



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB CON
BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO**

Jorge Roberto García Velásquez

Asesorado por el Ing. Carlos Eduardo Guzmán Salazar

Guatemala, mayo de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB
CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

JORGE ROBERTO GARCÍA VELÁSQUEZ

ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, MAYO DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Angel Roberto Sic García a.i.
EXAMINADOR	Ing. Carlos Eduardo Guzmán Salazar
EXAMINADOR	Ing. José Aníbal Silva De Los Ángeles
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 17 de agosto de 2017.



Jorge Roberto García Velásquez

Guatemala, 15 de noviembre de 2018

Ingeniero
Julio Solares Peñate
Coordinador Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado ingeniero Solares:

Hago de su conocimiento que he finalizado la revisión del trabajo de graduación del estudiante **Jorge Roberto García Velásquez**, titulado **“DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO”**, el cual ha cumplido con los objetivos que se plantearon para su elaboración.

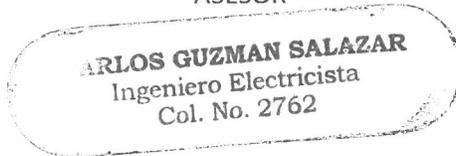
Por lo que, como ASESOR nombrado por la Escuela de Ingeniería Mecánica Eléctrica, doy mi APROBACIÓN al mismo. Tanto el suscrito como el estudiante García Velásquez somos responsables del contenido del trabajo en mención.

Reciba un cordial saludo,



Carlos Guzmán Salazar

ASESOR





FACULTAD DE INGENIERIA

Guatemala, 28 de enero de 2019

Señor Director
Ing. Otto Fernando Andrino González
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVES DE INTERFAZ WEB CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO**, desarrollado por el estudiante **Jorge Roberto García Velásquez**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS


Ing. Julio César Solares Peñate
Coordinador de Electrónica





REF. EIME 08. 2019.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación del estudiante: **JORGE ROBERTO GARCÍA VELÁSQUEZ** Titulado: **DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO**, procede a la autorización del mismo.


Ing. Otto Fernando Andrino González



GUATEMALA, 26 DE FEBRERO 2019.

Universidad de San Carlos
De Guatemala

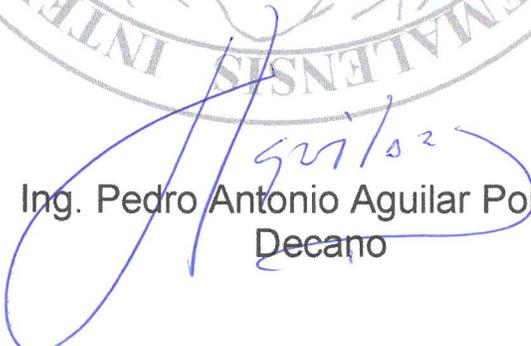


Facultad de Ingeniería
Decanato

Ref. DTG.246-2019

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica del trabajo de graduación titulado: **"DISEÑO DE SISTEMA DOMÓTICO CONTROLADO A TRAVÉS DE INTERFAZ WEB CON BASE EN LA PLACA DE DESARROLLO ORANGE PI Y ARDUINO"** presentado por el estudiante: **Jorge Roberto García Velásquez** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.


Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, mayo de 2019

/echm



ACTO QUE DEDICO A:

Dios

Quien, siendo Padre creador, Hijo salvador y Espíritu inspirador y consejero, es El ser supremo de la vida y Rey del universo, quien en su infinita misericordia me concedió la vida y ordenó las situaciones necesarias de la misma para hacer posible llegar a esta meta, al brindarme retos y la sabiduría para poder enfrentarlos.

Mis padres

Guillermo Antonio García Solís y María Luisa Velásquez Ruiz de García, quienes siempre me acompañaron en todas las circunstancias de mi vida y que sin importar si algunas fueran malas, siempre creyeron en mis capacidades para alcanzar esta meta y me brindaron su amor, cariño y apoyo incondicionalmente a lo largo de toda mi formación académica.

Mis hermanos

Julio Rodolfo y Mario Guillermo García Velásquez, quienes en todo momento me dieron su apoyo, comprensión y cariño a lo largo de la carrera y que de un modo u otro me brindaron su ayuda y tuvieron una palabra de motivación cuando más la necesité.

Mis abuelos

Ismael García y Filomena Solís, quienes ya se encuentran en el descanso eterno y de quienes tengo muy buenos recuerdos, y mi abuela Daniela Ruiz, con quien aún compartimos en vida.

Mis mejores amigos de la universidad

Pedro Pablo Samuel Muñoz Taniel, Julio Roberto Luna Reynoso y Darlineth Yajaira Alonzo Morales, con quienes a lo largo de la carrera compartimos alegrías, decepciones, problemas, ideas, retos y triunfos.

Mis mejores amigos del colegio

Carlos David Aroche Quevedo y Kevin Alejandro Guerra Juárez, con quienes al finalizar la carrera de diversificado continuamos manteniendo una excelente amistad.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por permitirme ser parte del grupo de estudiantes de tan prestigiosa universidad, la cual me permitió recibir mis estudios de nivel superior como el <i>alma mater</i> que es.
Facultad de Ingeniería	Por ser la facultad que me brindó la carrera de mi elección.
Escuela de Ingeniería Mecánica Eléctrica	Por ser la escuela que me brindó mi formación académica para desarrollarme como ingeniero.
Mi asesor	Ing. Carlos Eduardo Guzmán Salazar, por ser instructor en varios cursos a lo largo de mi carrera y por su ayuda y asesoría en este trabajo de graduación.
Mis amigos de la facultad	A todos mis amigos en general con quienes compartí diferentes etapas de la carrera.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. TEORÍA Y CONCEPTOS DEL SISTEMA DE DOMÓTICA Y DE LOS COMPONENTES DEL SISTEMA.....	1
1.1. Domótica	1
1.1.1. Definición de domótica	1
1.1.2. Aspectos de la domótica.....	1
1.1.2.1. Ahorro energético	2
1.1.2.2. Confort.....	2
1.1.2.3. Seguridad	2
1.1.2.4. Comunicaciones	3
1.1.3. Servicios de la domótica.....	3
1.1.4. Arquitecturas de un sistema de domótica	4
1.1.4.1. Arquitectura centralizada	4
1.1.4.2. Arquitectura descentralizada	4
1.1.4.3. Arquitectura distribuida	5
1.1.4.4. Arquitectura híbrida o mixta	5
1.1.5. Elementos a instalar en un sistema de domótica.....	5
1.2. Orange pi.....	6
1.2.1. Descripción.....	6

1.2.2.	Hardware de Orange Pi.....	7
1.2.3.	Modelos de Orange Pi.....	8
1.3.	Arduino.....	9
1.3.1.	Hardware de arduino.....	9
1.3.2.	Software de arduino.....	10
1.3.3.	Modelos de arduino.....	10
1.4.	Dimmer.....	11
1.5.	Módulo NRF 24L01.....	11
1.5.1.	Descripción.....	11
1.5.2.	Características.....	12
1.5.3.	Especificaciones.....	12
1.6.	HTML.....	13
1.7.	JavaScript.....	13
1.8.	CSS.....	14
1.9.	PHP.....	14
1.10.	Python.....	14
1.11.	Armbian.....	15
2.	ANÁLISIS DE LA SITUACIÓN Y DESCRIPCIÓN DEL PROBLEMA.....	17
2.1.	Situación.....	17
2.2.	Problema a nivel general.....	17
2.3.	Causas.....	18
2.4.	Efectos.....	19
3.	PLANTEAMIENTO DE DISEÑO DEL SISTEMA Y DISEÑO DE HARDWARE DEL PROTOTIPO E INSTALACIÓN.....	21
3.1.	Diseño del sistema.....	21
3.1.1.	Especificaciones del sistema.....	21
3.1.2.	Módulos inalámbricos.....	22

3.1.3.	Mando central	22
3.1.4.	Microcontrolador maestro	24
3.1.5.	Microcontrolador esclavo	25
3.1.6.	Topología de red.....	26
3.2.	Diseño del hardware.....	27
3.2.1.	Mando central	28
3.2.1.1.	Componentes del mando central	28
3.2.1.2.	Conexión del mando central	29
3.2.1.3.	Caja del mando central	30
3.2.2.	Microcontrolador maestro	33
3.2.2.1.	Funciones del microcontrolador maestro.....	33
3.2.2.2.	Componentes de microcontrolador maestro.....	34
3.2.2.3.	Conexión de microcontrolador maestro.....	34
3.2.2.4.	Diseño de placa de circuito impreso	36
3.2.2.5.	Diseño de la caja	37
3.2.3.	Microcontroladores esclavos	39
3.2.3.1.	Microcontrolador esclavo 1	40
3.2.3.1.1.	Funciones	40
3.2.3.1.2.	Componentes	41
3.2.3.1.3.	Conexión	41
3.2.3.1.4.	Diseño de circuito impreso.....	42
3.2.3.1.5.	Diseño de la caja	44
3.2.3.2.	Microcontrolador esclavo 2	45
3.2.3.2.1.	Funciones	46
3.2.3.2.2.	Componentes	46

	3.2.3.2.3.	Conexión	47
	3.2.3.2.4.	Diseño de circuito impreso	48
	3.2.3.2.5.	Diseño de la caja.....	49
3.3.		Instalación del sistema	49
	3.3.1.	Instalación del mando central.....	50
	3.3.2.	Instalación de microcontrolador maestro.....	50
	3.3.3.	Instalación de microcontrolador esclavo 1.....	51
	3.3.4.	Instalación de microcontrolador esclavo 2.....	52
	3.3.5.	Instalación en la casa	54
4.		SOFTWARE DE LOS ELEMENTOS DEL SISTEMA	57
	4.1.	Programación del microcontrolador maestro.....	57
	4.2.	Programación del microcontrolador esclavo 1	67
	4.3.	Programación del microcontrolador esclavo 2	71
	4.4.	Programación del mando central	75
	4.4.1.	Programación en Python	76
	4.4.2.	Programación en PHP y códigos de páginas	79
	4.4.2.1.	Página de acceso.....	79
	4.4.2.1.1.	Código PHP, HTML y CSS.....	80
	4.4.2.2.	Página principal de la aplicación	83
	4.4.2.2.1.	Códigos HTML, PHP y JavaScript.....	84
	4.4.2.3.	Página de control de iluminación.....	87
	4.4.2.3.1.	Códigos HTML y JavaScript	88
	4.4.2.3.2.	Código PHP	91
	4.4.2.3.3.	Código CSS	93

4.4.2.4.	Página de control de energía.....	96
4.4.2.4.1.	Códigos HTML y JavaScript.....	97
4.4.2.4.2.	Código PHP.....	100
4.4.2.4.3.	Código CSS.....	102
4.4.2.5.	Página de control remoto IR	106
4.4.2.5.1.	Código HTML y JavaScript.....	107
4.4.2.5.2.	Código PHP.....	112
4.4.2.5.3.	Código CSS.....	113
4.4.2.6.	Página de sensores	115
4.4.2.6.1.	Código HTML y JavaScript.....	116
4.4.2.6.2.	Código PHP.....	119
4.4.2.6.3.	Código CSS.....	121
CONCLUSIONES		125
RECOMENDACIONES		127
BIBLIOGRAFÍA.....		129

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Mini ordenador modelo Orange Pi Pc	6
2.	Distribución de pines de propósito general	8
3.	Módulo NRF 24L01	12
4.	Módulos inalámbricos.....	22
5.	Componentes de mando central	23
6.	Componentes de microcontrolador maestro	25
7.	Componentes de microcontrolador esclavo	26
8.	Topología de la red del sistema	27
9.	Diagrama de conexión de Orange Pi y NRF24L01	30
10.	Caja original para Orange Pi Pc Plus	31
11.	Caja modificada para Orange Pi Pc Plus y NRF24L01	31
12.	Dimensiones de la caja para Orange Pi y NRF24L01	32
13.	Vista superior de la caja sin tapa	32
14.	Diagrama de conexión de microcontrolador maestro	35
15.	Diseño del PCB de microcontrolador maestro	36
16.	Vista frontal de la caja	37
17.	Vistas laterales de la caja.....	38
18.	Dimensiones de la caja	38
19.	Diagrama de conexión	42
20.	Diseño del PCB.....	43
21.	Vistas de la caja	44
22.	Dimensiones de la caja	45
23.	Diagrama de conexión	47

24.	Diseño del PCB.....	48
25.	Vistas de la caja.....	49
26.	Instalación del mando central	50
27.	Instalación de microcontrolador maestro	51
28.	Instalación de microcontrolador esclavo 1	52
29.	Modificación a la instalación del interruptor	53
30.	Instalación de microcontrolador esclavo 2	54
31.	Instalación del sistema en una casa	55
32.	Página de acceso	80
33.	Página principal	84
34.	Página de iluminación.....	88
35.	Página de energía.....	97
36.	Página de control remoto IR	107
37.	Página de sensores	116

LISTA DE SÍMBOLOS

Símbolo	Significado
cm	Centímetro
dB	Decibel
g	Gramo
Ghz	Gigaherzio
Kbps	Kilobit por segundo
Mbps	Megabit por segundo
mA	Miliamperio
mAh	Miliamperio hora
m	Metro
mm	Milimetro
uA	Microamperio
V	Voltio

GLOSARIO

ADC	Son las siglas en inglés de Analog-to-Digital Converter, representan a un dispositivo electrónico que convierte señales analógicas en digitales.
ARM	Son las siglas en inglés de Advanced RISC Machine, arquitectura de un tipo de procesadores.
Banda ISM	Singnifican en inglés Industrial, Scientific and Medical, y hacen referencia a las bandas reservadas internacionalmente de uso no comercial de radiofrecuencia electromagnética.
CVBS	Son las siglas en inglés de Color, Video, Blanking and Sync, y hacen referencia al video compuesto.
Faradios	Es la unidad de capacidad eléctrica del sistema internacional de unidades.
<i>Framework</i>	Es un entorno de trabajo con estructura conceptual y tecnológica de asistencia definida para el desarrollo de software.
IDE	Son las siglas en inglés de Integrated Development Environment, y es una aplicación informática que

proporciona servicios y facilita el desarrollo de software.

IPX Son las siglas en inglés de Internetwork Packet Exchange, un antiguo protocolo de comunicación utilizado para transferir datos entre nodos.

I2C Es un protocolo síncrono y hace referencia a un bus de comunicación en serie y su nombre se refiere a Inter-Integrated Circuit.

Open Source Es un modelo de desarrollo de software basado en la colaboración abierta.

Optoacoplador Es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo led y recibida por un fototransistor o *fototriac* según sea la necesidad.

PCB Son las siglas en inglés de Printed Circuit Board, y hace referencia a una placa de circuito impreso y está construida por pistas de cobre para la colocación de componentes electrónicos.

PIR Son las siglas en inglés de Passive Infrared, y hace referencia a detectores de radiación de calor.

PWM	Son las siglas en inglés de Pulse Width Modulation, y hace referencia a la técnica en la que se modifica el ciclo de trabajo de una señal periódica.
SPI	Son las siglas en inglés de Serial Peripheral Interface, y hace referencia a un estándar de comunicación usado para la transferencia de datos entre circuitos integrados.
SVG	Son las siglas en inglés de Scalable Vector Graphics, y es un vector de gráficos vectoriales en formato XML.
Topología	Es el mapa físico o lógico para el intercambio de datos en una red.
<i>Triac</i>	Es un semiconductor de la familia de los tiristores, la corriente puede circular de manera bidireccional y por lo mismo es usado como un interruptor de corriente alterna.
UART	Son las siglas en inglés de Universal Asynchronous Receiver Transmitter, y hace referencia al dispositivo que controla los puertos seriales.
USB	Son las siglas en inglés de Universal Serial Bus, y hace referencia al estándar de bus de comunicación en tanto cables, conectores y protocolos para la conexión y comunicación de dispositivos.

USB OTG

OTG son de las siglas en inglés de On The Go, y es una extensión de la norma USB que permite a los dispositivos USB tener una mayor flexibilidad en la gestión de la interconexión.

XHTML

Son las siglas en inglés de Extensible Hyper Text Markup Language, y es básicamente html expresado como XML válido.

XML

Son las siglas en inglés de Extensible Markup Language, y hace referencia a un lenguaje de marcas de propósito general que no tiene sus etiquetas definidas y su objetivo general es el de compartir datos a través de diferentes sistemas.

XUL

Son las siglas en inglés de Xml User Interface Language, y hace referencia a un lenguaje basado en XML para la interfaz de usuario desarrollado por Mozilla.

RESUMEN

El presente trabajo expone las características de un sistema de domótica, desde los elementos principales que lo conforman hasta el tipo de sistemas y protocolos usados. Se especifica qué tipos de red de comunicación existen en estos sistemas, así como ventajas y desventajas de cada una.

En conjunto con las características de estos sistemas también se da una breve teoría y una explicación de las características de los elementos principales utilizados en el diseño propuesto, esto va desde los componentes físicos utilizados hasta la parte que implica el software que controla los mismos. Los componentes físicos principales que se utilizan en el diseño de este sistema son los microcontroladores Atmega 328p, el módulo de comunicación inalámbrica NRF24L01 y el mini ordenador Orange Pi. En la parte del software se hace uso de lenguajes tales como HTML, CSS, JavaScript, PHP, Python y el IDE de Arduino.

En la presentación del diseño del sistema propuesto se exponen todos los diagramas electrónicos que conforman el sistema, así como sus respectivos diagramas de circuito impreso y los diseños de las cajas propuestos para cada elemento del sistema.

OBJETIVOS

General

Presentar el diseño de un sistema domótico controlado a través de interfaz web con base en la placa de desarrollo Orange Pi y Arduino.

Específicos

1. Dar a conocer las características principales que componen un sistema de domótica.
2. Mostrar los diferentes dispositivos que conforman el diseño del sistema de domótica.
3. Dar a conocer una alternativa diferente a la Raspberry Pi en el campo de los mini ordenadores.
4. Evidenciar la ayuda que representa la presencia de un sistema de domótica en el hogar para los integrantes que la habitan.
5. Verificar las ventajas y desventajas que hay entre un sistema elaborado con este diseño y un sistema comercial.

INTRODUCCIÓN

La domótica es la encargada de todo lo referente a la automatización de una casa o vivienda, su función es el manejo automático de los recursos de una casa tales como la electricidad, agua, gas, apertura de puertas, entre otros. Su función es facilitar el uso de todos estos recursos y de las funciones de la misma de una manera automática y centralizada.

El uso de estos sistemas es poco usual en las viviendas. A pesar que la domótica tiene varios años de existir aun es poco el uso y el conocimiento que muchas personas pueden tener acerca de ella. En la actualidad existe una variedad de sistemas de domótica, los cuales pueden variar sus funciones y el modo en el que son instalados, ya que muchos pueden utilizar diferentes medios para la comunicación de los dispositivos que son utilizados por el sistema.

En este proyecto se propone un sistema de domótica que tiene como base el mini ordenador Orange Pi, el cual funciona como un servidor web que da acceso al sistema por medio de un navegador web que puede ser visualizado por un dispositivo móvil. Este mini ordenador se comunica con sus dispositivos de manera inalámbrica a través de un módulo NRF24L01, lo cual hace que el sistema sea modular y permite que su instalación sea simple y sin necesidad de alterar o modificar en gran manera alguna estructura de la vivienda.

1. TEORÍA Y CONCEPTOS DEL SISTEMA DE DOMÓTICA Y DE LOS COMPONENTES DEL SISTEMA

1.1. Domótica

La palabra domótica tiene su origen en la unión de dos palabras, comenzando con domus en latín que significa casa y tica que hace referencia a la palabra en griego automática, lo cual se refiere a una casa automática o que se controla sola.

1.1.1. Definición de domótica

Domótica es el nombre científico que se le da a un conjunto de tecnologías que se encargan de crear un control centralizado de servicios de una casa o vivienda y automatizarlos por medio del uso de la mecánica, electrónica, informática y otros. Entre los servicios puede destacarse el uso de la iluminación, la multimedia, cerraduras de puertas, ventanas y otras tareas propias del hogar, para así poder brindar comodidad, seguridad y un mejor manejo en el consumo energético de las personas que habitan la casa.

1.1.2. Aspectos de la domótica

La domótica puede ofrecer una variedad de servicios a un usuario, pero todos los servicios comparten uno o varios aspectos propios de la domótica, dichos aspectos principales pueden ser divididos en 4 partes:

1.1.2.1. Ahorro energético

Un sistema de domótica puede influir directamente en la optimización del uso de la energía y así lograr un ahorro energético, ya que muchas veces no es necesario sustituir los aparatos que posee una casa, a veces basta con darle un manejo adecuado al uso de los mismos, lo cual puede hacerlo el sistema.

Entre las maneras que se puede hacer uso del ahorro energético se pueden mencionar:

- Climatización programada
- Desconexión de equipos de uso no prioritario según horarios
- Alertas de consumo de energía de un aparato en función del tiempo

1.1.2.2. Confort

El confort es uno de los principales aspectos del sistema, ya que este es el encargado de permitirle al usuario realizar tareas y funciones de una manera sencilla e intuitiva al mismo tiempo, ya sea para el uso de las funciones de multimedia o tareas básicas de la vivienda, básicamente el confort se basa en brindar una forma cómoda del manejo de los recursos de la vivienda.

1.1.2.3. Seguridad

Consiste en una red que se basa en la visualización de cámaras y la lectura de sensores para dar paso a la gestión de alertas, las cuales podrían ser:

- Alerta de intrusos o aviso de personas que se acercan a la propiedad
- Detección de gas

- Detección de humedad
- Detección de apertura de puertas
- Detección de apertura o daño de ventanas
- Acceso a cámaras
- Cierre de puertas y ventanas

1.1.2.4. Comunicaciones

La comunicación en lo que respecta al sistema de domótica se refiere a una eficiente forma de poder comunicarse a un área de la vivienda en específico, ya sea por video y/o audio.

1.1.3. Servicios de la domótica

Existe una gran variedad de servicios que puede brindar un sistema de domótica, los servicios que sean incluidos dependerán del sistema instalado en la vivienda, pero entre los más comunes se pueden destacar:

- Gestión de energía eléctrica
- Control de la iluminación (apagado/encendido general o por secciones)
- Manejo centralizado de electrodomésticos y sistemas de multimedia
- Automatización en la apertura y cierre de ventanas y puertas
- Cámaras de seguridad
- Alerta de intrusos y alarma de sensores
- Intercomunicación entre ambientes
- Controles de acceso

1.1.4. Arquitecturas de un sistema de domótica

La clasificación de la arquitectura de un sistema de domótica dependerá de dos cosas: la red y el lugar en donde se encuentre el cerebro o mando del sistema, el cual se encarga de gestionar todos los procesos de un sistema, las arquitecturas más comunes en los sistemas de domótica son:

1.1.4.1. Arquitectura centralizada

La arquitectura centralizada de un sistema de domótica tiene la característica de poseer un controlador central (cerebro) que posee toda la inteligencia del sistema y recibe la información de manera directa de los sensores, y con base en los estados de los sensores se encarga de tomar las decisiones y con ellas controlar directamente un grupo de actuadores según sea la programación y configuración que tenga establecida.

1.1.4.2. Arquitectura descentralizada

La arquitectura descentralizada de un sistema de domótica tiene la característica de poseer varios controladores que interactúan con sus propias interfaces de actuadores y sensores cada uno, y todos los controladores se comunican entre ellos por medio de un bus de comunicación que puede ser por cualquier medio para así compartir su información, si esta es del interés de otro controlador.

1.1.4.3. Arquitectura distribuida

La arquitectura distribuida de un sistema de domótica tiene la característica de que todos los elementos del sistema tienen un controlador, otra forma de exponerlo por ejemplo es que un actuador tiene un controlador dedicado únicamente a ocuparse de esa tarea, así también un sensor tiene un controlador para recibir únicamente los datos del sensor, todos los controladores se comunican por un bus de comunicación para enviar y recibir datos y así actuar según sea la configuración y programación que le corresponde a cada uno.

1.1.4.4. Arquitectura híbrida o mixta

La arquitectura híbrida o mixta de un sistema de domótica tiene la característica de utilizar todas o algunas de las arquitecturas anteriormente mencionadas, según sea la conveniencia o necesidad del sistema, puede tener un controlador principal (cerebro) y controlar varios controladores que se encuentren en una arquitectura descentralizada y/o distribuida, dependiendo de la forma de la red de comunicación que exista entre los controladores y la localización física de los actuadores y sensores.

1.1.5. Elementos a instalar en un sistema de domótica

Se puede describir una gran cantidad de elementos que serán usados en un sistema de domótica, pero en términos más generales se pueden describir:

- Controlador o mando central
- Sensores
- Actuadores

- Red de comunicación de dispositivos

1.2. Orange Pi

Se asigna el nombre de Orange Pi a este mini ordenador para hacer una referencia al mini ordenador Raspberry Pi el cual fue el primero en su tipo.

1.2.1. Descripción

Es un mini ordenador multipropósito que integra todo en una sola placa que es de código abierto, el mini ordenador es fabricado por la compañía Shenzhen Xunlong Software CO., Limited. La Orange Pi puede ser utilizada con una variedad de sistemas operativos, entre los cuales los más famosos serían Android, Ubuntu y Debian, también puede utilizar imágenes de los sistemas operativos de Raspberry Pi y Banana Pi.

A continuación se muestra una imagen de referencia de uno de los modelos de un mini ordenador Orange Pi:

Figura 1. **Mini ordenador modelo Orange Pi Pc**



Fuente: *Orange Pi PC*. http://img.dxcn.com/productimages/sku_455299_1.jpg.

Consulta: 10 de enero de 2018.

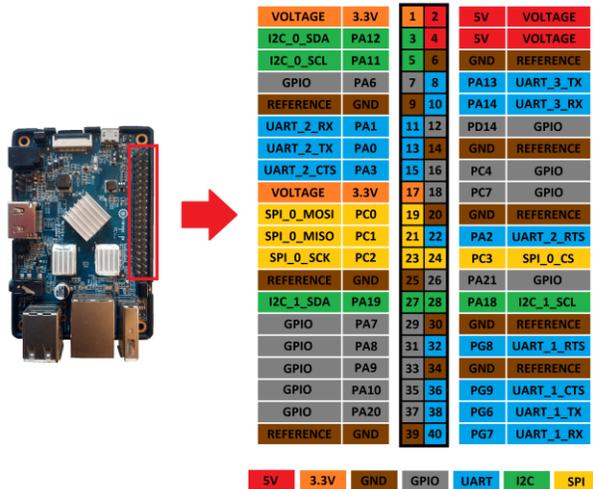
1.2.2. Hardware de Orange Pi

La Orange Pi utiliza los microprocesadores allwinner H2, H3 y H5 SoC, A64 quad-core Cortex-A53 64bit o ARM Cortex-A5 32bit, entre otros, y pueden tener desde 256MB hasta 2GB de ram ddr3, pudiendo tener según el modelo una variedad de interfaces, las más importantes son:

- 40 pines de propósito general que son compatibles con raspberry pi B+
- USB 2.0, USB-otg y en algunos modelos puerto Sata
- Puerto ethernet, wifi y en algunos modelos, modulo bluetooth y 2G
- HDMI, CVBS y entrada de cámara
- Micrófono y Jack de audio de 3.5mm
- Lector de tarjeta SD
- Receptor de IR y puerto de depuración uart

A continuación se muestra una imagen donde se detalla la distribución de los 40 pines de propósito general de la Orange Pi como también las interfaces existentes en los mismos:

Figura 2. **Distribución de pines de propósito general**



Fuente: *Pines de propósito general.*

<https://i1.wp.com/telegra.ph/file/889613796031b80386615.png>.

Consulta: 10 de enero de 2018.

1.2.3. Modelos de Orange Pi

Según la página oficial, existen alrededor de 18 modelos de placas diferentes de mini ordenadores Orange Pi, cada versión posee variaciones en características del hardware y el propósito para el que sea destinado, estos modelos pueden ser adquiridos en tiendas de Internet, a un costo variado según sea el lugar de compra. Entre los modelos más famosos se pueden mencionar:

- Orange Pi One.
- Orange Pi Pc.
- Orange Pi Pc Plus.
- Orange Pi Plus 2E.
- Orange Pi Plus 2.

- Orange Pi Zero.
- Orange Pi 2G IOT.

1.3. Arduino

Arduino es una plataforma de prototipos electrónicos de código abierto y está basado en un hardware y software intuitivo de fácil uso. El arduino está pensado para diseñadores, desarrolladores, estudiantes de programación y electrónica. Consta de una placa principal que contiene los elementos electrónicos necesarios para poder ser utilizado y programado desde una computadora sin tener la necesidad de agregar circuitería adicional, solamente es necesaria la instalación del programa de desarrollo y los controladores requeridos de la placa Arduino en la computadora para poder así comenzar a utilizarla.

1.3.1. Hardware de Arduino

El hardware de Arduino se basa en microcontroladores *atmel avr*, estos controladores son de 8 bits y se programan con lenguaje de alto nivel. Este microcontrolador tiene la capacidad de realizar procesos lógicos y matemáticos dentro del mismo. El diseño de Arduino tiene la capacidad de controlar una gran variedad de dispositivos de tipo digital, analógico y de comunicación, ya que el microcontrolador cuenta con pines de propósito general, módulos PWM, módulos de ADC y módulos de comunicación (SPI, UART, I2C), lo cual permite el uso de sensores, botones de control, actuadores y placas conocidas como *shields* que son fabricadas para ser usadas con el diseño específico de Arduino.

Los microcontroladores más utilizados para los diseños de las placas de Arduino son el Atmega 168, Atmega 328, Atmega 1280 y el Atmega 8, la razón

por la que estos microcontroladores son los más utilizados es debido al bajo costo que tienen. El Arduino también cuenta con un *chip* de conversión de USB a serie, y esto es para la comodidad del usuario, ya que para el uso es necesario un puerto serie, pero en la actualidad no todas las computadoras traen dicho puerto, en especial las computadoras portátiles, y es por esta razón que es agregado este conversor para así brindar una mayor comodidad en su uso.

1.3.2. Software de Arduino

Entre las principales características del software de Arduino se puede mencionar que este es multiplataforma, ya que funciona en los 3 sistemas operativos principales: Windows, Macintosh OSX y Linux. El entorno de programación es simple y directo y de fácil uso para principiantes y brinda flexibilidad para usuarios avanzados.

El lenguaje que opera dentro de un Arduino se llama *wiring*, el cual está basado en la plataforma Processing y primordialmente en el lenguaje C/C++, el software de Arduino se caracteriza por ser de código abierto, lo cual permite a los usuarios poder estudiar, modificar y mejorar su diseño mediante la disponibilidad del código fuente.

1.3.3. Modelos de arduino

Existe una amplia variedad de modelos de Arduino y la selección de la placa a utilizar dependerá de las necesidades a cubrir y del costo del mismo, entre algunos modelos de Arduino se pueden mencionar el Arduino Uno, Arduino Duemilanove, Arduino Mega, Arduino Ethernet, Arduino Mini y Arduino Nano.

1.4. Dimmer

Un *dimmer* es un dispositivo electrónico que se encarga de regular la tensión eléctrica para lograr una variación en la iluminación de una lámpara o un grupo de lámparas que permitan la variación de su iluminación con respecto a la variación en la tensión. Existen diferentes tipos de *dimmer* y entre los más usuales se pueden mencionar el *dimmer* manual y el automático o digital. Un *dimmer* representa un ahorro en el consumo de energía, ya que una lámpara puede variar su consumo a un determinado nivel de capacidad en el momento de que no sea necesario toda la capacidad de la misma.

1.5. Módulo NRF 24L01

Modulo de comunicación bidireccional que tiene como base el chip del fabricante Nordic semiconductor.

1.5.1. Descripción

El nRF24L01 es un circuito altamente integrado, de ultra bajo consumo, es un transceptor de RF de 2Mbps para la banda ISM de 2,4 GHz. Con corrientes inferiores a 14 mA, administración avanzada de energía y una gama de alimentación de 1,9 a 3,6 V, el nRF24L01, ofrece una solución que permite una duración verdadera de meses a años de una batería. El nRF24L01 integra un transceptor RF de 2,4 GHz que soporta una interfaz SPI de alta velocidad para el controlador de la aplicación, posee un sintetizador de RF y algoritmos de control de errores. No se requieren filtros externos, resonadores o diodos VCO.

Debido a su diseño y tamaño tiene un gran número de aplicaciones que pueden ser como por ejemplo para periféricos de computadora, mandos de videojuegos, telemetría, juguetes y electrónica de consumo. A continuación se muestra una imagen del módulo NRF 24L01.

Figura 3. **Módulo NRF 24L01**



Fuente: *Módulo NRF*. <http://www.evoluta.cl/images/nfrard.jpg>. Consulta: 12 de enero de 2018.

1.5.2. Características

- Trabaja en la banda libre de 2.4GHz
- Velocidades de 250kbps, 1Mbps y 2Mbps
- Muy bajo consumo en el orden de los 20 uA

1.5.3. Especificaciones

- Alimentación: 1.9~3.6V
- Voltaje puertos IO: 0~3.3V / 5V
- Nivel de salida: +7dB
- Sensibilidad de recepción: ≤ -90 dB
- Alcance: 15~30 m (lugares cerrados) hasta 100 m (áreas abiertas)
- Dimensiones: 15x29mm

1.6. HTML

HTML es un lenguaje de etiquetas que es utilizado para la creación de sitios web, no es considerado como un lenguaje de programación sino más bien de diseño web, ya que este tiene como función formar una estructura por medio de las etiquetas para ordenar el contenido de la página como puede ser texto, imágenes, videos, enlaces y otros. HTML significa HyperText Markup Language y es un estándar a cargo del World Wide Web Consortium o Consorcio WWW, esta es una organización que se dedica a la estandarización de todas las tecnologías ligadas a la web, en lo que se refiere a la escritura e interpretación. Se considera el lenguaje web más importante, siendo su invención crucial en la aparición, desarrollo y expansión.

1.7. JavaScript

Al igual que HTML, JavaScript es un lenguaje de programación que se puede utilizar para construir sitios web y para hacerlos más interactivos. Aunque comparte muchas de las características y de las estructuras del lenguaje Java, fue desarrollado independientemente. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. Este lenguaje hace fácil la respuesta a los acontecimientos iniciados por usuarios como la introducción de datos en formularios. El lenguaje JavaScript es *open source*, por lo que cualquier persona puede utilizarlo sin comprar una licencia. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

1.8. CSS

CSS que se refiere a hojas de estilo en cascada en inglés, este es un lenguaje dedicado al estilo que se le puede dar a un documento de etiquetas (HTML). El CSS también fue desarrollado por World Wide Web Consortium para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra y demás elementos referentes al estilo de una página web. El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo.

1.9. PHP

Es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP al igual que JavaScript es un lenguaje que se utiliza para crear páginas web dinámicas, pero se diferencian en que PHP es un lenguaje que es ejecutado del lado del servidor y no del lado del cliente, cuando se ejecuta un *script* de PHP este genera código HTML, pero como este es ejecutado del lado del servidor es imposible ver el código del lado del cliente, solo es posible ver su resultado en web.

1.10. Python

Python es un lenguaje de *scripting* independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones de Windows hasta servidores de red o incluso páginas web. Es un

lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

1.11. Armbian

Es una distribución liviana basada en Debian o Ubuntu, especializada en placas de desarrollo ARM, y está compilado desde cero. Tiene poderosas herramientas de compilación y desarrollo de software, además posee una comunidad de apoyo para facilitar el uso de este sistema operativo. Armbian es compatible con una gran cantidad de microprocesadores arm y estos son:

- Allwinner A10, A20, A31, H2 +, H3, H5, A64.
- Amlogic S805 y S905 (placas Odroid), S802 / S812, S805, S905, S905X y S912.
- Actionsemi S500.
- Freescale / NXP iMx6.
- Marvell Armada A380.
- Rockchip RK3288.
- Samsung Exynos 5422.

2. ANÁLISIS DE LA SITUACIÓN Y DESCRIPCIÓN DEL PROBLEMA

2.1. Situación

En Guatemala se puede observar un crecimiento en el uso diario de la tecnología y entre esto se puede resaltar un punto en específico, como lo es el uso de los dispositivos de telefonía, que tienen la capacidad de establecer una conexión a una red de datos móviles o también una conexión de red Wifi, esto para lograr la comunicación diaria y así poder estar en contacto con personas por medio de redes sociales o para el uso de otras aplicaciones. Dicho esto, a pesar del uso prolongado de la tecnología, aún existen temas que una gran parte de la población desconoce de la misma o que por algún motivo económico no se utiliza, como lo es el caso de la domótica, en este tiempo la domótica más que un lujo se puede considerar una ayuda eficaz y hasta una necesidad para así obtener un correcto uso de los recursos de una casa.

2.2. Problema a nivel general

A pesar del crecimiento en el uso de la tecnología, en una casa todo lo que se utiliza es controlado de una manera mecánica y requiere de la presencia de una persona en el lugar, para así poder supervisar que la función sea realizada como es debido o para no hacer un uso indebido de un recurso como el flujo de agua o la utilización de la energía eléctrica, pero el problema existente es la falta de tiempo y como consecuencia la realización de tareas simultáneas, lo cual termina por provocar el uso innecesario de los recursos por la falta de la supervisión de una persona.

Dicho de otra manera, el tiempo muchas veces resulta ser escaso para el cumplimiento total de las tareas a realizar, y a causa de esto se genera el problema de no poder darle la atención debida a una tarea referente a la casa, y a causa de esto se generan gastos innecesarios en los recursos de la misma, lo cual equivale a un gasto económico.

2.3. Causas

Para este caso puede existir más de una causa y por cada causa puede existir un motivo o acción diferente, pero en este trabajo únicamente se tomarán en consideración las causas que se centran en la ausencia de la domótica en una casa. Las causas más probables por las que en una casa no existe un sistema de domótica pueden ser:

- No saber de la existencia de un equipo que es capaz de centralizar servicios (sistema de domótica).
- Falta de interés en adquirir este tipo de tecnología.
- Costos elevados de equipos de domótica.
- No poseer la economía para obtener un sistema de este tipo.
- Por no querer realizar un cambio a la arquitectura de la casa para la instalación del sistema.
- Incomodidad en el uso de la tecnología.
- Los equipos existentes no se ajustan a la comodidad deseada o no se ajustan a casos específicos de una casa.
- Los equipos no son compatibles con los equipos que se encuentran en la casa.
- El número de personas que utilizan estos sistemas aún es bajo y debido a esto no cuentan con la suficiente popularidad.

2.4. Efectos

Los efectos que pueden existir por la falta de un sistema de domótica dependerán de las costumbres de las personas que habiten en la casa, ya que según su conducta pueden hacer o no un mal uso de los recursos, o también según sea su concentración pueden dejar o no la casa desprotegida en el caso que no tomen las medidas necesarias. Entre los efectos más probables pueden mencionarse:

- No existe un manejo eficiente en la utilización de los recursos.
- Un consumo no adecuado de los recursos.
- Gasto económico por el excesivo consumo de los recursos.
- No poder monitorear el estado de un área en específico sin tener que estar presencialmente en el lugar.
- Debido al poco control que se tiene la seguridad de la casa se vuelve vulnerable.
- No tener una comodidad completa en el uso de los equipos de multimedia.
- Pérdida de tiempo en tareas de rutina.

3. PLANTEAMIENTO DE DISEÑO DEL SISTEMA Y DISEÑO DE HARDWARE DEL PROTOTIPO E INSTALACIÓN

3.1. Diseño del sistema

En el diseño del sistema se especifica el modo de elaboración del sistema, así como el tipo de arquitectura con la que el sistema desarrolla sus funciones.

3.1.1. Especificaciones del sistema

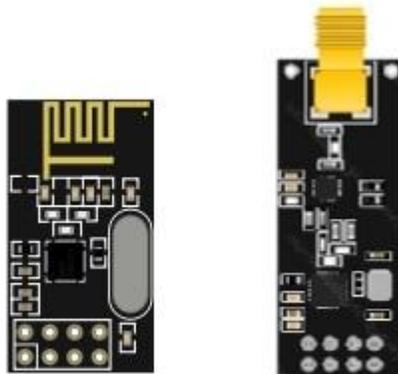
La arquitectura que será utilizada en el diseño de este sistema es la híbrida o mixta, la cual establece que puede tener un controlador principal (cerebro) y controlar varios controladores que se encuentren en una arquitectura descentralizada y/o distribuida dependiendo de la forma de la red de comunicación que exista entre los controladores.

En este diseño se emplea un mando central (cerebro), el cual es la placa de desarrollo Orange Pi y los microcontroladores que son representados por las placas de desarrollo Arduino. Este mando central tiene como función proveer una interfaz para interactuar entre el usuario y los microcontroladores, en el caso de los microcontroladores se pueden ver como principales y secundarios o maestros y esclavos, ya que entre los microcontroladores hay mandos centrales propios de un área, los cuales se encargan de gestionar las instrucciones entre la Orange Pi y los microcontroladores esclavos que pertenecen al área de su microcontrolador maestro, los cuales se comunican por la topología de red inalámbrica en árbol.

3.1.2. Módulos inalámbricos

El modulo inalámbrico utilizado se basa en el *chip* nrf24l01, este módulo trabaja a 2.4Ghz y tiene 125 canales para utilizar. En este diseño se utilizarán dos tipos diferentes de módulos basados en el mismo *chip*, la diferencia está en sus capacidades de alcance, debido a que uno tiene un amplificador de potencia y ganancia que le permite tener un mayor alcance, y este es el módulo inalámbrico que funcionará en conjunto con el módulo central, mientras que los módulos que trabajan solo con el *chip* nrf24l01 funcionarán con los microcontroladores. Los dos diferentes tipos de módulos inalámbricos se muestran en la siguiente imagen:

Figura 4. Módulos inalámbricos



Fuente: elaboración propia, empleando programa Paint.

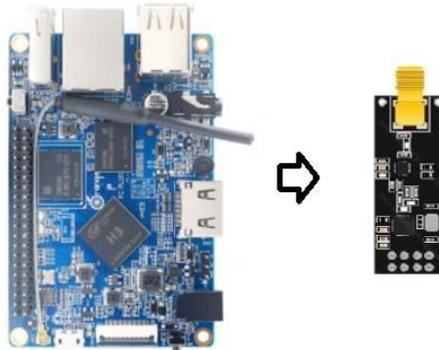
3.1.3. Mando central

El dispositivo utilizado como mando central o cerebro es la placa de desarrollo Orange Pi, esta se encarga de proveer la interfaz en web para

brindar al usuario una interfaz amigable e intuitiva, gestiona las bases de datos de cada usuario, las cuales tiene su información, contraseña, estados de los electrodomésticos, sensores y preferencias del usuario.

El proceso de gestión comienza en la interfaz web, esta le permite el acceso al usuario a las funciones principales del hogar y en especial las funciones de su habitación, luego de esta gestión se procesa esta información por medio de Python en la Orange Pi y luego se procede a enviar las instrucciones a los módulos inalámbricos en el protocolo SPI, para luego enviar la instrucción a cada microcontrolador involucrado en la función seleccionada. El mando central está formado por una placa de desarrollo Orange Pi y un módulo de comunicación inalámbrico NRF24I01, el conjunto puede verse en la siguiente imagen:

Figura 5. **Componentes de mando central**



Fuente: elaboración propia, empleando programa Paint.

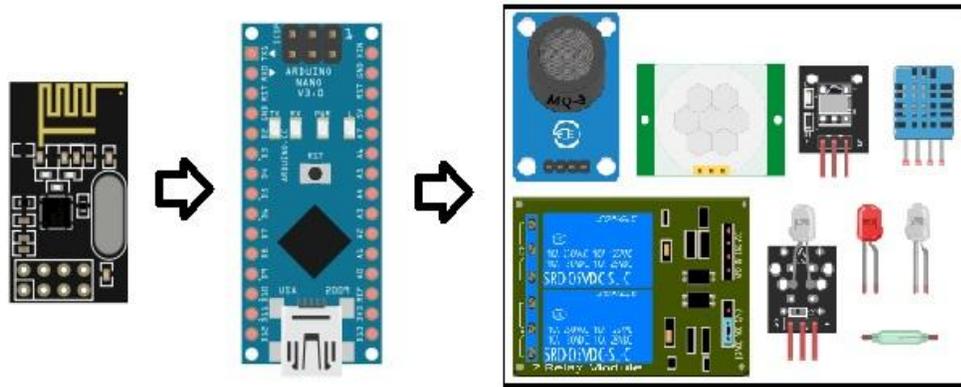
3.1.4. Microcontrolador maestro

El dispositivo utilizado como microcontrolador maestro es un Arduino Nano, este se encarga de centralizar y gestionar los servicios de un área en específico, tiene a su cargo la lectura de sensores conectados directamente a este microcontrolador, entre los cuales se pueden mencionar un sensor de gas, temperatura, movimiento y la detección de apertura de puertas. Además, puede controlar el accionar de dos dispositivos por medio de contactos (relé) para poder controlar la apertura de puertas o aspersores de agua en áreas verdes. Cuenta con una batería recargable y una lámpara led para brindar una luz de emergencia en el caso de que ocurra un corte de energía. También cuenta con un zumbador que le permite dar algún tipo de alarma al usuario.

El proceso de funcionamiento es la recepción de instrucciones que se envían desde el mando central por medio de los módulos inalámbricos y gestionar las instrucciones para ejecutarlas en sí mismo o reenviar a los microcontroladores esclavos por medio del módulo inalámbrico. El microcontrolador maestro está formado por un Arduino Nano, un módulo inalámbrico NRF24I01 y módulos de sensores y actuadores.

El conjunto de componentes del microcontrolador maestro puede verse en la siguiente imagen:

Figura 6. **Componentes de microcontrolador maestro**



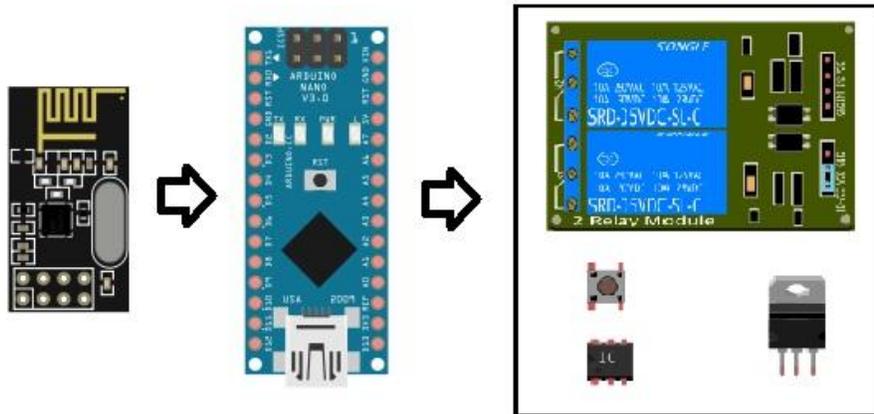
Fuente: elaboración propia, empleando programa Paint.

3.1.5. **Microcontrolador esclavo**

El dispositivo utilizado como microcontrolador esclavo es un Arduino, este se encarga de ejecutar las instrucciones enviadas por su microcontrolador maestro desde el módulo central, su función puede variar según sea el propósito para el cual fue diseñado. Su proceso es la recepción de las instrucciones por medio del módulo inalámbrico y la ejecución de su función.

El conjunto de componentes del microcontrolador esclavo puede verse en la siguiente imagen:

Figura 7. **Componentes de microcontrolador esclavo**



Fuente: elaboración propia, empleando programa Paint.

3.1.6. Topología de red

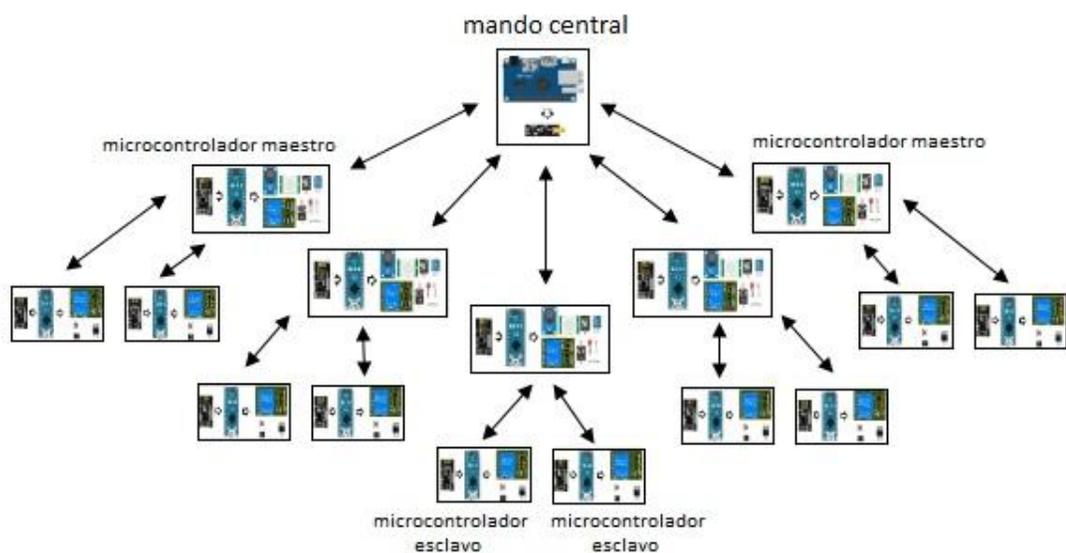
La topología de red en el diseño del sistema de domótica es de tipo árbol o de estrellas extendidas, ya que su funcionamiento se basa en la organización de jerarquías, partiendo de un servidor principal o mando central, el cual se encarga de ordenar y enviar las instrucciones y datos a los diferentes nodos, los nodos se comunican a través de señales inalámbricas por medio del módulo NRF24L01 y las instrucciones son enviadas a los microcontroladores maestros, los cuales trabajando en conjunto con el módulo inalámbrico pueden ser llamados en la red como concentradores secundarios o ramas, estos se encargan de la distribución de la información a los microcontroladores esclavos designados.

En este diseño de red la jerarquía más alta corresponde al mando central, este actúa como servidor tanto para los dispositivos de la red de módulos inalámbricos como para el mismo usuario por medio de la interfaz web. El

mando central distribuye las instrucciones y las envía con la dirección del dispositivo destino, el cual, si es referente a un microcontrolador esclavo, esta información primero es recibida por el microcontrolador maestro de ese segmento, y este a su vez busca en su lista de direcciones de dispositivos esclavos a su cargo y reenvía la información al microcontrolador destinado.

A continuación se muestra en la imagen la representación de la red del sistema de domótica:

Figura 8. **Topología de la red del sistema**



Fuente: elaboración propia, empleando programa Paint.

3.2. **Diseño del hardware**

En el diseño del hardware se especifican todas las conexiones de los componentes físicos de sistema como los módulos y microcontroladores.

3.2.1. Mando central

El mando central está conformado únicamente por dos elementos físicos, el principal es el mini ordenador Orange Pi y el segundo es el módulo inalámbrico NRF24L01.

3.2.1.1. Componentes del mando central

El modelo del mini ordenador Orange PI que será utilizado en este diseño es el Orange Pi PC Plus, el cual dispone de puertos USB, puerto Ethernet, HDMI, micrófono, lector de tarjeta SD, sensor IR, conector de audio y video por medio de *plug* de 3,5mm, USB OTG, interfaz de wifi y pines de propósito general. Las dimensiones de este modelo son de 8,5 cm por 5,5 cm y tiene un peso de 70 g.

En el caso del módulo inalámbrico NRF24L01, es utilizado el modelo que permite una alta distancia, que permite hasta 1100 m sin obstáculos, este modelo es desarrollado para alta potencia y alta sensibilidad, lo cual lo hace ideal para la comunicación a los sensores que se encuentren a distancias largas o con muros u obstáculos de por medio, el módulo cuenta con una base para la colocación de una antena IPX, lo cual le permite una mejor propagación y recepción de las señales de radiofrecuencia. Las dimensiones de este módulo son de 4,1 cm por 1,55 cm. Al módulo NRF24L01 es necesario agregarle un complemento, el cual se encarga de regular el nivel de tensión a 3,3 voltios y cuenta con el filtrado necesario para el correcto funcionamiento.

3.2.1.2. Conexión del mando central

La conexión entre el mini ordenador Orange Pi y el módulo NRF24L01 se realiza con cables tipo Dupont, ambos extremos con terminales hembras, en total se utiliza 7 cables tipo Dupont para realizar esta conexión, de los cuales 5 son de señales de comunicación usando el protocolo SPI y los dos restantes son de alimentación, siendo estos 5 y 0 voltios. En los 5 pines de comunicación por protocolo SPI los pines están identificados en las especificaciones de cada dispositivo y los pines son nombrados de la siguiente manera:

- CE
- CS
- SCK
- MOSI
- MISO

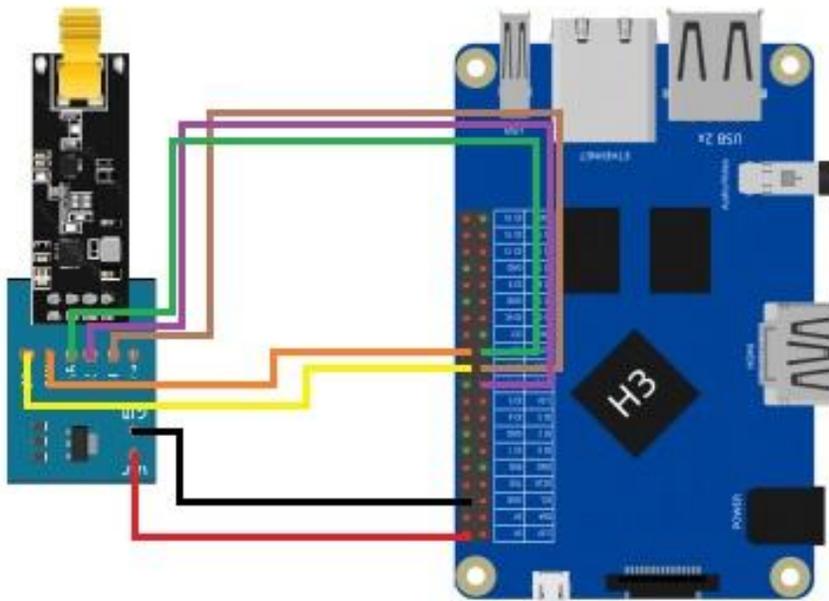
En la nomenclatura utilizada para este módulo inalámbrico CE son las siglas que hacen referencia a *chip enable*, cuya función es activar el *chip* de comunicación, y en el caso de CS las siglas hacen referencia a Chip Select, lo cual indica la selección del *chip* que se utilizará en el SPI, estas dos señales mencionadas anteriormente se conocen como señales de control.

En el caso de SCK las siglas hacen referencia a Serial Clock, cuya función es la sincronización del SPI proveyendo el dispositivo maestro un reloj al dispositivo esclavo, a esta señal se le conoce como señal de sincronización.

Para MOSI y MISO las siglas hacen referencia a Master Out, Slave In, y Master In, Slave Out, respectivamente, y estas señales son conocidas como señales de comunicación o de datos.

El diagrama de conexión de los dos dispositivos se muestra en la siguiente figura:

Figura 9. **Diagrama de conexión de Orange Pi y NRF24L01**



Fuente: elaboración propia, empleando programa Paint.

3.2.1.3. **Caja del mando central**

La caja del mando central está elaborada a partir de un diseño que fue creado específicamente para el resguardo o protección de la placa Orange Pi Pc Plus, el diseño está elaborado para ser realizado en impresora 3-D. En las figuras 10 y 11 se muestra la caja original y la caja modificada específicamente para este proyecto.

Figura 10. **Caja original para Orange Pi Pc Plus**



Fuente: *Caja para Orange Pi.*

<https://d2t1xqejof9utc.cloudfront.net/screenshots/pics/7912e328b87d4875cc896725e8c3b799/large.JPG>. Consulta: 12 de septiembre de 2018.

Figura 11. **Caja modificada para Orange Pi Pc Plus y NRF24L01**



Fuente: elaboración propia, empleando programa Paint.

Debido a que se agregó el módulo inalámbrico al diseño original que está en la libertad de ser modificado, las dimensiones originales de la caja se

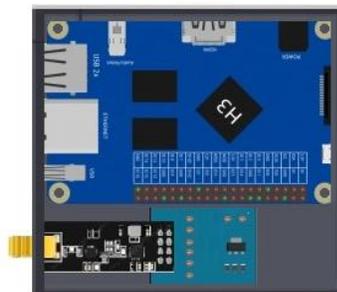
modificaron y, teniendo en cuenta las dimensiones del módulo inalámbrico, se realiza la modificación y las dimensiones de la nueva caja se muestran en la figura 12, y en la figura 13 se muestra la caja sin tapa desde vista superior.

Figura 12. **Dimensiones de la caja para Orange Pi y NRF24L01**



Fuente: elaboración propia, empleando programa Microsoft Word 2016.

Figura 13. **Vista superior de la caja sin tapa**



Fuente: elaboración propia, empleando programa Paint.

3.2.2. Microcontrolador maestro

El microcontrolador maestro está conformado por varios módulos y elementos electrónicos, para la elaboración de este diseño es necesaria la elaboración de una tarjeta de circuito impreso, la cual incluye la conexión de todos los componentes y tiene el tamaño adecuado que se ajusta al modelo de caja diseñado para este microcontrolador.

3.2.2.1. Funciones del microcontrolador maestro

Este microcontrolador está diseñado para funcionar de una manera cuasi autónoma, ya que está en libertad de ejecutar algunas funciones por cuenta propia sin depender del mando central, sin embargo, la mayoría de funciones dependen de las órdenes del mando central, las funciones del sistema se enlistan a continuación:

- Iluminación led
- Detección de movimiento
- Recepción y aprendizaje de controles remotos por IR
- Transmisión de códigos de controles remotos por IR
- Detección de temperatura y humedad en el ambiente
- Detección de humo y monóxido de carbono
- Led indicador de funciones
- Detección de apertura de puertas
- Mando de apertura de puertas
- Batería recargable para iluminación de emergencia por corte de energía
- Transmisión y recepción de señales por módulo NRF24L01
- Función de alarma
- Zumbador indicador de alarma de humo o gas

3.2.2.2. Componentes de microcontrolador maestro

Los componentes principales para elaborar este diseño del microcontrolador maestro se muestran en la siguiente lista:

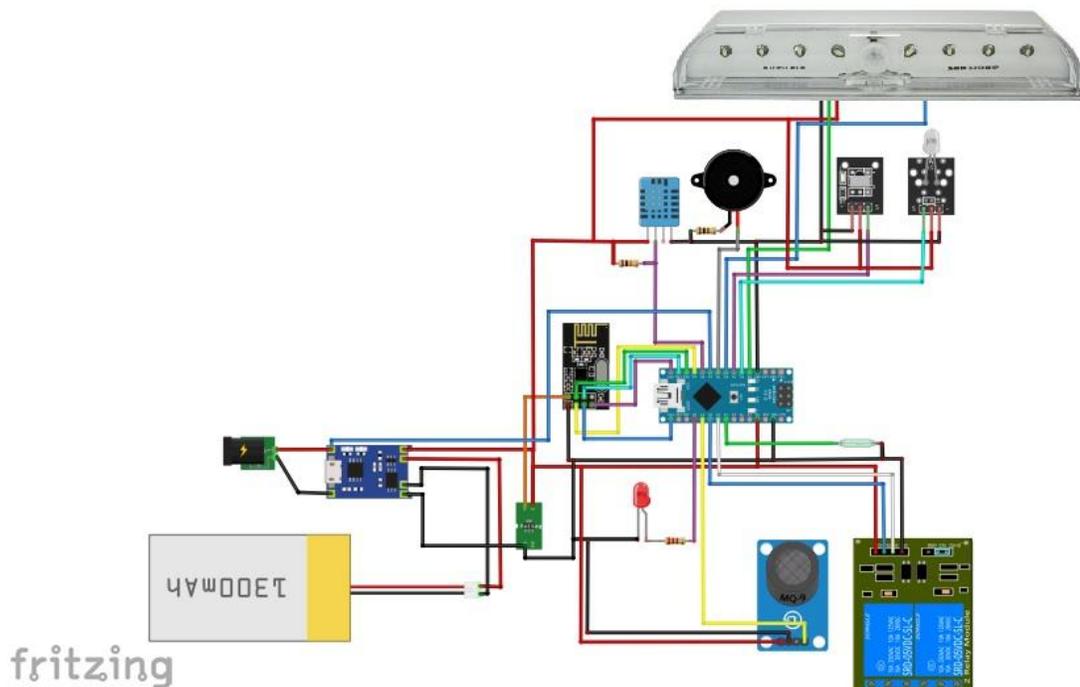
- Arduino Nano
- Módulo de iluminación led con sensor PIR HC-SR501 o de movimiento
- Módulo KY-005, emisor de IR
- Sensor receptor infrarrojo
- Sensor de temperatura DHT11
- Módulo inalámbrico NRF24L01
- Zumbador o transductor electroacústico
- 2 relé SRD-05VDC-SL-C
- Módulo de carga de baterías de litio, TP4056
- Led indicador
- Plug de poder
- Batería de litio de 3,3 voltios
- Tarjeta de circuito impreso
- Fuente de alimentación de 5 voltios de corriente directa

3.2.2.3. Conexión de microcontrolador maestro

La conexión del microcontrolador maestro es la más extensa debido al número de sensores y actuadores conectados a este, ocupa casi en su totalidad los pines de propósito general del microcontrolador Arduino Nano, siendo necesario el uso de entradas y salidas digitales y analógicas. La comunicación es por medio del módulo inalámbrico NRF24L01, por medio de este módulo el microcontrolador se encarga de la recepción y transmisión de información al módulo central como a los microcontroladores esclavos.

Al microcontrolador maestro es al único dispositivo que se le agrega una batería recargable de litio en conjunto con un módulo de recarga, este conjunto se encarga de convertir este dispositivo en una lámpara de emergencia en caso de corte de energía, la lámpara que se utiliza es de iluminación led e incorpora un sensor PIR. Las conexiones de estos elementos, junto con los demás módulos, se pueden ver en la figura 14.

Figura 14. Diagrama de conexión de microcontrolador maestro

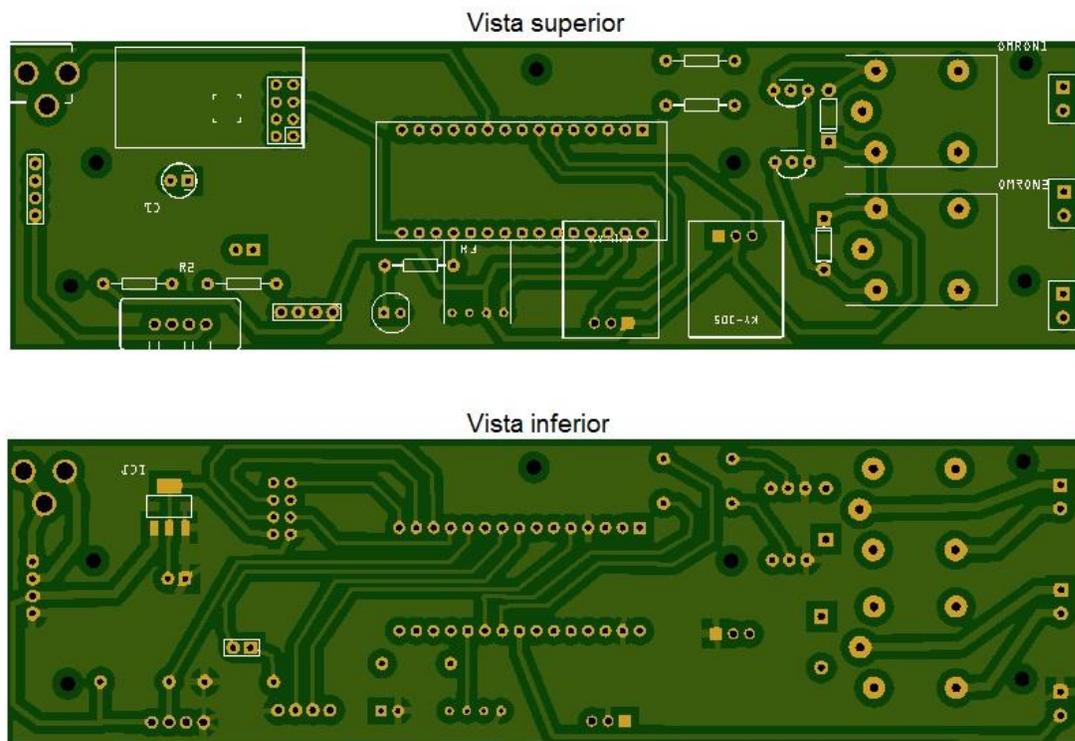


Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.2.4. Diseño de placa de circuito Impreso

El diseño del circuito impreso fue elaborado cuidadosamente, tomando en cuenta el tamaño del dispositivo de iluminación led y de la posición donde debe ser colocado el mismo, todos los módulos están organizados de tal manera que puedan salir al exterior de la caja de este dispositivo para realizar sus debidas funciones.

Figura 15. **Diseño del PCB de microcontrolador maestro**



Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.2.5. Diseño de la caja

La caja del microcontrolador maestro está diseñada para que sea de un tamaño reducido y que tenga la capacidad de proteger la placa de circuito impreso que contiene todos los módulos de este dispositivo, su tamaño y forma se basan en principio en la forma de la lámpara de iluminación led que se agrega en la parte superior de la caja, y en segundo lugar se toma en cuenta el sitio donde debe estar posicionada, dado que tiene que tener una vista que cubra la mayor cantidad de área del ambiente que se encuentra a su cargo.

La caja cuenta con las dimensiones necesarias para abarcar todos los componentes de este microcontrolador, tomando en cuenta los agujeros necesarios para que los módulos instalados puedan realizar su función de manera correcta, y que además pueda albergar la batería de litio y el módulo encargado de la carga de la batería.

Figura 16. Vista frontal de la caja

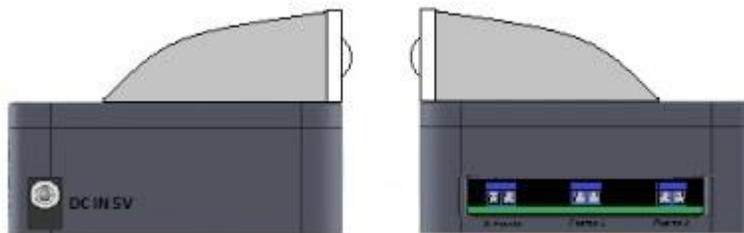


Fuente: elaboración propia, empleando programa Paint.

En la vista frontal mostrada en la figura 16 se puede observar que ahí se encuentran todos los sensores y el led indicador e infrarrojo, así como la lámpara de iluminación led sobre la caja.

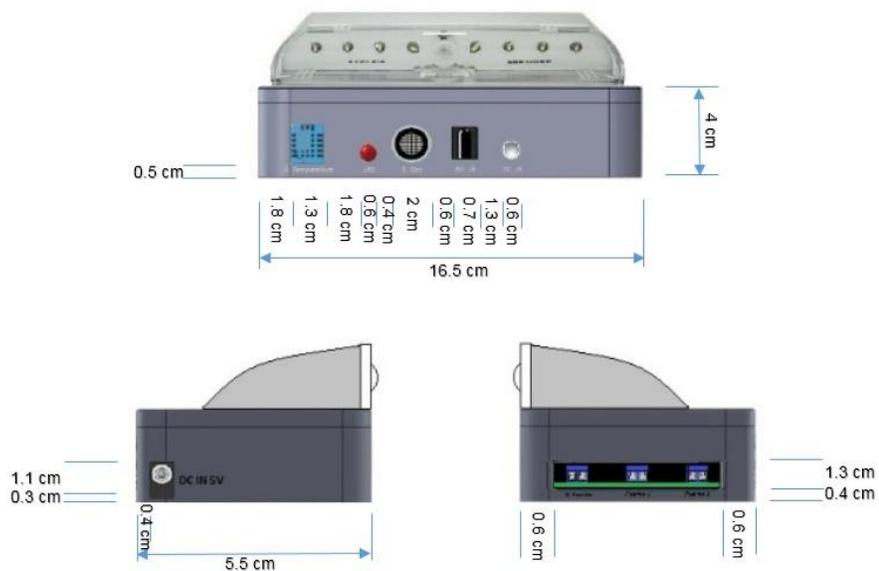
En los lados laterales de la caja se encuentran el Jack de energía y dos salidas para apertura de puertas o activación de algún otro dispositivo y también la entrada del sensor de detección de puerta abierta.

Figura 17. **Vistas laterales de la caja**



Fuente: elaboración propia, empleando programa Paint.

Figura 18. **Dimensiones de la caja**



Fuente: elaboración propia, empleando programa Microsoft Word 2016.

3.2.3. Microcontroladores esclavos

En el diseño de este sistema existen dos tipos de microcontrolador tipo esclavo, los dos microcontroladores ejecutan dos funciones diferentes y el diseño de ambos es distinto, los dos microcontroladores que serán tratados en este caso son considerados como esclavos, dado que son los últimos dispositivos en la red a los que se les envía una instrucción desde el mando central. A diferencia del microcontrolador maestro, estos microcontroladores esclavos son completamente dependientes de las instrucciones que transmite el mando central para poder realizar su función.

El primer microcontrolador esclavo tiene como función principal el control de un grupo de 6 tomacorrientes, el cual provee de 120 voltios de corriente alterna a una variedad de electrodomésticos, estos son conectados de la manera tradicional a través de un tomacorriente para así ser controlado el suministro de energía a los dispositivos por medio de la interfaz web.

El segundo microcontrolador esclavo tiene como función el manejo de los interruptores de energía que controlan la iluminación del ambiente, el cual puede recibir la instrucción de un usuario de la manera tradicional o por medio de una instrucción del mando central a través de la interfaz web.

Ambos microcontroladores esclavos se comunican directamente a su microcontrolador maestro diseñado específicamente para ambos, ya que comparten el mismo tipo de tuberías en la comunicación inalámbrica, las instrucciones del mando central son recibidas primero por el microcontrolador maestro y luego este se encarga de reenviar las instrucciones al microcontrolador esclavo que corresponda.

3.2.3.1. Microcontrolador esclavo 1

El primer microcontrolador esclavo está conformado por varios componentes electrónicos, a diferencia del microcontrolador maestro, este microcontrolador en su mayoría es circuitería y el único módulo que utiliza es el de comunicación NRF24L01 y un Arduino Nano. Al igual que el microcontrolador maestro, este cuenta con el diseño de una placa de circuito impreso que contiene todos los elementos necesarios. La función de este microcontrolador esclavo es controlar el estado de 6 tomas de corriente alterna, de los cuales 4 son controlados por relé y dos son controlados por un variador de voltaje o *dimmer*.

3.2.3.1.1. Funciones

Este microcontrolador cuenta únicamente con dos funciones, las cuales son el control del estado de encendido o apagado de 4 tomacorrientes, que pueden alimentar electrodomésticos o algún otro aparato eléctrico que se alimente de 120 voltios de corriente alterna, y el control de 2 tomacorrientes que tienen la función de variar el voltaje por medio de un *dimmer*, esto para el control de la intensidad de alguna lámpara decorativa, el manejo de la velocidad de algún ventilador o algún otro aparato que permita variar el voltaje de entrada. En resumen las funciones se enlistan a continuación:

- 4 tomacorrientes de 0 ó 120 voltios de corriente alterna.
- 2 tomacorrientes que pueden variar de 0 a 120 voltios de corriente alterna.

3.2.3.1.2. Componentes

Los componentes principales utilizados en el diseño de este microcontrolador se muestran en la siguiente lista:

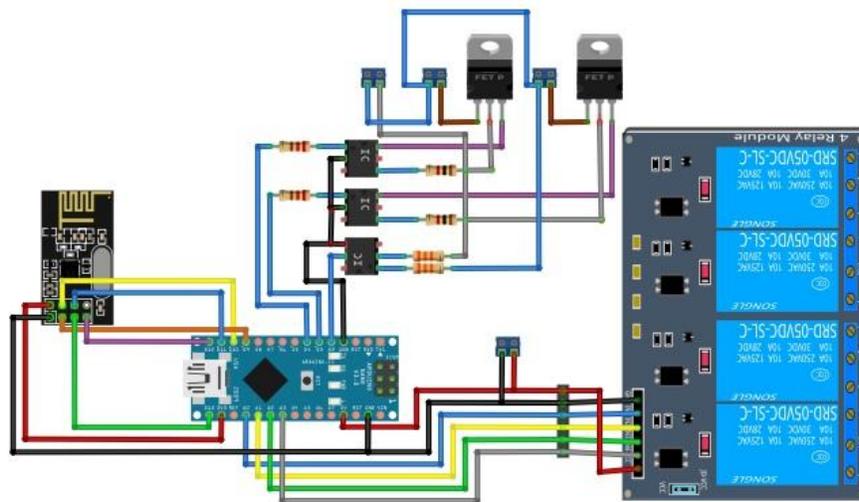
- Arduino Nano.
- Modulo inalámbrico NRF24L01.
- 4 relé SRD-05VDC-SL-C.
- 1 opto-acoplador TIL186-4.
- 2 opto-triac moc 3020.
- 2 *triacs* BT138 600E.
- 4 transistores 2n3904.
- 2 resistencias de 220, 1000 y 33000 ohmios.
- Fusible de 500 voltios de corriente alterna y 20 amperios.
- Fuente de poder de uso interno de 5 voltios de corriente directa y 1 amperio.
- 6 tomacorriente individuales.

3.2.3.1.3. Conexión

Este diseño cuenta con 2 variadores de voltaje, los cuales son posibles por el uso de componentes electrónicos como los *triacs*, para el control de la corriente alterna, y de un optoacoplador para la detección de cruce por cero en la corriente alterna, lo cual permite la sincronización de la señal de la red eléctrica para así poder controlar la tasa de eficiencia de la señal de salida. Para las otras 4 salidas restantes, solo es necesario el uso de transistores bipolares para la amplificación de corriente para la activación de los relés, los cuales se encargan de funcionar como un interruptor para las salidas.

Las conexiones de estos componentes electrónicos se muestran en la siguiente figura:

Figura 19. Diagrama de conexión



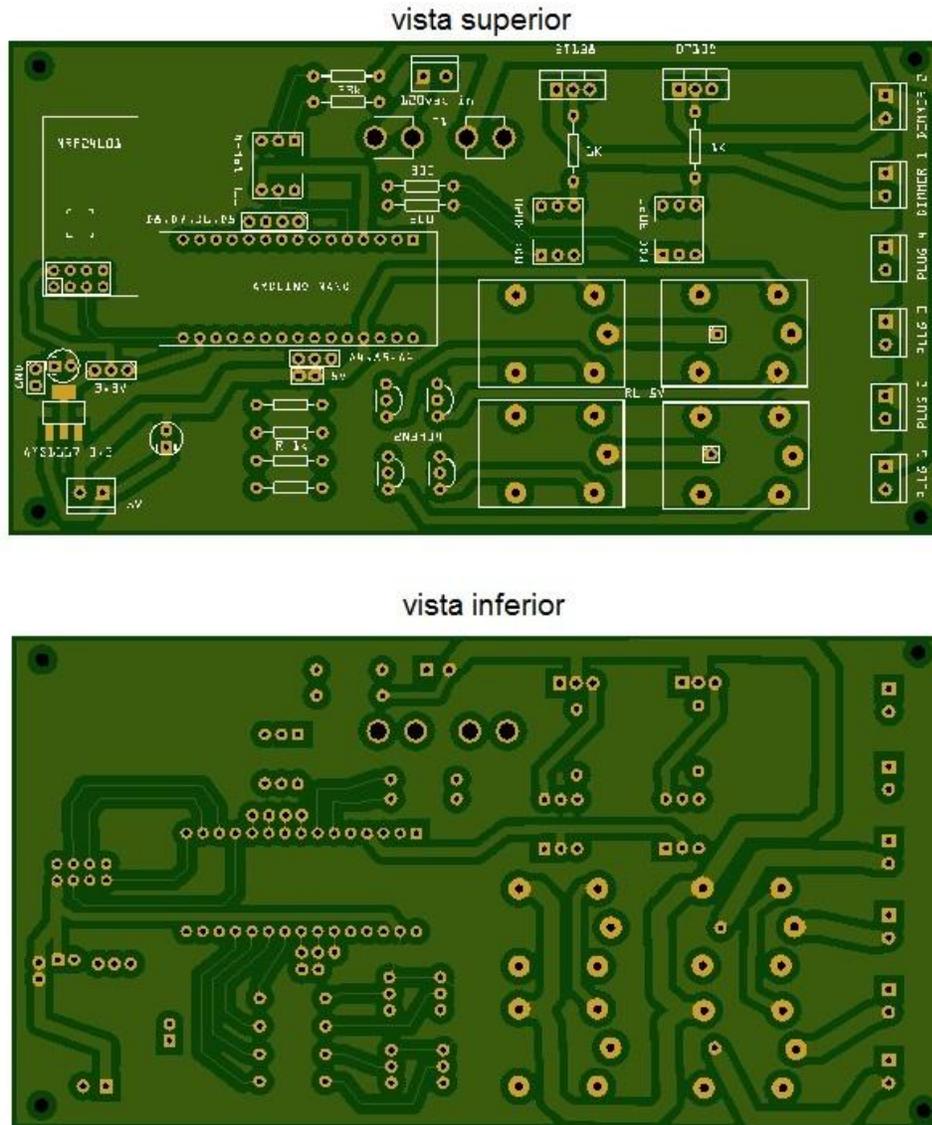
fritzing

Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.3.1.4. Diseño de circuito impreso

El diseño de la placa de circuito impreso fue elaborado para ser realizado en una placa que posea cobre en ambas caras, esto para tener un mejor orden en las posiciones de los elementos y también reducir el tamaño de la placa, para evitar que las dimensiones del diseño armado no se ajusten a los lugares donde pueda ser conectado. El diseño de este circuito no incluye la fuente de alimentación, así que debe agregarse una y conectarse en el lugar indicado, y colocarse dentro del diseño armado.

Figura 20. Diseño del PCB

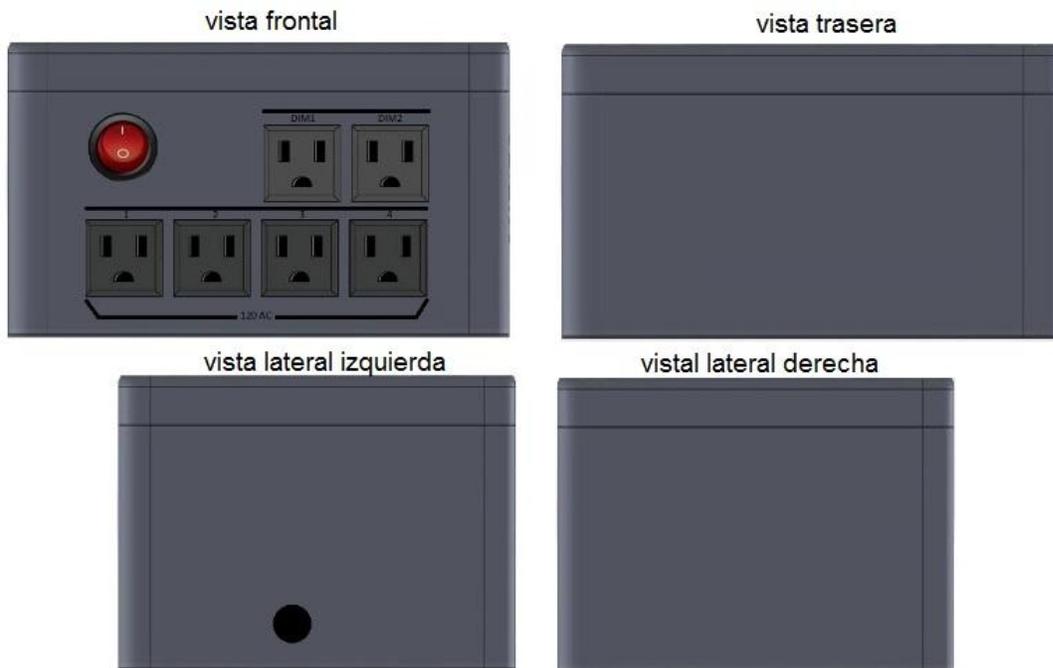


Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.3.1.5. Diseño de la caja

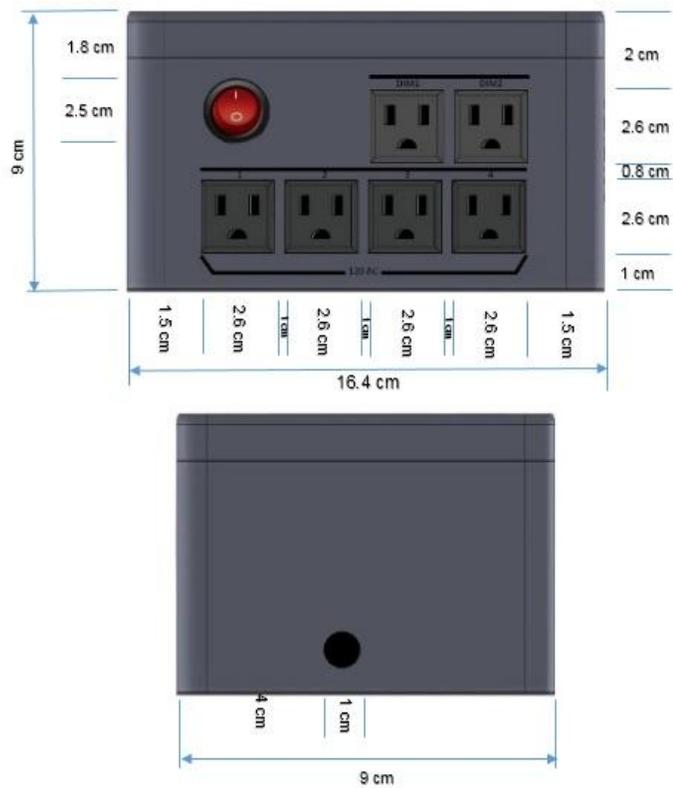
El diseño de la caja está realizado conforme a las dimensiones del diseño de la placa de circuito impreso y el tamaño de las tomas de corriente individuales, los elementos internos se ordenan de manera que se reduzcan las dimensiones de la misma y así no ocupar más lugar del necesario y que no presente gran dificultad al momento de acomodarla en el lugar de la instalación. Las tomas de corriente están divididas en dos filas, las cuales separan las tomas con *dimmer* y las tomas de interrupción, ambas se encuentran indicadas en el diseño de la caja.

Figura 21. Vistas de la caja



Fuente: elaboración propia, empleando programa Paint.

Figura 22. Dimensiones de la caja



Fuente: elaboración propia, empleando programa Microsoft Word 2016.

3.2.3.2. Microcontrolador esclavo 2

El segundo microcontrolador esclavo, al igual que el primero, en su mayoría está conformado por circuitería, pero en este caso el único módulo con el que cuenta es con el de comunicación inalámbrica NRF24L01, ya que, por el reducido espacio, el microcontrolador Atmega 328 se encuentra soldado directamente en el diseño de la placa de circuito impreso, y también a diferencia de los demás microcontroladores este cuenta con su propia fuente de alimentación, conectándose directo a la red de energía.

3.2.3.2.1. Funciones

La función de este microcontrolador permite el encendido o apagado de 2 luces de una habitación por medio de botones como se realiza convencionalmente o por medio de las acciones realizadas en la interfaz web, esto permite a este microcontrolador, a diferencia del esclavo 1, interactuar con este de dos maneras, ya sea de manera presencia o remota. También cabe resaltar que el manejo de las dos luces a su cargo lo hace a través de variadores de voltaje, lo cual permite variar la iluminación al gusto y así tener un mejor manejo de la iluminación y del consumo de energía.

3.2.3.2.2. Componentes

Los componentes principales utilizados en el diseño de este microcontrolador se muestran en la siguiente lista:

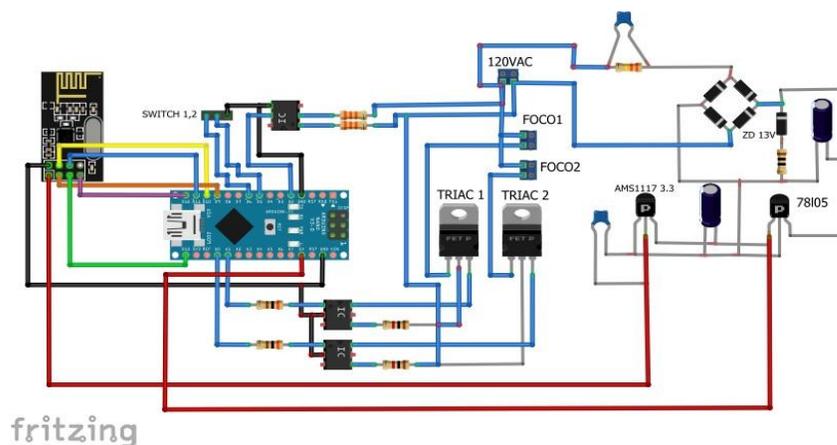
- Microcontrolador Atmega 328p AU 1612
- Módulo inalámbrico NRF24L01
- 1 opto-acoplador TIL186-4
- 2 opto-*triac* moc 3020
- 2 *triacs* BT138 600E
- 2 resistencias de 1000, 10000, 33000 ohmios
- 1 resistencia de 10 y 330,000 ohmios
- 1 puente de diodos RB-154
- 1 diodo *zener* de 13 voltios a 1 amperio
- 1 capacitor cerámico de 1 microfaradios a 250 voltios
- 1 capacitor cerámico de 100 nanofaradios a 100 voltios
- 2 capacitores electrolíticos de 100 microfaradios a 10 voltios
- 2 botones tipo *push*

- 1 regulador de voltaje 78I05
- 1 regulador de voltaje AMS1117-3.3
- 3 borneras de 2 polos de 2,54 milímetros
- 1 fusible de 250 voltios de corriente alterna a 3 amperios

3.2.3.2.3. Conexión

La conexión de este diseño inicia desde la entrada de 120 voltios de corriente alterna, este voltaje alimenta al microcontrolador sin utilizar un transformador o fuente de energía, ya que la fuente utilizada en este diseño es de tipo capacitiva, lo cual permite suministrar la energía necesaria para el funcionamiento de este circuito sin la necesidad de ocupar una fuente convencional de 5 voltios, esto se hace debido al poco espacio que hay en el lugar de la instalación, la conexión de esta fuente en conjunto con los demás componentes electrónicos se muestran en la siguiente figura:

Figura 23. Diagrama de conexión

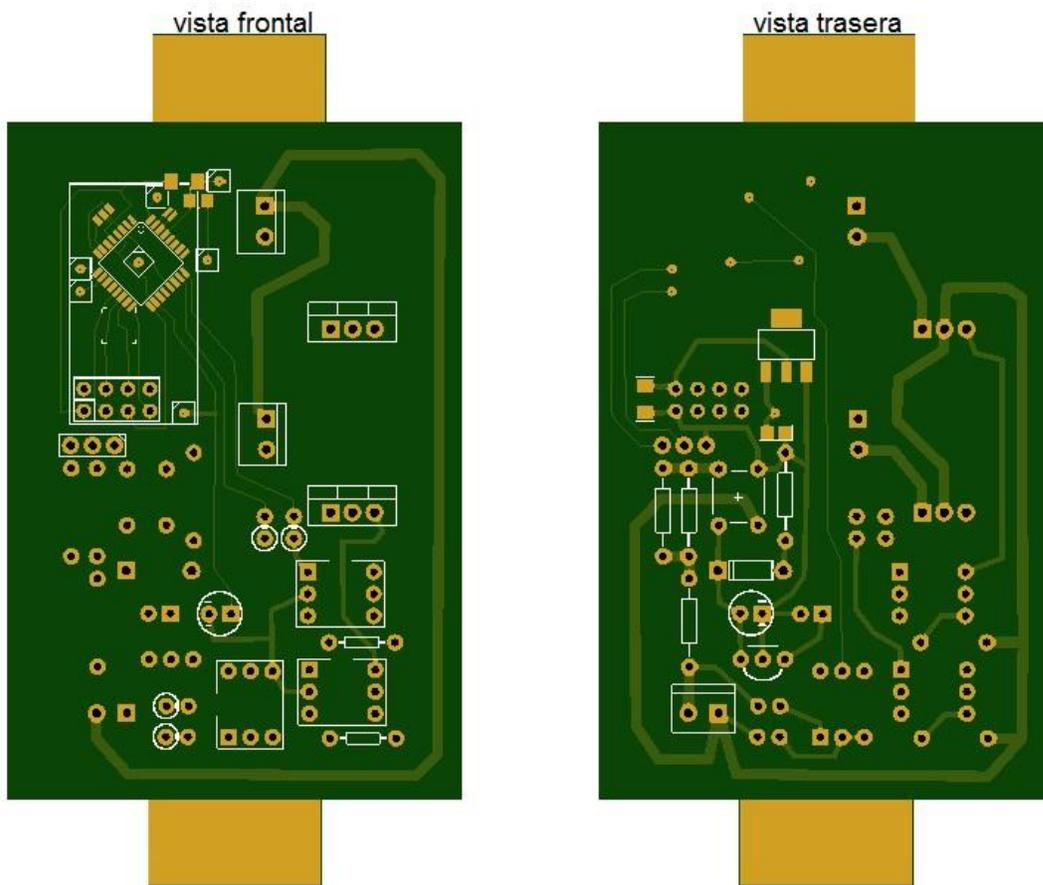


Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.3.2.4. Diseño de circuito impreso

El diseño de placa está hecho de tal manera que sus dimensiones puedan encajar en la caja donde normalmente está la instalación de los interruptores de la iluminación de una habitación o ambiente, debido al escaso espacio es necesaria la instalación del microcontrolador de manera directa a la placa. Este diseño se muestra a continuación en la siguiente figura:

Figura 24. Diseño del PCB

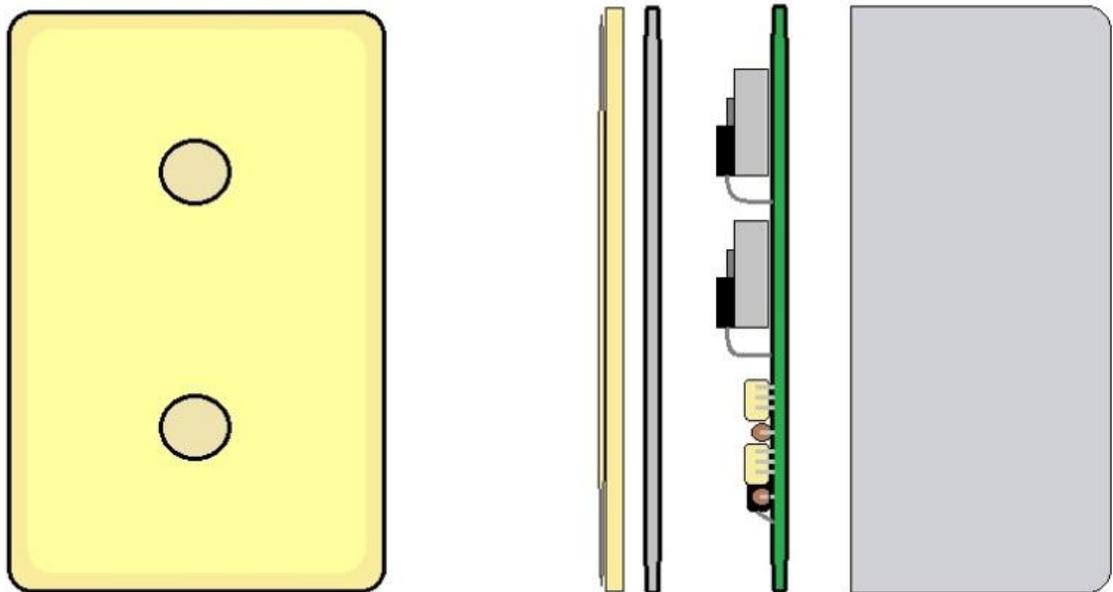


Fuente: elaboración propia, empleando programa Fritzing Beta versión 0.9.3.

3.2.3.2.5. Diseño de la caja

A diferencia de los demás diseños de microcontroladores, este es el único que no cuenta con un diseño de caja, esto debido a que se coloca en una instalación propia de la casa, cambiando únicamente el tipo de interruptor que comúnmente se utiliza, tal y como se muestra en la siguiente figura:

Figura 25. Vistas de la caja



Fuente: elaboración propia, empleando programa Paint.

3.3. Instalación del sistema

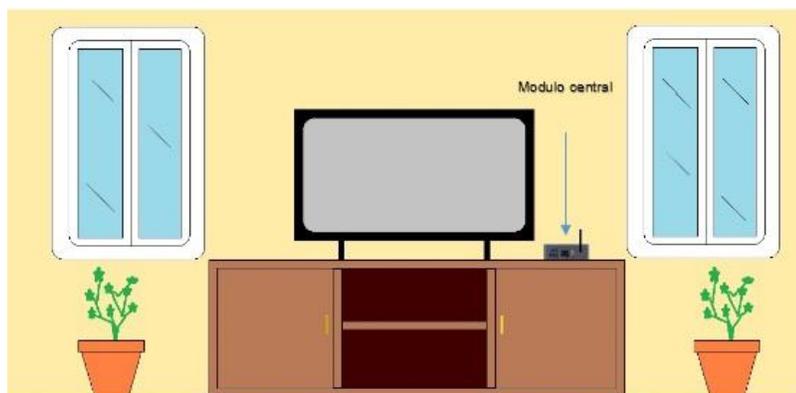
Los lugares propuestos para la instalación de cada uno de estos dispositivos van de acuerdo a sus funciones y también en donde pueden ejecutar de mejor manera las mismas. Las instalaciones de estos dispositivos en su mayoría son simples a excepción del microcontrolador esclavo 2, el cual

requiere de una modificación simple en la instalación eléctrica de la habitación o lugar donde sea instalado.

3.3.1. Instalación del mando central

La instalación del mando central es simple, ya que solo requiere conectarse en un lugar cercano a un televisor con entrada *HDMI* o entrada de video compuesto y también con disponibilidad de una toma de corriente para la conexión de una fuente de poder de 5 voltios de corriente directa a 3 amperios, para la alimentación del mando central.

Figura 26. Instalación del mando central



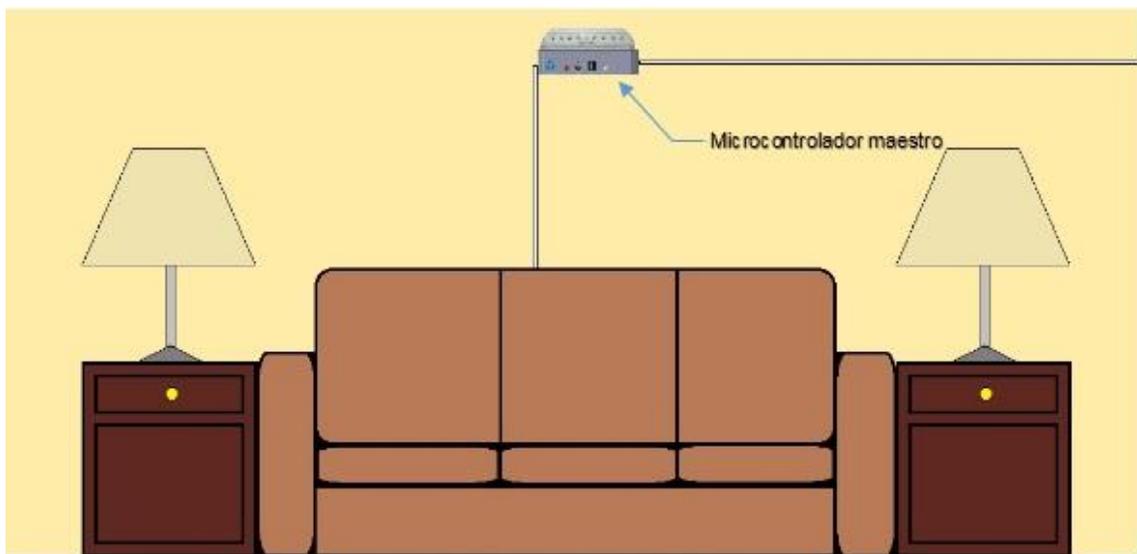
Fuente: elaboración propia, empleando programa Paint.

3.3.2. Instalación de microcontrolador maestro

La instalación del microcontrolador central requiere de la conexión de algunos elementos electrónicos extras al módulo para completar todas sus funciones, las funciones que requieren de estos elementos son para la

detección de puerta abierta y de apertura de la misma, la función de apertura de puerta tiene la capacidad para 2 puertas. La posición óptima para la instalación de este dispositivo es el lugar que abarque la mayoría de área del lugar donde se encuentre instalado, ya que es necesario para la detección de movimiento y la transmisión de IR. La instalación requiere de una toma de corriente cercana para la conexión de una fuente de 5 voltios de corriente directa a 1 amperio para la alimentación del mismo.

Figura 27. **Instalación de microcontrolador maestro**



Fuente: elaboración propia, empleando programa Paint.

3.3.3. **Instalación de microcontrolador esclavo 1**

La instalación de este microcontrolador es sencilla, pues basta con ser conectado a una toma de corriente y encenderlo. De preferencia este microcontrolador se debe conectar a la toma de corriente de un regulador de

voltaje y acomodarlo en el lugar más conveniente que sea posible y que esté en un lugar donde se encuentre una mayoría de electrodomésticos, para así poder centralizar el control de distribución de energía de los mismos.

Figura 28. **Instalación de microcontrolador esclavo 1**

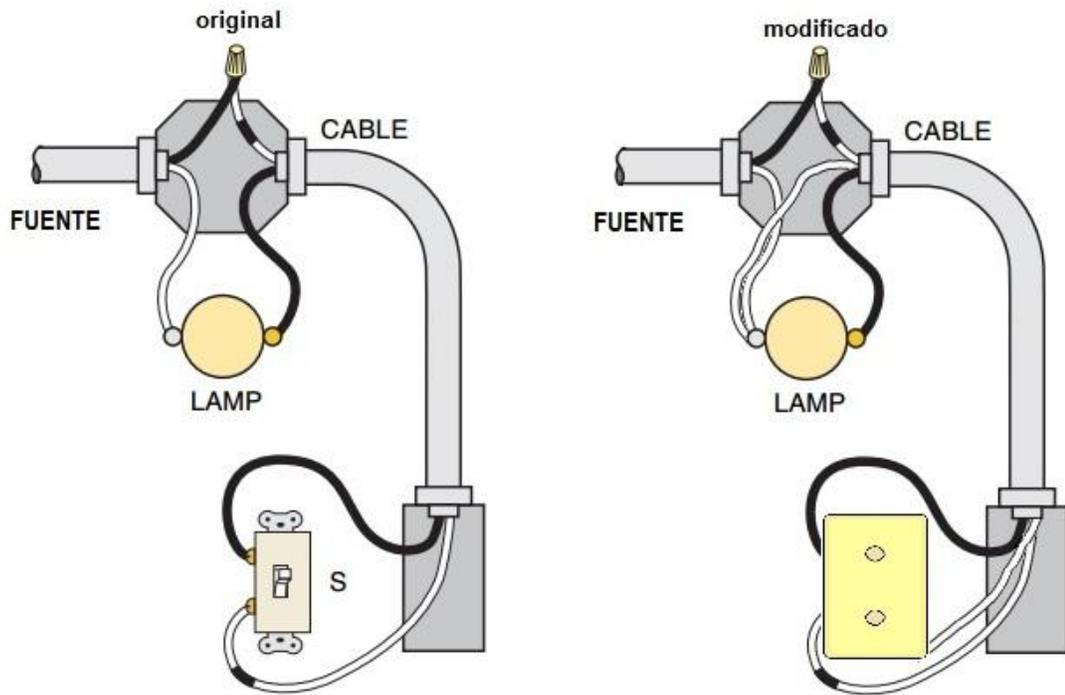


Fuente: elaboración propia, empleando programa Paint.

3.3.4. Instalación de microcontrolador esclavo 2

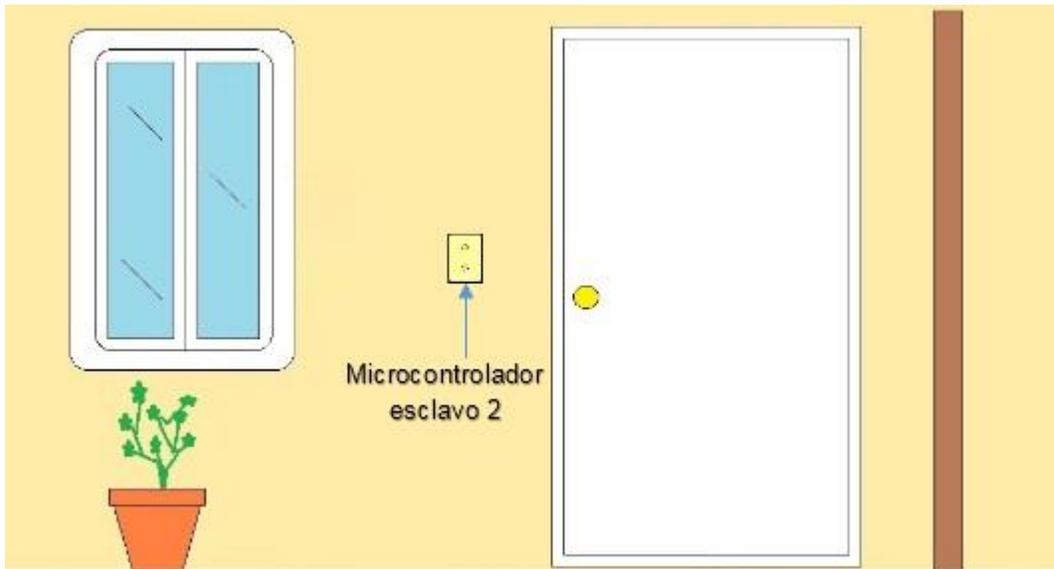
La instalación del microcontrolador esclavo 2 es la única que requiere hacer un cambio significativo en la instalación eléctrica de cada lugar donde es instalado, ya que se requiere agregar un cable a la instalación para la conducción de la fase a la caja del interruptor, para así poder suministrar 120 voltios de corriente alterna al circuito de este microcontrolador, el cambio a realizar se muestra en la siguiente imagen:

Figura 29. **Modificación a la instalación del interruptor**



Fuente: *Modificación del interruptor*. <http://1.bp.blogspot.com/-ENoLW-HsjsY/UsIC-SeioRI/AAAAAAAAAZw/htLtFziZwbw/s1600/Conexion+de+interruptor+simple+con+fuentes+en+l+lampara.png>. Consulta: 18 de septiembre de 2018.

Figura 30. **Instalación de microcontrolador esclavo 2**



Fuente: elaboración propia, empleando programa Paint.

3.3.5. **Instalación en la casa**

La instalación de todos los dispositivos en la casa puede variar según los ambientes y lugares que se deseen automatizar, de eso dependerá la distribución de los mismos. El mando central se recomienda que se instale en el lugar más céntrico de la casa, esto podría ser, por ejemplo, la sala, aunque dependerá del usuario. En la figura 31 se puede observar un ejemplo de una distribución del sistema en una casa.

Figura 31. **Instalación del sistema en una casa**



Fuente: *Instalación del sistema*. https://images-platform.99static.com/15Xz_xiAJZLLVgxUQyQZGDV8pDE=/500x500/top/smart/99designs-contests-attachments/43/43190/attachment_43190896. Consulta: 18 de septiembre de 2018.

4. SOFTWARE DE LOS ELEMENTOS DEL SISTEMA

En el diseño de este sistema es necesario el uso del entorno de programación de Arduino, el cual es utilizado para programar todos los microcontroladores, siendo estos el microcontrolador maestro y los dos microcontroladores esclavos, cada microcontrolador tiene su propio código para ejecutar sus funciones correspondientes.

A diferencia de los microcontroladores, el mando central no es programado por el entorno de programación de Arduino, ya que el mando central utiliza como cerebro el mini ordenador Orange Pi, el cual ejecuta un sistema operativo Linux, en el cual se utilizan varios lenguajes de programación, entre los principales se pueden mencionar Python y PHP, los cuales se encargan de la ejecución de las instrucciones y la gestión de las mismas hacia los microcontroladores y brindar una interfaz en web para el manejo del usuario.

4.1. Programación del microcontrolador maestro

Este código es el encargado del manejo de todos los sensores del sistema y de recibir las instrucciones directas del mando central y reenviarlas a los microcontroladores esclavos. En este código se incluyen todas las direcciones de los dispositivos esclavos que pertenezcan al área que estén a su cargo, así como la dirección del mando central, el código siempre es el mismo para todo microcontrolador maestro, solo se debe cambiar la lista de direcciones que se encuentra en el código y que corresponda con el mismo canal de dirección de los microcontroladores esclavos, la dirección es un vector de 5 números hexadecimales y el primero es la dirección del mando central.

A continuación se muestra el código utilizado para el microcontrolador maestro:

```
#include<SPI.h>
#include<RF24.h>
#include <IRremote.h>
#include <stdlib.h>
#include <DHT.h>

//DIRECCIONES DE LOS CANALES
const uint64_t pipes[6] =
{0x504f525441130LL,0x53657441130LL,0x53657441131LL,0x53657441132LL,0x53657441133LL,0x53657441134LL};
RF24 radio(9, 10);// ce, csn pines nrf24l01

//APERTURA DE PUERTAS 1 Y 2 Y SENSORES DE APERTURA
const byte PUERTA1 = A2;
const byte PUERTA2 = A3;
const byte SENSPUERTA = A4;
boolean P1 = false;
boolean P2 = false;
boolean P = true;

//BANCO DE BATERIAS
byte pin5v = 7;
boolean batt = true;

//SENSOR DE GAS Y HUMO
const byte MQ_PIN = A1;
byte buzzer = 6;
byte ngas1_muestra = 0;
byte ngas2_muestra = 0;
byte ngas3_muestra = 0;
int ttemp = 98;
unsigned int t0 = 0;
unsigned int t01 = 0;
unsigned int t_muestras = 2000;
boolean gas1 = false;
boolean gas2 = false;
boolean pulso = false;
boolean bandera_ngas3 = false;
boolean bgas = true;

//SENSOR DE TEMPERATURA Y HUMEDAD
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
String comandoT = "";

//TRANSMISION Y RECEPCION DE IR
IRsend irsend;
int RECV_PIN = 4;
IRrecv irrecv(RECV_PIN);
decode_results results;
char s[15];
long codigo;
long nbit;
boolean IRREC = false;
String comandoIR = "";
String codigohex = "";
String marca[]={ "sin marca", "RC5", "RC6", "NEC", "SONY", "PANASONIC", "JVC",
                "SAMSUNG", "WHYNTER", "AIWA_RC_T501", "LG", "SANYO",
                "MITSUBISHI", "DISH", "SHARP", "DENON", "PRONTO", "LEGO_PF"}; //18
```

```

//RECEPCION Y ENVIO DE MENSAJE
char receivedMessage[32] = {0};
char senderMessage[32]={0};
char text[] = "aTX0AbOK1cdz";
char text0[] = "aTX0bPREScSldz";
char text1[] = "aTX0bPREScNOdz";
char text2[] = "aTX0bPUERTAcABIERTAdz";
char text3[] = "aTX0bPUERTAcCERRADAdz";
char text4[] = "aTX0bGAScSldz";
char text5[] = "aTX0bGAScNOdz";

//LEDS
int ledPin = 5;
int ledindicador = A0;
long intensidad;
boolean parpadeo = false;

//ANALISIS DE MENSAJE
char caracter;
//variables de valores recibidos
String tx="";
String comando="";
String valor="";
String modo="";
////////////////////
String txx="";
String com="";
String val="";
String mod="";
//valores de condicion o banderas
boolean TX = false;
boolean COMANDO = false;
boolean VALOR = false;
boolean MODO = false;

//LUZ DE EMERGENCIA Y ALARMA
const int intervalo = 50;
byte conteo = 0;
byte conteo1 = 0;
int conteo2 = 0;
unsigned int tiempo0 = 0;
unsigned int tiempo01 = 0;
unsigned int tiempo02 = 0;
unsigned int tiempo03 = 0;
unsigned int tiempo04 = 0;
unsigned int tiempo1 = 0;
boolean estado = LOW;
boolean sensor = true;
boolean encender = false;
boolean tencendido = false;
boolean apagar = false;
boolean apagarIR = false;
boolean Lemergencia = false;
boolean Lemergencia1 = false;
boolean alarma = false;

void setup(void){
  while(!Serial);
  Serial.begin(9600);
  dht.begin();
  pinMode(pin5v, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(PUERTA1, OUTPUT);
  pinMode(PUERTA2, OUTPUT);
  pinMode(SENSPUERTA, INPUT);
}

```

```

pinMode(ledindicador, OUTPUT);
digitalWrite(PUERTA1, HIGH);
digitalWrite(PUERTA2, HIGH);
digitalWrite(SENSPUERTA, HIGH);
radio.begin();
radio.setPALevel(RF24_PA_MAX);
radio.setChannel(0x4c);
radio.openReadingPipe(0,pipes[0]);
radio.openReadingPipe(1,pipes[1]);
radio.openReadingPipe(2,pipes[2]);
radio.openReadingPipe(3,pipes[3]);
radio.openReadingPipe(4,pipes[4]);
radio.openReadingPipe(5,pipes[5]);
radio.enableDynamicPayloads();
radio.powerUp();
radio.startListening();
}

void loop(void){
  if(radio.available()){
    radio.read(receivedMessage, sizeof(receivedMessage));
    Serial.println(receivedMessage);
    AnalizarComando();
    ruteo();
    logica();
  }
  funciones();
  delay(10);
}
//RUTEO DE MENSAJES
void ruteo(){
  if (txx == "TX0"){
    envio(receivedMessage, 0, sizeof(receivedMessage));
  }else if (txx == "TX1"){
    envio(text, 0, sizeof(text));
    logica();
  }else if (txx == "TX2"){
    envio(receivedMessage, 1, sizeof(receivedMessage));
  }else if (txx == "TX3"){
    envio(receivedMessage, 2, sizeof(receivedMessage));
  }else if (txx == "TX4"){
    envio(receivedMessage, 3, sizeof(receivedMessage));
  }else if (txx == "TX5"){
    envio(receivedMessage, 4, sizeof(receivedMessage));
  }else if (txx == "TX6"){
    envio(receivedMessage, 5, sizeof(receivedMessage));
  }
  txx = "";
}
//ENVIO
void envio(char mensaje[], int canal, int tam){
  radio.stopListening();
  radio.openWritingPipe(pipes[canal]);
  radio.write(mensaje, tam);
  radio.startListening();
}
//LOGICA
void logica(){
  //PUERTAS
  if (com == "PUERTA1"){
    P1 = true;
    com="";
  }
  if (com == "PUERTA2"){
    P2 = true;
    com="";
  }
}

```

```

}
//ILUMINACION LED
if(com=="LED"){
  if(val=="ON"){
    digitalWrite(ledPin, HIGH);
    Lemergencia1 = Lemergencia;
    Lemergencia = false;
  }else if(val=="OFF"){
    digitalWrite(ledPin, LOW);
    Lemergencia = Lemergencia1;
  }
  com="";
  val="";
}
//DIMMER DE ILUMINACION LED
if(com=="DIM"){
  intensidad=val.toInt();
  analogWrite(ledPin, intensidad*2.55);
  com="";
  val="";
}
//MODULO EMISOR IR
if (mod != ""){
  val.toCharArray(s, 15);
  codigo = strtoul(s,NULL,16);
  nbit=mod.toInt();

for (int i = 0; i < 5; i++) {
  if (com=="RC5"){ irsend.sendRC5(codigo, nbit); }
  if (com=="RC6"){ irsend.sendRC6(codigo, nbit); }
  if (com=="NEC"){ irsend.sendNEC(codigo, nbit); }
  if (com=="SONY"){ irsend.sendSony(codigo, nbit); }
  if (com=="PANASONIC"){ irsend.sendPanasonic(codigo, nbit); }
  //if (com=="JVC"){ irsend.sendJVC(codigo, nbit); }
  //if (com=="SAMSUNG"){ irsend.sendSamsung(codigo, nbit); }
  if (com=="WHYNTER"){ irsend.sendWhynter(codigo, nbit); }
  //if (com=="AIWA_RC_T501"){ irsend.sendSony(codigo, nbit); }
  if (com=="LG"){ irsend.sendLG(codigo, nbit); }
  //if (com=="SANYO"){ irsend.sendSanyo(codigo, nbit); }
  //if (com=="MITSUBISHI"){ irsend.sendMitsubishi(codigo, nbit); }
  if (com=="DISH"){ irsend.sendDISH(codigo, nbit); }
  if (com=="SHARP"){ irsend.sendSharp(codigo, nbit); }
  if (com == "DENON"){ irsend.sendDenon(codigo, nbit); }
  //if (com=="PRONTO"){ irsend.sendPronto(codigo, nbit); }
  //if (com=="LEGO_PF"){ irsend.sendSony(codigo, nbit); }
  delay(26);
}
  com="";
  val="";
  mod="";
}
//MODULO RECEPTOR IR
if (com == "IRREC"){
  IRREC = true;
  apagarIR = true;
  irrecv.enableIRIn();
  estado = HIGH;
  tiempo04 = millis();
  com="";
}
//MODULO DE ALARMA
if (com == "ALARMA"){
  if (val == "ON"){
    digitalWrite(ledPin, LOW);
    Lemergencia1 = Lemergencia;
    Lemergencia = false;
  }
}

```

```

conteo1 = 0;
alarma = true;
encender = false;
tencendido = false;
apagar = false;
}else if(val == "OFF"){
  alarma = false;
  Lemergencia = Lemergencia1;
}
com="";
val="";
}
//TEMPERATURA
if (com == "TEMP"){
  temperatura();
  com="";
}
//LUZ DE EMERGENCIA
if (com == "LEMERG"){
  if (val == "ON"){
    Lemergencia = true;
  }else if (val == "OFF"){
    Lemergencia = false;
    apagar = false;
    conteo1 = 0;
    apagar = false;
    tencendido = false;
    encender = false;
    analogWrite(ledPin, conteo1);
  }
  com="";
  val="";
}
}
//SENSOR DE MONOXIDO DE CARBONO
void gas(){
  int raw_adc = analogRead(MQ_PIN);
  float value_adc = raw_adc * (5.0 / 1023.0);
  ttemp = ttemp + 1;
  if (ttemp == 75){
    temperatura();
  }
  if (ttemp == 125){
    if (!digitalRead(SENSPUERTA)){
      envio(text2, 0, sizeof(text2));
    }else{
      envio(text3, 0, sizeof(text3));
    }
  }
  if (ttemp == 150){
    if (bitRead(PIND,2)){

    }else{
      if (alarma){
        envio(text1, 0, sizeof(text1));
      }
    }
    ttemp = 0;
  }
  if (value_adc >= 0.45 && value_adc < 1.00){
    if (ngas1_muestra < 5){
      ngas1_muestra = ngas1_muestra + 1;
      if (ngas1_muestra == 5){
        digitalWrite(ledPin, HIGH);
        Lemergencia1 = Lemergencia;
        Lemergencia = false;
      }
    }
  }
}

```

```

        gas1 = true;
    }
}
ngas3_muestra = 0;
}else if (value_adc >= 1.00){
    if (ngas2_muestra < 5){
        ngas2_muestra = ngas2_muestra + 1;
        if (ngas2_muestra == 5){
            digitalWrite(ledPin, HIGH);
            Lemergencia1 = Lemergencia;
            Lemergencia = false;
            gas2 = true;
        }
    }
    ngas3_muestra = 0;
}else{
    if (ngas3_muestra < 5){
        ngas3_muestra = ngas3_muestra + 1;
        if (ngas3_muestra == 5){
            if (ngas1_muestra == 5 || ngas2_muestra == 5){
                Lemergencia = Lemergencia1;
                if (tencendido == true){
                    tencendido = false;
                    Lemergencia = true;
                }
            }
            apagar = true;
            bandera_ngas3 = true;
        }
        ngas1_muestra = 0;
        ngas2_muestra = 0;
        digitalWrite(buzzer,LOW);
        estado = LOW;
        gas1 = false;
        gas2 = false;
    }
}
}
if (gas1 || gas2){
    if (bgas == true){
        envio(text4, 0, sizeof(text4));
        bgas = false;
    }
}else{
    if (bgas == false){
        envio(text5, 0, sizeof(text5));
        bgas = true;
    }
}
}
//TEMPERATURA
void temperatura(){
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float hic = dht.computeHeatIndex(t, h, false);
    comandoT = "aTX0bTEMPc" + String(t) + "d" + String(h) + "e" + String(hic) + "z";
    Serial.println(comandoT);
    comandoT.toCharArray(senderMessage, (comandoT.length()+1));
    envio(senderMessage, 0, (comandoT.length()+1));
    comandoT = "";
}
//FUNCIONES QUE CORREN SIEMPRE
void funciones(){
    digitalWrite(ledindicador, estado);
    if(parpadeo or encender or tencendido or apagar or apagarIR){
        tiempo1 = millis();
        if (apagarIR){

```

```

if (tiempo1 - tiempo04 >= 500*intervalo) {
  apagarIR = false;
  parpadeo = false;
  estado = LOW;
  conteo = 0;
  IRREC = false;
  codigohex = "";
  comandoIR = "";
}
}
if (apagar){
  if (tiempo1 - tiempo03 >= intervalo/5) {
    conteo1--;
    if (conteo1 == 0){
      apagar = false;
      if (!bandera_ngas3){
        Lemergencia = true;
      }else{
        bandera_ngas3 = false;
      }
    }
    analogWrite(ledPin, conteo1);
  }
}
if (tencendido){
  if (tiempo1 - tiempo02 >= 600*intervalo) {
    tencendido = false;
    apagar = true;
    tiempo03 = millis();
  }
}
if (encender){
  if (tiempo1 - tiempo01 >= intervalo/5) {
    conteo1++;
    if (conteo1 == 255){
      encender = false;
      tencendido = true;
      tiempo02 = millis();
    }
    analogWrite(ledPin, conteo1);
  }
}
if (parpadeo){
  if (tiempo1 - tiempo0 >= intervalo) {
    tiempo0 = tiempo1;
    estado = !estado;
    conteo++;
    if (conteo == 4){
      parpadeo = false;
      estado = HIGH;
      conteo = 0;
    }
  }
}
}
}
}
////RECEPTOR IR
if (IRREC == true){
  if (irrecv.decode(&results)) {
    codigohex = String(results.value, HEX);
    codigohex.toUpperCase();
    comandoIR = "aTX1IRb" + marca[results.decode_type] + "c" + codigohex + "d" + String(results.bits) + "z";
    Serial.println(comandoIR);
    tiempo04 = millis();
    apagarIR = true;
    if (results.decode_type >= 1 && results.decode_type <= 18 && results.bits != 0){
      comandoIR.toCharArray(senderMessage, (comandoIR.length()+1));
    }
  }
}
}
}
}

```



```

    encender = false;
    analogWrite(ledPin, conteo1);
    batt = false;
}
}else if (ldigitalRead(pin5v)){
    if (!batt){
        Lemergencia = true;
        batt = true;
    }
}
//TIEMPO DE MUESTRAS DE SENSOR DE GAS Y APERTURA DE PUERTAS
unsigned int t1 = millis();
if ((unsigned int)(t1 - t0) >= t_muestras){
    gas();
    if (P1){
        digitalWrite(PUERTA1, LOW);
    }else{
        digitalWrite(PUERTA1, HIGH);
    }
    if (P2){
        digitalWrite(PUERTA2, LOW);
    }else{
        digitalWrite(PUERTA2, HIGH);
    }
    t0 = millis();
    P1 = P2 = false;
}
if (gas1 || gas2){
    if ((unsigned int)(t1 - t01) >= (t_muestras - 1500)){
        t01 = millis();
        if (pulso){
            estado = HIGH;
            analogWrite(buzzer, 128);
        }else if (!pulso){
            estado = LOW;
            digitalWrite(buzzer, LOW);
        }
        pulso = !pulso;
    }
}
}
void AnalizarComando() {
    for (int i=0; i < sizeof(receivedMessage) ; i++){
        caracter = receivedMessage[i];
        if (caracter=='z'){
            VALOR=false;
            MODO=false;
            txx=tx;
            com=comando;
            val=valor;
            mod=modo;
        }
        if (MODO){
            modo += caracter;
        }
        if (caracter=='d'){
            VALOR=false;
            MODO= true;
        }
        if (VALOR){
            valor += caracter;
        }
        if (caracter=='c'){
            COMANDO=false;
            VALOR=true;
        }
    }
}

```

```

if (COMANDO){
  comando += caracter;
}
if(caracter == 'b'){
  TX=false;
  COMANDO=true;
}
if (TX){
  tx += caracter;
}
if(caracter=='a'){
  TX=true;
}
if (receivedMessage[i]==0){
  tx="";
  comando="";
  valor="";
  modo="";
  break;
}
}
}

```

4.2. Programación del microcontrolador esclavo 1

Este código es el encargado de la lógica del microcontrolador que se encarga del control del suministro de energía de 120 voltios de corriente alterna para los electrodomésticos que se encuentren conectados al microcontrolador, en este código, a diferencia que en el código del microcontrolador maestro, no hay una lista de direcciones o de canales, ya que a este microcontrolador solo le interesa saber su dirección de canal que le corresponde y que comparte con el microcontrolador maestro. El código de este microcontrolador esclavo 1 es el mismo para todos los de su tipo, solo se debe modificar la dirección de canal que corresponda con el asignado por el microcontrolador maestro y este se expresa de la misma forma por un vector de 5 números hexadecimales.

A continuación se muestra el código utilizado para el microcontrolador esclavo 1:

```

#include<SPI.h>
#include<RF24.h>
#include<TimerOne.h>

//DIRECCION DE CANAL
const uint64_t pipes[6] = {0x5365744130LL}; //canal: SetA0 TX2
RF24 radio(9, 10); // ce, csn pines nrf24l01

//PLUGS
const byte RPin1 = A0; //pin A0 relay1
const byte RPin2 = A1; //pin A1 relay2
const byte RPin3 = A2; //pin A2 relay3
const byte RPin4 = A3; //pin A3 relay4
int nplug = 0; //posicion de plug
char plug[8] = {0}; //valores de plug

//DIMMERS
const byte IntPin = 2; //interrupcion
const byte ACPin1 = 3; //pin3
const byte ACPin2 = 4; //pin4
const int TPeriodo = 33; //periodo de timmer
volatile int i = 0; //cuenta de periodos dimmer1
volatile int j = 0; //cuenta de periodos dimmer2
int dim1 = 255; //intensidad dimmer1
int dim2 = 255; //intensidad dimmer2
int DIM1 = 255; //intensidad dimmer1
int DIM2 = 255; //intensidad dimmer2
volatile boolean meta1 = false; //bandera dimmer1
volatile boolean meta2 = false; //bandera dimmer2

//RECEPCION DE MENSAJE Y ENVIO
char receivedMessage[32] = {0}; //variable de recepcion de mensaje
const char text[] = "aTX0bOK2cdz"; //ACK TX2 PORT A

//ANALISIS DE MENSAJE
char caracter; //variable receptora de nrf24l01
//variables de valores recibidos
String tx="";
String comando="";
String valor="";
String modo="";
////////////////////
String txx="";
String com="";
String val="";
String mod="";
//valores de condicion o banderas
boolean TX = false;
boolean COMANDO = false;
boolean VALOR = false;
boolean MODO = false;

void setup(void){
  Serial.begin(9600);
  pinMode(ACPin1, OUTPUT);
  pinMode(ACPin2, OUTPUT);
  pinMode(RPin1, OUTPUT);
  pinMode(RPin2, OUTPUT);
  pinMode(RPin3, OUTPUT);
  pinMode(RPin4, OUTPUT);
  pinMode(IntPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(IntPin), CRUCExCERO, RISING);
  Timer1.initialize(TPeriodo);
  Timer1.attachInterrupt(temporizador);
  radio.begin();
  radio.setPALevel(RF24_PA_MAX);

```

```

radio.setChannel(0x4c);
radio.openReadingPipe(0,pipes[0]);
radio.enableDynamicPayloads();
radio.powerUp();
radio.startListening();
}

void loop(void){
if(radio.available()){
radio.read(receivedMessage, sizeof(receivedMessage));
Serial.println(receivedMessage);
AnalizarComando();
envio(text, 0, sizeof(text));
logica();
}
funciones();
delay(10);
}
//INTERRUPCION DE CRUCE POR CERO
void CRUCExCERO(){
meta1 = true;
meta2 = true;
i = 0;
j = 0;
digitalWrite(ACPin1, LOW);
digitalWrite(ACPin2, LOW);
}
//INTERRUPCION DE TEMPORIZACION
void temporizador(){
if(meta1){
if(i >= DIM1){
digitalWrite(ACPin1, HIGH);
i = 0;
meta1 = false;
}else{
i++;
}
}
if(meta2){
if(j >= DIM2){
digitalWrite(ACPin2, HIGH);
j = 0;
meta2 = false;
}else{
j++;
}
}
}
}
//ENVIO
void envio(char mensaje[], int canal, int tam){
radio.stopListening();
radio.openWritingPipe(pipes[canal]);
radio.write(mensaje, tam);
radio.startListening();
}
//LOGICA
void logica(){
//RELAYS DE TOMA CORRIENTE
if(com=="PLUG"){
if(plug[0]=='1'){
digitalWrite(RPin1, HIGH);
}else{
digitalWrite(RPin1, LOW);
}
if(plug[1]=='1'){
digitalWrite(RPin2, HIGH);
}
}
}

```

```

}else{
  digitalWrite(RPin2, LOW);
}
if(plug[2]=='1'){
  digitalWrite(RPin3, HIGH);
}else{
  digitalWrite(RPin3, LOW);
}
if(plug[3]=='1'){
  digitalWrite(RPin4, HIGH);
}else{
  digitalWrite(RPin4, LOW);
}
com="";
val="";
}
//DIM
if(com=="DIM1"){
  dim1 = 255 - 2.55*val.toInt();
  com="";
  val="";
}
if(com=="DIM2"){
  dim2 = 255 - 2.55*val.toInt();
  com="";
  val="";
}
}
//FUNCIONES QUE CORREN SIEMPRE
void funciones(){
  //GRADUACION DE INTENSIDAD DIM1
  if (DIM1 < dim1){
    DIM1 = DIM1 + 1;
    delay(2);
  }else if (DIM1 > dim1){
    DIM1 = DIM1 - 1;
    delay(2);
  }
  //GRADUACION DE INTENSIDAD DIM2
  if (DIM2 < dim2){
    DIM2 = DIM2 + 1;
    delay(2);
  }else if (DIM2 > dim2){
    DIM2 = DIM2 - 1;
    delay(2);
  }
}
//ANALISIS
void AnalizarComando() {
  for (int i=0; i < sizeof(receivedMessage) ; i++){

    if (comando == "PLUG" && VALOR){
      plug[nplug] = receivedMessage[i];
      nplug++;
    }else{
      nplug = 0;
    }
    caracter = receivedMessage[i];
    if (caracter=='z'){
      VALOR=false;
      MODO=false;
      txx=tx;
      com=comando;
      val=valor;
      mod=modo;
    }
  }
}

```

```

if (MODO){
  modo += caracter;
}
if (caracter=='d'){
  VALOR=false;
  MODO= true;
}
if (VALOR){
  valor += caracter;
}
if (caracter=='c'){
  COMANDO=false;
  VALOR=true;
}
if (COMANDO){
  comando += caracter;
}
if(caracter == 'b'){
  TX=false;
  COMANDO=true;
}
if (TX){
  tx += caracter;
}
if(caracter=='a'){
  TX=true;
}
if (receivedMessage[i]==0){
  tx="";
  comando="";
  valor="";
  modo="";
  break;
}
}
}
}

```

4.3. Programación del microcontrolador esclavo 2

La programación de este microcontrolador se encarga de la *logia* del manejo de los interruptores de la iluminación del área donde se encuentre y permite al usuario el manejo por medio de botones físicos o a través de órdenes por medio de entorno web. El código de este microcontrolador esclavo 2 es el mismo para todos los microcontroladores de este tipo, solo se debe modificar la dirección del canal que esté utilizando el microcontrolador maestro y que posea el mismo vector de 5 números hexadecimales en su lista.

A continuación se muestra el código utilizado para el microcontrolador esclavo 2:

```

#include<SPI.h>
#include<RF24.h>
#include<TimerOne.h>

//DIRECCION DE CANAL
const uint64_t pipes[6] = {0x5365744131LL}; //canal: SetA1 (TX3)
RF24 radio(9, 10); // ce, csn pines nrf24l01

//DIMMERS
const byte BOTON1 = 5; //boton1
const byte BOTON2 = 6; //boton2
const byte IntPin = 2; //interrupcion
const byte ACPin1 = A0; //pinA0
const byte ACPin2 = A1; //pinA1
const int TPeriodo = 33; //periodo de timmer
volatile int i = 0; //cuenta de periodos dimmer1
volatile int j = 0; //cuenta de periodos dimmer2
int dim1 = 255; //intensidad dimmer1
int dim2 = 255; //intensidad dimmer2
int DIM1 = 255; //intensidad dimmer1
int DIM2 = 255; //intensidad dimmer2
boolean bandera1 = true; //bandera de ajuste de dimmer1
boolean bandera2 = true; //bandera de ajuste de dimmer2
volatile boolean meta1 = false; //bandera dimmer1
volatile boolean meta2 = false; //bandera dimmer2

//RECEPCION DE MENSAJE Y ENVIO
char receivedMessage[32] = {0}; //variable de recepcion de mensaje
const char text[] = "aTX0bOK3cdz"; //ACK TX3 PORT A
const char text1[] = "aTX0bSW1OFFcdz";
const char text2[] = "aTX0bSW1ONcdz";
const char text3[] = "aTX0bSW2OFFcdz";
const char text4[] = "aTX0bSW2ONcdz";

//ANALISIS DE MENSAJE
char caracter; //variable receptora de nrf24l01
//variables de valores recibidos
String tx="";
String comando="";
String valor="";
String modo="";
//////////
String txx="";
String com="";
String val="";
String mod="";
//valores de condicion o banderas
boolean TX = false;
boolean COMANDO = false;
boolean VALOR = false;
boolean MODO = false;

void setup(void){
  pinMode(ACPin1, OUTPUT);
  pinMode(ACPin2, OUTPUT);
  pinMode(IntPin, INPUT_PULLUP);
  pinMode(BOTON1, INPUT_PULLUP);
  pinMode(BOTON2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(IntPin), CRUCEXCERO, RISING);
  Timer1.initialize(TPeriodo);
  Timer1.attachInterrupt(temporizador);
  radio.begin();
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(0x4c);
  radio.openReadingPipe(0,pipes[0]);
  radio.enableDynamicPayloads();

```

```

radio.powerUp();
radio.startListening();
}

void loop(void){
  if(radio.available()){
    radio.read(receivedMessage, sizeof(receivedMessage));
    Serial.println(receivedMessage);
    AnalizarComando();
    envio(text, 0, sizeof(text));
    logica();
  }
  funciones();
  delay(10);
}
//INTERRUPCION DE CRUCE POR CERO
void CRUCExCERO(){
  meta1 = true;
  meta2 = true;
  i = 0;
  j = 0;
  digitalWrite(ACPin1, LOW);
  digitalWrite(ACPin2, LOW);
}
//INTERRUPCION DE TEMPORIZACION
void temporizador(){
  if(meta1){
    if(i >= DIM1){
      digitalWrite(ACPin1, HIGH);
      i = 0;
      meta1 = false;
    }else{
      i++;
    }
  }
  if(meta2){
    if(j >= DIM2){
      digitalWrite(ACPin2, HIGH);
      j = 0;
      meta2 = false;
    }else{
      j++;
    }
  }
}
//ENVIO
void envio(char mensaje[], int canal, int tam){
  radio.stopListening();
  radio.openWritingPipe(pipes[canal]);
  radio.write(mensaje, tam);
  radio.startListening();
}
//LOGICA
void logica(){
  //DIM
  if(com=="DIM1"){
    dim1 = 255 - 2.55*val.toInt();
    com="";
    val="";
  }
  if(com=="DIM2"){
    dim2 = 255 - 2.55*val.toInt();
    com="";
    val="";
  }
}
}

```

```

//FUNCIONES QUE CORREN SIEMPRE
void funciones(){
//BOTON1
if (!digitalRead(BOTON1) && bandera1){
  if (dim1 < 255 ){
    dim1 = 255;
    envio(text1, 0, sizeof(text1));
  }else if (dim1 == 255){
    dim1 = 0;
    envio(text2, 0, sizeof(text2));
  }
  bandera1 = false;
} else if (digitalRead(BOTON1) && !bandera1){
  bandera1 = true;
  delay(10);
}
//BOTON2
if (!digitalRead(BOTON2) && bandera2){
  if (dim2 < 255 ){
    dim2 = 255;
    envio(text3, 0, sizeof(text3));
  }else if (dim2 == 255){
    dim2 = 0;
    envio(text4, 0, sizeof(text4));
  }
  bandera2 = false;
} else if (digitalRead(BOTON2) && !bandera2){
  bandera2 = true;
  delay(10);
}
//GRADUACION DE INTENSIDAD DIM1
if (DIM1 < dim1){
  DIM1 = DIM1 + 1;
  delay(2);
}else if (DIM1 > dim1){
  DIM1 = DIM1 - 1;
  delay(2);
}
//GRADUACION DE INTENSIDAD DIM2
if (DIM2 < dim2){
  DIM2 = DIM2 + 1;
  delay(2);
}else if (DIM2 > dim2){
  DIM2 = DIM2 - 1;
  delay(2);
}
}
//ANALISIS
void AnalizarComando() {
for (int i=0; i < sizeof(receivedMessage) ; i++){
  caracter = receivedMessage[i];
  if (caracter=='z'){
    VALOR=false;
    MODO=false;
    txx=tx;
    com=comando;
    val=valor;
    mod=modo;
  }
  if (MODO){
    modo += caracter;
  }
  if (caracter=='d'){
    VALOR=false;
    MODO= true;
  }
}
}

```

```

if (VALOR){
    valor += caracter;
}
if (caracter=='c'){
    COMANDO=false;
    VALOR=true;
}
if (COMANDO){
    comando += caracter;
}
if(caracter == 'b'){
    TX=false;
    COMANDO=true;
}
if (TX){
    tx += caracter;
}
if(caracter=='a'){
    TX=true;
}
if (receivedMessage[i]==0){
    tx="";
    comando="";
    valor="";
    modo="";
    break;
}
}
}

```

4.4. Programación del mando central

La programación del mando central se divide en dos lenguajes diferentes de programación: Python y PHP, los cuales, trabajando en conjunto se encargan de la ejecución y gestión de las funciones correspondientes entre el usuario y el sistema. La programación en Python se encarga de la comunicación entre PHP y los diferentes microcontroladores de tipo maestro que estén conectados, y la programación en PHP se encarga de la comunicación entre el usuario y la programación que ejecuta Python. La comunicación entre el usuario y la programación en PHP es posible con el uso de lenguaje HTML que interactúa con el usuario y programación en JavaScript que interactúa con PHP. El modo de interactuar es por navegador web.

El sistema cuenta con un conjunto de páginas web que hace uso de las herramientas anteriormente mencionadas, cada página cuenta con su

documento de estilos escrito en lenguaje CSS, el cual se encarga de mejorar la apariencia de la página escrita en HTML.

4.4.1. Programación en Python

El código de programación en Python se encarga de la recepción de comandos por medio de un *socket* entre Python y PHP, luego de la recepción por medio del *socket* se hace la selección de la dirección correcta del microcontrolador maestro al que va destinado el comando.

Cuando en Python se recibe un comando de parte de uno de los microcontroladores, este procesa la información y, según sea el comando, almacena una respuesta o un comando en una base de datos para que así pueda ser verificado en PHP y permita ser visualizada la respuesta por el usuario por medio de la página web.

A continuación se muestra el código escrito en Python para el servidor:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from lib_nrf24 import NRF24
import Queue as queue
import threading
import time
import socket
import MySQLdb

b = 1
def comusuario():
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_address = ('localhost', 10000)
    print 'Starting up on %s port %s' % server_address
    sock.bind(server_address)
    sock.listen(5)
    while (b):
        print 'Waiting for a connection'
        connection, client_address = sock.accept()
        try:
            print 'Connection from: ', client_address
            try:
                all_data = ""
                while (b):
                    data = "
```

```

try:
    data = connection.recv(16, socket.MSG_DONTWAIT)
except:
    pass
if data:
    all_data += data
else:
    break
print 'Received "%s"' % all_data
    if (all_data != " " and all_data != " " and all_data != " "):
        usuario.put(all_data)
except:
    break
finally:
    try:
        connection.close()
    except:
        pass

def comsistema():
    area = ['sala', 'cocina', 'hab1', 'hab2', 'hab3', 'patio']
    while (b):
        if not sistema.empty():
            item2 = sistema.get()
            if (item2[5:7] == "IR"):
                narea = int(item2[0]) - 1
                comando = str(narea) + "aTX" + item2[0] + item2[7:]
                db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
                cursor = db.cursor()
                query = "UPDATE control SET funx='%s' WHERE id = 1" % (comando)
                cursor.execute(query)
                db.commit()
                cursor.close()
                db.close()
            if (item2[6:8] == "SW"):
                narea = int(item2[0]) - 1
                db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
                cursor = db.cursor()
            if (item2[9:11] == "ON"):
                query = "UPDATE luces SET estado='%s' WHERE id = '%s'" % (item2[9:11], item2[8:9])
            else:
                query = "UPDATE luces SET estado='%s' WHERE id = '%s'" % (item2[9:12], item2[8:9])
                cursor.execute(query)
                db.commit()
                cursor.close()
                db.close()
            if (item2[6:10] == "TEMP"):
                narea = int(item2[0]) - 1
                db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
                cursor = db.cursor()
                query = "UPDATE sensores SET temperatura='%s', humedad='%s', icalor='%s' WHERE id = '1'" %
                (item2[11:16], item2[17:22], item2[23:28])
                cursor.execute(query)
                db.commit()
                cursor.close()
                db.close()
            if (item2[6:9] == "GAS"):
                narea = int(item2[0]) - 1
                db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
                cursor = db.cursor()
                query = "UPDATE sensores SET co2='%s' WHERE id = '1'" % (item2[10:12])
                cursor.execute(query)
                db.commit()
                cursor.close()
                db.close()
            if (item2[6:12] == "PUERTA"):

```

```

narea = int(item2[0]) - 1
db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
cursor = db.cursor()
query = "UPDATE sensores SET puertas=%s' WHERE id = '1'" % (item2[13:20])
cursor.execute(query)
db.commit()
cursor.close()
db.close()
if (item2[6:10] == "PRES"):
narea = int(item2[0]) - 1
db = MySQLdb.connect(host="localhost", user="root", passwd="orangepi", db=area[narea])
cursor = db.cursor()
query = "UPDATE sensores SET estado=%s' WHERE id = '1'" % (item2[11:13])
cursor.execute(query)
db.commit()
cursor.close()
db.close()
print "comando de entrada: ", item2
time.sleep(1)

usuario = queue.Queue()
sistema = queue.Queue()
t = threading.Thread(target=comusuario)
u = threading.Thread(target=comsistema)
t.start()
u.start()

pipes = [[0x50, 0x4f, 0x52, 0x54, 0x41], [0x50, 0x4f, 0x52, 0x54, 0x42], [0x50, 0x4f, 0x52, 0x54, 0x43], [0x50, 0x4f, 0x52,
0x54, 0x44], [0x50, 0x4f, 0x52, 0x54, 0x45], [0x50, 0x4f, 0x52, 0x54, 0x46]]

radio = NRF24()
radio.begin(0, 2)

radio.setPayloadSize(32)
radio.setChannel(0x4c)
radio.setDataRate(NRF24.BR_1MBPS)
radio.setPALevel(NRF24.PA_MAX)

radio.setAutoAck(True)
radio.enableDynamicPayloads()
radio.enableAckPayload()

radio.openWritingPipe(pipes[0])
radio.openReadingPipe(0, pipes[0])
radio.openReadingPipe(1, pipes[1])
radio.openReadingPipe(2, pipes[2])
radio.openReadingPipe(3, pipes[3])
radio.openReadingPipe(4, pipes[4])
radio.openReadingPipe(5, pipes[5])

radio.printDetails()
a = 1
try:
while(a):
# INICIA ENVIO
if not usuario.empty():
item = usuario.get()
print "comando de salida: ", item
radio.openWritingPipe(pipes[int(item[0:1])])
mensaje = list(item[1:])
while len(mensaje) < 32:
mensaje.append(0)
radio.stopListening()
radio.write(mensaje)
start = time.time()
radio.startListening()

```

```

while not radio.available():
    time.sleep(1 / 100)
    if time.time() - start > 2:
        print("Timed out.")
        break
# FIN ENVIO

# INICIA RECEPCION
for i in range(5):
    pipe = []
    pipe = [i]
    if radio.available(pipe):
        receivedMessage = []
        radio.read(receivedMessage, radio.getDynamicPayloadSize())
        string = ""
        m = 0
        m = i + 1
        string += str(m)
        for n in receivedMessage:
            if (n >= 32 and n <= 126):
                string += chr(n)
            sistema.put(string)
        time.sleep(0.4)
except:
    print "fin del programa"
    a = 0
    b = 0
    t.join()
    u.join()

```

4.4.2. Programación en PHP y códigos de páginas

Las páginas con las que interactúa un usuario son un compendio de varios lenguajes y códigos de programación, en este caso las páginas están conformadas por lenguaje HTML, CSS, JavaScript y PHP. Por lo tanto, se mostrarán los 4 códigos por cada página que se presenta.

4.4.2.1. Página de acceso

La página de registro tiene como función brindar seguridad a la aplicación del sistema por web, ya que a través de ella se concede el acceso a la página de la aplicación, para ello es necesario tener un usuario y una contraseña que estén almacenados en la base de datos de usuarios del sistema.

La apariencia de la página es simple pues solo cuenta con las instrucciones y el formulario para poder ser llenado y enviado al servidor al presionar el botón de entrar. La página se conforma por 4 archivos PHP que están dedicados a la seguridad de la aplicación y solamente en el archivo principal se utiliza lenguaje HTML y CSS. A continuación se muestra la apariencia de la página, así como los códigos que la conforman:

Figura 32. **Página de acceso**



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.1.1. **Código PHP, HTML y CSS**

El primer código tiene como función desplegar un formulario de acceso para permitir el ingreso a un usuario.

```

<html>
<head>
<title>Autenticacion </title>
<style type="text/css">
body {
    color: white;
    background-color: #E8ECED;
}
div#principal{
    margin:auto;
    margin-top:30px;
    width:960px;
    height:560px;
    background-color:navy;
    text-align:center;
    border-radius:15px 15px 15px 15px;
}
div#secundario{
    width:100%;
    background-color:black;
    text-align:center;
    border-radius:15px 15px 0px 0px;
}
div#tercero{
    width:100%;
    height:150px
}
div#login{
    width:40%;
    margin:auto;
    background:grey;
    border-radius:15px 15px 15px 15px;
    border-color:black;
}
@media screen and (max-width: 400px) {
    div#principal {
        width:50%;
        height:400px;
        text-align:center;
    }
    div#secundario {
        width:50%;
        height:200px;
        text-align:center;
    }
    div#tercero{
        }
}
</style>
</head>
<body>
<div id="principal">
<div id="secundario">
<h1 align="center" color="black">SISTEMA DE DOMOTICA</h1>
</div>
<div id ="tercero">
</div>
<form action="controll.php" method="POST">
<div id="login">
<table align="center" width="225" cellspacing="2" cellpadding="2" border="0">
<tr>
<td colspan="2" align="center"
    <?php if ($_GET["errorusuario"]=="si"){ ?>
        bgcolor=red><span style="color:ffffff"><b>Datos incorrectos</b></span>
    <?php }else{ ?>

```

```

                bgcolor=#cccccc>Introduce tu clave de acceso
        <?php } ?>
</td>
</tr>
<tr>
        <td align="right">USER:</td>
        <td><input type="Text" name="usuario" size="8" maxlength="50" style="font-weight: bold; color: white;
background-color: navy"></td>
</tr>
<tr>
        <td align="right">PASSWD:</td>
        <td><input type="password" name="contrasena" size="8" maxlength="50" style="font-weight: bold; color:
white; background-color: navy"></td>
</tr>
<tr>
        <td colspan="2" align="center"><input type="Submit" value="ENTRAR" style="font-weight: bold; color: white;
background-color: navy"></td>
</tr>
</table>
</div>
</form>
</div>
</body>
</html>

```

El siguiente código tiene como función comprobar si los datos del usuario son válidos y crear una sesión:

```

<?php
function Conectarse()
{
    if (!($link=mysql_connect("localhost","root","orangeipi")))
    {
        exit();
    }
    if (!mysql_select_db("usuarios",$link))
    {
        exit();
    }
    return $link;
}
$link=Conectarse();
$result=mysql_query("select * from datos",$link);
$a = 1;
while($row = mysql_fetch_array($result)) {
if ($_POST["usuario"] == $row["user"]){
if ($_POST["contrasena"] == $row["password"]){
    session_start();
    $_SESSION["autenticado"]="SI";
    $a=0;
    header ("Location: aplicacion.php");
}
}
}
mysql_free_result($result);
mysql_close($link);
if ($a == 1){
header("Location: index.php?errorusuario=si");
} ?>

```

El siguiente código verifica si hay una sesión creada al ingresar a la página web de la aplicación:

```
<?php
//Inicio la sesión
session_start();
//COMPRUEBA QUE EL USUARIO ESTA AUTENTIFICADO
if ($_SESSION["autenticado"] != "SI") {
    //si no existe, envio a la página de autentificacion
    header("Location: index.php");
    //ademas salgo de este script
    exit();
}
?>
```

El último código de esta página permite salir de la página de aplicación y cerrar sesión en el navegador:

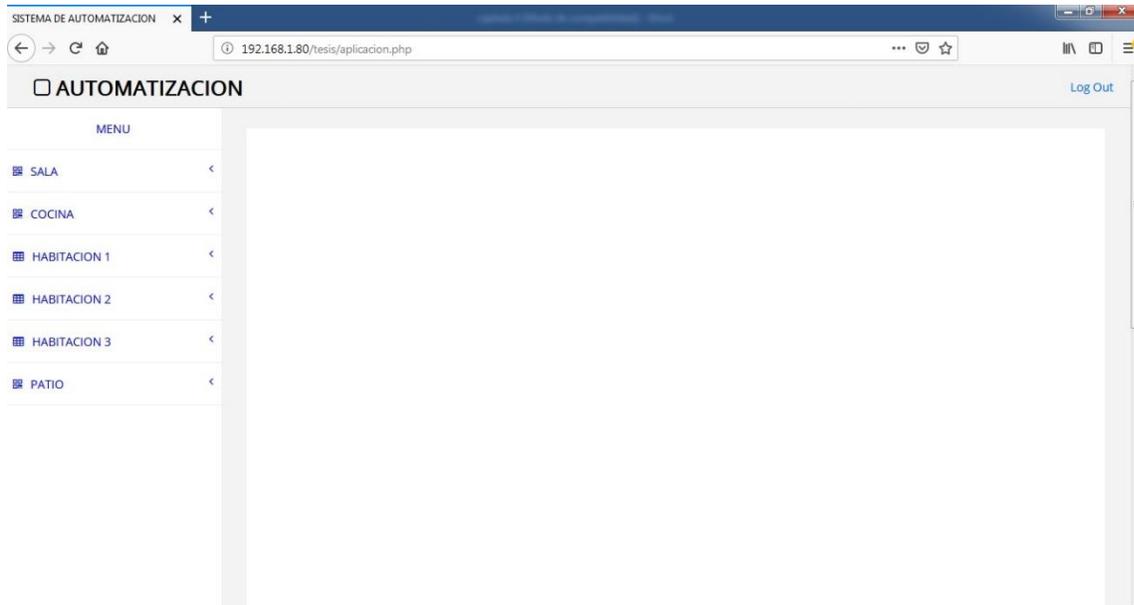
```
<?php
session_start();
session_destroy();
header("Location: index.php?errorusuario=no");
?>
```

4.4.2.2. Página principal de la aplicación

La página principal tiene como base una plantilla del *framework* Bootstrap y es utilizada para el menú principal del sistema. En este caso únicamente se muestra el código combinado de HTML, JavaScript y PHP. El código de PHP verifica con uno de los códigos de seguridad si se ha iniciado la sesión correctamente y el JavaScript es para desplegar las páginas que controlan las funciones del sistema.

La página muestra seis áreas disponibles para controlar, las cuales van desde la sala hasta el patio de la casa. La apariencia de la página se muestra a continuación en la figura 33:

Figura 33. **Página principal**



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.2.1. **Código HTML, PHP y JavaScript**

```
<?php include ("seguridad.php"); ?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>SISTEMA DE AUTOMATIZACION</title>
<link href="assets/css/bootstrap.css" rel="stylesheet" />
<link href="assets/css/font-awesome.css" rel="stylesheet" />
<link href="assets/css/custom.css" rel="stylesheet" />
<link href="assets/fonts/fonts.css" rel="stylesheet" type="text/css" />
<script src="assets/js/jquery.js"></script>
<script type="text/javascript">
var area = "";
var areas = ["sala", "cocina", "hab1", "hab2", "hab3", "patio"];
function control(narea, URL){
area = areas[parseInt(narea)];
$.ajax({
    type: "POST",
    url: URL,
    success: function(response) {
```

```

        $('#div-results').html(response);
    }
});
}
</script>
</head>
<body>
<div id="wrapper">
<div class="navbar navbar-inverse navbar-fixed-top">
<div class="adjust-nav">
<div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".sidebar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" ><i class="fa fa-square-o "></i>&nbsp;AUTOMATIZACION</a>
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav navbar-right">
        <li><a href="salir.php">Log Out</a></li>
    </ul>
</div>
</div>
</div>
</div>
<nav class="navbar-default navbar-side" role="navigation">
<div class="sidebar-collapse">
<ul class="nav" id="main-menu">
<li class="text-center user-image-back">
    <a> MENU </a>
</li>
<li>
<a><i class="fa fa-qrcode "></i>SALA<span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
    <li>
        <a onclick = "control('0', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
    </li>
    <li>
        <a onclick = "control('0', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
    </li>
    <li>
        <a onclick = "control('0', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
    </li>
    <li>
        <a onclick = "control('0', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
    </li>
</ul>
</li>
<li>
<a><i class="fa fa-qrcode "></i>COCINA<span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
    <li>
        <a onclick = "control('1', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
    </li>
    <li>
        <a onclick = "control('1', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
    </li>
    <li>
        <a onclick = "control('1', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
    </li>
</ul>
</li>

```

```

        </li>
        <li>
            <a onclick = "control('1', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
        </li>
    </ul>
</li>
<li>
<a><i class="fa fa-table "></i>HABITACION 1<span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
    <li>
        <a onclick = "control('2', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
    </li>
    <li>
        <a onclick = "control('2', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
    </li>
    <li>
        <a onclick = "control('2', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
    </li>
    <li>
        <a onclick = "control('2', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
    </li>
</ul>
</li>
<li>
<a><i class="fa fa-table "></i>HABITACION 2<span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
    <li>
        <a onclick = "control('3', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
    </li>
    <li>
        <a onclick = "control('3', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
    </li>
    <li>
        <a onclick = "control('3', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
    </li>
    <li>
        <a onclick = "control('3', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
    </li>
</ul>
</li>
<li>
<a><i class="fa fa-table "></i>HABITACION 3<span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
    <li>
        <a onclick = "control('4', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
    </li>
    <li>
        <a onclick = "control('4', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
    </li>
    <li>
        <a onclick = "control('4', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
    </li>
    <li>

```

```

                <a onclick = "control('4', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
            </li>
        </ul>
    </li>
    <li>
        <a <i class="fa fa-qrcode "></i>PATIO<span class="fa arrow"></span></a>
        <ul class="nav nav-second-level">
            <li>
                <a onclick = "control('5', 'luces')" class="btn btn-success"><i class="fa fa-desktop
"></i>ILUMINACION</a>
            </li>
            <li>
                <a onclick = "control('5', 'enchufes')" class="btn btn-success"><i class="fa fa-desktop
"></i>ENERGIA</a>
            </li>
            <li>
                <a onclick = "control('5', 'control')" class="btn btn-success"><i class="fa fa-desktop
"></i>CONTROL</a>
            </li>
            <li>
                <a onclick = "control('5', 'sensores')" class="btn btn-success"><i class="fa fa-desktop
"></i>SENSORES</a>
            </li>
        </ul>
    </li>
</ul>
</li>
</ul>
</div>
</nav>
<div id="page-wrapper">
<div id="page-inner">
<div id="div-results">
<script type="text/javascript">

</script>
</div>
</div>
</div>
</div>
</div>
<script src="assets/js/jquery-1.10.2.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script src="assets/js/jquery.metisMenu.js"></script>
<script src="assets/js/custom.js"></script>
</body>
</html>

```

4.4.2.3. Página de control de iluminación

El código de la página de iluminación es general para todas las áreas, ya que en la selección del menú principal se le indica al código de iluminación qué área se desea controlar. Esta página permite el manejo de la iluminación principal del área seleccionada, con un máximo de dos luces con opción a ser utilizadas con variación de intensidad por medio del menú.

La página también permite el manejo de la lámpara led de emergencia que posee el microcontrolador maestro, las opciones que pueden ser seleccionadas sobre esta lámpara son el encendido y apagado con opción a variar la intensidad. También se da como opción usar la iluminación de la lámpara cuando se detecte presencia de una persona por medio del sensor PIR. La apariencia de la página se muestra a continuación:

Figura 34. **Página de iluminación**



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.3.1. **Código HTML y JavaScript**

Este código es el encargado de ejecutar el contenido HTML para el navegador y la parte de JavaScript que se encarga de la programación de los botones de la página, los deslizadores para la variación de la iluminación y la conexión con el servidor para transferencia de datos.

```

<!doctype html>
<html lang="es">
<head>
<title></title>
<meta http-equiv="content-Type" content="text/html; charset=iso-8859-1">
<link href="luces/index_style_luces.css" rel="stylesheet" type="text/css" />
<script src="assets/js/jquery.js"></script>
<script type="text/javascript">
var area;
var intervalo;
$(document).ready(function(){
  clearInterval(intervalo);
  actualizar();
  intervalo = setInterval("actualizar()", 5000);
});
function actualizar(){
  var parametros1 = {area};
  $.ajax({
    data: parametros1,
    url: 'luces/actualizar.php',
    type: 'post',
    success: function (response) {
      var myArray = jQuery.parseJSON(response);
      var i;
      var j = 0;
      for (i = 0; i < 4; i++) {
        j = i + 1;
        var id = "#" + j.toString();
        var id1 = "#valor" + j.toString();
        var id2 = "#porcentaje" + j.toString();
        if (j < 4){
          $(id1).val(myArray.valor[j]);
          $(id2).html(myArray.valor[j]);
        }
        if (myArray.estado[j] == "ON"){
          $(id).css('background-color', 'red');
          if (j == 4){
            $('#valor3').prop( "disabled", true );
            bandera4 = 0;
          }
        }
        }else{
          $(id).css('background-color', 'blue');
          if (j == 4){
            $('#valor3').prop( "disabled", false );
            bandera4 = 1;
          }
        }
      }
    };
  }
});
$('#valor1').change(function() {
  slider = "1";
  proceso('1', $(this).val());
  $('#porcentaje1').html($(this).val());
});
$('#valor2').change(function() {
  slider = "1";
  proceso('2', $(this).val());
  $('#porcentaje2').html($(this).val());
});
$('#valor3').change(function() {
  slider = "1";
  proceso('3', $(this).val());
  $('#porcentaje3').html($(this).val());
});

```

```

var slider = "0";
var bandera4 = 1;
var tabla = "luces";

function proceso(funcion, val){
  if (funcion == 3 & bandera4 == 0){
    proceso.break();
  }
  var id = "#" + funcion.toString();
  var parametros = {funcion, val, area, slider, tabla};
  $.ajax({
    data: parametros,
    url: 'luces/ajax_proceso.php',
    type: 'post',
    success: function (response) {
      if (response == "ON"){
        $(id).css('background-color', 'red');
        if (funcion == 4){
          $('#3').css('background-color', 'blue');
          $('#valor3').prop( "disabled", true );
          bandera4 = 0;
        }
      }else{
        $(id).css('background-color', 'blue');
        if (funcion == 4){
          $('#valor3').prop( "disabled", false );
          bandera4 = 1;
        }
      }
      slider = "0";
    }
  });
}
</script>
</head>
<body>
<div id="general">
<div>
</img>
</div>
<div id="menu">
<label id="titulo" class="texto">ILUMINACION</label>
</div>
<div id="control">
<div id="contenedor">
<div class="control1">
<div>
<label>LUZ 1</label>
</div>
<a id = "1" onclick = "proceso('1','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
<div class="slider">
<label>intensidad:</label>
<input type="range" min="0" max="100" id="valor1" name="valoracion" type="range" step=1
value="0"></input>
<label id="porcentaje1" ></label>&#37;
</div>
</div>
<div class="control1">
<div>
<label>LUZ 2</label>
</div>
<a id = "2" onclick = "proceso('2','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
<div class="slider">
<label>intensidad:</label>

```

```

        <input type="range" min="0" max="100" id="valor2" name="valoracion" type="range" step=1
value="0"></input>
        <label id="porcentaje2" ></label>&#37;
    </div>
</div>
<div class="control1">
    <div>
        <label>LUZ LED</label>
    </div>
    <a id = "3" onclick = "proceso('3','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
    <a id = "4" onclick = "proceso('4','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">PIR</a>
    <div class="slider">
        <label>intensidad:</label>
        <input type="range" min="0" max="100" id="valor3" name="valoracion" type="range" step=1
value="0"></input>
        <label id="porcentaje3" ></label>&#37;
    </div>
</div>
</div>
</body>
</html>

```

4.4.2.3.2. Código PHP

Esta página contiene dos códigos PHP, el primero se encarga de la actualización de los datos mostrados en la página, la llamada a este código la hace JavaScript cada cierto período de tiempo establecido en la página. A continuación se muestra el código de la actualización:

```

<?php
$area = $_POST['area'];
$conn = mysql_connect("localhost","root","orangeipi");
mysql_select_db($area,$conn);
$result=mysql_query("SELECT * FROM luces",$conn);
$i = 1;
while($row = mysql_fetch_array($result)) {
    $comando=$row;
    $estado[$i]= $comando[1];
    $valor[$i]= $comando[2];
    $i = $i + 1;
}
mysql_free_result($result);
mysql_close($conn);
$datos = array(
'estado' => $estado,
'valor' => $valor
);
echo json_encode($datos);
?>

```

El segundo código se encarga de la gestión de las opciones que seleccionó el usuario, las procesa y las ordena según el área que corresponde a la opción, luego almacena en una base de datos los valores o estados establecidos y arma un comando que envía por medio de un *socket* al programa que corre en Python.

```
<?php
$funcion = $_POST['funcion'];
$area = $_POST['area'];
$valor = $_POST['val'];
$slider = $_POST['slider'];
$tabla = $_POST['tabla'];

if ($area == "sala"){ $narea = "0aTX3bDIM"; }
if ($area == "cocina"){ $narea = "1aTX3bDIM"; }
if ($area == "hab1"){ $narea = "2aTX3bDIM"; }
if ($area == "hab2"){ $narea = "3aTX3bDIM"; }
if ($area == "hab3"){ $narea = "4aTX3bDIM"; }
if ($area == "patio"){ $narea = "5aTX3bDIM"; }

if ($funcion == 4){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT estado, id FROM $tabla WHERE id='$funcion'", $conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    if ($comando[0] == "OFF"){
        $comando[0] = "ON";
    }else{
        $comando[0] = "OFF";
    }
    $com = substr($narea, 0, 6) . "LEMERGc" . $comando[0] . ".dz";
    $ssql = "UPDATE $tabla SET estado='{$comando[0]}' WHERE id = '$funcion'";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
    $ssql = "UPDATE $tabla SET estado='OFF' WHERE id = '3'";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
}
else{
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT estado, valor FROM $tabla WHERE id='$funcion'", $conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    if ($slider == "0"){
        if ($comando[0] == "OFF"){
            $comando[0] = "ON";
            if ($funcion <= 2){
                $com = $narea . $funcion . "c" . $comando[1] . ".dz";
            }else{
                $com = $narea . "c" . $comando[1] . ".dz";
            }
        }
        }else{
            $comando[0] = "OFF";
            if ($funcion <= 2){
                $com = $narea . $funcion . "c" . "0" . ".dz";
            }
        }
    }
}

```

```

    }else{
        $com = $narea . "c" . "0" . "dz";
    }
}
}
$ssql = "UPDATE $tabla SET estado='{$comando[0]}' WHERE id = '$funcion'";
$rs = mysql_query($ssql,$conn);
mysql_free_result($rs);
}else if($slider == "1"){
    $ssql = "UPDATE $tabla SET valor='{$valor}' WHERE id = '$funcion'";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
    if ($comando[0] == "ON"){
        if ($funcion <= 2){
            $com = $narea . $funcion . "c" . $valor . "dz";
        }else{
            $com = $narea . "c" . $valor . "dz";
        }
    }else{
        $com = " ";
    }
}
}
}
mysql_free_result($result);
mysql_close($conn);

if ($com != " "){
    error_reporting(E_ALL);
    $address = gethostbyname('localhost');
    $service_port = 10000;
    $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
    $result = socket_connect($socket, $address, $service_port);
    socket_write($socket, $com, strlen($com));
    socket_close($socket);
    echo $comando[0];
}
?>

```

4.4.2.3.3. Código CSS

```

*{
    margin:0px;
    padding:0px;
}

.button {
    font:bold 20px Verdana;
    color:white;
    margin:10px 0px;
    margin-left:8px;
    margin-right:8px;
    margin-top:6px;
    margin-bottom:6px;
    border:4px solid #fff;
    padding:10px 13px 10px 13px;
    display:inline-block;
    box-sizing:border-box;
    border-radius:15px;
    text-align:center;
    width:200;
    text-decoration:none;
    transition:box-shadow .5s,padding-top .5s,background .5s;
}

```

```

}

.texto{
  font:bold 25px Verdana;
  color:white;
}

.button.blue {
  background:#3498db;
  box-shadow:0px 3px 0px #266792;
}

.button.grey {
  background:#7f8c8d;
  box-shadow:0px 3px 0px #373c3c;
}

.button.medium {
  width:350px;
}

.button.large {
  width:450px;
}

.button.radius {
  border-radius:50px;
  font:bold 20px Verdana;
}

.button.hover {
  box-shadow:0px 0px 0px;
  padding-top:7px;
  background:#3498db;
}

div#general{
  margin:auto;
  margin-top:30px;
  width:960px;
  height:560px;
  background-color:navy;
  text-align:center;
}

img.imagen{
  width: 960px;
  height: 220px;
}

div#menu{
  margin:auto;
  width:960px;
  height:50px;
  line-height: 50px;
  background-color:grey;
  text-align:center;
}

label#titulo{
  display: inline-block;
  vertical-align: middle;
  line-height: normal;
}

div#contenedor{

```

```

margin:auto;
margin-top:15px;
width:100%;
height:190px;
background-color:grey;
border-radius:15px 15px 15px 15px;
}

div#control{
margin:auto;
margin-top:30px;
width:92%;
height:200px;
background-color:grey;
border-radius:15px 15px 15px 15px;
text-align:center;
}

.control1{
margin:auto;
margin-top:15px;
float:left;
margin-left:1px;
margin-right:1px;
width:33%;
height:160px;
background-color:white;
border:solid;
border-width:3px;
border-color:black;
border-radius:15px 15px 15px 15px;
text-align:center;
}

.slider{
margin-right:auto;
margin-left:auto;
margin-top:10px;
width:90%;
}

@media screen and (max-width: 400px) {
div#general {
width:100%;
height:700px;
text-align:center;
}
div#menu{
width:100%;
height:30px;
line-height: 30px;
text-align:center;
}
label#titulo{
display: inline-block;
vertical-align: middle;
line-height: normal;
}
div#control{
margin:auto;
margin-top:20px;
width:90%;
height:540px;
text-align:center;
}
div#contenedor{

```

```

margin-top:15px;
margin:auto;
float:center;
width:100%;
height:540px;
background-color:grey;
border-radius:15px 15px 15px 15px;
text-align:center;
}
.control1{
margin:auto;
margin-top:15px;
margin-left:14px;
margin-right:15px;
width:90%;
height:160px;
background-color:white;
border:solid;
border-width:3px;
border-color:black;
border-radius:15px 15px 15px 15px;
text-align:center;
}
.texto{
color:#fff;
margin-left:10px;
margin-right:10px;
}
.button{
font:bold 10px Verdana;
width:100;
margin-left:3px;
margin-right:3px;
}
.button.radius{
font:bold 10px Verdana;
width:100;
margin-left:3px;
margin-right:3px;
}
img.imagen{
width:100%;
height: 90px;
}
}

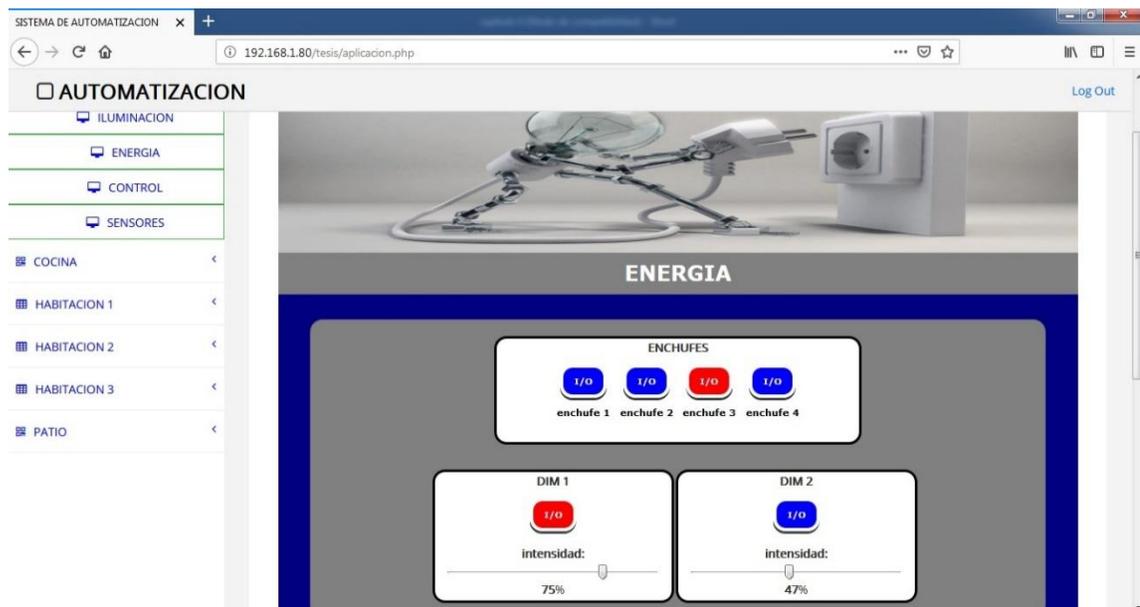
```

4.4.2.4. Página de control de energía

La página del control de la energía está dedicada enteramente al microcontrolador esclavo 1, ya que su función es controlar el encendido o apagado de los 6 tomacorrientes que posee el microcontrolador. La página muestra un control para manejar 4 tomacorrientes con función de encendido y apagado y dos controles para dos tomacorrientes con opción a variación de voltaje para el uso de cargas inductivas como luces o ventiladores.

El código de la página es general para todas las áreas de selección de la página principal, ya que el valor del área se establece al seleccionar una. La apariencia de la página muestra a continuación:

Figura 35. **Página de energía**



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.4.1. **Código HTML y JavaScript**

El código HTML establece el orden de la página para el navegador y trabajando en conjunto con JavaScript se realizan las funciones que el usuario selecciona en la página. El código en JavaScript controla algunas características de la página y actualiza los datos de la misma al conectarse por medio de Ajax con los programas de PHP. El código del programa se muestra a continuación:

```

<!doctype html>

<html lang="es">
<head>
<title></title>
<meta http-equiv="content-Type" content="text/html; charset=iso-8859-1">
<link href="enchufes/index_style_enchufes.css" rel="stylesheet" type="text/css" />
<script src="assets/js/jquery.js"></script>
<script type="text/javascript">
var intervalo;
var area;
$(document).ready(function(){
  clearInterval(intervalo);
  actualizar();
  intervalo = setInterval("actualizar()", 5000);
});
function actualizar(){
  var parametros1 = {area};
  $.ajax({
    data: parametros1,
    url: 'enchufes/actualizar.php',
    type: 'post',
    success: function (response) {
      var myArray = jQuery.parseJSON(response);
      var i;
      var j = 0;
      for (i = 0; i < 6; i++) {
        j = i + 1;
        var id = "#" + j.toString();
        var id1 = "#valor" + j.toString();
        var id2 = "#porcentaje" + j.toString();
        if (j > 4){
          $(id1).val(myArray.valor[j]);
          $(id2).html(myArray.valor[j]);
        }
        if (myArray.estado[j] == "ON" || myArray.estado[j] == "1"){
          $(id).css('background-color', 'red');
        }else{
          $(id).css('background-color', 'blue');
        }
      }
    };
  });
}
$('#valor5').change(function() {
  slider = "1";
  proceso('5', $(this).val());
  $('#porcentaje5').html($(this).val());
});
$('#valor6').change(function() {
  slider = "1";
  proceso('6', $(this).val());
  $('#porcentaje6').html($(this).val());
});
var slider = "0";
var tabla = "enchufes";

function proceso(funcion, val){
  var id = "#" + funcion.toString();
  var parametros = {funcion, val, area, slider, tabla};
  $.ajax({
    data: parametros,
    url: 'enchufes/ajax_proceso.php',
    type: 'post',
    success: function (response) {
      if (response == "ON" || response == "1"){

```

```

        $(id).css('background-color', 'red');
    }else{
        $(id).css('background-color', 'blue');
    }
    slider = "0";
}
});
}
</script>
</head>
<body>
<div id="general">
    <div>
        </img><!--https://viajairan.com/enchufes-en-iran/>
    </div>
    <div id="menu">
        <label id="titulo" class="texto">ENERGIA</label>
    </div>
    <div id="control">
        <div id="contenedor2">
            <div class="control2">
                <div>
                    <label>ENCHUFES</label>
                </div>
                <div>
                    <a id = "1" onclick="proceso('1','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
                    <a id = "2" onclick="proceso('2','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
                    <a id = "3" onclick="proceso('3','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
                    <a id = "4" onclick="proceso('4','101')" class="button grey" style="background:blue; font:bold 10px
Verdana;">I/O</a>
                </div>
                <div>
                    <label class="texto2">enchufe 1</label>
                    <label class="texto2">enchufe 2</label>
                    <label class="texto2">enchufe 3</label>
                    <label class="texto2">enchufe 4</label>
                </div>
            </div>
        </div>
        <div id="contenedor">
            <div class="control1">
                <div>
                    <label>DIM 1</label>
                </div>
                <a id = "5" onclick="proceso('5','101')" class="button grey" style="background:red; font:bold 10px
Verdana;">I/O</a>
                <div class="slider">
                    <label>intensidad:</label>
                    <input type="range" min="0" max="100" id="valor5" name="valoracion" type="range" step=1 value="0"></input>
                    <label id="porcentaje5" ></label>&#37;
                </div>
            </div>
            <div class="control1">
                <div>
                    <label>DIM 2</label>
                </div>
                <a id = "6" onclick="proceso('6','101')" class="button grey" style="background:red; font:bold 10px
Verdana;">I/O</a>
                <div class="slider">
                    <label>intensidad:</label>
                    <input type="range" min="0" max="100" id="valor6" name="valoracion" type="range" step=1 value="0"></input>
                    <label id="porcentaje6" ></label>&#37;
                </div>
            </div>
        </div>
    </div>

```

```
</div>
</div>
</div>
</div>
</body>
</html>
```

4.4.2.4.2. Código PHP

Esta página contiene dos códigos PHP, el primero se encarga de la actualización de los datos mostrados en la página, la llamada a este código la hace JavaScript cada cierto período de tiempo establecido en la página. A continuación se muestra el código de la actualización:

```
<?php
$area = $_POST['area'];
$conn = mysql_connect("localhost","root","orangepi");
mysql_select_db($area,$conn);
$result=mysql_query("SELECT * FROM enchufes",$conn);
$i = 1;
while($row = mysql_fetch_array($result)) {
    $comando=$row;
    $estado[$i]= $comando['1'];
    $valor[$i]= $comando['2'];
    $i = $i + 1;
}
mysql_free_result($result);
mysql_close($conn);
$datos = array(
'estado' => $estado,
'valor' => $valor
);
echo json_encode($datos);
?>
```

El segundo código se encarga de la gestión de las opciones que seleccionó el usuario, las procesa y las ordena según el área que corresponde a la opción, luego almacena en una base de datos los valores o estados establecidos y arma un comando que envía por medio de un *socket* al programa que corre en Python.

```

<?php
$funcion = $_POST['funcion'];
$area = $_POST['area'];
$valor = $_POST['valor'];
$slider = $_POST['slider'];
$tabla = $_POST['tabla'];
$fun = $funcion - 4;

if ($area == "sala"){ $narea = "0aTX2b";}
if ($area == "cocina"){ $narea = "1aTX2b";}
if ($area == "hab1"){ $narea = "2aTX2b";}
if ($area == "hab2"){ $narea = "3aTX2b";}
if ($area == "hab3"){ $narea = "4aTX2b";}
if ($area == "patio"){ $narea = "5aTX2b";}

$conn = mysql_connect("localhost","root","orange");
mysql_select_db($area,$conn);
$result=mysql_query("SELECT * FROM $tabla",$conn);
$i = 1;
while($row = mysql_fetch_array($result)) {
    $comando=$row;
    $estado[$i]= $comando[1];
    $val[$i]= $comando[2];
    $i = $i + 1;
}
if ($slider == "0"){
    if ($estado[$funcion] == "OFF" || $estado[$funcion] == "0"){
        if ($funcion < 5){
            $estado[$funcion] = "1";
            $com = $narea . "PLUG" . "c" . $estado[1].$estado[2].$estado[3].$estado[4] . "dz";
        }
        if ($funcion > 4){
            $estado[$funcion] = "ON";
            $com = $narea . "DIM" . $fun . "c" . $val[$funcion] . "dz";
        }
    }
    }else{
        if ($funcion < 5){
            $estado[$funcion] = "0";
            $com = $narea . "PLUG" . "c" . $estado[1].$estado[2].$estado[3].$estado[4] . "dz";
        }
        if ($funcion > 4){
            $estado[$funcion] = "OFF";
            $com = $narea . "DIM" . $fun . "c" . "0" . "dz";
        }
    }
    $ssql = "UPDATE $tabla SET estado='{$estado[$funcion]}' WHERE id = '$funcion'";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
}
}
else{
    $ssql = "UPDATE $tabla SET valor='{$valor}' WHERE id = '$funcion'";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
    if ($estado[$funcion] == "ON"){
        $com = $narea . "DIM" . $fun . "c" . $valor . "dz";
    }
    }else{
        $com = " ";
    }
}
mysql_free_result($result);
mysql_close($conn);

if ($com != " "){
    error_reporting(E_ALL);
    $address = gethostbyname('localhost');
    $service_port = 10000;
    $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
}

```

```

        $result = socket_connect($socket, $address, $service_port);
        socket_write($socket, $com, strlen($com));
        socket_close($socket);
    }
    echo $estado[$funcion];
?>

```

4.4.2.4.3. Código CSS

```

*{
    margin:0px;
    padding:0px;
}

.button {
    font:bold 20px Verdana;
    color:white;
    margin:10px 0px;
    margin-left:8px;
    margin-right:8px;
    margin-top:6px;
    margin-bottom:6px;
    border:4px solid #fff;
    padding:10px 13px 10px 13px;
    display:inline-block;
    box-sizing:border-box;
    border-radius:15px;
    text-align:center;
    width:200;
    text-decoration:none;
    transition:box-shadow .5s,padding-top .5s,background .5s;
}

.texto{
    font:bold 25px Verdana;
    color:white;
}

.texto2{
    font:bold 11px Verdana;
    color:black;
    margin-left:4px;
    margin-right:4px;
}

.button.blue {
    background:#3498db;
    box-shadow:0px 3px 0px #266792;
}

.button.grey {
    background:#7f8c8d;
    box-shadow:0px 3px 0px #373c3c;
}

.button.medium {
    width:350px;
}

.button.large {
    width:450px;
}

```

```

}

.button.radius {
    border-radius:50px;
    font:bold 20px Verdana;
}

.button:hover {
    box-shadow:0px 0px 0px;
    padding-top:7px;
    background:#3498db;
}

div#general{
    margin:auto;
    margin-top:30px;
    width:960px;
    height:685px;
    background-color:navy;
    text-align:center;
}

img.imagen{
    width: 960px;
    height: 220px;
}

div#menu{
    margin:auto;
    width:960px;
    height:50px;
    line-height: 50px;
    background-color:grey;
    text-align:center;
}

label#titulo{
    display: inline-block;
    vertical-align: middle;
    line-height: normal;
}

div#contenedor{
    margin:auto;
    margin-top:15px;
    width:67%;
    height:190px;
    background-color:grey;
    border-radius:15px 15px 15px 15px;
}

div#contenedor2{
    margin:auto;
    width:50%;
    height:150px;
    background-color:grey;
    border-radius:15px 15px 15px 15px;
}

div#control{
    margin:auto;
    margin-top:30px;
    width:92%;
    height:358px;
    background-color:grey;
    border-radius:15px 15px 15px 15px;
}

```

```

        text-align:center;
    }

    .control1{
        margin:auto;
        margin-top:15px;
        float:left;
        margin-left:1px;
        margin-right:1px;
        width:49%;
        height:160px;
        background-color:white;
        border:solid;
        border-width:3px;
        border-color:black;
        border-radius:15px 15px 15px 15px;
        text-align:center;
    }

    .control2{
        float:left;
        width:100%;
        height:130px;
        margin-top:20px;
        background-color:white;
        border:solid;
        border-width:3px;
        border-color:black;
        border-radius:15px 15px 15px 15px;
        text-align:center;
    }

    .slider{
        margin-right:auto;
        margin-left:auto;
        margin-top:10px;
        width:90%;
    }

    @media screen and (max-width: 400px) {
    div#general {
        width:100%;
        height:670px;
        text-align:center;
    }

    div#menu{
        width:100%;
        height:30px;
        line-height: 30px;
        text-align:center;
    }

    label#titulo{
        display: inline-block;
        vertical-align: middle;
        line-height: normal;
    }

    div#control{
        margin:auto;
        margin-top:20px;
        width:90%;
        height:365px;
        text-align:center;
    }

```

```

div#contenedor{
    margin-top:15px;
    margin:auto;
    float:center;
    width:100%;
    height:365px;
    background-color:grey;
    border-radius:15px 15px 15px 15px;
    text-align:center;
}

div#contenedor2{
    margin-top:15px;
    margin:auto;
    float:center;
    width:100%;
    height:145px;
    background-color:grey;
    border-radius:15px 15px 15px 15px;
    text-align:center;
}

.control1{
    margin:auto;
    margin-top:15px;
    margin-left:14px;
    margin-right:15px;
    width:90%;
    height:160px;
    background-color:white;
    border:solid;
    border-width:3px;
    border-color:black;
    border-radius:15px 15px 15px 15px;
    text-align:center;
}

.control2{
    margin:auto;
    margin-top:15px;
    margin-left:14px;
    margin-right:15px;
    width:90%;
    height:130px;
    background-color:white;
    border:solid;
    border-width:3px;
    border-color:black;
    border-radius:15px 15px 15px 15px;
    text-align:center;
}

.texto{
    color:#fff;
    margin-left:10px;
    margin-right:10px;
}

.button{
    font:bold 10px Verdana;
    width:100;
    margin-left:3px;
    margin-right:3px;
}

.button.radius{

```

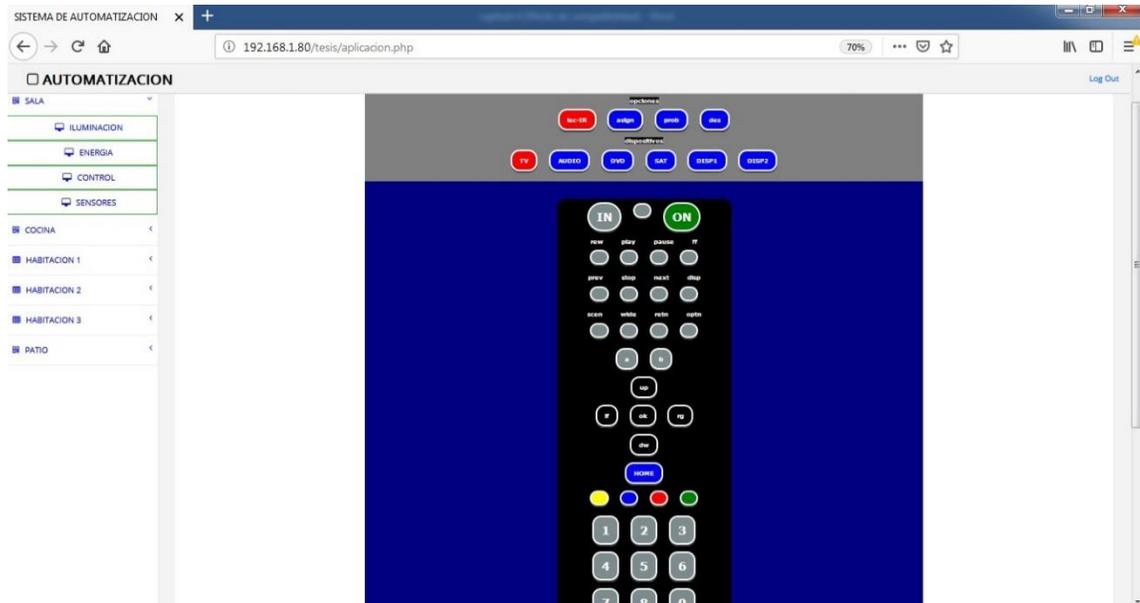
```
font:bold 10px Verdana;
width:100;
margin-left:3px;
margin-right:3px;
}
img.imagen{
width:100%;
height: 90px;
}
}
```

4.4.2.5. Página de control remoto IR

El código de la página de control remoto IR le brinda al usuario la opción de utilizar un control remoto universal para hacer uso de una variedad de dispositivos que son controlados por IR a través de un navegador web, al cual se puede acceder desde una computadora o un teléfono inteligente. La página tiene la capacidad de copiar hasta 6 controles IR, lo cual se logra utilizando las opciones que ofrece la página web, el modo de hacerlo es copiar un comando IR enviando la orden de lectura al microcontrolador maestro destino, el cual, al recibir la orden, iluminará el led indicador para mostrar qué espera el comando del control original del dispositivo. Al realizar esta función el comando queda guardado para ser asignado a un botón.

La página se conforma por un conjunto de lenguaje HTML y JavaScript para el funcionamiento de la página y algunas de las animaciones de la misma, la página también hace uso del lenguaje de PHP, el cual es invocado por medio de una llamada de JavaScript a través de Ajax para el envío de los comandos seleccionados por el usuario, el envío se realiza a través de un *socket* entre Python y PHP, o también según la función que sea seleccionada puede almacenar información en la base de datos que corresponde a esta página. La apariencia de la página se muestra en la siguiente figura:

Figura 36. Página de control remoto IR



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.5.1. Código HTML y JavaScript

```

<!doctype html>
<html lang="es">
<head>
<title></title>
<meta http-equiv="content-Type" content="text/html; charset=iso-8859-1">
<link href="control/index_style_control.css" rel="stylesheet" type="text/css" />
<script src="assets/js/jquery.js"></script>
<script type="text/javascript">
var intervalo;
var area;
$(document).ready(function(){
    clearInterval(intervalo);
});
var disp = 1;
var asignar = 0;
function asignar(){
    if (asignar == 0){
        var opcion = confirm("¿de desea asignar un comando IR?");
        if (opcion == true) {
            asignar = 1;
            $('#asign').css('background-color', 'green');
        }
    }
}
else{
    asignar = 0;
    $('#asign').css('background-color', 'blue');
}
    
```

```

}
}
function tv(){
disp = 1;
    $('#tv').css('background-color', 'red');
    $('#audio').css('background-color', 'blue');
    $('#dvd').css('background-color', 'blue');
    $('#sat').css('background-color', 'blue');
    $('#disp1').css('background-color', 'blue');
    $('#disp2').css('background-color', 'blue');
}
function audio(){
disp = 2;
    $('#tv').css('background-color', 'blue');
    $('#audio').css('background-color', 'red');
    $('#dvd').css('background-color', 'blue');
    $('#sat').css('background-color', 'blue');
    $('#disp1').css('background-color', 'blue');
    $('#disp2').css('background-color', 'blue');
}
function dvd(){
disp = 3;
    $('#tv').css('background-color', 'blue');
    $('#audio').css('background-color', 'blue');
    $('#dvd').css('background-color', 'red');
    $('#sat').css('background-color', 'blue');
    $('#disp1').css('background-color', 'blue');
    $('#disp2').css('background-color', 'blue');
}
function sat(){
disp = 4;
    $('#tv').css('background-color', 'blue');
    $('#audio').css('background-color', 'blue');
    $('#dvd').css('background-color', 'blue');
    $('#sat').css('background-color', 'red');
    $('#disp1').css('background-color', 'blue');
    $('#disp2').css('background-color', 'blue');
}
function disp1(){
disp = 5;
    $('#tv').css('background-color', 'blue');
    $('#audio').css('background-color', 'blue');
    $('#dvd').css('background-color', 'blue');
    $('#sat').css('background-color', 'blue');
    $('#disp1').css('background-color', 'red');
    $('#disp2').css('background-color', 'blue');
}
function disp2(){
disp = 6;
    $('#tv').css('background-color', 'blue');
    $('#audio').css('background-color', 'blue');
    $('#dvd').css('background-color', 'blue');
    $('#sat').css('background-color', 'blue');
    $('#disp1').css('background-color', 'blue');
    $('#disp2').css('background-color', 'red');
}
function proceso(funcion){
var bandera = 1;
if (funcion == 48){
    if (asignar == 1){
        asignar = 0;
        $('#asign').css('background-color', 'blue');
    }
    var opcion = confirm("¿desesa leer un comando IR?");
    if (opcion == false){
        bandera = 0;
    }
}
}

```

```

    }
}
if (funcion == 49){
    if (asignar == 1){
        asignar = 0;
        $('#asign').css('background-color', 'blue');
    }
}
if (funcion == 50){
    if (asignar == 1){
        asignar = 0;
        $('#asign').css('background-color', 'blue');
    }
    var opcion = confirm("¿quiere desechar el comando IR?");
    if (opcion == false){
        bandera = 0;
    }
}
if (bandera == 1){
    var parametros = {funcion, disp, area, asignar};
    $.ajax({
        data: parametros,
        url: 'control/ajax_proceso.php',
        type: 'post',
        success: function (response) {
        }
    });
    if (asignar == 1){
        asignar = 0;
        $('#asign').css('background-color', 'blue');
    }
}
}
</script>
</head>
<body>
<div id="general">
<div id="menu">
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">opciones</a>
</div>
<div>
    <a onclick="proceso('48');return false;" class="button grey" style="background:red; font:bold 10px
Verdana;">lec-IR</a>
    <a id="asign" onclick="asign()" class="button grey" style="background:blue; font:bold 10px
Verdana;">asign</a>
    <a onclick="proceso('49');return false;" class="button grey" style="background:blue; font:bold 10px
Verdana;">prob</a>
    <a onclick="proceso('50');return false;" class="button grey" style="background:blue; font:bold 10px
Verdana;">des</a>
</div>
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">dispositivos</a>
</div>
<div>
    <a id="tv" onclick="tv()" class="button grey" style="background:red; font:bold 10px Verdana;">TV</a>
    <a id="audio" onclick="audio()" class="button grey" style="background:blue; font:bold 10px
Verdana;">AUDIO</a>
    <a id="dvd" onclick="dvd()" class="button grey" style="background:blue; font:bold 10px Verdana;">DVD</a>
    <a id="sat" onclick="sat()" class="button grey" style="background:blue; font:bold 10px Verdana;">SAT</a>
    <a id="disp1" onclick="disp1()" class="button grey" style="background:blue; font:bold 10px
Verdana;">DISP1</a>
    <a id="disp2" onclick="disp2()" class="button grey" style="background:blue; font:bold 10px
Verdana;">DISP2</a>
</div>
</div>
</div>

```

```

<div id="control">
<div>
    <a onclick="proceso('1');return false;" class="button radius grey">IN</a>
    <a onclick="proceso('2');return false;" class="button radius grey"> </a>
    <a onclick="proceso('3');return false;" class="button radius grey" style="background:green;">ON</a>
</div>
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">rew</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">play</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">pause</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">ff</a>
</div>
<div>
    <a onclick="proceso('4');return false;" class="button grey"></a>
    <a onclick="proceso('5');return false;" class="button grey"></a>
    <a onclick="proceso('6');return false;" class="button grey"></a>
    <a onclick="proceso('7');return false;" class="button grey"></a>
</div>
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">prev</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">stop</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">next</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">disp</a>
</div>
<div>
    <a onclick="proceso('8');return false;" class="button grey"> </a>
    <a onclick="proceso('9');return false;" class="button grey"> </a>
    <a onclick="proceso('10');return false;" class="button grey"> </a>
    <a onclick="proceso('11');return false;" class="button grey"> </a>
</div>
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">scen</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">wide</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">retn</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">optn</a>
</div>
<div>
    <a onclick="proceso('12');return false;" class="button grey"> </a>
    <a onclick="proceso('13');return false;" class="button grey"> </a>
    <a onclick="proceso('14');return false;" class="button grey"> </a>
    <a onclick="proceso('15');return false;" class="button grey"> </a>
</div>
<div>
    <a onclick="proceso('16');return false;" class="button grey" style="font:bold 10px Verdana;">a</a>
    <a onclick="proceso('17');return false;" class="button grey" style="font:bold 10px Verdana;">b</a>
</div>
<div>
    <a onclick="proceso('18');return false;" class="button grey" style="background:black; font:bold 10px
Verdana;">up</a>
</div>
<div>
    <a onclick="proceso('19');return false;" class="button grey" style="background:black; font:bold 10px
Verdana;">lf</a>
    <a onclick="proceso('20');return false;" class="button grey" style="background:black; font:bold 10px
Verdana;">ok</a>
    <a onclick="proceso('21');return false;" class="button grey" style="background:black; font:bold 10px
Verdana;">rg</a>
</div>
<div>
    <a onclick="proceso('22');return false;" class="button grey" style="background:black; font:bold 10px
Verdana;">dw</a>
</div>
<div>
    <a onclick="proceso('23');return false;" class="button grey" style="background:blue; font:bold 10px
Verdana;">HOME</a>
</div>

```

```

</div>
    <a onclick="proceso('24');return false;" class="button grey" style="background:yellow;"> </a>
    <a onclick="proceso('25');return false;" class="button grey" style="background:blue;"> </a>
    <a onclick="proceso('26');return false;" class="button grey" style="background:red;"> </a>
    <a onclick="proceso('27');return false;" class="button grey" style="background:green;"> </a>
</div>
<div>
    <a onclick="proceso('28');return false;" class="button grey">1</a>
    <a onclick="proceso('29');return false;" class="button grey">2</a>
    <a onclick="proceso('30');return false;" class="button grey">3</a>
</div>
<div>
    <a onclick="proceso('31');return false;" class="button grey">4</a>
    <a onclick="proceso('32');return false;" class="button grey">5</a>
    <a onclick="proceso('33');return false;" class="button grey">6</a>
</div>
<div>
    <a onclick="proceso('34');return false;" class="button grey">7</a>
    <a onclick="proceso('35');return false;" class="button grey">8</a>
    <a onclick="proceso('36');return false;" class="button grey">9</a>
</div>
<div>
    <a onclick="proceso('37');return false;" class="button grey">M</a>
    <a onclick="proceso('38');return false;" class="button grey">0</a>
    <a onclick="proceso('39');return false;" class="button grey">J</a>
</div>
<div>
    <a onclick="proceso('40');return false;" class="button radius grey">v+</a>
    <a onclick="proceso('41');return false;" class="button radius grey" style="">c+</a>
</div>
<div>
    <a onclick="proceso('42');return false;" class="button radius grey">v-</a>
    <a onclick="proceso('43');return false;" class="button radius grey" style="">c-</a>
</div>
<div>
    <a class="texto" style="background:black; font:bold 10px Verdana;">bt-a</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">bt-b</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">bt-c</a>
    <a class="texto" style="background:black; font:bold 10px Verdana;">bt-d</a>
</div>
<div>
    <a onclick="proceso('44');return false;" class="button grey"> </a>
    <a onclick="proceso('45');return false;" class="button grey"> </a>
    <a onclick="proceso('46');return false;" class="button grey"> </a>
    <a onclick="proceso('47');return false;" class="button grey"> </a>
</div>
<div>
    <a class="texto" style="font:bold 20px Verdana;">TESIS</a>
</div>
</div>
</div>
</div>
</body>
</html>

```

4.4.2.5.2. Código PHP

```
<?php
$func = $_POST['funcion'];
$func = 'fun';
$func .= $nfunc;
$disp = $_POST['disp'];
$area = $_POST['area'];
$asign = $_POST['asignar'];

if ($asign == 0 && $func < 48){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT $func, id FROM control WHERE id='$disp',$conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    mysql_free_result($result);
    mysql_close($conn);
    if ($comando[0] != ""){
        error_reporting(E_ALL);
        $address = gethostbyname('localhost');
        $service_port = 10000;
        $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
        $result = socket_connect($socket, $address, $service_port);
        socket_write($socket, $comando[0], strlen($comando[0]));
        socket_close($socket);
    }
    echo $comando[0];
}

if ($asign == 1){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT funx, id FROM control WHERE id=1",$conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    if ($comando[0] != ""){
        $ssql = "UPDATE control SET $func='{$comando[0]}' WHERE id = '$disp'";
        $rs = mysql_query($ssql,$conn);
        mysql_free_result($rs);
    }
    mysql_close($conn);
    echo $comando[0];
}

if ($nfunc == 48){
    if ($area == "sala"){ $in = "0aTX1bIRRECcdz";}
    if ($area == "cocina"){ $in = "1aTX1bIRRECcdz";}
    if ($area == "hab1"){ $in = "2aTX1bIRRECcdz";}
    if ($area == "hab2"){ $in = "3aTX1bIRRECcdz";}
    if ($area == "hab3"){ $in = "4aTX1bIRRECcdz";}
    if ($area == "patio"){ $in = "5aTX1bIRRECcdz";}
    error_reporting(E_ALL);
    $address = gethostbyname('localhost');
    $service_port = 10000;
    $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
    $result = socket_connect($socket, $address, $service_port);
    socket_write($socket, $in, strlen($in));
    socket_close($socket);
}
```

```

if ($func == 49){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT funx, id FROM control WHERE id=1",$conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    mysql_free_result($result);
    mysql_close($conn);
    if ($comando[0] != ""){
        error_reporting(E_ALL);
        $address = gethostbyname('localhost');
        $service_port = 10000;
        $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
        $result = socket_connect($socket, $address, $service_port);
        socket_write($socket, $comando[0], strlen($comando[0]));
        socket_close($socket);
    }
    echo $comando[0];
}

if ($func == 50){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT funx, id FROM control WHERE id=2",$conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    $ssql = "UPDATE control SET funx='{$comando[0]}' WHERE id = 1";
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
    mysql_close($conn);
    echo $comando[0];
}
?>

```

4.4.2.5.3. Código CSS

```

*{
    margin:0px;
    padding:0px;
}

.button {
    font:bold 20px Verdana;
    color:#fff;
    margin:10px 0px;
    margin-left:8px;
    margin-right:8px;
    margin-top:6px;
    margin-bottom:6px;
    border:4px solid #fff;
    padding:10px 13px 10px 13px;
    display:inline-block;
    box-sizing:border-box;
    border-radius:15px;
    text-align:center;
    width:200;
    text-decoration:none;
    transition:box-shadow .5s,padding-top .5s,background .5s;
}

```

```

.texto{
    color:#fff;
    margin-left:14px;
    margin-right:14px;
    margin-top:2px;
    margin-bottom:2px;
}

.button.blue {
    background:#3498db;
    box-shadow:0px 3px 0px #266792;
}

.button.grey {
    background:#7f8c8d;
    box-shadow:0px 3px 0px #373c3c;
}

.button.medium {
    width:350px;
}

.button.large {
    width:450px;
}

.button.radius {
    border-radius:50px;
    font:bold 20px Verdana;
}

.button:hover {
    box-shadow:0px 0px 0px;
    padding-top:7px;
    background:#3498db;
}

div#general{
    margin:auto;
    margin-top:30px;
    width:960px;
    height:1300px;
    background-color:navy;
    text-align:center;
}

div#menu{
    margin:auto;
    width:960px;
    height:150px;
    background-color:grey;
    text-align:center;
}

div#control{
    margin:auto;
    margin-top:30px;
    width:300px;
    height:1100px;
    background-color:black;
    border-radius:15px 15px 15px 15px;
}

@media screen and (max-width: 400px) {

```

```

div#general {
    width:100%;
    height:1250px;
    text-align:center;
}

div#menu{
    width:100%;
    height:200px;
    text-align:center;
}

div#control{
    width:60%;
    height:1000px;
}

.texto{
    color:#fff;
    margin-left:10px;
    margin-right:10px;
}

.button{
    font:bold 10px Verdana;
    width:100;
    margin-left:3px;
    margin-right:3px;
}

.button.radius{
    font:bold 10px Verdana;
    width:100;
    margin-left:3px;
    margin-right:3px;
}
}

```

4.4.2.6. Página de sensores

La página de sensores tiene el objetivo de mostrar las lecturas de los sensores del sistema en el área seleccionada, los sensores con los que cuenta el sistema son: sensor de temperatura, sensor de movimiento, sensor de gas combustible y sensor de puertas. El sistema en la sección de sensores también permite el modo alarma, el cual al activarlo muestra en la página si hay una detección de movimiento por el sensor PIR o una apertura de puerta. La alarma de gas siempre se encuentra activada e indicará en la página si detecta algún gas, además en el dispositivo se emitirá un sonido y se encenderá la luz de emergencia.

La página cuenta con 3 botones, uno de ellos es para activar o desactivar el modo de alarma del sistema, el cual se mencionó anteriormente. Los otros dos botones son para apertura de 2 puertas de una misma área.

Figura 37. **Página de sensores**



Fuente: elaboración propia, empleando navegador web Firefox.

4.4.2.6.1. Código HTML y JavaScript

```
<!doctype html>

<html lang="es">
<head>
<title></title>
<meta http-equiv="content-Type" content="text/html; charset=iso-8859-1">
<link href="sensores/index_style_sensores.css" rel="stylesheet" type="text/css" />
<script src="assets/js/jquery.js"></script>

<script type="text/javascript">
var intervalo;
var area;
var tabla = "sensores";

$(document).ready(function(){
    clearInterval(intervalo);
```

```

    actualizar();
    intervalo = setInterval("actualizar()", 5000);
    $('#puertas').css('background-color', '#EAEBED');
    $('#alarma').css('background-color', '#EAEBED');
    $('#temperatura').css('background-color', '#EAEBED');
    $('#humedad').css('background-color', '#EAEBED');
    $('#icalor').css('background-color', '#EAEBED');
    $('#CO2').css('background-color', '#EAEBED');
});

function actualizar(){
var parametros1 = {area};
$.ajax({
    data: parametros1,
    url: 'sensores/actualizar.php',
    type: 'post',
    success: function (response) {
        var myArray = jQuery.parseJSON(response);
        $("#puertas").val(myArray.puertas);
        $("#alarma").val(myArray.alarma);
        $("#temperatura").val(myArray.temperatura + " °C");
        $("#humedad").val(myArray.humedad + " % HR");
        $("#icalor").val(myArray.icalor + " °C");
        $("#CO2").val(myArray.CO2);
        if (myArray.estado == "SI"){
            $('#alarma').css('background-color', 'red');
        }else{
            $('#alarma').css('background-color', '#EAEBED');
        }
        if (myArray.alarma == "activada"){
            $('#3').css('background-color', 'red');
            if (myArray.puertas == "ABIERTA"){
                $('#puertas').css('background-color', 'red');
            }else{
                $('#puertas').css('background-color', '#EAEBED');
            }
        }else{
            $('#3').css('background-color', 'blue');
            $('#puertas').css('background-color', '#EAEBED');
            $('#alarma').css('background-color', '#EAEBED');
        }
        if (myArray.CO2 == "SI"){
            $('#CO2').css('background-color', 'red');
        }else{
            $('#CO2').css('background-color', '#EAEBED');
        }
    }
});
}

function puerta1(funcion, val){
    $('#1').css('background-color', 'red');
    proceso(1)
}
function puerta2(funcion, val){
    $('#2').css('background-color', 'red');
    proceso(2)
}
function alarma(funcion, val){
    $('#3').css('background-color', 'red');
    proceso(3)
}
function proceso(funcion){
    var id = "#" + funcion.toString();
    var parametros = {funcion, area, tabla};
    $.ajax({

```

```

data: parametros,
url: 'sensores/ajax_proceso.php',
type: 'post',
success: function (response) {
    if (funcion != 3){
        $(id).css('background-color', 'blue');
    }else if (response == "ON"){
        $(id).css('background-color', 'red');
    }else{
        $(id).css('background-color', 'blue');
    }
}
});
}
</script>
</head>
<body>
<div id="general">
<div>
    </img>
</div>
    <!http://www.metrocuadrado.com/noticias/sites/default/files/styles/full_image/public/field/image/dibujo_de_casa
_con_>
    <!herramientas_al_rededor.jpg?itok=KaJ9ppzk>
<div id="menu">
    <label id="titulo" class="texto">SENSORES</label>
</div>
<div id="control">
<div id="contenedor">
<div class = "infosensores">
    <div class = "info">
        <label class="texto2">PUERTA:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "puertas" size="15" disabled value = "----" />
    </div>
    <div class = "info">
        <label class="texto2">ALARMA:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "alarma" size="15" disabled value = "----" />
    </div>
    <div class = "info">
        <label class="texto2">TEMPERATURA:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "temperatura" size="15" disabled value = "----" />
    </div>
    <div class = "info">
        <label class="texto2">HUMEDAD:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "humedad" size="15" disabled value = "----" />
    </div>
    <div class = "info">
        <label class="texto2">INDICE DE CALOR:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "icalor" size="15" disabled value = "----" />
    </div>
    <div class = "info">
        <label class="texto2">ALARMA DE GAS:&nbsp;</label>
    </div>
    <div class = "inputtext">
        <input type="text" class="texto2" align='right' id = "CO2" size="15" disabled value = "----" />
    </div>
</div>

```

```

</div>
</div>
<div id="contenedor2">
<div>
    <a id = "1" onclick="puerta1()" class="button grey" style="background:blue; font:bold 10px Verdana;">PUERTA
1</a>
</div>
<div>
    <a id = "2" onclick="puerta2()" class="button grey" style="background:blue; font:bold 10px Verdana;">PUERTA
2</a>
</div>
<div>
    <a id = "3" onclick="alarma()" class="button grey" style="background:blue; font:bold 10px
Verdana;">ALARMA</a>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

4.4.2.6.2. Código PHP

Esta página contiene dos códigos PHP, el primero se encarga de la actualización de los datos mostrados en la página, la llamada a este código la hace JavaScript cada cierto período de tiempo establecido en la página. A continuación se muestra el código de la actualización:

```

<?php
$area = $_POST['area'];
$conn = mysql_connect("localhost","root","orangeipi");
mysql_select_db($area,$conn);
$result=mysql_query("SELECT * FROM sensores",$conn);
while($row = mysql_fetch_array($result)) {
    $comando=$row;
}
mysql_free_result($result);
mysql_close($conn);
$datos = array(
'puertas' => $comando['1'],
'alarma' => $comando['2'],
'temperatura' => $comando['3'],
'humedad' => $comando['4'],
'icalor' => $comando['5'],
'CO2' => $comando['6'],
'estado' => $comando['7']
);
echo json_encode($datos);
?>

```

El segundo código se encarga de la gestión de las opciones que seleccionó el usuario, las procesa y las ordena según el área que corresponde a la opción, luego almacena en una base de datos los valores o estados establecidos y arma un comando que envía por medio de un *socket* al programa que corre en Python.

```

<?php
$funcion = $_POST['funcion'];
$area = $_POST['area'];
$tabla = $_POST['tabla'];

if ($area == "sala"){ $narea = "0aTX1b";}
if ($area == "cocina"){ $narea = "1aTX1b";}
if ($area == "hab1"){ $narea = "2aTX1b";}
if ($area == "hab2"){ $narea = "3aTX1b";}
if ($area == "hab3"){ $narea = "4aTX1b";}
if ($area == "patio"){ $narea = "5aTX1b";}

if($funcion == 1){
    $com = $narea . "PUERTA" . $funcion . "cdz";
}

if($funcion == 2){
    $com = $narea . "PUERTA" . $funcion . "cdz";
}

if($funcion == 3){
    $conn = mysql_connect("localhost","root","orangepi");
    mysql_select_db($area,$conn);
    $result=mysql_query("SELECT alarma FROM $tabla",$conn);
    while($row = mysql_fetch_array($result)) {
        $comando=$row;
    }
    if ($comando[0] == "activada"){
        $ssql = "UPDATE $tabla SET alarma='desactivada', estado = 'NO' WHERE id = '1'";
        $com = $narea . "ALARMaOFFdz";
        $estado = "OFF";
    } else if ($comando[0] == "desactivada"){
        $ssql = "UPDATE $tabla SET alarma='activada' WHERE id = '1'";
        $com = $narea . "ALARMaONdz";
        $estado = "ON";
    }
    $rs = mysql_query($ssql,$conn);
    mysql_free_result($rs);
    mysql_free_result($result);
    mysql_close($conn);
}

if ($com != ""){
    error_reporting(E_ALL);
    $address = gethostbyname('localhost');
    $service_port = 10000;
    $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
    $result = socket_connect($socket, $address, $service_port);
    socket_write($socket, $com, strlen($com));
    socket_close($socket);
    echo $estado;
}

sleep(1);
?>

```

4.4.2.6.3. Código CSS

```

*{
    margin:0px;
    padding:0px;
}
.button {
    font:bold 20px Verdana;
    color:white;
    margin:10px 0px;
    margin-left:8px;
    margin-right:8px;
    margin-top:20px;
    margin-bottom:15px;
    border:4px solid #FFF;
    padding:10px 13px 10px 13px;
    display:inline-block;
    box-sizing:border-box;
    border-radius:15px;
    text-align:center;
    width:150px;
    text-decoration:none;
    transition:box-shadow .5s,padding-top .5s,background .5s;
}
.texto{
    font:bold 25px Verdana;
    color:white;
}
.texto2{
    font:bold 13px Verdana;
    color:black;
    margin-left:4px;
    margin-right:4px;
}
.button.blue {
    background:#3498db;
    box-shadow:0px 3px 0px #266792;
}
.button.grey {
    background:#7f8c8d;
    box-shadow:0px 3px 0px #373c3c;
}
.button.medium {
    width:350px;
}
.button.large {
    width:450px;
}
.button.radius {
    border-radius:50px;
    font:bold 20px Verdana;
}
.button:hover {
    box-shadow:0px 0px 0px;
    padding-top:7px;
    background:#3498db;
}
div#general{
    margin:auto;
    margin-top:30px;
    width:960px;
    height:595px;
    background-color:navy;
}

```

```

        text-align:center;
    }
    img.imagen{
        width: 960px;
        height: 220px;
    }
    div#menu{
        margin:auto;
        width:960px;
        height:50px;
        line-height: 50px;
        background-color:grey;
        text-align:center;
    }
    label#titulo{
        display: inline-block;
        vertical-align: middle;
        line-height: normal;
    }
    div#contenedor{
        float:left;
        margin-top:15px;
        width:50%;
        height:230px;
        background-color:white;
        border:solid;
        border-width:3px;
        border-color:black;
        border-radius:15px 15px 15px 15px;
    }
    .infosensores{
        margin:auto;
        width:96%;
        height:30px;
        background-color:white;
        border-radius:15px 15px 15px 15px;
    }
    .info{
        float:left;
        margin-top:15px;
        width:50%;
        height:20px;
        background-color:white;
        text-align:right;
    }
    .inputtext{
        float:right;
        margin-top:15px;
        width:50%;
        background-color:white;
    }
    div#contenedor2{
        margin:auto;
        float:left;
        margin-top:15px;
        width:50%;
        height:230px;
        background-color:#C4C5C6;
        border:solid;
        border-width:3px;
        border-color:black;
        border-radius:15px 15px 15px 15px;
    }
    div#control{
        margin:auto;
        margin-top:30px;
    }

```

```

width:70%;
height:260px;
background-color:blue;
border-radius:15px 15px 15px 15px;
text-align:center;
}
@media screen and (max-width: 400px) {
div#general {
width:100%;
height:635px;
text-align:center;
}
div#menu{
width:100%;
height:30px;
line-height: 30px;
text-align:center;
}
label#titulo{
display: inline-block;
vertical-align: middle;
line-height: normal;
}
div#control{
margin:auto;
margin-top:20px;
width:80%;
height:365px;
text-align:center;
}
div#contenedor{
margin-top:15px;
margin:auto;
float:center;
width:100%;
height:230px;
background-color:white;
border:solid;
border-width:3px;
border-color:black;
border-radius:15px 15px 15px 15px;
text-align:center;
}
div#contenedor2{
margin-top:15px;
width:100%;
height:230px;
background-color:#C4C5C6;
border-radius:15px 15px 15px 15px;
border:solid;
border-width:3px;
border-color:black;
text-align:center;
}
.control1{
margin:auto;
margin-top:15px;
margin-left:14px;
margin-right:15px;
width:90%;
height:160px;
background-color:white;
border:solid;
border-width:3px;
border-color:black;
border-radius:15px 15px 15px 15px;
}

```

```

        text-align:center;
    }
    .control2{
        margin:auto;
        margin-top:15px;
        margin-left:14px;
        margin-right:15px;
        width:90%;
        height:130px;
        background-color:white;
        border:solid;
        border-width:3px;
        border-color:black;
        border-radius:15px 15px 15px 15px;
        text-align:center;
    }
    .infosensores{
        margin:auto;
        width:96%;
        height:30px;
        background-color:white;
        border-radius:15px 15px 15px 15px;
    }
    .info{
        float:left;
        margin-top:15px;
        width:50%;
        height:20px;
        background-color:white;
        text-align:right;
    }
    .inputtext{
        float:right;
        margin-top:15px;
        width:50%;
        background-color:white;
    }
    .texto{
        color:#fff;
        margin-left:10px;
        margin-right:10px;
    }
    .texto2{
        font:bold 10px Verdana;
    }
    .button{
        font:bold 10px Verdana;
        width:100;
        margin-left:3px;
        margin-right:3px;
    }
    .button.radius{
        font:bold 10px Verdana;
        width:100;
        margin-left:3px;
        margin-right:3px;
    }
    img.imagen{
        width:100%;
        height: 90px;
    }
}

```

CONCLUSIONES

1. Existe una gran variedad de alternativas a la Raspberry Pi que no son muy conocidas en este país y que son de menor costo.
2. Un sistema de domótica es de gran ayuda para el manejo de los recursos de un hogar, pero no todas las personas están acostumbradas al uso de este tipo de tecnologías o al manejo de un dispositivo móvil.
3. Existen varios sistemas de domótica que son de tipo comercial, muchos son excesivamente caros y de difícil instalación, pero cubren una gran variedad de funciones con respecto al diseño propuesto, y algunos más económicos y de instalación simple, pero solo se encargan de la automatización de una función en específico.
4. En Internet existe más información para el uso de los mini ordenadores Raspberry Pi que para la Orange Pi.
5. Se determinó que es posible y es viable la realización de un sistema de domótica con el mini ordenador Orange Pi.

RECOMENDACIONES

1. Conocer el funcionamiento de cada uno de los dispositivos que son utilizados en este sistema.
2. Conocer el funcionamiento y el tipo de direcciones que utiliza el módulo inalámbrico NRF24L01.
3. Leer las instrucciones en los códigos de programación en el capítulo 4 de este trabajo, ya que para el correcto funcionamiento de este sistema se debe respetar el direccionamiento establecido de las tuberías de comunicación del módulo NRF24L01.
4. Verificar que el microcontrolador 1 siempre sea instalado en línea vista con los dispositivos a controlar por medio de IR para su correcto funcionamiento.

BIBLIOGRAFÍA

1. Arduino. *¿Qué es Arduino?* [en línea]. <<https://arduinodhtics.weebly.com/iquestqueacute-es.html>>. [Consulta: 11 de enero de 2018].
2. ARMBIAN. *Linux for arm development boards*. [en línea]. <<https://docs.armbian.com/>>. [Consulta: 16 de enero de 2018].
3. BOTSCIENCE. *Módulo RF transceiver NRF24L01*. [en línea] <http://botscience.net/store/index.php?route=product/product&product_id=105>. [Consulta: 12 de enero de 2018].
4. CEDOM. *¿Qué es domótica?* [en línea]. <<http://www.cedom.es/sobre-domotica/que-es-domotica>>. [Consulta: 7 de enero de 2018].
5. Domótica. *Características generales*. [en línea]. <<http://domotica7.blogspot.com/p/caracteristicas.html>>. [Consulta: 8 de enero de 2018].
6. _____. *La arquitectura del sistema*. [en línea]. <<http://domotica1003.weebly.com/la-arquitectura-del-sistema-de-domoacutetica.html>>. [Consulta: 8 de enero de 2018].
7. MASADELANTE. *Javascript*. [en línea]. <<http://www.masadelante.com/faqs/javascript>>. [Consulta: 16 de enero de 2018].

8. _____ . *¿Qué son las hojas de estilo?* [en línea].
<<http://www.masadelante.com/faqs/css>>. [Consulta: 16 de enero de 2018].
9. Monografías. *Domótica*. [en línea].
<<http://www.monografias.com/trabajos35/domotica/domotica.shtml>>. [Consulta: 7 de enero de 2018].
10. Orange Pi en español. *¿Qué es Orange Pi?* [en línea].
<<http://orangepiweb.es/>>. [Consulta: 10 de enero de 2018].
11. Orange Pi. *What's Orange Pi PC?* [en línea].
<<http://www.orangepi.org/orangepipc/>>. [Consulta: 10 de enero de 2018].
12. PHP. *¿Qué es PHP?* [en línea].
<<http://php.net/manual/es/intro-what-is.php>>. [Consulta: 16 de enero de 2018].
13. VENDOMOTICA. *¿Qué es un dimmer?* [en línea].
<<http://vendomotica.com/blog/que-es-un-dimmer/>>. [Consulta: 12 de enero de 2018].