMA

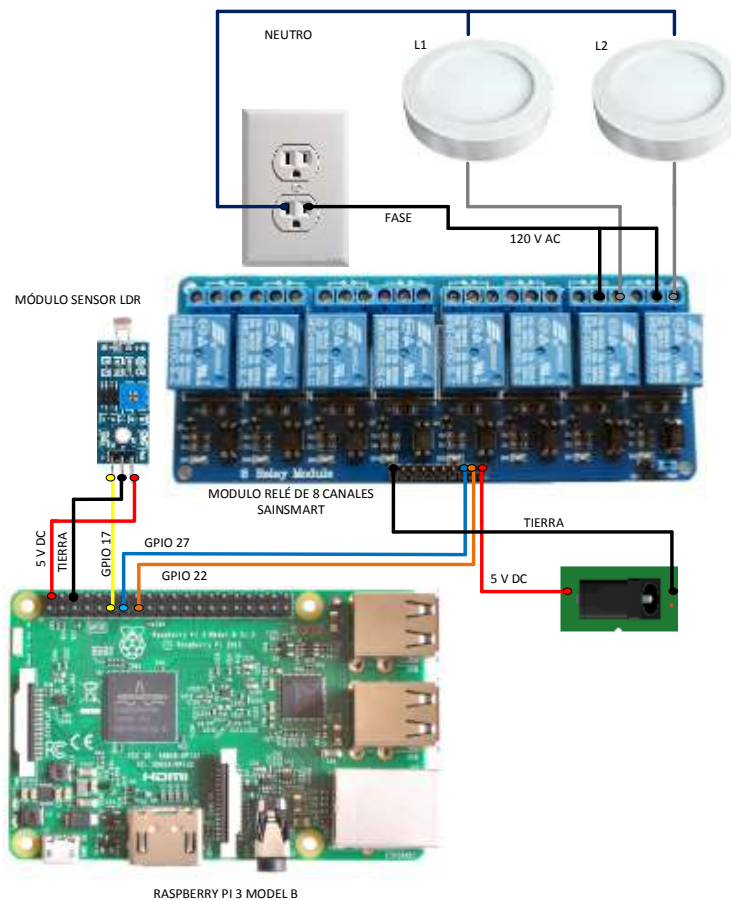


Figura 114 Circuito del sistema de iluminación

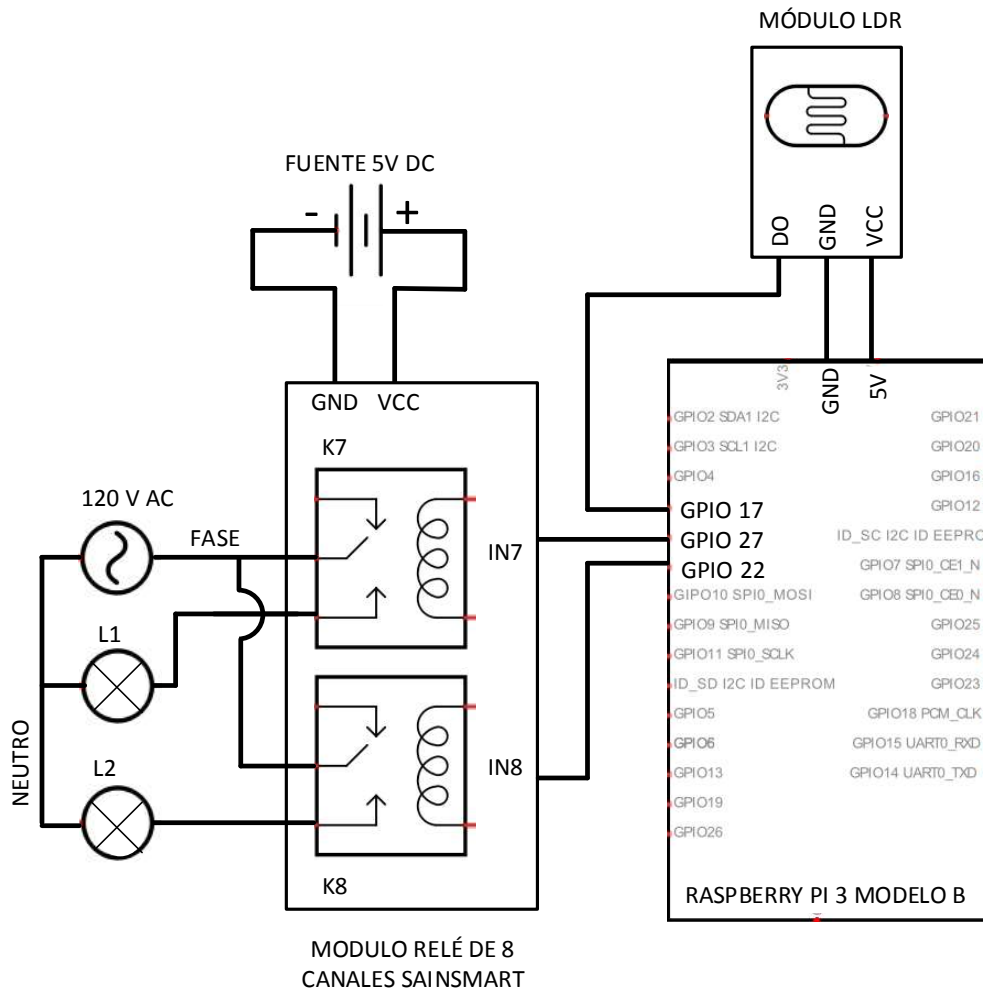


Figura 115 Diagrama de conexiones del sistema de iluminación

### 3.6.2.5 VARIABLES DEL SISTEMA DE ILUMINACIÓN

#### **MODO AUTOMÁTICO:**

Cambia la variable I\_M por 0

#### **SENSOR:**

Cambia la variable I\_MA por 0

## **OK**

Llama al programa iluminacions.php el cual escribe los valores antes detallados en LUCES.txt

Cada vez que se ejecuta iluminacions.php se reinicia el subprograma de control cambiando el valor de IR = 1

## **HORARIO**

Cambia la variable I\_MA por 1

## **DESDE**

Recoge el valor y lo guarde en variables para que se enciendan en la hora indicada.

**H\_H: valor de la hora**

**M\_H: valor del minuto**

## **HASTA**

Recoge el valor y lo guarde en variables para que se apaguen en la hora indicada.

**H\_L: valor de la hora**

**M\_L: valor del minuto**

## **OK**

Recoge los valores de tiempos ingresados, llama al programa iluminacionh.php el cual escribe los tiempos y valores antes detallados en LUCES.txt

Cada vez que se ejecuta iluminacionh.php se reinicia el subprograma de control cambiando el valor de IR = 1.

## **MODO MANUAL**

### **LUCES AMBIENTE 1**

#### **ON**

Ejecuta ilumanual.php el cual desactiva el modo automático cambiando los valores en el txt y llama al programa python LUZ-MANUAL.py con argumento 1.

#### **OFF:**

Ejecuta ilumanual.php el cual desactiva el modo automático cambiando los valores en el txt y llama al programa python LUZ-MANUAL.py con argumento 2.

### **LUCES AMBIENTE 2**

#### **ON**

Ejecuta ilumanual.php el cual desactiva el modo automático cambiando los valores en el txt y llama al programa python LUZ-MANUAL.py con argumento 3.

#### **OFF**

Ejecuta ilumanual.php el cual desactiva el modo automático cambiando los valores en el txt y llama al programa python LUZ-MANUAL.py con argumento 4.

#### **OK**

No realiza ninguna acción, las acciones se ejecutan directamente al ser seleccionadas, cierra pestaña.

## DETALLE DE VARIABLES EN LUCES.TXT

El orden debe ser tal cual se describe en la tabla:

Tabla 8 Variables de LUCES.txt

1	IR	Si es 1 manda reiniciar el programa de control
1	I_M	Si es 1 modo manual si es 0 modo automático
1	I_MA	Si es 0 modo sensor y si es 1 modo horario
20	H_H	Hora de encendido
18	M_H	Minuto de encendido
05	H_L	Hora de apagado
12	M_L	Minuto de apagado

Fuente: Propia

### 3.6.2.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA

#### LUMINARIAS

Ambas luminarias son sobrepuestas, una se encuentra en la zona de descanso, La otra luminaria está ubicada en la zona de alimentación.

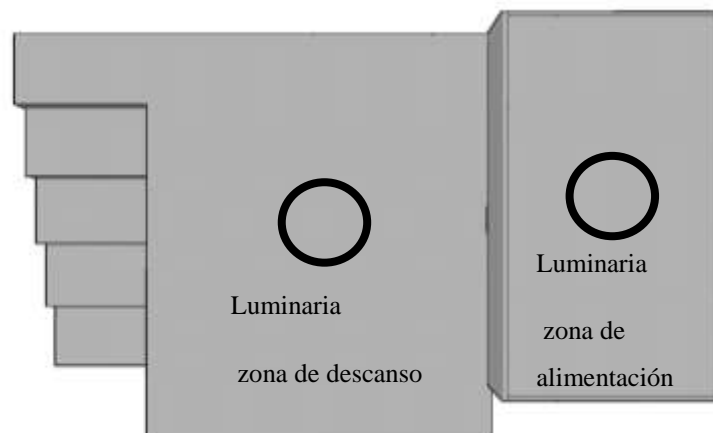


Figura 116 Ubicación referencial de luminarias del sistema



Figura 117 Luminarias instaladas en zona de alimentación y descanso

## SENSOR LDR

El sensor LDR se encuentra ubicado en el lado izquierdo de la casa, sobre las escaleras pegado a la pared exterior de la zona de descanso. El mismo se encuentra dentro del techo falso, sin embargo, se realizó una perforación y colocación de acrílico en su parte exterior para que detecte luz del sol.

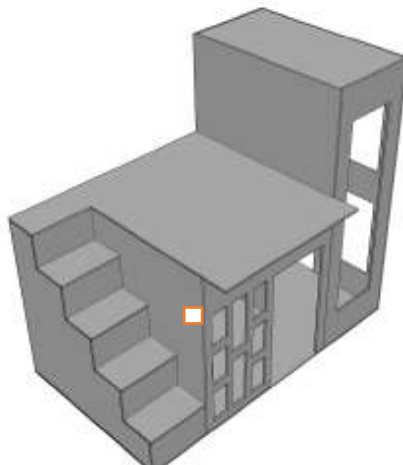


Figura 118 Ubicación del sensor LDR



Figura 119 Sensor LDR conectado al sistema

### **3.6.3 SISTEMA DE ACCESO**

Por medio de una puerta deslizante automática se permite o no el ingreso del perro hacia el interior de la casa.

Permite evitar el ingreso de otros animales o daños de la intemperie en el interior cuando la casa no está en uso, además en conjunto al sistema de ventilación mantiene un ambiente y temperatura agradable en el interior cuando la casa se encuentra cerrada.

Puede controlarse de forma manual o a través de sensores que detectan al perro y abren la puerta al querer ingresar o salir del mismo, de la misma manera se cierra cuando no detecta movimiento alguno.

#### **3.6.3.1 DISEÑO DE PUERTA AUTOMÁTICA**

##### **ESTRUCTURA**

Las dimensiones de la estructura de la puerta fueron en base a la apertura del ingreso de la casa que se ubicó en su parte frontal. La puerta será deslizante en el interior y al estar abierta permanece oculta por la pared frontal de la zona de descanso.

Se desliza a través de un riel sujeto en la parte superior de la cara interior de la pared frontal, de igual manera en la parte inferior posee un riel que hace de guía para el mecanismo.

##### **MATERIAL DE LA PUERTA**

La puerta está elaborada por un vidrio de 6 mm de espesor con marco de aluminio.



## **ELEMENTOS DEL MECANISMO UTILIZADO**

### **CREMALLERA**

Por medio de una cremallera sujeta a la parte superior de la puerta se realiza el movimiento horizontal que desplaza en sentido izquierdo y derecho la puerta.

### **MOTOR**

Un motor de 24 v DC con caja reductora es el encargado de transferir el movimiento circular del engranaje ubicado en su extremo hacia la cremallera provocando un movimiento lineal.

### **FUENTE DE PODER**

Una fuente de 24 DC a 2.5 Amperios es el encargado de energizar el motor durante su activación.

## **ELEMENTOS DEL SISTEMA**

### **FINALES DE CARRERA**

Están ubicados en los extremos del carril izquierdo y derecho de la puerta, su función es detectar cuando la puerta se abre o cierra en su totalidad. Al contacto con la puerta se presiona el interruptor del final de carrera, lo cual envía la señal al Raspberry que desactiva el motor.

## **SENSOR PIR**

Su función es la de detectar la presencia de la mascota al ingreso y salida de la casa lo cual activa la apertura de la puerta.

Son dos sensores uno ubicado en la parte exterior, sobre la puerta, y otro al interior, sobre la puerta. Su rango de visibilidad está direccionado al área de interés para evitar falsas alarmas.

## **MÓDULO L298**

A través de su circuito permite realizar el control de la inversión de giro del motor acorde al puerto que se active en el Raspberry PI. Abre o cierra la puerta en base al sentido de giro del motor.

## **RASPBERRY PI 3**

Recibe las instrucciones ingresadas por medio de la interfaz web, las procesa, recibe información de los sensores y ejecuta los procesos principales que controlan el sistema para interactuar con los elementos de salida.

### **3.6.3.2 CONTROL DEL SISTEMA**

El control del sistema se lleva a cabo a través de un proceso en ejecución sobre la Raspberry PI, el programa Python ACCESO.py es el encargado de detectar las instrucciones recibidas por parte del usuario en la interfaz web y ejecutar las secuencias respectivas de apertura o cierre de la puerta, además en caso de falla de luz o interrupción de corriente se ha considerado una secuencia en la programación que abre la puerta inmediatamente a través del respaldo de energía propio del sistema.

## MENU DE ACCESO

Al seleccionar el botón de acceso en el menú principal se muestra una ventana emergente con las opciones del control de acceso, las cuales son 3: Abierto, Cerrado y automático.

El accionamiento de la puerta es inmediato en base a la selección que se tome.



Figura 120 Menú del sistema de acceso

La elección que se tome es detectada por el programa acceso.php el cual registra el valor de la variable seleccionada en un archivo txt llamado ACCESO.txt, estos valores son leídos e interpretados por el programa Python ACCESO.py para ejecutar las instrucciones de apertura o cierre de la puerta.

El modo automático se lleva a cabo por medio de una clase tipo Thread incluida en el código de manera que no se interrumpe el proceso principal del programa.

El código está dispuesto de tal manera que durante el cierre de la puerta en caso de que se detecte movimiento o presencia del perro se interrumpe el cierre y se abra por un tiempo determinado antes de volver a intentar cerrarse.

## PROTECCIÓN CONTRA FALLAS DE LUZ

Para evitar que durante una interrupción de energía o corte de luz el perro quede encerrado al interior de la casa, se consideró una función dentro del código Python que censa el estado de un puerto GPIO determinado. Mientras este puerto se encuentre energizado se llevará a cabo el proceso principal del programa, en caso de que no se

detecte el puerto activo de inmediato se solicitará al sistema que realice la apertura de la puerta.

Esta acción se realiza a través del respaldo de energía UPS que posee integrado todo el sistema de control, la cual da una autonomía de aproximadamente 40 minutos.

Al restablecerse la energía el programa retoma su proceso principal y continúa funcionando en base a la última elección tomada por el usuario.

### 3.6.3.3 PUERTOS GPIO ASIGNADOS DEL SISTEMA DE ACCESO

Tabla 9 Puertos GPIO del sistema de acceso

SISTEMA	ELEMENTOS DE ENTRADA	PUERTO ENTRADA	FUNCIÓN
ACCESO	Sensor PIR interno	GPIO 5	Detecta presencia
	Sensor PIR externo	GPIO 6	Detecta presencia
	Final de carrera ABRE	GPIO 20	Detecta apertura total de la puerta
	Final de carrera CIERRA	GPIO 21	Detecta cierre total de la puerta
	Estado corriente, circuito de protección	GPIO 0	Detecta corte de energía
	<b>ELEMENTOS DE SALIDA</b>	<b>PUERTO SALIDA</b>	<b>FUNCIÓN</b>
	L298	GPIO 19	Activa la apertura de la puerta
GPIO 26		Activa el cierre de la puerta	

Fuente: Propia

### 3.6.3.4 CIRCUITO DE CONTROL DEL SISTEMA

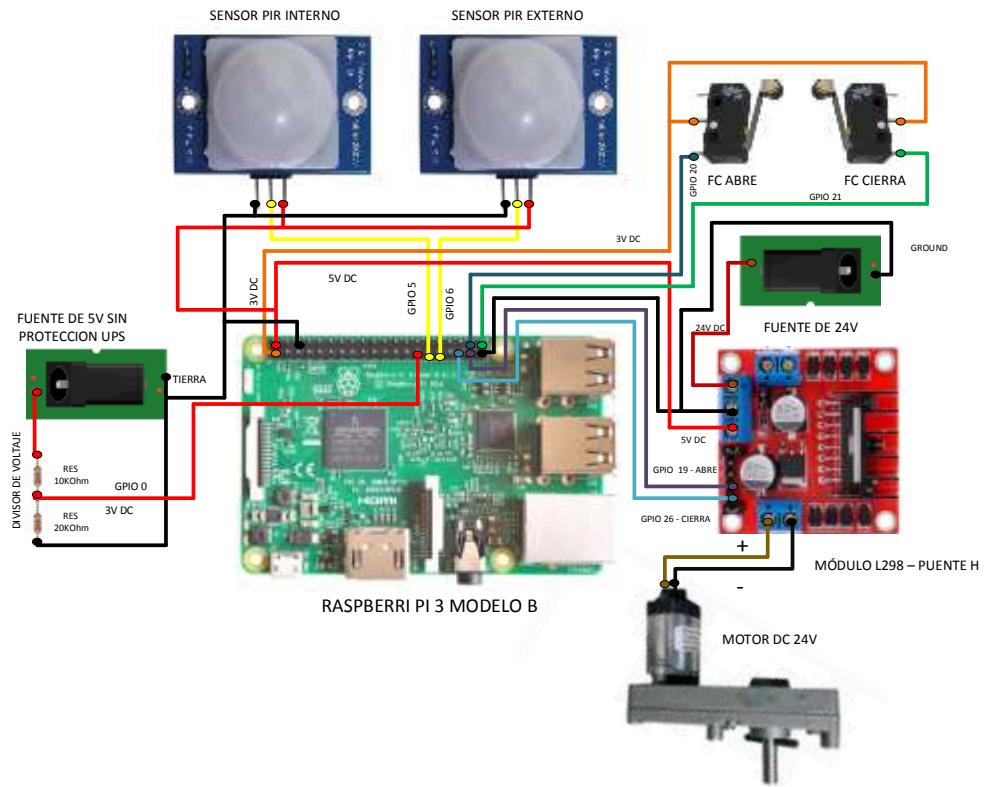


Figura 121 Circuito del sistema de acceso

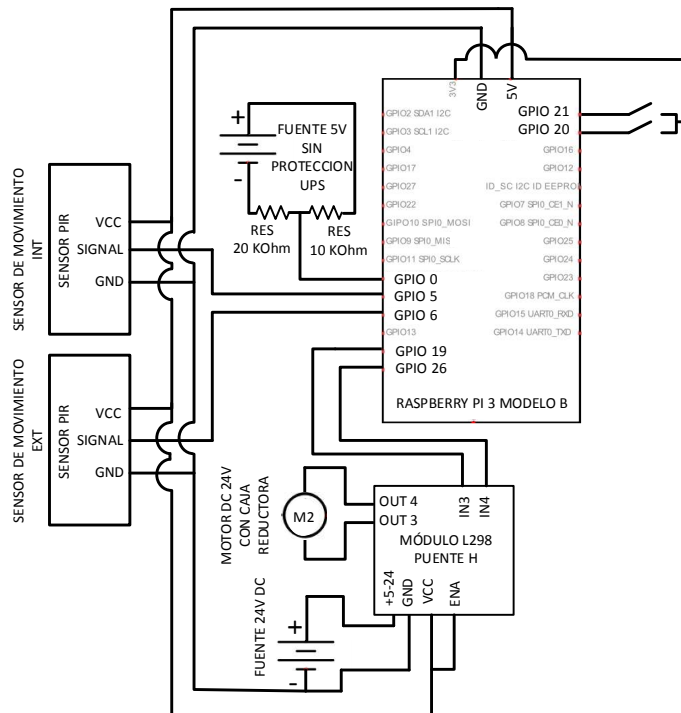


Figura 122 Diagrama de conexiones del sistema de acceso

Fuente: Propia

### 3.6.3.5 VARIABLES DEL SISTEMA DE ACCESO

#### ABIERTO

Le asigna a la variable P\_M el valor de 1

Manda a escribir en ACCESO.txt el valor de P\_M y cambia el valor de PR por 1

#### CERRADO

Le asigna a la variable P\_M el valor de 2

Manda a escribir en ACCESO.txt el valor de P\_M y cambia el valor de PR por 1

#### AUTOMÁTICO

Le asigna a la variable P\_M el valor de 3

Manda a escribir en ACCESO.txt el valor de P\_M y cambia el valor de PR por 1

PR se cambia al final para que al reiniciarse el programa tome el valor ya actualizado de P\_M.

#### DETALLE DE VARIABLES EN ACCESO.TXT

El orden debe ser tal cual se describe en la tabla:

Tabla 100 Variables de ACCESO.txt

PR	1	Indica un cambio en el txt al programa
P_M	1 – 2 – 3	Indica la selección realizada

Fuente: Propia

### 3.6.3.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA

La puerta está ubicada en la parte frontal de la zona de descanso, se desplaza horizontalmente de derecha a izquierda, dando paso u obstaculizado la entrada principal.

El motor está sujeto al interior del tumbado falso sobre la zona de descanso y a través de un orificio sobresale el engrane que se conecta a la cremallera para la transferencia de movimiento.

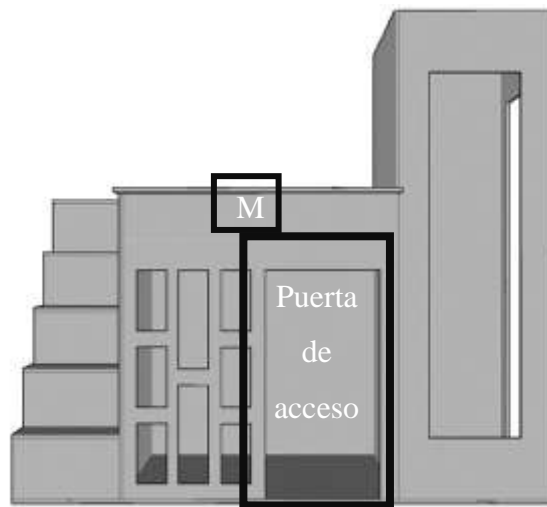


Figura 123 Ubicación referencial de puerta y motor del sistema de acceso



Figura 124 Engrane, cremallera, puerta y FC abre del sistema de acceso



Figura 125 FC Cierra, sensor PIR interno y externo del sistema de acceso

### **3.6.4 SISTEMA DE VENTILACIÓN**

El sistema de ventilación fue desarrollado bajo el concepto de mantener una temperatura y ambiente agradable al interior de la casa, especialmente en la zona de descanso siendo este el lugar donde el perro permanece la mayoría de tiempo.

El sistema posee tres extractores invertidos, alineados horizontalmente en la pared posterior de la zona de descanso, los mismos se encuentran protegidos bajo una caja de madera y rejilla metálica para prevenir que el perro tenga acceso directo a ellos y pueda lastimarse.

El control se realiza a través de un Arduino conectado al Raspberry PI por medio de la interfaz USB, las opciones elegidas a través de la interfaz web son enviadas vía serial hacia el Arduino.

#### **3.6.4.1 DISEÑO DEL SISTEMA DE VENTILACIÓN**

El sistema se desarrolló bajo las siguientes condiciones:

- Tener bajo consumo energético
- De fácil limpieza
- De fácil reemplazo



- No generar ruido
- Mantener flujo constante de aire

## **ELEMENTOS DEL SISTEMA**

### **EXTRACTORES**

Se hace uso de dos extractores de aire de 4" de diámetro y uno de 5" de diámetro. El de 5" está ubicado estratégicamente en el centro de la parte posterior de la zona de descanso y los de 4" uno a cada lado.

Están dispuestos de tal manera que extraigan aire de la parte posterior externa de la casa y lo ventilen hacia el interior.

Con los tres ventiladores encendidos se mantiene un flujo constante de aire, además la rejilla de protección que posee en su interior redirecciona este aire hacia la parte baja de la casa donde reposa el perro.

### **ARDUINO MEGA 2560 R3**

El Arduino Mega 2560 es el encargado de recibir y procesar la información que detecta el sensor de temperatura. A través del puerto USB se energiza y recibe instrucciones vía serial desde el Raspberry PI con las cuales cambia su función, de la misma manera es capaz de interpretar los datos del sensor y enviarlos vía serial hacia el Raspberry PI para que sean mostrados en la interfaz web.

El proceso principal que ejecuta el Arduino es independiente del Raspberry PI, sin embargo, en base a las variables que recibe procesa su código. Según la opción escogida el Arduino se encargará de habilitar o no el encendido de los ventiladores a través del módulo Relé.

De reiniciarse el Arduino por defecto cambiará su modo de trabajo a Automático.

## **MÓDULO SENSOR LM35**

Censa la temperatura del interior de la casa y la envía de manera análoga hacia el Arduino. Está ubicado estratégicamente bajo el marco que contiene la ventilación.

## **MÓDULO RELÉ DE 8 CANALES**

Para activar o desactivar el encendido de los extractores se hace uso de relés. Un módulo de 8 relés facilita la conexión de estos hacia el Arduino. Es necesario hacer uso de relés ya que los extractores trabajan a 110V.

La alimentación del módulo es realizada por medio de una fuente de 5v a 2.5 Amperios.

## **RASPBERRY PI 3**

Recibe las instrucciones seleccionadas a través de la interfaz web y las envía vía serial hacia el Arduino, de igual manera recibe el valor de la temperatura actual cada vez que el usuario ingrese al menú de temperatura para mostrarlo en la interfaz.

### **3.6.4.2 CONTROL DEL SISTEMA**

El control del sistema se lleva a cabo por el Arduino Mega, las instrucciones son recibidas desde el Raspberry PI vía serial y las ejecuta.

## MENÚ DE VENTILACIÓN: SELECCIÓN DE MODO

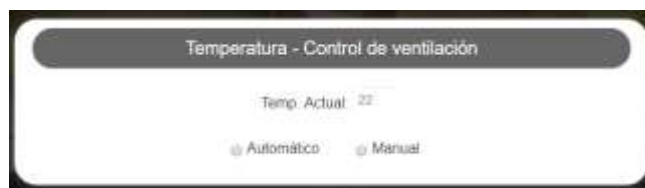


Figura 126 Menú del sistema de ventilación

Al dar clic sobre la opción Temperatura del Menú principal se ejecuta el programa temperatura.php, este programa realiza la comunicación de la interfaz web con el servidor indicándole al sistema que ejecute el programa Python TEMP\_SERIAL.py con argumento 400.

TEMP\_SERIAL.py envía vía serial el valor de 400 al Arduino.

El Arduino lee el parámetro recibido y solicita en tiempo real el valor de la temperatura que detecta el sensor. Este valor lo envía vía serial como respuesta al Raspberry.

TEMP\_SERIAL.py detecta el valor recibido y lo imprime como resultado.

Temperaturapy.php recoge el valor impreso por el código Python y lo comunica hacia la interfaz web.

Finalmente se abre una ventana emergente del menú de opciones del control de ventilación en la cual se muestra la Temperatura actual del interior de la casa.

Todo este proceso tarda alrededor de 1 a dos segundos.

## MODO AUTOMÁTICO



Figura 127 Submenú de modo automático

El Arduino Mega por defecto al encender se ubica en modo automático y activa los relés cuando detecta temperaturas superiores a 28°C.

## **OK**

En modo automático si se desea cambiar este parámetro se debe señalar usando las flechas de navegación el nuevo valor a partir del cual se debe activar la ventilación. Este valor debe estar comprendido entre 20 y 40°C.

Al presionar OK se ejecuta el programa temperaturaseteo.php, este programa envía por serial a través de TEMP\_SERIAL.py el valor ingresado en la interfaz web luego envía de igual manera el parámetro 100.

El Arduino recibe el valor ingresado en la interfaz y actualiza el umbral sobre el cual debe activar los relés que encienden la ventilación. Luego recibe el parámetro 100 para activar el modo automático.

## **CANCELAR**

Cierra la ventana del menú sin realizar cambio alguno

## **MODO MANUAL**



Figura 128 Submenú de modo manual

## **ON**

Al seleccionar On se ejecuta el programa temperaturaon.php, este programa envía por serial a través de TEMP\_SERIAL.py el valor de **200**.

El Arduino al recibir el valor 200 desactiva el modo automático y activa los relés que encienden la ventilación.

## **OFF**

Al seleccionar Off se ejecuta el programa temperaturaoff.php, este programa envía por serial a través de TEMP\_SERIAL.py el valor de **300**.

El Arduino al recibir el valor 300 desactiva el modo automático y los relés para apagar la ventilación.

## **SALIR**

Cierra la pestaña del menú y no realiza cambio alguno

### **3.6.4.3 PUERTOS DE ARDUINO ASIGNADOS AL SISTEMA DE VENTILACIÓN**

Tabla 11 Puertos del arduino para el sistema de ventilación

SISTEMA	ELEMENTO DE ENTRADA	PUERTO ENTRADA	FUNCIÓN
Ventilación	Sensor de temperatura	A1	Censa temperatura
	ELEMENTO DE SALIDA	PUERTO SALIDA	FUNCIÓN
	Relé	23	Ventilación central
		25	Ventilación lateral

Fuente: Propia

### 3.6.4.4 CIRCUITO DE CONTROL DEL SISTEMA

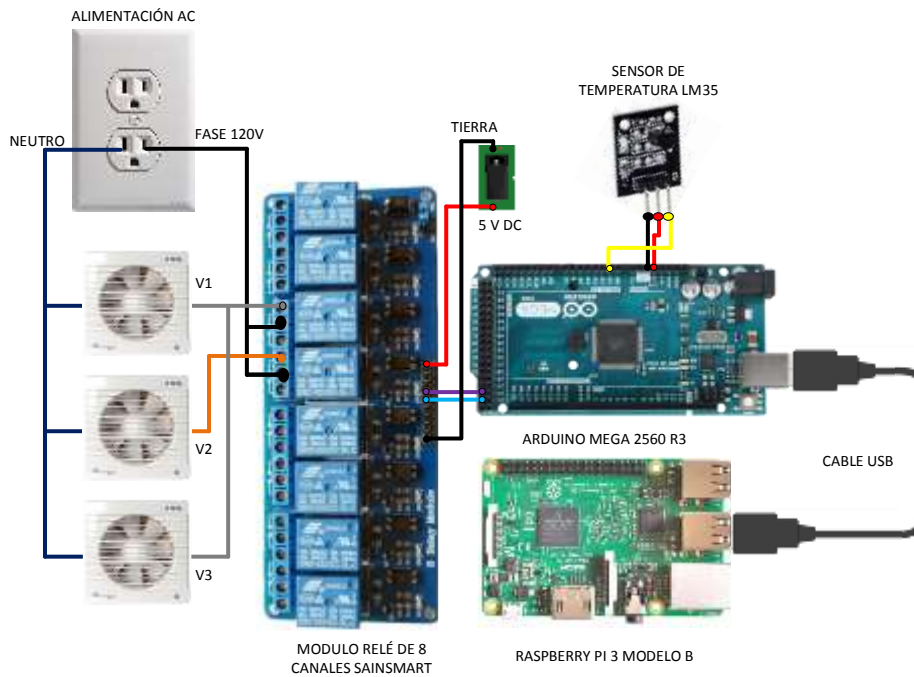


Figura 129 Circuito del sistema de ventilación

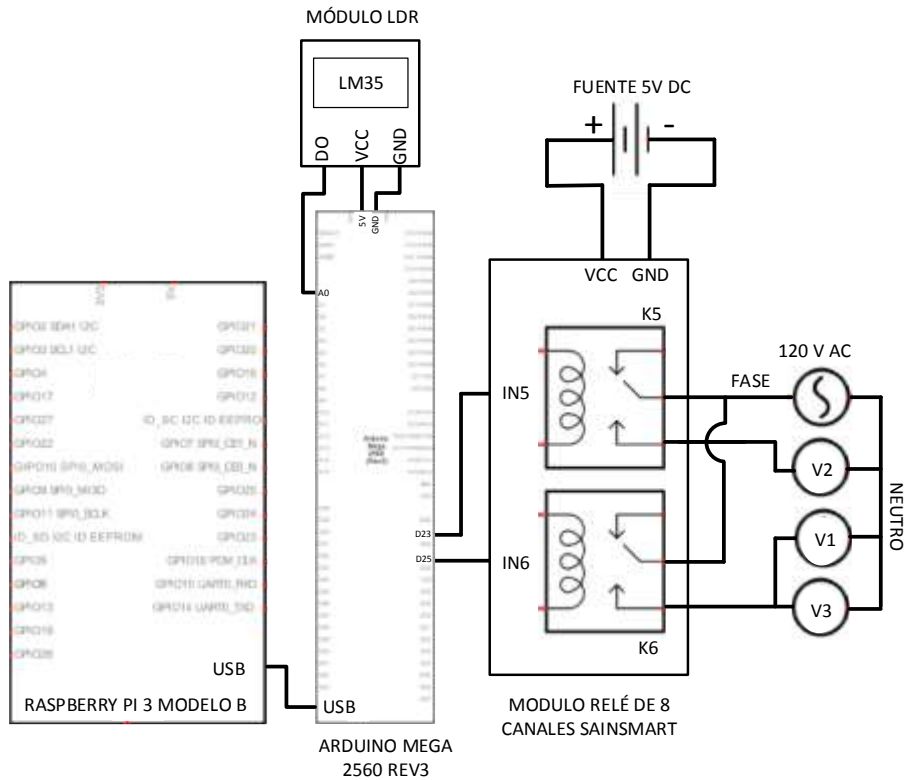


Figura 130 Diagrama de conexiones del sistema de ventilación

### 3.6.4.5 ARGUMENTOS DEL SISTEMA DE VENTILACIÓN

#### CUADRO DE ARGUMENTOS DEL SISTEMA DE VENTILACIÓN RECIBIDOS POR EL ARDUINO

Tabla 12 Parámetros que envía el raspberry pi al arduino vía serial

T_M	100	ACTIVA MODO AUTOMATICO
Tem_d	20 <=X <=40	GRADO EN QUE ENCIENDE VENTILACIÓN
T_M	200	ENCIENDE VENTILACIÓN
T_M	300	APAGA VENTILACIÓN
T_A	400	PREGUNTA Y SOLICITA TEMP. ACTUAL

Fuente: Propia

### 3.6.4.6 UBICACIÓN DE ELEMENTOS DEL SISTEMA

Los tres extractores se encuentran en la Pared posterior de la zona de descanso, el sensor de temperatura se encuentra bajo la caja de protección de los tres extractores en un encapsulado de acrílico.

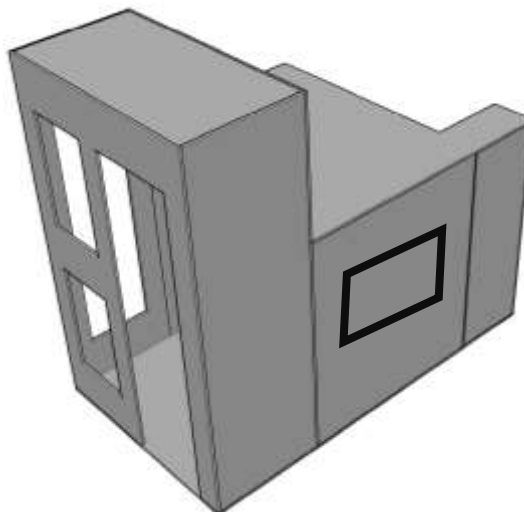


Figura 131 Ubicación referencial de extractores y sensor de temperatura



Figura 132 Extractores, caja contenedora y sensor de temperatura en zona de descanso

### 3.6.5 SISTEMA DE VIDEOVIGILANCIA

El sistema de video vigilancia se diseñó con la finalidad de poder monitorear visualmente el estado de la mascota, es decir si se encuentra al interior o no de la casa domótica para perros. El rango de visión abarca la zona de descanso, parte de la zona



de alimentación y a su vez a través de los vidrios es posible conocer las condiciones del área exterior.

Es una herramienta útil para validar el correcto encendido y apagado de las luces, así como el sistema de acceso.

La transmisión de video es observada desde una pestaña de la interfaz web en tiempo real por medio del software RPI-Cam-Web-Interface. Esta captura la imagen a través de la cámara RASPI NOIR V2.1 y la retransmite a través de una propia interfaz web desde donde se puede gestionar distintas opciones.

### **3.6.5.1 DISEÑO DEL SISTEMA DE VIDEOVIGILANCIA**

Las consideraciones que se tomaron en cuenta para el sistema son:

- Cámara no debe generar carga excesiva en procesamiento del sistema
- Cámara debe ser compatible con la plataforma Raspberry PI
- Cámara debe ser compatible con el software de transmisión web
- Fácil conexión
- Fácil reemplazo
- Ajuste rápido

## **ELEMENTOS DEL SISTEMA**

### **CAMARA RASPI NOIR V2.1**

Se conecta al Raspberry PI 3 vía puerto DSI. Captura la imagen del interior de la casa y la retransmite hacia la interfaz web por medio de Software.

## **RASPBERRY PI 3**

Ejecuta el servidor RPi CAM Web Interface para la transmisión de video vía interfaz web. Procesa el vídeo que recibe de la cámara.

### **3.6.5.2 SOFTWARE RPICAM WEB INTERFACE**

Con la finalidad de transmitir la imagen capturada por la cámara en una interfaz web realizamos la instalación del software RPi CAM Web Interface.

La información del Software puede ser revisada desde el siguiente repositorio:

<https://elinux.org/RPi-Cam-Web-Interface#Support>

### **INSTALACIÓN DE RPICAM WEB INTERFACE**

Los pasos para la instalación son detallados desde el propio repositorio del Software.

Para iniciar la instalación del software, se digita vía SSH o ventana terminal los siguientes comandos:

```
git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
cd RPi_Cam_Web_Interface
./install.sh
```

Se mantiene las configuraciones por defecto, solo se cambia el nombre de la carpeta en donde se instalará el archivo el cual llamamos xCAM.



```
<div class="container" id="content">  
  <div id="vigilancia" style="text-align: center;">  
    <iframe width="800" height="600" src="/xCAM/min.php">  
    </iframe>  
  </div>  
</div>
```

Figura 135 Iframe de transmisión de video en videovigilancia.html

### 3.6.5.3 UBICACIÓN DE ELEMENTOS DEL SISTEMA

La cámara PI NOIR V2 se encuentra ubicada en la esquina superior izquierda de la parte posterior de la zona de descanso. Esta se encuentra sujeta al techo falso de manera invertida.

Su ubicación es estratégica para poder abarcar el mayor campo de visión

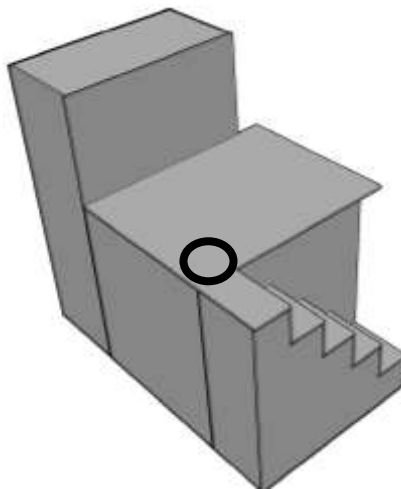


Figura 136 Ubicación referencial de cámara PI NOIR V2



Figura 137 Ubicación de cámara PI NOIR V2 en zona de descanso

### **3.7 DISEÑO Y ELABORACIÓN DEL CENTRO DE CONTROL**

El centro de control contiene los elementos que procesan y reciben la información de todos los sensores, de igual manera es el que activa y desactiva los distintos actuadores del sistema.

#### **3.7.1 ELEMENTOS DEL TABLERO DE CONTROL**

Contiene los siguientes elementos:

- 1 Raspberry Pi 3 modelo B
- 1 Arduino Mega 2560 R3
- 2 módulos L298
- 1 módulo Relé de 8 canales
- Panel de conexiones frontal
- Circuito de interconexión

#### **3.7.2 ELABORACIÓN DEL CENTRO DE CONTROL**

Para determinar la ubicación más funcional de los elementos se realizó varias pruebas, finalmente en base a un prototipo se determinó la ubicación idónea de cada uno de ellos.



Figura 138 Prototipo del centro de control

En base a esta plantilla se diseñó la caja que contiene la unidad de control, se tomó en consideración la ubicación de los elementos de entrada, salida y alimentación del panel.

El material elegido para su construcción es acrílico de 2 mm, lo cual es suficiente para asegurar su estabilidad y rigidez, este mismo material se usó para la elaboración de las cajas que contendrán los sensores del sistema.

El corte de las piezas se realizó en una cortadora laser acorde a una plantilla vectorial previamente diseñada en Illustrator.

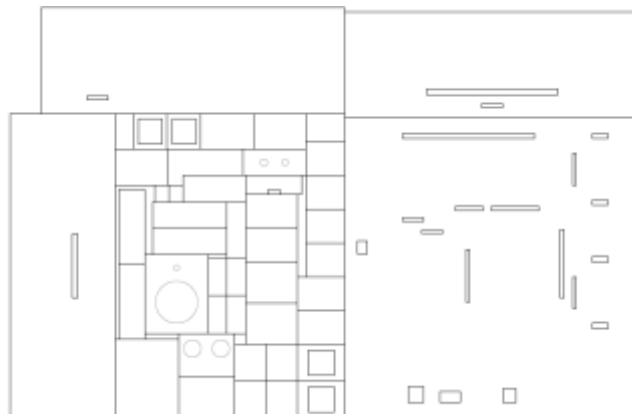


Figura 139 Plantilla vectorial para el corte laser de la plancha de acrílico.



Figura 140 Corte laser de las piezas del centro de control y contenedor de sensores



Figura 141 Ubicación de elementos en tablero de acrílico

Las piezas cortadas en laser fueron unidas con un pegamento especial para acrílico.



Figura 142 Resultado final del armado de las cajas de acrílico

### 3.7.3 DISEÑO DE CIRCUITOS

Para simplificar la conexión e integración del centro de control a la estructura de la casa domótica se desarrolló dos circuitos impresos. En ambos circuitos se colocó borneras de respaldo con la intención de ofrecer escalabilidad al centro de control en caso de que se plantee alguna mejora o adición de funcionalidades.

#### PANEL DE CONEXIONES FRONTAL

Simplifica la interconexión de los distintos sensores y actuadores hacia los puertos GPIO del Raspberry Pi y Arduino.

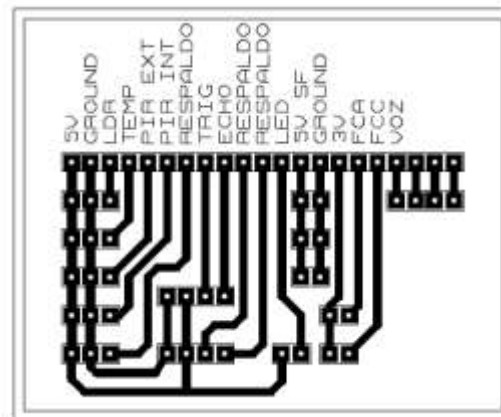


Figura 143 Diseño final de la placa

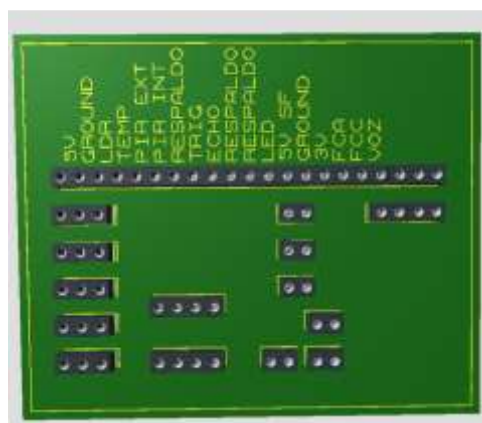


Figura 144 Vista 3D del panel de conexión frontal



## CIRCUITO DE INTERCONEXIÓN

Contiene borneras que unifican y simplifican las conexiones que se lleva a cabo en el centro de control. Además, contiene resistencias necesarias para el circuito, así como el divisor de corriente para el circuito de protección.

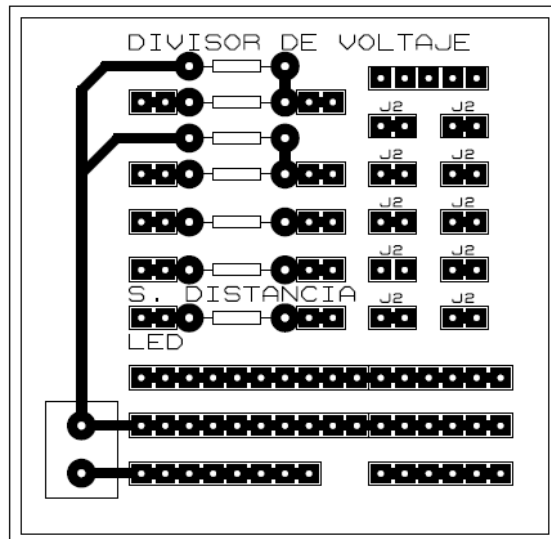


Figura 145 Diseño final del circuito de interconexión

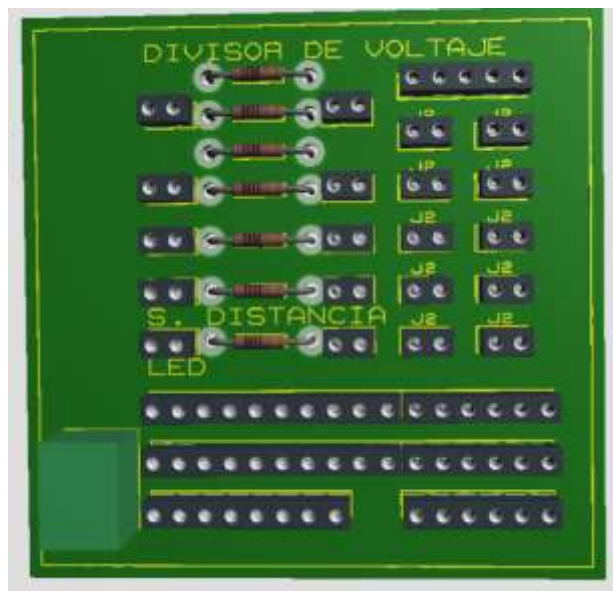


Figura 146 Vista 3D del circuito de interconexión

## IMPRESIÓN, PERFORACIÓN Y SOLDADO DE LOS CIRCUITOS

El circuito se transfirió a la baquelita de cobre a través del método de planchado sobre papel transfer.

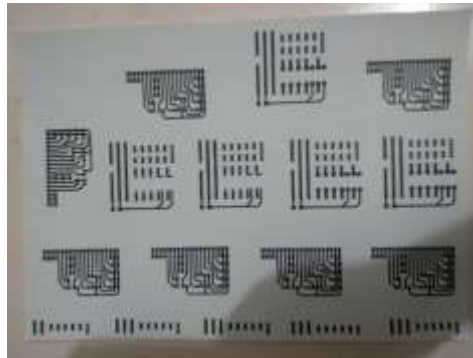


Figura 147 Circuito impreso en papel transfer

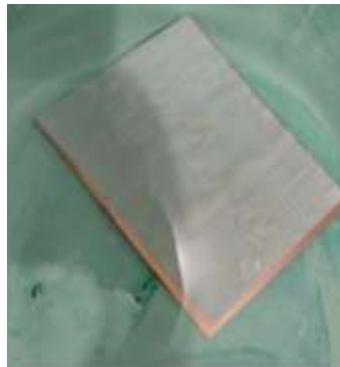


Figura 148 Transferencia de impresión a baquelita de cobre



Figura 149 Circuito impreso en Baquelita

La baquelita es remojada por varias horas en Percloruro Férrico para disolver el cobre no cubierto por la impresión.

Una vez disuelto todo el cobre no cubierto de la baquelita se limpia la impresión con acetona.

Se perfora los orificios y se procede con el soldado de los distintos elementos.



Figura 150 Circuito de interconexión terminado

### **3.7.4 CABLEADO DEL CENTRO DE CONTROL**

Los distintos elementos del centro de control fueron cableados con cable puente acorde al tipo de conectores de cada elemento.

Los elementos energizados por la fuente de 24 V DC fueron conectados con cable flexible 16 AWG.

La alimentación de 110V AC que se conecta hacia el relé fue realizado con cable flexible 14 AWG, de igual manera se cableó la salida de estos hacía una bornera fija en el lado derecho de la caja.



Figura 151 Conexión de relés hacia bornera



Figura 152 Cable puente conectado en puertos GPIO del raspberry pi

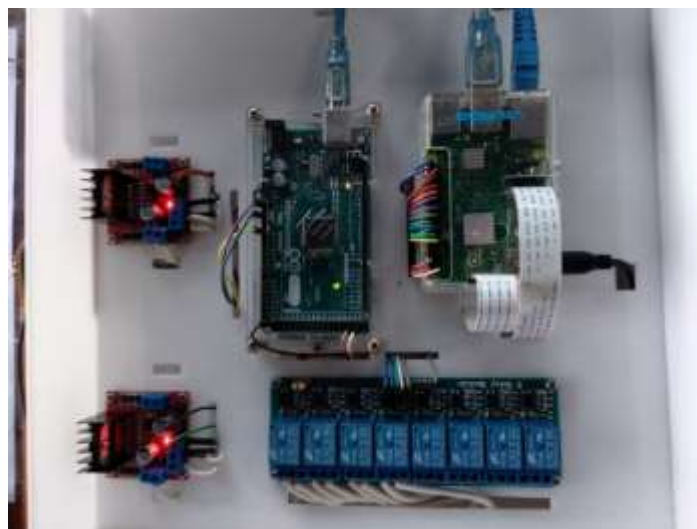


Figura 153 Tablero de control cableado y energizado

### **3.8 DISEÑO DE FUENTE DE ALIMENTACIÓN DEL SISTEMA**

El sistema total de la casa es energizado a través de un UPS el cual regula y protege los elementos que llevan a cabo los procesos principales.

Desde la salida del UPS que posee protección y respaldo de energía se conecta una fuente de 5V DC a 2.5 Amperios y la fuente de 24 V DC a 2.5 Amperios, esto se realizó con la finalidad de dar autonomía y proteger de cortes o fallas de corriente los procesos principales como lo son el acceso y la alimentación.

La autonomía total es de 40 minutos aproximadamente, además se evita que el sistema se apague de manera repentina lo q evita daño y afectaciones al sistema.

Desde la salida con protección, pero sin respaldo de energía del UPS en cambio se conecta una fuente de 5 V DC a 2.5 Amperios y una línea directa de 110 V para la energización de las lámparas y ventiladores cuando son activados por el relé.

Esta fuente de 5V es la que da aviso al Raspberry cuando se tiene un corte de energía para que proceda con la apertura de la puerta, los demás procesos en que interviene son más prescindibles y pueden ser retomados con total normalidad al restablecerse la energía.

Los elementos de 110 V como lo son la iluminación y ventilación no se les ofrece respaldo de energía con la finalidad de dar más tiempo de autonomía a la batería del UPS.

La alimentación de los distintos elementos con fuentes independientes se planteó con la finalidad de que cada elemento de la casa sea de fácil reparación o reemplazo, minimizando de esta forma la afectación y tiempo de solución ante cualquier falla.

### 3.8.1 DIAGRAMA DE ALIMENTACIÓN GENERAL DEL SISTEMA

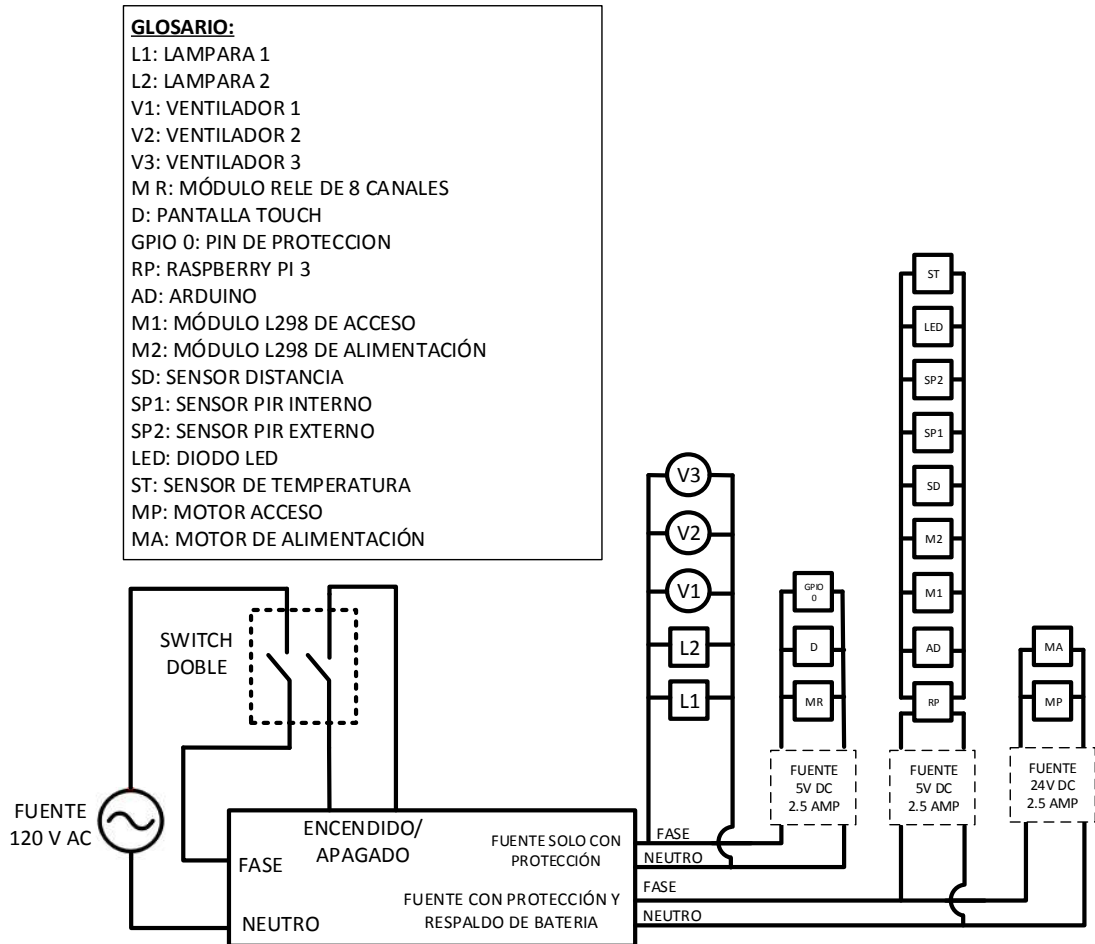


Figura 154 Diagrama de alimentación general del sistema

### 3.8.2 CONEXIÓN DE FUENTE DE ALIMENTACIÓN GENERAL DEL SISTEMA

Se modificó la conexión del interruptor de encendido del UPS de manera que pueda controlarse desde un SW doble ubicado en la parte posterior de la casa.



Figura 155 UPS con cables soldados directo al interruptor de encendido del UPS



Figura 156 Conexión de fuente de alimentación general del sistema



Figura 157 Tomacorrientes con protección UPS

### 3.9 CONEXIÓN Y CABLEADO GENERAL DEL SISTEMA

Sensores y elementos cuya alimentación es de 5V DC fueron cableados con cable tipo puente y ponchados en sus extremidades acorde al conector de cada elemento.

Motores DC de 24 V fueron cableados con cable flexible 16 DWG.

Los elementos de 110 V AC fueron conectados con cable flexible 14 DWG.



Figura 158 Ponchado de cables tipo puente para sensores



Figura 159 Conexión y pruebas de extractores del sistema de ventilación



Figura 160 Conexión del sensor de Temperatura LM35





Figura 161 Pruebas de campo de detección del sensor PIR

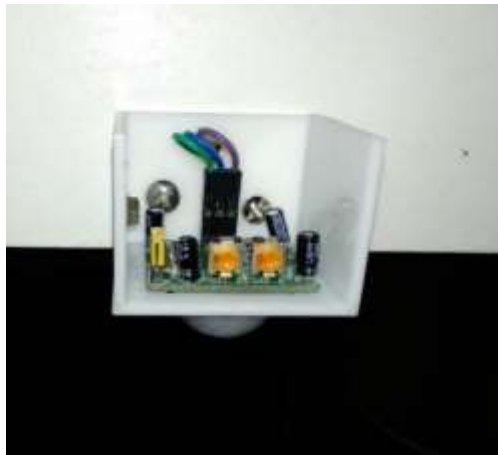


Figura 162 Conexión del sensor PIR externo



Figura 163 Conexión del Dosificador de alimentos



Figura 164 Elaboración de la caja de resonancia para circuito reproductor de voz



Figura 165 Conexión del circuito reproductor de voz y led en caja de resonancia



Figura 166 Conexión del sensor de distancia del sistema de Alimentación



Figura 167 Conexión de pantalla táctil y cable de red



Figura 168 Conexión de sensores en panel de conexiones frontal



Figura 169 Integración del cableado a centro de control



Figura 170 Cableado final del centro de control

## 4. RESULTADOS

### 4.1 SISTEMA DE ALIMENTACIÓN

Se diseñó un sistema de alimentación funcional que permite determinar la cantidad de alimento que debe servir, es programable y adaptable a distintas exigencias que requiera el usuario para con su mascota.

Posee un sistema que alerta al usuario cuando debe recargar alimento, además el dosificador es desmontable para su lavado y mantenimiento.

Detalle de funcionamiento:

Se coloca alimento en el contenedor del dispensador de comidas



Figura 171 Vaciado de alimento en contenedor del dosificador



Figura 172 Alimento dentro del dosificador

## PRUEBA DE LA OPCIÓN SERVIR PORCIÓN



Figura 173 Selección de opción “Servir Porción” para raza pequeña

Se acciona el dosificar y expulsa una leve porción de comida para perros acorde a la selección dada. El led permanece apagado debido a que aún se tiene alimento en el dispensador. Se activa el reproductor de voz y emite un sonido previamente guardado.



Figura 174 Sistema de dosificación

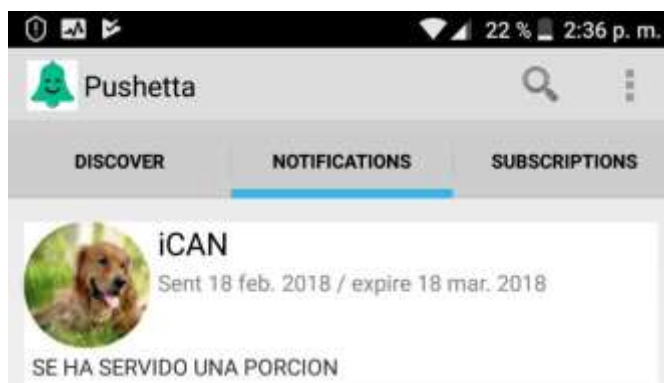


Figura 175 Notificación en Pushetta de porción servida

## PRUEBA DEL MODO AUTOMÁTICO

Se configura para que a las 2:46 PM se sirva una porción para raza de perro grande.



Figura 176 Selección en modo automático del dispensador de comida

Se sirve una porción y se deja poco alimento en el interior del envase para validar que salte la alarma de recarga de alimento.

En efecto sirve la porción de alimento.



Figura 177 Porción de alimento servida por el dosificador para raza grande

Se reproduce el sonido de llamado al perro y luego se activa el led que indica que se debe recargar alimento en el contenedor



Figura 178 Led de recarga de alimento encendido

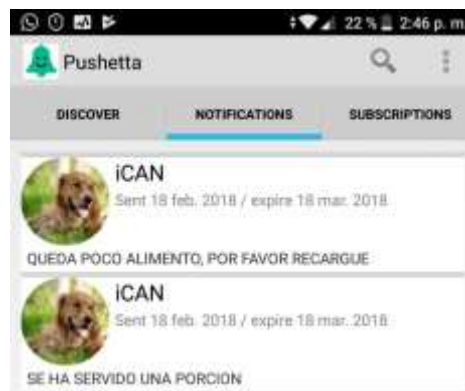


Figura 179 Notificación en Pushetta de recarga de alimento

## 4.2 SISTEMA DE ILUMINACIÓN

El sistema de iluminación diseñado se controla de manera eficaz en base a la elección tomada por el usuario en la interfaz web. Posee dos modos: automático y manual. Las luminarias no generan calor, no consumen mucha corriente, su intensidad es adecuada y su luz cubre toda el área interior de la estructura.

Detalle del funcionamiento:

Inicialmente ambas luminarias se encuentran apagadas





Figura 180 Luminaria L2 apagada



Figura 181 Luminaria de L1 apagada

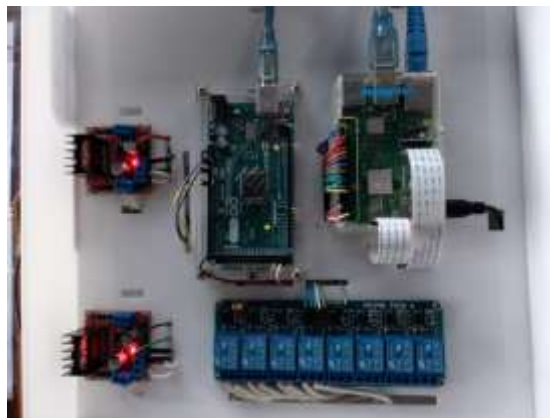


Figura 182 Relés de iluminación desactivados

## PRUEBA DEL MODO MANUAL

Inicialmente se solicita Encender L1



Figura 183 Selección On de “Luces Ambiente 1”

Inmediatamente se activa el Relé que controla la iluminación de L1

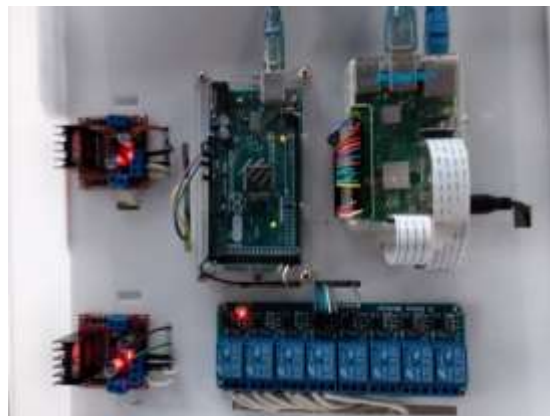


Figura 184 Relé activo de L1



Figura 185 Encendido de L1

Se solicita apagar L1 y encender L2.



Figura 186 Selección Off de L1 y On de L2

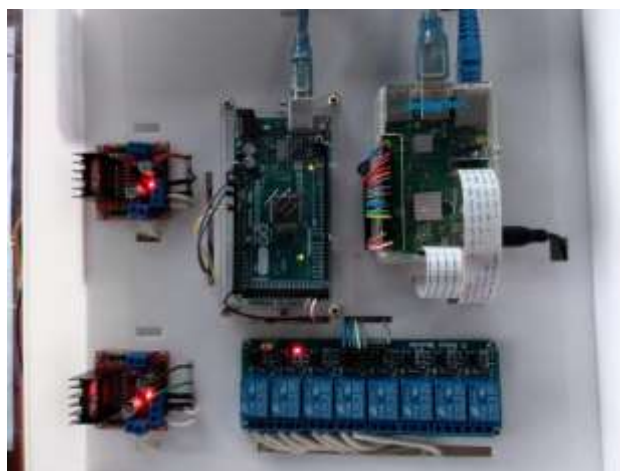


Figura 187 Relé de L1 apagado de y L2 encendido



Figura 188 Luminaria L2 encendida

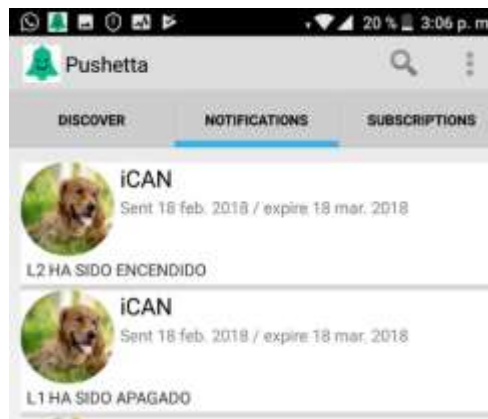


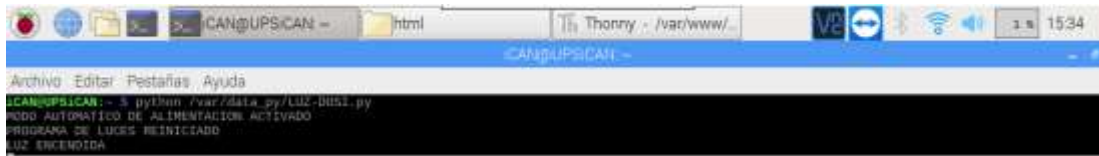
Figura 189 Notificación Push de las acciones ejecutadas

## PRUEBA DEL MODO AUTOMÁTICO – HORARIO



Figura 190 Se selecciona el modo horario automático

Las Luces se encienden a las 3:34 PM como se indicó



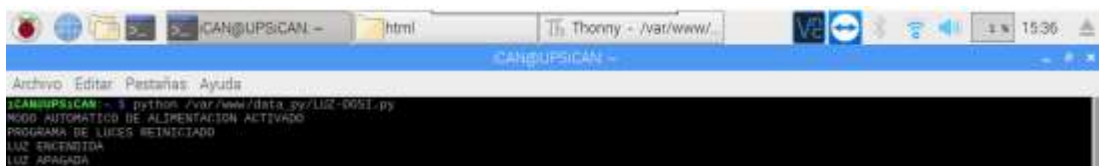
```
ican@UPSICAN ~$ python /var/www/html/luz-dosi.py
MODO AUTOMÁTICO DE ALIMENTACION ACTIVADO
PROGRAMA DE LUCES REINICIADO
LUZ ENCENDIDA
```

Figura 191 Depuración de LUZ-DOSI.py para observar acciones en tiempo real



Figura 192 Se activa ambos relés de L1 y L2

A las 3:36 PM según la instrucción dada se apagan:



```
ican@UPSICAN ~$ python /var/www/html/luz-dosi.py
MODO AUTOMÁTICO DE ALIMENTACION ACTIVADO
PROGRAMA DE LUCES REINICIADO
LUZ ENCENDIDA
LUZ APAGADA
```

Figura 193 Depuración de LUZ-DOSI.py para observar acciones en tiempo real

Se recibe notificación al celular Android con las acciones ejecutadas por parte del programa.

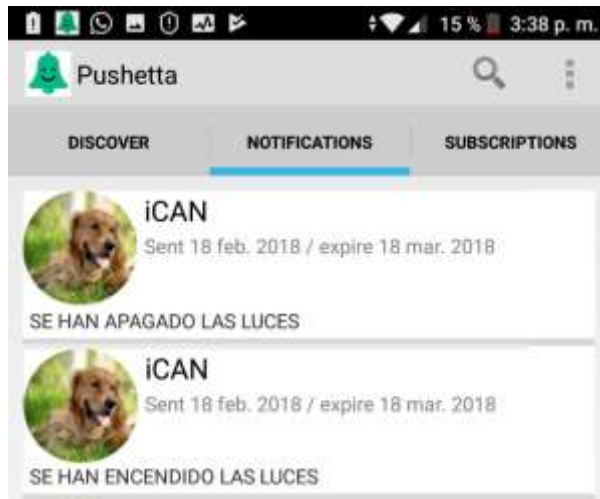


Figura 194 Notificación Push del programa ejecutado

## PRUEBA DEL MODO AUTOMÁTICO – SENSOR



Figura 195 Selección del modo automático sensor

Obstaculizando intencionalmente el lado donde se encuentra el sensor de temperatura se consigue Encender o apagar las luces según sea el caso.

```
iCAN@UPSiCAN:~ $ python /var/www/data_py/LUZ-DOSI.py
LUCES APAGADAS
SENSOR DE LUZ ACTIVADO
MODO AUTOMATICO DE ALIMENTACION ACTIVADO
LUCES ENCENDIDAS
LUCES APAGADAS
```

Figura 196 Depuración del proceso principal LUZ-DOSI.py

### 4.3 SISTEMA DE ACCESO

El sistema de acceso implementado se realizó de manera que brinde todas las seguridades necesarias para el perro durante su funcionamiento. Su apertura puede ser controlada de manera manual o automática a través de los sensores ubicados en la parte interna y exterior de la puerta, los mismos detectan y evitan el cierre de la puerta si se encuentran algún obstáculo en su recorrido. Además el sistema de protección que detecta cortes de **energía** evita el riesgo que el perro pueda quedarse encerrado en el interior de **esta**.

Detalle del funcionamiento:

#### PRUEBA DE APERTURA DE LA PUERTA

Iniciamos el escenario de pruebas con la puerta cerrada, luego seleccionamos Abierto.



Figura 197 Selección de abierto en el menú de acceso

La puerta empieza a abrirse hasta que acciona el final de carrera ABRE y se detiene.

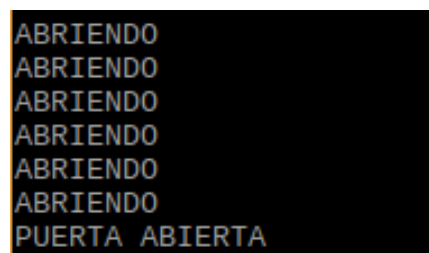


Figura 198 Depuración de ACCESO.py para observar en tiempo real el proceso



Figura 199 Puerta abierta de la casa domótica

Se recibe una notificación Push indicando que la puerta ha sido abierta.

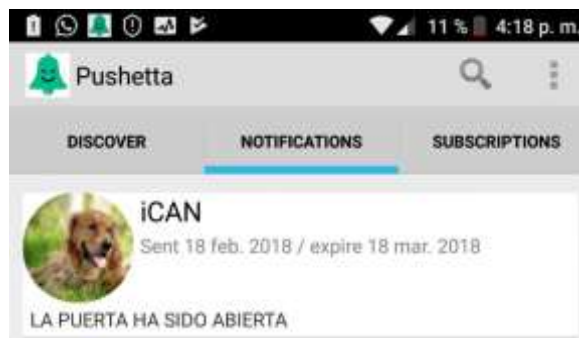


Figura 200 Notificación Push que informa sobre la apertura de la puerta

## PRUEBA DE CIERRE DE LA PUERTA



Figura 201 Selección de cierre de la puerta



La puerta empieza a cerrarse hasta que acciona el final de carrera cierra y se detiene

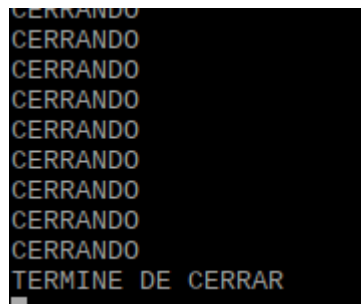


Figura 202 Depuración de ACCESO.py para observar en tiempo real el proceso

Se recibe una notificación Push indicando que la puerta ha sido cerrada.



Figura 203 Notificación Push que informa sobre el cierre de la puerta



Figura 204 Puerta cerrada

## PRUEBA DE SEGURIDAD ANTE PRESENCIA DE OBSTACULOS

Si la puerta se cierra y uno de los dos sensores PIR detecta movimiento cerca de la puerta enseguida detienen el cierre y abren la puerta

```
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
SE ABRE POR OBSTACULO
```

Figura 205 Depuración de ACCESO.py con ejecución de seguridad para el cierre

Una vez abierta espera un determinado tiempo antes de intentar volver a cerrar, si esta vez no detecta ningún movimiento cerca del área se cierra exitosamente. Este proceso se puede repetir las veces que sean necesarias hasta que el sistema detecte que es seguro cerrar.

```
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
CERRANDO
TERMINE DE CERRAR
TERMINE DE CERRAR
```

Figura 206 La puerta intenta volver a cerrarse y esta vez es exitoso

## PRUEBA DE APERTURA ANTE CORTE DE ENERGÍA

Si la puerta está cerrada y llegase haber un corte de energía de inmediato se detecta y se envía a abrir la puerta.

Este corte lo simulamos desconectando de la parte trasera de la casa el cable de poder.

```
TERMINE DE CERRAR
SISTEMA DE PROTECCIÓN
ABRIENDO
ABRIENDO
ABRIENDO
ABRIENDO
ABRIENDO
ABRIENDO
```

Figura 207 Depuración de ACCESO.py para validar los procesos en tiempo real

En la Figura se aprecia el instante en que existe un corte energía y se dispara la alarma del sistema de protección. Inmediatamente empieza a abrir la puerta.

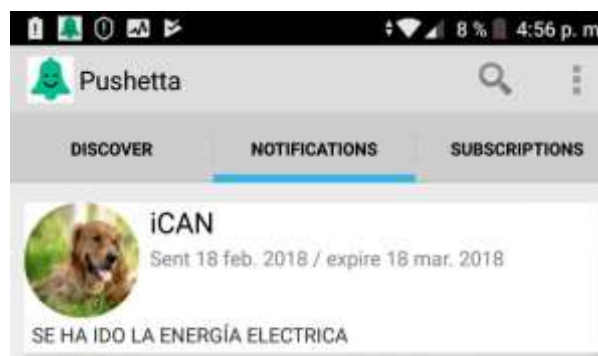


Figura 208 Notificación Push de corte de energía

#### 4.4 SISTEMA DE VENTILACIÓN

El sistema de ventilación implementado permite mantener un ambiente agradable al interior de la casa. Puede ser activado de manera manual o automático en base a la elección que tome el usuario en la interfaz web. Por medio del sensor de temperatura se puede conocer la temperatura actual del interior y definir a partir de qué medida se debe encender el sistema. Al hacer uso de tres extractores se mantiene un flujo constante de aire direccionado hacia la parte baja de la casa por medio de la rejilla metálica de protección que se ubicó al interior de la casa.

Detalle del funcionamiento

Estado inicial de los relés que controlan los extractores de aire



Figura 209 Estado inicial de relés del sistema de ventilación

## PRUEBA DEL MODO MANUAL

Encendido manual del sistema de Ventilación a través de las opciones del Menú. Observamos que al desplegarse el menú en efecto nos muestra la temperatura actual.



Figura 210 Selección de modo manual del sistema de ventilación

Al seleccionar ON se enciende a través del arduino los dos relés que activan los extractores. En seguida empieza a sentirse un flujo de aire en la parte baja de la zona de descanso.

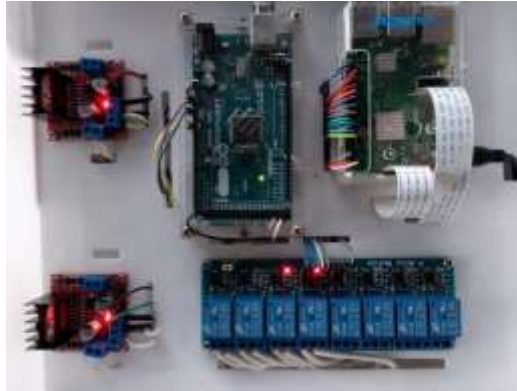


Figura 211 Estado activo de relés del sistema de ventilación



Figura 212 Extractores encendidos dentro de la zona de descanso.

### **PRUEBA DEL MODO AUTOMÁTICO**

Se fija la temperatura de encendido a un valor superior al de la Temperatura actual.



Figura 213 Fijado 30 grados el valor de encendido del sistema

Inmediatamente luego de este cambio los extractores se apagan.

Se fija la temperatura de encendido a un valor inferior del de la Temperatura actual.



Figura 214 Fijado 30 grados el valor de encendido del sistema

Inmediatamente luego de este cambio los extractores vuelven a encender.

#### **4.5 SISTEMA DE VIDEOVIGILANCIA**

El sistema de video vigilancia implementado en el sistema se integra a la interfaz web de manera óptima. La transmisión de video es fluida y puede ser modificada para que se adapte a las condiciones del ambiente así como a la velocidad de internet que posea el usuario.

La cámara usada es compacta y no genera mucha carga al procesador del Raspberry PI.

Detalle del funcionamiento:

#### **PRUEBAS DE TRANSMISIÓN DE VIDEO A TRAVÉZ DE INTERFAZ WEB**

Como se observa en la Figura la transmisión de video es óptima. La calidad de imagen es buena y fluida. Está integrado en una página web de características similares a nuestro menú principal.

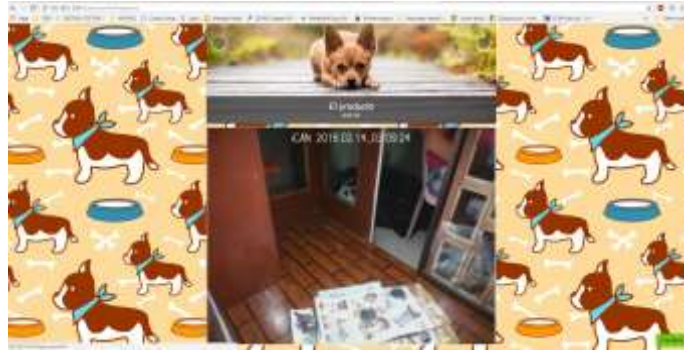


Figura 215 Interfaz web de videovigilancia

#### **4.6 PRUEBAS DE CONTROL POR INTERFAZ Y PANTALLA TÁCTIL**

La pantalla táctil permite manejar y seleccionar las distintas opciones del sistema.

Como la interfaz es dinámica y las elecciones se realizan a través de flechas de desplazamiento es posible manejarlo sin necesidad de teclado en pantalla.

Además, es útil en caso de soportes o inconvenientes que se llegasen a presentar. En su parte posterior se colocó un cable de red que va conectado al raspberry por medio del cual se puede realizar reparaciones o ejecutar instrucciones de primera instancia.



Figura 216 Pantalla táctil con interfaz web.

#### 4.7 GRADO DE SATISFACCIÓN E INTEGRACIÓN DE LA CASA DOMÓTICA CON PERROS



Figura 217 Perro asomado en el balcón de la casa

Como se observa en las siguientes imágenes la casa tuvo acogida inmediata por parte de los perros, inclusive durante su implementación ya previamente se había realizado pruebas con perros constatando que logren familiarizarse de manera eficaz.

Hacen uso correcto del sistema de alimentación y se sienten cómodos con la ventilación que se posee en su interior.

La estructura de la casa es apta para el uso de todo tipo de perros. Los materiales elegidos son resistentes a condiciones del ambiente y fáciles de mantener.

El diseño de la casa protege y centraliza el panel de control dando facilidades de acceso y mantenimiento al sistema.



Figura 218 Perros descansando al interior de la casa





Figura 219 Perro en la zona de descanso de la casa domótica



Figura 220 Perro recibiendo aire del sistema de ventilación



Figura 221 Perro descansando luego de haberse alimentado en la zona de alimentación



Figura 222 Plano general de la casa



Figura 223 Vista lateral

#### **4.8 ANÁLISIS DE RESULTADOS**

Los cinco sistemas principales que comprenden el sistema domótico funcionan de manera correcta, su control es amigable y de fácil aprendizaje por parte del usuario.

El perro se adapta e integra fácilmente al funcionamiento de los sistemas.

Perros se acercan a la puerta y esperan a que la puerta se abra para ingresar a la misma, se ubican en la zona de descanso a refrescarse con el sistema de ventilación implementado.

Se familiarizan con el sistema de alimentación y al escuchar el sonido de la voz del dueño se acercan de inmediato al interpretar que es hora de la comida.

Durante su implementación se realizó pruebas con perros pertenecientes a la administración del hostel quienes hacían uso de este durante las últimas etapas del proyecto.

El material con el que se contruyó la estructura de la casa ha resistido lluvias y las altas temperaturas de Chipipe – Salinas, el techo evita que caiga la luz del sol y a su vez la madera hace de aislante natural para evitar temperaturas elevadas o fríos intensos durante la noche.

Se tuvo un impacto positivo para el servicio ofrecido por el hotel ya que durante su construcción se tuvieron varias personas interesadas en su funcionamiento y que tiempo después se acercaban con sus mascotas para observar su reacción.

El impacto creado en las personas al observar la casa domótica para perros fue positivo. Interpretando su funcionamiento y dando constancia de la utilidad que tendría para el cuidado de sus mascotas.

Luego de la reacción e impacto generado en la clientela se vió la necesidad y oportunidad a mediano plazo de implementar una segunda casa para ofrecer el servicio de cuidado de mascotas para las personas que acudan al balneario.

Con la implementación de la casa se evitará la restricción del hostel de admitir perros grandes en los cuartos ya que los mismos serán ubicados en la terraza, lugar donde se encuentra la casa domótica, y se brindara a los huéspedes las credenciales de acceso a la interfaz web para que puedan monitorear, observar y darle la atención que requiera el perro durante su estancia.

## 5. CONCLUSIONES

La estructura es apta para cualquier tipo de tamaño y raza de perros, tiene buena aceptación por parte de estos y logran adaptarse a su funcionamiento.

Se concluye que el sistema de dosificación permite asegurar la correcta alimentación del perro de manera automatizada. Es de fácil configuración y mantenimiento.

Se determina que el sistema de ventilación implementado en el proyecto proporciona un ambiente agradable y fresco al interior de la casa. Es automático, de fácil configuración y mantenimiento.

El sistema de iluminación es idóneo para las dimensiones de la casa y adecuado para el confort del perro. Su modo automático es funcional y de fácil configuración.

El sistema de video vigilancia del proyecto permite observar en tiempo real a través de la interfaz web lo que sucede al interior de la casa domótica.

El sistema de acceso del sistema permite controlar la apertura manual o automática de la puerta de ingreso a la casa de manera exitosa en base a la elección tomada por el usuario desde la interfaz web.

Se concluye que los sensores integrados a la casa domótica en conjunto permiten conocer el estado del interior de la casa y el ambiente en que se encuentra el perro. Se encuentran en módulos y son de fácil reemplazo.

La comunicación entre Arduino y Raspberry Pi se realiza exitosamente a través de puerto serial, comparten información de los sensores.

La interfaz web implementada para el control del sistema es dinámica y muestra de manera clara y objetiva los distintos tipos de control y opciones que posee cada sistema.

Los procesos programados en Python recogen información de los sensores, interactúan con la interfaz web y actuadores del sistema domótico de manera coordinada y funcional.

Con respecto a la tarjeta Arduino Mega se configuró para llevar a cabo el proceso del sistema de ventilación. Sensa la temperatura y la comunica al Raspberry Pi cada vez que el usuario lo solicita.

La interfaz web permite controlar el funcionamiento de los sistemas, recibe y procesa parámetros y muestra información de sensores como la temperatura y captura de video. Se tiene gestión de este a través de Internet.

La fuente de energía de todo el sistema brinda protección y respaldo de corriente a los procesos principales.

## **6. RECOMENDACIONES**

Con respecto al diseño se recomienda conocer el ambiente y condiciones en que estará ubicada la casa domótica, de ser posible el tipo de perro que va a dar uso y si requiere de algún tipo de cuidado especial. Su distribución de espacios debe ser modular, de fácil ensamble y desarme. Acorde a las exigencias se puede hacer uso de distintos tipos de materiales en base al presupuesto y necesidades.

Se recomienda identificar cada una de las conexiones y circuitos del centro de control para brindar facilidades de mantenimiento.

Con respecto al sistema de dosificación se recomienda al usuario realizar la limpieza del dosificador por lo menos una vez dentro de uno o dos meses.

El sistema de ventilación usado en el sistema debe brindar un flujo de aire constante y ligero, no se puede dar un flujo demasiado fuerte debido a que incomodaría al perro y a su vez genera mucho ruido. La ventilación debe estar bien protegida para evitar que el perro sufra algún accidente.

El control elegido a través de la interfaz para el sistema de iluminación debe ser el más adecuado en base al uso que va darle el perro a la casa y a las condiciones ambientales en que se encuentra, esto con el fin de evitar el uso innecesario de iluminación y evitar gastos energéticos.

Para mantener una fluida transmisión de video del sistema hacia la interfaz web desde un host externo es necesario disponer de un ancho de banda de mínimo 10 MB.

La puerta de acceso principal no debe intentar abrirse o cerrarse de manera manual sino solo a través de la interfaz web.

Identificar por parte del usuario la ubicación de los sensores y no obstaculizar o realizar maniobras que puedan afectar su funcionamiento.

Se debe tener en cuenta que la configuración del puerto serial debe ser la misma para ambos equipos, así como asegurarse de disponer de las librerías necesarias.

La interfaz no debe recargarse de mucho contenido con la finalidad de que la navegación sea fluida y el tiempo de carga sea mínimo.

Se debe hacer uso de librerías y clases que simplifiquen los procesos ejecutados por la raspberry. Cada uno de los procesos debe tener instrucciones y protecciones en caso de falla para el reinicio y restablecimiento del sistema.

Para el correcto funcionamiento del arduino mega 2560 se recomienda colocar un capacitor de 10 uF entre el puerto Ground y Reset, sus puertos analógicos pueden ser usados como mejora del sistema para la integración de más sensores.

Se recomienda hacer uso de software de notificaciones PUSH con la finalidad de conocer en tiempo real las acciones que ejecuta el sistema.

Se recomienda ser pacientes durante el tiempo de adaptación del perro con su nueva casa domótica, hacer demostración de los usos que le puede dar y asegurarse que su estancia sea agradable

En los circuitos se debe referenciar la masa común de todas las fuentes para disminuir el ruido eléctrico que se genera.

Se recomienda mantener una limpieza continua de la casa para extender su buen funcionamiento.

La ubicación de la casa domótica debe poseer buena intensidad de señal wifi de lo contrario se recomienda el uso de repetidores con la finalidad de asegurar cobertura inalámbrica.



## 7. BIBLIOGRAFÍA.

- [1] K. Ochoa. [En línea]. Available: <https://www.metroecuador.com.ec/ec/noticias/2014/10/02/hasta-80-denuncias-diarias-animales-abandonados.html>. [Último acceso: 19 Noviembre 2017].
- [2] G. A. y. P. Sra. Ministra de Salud Pública y el Ministerio de Agricultura, «Petnet.ec,» 19 Febrero 2009. [En línea]. Available: <http://petnet.ec/video/REGLAMENTO.pdf>. [Último acceso: 19 Noviembre 2017].
- [3] L. T. Ojeda, «arduino.cl,» Ingeniería MCI Ltda, [En línea]. Available: <http://arduino.cl/que-es-arduino/>. [Último acceso: 19 Noviembre 2017].
- [4] L. T. Ojeda, «arduino.cl,» Ingeniería MCI Ltda., [En línea]. Available: <http://arduino.cl/arduino-mega-2560/>. [Último acceso: 19 Noviembre 2017].
- [5] L. Llamas, «Luis Llamas,» 15 Octubre 2017. [En línea]. Available: <https://www.luisllamas.es/que-es-raspberry-pi/>. [Último acceso: 19 Noviembre 2017].
- [6] Raspiman, «Raspiman,» 16 Julio 2014. [En línea]. Available: <http://www.raspiman.com/que-es-una-raspberry-pi/>. [Último acceso: 19 Noviembre 2017].
- [7] P. Moya, «Omicrono, la tecnología de EL Español,» 29 Febrero 2016. [En línea]. Available: <http://omicrono.elespanol.com/2016/02/raspberry-pi-3-model-b/>. [Último acceso: 19 Noviembre 2017].
- [8] Python, «Phyton.org,» [En línea]. Available: <http://brochure.getpython.info/>. [Último acceso: 19 Noviembre 2017].
- [9] infostudio, «infostudio,» 31 Agosto 2011. [En línea]. Available: [http://infostudio-blog.blogspot.com/2011/08/principales-caracteristicas-del\\_31.html](http://infostudio-blog.blogspot.com/2011/08/principales-caracteristicas-del_31.html). [Último acceso: 19 Noviembre 2017].

- [10] Raspberrypi, «Raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/documentation/remote-access/web-server/README.md>. [Último acceso: 19 Noviembre 2017].
- [11] Raspberrypi, «Raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>. [Último acceso: 19 Noviembre 2017].
- [12] Raspberrypi, «Raspberrypi,» [En línea]. Available: <https://www.raspberrypi.org/documentation/remote-access/web-server/nginx.md>. [Último acceso: 19 Noviembre 2017].
- [13] J. Barberá, «Horizonweb,» [En línea]. Available: <http://www.xn--diseowebmurcia1-1qb.es/interfaz-web/>. [Último acceso: 19 Noviembre 2017].
- [14] Manz, «lenguajecss,» [En línea]. Available: <https://lenguajecss.com/p/css/introduccion/que-es-css>. [Último acceso: 19 Noviembre 2017].
- [15] L. Chuburu, «laurachuburu,» [En línea]. Available: <http://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>. [Último acceso: 19 Noviembre 2017].
- [16] R. G. Latinoamericana, «redgrafica,» [En línea]. Available: <http://redgrafica.com/El-lenguaje-de-programacion-PHP>. [Último acceso: 20 Noviembre 2017].
- [17] Gumino.com, «Pushetta,» 2014. [En línea]. Available: <http://www.pushetta.com/pushetta-docs/>. [Último acceso: 26 Febrero 2018].
- [18] G. M, «Concepto definicion,» [En línea]. Available: <http://conceptodefinition.de/motor-electrico/>. [Último acceso: 20 Noviembre 2017].
- [19] M. ELéctricos, «motoreselectricos7,» 2013. [En línea]. Available: <http://motoreselectricos7.webnode.es/cambio-de-sentido-de-giro-/>. [Último acceso: 20 Noviembre 2017].

- [20] J. P. Forero, «blogspot.com,» 18 Noviembre 2015. [En línea]. Available: <http://taladrodearbolinstrumentacion.blogspot.com/2015/11/motores-con-caja-reductora-y-puente-en-h.html>. [Último acceso: 20 Noviembre 2017].
- [21] glosario.net, «glosario.net,» 9 Noviembre 2006. [En línea]. Available: <http://arte-y-arquitectura.glosario.net/construccion-y-arquitectura/puerta-autom%EAtica-7495.html>. [Último acceso: 21 Noviembre 2017].
- [22] instelec, «instelec,» [En línea]. Available: <http://puertaautomatica.es/puertas-automaticas>. [Último acceso: 22 Noviembre 2017].
- [23] Tecnoentusiastas, «Tecnología y Educación,» 30 Mayo 2012. [En línea]. Available: <http://www.tecnologiayeducacion.com/%C2%BFque-es-luz-led/>. [Último acceso: 22 Noviembre 2017].
- [24] p. led, «Pantallas LED,» 2006. [En línea]. Available: <https://www.pantallasled.com.mx/productos/iluminacion-interior/>. [Último acceso: 22 Noviembre 2017].
- [25] supradesarrollos, «supradesarrollos,» [En línea]. Available: [http://www.supradesarrollos.com/catalogo/led\\_comercial/para-plafon/](http://www.supradesarrollos.com/catalogo/led_comercial/para-plafon/). [Último acceso: 22 Noviembre 2017].
- [26] DefiniciónABC, «DefiniciónABC,» [En línea]. Available: <https://www.definicionabc.com/tecnologia/ventilador.php>. [Último acceso: 22 Noviembre 2017].
- [27] tiposde, «tipos de org,» [En línea]. Available: <http://www.tiposde.org/cotidianos/665-tipos-de-ventiladores/>. [Último acceso: 22 Noviembre 2017].
- [28] Tecnología, «Recursos para el aula de tecnología,» [En línea]. Available: <http://www.tallerdetecnologia.es/jomla37/index.php/materiales/13-contrachapado>. [Último acceso: 22 Noviembre 2017].

- [29] L. Albano, «Mi carpintería,» [En línea]. Available: <https://micarpinteria.wordpress.com/2011/02/17/triplay-que-es/>. [Último acceso: 22 Noviembre 2017].
- [30] Tablecom, «Tablecom,» [En línea]. Available: <http://www.tablecom.com.ec/index.php/component/virtuemart/plywood/plywood-marino-detail?Itemid=0>. [Último acceso: 22 Noviembre 2017].
- [31] Pelikano, «emfalu,» [En línea]. Available: <http://www.emfalu.com/images/plywood-marino.pdf>. [Último acceso: 22 Noviembre 2017].
- [32] p. molina, «profesor molina,» [En línea]. Available: [http://www.profesormolina.com.ar/tecnologia/sens\\_transduct/que\\_es.htm](http://www.profesormolina.com.ar/tecnologia/sens_transduct/que_es.htm). [Último acceso: 22 Noviembre 2017].
- [33] m. temperatura, «medir temperatura,» [En línea]. Available: <http://medirtemperatura.com/sensor-temperatura.php>. [Último acceso: 22 Noviembre 2017].
- [34] Teslabem, «GeekBot,» 2017. [En línea]. Available: <http://teslabem.com/lm35-sensor-de-temperatura.html>. [Último acceso: 2 Marzo 2018].
- [35] Elextrocrea, «electrocrea,» [En línea]. Available: <https://electrocrea.com/blogs/tutoriales/34105155-sensor-fotoelectronico>. [Último acceso: 22 Noviembre 2017].
- [36] I. artificial, «Inteligencia Artificial,» [En línea]. Available: <http://www.inteligenciaartificialyrobotica.com/esp/item/389/sen-ldr-modulo-sensor-ldr>. [Último acceso: 22 Noviembre 2017].
- [37] naylamp, «naylamp,» [En línea]. Available: <http://www.naylampmechatronics.com/sensores-luz-y-sonido/135-modulo-ldr.html>. [Último acceso: 22 Noviembre 2017].

- [38] p. d. tecnología, «Proyectos de tecnología,» [En línea]. Available: <https://elblogdelprofesordetecnologia.blogspot.com/2016/02/microrruptor-o-final-de-carrera.html>. [Último acceso: 22 Noviembre 2017].
- [39] prometec, «prometec,» [En línea]. Available: <https://www.prometec.net/sensor-pir/>. [Último acceso: 22 Noviembre 2017].
- [40] Electronilab, «Electronilab,» [En línea]. Available: <https://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>. [Último acceso: 22 Noviembre 2017].
- [41] electronrtools, «electronrtools,» [En línea]. Available: <http://www.electronrtools.com/Home/WP/2016/05/06/puente-h-con-driver-l298/>. [Último acceso: 22 Noviembre 2017].
- [42] L. Llamas, «Luis LLamas,» 26 Mayo 2016. [En línea]. Available: <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>. [Último acceso: 23 Noviembre 2017].
- [43] Electrocrea, «Electrocrea,» [En línea]. Available: <https://electrocrea.com/blogs/tutoriales/33306499-sensor-ultrasonico>. [Último acceso: 23 Noviembre 2017].
- [44] electronilab, «electronilab,» [En línea]. Available: <https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>. [Último acceso: 23 Noviembre 2017].
- [45] Tecnología, «area tecnologia,» [En línea]. Available: <http://www.areatecnologia.com/electricidad/rele.html>. [Último acceso: 23 Noviembre 2017].
- [46] ecopech, «ecopech,» [En línea]. Available: <http://www.ecopech.net/modulo-relay-optoacoplado-de-8-canales.html>. [Último acceso: 23 Noviembre 2017].
- [47] Prometec, «Prometec,» [En línea]. Available: <https://www.prometec.net/producto/modulo-de-grabacion-de-voz/>. [Último acceso: 23 Noviembre 2017].

- [48] e. pack, «easy pack,» [En línea]. Available: <http://easypack.mx/dosificadora-a-03/>. [Último acceso: 23 Noviembre 2017].
- [49] Ingesir, «Ingesir,» [En línea]. Available: <http://www.ingesir.com.ar/dosific.htm>. [Último acceso: 23 Noviembre 2017].
- [50] WordReference, «WordReference,» [En línea]. Available: <http://www.wordreference.com/definicion/tolva>. [Último acceso: 23 Noviembre 2017].
- [51] M. Perruno, «Muy Perruno,» [En línea]. Available: <https://www.muyperruno.com/casas-para-perros/>. [Último acceso: 23 Noviembre 2017].
- [52] I. Cedeño, «ABOUTESPAÑOL,» 29 Julio 2017. [En línea]. Available: <https://www.aboutespanol.com/cuidados-basicos-que-todo-perro-necesita-2641887>. [Último acceso: 23 Noviembre 2017].
- [53] tecnología, «área tecnología,» [En línea]. Available: <http://www.areatecnologia.com/materiales/acero-inoxidable.html>. [Último acceso: 23 Noviembre 2017].
- [54] tecnosmart, «tecnosmart,» [En línea]. Available: <http://www.tecnosmart.com.ec/v2/forza-ups-nt-502u-500va-250w-220v-240v4-4-salidas.html>. [Último acceso: 23 Noviembre 2017].
- [55] A. L. S. Iglesias, «ABOUTESPAÑOL,» 28 Julio 2016. [En línea]. Available: <https://www.aboutespanol.com/que-es-un-router-841387>. [Último acceso: 23 Noviembre 2017].
- [56] elinux.org, «elinux,» [En línea]. Available: <https://elinux.org/RPi-Cam-Web-Interface>. [Último acceso: 26 Febrero 2018].
- [57] Adafruit, «Adafruit,» [En línea]. Available: <https://www.adafruit.com/product/3100>. [Último acceso: 26 Febrero 2018].

- [58] FIMARCA, «FIMARCA,» [En línea]. Available: <http://fimarca.com/productos/plywood-tipos/>. [Último acceso: 23 Noviembre 2017].
- [59] Raspberry, «Raspberry,» 25 Julio 2017. [En línea]. Available: <https://www.raspberrypi.org/downloads/raspbian/>. [Último acceso: 23 Noviembre 2017].
- [60] M. Garcia, «Experto Animal,» [En línea]. Available: <https://www.expertoanimal.com/cantidad-de-comida-diaria-para-perros-20020.html>. [Último acceso: 23 Noviembre 2017].

# ANEXOS



## MEDIDAS DEL DISEÑO ESTRUCTURAL DE LA CASA

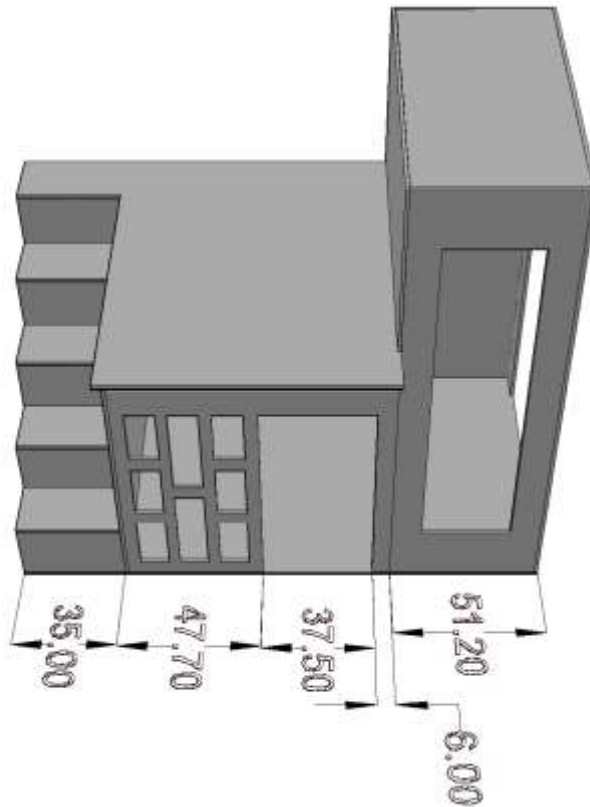


Figura 224 Medidas de la cara frontal de la estructura

Fuente: Propia

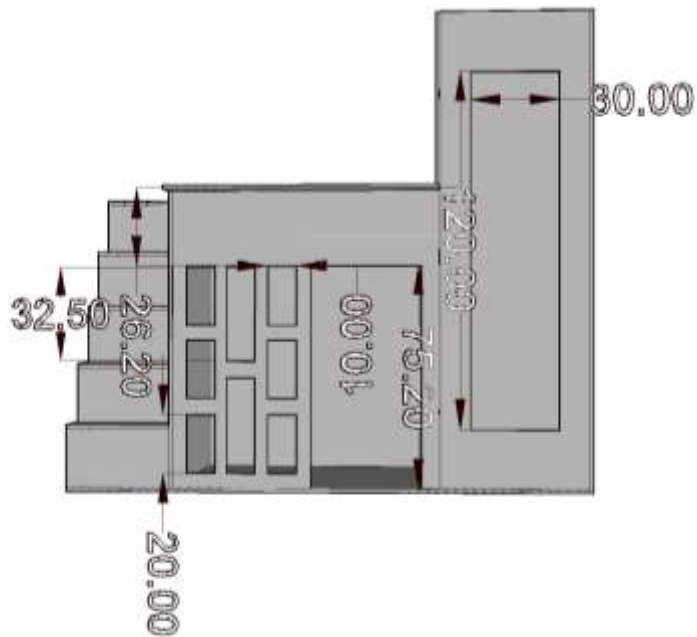


Figura 225 Medidas de la cara frontal de la estructura

Fuente: Propia

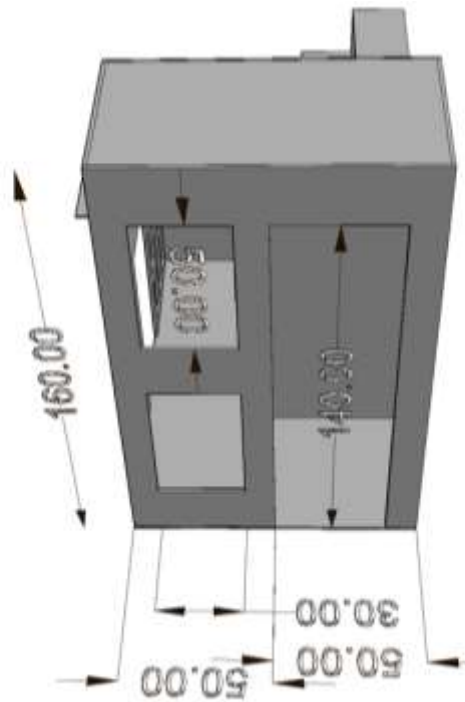


Figura 226 Medidas de la cara lateral derecha de la estructura

Fuente: Propia

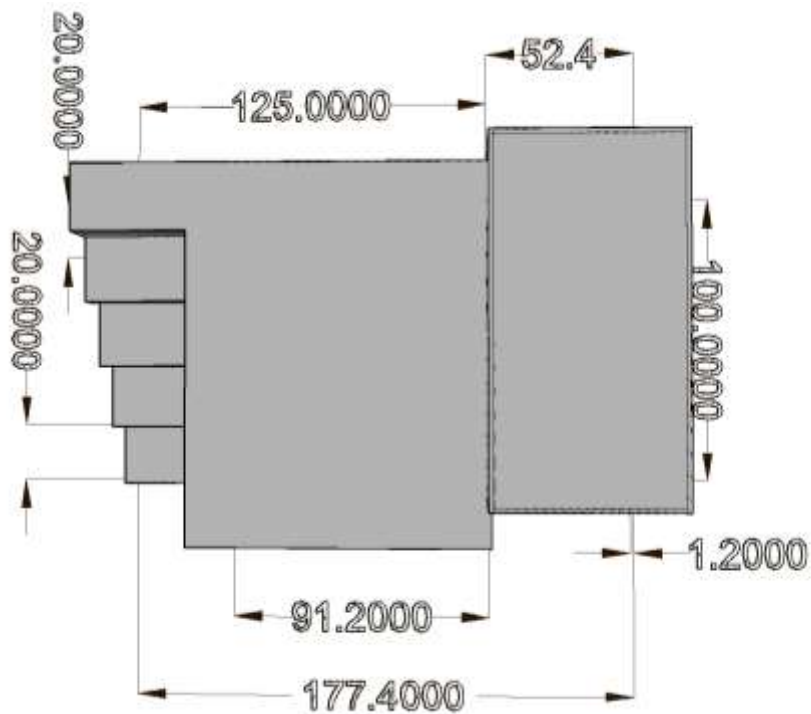


Figura 227 Medidas de la estructura visto de cara superior.

Fuente: Propia

A continuación, se adjunta código de los distintos archivos y programas del sistema:

## INDEX.HTML

```
<!DOCTYPE html>
<html >
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link rel='stylesheet prefetch' href='css/kickstrap.css'>
    <link rel="stylesheet" href="css/light_pop_up.css">
    <link rel="stylesheet" href="css/secondMenu.css">
    <link rel="stylesheet" href="css/style.css">
    <script>

        function closePopUp(radio) {
            var pop_up =
radio.parentNode.parentNode.parentNode;
            pop_up.style.display="none";// list[i] is a node with the
desired class name
        }
        function openPopUp(pop) {
            var pop_up = document.getElementById(pop);
            pop.style.display="block";// list[i] is a node with the
desired class name

            pop.style.outline="none";
            pop.style.background="rgba(0,0,0,0.8)";
            location.href='#'+pop.id;
        }
        function verSubOpcion(pop) {
            var div_nombre= 'div_'+pop.value;
            var div_subOpcion =
document.getElementById(div_nombre);

            for (var i = 0; i <
div_subOpcion.parentNode.childNodes.length; i++) {
                if
(div_subOpcion.parentNode.childNodes[i].className == "container") {

                    div_subOpcion.parentNode.childNodes[i].style.display="none";
                }
            }
            div_subOpcion.style.display="block";
        }
        function timeKeyPress()
        {
            $(this).on("keypress", function(e) {

                return e.metaKey || // cmd/ctrl
                e.which <= 0 || // arrow keys
            }
        }
    </script>
  </head>
  <body>
  </body>
</html>
```

```

        e.which == 8 || // delete key
        /[\].test(String.fromCharCode(e.which)); //
numbers
        });
    }
    function blockTime(b)
    {
        for (var i = 0; i < b.parentNode.childNodes.length; i++)
    {
        if (b.parentNode.childNodes[i].className ==
"al_time") {
            if (b.checked) {

                b.parentNode.childNodes[i].style="background:white";

                b.parentNode.childNodes[i].type="time";

                b.parentNode.childNodes[i].readOnly = false;
                    }
                    else
                    {

                b.parentNode.childNodes[i].style="background:transparent;width:100px;text-align:center;line-height: auto;";

                b.parentNode.childNodes[i].type="text";

                b.parentNode.childNodes[i].readOnly = true;
                    }
                }
            }
        }
        document.addEventListener("DOMContentLoaded",
function(event) {
            var partes_dia= document.getElementsByName("al");
            for (var i = 0; i < partes_dia.length; i++) {
                blockTime(partes_dia[i]);
            }
        });
    </script>
</head>

<body>
    <div id="mainTopNav" class="navbar navbar-static-top navbar-inverse frontpage">
        <div class="navbar-inner">
            <a class="btn btn-navbar" data-toggle="collapse" data-target="#mainTopNav .nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>

```

```

        <span class="icon-bar"></span>
    </a>
    <a href="#" class="brand" data-toggle="collapse" data-
target="#mainTopNav .nav-collapse">
        Logo
    </a>
    <div class="nav-collapse collapse">
        <ul class="nav pull-right">
            <li>
                <a href="#index.html"
alt="inicio">
                    <i class="icon-small icon-
external-link">Inicio
                        </i>
                </a>
            </li>
            <li>
                <a
onclick="openPopUp(instrucciones)" alt="instrucciones">
                    <i class="icon-small icon-
external-link">Instrucciones</i>
                </a>
            </li>
            <li>
                <a onclick="openPopUp(acerca)"
alt="acerca">
                    <i class="icon-small icon-
external-link">Acerca</i>
                </a>
            </li>
            <li>
                <a
onclick="openPopUp(sistema)" alt="sistema">
                    <i class="icon-small icon-
external-link">Sistema</i>
                </a>
            </li>
        </ul>
    </div>
</div>
</div>
<!-- Logotype, .header-image gets inline style for background image,
uploaded by user
=====
-->
<div class="utc-logo row">
    <div id="carousel-example" class="carousel slide" data-
ride="carousel">
        <ol class="carousel-indicators">

```

```

to="0" class="active"></li>
to="1"></li>
to="2"></li>
</ol>
<div class="carousel-inner">
  <div class="item active">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>iCAN</h3>
      <p>Casa domotizada para perros
</p>
    </div>
  </div>
  <div class="item">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>Las ventajas</h3>
      <p>ventajas</p>
    </div>
  </div>
  <div class="item">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>Que usa</h3>
      <p>Arduino/raspberry</p>
    </div>
  </div>
</div>
<a class="left carousel-control" href="#carousel-
example" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-
left"></span>
</a>
<a class="right carousel-control" href="#carousel-
example" data-slide="next">
  <span class="glyphicon glyphicon-chevron-
right"></span>
</a>
</div>
</div>
<div class="container" id="content">

```

```

<div class="s3-menu">
  <div id="s3-menu">
    <ul>
      <li style="background:#e75027">
        <a onclick=""
id="btnTemperatura">
          
          <br/>Temperatura
        </a>
      </li>
      <li style="background:#d13076">
        <a
onclick="openPopUp(iluminacion)">
          
          <br/>Iluminación
        </a>
      </li>
      <li style="background:#5587a2">
        <a
onclick="openPopUp(acceso)">
          
          <br/>Acceso
        </a>
      </li>
      <li style="background:#0c4c8a">
        <a
onclick="openPopUp(alimentacion)">
          
          <br/>Alimentación
        </a>
      </li>
      <li style="background:#76bf31">
        <a
onclick="location.href='Vigilancia.html#vigilancia'">
          
          <br/>Vigilancia
        </a>
      </li>
    </ul>
  </div>
</div>

```

```

<div class="lightbox" id="acceso">
  <div class="box" style="height: auto; ">

```

```

entradas</p>
    <p class="title">Acceso - Control de
    <div class="container">
        <div class="block" >
            <br/>
            <input id="A_open"
type="radio" name="acceso" value="abierto">Abierto</input><br>
        </div>
        <div class="block" >
            <br/>
            <input id="A_closed"
type="radio" name="acceso" value="cerrado">Cerrado</input>
        </div>
        <div class="block" >
            <br/>
            <input id="A_auto" type="radio"
name="acceso" value="automatico">Automático</input>
        </div>
        <br/>
        <input type="button" value="Salir"
onclick="closePopUp(this);"/>
    </div>
</div>
<div class="clear"></div>
</div>

```

```

<div class="lightbox" id="temperatura">
    <div class="box" style="height: auto;">
        <p class="title">Temperatura - Control de
ventilación</p>

```

```

    <div class="container">
        <div class="block"
style="width:200px;">
            Temp. Actual:
            <input value="22" type="text"
name="temp" id="idTemp" readonly="readonly" class="form" step="any" min="1"
max="40" style="width:50px; background: transparent; border: none;"/>
        </div>
        <br/>
        <div class="block">
            <input type="radio"
onclick="verSubOpcion(this);" name="vent_mode" value="vent_auto">
Automático</input>
        </div>
    </div>
</div>

```



```

                                <input type="radio"
onclick="verSubOpcion(this);" name="vent_mode" value="vent_man">
Manual</input>
                                </div>

                                <div class="container"
id="div_vent_auto" style="display:none;width: 80%;">
                                <div class="content"
style="height: auto; width:100%;">
                                <label
for="temp">Mantener temperatura en: </label>
                                <input type="number"
name="temp" id="txtemp" class="form" step="any" min="1" max="40"
style="width:50px;"/>
                                </div>
                                <br/>
                                <br/>
                                <input type="button" value="Ok"
id="btnTempguardar"/>
                                <input type="button"
value="Cancelar" onclick="closePopUp(this.parentNode);"/>
                                </div>
                                <div class="container"
id="div_vent_man" style="display:none">
                                <div style="clear:both"></div>
                                <div class="block" >
                                
                                <input type="radio"
name="vent" value="vent_on" id="btnTempon">On</input><br>
                                </div>
                                <div class="block" >
                                
                                <input type="radio"
name="vent" value="vent_off" id="btnTempoff">Off</input>
                                </div>
                                <br/>
                                <br/>
                                <input type="button"
value="Salir" onclick="closePopUp(this.parentNode);"/>
                                </div>
                                </div>
                                <div class="clear"></div>
                                </div>
                                </div>

                                <div class="lightbox" id="iluminacion">
                                <div class="box" style="height: auto; ">

```

```

    <p class="title">Iluminación - Control de
luces</p>

    <div class="container">
        <br/>
        <div class="block">
            <input type="radio"
onclick="verSubOpcion(this);" id="chkmode" name="il_mode" value="il_auto">
Automático</input>
        </div>
        <div class="block">
            <input type="radio"
onclick="verSubOpcion(this);" id="chkmode" name="il_mode" value="il_man">
Manual</input>
        </div>

        <div class="container" id="div_il_auto"
style="display:none;width: 80%;">
            <div class="block" >
                
                <input type="radio"
onclick="verSubOpcion(this);" id="chkmode_auto" name="il_mode_auto"
value="il_sensor">Sensor</input><br>
            </div>
            <div class="block" >
                
                <input type="radio"
onclick="verSubOpcion(this);" id="chkmode_auto" name="il_mode_auto"
value="il_horario">Horario</input>
            </div>
            <div class="container"
id="div_il_sensor" style="display:none;width: 80%;">
                <br/>
                <input type="button"
value="Ok" id="btnIluminacion_s" />
                <input type="button"
value="Cancelar" onclick="closePopUp(this.parentNode.parentNode);"/>
            </div>
            <div class="container"
id="div_il_horario" style="display:none; text-align:center">
                <div class="col-sm-6"
style="height:auto;text-align:center">
                    Desde:<br>
                    <input
type="time" id="txtIDesde" class="al_time" min="00:00" max="23:59"
onclick="timeKeyPress(this)">
                </div>
            </div>

```

```

style="height:auto;text-align:center">
Hasta:<br>
<input
type="time" id="txtIHasta" class="al_time" min="00:00" max="23:59"
onclick="timeKeyPress(this)">
</div>
<div
style="clear:both"></div>
<br/>
<input type="button"
value="Ok" id="btnIluminacion_h"
onclick="closePopUp(this.parentNode.parentNode);"/>
<input type="button"
value="Cancelar" onclick="closePopUp(this.parentNode.parentNode);"/>
</div>
</div>
<div class="container" id="div_il_man"
style="display:none">
<div style="clear:both">Luces
Ambiente 1</div>
<div class="block" >
<br/>
<input type="radio"
id="il_1_on" name="il_1" value="il_1_on"> On</input><br>
</div>
<div class="block" >
<br/>
<input type="radio"
id="il_1_off" name="il_1" value="il_1_off"> Off</input>
</div>
<br/>
<div style="clear:both">Luces
Ambiente 2</div>
<div class="block" >
<br/>
<input type="radio"
id="il_2_on" name="il_2" value="il_2_on"> On</input><br>
</div>
<div class="block" >
<br/>
<input type="radio"
id="il_2_off" name="il_2" value="il_2_off"> Off</input>
</div>
<br/>
<br/>

```

```

value="Salir" onclick="closePopUp(this.parentNode);"/>
</div>
</div>
<div class="clear"></div>
</div>
</div>

<div class="lightbox" id="alimentacion" >
  <div class="box" style="height: auto;">
    <p class="title">Alimentación - Dispensar
comida</p>

    <div class="container">
      <div
style="clear:both">Seleccione una de la siguientes opciones:</div>
        <div class="block" >
          <br/>
          <input type="radio"
id="PEQUEÑO" name="Porción" value="p_pequeño"><br> Razas pequeñas o
Cachorros</input><br>
          </div>
          <div class="block" >
            <br/>
            <input type="radio"
id="MEDIANO" name="Porción" value="p_mediano"><br> Razas medianas o
Jovenes</input><br>
            </div>
            <div class="block" >
              <br/>
              <input type="radio"
id="GRANDE" name="Porción" value="p_grande"><br> Razas grandes o
Adultos</input><br>
              </div>
            </div class="container" id="div_al_auto"
style="width:100%">
              <input type="button"
style="width: 125px;" value="Servir Porción" id="btnServir"/>
              <p style="padding-bottom: 10px"
/>
              <div class="content"
style="float:left;width:33.3%;height:auto;text-align:center">
                <br/>

```

```

                                <input
onclick="blockTime(this);" type="checkbox" class="al" id="chkAManana"
name="al" value="al_manana">Mañana</input><br>
                                <input id="txtAManana"
type="time" class="al_time" value="06:00" min="06:00" max="11:59"
onclick="timeKeyPress(this)">
                                </div>
                                <div class="content"
style="float:left;width:33.4%;height:auto;text-align:center">
                                <br/>
                                <input
onclick="blockTime(this);" type="checkbox" class="al" id="chkATarde" name="al"
value="al_tarde">Tarde</input><br>
                                <input id="txtATarde"
type="time" class="al_time" value="12:00" min="12:00" max="18:00"
onclick="timeKeyPress(this)">
                                </div>
                                <div class="content"
style="float:left;width:33.3%;height:auto;text-align:center">
                                <br/>
                                <input
onclick="blockTime(this);" type="checkbox" class="al" id="chkANoche"
name="al" value="al_noche">Noche</input><br>
                                <input type="time"
id="txtANoche" class="al_time" value="18:01" min="18:01" max="23:55"
onclick="timeKeyPress(this)">
                                </div>
                                <div style="clear:both"></div>
                                <br/>
                                <input type="button"
value="Guardar" id="btnAlimentacion"/>
                                <input type="button"
value="Cancelar" onclick="closePopUp(this.parentNode);"/>
                                </div>
                                </div>
                                <div class="clear"></div>
                                </div>
                                </div>
                                <div class="lightbox" id="instrucciones">
                                <div class="box" style="height: auto; ">
                                <p class="title">Instrucciones - Como usar</p>
                                <div class="container">
                                <div class="block" style="width:
100%;">
                                <p>

```

```

InstruccionesInstruccionesInstruccionesInstruccionesInstrucciones

InstruccionesInstruccionesInstruccionesInstruccionesInstrucciones
        </p>
    </div>
    <br/>
    <input type="button" value="Cerrar"
onclick="closePopUp(this);"/>
    </div>
    <div class="clear"></div>
</div>
</div>
<div class="lightbox" id="acerca">
    <div class="box" style="height: auto; ">
        <p class="title">Acerca - Quienes Somos</p>

        <div class="container">
            <div class="block" style="width:
100%;">
                <p>
                    Somos
                </p>
            </div>
            <br/>
            <input type="button" value="Cerrar"
onclick="closePopUp(this);"/>
        </div>
    <div class="clear"></div>
</div>
</div>
<div class="lightbox" id="sistema">
    <div class="box" style="height: auto; ">
        <p class="title">Opciones de sistema</p>
        <div class="container">
            <div class="block" style="width: auto; ">
                <br>
                <input type="button" value="APAGAR"
id="apaga" onclick="closePopUp(this.parentNode)"/>
                <input type="button"
value="REINICIAR" id="reinicia" onclick="closePopUp(this.parentNode)"/>
                <br>
                <br>
            </div>
            <br/>
            <input type="button" value="CERRAR"
onclick="closePopUp(this);"/>
        </div>
    </div>
</div>

```

```

</div>
<!--div class="container" style="width:100%; background:#1E5799;
color:white; margin: 10px auto 0; padding: 10px">
    fsdf
</div-->
<script src='js/jquery.min.js'></script>
<script src='js/bootstrap.min.js'></script>

<script>
    $(function(){
        $("#btnServir").on('click',function(){
            var array=[];
            if($("#PEQUEÑO").is(":checked")){
                array.push($("#PEQUEÑO").val());
            }
            if($("#MEDIANO").is(":checked")){
                array.push($("#MEDIANO").val());
            }
            if($("#GRANDE").is(":checked")){
                array.push($("#GRANDE").val());
            }
            $.post( "/controles/alimentacionpy.php", { accion:
'EJECUTAR',type:array} )
                .done(function( data ) {
                    alert( "Proceso ejecutado correctamente");
                })
                .fail(function (data) {
                    alert("Error de proceso "+data.message);
                });
        });

        $("#btnAlimentacion").on('click',function(){
            var array=[];
            if($("#chkAManana").is(":checked")){
                array.push($("#chkAManana").val());
            }
            if($("#chkATarde").is(":checked")){
                array.push($("#chkATarde").val());
            }
            if($("#chkANoche").is(":checked")){
                array.push($("#chkANoche").val());
            }
            if($("#PEQUEÑO").is(":checked")){
                array.push($("#PEQUEÑO").val());
            }
            if($("#MEDIANO").is(":checked")){
                array.push($("#MEDIANO").val());
            }
            if($("#GRANDE").is(":checked")){

```

```

        array.push($("#GRANDE").val());
    }
    $.post( "/controles/alimentacion.php", { m: $("#txtAManana").val(),
    t: $("#txtATarde").val(),n: $("#txtANoche").val(),type:array} )
    .done(function( data ) {
        alert( "Proceso ejecutado correctamente");
    })
    .fail(function (data) {
        alert("Error de proceso "+data.message);
    });
    closePopUp(this.parentNode);
    });

    $("#btnIluminacion_s").on('click',function(){
        $.post( "/controles/iluminacions.php", {
accion: 'EJECUTAR'})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
        closePopUp(this.parentNode.parentNode);
        });

    $("#btnIluminacion_h").on('click',function(){
        $.post( "/controles/iluminacionh.php", {
D: $("#txtIDesde").val(),
H: $("#txtIHasta").val()})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
        closePopUp(this.parentNode.parentNode);
        });

    $("#il_1_on").on('click',function){
        $.post( "/controles/ilumanual.php", {
estado: '1'})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });
    });

    $("#il_1_off").on('click',function(){

```



```

                                $.post( "/controles/ilummanual.php", {
estado: '2'})
                                .done(function( data ) {
                                    alert( "Proceso ejecutado correctamente");
                                })
                                .fail(function (data) {
                                    alert("Error de proceso "+data.message);
                                });
});

$("#il_2_on").on('click',function(){
                                $.post( "/controles/ilummanual.php", {
estado: '3'})
                                .done(function( data ) {
                                    alert( "Proceso ejecutado correctamente");
                                })
                                .fail(function (data) {
                                    alert("Error de proceso "+data.message);
                                });
});

$("#il_2_off").on('click',function(){
                                $.post( "/controles/ilummanual.php", {
estado: '4'})
                                .done(function( data ) {
                                    alert( "Proceso ejecutado correctamente");
                                })
                                .fail(function (data) {
                                    alert("Error de proceso "+data.message);
                                });
});

$("#A_open").on('click',function(){
                                $.post( "/controles/acceso.php", {
acceso: 'ABRIR'})
                                .done(function( data ) {
                                    alert( "Proceso ejecutado correctamente");
                                })
                                .fail(function (data) {
                                    alert("Error de proceso "+data.message);
                                });
});

$("#A_closed").on('click',function(){
                                $.post( "/controles/acceso.php", {
acceso: 'CERRAR'})
                                .done(function( data ) {
                                    alert( "Proceso ejecutado correctamente");
                                })

```

```

        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });

    $("#A_auto").on('click',function(){
        $.post( "/controles/acceso.php", {
acceso: 'AUTO'})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });

    $("#apaga").on('click',function(){
    $.post( "/controles/apaga.php", { accion: 'EJECUTAR'})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });

    $("#reinicia").on('click',function(){
    $.post( "/controles/reinicia.php", { accion: 'EJECUTAR'})
        .done(function( data ) {
            alert( "Proceso ejecutado correctamente");
        })
        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });

    $("#btnTemperatura").on('click',function(){

        $.post( "/controles/temperaturapy.php", { accion:
'EJECUTAR'})
            .done(function( data ) {

                $("#idTemp").val(data.message);
                $('#temperatura').show();
            })
    });

```

```

        .fail(function (data) {
            alert("Error de proceso "+data.message);
        });
    });

    $("#btnTempon").on('click',function(){

$.post( "/controles/temperaturaon.php", { accion: 'EJECUTAR'})
    .done(function( data ) {
        alert( "Proceso ejecutado correctamente");
    })
    .fail(function (data) {
        alert("Error de proceso "+data.message);
    });

});

    $("#btnTempoff").on('click',function(){

$.post( "/controles/temperaturaoff.php", { accion: 'EJECUTAR'})
    .done(function( data ) {
        alert( "Proceso ejecutado correctamente");
    })
    .fail(function (data) {
        alert("Error de proceso "+data.message);
    });

});

    $("#btnTempguardar").on('click',function(){
        $.post(
"/controles/temperaturaseteo.php", { T: $("#txtemp").val()})
    .done(function( data ) {
        alert( "Proceso ejecutado correctamente");
    })
    .fail(function (data) {
        alert("Error de proceso "+data.message);
    });
    closePopUp(this.parentNode);
    });

});
</script>
</body>

</html>

```

## VIGILANCIA.HTML

```
<!DOCTYPE html>
<html >
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link rel='stylesheet prefetch' href='css/kickstrap.css'>
    <link rel="stylesheet" href="css/light_pop_up.css">
    <link rel="stylesheet" href="css/secondMenu.css">
    <link rel="stylesheet" href="css/style.css">
    <script>

      function closePopUp(radio) {
        var pop_up =
radio.parentNode.parentNode.parentNode;
        pop_up.style.display="none";// list[i] is a node with the
desired class name
      }
      function openPopUp(pop) {
        var pop_up = document.getElementById(pop);
        pop.style.display="block";// list[i] is a node with the
desired class name

        pop.style.outline="none";
        pop.style.background="rgba(0,0,0,0.8)";
        location.href='#'+pop.id;
      }
      function verSubOpcion(pop) {
        var div_nombre= 'div_'+pop.value;
        var div_subOpcion =
document.getElementById(div_nombre);

        for (var i = 0; i <
div_subOpcion.parentNode.childNodes.length; i++) {
          if
(div_subOpcion.parentNode.childNodes[i].className == "container") {
            div_subOpcion.parentNode.childNodes[i].style.display="none";;
          }
        }
        div_subOpcion.style.display="block";
      }
      function timeKeyPress()
      {
        $(this).on("keypress", function(e) {

          return e.metaKey || // cmd/ctrl
          e.which <= 0 || // arrow keys
          e.which == 8 || // delete key
        }
      }
    </script>
  </head>
  <body>
  </body>
</html>
```

```

numbers
    /[]/.test(String.fromCharCode(e.which)); //
    });
    }
    function blockTime(b)
    {
        for (var i = 0; i < b.parentNode.childNodes.length; i++)
    {
        if (b.parentNode.childNodes[i].className ==
"al_time") {
            if (b.checked) {

                b.parentNode.childNodes[i].style="background:white";

                b.parentNode.childNodes[i].type="time";

                b.parentNode.childNodes[i].readOnly = false;
                }
                else
                {

                b.parentNode.childNodes[i].style="background:transparent;width:100px;text-
align:center;line-height: 34px;";

                b.parentNode.childNodes[i].type="text";

                b.parentNode.childNodes[i].readOnly = true;
                }
            }
        }
    }
    document.addEventListener("DOMContentLoaded",
function(event) {
        var partes_dia= document.getElementsByName("al");
        for (var i = 0; i < partes_dia.length; i++) {
            blockTime(partes_dia[i]);
        }
    });
</script>
</head>

<body>
    <div id="mainTopNav" class="navbar navbar-static-top navbar-
inverse frontpage">
        <div class="navbar-inner">
            <a class="btn btn-navbar" data-toggle="collapse" data-
target="#mainTopNav .nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>

```

```

        </a>
        <a href="#" class="brand" data-toggle="collapse" data-
target="#mainTopNav .nav-collapse">
            Logo
        </a>
        <div class="nav-collapse collapse">
            <ul class="nav pull-right">
                <li>
                    <a
onclick="location.href='index.html'" alt="inicio">
                        <i class="icon-small icon-
external-link">Inicio
                    </i>
                </a>
                </li>
                <li>
                    <a
onclick="openPopUp(instrucciones)" alt="instrucciones">
                        <i class="icon-small icon-
external-link">Instrucciones</i>
                    </a>
                </li>
                <li>
                    <a onclick="openPopUp(acerca)"
alt="acerca">
                        <i class="icon-small icon-
external-link">Acerca</i>
                    </a>
                </li>
                <li>
                    <a
onclick="openPopUp(sistema)" alt="sistema">
                        <i class="icon-small icon-
external-link">Sistema</i>
                    </a>
                </li>
            </ul>
        </div>
    </div>
</div>
<!-- Logotype, .header-image gets inline style for background image,
uploaded by user
=====
-->
<div class="utc-logo row">
    <div id="carousel-example" class="carousel slide" data-
ride="carousel">
        <ol class="carousel-indicators">

```

```

to="0" class="active"></li>
to="1"></li>
to="2"></li>
</ol>
<div class="carousel-inner">
  <div class="item active">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>El producto</h3>
      <p>que es</p>
    </div>
  </div>
  <div class="item">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>Las ventajas</h3>
      <p>ventajas</p>
    </div>
  </div>
  <div class="item">
    <a href="#"></a>
    <div class="carousel-caption">
      <h3>Que usa</h3>
      <p>Arduino/raspberry</p>
    </div>
  </div>
</div>
<a class="left carousel-control" href="#carousel-
example" data-slide="prev">
  <span class="glyphicon glyphicon-chevron-
left"></span>
</a>
<a class="right carousel-control" href="#carousel-
example" data-slide="next">
  <span class="glyphicon glyphicon-chevron-
right"></span>
</a>
</div>
</div>
<div class="container" id="content">
  <div id="vigilancia" style="text-align: center;">

```

```

src="/xCAM/min.php">
    <iframe width="800" height="600"
    </iframe>
</div>
<input type="button" onclick="location.href='index.html'"
style="position: fixed; bottom: 10px; right: 20px; padding: 8px; background:
rgba(100,200,10,0.9); border: none;" value="ir al menú">

<div class="lightbox" id="instrucciones">
    <div class="box" style="height: auto; ">
        <p class="title">Instrucciones - Como usar</p>

        <div class="container">
            <div class="block" style="width:
100%;">
                <p>
                    Instrucciones
                </p>
            </div>
            <br/>
            <input type="button" value="Cerrar"
onclick="closePopUp(this);"/>
        </div>
        <div class="clear"></div>
    </div>
</div>

<div class="lightbox" id="acerca">
    <div class="box" style="height: auto; ">
        <p class="title">Acerca - Quienes Somos</p>

        <div class="container">
            <div class="block" style="width:
100%;">
                <p>
                    Somos
                </p>
            </div>
            <br/>
            <input type="button" value="Cerrar"
onclick="closePopUp(this);"/>
        </div>
        <div class="clear"></div>
    </div>
</div>

<div class="lightbox" id="sistema">
    <div class="box" style="height: auto; ">
        <p class="title">Opciones de sistema</p>

```



```

        <div class="container">
            <div class="block" style="width: auto;">
                <br>
                <input type="button" value="APAGAR"
id="apaga" onclick="closePopUp(this.parentNode)"/>
                <input type="button"
value="REINICIAR" id="reinicia" onclick="closePopUp(this.parentNode)"/>
                <br>
                <br>
            </div>
        <br/>
        <input type="button" value="CERRAR"
onclick="closePopUp(this);"/>
    </div>
</div>
<!--div class="container" style="width:100%; background:#1E5799;
color:white; margin: 10px auto 0; padding: 10px">
    fsdf
</div-->
<script src='js/jquery.min.js'></script>
<script src='js/bootstrap.min.js'></script>

<script>

$(function(){

$("#apaga").on('click',function(){
$.post( "/controles/apaga.php", { accion: 'EJECUTAR'})
.done(function( data ) {
    alert( "Proceso ejecutado correctamente");
})
.fail(function (data) {
    alert("Error de proceso "+data.message);
});

});

$("#reinicia").on('click',function(){
$.post( "/controles/reinicia.php", { accion: 'EJECUTAR'})
.done(function( data ) {
    alert( "Proceso ejecutado correctamente");
})
.fail(function (data) {
    alert("Error de proceso "+data.message);
});

});

});

```

```

        </script>
    </body>

</html>

```

## PROGRAMAS PHP DEL DIRECTORIO CONTROLES:

### ACCESO.PHP

```

<?php
include 'libreria.php';

try{

    $var = $_POST['acceso'];
    $file=fopen(__DIR__.'../files/ACCESO.txt','r+');
    fputs($file,"0\n");
    if ($var == 'ABRIR') {
        $resultType['modo_acceso']=1;
    }
    if($var=='CERRAR'){
        $resultType['modo_acceso']=2;
    }
    if($var=='AUTO'){
        $resultType['modo_acceso']=3;
    }
    fputs($file,$resultType['modo_acceso']);
    fclose($file);
    $file=fopen(__DIR__.'../files/ACCESO.txt','r+');
    fputs($file,"1");
    fclose($file);
    echo json_response( 'ok',200);

}catch (Exception $ex){
    json_response( 'Error interno '.$ex->getMessage(),500);
}

```

### ALIMENTACION.PHP

```

<?php

include 'libreria.php';

try{

    $txtAManana=explode(":",$_POST['m']);
    $txtATarde=explode(":",$_POST['t']);
    $txtANoche=explode(":",$_POST['n']);

```

```

$type=$_POST['type'];
    $PWM=90;
    $TME=3;
$resultType=['almanana'=>0,'altarde'=>0,'alnoche'=>0];
$file=fopen(__DIR__.'../files/HDOSIFICADOR.txt','r+');
fputs($file,"0\n");
foreach ($type as $item){
    if($item=='al_manana'){
        $resultType['almanana']=1;
    }
    if($item=='al_tarde'){
        $resultType['altarde']=1;
    }
    if($item=='al_noche'){
        $resultType['alnoche']=1;
    }
    if($item=='p_pequeño'){
        $PWM=30;
        $TIME=1;
    }
    if($item=='p_mediano'){
        $PWM=45;
        $TIME=1;
    }
    if($item=='p_grande'){
        $PWM=60;
        $TIME=1;
    }
}

}
fputs($file,$TIME."\n");
fputs($file,$PWM."\n");

fputs($file,$resultType['almanana']."\n");
fputs($file,$txtAManana[0]."\n");
fputs($file,$txtAManana[1]."\n");

fputs($file,$resultType['altarde']."\n");
fputs($file,$txtATarde[0]."\n");
fputs($file,$txtATarde[1]."\n");

fputs($file,$resultType['alnoche']."\n");
fputs($file,$txtANoche[0]."\n");
fputs($file,$txtANoche[1]);

fclose($file);
$file=fopen(__DIR__.'../files/HDOSIFICADOR.txt','r+');
fputs($file,"1");
fclose($file);
echo json_response( 'ok',200);

```

```

} catch (Exception $ex){
    json_response( 'Error interno '.$ex->getMessage(),500);
}

```

### **ALIMENTACIONPY.PHP**

```

<?php
include 'libreria.php';

try {
    $result=[];
    $var = $_POST['accion'];
    $type=$_POST['type'];
    $PWM=30;
    $TME=1;
    if ($var == 'EJECUTAR') {
        foreach ($type as $item){
            if($item=='p_pequeño'){
                $PWM=30;
                $TIME=1;
            }

            if($item=='p_mediano'){
                $PWM=45;
                $TIME=1;
            }

            if($item=='p_grande'){
                $PWM=60;
                $TIME=1;
            }

        }

        $file=fopen(__DIR__.'../files/HDOSIFICADOR.txt','r+');
        fputs($file,"0\n");
        fputs($file,$TIME."\n");
        fputs($file,$PWM."\n");
        fclose($file);
        exec("sudo python ".PATH_PY."DOSI-MANUAL.py",$result);
    }

    echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno ' . $ex->getMessage(),500);
}

```

### **APAGA.PHP**

```

<?php
include 'libreria.php';

```

```

try {
    $result=[];
    $var = $_POST['accion'];
    if ($var == 'EJECUTAR') {
        exec("sudo poweroff",$result);
    }
    echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno ' . $ex->getMessage(),500);
}

```

## **ILUMANUAL.PHP**

```
<?php
```

```
include 'libreria.php';
```

```
try{
```

```
    $result=[];
```

```
    $item=$_POST['estado'];
```

```
    if($item=='1'){
```

```
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
```

```
        fputs($file,"0\n");
```

```
        fputs($file,"1\n");
```

```
        fclose($file);
```

```
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
```

```
        fputs($file,"1\n");
```

```
        fclose($file);
```

```
        exec("sudo python ".PATH_PY."LUZ_MANUAL.py 1",$result);
```

```
    }
```

```
    if($item=='2'){
```

```
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
```

```
        fputs($file,"0\n");
```

```
        fputs($file,"1\n");
```

```
        fclose($file);
```

```
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
```

```
        fputs($file,"1\n");
```

```

        fclose($file);
        exec("sudo python ".PATH_PY."LUZ_MANUAL.py 2",$result);
    }
    if($item=='3'){
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
        fputs($file,"0\n");
        fputs($file,"1\n");
        fclose($file);
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
        fputs($file,"1\n");
        fclose($file);
        exec("sudo python ".PATH_PY."LUZ_MANUAL.py 3",$result);
    }
    if($item=='4'){
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
        fputs($file,"0\n");
        fputs($file,"1\n");
        fclose($file);
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
        fputs($file,"1\n");
        fclose($file);
        exec("sudo python ".PATH_PY."LUZ_MANUAL.py 4",$result);
    }

    echo json_response( 'ok',200);
} catch (Exception $ex){
    json_response( 'Error interno '.$ex->getMessage(),500);
}

```

## **ILUMINACIONH.PHP**

```

<?php
include 'libreria.php';

```

```

try{

    $txtIDesde=explode(":",$_POST['D']);
    $txtIHasta=explode(":",$_POST['H']);
    $file=fopen(__DIR__.'../files/LUCES.txt','r+');
    fputs($file,"0\n");
    fputs($file,"0\n");
    fputs($file,"1\n");
    fputs($file,$txtIDesde[0]."\n");
    fputs($file,$txtIDesde[1]."\n");
    fputs($file,$txtIHasta[0]."\n");
    fputs($file,$txtIHasta[1]);

    fclose($file);
    $file=fopen(__DIR__.'../files/LUCES.txt','r+');
    fputs($file,"1");
    fclose($file);
    echo json_response( 'ok',200);
}catch (Exception $ex){
    json_response( 'Error interno '.$ex->getMessage(),500);
}

```

## **ILUMINACION.S.PHP**

```

<?php
include 'libreria.php';

try{

    $var = $_POST['accion'];
    if ($var == 'EJECUTAR') {
        $file=fopen(__DIR__.'../files/LUCES.txt','r+');
        fputs($file,"0\n");
        fputs($file,"0\n");
    }
}

```

```

    fputs($file,"0");
    fclose($file);
    $file=fopen(__DIR__.'../files/LUCES.txt','r+');
    fputs($file,"1");
    fclose($file);
    echo json_response( 'ok',200);
}
}catch (Exception $ex){
    json_response( 'Error interno '.$ex->getMessage(),500);
}

```

## **LIBRERÍA.PHP**

```

<?php
/**
 * Created by PhpStorm.
 * User: blacksato
 * Date: 20/07/17
 * Time: 18:26
 */
function json_response($message = null, $code = 200)
{
    // clear the old headers
    header_remove();
    // set the actual code
    http_response_code($code);
    // set the header to make sure cache is forced
    header("Cache-Control: no-transform,public,max-age=300,s-maxage=900");
    // treat this as json
    header('Content-Type: application/json');
    $status = array(
        200 => '200 OK',
        400 => '400 Bad Request',
        422 => 'Unprocessable Entity',

```



```

        500 => '500 Internal Server Error'
    );
    // ok, validation error, or failure
    header('Status: '.$status[$code]);
    // return the encoded json
    return json_encode(array(
        'status' => $code < 300, // success or not?
        'message' => $message
    ));
}
define('PATH_PY','/var/www/data_py/');

```

### **REINICIA.PHP**

```

<?php
include 'libreria.php';

try {
    $result=[];
    $var = $_POST['accion'];
    if ($var == 'EJECUTAR') {
        exec("sudo reboot",$result);
    }
    echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno ' . $ex->getMessage(),500);
}

```

### **TEMPERATURAOFF.PHP**

```

<?php
include 'libreria.php';

try {

```

```

$result=[];
$var = $_POST['accion'];
if ($var == 'EJECUTAR') {
    //exec("python ".PATH_PY."DOSI-MANUAL.py > /dev/null &");
    exec("python ".PATH_PY."TEMP_SERIAL.py 300",$result);
}
echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno ' . $ex->getMessage(),500);
}

```

### **TEMPERATURAON.PHP**

```

<?php
include 'libreria.php';

try {
    $result=[];
    $var = $_POST['accion'];
    if ($var == 'EJECUTAR') {
        //exec("python ".PATH_PY."DOSI-MANUAL.py > /dev/null &");
        exec("python ".PATH_PY."TEMP_SERIAL.py 200",$result);
    }
    echo json_response( 'ok',200);
} catch (Exception $ex) {
    json_response('Error interno ' . $ex->getMessage(),500);
}

```

### **TEMPERATURAPY.PHP**

```

<?php
include 'libreria.php';

try {

```

```

$result=[];
$var = $_POST['accion'];
if ($var == 'EJECUTAR') {
    //exec("python ".PATH_PY."DOSI-MANUAL.py > /dev/null &");
    exec("python ".PATH_PY."TEMP_SERIAL.py 400",$result);
}
echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno '. $ex->getMessage(), 500);
}

```

### **TEMPERATURASETEO.PHP**

```

<?php
include 'libreria.php';
try {
    $result=[];
    $Tempset = $_POST['T'];
    exec("python ".PATH_PY."TEMP_SERIAL.py $Tempset",$result);
    sleep(1);
    exec("python ".PATH_PY."TEMP_SERIAL.py 100",$result);
    echo json_response( $result[0],200);
} catch (Exception $ex) {
    json_response('Error interno '. $ex->getMessage(),500);
}

```

### **SE ADJUNTA PROGRAMAS PYTHON DEL DIRECTORIO DATA\_PY**

#### **ACCESO.PY**

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
import push
import time

```

```

import threading
import RPi.GPIO as GPIO, time
#DECLARACION DE CONSTANTES DE PUERTOS GPIO
SE= 5
SI = 6
ABRE = 19
CIERRA = 26
FCABRE = 20
FCCIERRA = 21
#DECLARACION DE VARIABLES E INICIALIZACION
auto = 0
PROTECCION=0
#PARAMETRIZACIÓN DE PUERTOS GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(SE,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(SI,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(PROTECCION,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(FCABRE,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(FCCIERRA,GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(ABRE, GPIO.OUT)
GPIO.setup(CIERRA, GPIO.OUT)
GPIO.output(ABRE, False)
GPIO.output(CIERRA, False)

#PROGRAMA TIPO THREAD QUE REALIZA EL CONTROL AUTOMÀTICO
DE LA PUERTA
class automatico(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.keep_running = True
        print("SISTEMA DE PUERTAS AUTOMATICO INICIADO!")

    def run(self):

```

```

try:
    while self.keep_running:
        time.sleep(0.5)
        if GPIO.input(SI)== True or GPIO.input(SE)== True:
            if GPIO.input(FCABRE) == False:
                #print("MA - ABRIENDO")
                abrir()
            time.sleep(20)
        else:
            if GPIO.input(FCCIERRA) == False:
                #print("MA - CERRANDO")
                cerrar()
    except:
        print("ERROR DE AUTOMATICO")
        return
def just_die(self):
    self.keep_running = False

def abrir():
    while GPIO.input(FCABRE) == False:
        GPIO.output(CIERRA, False)
        GPIO.output(ABRE, True)
        time.sleep(0.4)
        print("ABRIENDO")
    print("PUERTA ABIERTA")
    GPIO.output(ABRE, False)
    GPIO.output(CIERRA, False)
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","LA PUERTA HA SIDO ABIERTA")
def cerrar():
    while GPIO.input(FCCIERRA) == False:
        time.sleep(0.3)
        if GPIO.input(SI)== True or GPIO.input(SE)== True:
            GPIO.output(CIERRA, False)

```

```

time.sleep(0.3)
while GPIO.input(FCABRE) == False:
    time.sleep(0.3)
    GPIO.output(CIERRA, False)
    GPIO.output(ABRE, True)
    print("SE ABRE POR OBSTACULO")
    GPIO.output(ABRE, False)
    time.sleep(15)
    #print("ESPERANDO PARA CERRAR")
else:
    GPIO.output(ABRE, False)
    GPIO.output(CIERRA, True)
    #time.sleep(0.4)
    #print("CERRANDO")
print("TERMINE DE CERRAR")
GPIO.output(ABRE, False)
GPIO.output(CIERRA, False)
push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN", "LA PUERTA SE HA CERRADO")
#PROGRAMA INICIAL
abrir()
#print("termine de abrir")

try:

while True:
if GPIO.input(PROTECCION) == True:
infile = open('/var/www/files/ACCESO.txt', 'r')
infile.seek(0)
a=infile.readline()
b=infile.readline()
infile.close()
time.sleep(0.6)

```

```

if a=="1\n":
    if b=="1":
        print("abre")
        if auto == 1:
            Automatico.just_die()
            auto = 0
        outfile = open('/var/www/files/ACCESO.txt', 'r+')
        outfile.write('0\n')
        outfile.close()
        abrir()
    if b=="2":
        print("Cierra")
        if auto == 1:
            Automatico.just_die()
            auto = 0
        outfile = open('/var/www/files/ACCESO.txt', 'r+')
        outfile.write('0\n')
        outfile.close()
        cerrar()

    if b == "3":
        print("auto")
        if auto == 1:
            Automatico.just_die()
        Automatico = automatico()
        auto = 1
        #print("AUTO")
        outfile = open('/var/www/files/ACCESO.txt', 'r+')
        outfile.write('0\n')
        outfile.close()
        Automatico.start()

else:
    print("protegete")

```

```

if GPIO.input(FCABRE) == False:
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN", "SE HA IDO LA ENERGÍA ELECTRICA")
    print("protegete")
    abrir()
    #print("protegete")

except:
    print("SALIENDO DEL SISTEMA")
    GPIO.cleanup()
    if auto == 1:
        Automatico.just_die()
        auto = 0

```

## **DOSI-MANUAL.PY**

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-
import push
import RPi.GPIO as GPIO, time
import string
from time import sleep

#Setup
#DEFINIMOS CONSTANTES DE LOS PUERTOS GPIO
PPWM= 24
ARRANQUE=2
VOZ=23
GPIO_TRIGGER = 9 #Usamos el pin GPIO 9 como TRIGGER
GPIO_ECHO = 11 #Usamos el pin GPIO 11 como ECHO
LED= 25
#DEFINIMOS VARIABLES Y SU VALOR INICIAL
PW=90
Time=2

```



```

#PARAMETRIZACION DE LOS PUERTOS
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(GPIO_TRIGGER,GPIO.OUT) #Configuramos Trigger como salida
GPIO.setup(VOZ, GPIO.OUT) #PIN DEL REPRODUCTOR DE VOZ
GPIO.setup(PPWM, GPIO.OUT) #PUERTO DEL MOTOR
GPIO.setup(ARRANQUE, GPIO.OUT)
GPIO.setup(LED, GPIO.OUT) #PUERTO DEL LED
GPIO.setup(GPIO_ECHO,GPIO.IN) #Configuramos Echo como entrada
GPIO.output(GPIO_TRIGGER,False) #Ponemos el pin 25 como LOW
GPIO.output(LED, False) #Pin del LED LOW
GPIO.output(VOZ, False) #Pin de VOZ LOW
GPIO.output(ARRANQUE, False)
GPIO.output(PPWM, False)
PWM = GPIO.PWM(PPWM, 100) #INICIALIZACION DEL PWM
#PROGRAMA DEL SENSOR DE DISTANCIA:
def Nivel():
    GPIO.output(GPIO_TRIGGER,True) #Enviamos un pulso de ultrasonidos
    time.sleep(0.00001) #Una pequeñña pausa
    GPIO.output(GPIO_TRIGGER,False) #Apagamos el pulso
    start = time.time() #Guarda el tiempo actual mediante time.time()
    while GPIO.input(GPIO_ECHO)==0: #Mientras el sensor no reciba señal
        start = time.time() #Mantenemos el tiempo actual mediante time.time()
    while GPIO.input(GPIO_ECHO)==1: #Si el sensor recibe señal
        stop = time.time() #Guarda el tiempo actual mediante time.time() en otra
variable
    elapsed = stop-start #Obtenemos el tiempo transcurrido entre envío y
recepción
    distance = (elapsed*34300)/2 #Distancia es igual a tiempo por velocidad partido
por 2  $D = (T \times V)/2$ 
    #print(distance) #Devolvemos la distancia (en centímetros) por pantalla
    time.sleep(1) #Pequeña pausa para no saturar el procesador de la
Raspberry
    if distance >= 20:

```

```

GPIO.output(LED,True)#SI LA COMIDA ESTÁ POR DEBAJO DEL
NIVEL ENVÍA NOTIFICACIÓN Y ENCIENDE LED
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","QUEDA POCO ALIMENTO, POR FAVOR RECARGUE")
    print("SE PRENDIO")
else:
    GPIO.output(LED,False) #SI LA COMIDA ESTÀ POR ENCIMA DEL
NIVEL APAGA EL LED.
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","SE HA SERVIDO UNA PORCION")
    print("NO SE PRENDIO")
#Programa principal
#LECTURA DE VARIABLES
infile = open('/var/www/files/HDOSIFICADOR.txt', 'r')
infile.seek(0)
T=int(infile.readline())
Time=int(infile.readline())
PW=int(infile.readline())
#EJECUION DE PROGRAMA
GPIO.output(VOZ,True)
time.sleep(10)
GPIO.output(VOZ,False)
#GPIO.output(ARRANQUE,True)
#time.sleep(0.5)
GPIO.output(ARRANQUE,False)
time.sleep(0.5)
PWM.start(100)
#PWM.ChangeDutyCycle(100)
time.sleep(0.8)
PWM.ChangeDutyCycle(PW)
#GPIO.output(VOZ,False)
time.sleep(Time)
PWM.stop()
Nivel()

```

## LUZ-DOSI.PY

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import push
import time
import os
import threading
import RPi.GPIO as GPIO, time
from datetime import datetime
#Setup
#Definimos constantes para puertos GPIO
Luz1=27
Luz2=22
PPWM=24
VOZ=23
LDR=17
GPIO_TRIGGER = 9
GPIO_ECHO = 11
LED= 25
#Definimos variables y sus valores iniciales
PW=80
Time=1
Luces = 0
Activo = 0
flag=0
#Configuración de los puertos GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(Luz1, GPIO.OUT)## GPIO SALIDA LAMPARA1
GPIO.setup(Luz2, GPIO.OUT)## GPIO SALIDA LAMPARA2
GPIO.setup(PPWM, GPIO.OUT)## GPIO DOSIFICADOR
GPIO.setup(LED, GPIO.OUT) ## PWM DEL MOTOR
GPIO.setup(VOZ, GPIO.OUT)#se da estado off al puerto del led
```

```

GPIO.setup(GPIO_TRIGGER,GPIO.OUT) #Configuramos Trigger como salida
GPIO.setup(LDR,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)#Configuramos la
entrada del LDR
GPIO.setup(GPIO_ECHO,GPIO.IN)#Configuramos Echo como entrada
GPIO.output(Luz1, True)#Lógica inversa
GPIO.output(Luz2, True)#Lógica inversa
GPIO.output(LED, False)#se da estado off al puerto del led
GPIO.output(VOZ, False)#se da estado off al puerto del led
GPIO.output(PPWM, False)#Se da estado off al puerto del dosificador
GPIO.output(GPIO_TRIGGER,False)#Ponemos el pin 25 como LOW
PWM = GPIO.PWM(PPWM, 100)

```

#Programa tipo thread que controla iluminación

```

class Iluminacion(threading.Thread):
    def __init__(self, hour_on, minute_on, hour_off, minute_off):
        super(Iluminacion, self).__init__()
        self.hours_on = str(int(hour_on))
        self.minutes_on = str(int(minute_on))
        self.hours_off = str(int(hour_off))
        self.minutes_off = str(int(minute_off))
        self.keep_running = True

    def run(self):
        horaon = datetime.strptime(self.hours_on+":"+self.minutes_on+":00",
"%X").time()
        horaoff = datetime.strptime(self.hours_off+":"+self.minutes_off+":00",
"%X").time()

        try:

            while self.keep_running:
                time.sleep(1)
                hora_act = datetime.now().time()
                if (horaoff >= horaon):

```

```

        if horaon <= hora_act <= horaoff:
            #print("LUCES ENCENDIDAS!2")
            Enciende()
        else:
            #print("LUCES APAGADAS!1")
            Apaga()
    else:

        if horaoff <= hora_act <= horaon:
            #print("LUCES APAGADAS!3")
            Apaga()
        else:
            #print("LUCES ENCENDIDAS!4")
            Enciende()

    # time.sleep(60)
except:
    #return
    print("ERROR EN SISTEMA DE ILUMINACION")
def just_die(self):
    self.keep_running = False

#Programa tipo thread que controla alimentación
class Dosificador(threading.Thread):
    def __init__(self,time,pw,a1, hours1,
minutes1,a2,hours2,minutes2,a3,hours3,minutes3):
        super(Dosificador, self).__init__()
        self.Time = int(time)
        self.PW = int(pw)
        self.a1 = int(a1)
        self.hours1 = int(hours1)
        self.minutes1= int(minutes1)
        self.a2 = str(a2)
        self.hours2 = int(hours2)

```

```

self.minutes2 = int(minutes2)
self.a3 = int(a3)
self.hours3 = int(hours3)
self.minutes3 = int(minutes3)
self.keep_running = True

def run(self):
    try:
        while self.keep_running:
            time.sleep(1)
            now = time.localtime()
            if self.a1==1:
                if (now.tm_hour == self.hours1 and now.tm_min == self.minutes1):
                    #print("PORCION SERVIDA!1")
                    Servir()
                    time.sleep(60)
            if self.a2==1:
                if (now.tm_hour == self.hours2 and now.tm_min == self.minutes2):
                    #print("PORCION SERVIDA!2")
                    Servir()
                    time.sleep(60)
            if self.a3==1:
                if (now.tm_hour == self.hours3 and now.tm_min == self.minutes3):
                    #print("PORCION SERVIDA!3")
                    Servir()
                    time.sleep(60)
        except:
            #return
            print("ERROR EN DOSIFICACION")
    def just_die(self):
        self.keep_running = False

#PROGRAMA DEL MODO SENSOR LUCES
def Automatico(channel):

```

```

time.sleep(0.5)
if (GPIO.input(LDR)== GPIO.HIGH): # if port 17 == 1
    print ("LUCES ENCENDIDAS")
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN","SE HAN ENCENDIDO LAS LUCES")
    #INSTAPUSH.Notificacion('iCAN: SE HAN ENCENDIDO LAS LUCES')
    GPIO.output(Luz1, False)
    GPIO.output(Luz2, False)
else:
    if (GPIO.input(LDR) != GPIO.HIGH): #if port 17 != 1
        print ("LUCES APAGADAS")
        #INSTAPUSH.Notificacion('iCAN: LUCES HAN SIDO APAGADAS')
        push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN","SE HAN APAGADO LAS LUCES")
        GPIO.output(Luz1, True)
        GPIO.output(Luz2, True)

```

#Clase principal del sistema de luces

```

def Luz():
    global iluminacion
    global Luces
    infile = open('/var/www/files/LUCES.txt', 'r')
    infile.seek(0)
    A0=infile.readline()
    A1=infile.readline() #INDICA: 1 modo MANUAL - 0 modo AUTOMATICO
    A2=infile.readline() #INDICA: 0 modo SENSOR - 1 modo HORARIO
    HE=infile.readline() #HORA ENCENDIDO
    ME=infile.readline() #MINUTO ENCENDIDO
    HA=infile.readline() #HORA APAGADO
    MA=infile.readline() #MINUTO APAGADO
    infile.close()
    #print(A1,A2,HE,ME,HA,MA)

if A1=="0\n": #MODO AUTOMATICO SELECCIONADO

```

```

if A2=="0\n":
    #Automatico(1)
    GPIO.add_event_detect(17, GPIO.BOTH, callback=Automatico,
bouncetime=200)
    print("SENSOR DE LUZ ACTIVADO")
    if Luces == 1:
        iluminacion.just_die()
    else:
        if Luces == 1:
            iluminacion.just_die()
            GPIO.remove_event_detect(17)
            iluminacion = Iluminacion(HE,ME,HA,MA)
            Luces = 1
            iluminacion.start()
        else:
            GPIO.remove_event_detect(17)
            if Luces == 1:
                iluminacion.just_die()
                Luces = 0
def Enciende():
    global flag
    if flag == 0:
        GPIO.output(Luz1, False)
        GPIO.output(Luz2, False)
        push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN","SE HAN ENCENDIDO LAS LUCES")
        flag=1

def Apaga():
    global flag
    if flag == 1:
        GPIO.output(Luz1, True)
        GPIO.output(Luz2, True)

```



```

        push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
        "iCAN","SE HAN APAGADO LAS LUCES")
        flag=0

#Clase principal del sistema de alimentacion
def Comida():
    global dosificador
    global Activo
    infile = open('/var/www/files/HDOSIFICADOR.txt', 'r')
    infile.seek(0)
    Time=infile.readline()
    PW=infile.readline()
    a0=infile.readline()
    a1=int(infile.readline())
    H1=infile.readline()
    M1=infile.readline()
    a2=int(infile.readline())
    H2=infile.readline()
    M2=infile.readline()
    a3=int(infile.readline())
    H3=infile.readline()
    M3=infile.readline()
    infile.close()
    if a1==0 and a2==0 and a3==0:
        if Activo == 1:
            dosificador.just_die()
            #print("MODO AUTOMATICO DE ALIMENTACION DESACTIVADO /
MODO MANUAL")
        else:
            dosificador = Dosificador(Time,PW,a1,H1,M1,a2,H2,M2,a3,H3,M3)
            Activo = 1
            dosificador.start()
            #print("MODO AUTOMATICO DE ALIMENTACION ACTIVADO")

```

```

def Nivel():
    GPIO.output(GPIO_TRIGGER,True) #Enviamos un pulso de ultrasonidos
    time.sleep(0.00001)           #Una pequeña pausa
    GPIO.output(GPIO_TRIGGER,False) #Apagamos el pulso
    start = time.time()           #Guarda el tiempo actual mediante time.time()
    while GPIO.input(GPIO_ECHO)==0: #Mientras el sensor no reciba señal...
        start = time.time()       #Mantenemos el tiempo actual mediante time.time()
    while GPIO.input(GPIO_ECHO)==1: #Si el sensor recibe señal...
        stop = time.time()        #Guarda el tiempo actual mediante time.time() en otra
variable
        elapsed = stop-start      #Obtenemos el tiempo transcurrido entre envío y
recepción
        distance = (elapsed * 34300)/2 #Distancia es igual a tiempo por velocidad partido
por 2  $D = (T \times V)/2$ 
        #print(distance)         #Devolvemos la distancia (en centímetros) por pantalla
        time.sleep(1)           #Pequeña pausa para no saturar el procesador de la
Raspberry
        if distance >= 20:
            GPIO.output(LED,True) #SI LA COMIDA ESTÁ POR DEBAJO DEL
NIVEL ENVÍA NOTIFICACIÓN Y ENCIENDE LED
            push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN","QUEDA POCO ALIMENTO, POR FAVOR RECARGUE")
            #print("SE ESTA ACABANDO LA COMIDA")
        else:
            GPIO.output(LED, False) #SI LA COMIDA ESTÀ POR ENCIMA DEL
NIVEL APAGA EL LED.
            push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN","SE HA SERVIDO UNA PORCION")
            #print("LIMITES NORMALES")

def Servir():
    GPIO.output(VOZ,True)
    time.sleep(10)
    GPIO.output(VOZ,False)

```

```

PWM.start(100)
PWM.ChangeDutyCycle(100)
time.sleep(0.5)
PWM.ChangeDutyCycle(PW)
time.sleep(Time)
#push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
"iCAN", "SE HA SERVIDO UNA PORCION")
PWM.stop()
Nivel()

```

#PROGRAMAS QUE SE EJECUTA AL INICIAR

```
Luz()
```

```
Comida()
```

```
try:
```

```
while True:
```

```
infile = open('/var/www/files/LUCES.txt', 'r')
```

```
infile.seek(0)
```

```
a=infile.readline()
```

```
infile.close()
```

```
infile = open('/var/www/files/HDOSIFICADOR.txt', 'r')
```

```
infile.seek(0)
```

```
b=infile.readline()
```

```
infile.close()
```

```
time.sleep(0.5)
```

```
if a=="1\n":
```

```
    if Luces == 1:
```

```
        iluminacion.just_die()
```

```
    #print("PROGRAMA DE LUCES REINICIADO")
```

```
    outfile = open('/var/www/files/LUCES.txt', 'r+')
```

```
    outfile.write('0\n')
```

```
    Luz()
```

```
    outfile.close()
```

```

if b=="1\n":
    if Activo == 1:
        dosificador.just_die()
        #print("PROGRAMA DE ALIMENTACION REINICIADO")
        outfile = open('/var/www/files/HDOSIFICADOR.txt', 'r+')
        outfile.write('0\n')
        Comida()
        outfile.close()

except:
    #DESACTIVA PROGRAMAS THREAD Y LIBERA PUERTOS
    if Luces == 1:
        iluminacion.just_die()
    if Activo == 1:
        dosificador.just_die()
    print("SALIENDO DEL SISTEMA")
    GPIO.cleanup()

```

## **LUZ-MANUAL.PY**

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-
import push
import RPi.GPIO as GPIO, time
import string
import sys
from time import sleep

#CONSTANTE QUE DEFINE PUERTO GPIO
Luz1=27
Luz2=22
#INICIALIZACION DE PUERTOS GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

```

```

GPIO.setup(Luz1, GPIO.OUT) ## PIN DE LAMPARA 1
GPIO.setup(Luz2, GPIO.OUT) ## PIN DE LAMPARA 2
#SOLICITA INGRESO DE VARIABLE DESDE TERMINAL
OP= sys.argv[1]
#PROGRAMA PRINCIPAL
if OP== "1":
    sleep(1)
    GPIO.output(Luz1, False)
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","L1 HA SIDO ENCENDIDO")

if OP== "2":
    sleep(1)
    GPIO.output(Luz1, True)
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","L1 HA SIDO APAGADO")

if OP== "3":
    sleep(1)
    GPIO.output(Luz2, False)
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","L2 HA SIDO ENCENDIDO")

if OP== "4":
    sleep(1)
    GPIO.output(Luz2, True)
    push.sendNotification("cd48ac7a917e2bb90d19a89345bb14a1fb4ddb0f",
    "iCAN","L2 HA SIDO APAGADO")

```

## **PUSH.PY**

```
#!/usr/local/bin/python
```

```
#https://www.linuxito.com/programacion/635-netisup-py-script-python-para-
verificar-el-estado-de-la-red
```

```
import subprocess
import urllib2
import json
import sys
```

```
hosts = ('8.8.8.8')
```

```
def sendNotification(token, channel, message):
```

```
    if ping(hosts):
        data = {
            "body" : message,
            "message_type" : "text/plain"
        }
```

```
        req =
urllib2.Request('http://api.pushetta.com/api/pushes/{0}'.format(channel))
        req.add_header('Content-Type', 'application/json')
        req.add_header('Authorization', 'Token {0}'.format(token))
        response = urllib2.urlopen(req, json.dumps(data))
        xstatus = 0
```

```
def ping(host):
```

```
    ret = subprocess.call(['ping', '-c', '2', '-W', '2', host],
        stdout=open('/dev/null', 'w'),
        stderr=open('/dev/null', 'w'))
    return ret == 0
```

## **TEMP\_SERIAL.PY**

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```

import serial
import time
import sys

#INICIALIZACION DE COMUNICACION CON ARDUINO
arduino=serial.Serial('/dev/ttyACM0',baudrate=9600, timeout = 3.0)
#arduino.open()

#PROGRAMA PRINCIPAL
dato= sys.argv[1] #RECOGE VALOR INGRESADO
time.sleep(0.2)
arduino.write(dato) #ENVIA VALOR POR SERIAL
time.sleep(0.2)

if dato == "400": #SI ENVIA EL VALOR DE 400 ESPERA A RECIBIR DATOS
POR PARTE DEL ARDUINO
    txt= arduino.read(4)
    time.sleep(0.02)
    print (txt)

arduino.close()

```

**SE ADJUNTA CÓDIGO CARGADO AL ARDUINO MEGA 2560 R3**

**LUCES\_FIN.INO**

```

/**
POR DEFECTO EL PROGRAMA INICIA EN MODO AUTOMATICO
100 ACTIVA MODO AUTOMATICO
200 MANUAL ENCIENDE
300 MANUAL APAGA
400 SOLICITA EL VALOR DE temp (VER SI ESTA BIEN AHI O SE CAMBIA
ESTADO)
CUALQUIER VALOR DEL 20 AL 40 LO TOMA COMO TEM UMBRAL

```

```

/**/

int x=0; //VARIABLE QUE LEE
int DTA;
int inputString = 0; // DATO QUE LEE
int y=30;/// TEMPERATURA INICIAL POR DEFECTO
int A=1;//1 modo automatico on else off
float tempC; // Variable para almacenar el valor obtenido del sensor (0 a 1023)
int pinLM35 = 0; // Variable del pin de entrada del sensor (A0)

void setup() {
  delay(1000);
  // INICIALIZA SERIAL
  analogReference(INTERNAL1V1);
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  pinMode(23, OUTPUT);
  pinMode(25, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    int inputString = Serial.parseInt();
    //Serial.println(inputString);
    x=inputString;
    //Serial.println(x);

    if(x==400){
      // Con analogRead leemos el sensor, recuerda que es un valor de 0 a 1023
      tempC = analogRead(pinLM35);
      // Calculamos la temperatura con la fórmula
      tempC = (1.1 * tempC * 100.0)/1024.0;
      DTA=(int)tempC;
    }
  }
}

```



```

Serial.print(DTA); //Se lee e imprime la temperatura en grados Centigrados
//Serial.print(sensors.getTempCByIndex(0)); //Se lee e imprime la temperatura en
grados Centigrados
  // Serial.println(" Grados Centigrados");
}

}

if( (x==200) ||(x== 300)){ //DETECTA SI ES MANUAL O AUTOMATICA
  A=0;//desactiva modo automatico
  //sensors.stop();//por que debe leer temp la pagina
  if (x==200){
    ON_OFF(0);
  }else{
    ON_OFF(1);
  }
}else{
  if(x==100){
    A=1;
  }
}

if ( (x>=20) && (x<=40))y=x;
if(A==1){
  tempC = analogRead(pinLM35);
  // Calculamos la temperatura con la fórmula
  tempC = (1.1 * tempC * 100.0)/1024.0;
  if (tempC>= y){
    ON_OFF(0); //logica inversa
  }else{
    ON_OFF(1);//logica inversa
  }
}
}

void ON_OFF (int val){
if (val==1){

```

```
digitalWrite(13, HIGH); //Si el valor de input es 1, se enciende el led
digitalWrite(23, HIGH); //Si el valor de input es 1, se enciende el led
digitalWrite(25, HIGH); //Si el valor de input es 1, se enciende el led
}
else
{
digitalWrite(13, LOW); //Si el valor de input es diferente de 1, se apaga el LED
digitalWrite(23, LOW); //Si el valor de input es diferente de 1, se apaga el LED
digitalWrite(25, LOW); //Si el valor de input es diferente de 1, se apaga el LED
}
}
```