

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DOMÓTICO PARA SEGURIDAD Y CONTROL DE ILUMINACIÓN DE UNA VIVIENDA UTILIZANDO UN MICROCOMPUTADOR RASPBERRY PI 3

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

MARCO AURELIO CAIZA TIPAN

marco_18nata@hotmail.com

DIRECTOR: ING. LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.

leandro.pazmino@epn.edu.ec

CODIRECTOR: ING. MONICA VINUEZA RHOR, MSC.

monivinueza@yahoo.com.ar

Quito, Febrero 2018

DECLARACIÓN

Yo CAIZA TIPAN MARCO AURELIO, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en éste documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a éste trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Caiza Tipan Marco Aurelio

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por **CAIZA TIPAN MARCO AURELIO**, bajo nuestra supervisión.

Ing. Leandro Pazmiño MSc.
DIRECTOR DE PROYECTO

Ing. Mónica Vinuesa Rhor MSc.
CODIRECTORA DEL PROYECTO

DEDICATORIA

A Dios y a la Virgencita Maria Auxiliadora por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad

.

A mis padres Aurelio Caiza y Carmen Tipán por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo, por ser un excelente ejemplo de vida a seguir.

A mis sobrinos Ariel y Christopher Nacimba, que con su cariño y afecto han llenado mi corazón de amor y me han dado fuerzas de seguir adelante.

Marco Aurelio Caiza

AGRADECIMIENTO

Agradezco a mis Amigos Diego Alvear, Marlon Erazo y Jenny Falcón por haber hecho de mi etapa estudiantil un trayecto lleno de vivencias que nunca olvidaré estando en los buenos y malos momentos en los que hemos pasado riendo, llorando, jugando, sufriendo en fin esas son las cosas que dan sentido a la vida.

A mis hermanos Katty, Evelyn, Diego por ser parte importante de mi vida, por su apoyo económico brindado durante mi vida estudiantil.

Le agradezco la confianza, apoyo y dedicación de tiempo a mi director de tesis Ing. Leandro Antonio Pazmiño Ortiz. Por haber compartido conmigo sus conocimientos y sobre todo su amistad.

A mi codirectora de tesis, Ing. Mónica Vinueza quien con su conocimiento y experiencia supo corregir mis errores y convertirme en un buen profesional.

A todos quienes contribuyeron de una u otra manera en la realización de este proyecto y que se me pasa por alto, Gracias. A veces el camino es largo, pero no imposible.

Y por último sin significar que sea menos importante a mi amigo de años Diego Quisphe por haberme facilitado materiales utilizados para el desarrollo de este trabajo de una manera desinteresada, por sus ideas locas pero funcionales, por haberme motivado a matricularme en la Poli años atrás, muchas gracias.

Marco Aurelio Caiza

CONTENIDO

DECLARACIÓN.....	II
CERTIFICACIÓN.....	III
DEDICATORIA	IV
AGRADECIMIENTO	V
CONTENIDO	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	X
RESUMEN.....	XI
ABSTRACT.....	XII
1. INTRODUCCIÓN.....	13
1.1 Planteamiento del Problema.....	14
1.2 Justificación.....	14
1.3 Marco Teórico	15
Análisis del microcomputador Raspberry Pi 3 y sus versiones.....	15
Entradas y salidas de propósito general (GPIO).....	17
Instalación y configuración del sistema operativo Raspbian	19
Configuración de <i>Raspbian</i>	20
Configuración de la Raspberry con una dirección IP fija	21
Configuración e instalación del servidor Web	24
Instalación de la librería WiringPi	25
2. METODOLOGÍA.....	27
2.1 Funcionamiento del proyecto.....	28
3. RESULTADOS Y DISCUSIÓN	29
3.1 Descripción general del sistema.....	29
3.2 Diseño de las conexiones del microcomputador Raspberry Pi 3.....	30
3.3 Página de login.....	31

3.4	Diseño del control de iluminación	35
3.5	Diseño del control de ventanas y puerta	43
3.6	Diseño del sistema de control de seguridad.....	48
3.7	Implementación de la maqueta y pruebas de funcionamiento.....	56
3.8	Diseño y ensamblaje de la maqueta.....	62
4.	CONCLUSIONES Y RECOMENDACIONES.....	64
	Conclusiones:	64
	Recomendaciones:	66
	BIBLIOGRAFÍA.....	68
	ANEXOS	73
	ANEXO A: Conceptos necesarios.....	73
	ANEXO B: Manual de usuario para el administrador del prototipo	109
	ANEXO C:Características de los dispositivos utilizados en el prototipo.	113
	ANEXO D: Maqueta del prototipo	115
	ANEXO E : Costo de implementación.....	116

ÍNDICE DE FIGURAS

Fig. 1.1 Ubicación de los pines en la placa	17
Fig. 1.2 Numeración GPIO y numeración física	18
Fig. 1.3 Configuración para cargar Raspbian en la Raspberry	20
Fig. 1.4 Menú que ofrece comando raspi-config	20
Fig. 1.5 Fichero interfaces modificado con las direcciones IP estáticas	22
Fig. 1.6 Configuración de la red inalámbrica	23
Fig. 1.7 Página web inicial de Apache	24
Fig. 1.8 Ubicación del archivo index.php utilizando WinScp	25
Fig. 1.9 Página para descargar WiringPi	26
Fig. 3.1 Diagrama general del sistema domótico	29
Fig. 3.2 Distribución de los pines acorde a las tareas a realizar	30
Fig. 3.3 Página de login	31
Fig. 3.4 Página principal sistema domótico	32
Fig. 3.5 Archivo php.ini instalación MySQL	33
Fig. 3.6 Archivo apache.config Instalación MySQL	33
Fig. 3.7 Página principal MySQL	34
Fig. 3.8 Circuito electrónico para el control de iluminación	35
Fig. 3.9 Circuito de acoplamiento AC-DC	35
Fig. 3.10 Diagrama de funcionamiento del control de iluminación	36
Fig. 3.11 Página Web del control de iluminación de la vivienda	37
Fig. 3.12 Circuito implementado para el control de ventanas	44
Fig. 3.13 Creación de imágenes para la página Web	45
Fig. 3.14 Diagrama de bloques sistema de control de ventanas	46
Fig. 3.15 Página principal del sistema de control de ventanas	46
Fig. 3.16 Circuito implementado del sistema de seguridad	48
Fig. 3.17 Imagen captada desde la Raspberry con Fsw webcam	50
Fig. 3.18 Códigos necesarios para utilizar en python	51
Fig. 3.19 Diagrama de bloques del sistema de seguridad	54
Fig. 3.20 Circuito esquemático diseñado en Proteus	56
Fig. 3.21 Circuito impreso diseñado en Ares	57
Fig. 3.22 Ping extendido	57
Fig. 3.23 Página principal del proyecto de titulación	58
Fig. 3.24 Página Web principal	58
Fig. 3.25 Ejemplo del funcionamiento del sistema de iluminación	59
Fig. 3.26 Ejemplo del funcionamiento del estado de ventanas y de la puerta	60

Fig. 3.27	Página Web sistema de seguridad	61
Fig. 3.28	Alarma activa con detección de intrusos	61
Fig. 3.29	Alarma desactivada sin detección de intrusos	62
Fig. 3.30	Alarma activa sin detección de intrusos	62
Fig. 3.31	Plano de la maqueta en 3D	63
Fig. 3.32	Plano de la maqueta	63
Fig. 4.1	Definición teórica de domótica mediante la disponibilidad de un bus doméstico	73
Fig. 4.2	Arquitectura domótica centralizada	74
Fig. 4.3	Raspberry Pi A	75
Fig. 4.4	Raspberry Pi A +	76
Fig. 4.5	Raspberry Pi Modelo B	76
Fig. 4.6	Raspberry Pi Modelo B+	77
Fig. 4.7	Raspberry Pi2 modelo B	77
Fig. 4.8	Raspberry Pi 3 Modelo B	78
Fig. 4.9	Sensor de movimiento PIR	79
Fig. 4.10	Lentes del sensor PIR.....	79
Fig. 4.11	Cambio diferencial producido por la detección de un cuerpo caliente	79
Fig. 4.12	Diagrama de pines del sensor PIR	80
Fig. 4.13	Entorno de trabajo win32 Disk Imager.....	81
Fig. 4.14	Entorno de trabajo Dreamweaver	82
Fig. 4.15	Entorno de transferencia de archivos WinScp	83
Fig. 4.16	Modelo cliente servidor	84
Fig. 4.17	Prototipo Domótico.....	115

ÍNDICE DE TABLAS

Tabla 1.1 Versiones del microcomputador Raspberry Pi	16
Tabla 1.2 Distribución de los pines del microcomputador Raspberry Pi 3.....	17
Tabla 3.1 Partes del objeto open y send	40
Tabla 4.1 Porcentaje de utilización de servidores	84
Tabla 4.2 Especificaciones técnicas sensor PIR	113
Tabla 4.3 Especificaciones técnicas sensor magnético	113
Tabla 4.4 Distribución de pines del microcomputador Raspberry Pi 3	114
Tabla 4.5 Características del módulo relés	114
Tabla 4.6 Costos directos.....	116
Tabla 4.7 Costos indirectos.....	116
Tabla 4.8 Costo total de implementación.....	116

RESUMEN

La finalidad del siguiente proyecto es el diseño de un prototipo domótico que sea capaz de realizar tres tareas, control de iluminación, verificación del estatus de ventanas y puertas y finalmente que sea capaz de controlar un sistema de seguridad de la vivienda diseñado para cuando el usuario deje su domicilio solo.

Este proyecto está compuesto por fundamentos teóricos, una breve descripción de cada uno de los dispositivos utilizados, además se crearon tablas de características de los elementos principales todo esto con la finalidad de que el lector pueda desarrollar y mejorar el sistema domótico, sirviendo éste proyecto como una fuente de consulta. También se realizó un resumen de las versiones anteriores del microcomputador *Raspberry Pi* con la finalidad de realizar un cuadro comparativo de todas las antiguas versiones, además de ver las ventajas que trae el microcomputador *Raspberry Pi 3*. Se detalla los softwares utilizados para éste proyecto y sus respectivos pasos para su instalación.

El prototipo está formado por elementos electrónicos; el principal elemento utilizado es el microcomputador *Raspberry Pi 3*, el cual actúa como la unidad central de procesamiento. Adicionalmente, se ocupó un sensor PIR y varios sensores magnéticos los cuales, se encargan de sensor y enviar información a la unidad de procesamiento, para que esta pueda realizar alguna acción dependiendo de los datos obtenidos.

La interfaz por medio de la cual es controlado el sistema domótico está formada por cuatro páginas Web para cada una de las tareas, dichas páginas se crearon utilizando el software *Dreamweaver*. Estas páginas funcionan con diferentes lenguajes de programación como: *Php*, *Java script* y *Python*, permitiendo de esta manera tener una interacción total y completa entre el sistema domótico y el usuario.

El proyecto cuenta con un sistema de control de acceso el cual solicita un usuario y contraseña para ingresar al sistema de control domótico. Para la creación de los usuarios se utilizó una base de datos sencilla empleando *MySQL*.

Al finalizar el presente proyecto, se presenta un prototipo funcional montado en una maqueta, la cual servirá para comprobar el funcionamiento y el cumplimiento de los objetivos propuestos, así como para plantear las conclusiones y recomendaciones del proyecto.

ABSTRACT

The purpose of the following project is to design a domotic prototype which is capable of performing three tasks: lighting control, verification of the status of *windows* and doors and finally to control a security system of the home designed to be used only when the user leaves your home.

This project is composed of theoretical foundations, a brief description of each of the devices used, and tables of characteristics of the main elements were created, all with the purpose of allowing the reader to develop and improve the home automation system, serving this project as a source of consultation. A summary of the previous versions of the *Raspberry Pi* microcomputer was also made in order to make a comparative table of all the old versions, as well as to notice the advantages that the *Raspberry Pi 3* microcomputer brings. The software used for this project with its respective steps for its installation are detailed.

The prototype is made up of electronic elements; the main element used is the *Raspberry Pi 3* microcomputer, which acts as the central processing unit, in addition a PIR sensor and several magnetic sensors were occupied which are responsible for sensing and sending information to the unit processing, so that it can perform some action depending on the data obtained.

The interface through which the home automation system is controlled consists of four Web pages for each of the tasks, these pages were created using the *Dreamweaver* software. These pages work with different programming languages such as: *Php*, Java script and *python*, allowing in this way to have a total and complete interaction between the home automation system and the user.

The project has an access control system which requests a username and a password to access to the home automation control system. For the creation of the users a simple database using *MySQL* is being used.

At the end of this project, a functional prototype is presented mounted on a model, which will serve to verify the operation and compliance with the proposed objectives, as well as to present the conclusions and recommendations of the project

1. INTRODUCCIÓN

Durante los últimos años se ha hablado mucho acerca del desarrollo de edificios inteligentes, viviendas domóticas, ciudades inteligentes. Esto se debe al desarrollo continuo de las tecnologías de la comunicación (TIC), término que ha ido penetrando en todos los ámbitos de la vida. La aplicación del término domótica ha venido impulsándose por tres factores principales: evolución tecnológica, cambios sociales y oportunidades de negocios [1].

Lo cierto es que el mundo ha llegado a una etapa donde gran parte de las viviendas cuentan con instalaciones básicas como la electricidad, televisión e Internet las cuales podrían complementarse con los beneficios que brinda la domótica como, por ejemplo: la automatización y control de la iluminación, control de la climatización, control de acceso, automatización de las tareas domésticas, alarmas de seguridad contra intrusos, alarma de incendios [1].

Para conseguir todos estos beneficios una vivienda necesita tener componentes conectados entre sí los cuales se encargarán de recoger la información como por ejemplo los estados de puertas y ventanas (abierta o cerrada), procesarlos y actuar. Dotando de esta manera a la vivienda de cierta inteligencia para automatizar tareas por ende se puede afirmar que la vivienda domótica va tomando más fuerza al facilitar las tareas del usuario. En el anexo A-1 se presenta un breve resumen del concepto de domótica incluyendo sus características y componentes [1].

Para crear un sistema domótico de bajo costo se empleó: un microcomputador *Raspberry Pi 3*, el cual es capaz de cumplir con las características que implica un proyecto domótico. Se utilizó varios elementos electrónicos y eléctricos muy comunes en el mercado, los cuales permitan tener el control sobre la iluminación de la vivienda y a su vez tener un control de seguridad cuando el usuario se encuentre fuera de su domicilio.

Además, se encontró la necesidad de contar con un sistema que permita reducir la cifra de robos en domicilios, ya que éste es uno de los delitos más comunes en el país. Muchos de los robos residenciales se dan por no contar con un apropiado sistema de seguridad.

El actual proyecto cuenta con dos etapas las cuales son: el diseño y la implementación del sistema domótico. Al finalizar el desarrollo del proyecto se presentará el prototipo que cumple con los objetivos planteados.

1.1 Planteamiento del Problema

Uno de los objetivos del avance tecnológico y del desarrollo de hardware y software es facilitar el modo de vida de los seres humanos, reduciendo el grado de dificultad al momento de realizar sus labores diarias; como por ejemplo, controlar el sistema de iluminación de sus casas desde un solo sitio, sin que ésta demande un esfuerzo mayor.

Día tras día surgen nuevos avances tecnológicos, como por ejemplo: la creación de tarjetas inteligentes, las cuales en su gran mayoría son de tamaño muy reducido, trabajan inalámbricamente y por lo general funcionan con voltajes bajos. Estas ofrecen grandes recursos y utilidades que facilitan la construcción de sistemas domóticos, pudiendo así instalarse de una manera sencilla sin la necesidad de realizar un cableado específico y reduciendo el trabajo en el proceso de instalación.

La capacidad de las personas para crear un sitio Web con el objetivo de publicar o transmitir diversidad de contenido es muy común; uno de los nuevos desafíos consiste en crear sistemas de control que puedan ser administrados desde una página Web. Estos nuevos avances se logran con la ayuda de software que facilita la creación de páginas Web a través del uso de distintos tipos de lenguajes de programación.

Los robos residenciales se dan por no contar con un apropiado sistema de seguridad de puertas y ventanas o por no poseer un sistema inteligente de alarmas. El robo a domicilios es uno de los delitos más comunes en el país y uno de los que mayor conmoción social ocasiona en la comunidad.

En un estudio llamado “la inseguridad en el Ecuador”, publicado por la empresa CEDATOS, determinó que el robo a domicilios es uno de los principales problemas que aquejan a los ecuatorianos [2], Esto se podría prevenir utilizando sistemas de seguridad, pero debido a los altos costos, éstos no son asequibles para el segmento residencial; sin embargo, son una solución que presenta muchas ventajas.

1.2 Justificación

En años recientes, los sistemas domóticos han asumido un papel cada vez más importante en el desarrollo y avances de la civilización moderna y la tecnología. Prácticamente, cada tarea de nuestra vida está afectada por algún tipo de automatización facilitando de esta manera las labores diarias [3].

Es vital crear sistemas de control que puedan ser administrados desde un celular o una laptop, ya que en la actualidad gran parte de las personas cuentan con un equipo con

acceso a Internet permitiendo de esta manera tener el control total de una vivienda desde cualquier lugar del mundo, estos nuevos avances se logran con la ayuda de software y hardware, todo esto lleva a mejorar el nivel de confort de las personas y aumentar la seguridad de la vivienda.

Algunas de las actividades más usuales que desarrolla un sistema domótico son: control automático de iluminación, seguridad anti-intrusos [4].

1.3 Marco Teórico

• Análisis del microcomputador Raspberry Pi 3 y sus versiones

El microcomputador *Raspberry Pi 3* es un ordenador de placa reducida de un tamaño muy diminuto aproximadamente de 85x54mm, sus componentes internos y externos varían de acuerdo con el modelo [5]. Para que el microcomputador *Raspberry Pi 3* funcione, se necesita contar con los siguientes elementos:

- Tarjeta de memoria SD de 8 GB en adelante y clase 4.
- Alimentador con conector micro-USB 5V, 1.2 A. como mínimo.
- Monitor o TV con conexión HDMI o DVI.
- Cable HDMI 1.3a o HDMI-DVI.
- Teclado y mouse con cable USB 2.0.
- Cable red ethernet RJ-45 para proporcionar acceso a Internet.

Una vez estudiada cada una de las versiones que se presenta en el anexo A-2, se determinó que todos los microcomputadores Raspberry tienen el mismo fin que es la automatización de las cosas. El microcomputador *Raspberry Pi 3* sobresale de la *Raspberry Pi2*, en primer lugar, por contar con un procesador mucho más rápido y potente que las versiones anteriores, el cual es un ARM C rtex A53 de 4 n cleos a 1.2 GHz y de 64 bits. Adem s, su rendimiento es al menos un 50% superior al microcomputador *Raspberry Pi2*.

Otra de las ventajas se encuentra en la conectividad Wifi inal mbrica y Bluetooth, los anteriores modelos s lo pod an conectarse a trav s de un cable ethernet o bien mediante adaptadores inal mbricos USB, por lo que esta versi n facilita las cosas y por esta raz n se decidi  utilizar en el presente proyecto. El microcomputador *Raspberry Pi 3* incluye as  un m dulo Bluetooth 4.1 y una tarjeta de red Wifi inal mbrica 802.11n.

A continuaci n, se elabor  un cuadro comparativo y un an lisis entre las versiones del microcomputador [6] el cual se presenta a continuaci n:

Tabla 1.1 Versiones del microcomputador Raspberry Pi [37]

Todos los modelos de Raspberry Pi	Raspberry Pi A	Raspberry Pi A+	Raspberry Pi B	Raspberry Pi B+	Raspberry Pi 2 B	Raspberry Pi 3 B
Tipo de chip	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2836	Broadcom BCM 2837
CPU	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM1176JZF-S a 700 MHz	ARM Cortex-A7 cuatro núcleos a 900 MHz	1.2Ghz QUAD ARM CORTEX-A53
Procesamiento Grafico (GPU)	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz
Memoria RAM	256 MB a 400 Mhz	256 MB a 400 Mhz	512 MB a 400 Mhz	512 MB a 400 Mhz	1 GB a 450 Mhz	1Gb
Entradas de vídeo	Cámara CSI	Cámara CSI	Cámara CSI	Cámara CSI	Cámara CSI	Cámara CSI
Salidas de vídeo	HDMI 1.4, conector RCA	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS	Jack HDMI
Salidas de audio	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI
Conectores USB 2.0	1	1	2	4	4	4
Tarjetas de almacenamiento	SD	microSD	SD	microSD	microSD	microSD
Conexión a red	No	No	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg WIFI BLUETOOTH
GPIO	8	17	8	17	17	17
Tamaño	85.6 x 56.5 mm	65 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm
Peso	45 gramos	23 gramos	45 gramos	45 gramos	45 gramos	45 gramos
Consumo	1.5W/5V	1W/5V	3.5W/5V	3W/5V	5V	2.5 A /12.5 W / 5V
Precio	25	20	30 16	35	35	35

La razón más importante y por la que sobresale el microcomputador *Raspberry Pi 3* de los demás microcomputadores disponibles en el mercado se debe a la variedad de pines GPIO que posee. En el anexo A-4 se presenta una definición de los puertos GPIO.

- **Entradas y salidas (GPIO) del microcomputador Raspberry Pi 3**

Es importante y necesario conocer la distribución y función de los pines pues es la base del presente proyecto, los mismos que se encuentran en el borde superior de la placa como se puede apreciar en la figura 1.1.



Fig. 1.1 Ubicación de los pines en la placa

El microcomputador *Raspberry Pi 3* tiene 40 pines distribuidos de la siguiente manera:

Tabla 1.2 Distribución de los pines del microcomputador *Raspberry Pi 3*

Función	Número de pines
Entradas / Salidas	26
Pines de alimentación y tierra	12
ID EEPROM	2
Total	40

Es necesario entender como referirse a los pines del microcomputador *Raspberry Pi 3* para poder programarlo; existen dos maneras, las cuales se explican a continuación:

- **Numeración GPIO:** esta numeración es la que entiende el microcomputador mientras que para el ser humano esta numeración no tiene ningún sentido ya que no sigue un orden.
- **Numeración física:** es una manera fácil de referirse a los pines, simplemente se debe contar de izquierda a derecha en forma ordenada.

En la figura 1.2 se muestra la distribución de cada uno de los pines tanto en la numeración física como la numeración GPIO, esta última será utilizada al momento de crear los códigos

de programación en *Php* y *Python*. Los cuadros en color rojo representan los pines en numeración GPIO, mientras que el cuadro de color amarillo representa la numeración física.

```

pi@MarcoCaizaE:~$ gpio readall

```

GPIO					FISICO		GPIO				
BCM	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	BCM
		3.3v			1	2			5v		
2	8	SDA.1	IN	1	3	4			5v		
3	9	SCL.1	IN	1	5	6			0v		
4	7	GPIO. 7	IN	1	7	8	0	IN	TxD	15	14
		0v			9	10	1	IN	RxD	16	15
17	0	GPIO. 0	OUT	1	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14			0v		
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20			0v		
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30			0v		
6	22	GPIO.22	IN	0	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34			0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21

Fig. 1.2 Numeración GPIO y numeración física

Todo ordenador necesita un sistema operativo y es por ello por lo que se adjunta en el anexo A-5 una variedad de sistemas operativos compatibles con Raspberry. Una parte importante del microcomputador *Raspberry Pi* es el sistema operativo que deberá ser instalado. El proceso de instalación es diferente al seguido en otros ordenadores, ya que éste está instalado en una tarjeta SD, lo que facilita cambiar de sistema operativo tan solo con cambiar de tarjeta SD, básicamente se tiene que escribir la imagen ISO sobre una tarjeta de memoria y encender el dispositivo para seguir con el proceso de instalación.

Para éste proyecto se escogió *Raspbian* como sistema operativo por la velocidad en el rendimiento y por ser un sistema operativo para trabajo a diferencia de otros sistemas operativos que son diseñados para diversión (ocio). Adicionalmente, *Raspbian* es considerado como el sistema operativo oficial de *Raspberry Pi*.

A continuación, se detalla los pasos para instalar el sistema operativo y las configuraciones básicas para el correcto funcionamiento del microcomputador *Raspberry Pi 3*.

- **Instalación y configuración del sistema operativo Raspbian en el microcomputador Raspberry Pi 3**

Para iniciar con la instalación se necesita contar con los siguientes elementos:

- Una tarjeta SD de 4 Gb o de mayor capacidad.
- La imagen ISO del sistema operativo *Raspbian*.
- El programa Win32 Imager (Anexo A – 6.a)

Primero se necesita descargar la imagen ISO perteneciente a *Raspbian* Jessie Lite, la cual se encuentra disponible de forma gratuita en la página Web de Raspberry (<https://www.raspberrypi.org/downloads/raspbian/>) y continuar con los siguientes pasos:

- Descargar e instalar en el computador el programa Win32 Disk Imager, se puede descargar del siguiente enlace: <https://sourceforge.net/projects/win32diskimager/>, se descargará un archivo en formato ZIP dentro del cual se encuentra el archivo .img, el cual hay que extraerlo.
- Insertar la tarjeta SD en el computador.
- Ahora se procede a cargar el sistema operativo en la tarjeta de memoria SD, se debe abrir el programa win32 Disk Imager como se muestra en la figura 1.3. En el ícono de la carpeta se debe buscar el archivo 2017-03-02-raspbian-jessie-lite.img (ISO) y en el ícono device se debe escoger la unidad en donde se encuentre la tarjeta SD.
- Presionar el botón “Write”.
- Colocar la tarjeta micro SD en el microcomputador *Raspberry Pi 3*.
- Conectar un teclado y un monitor o televisor con entrada HDMI.
- Una vez insertada la tarjeta SD se procede a conectar la alimentación de energía eléctrica (5VDC.), ya que el microcomputador Raspberry no posee un botón de encendido
- Al prender el microcomputador *Raspberry Pi 3* se solicitará ingresar un login y password por defecto la Raspberry viene configurada con el usuario: “pi” y el password: “raspberrry” [7].

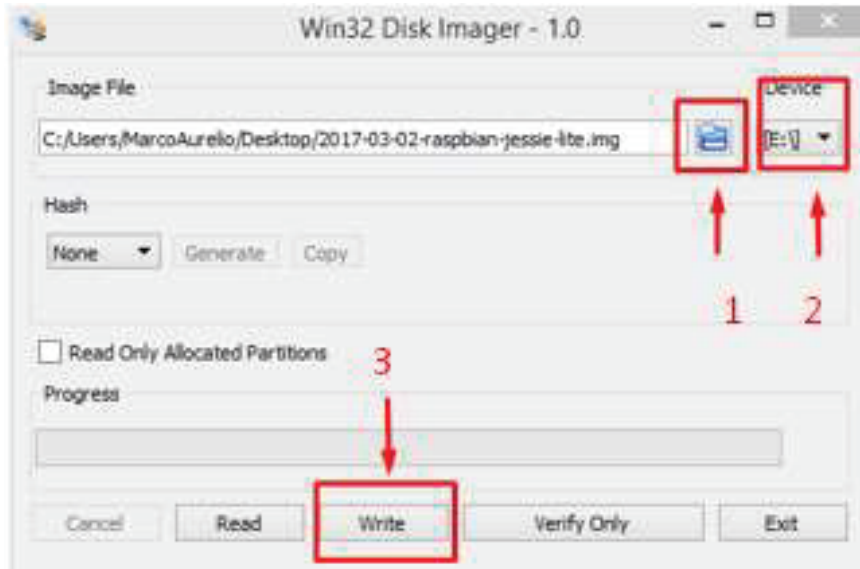


Fig. 1.3 Configuración para cargar Raspbian en la Raspberry

Luego del proceso de inicio del sistema operativo, se debe abrir la herramienta raspi-config, la cual se utiliza para la configuración de *Raspbian* a continuación se detalla las principales configuraciones.

- **Configuración de *Raspbian***

La herramienta raspi-config se utiliza para la configuración de *Raspbian*. Una vez que se haya encendido el microcomputador Raspberry mostrará la pantalla que se puede ver en la figura 1.4. Este proceso se lo puede realizar cuando el usuario desee, escribiendo en un terminal el comando *raspi-config*.

A continuación, se muestra en detalle y de manera resumida cada una de las opciones que ofrece la configuración de *Raspbian* y que deben ser modificadas obligatoriamente:

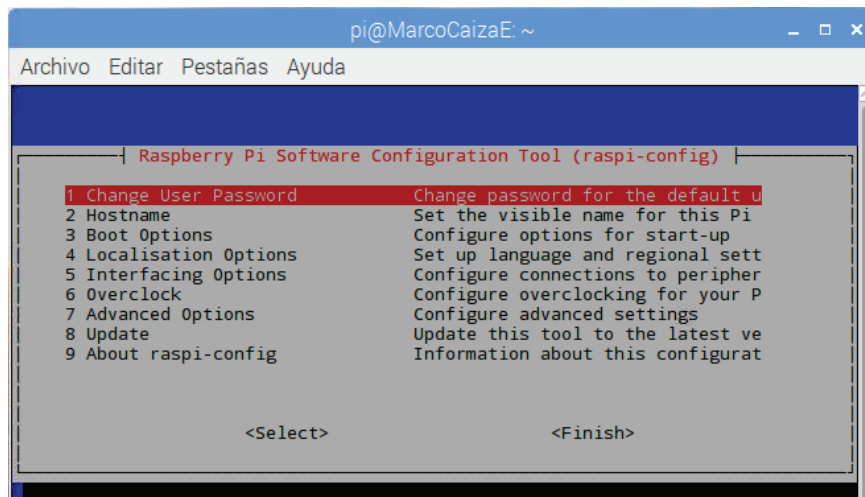


Fig. 1.4 Menú que ofrece comando raspi-config

- **Change user password:** por defecto el microcomputador *Raspberry Pi* trae la contraseña “raspberrypi”, la cual debe ser cambiada por una contraseña nueva.
- **Hostname:** en éste menú se puede cambiar el nombre asignado a la Raspberry.
- **Boot Options:** permite escoger la manera de iniciar el microcomputador *Raspberry Pi* en modo de terminal o gráfico, también permite elegir si desea que el microcomputador Raspberry solicite la contraseña al momento de inicializarse.
- **Localisation Options:** permite modificar la zona horaria, así como también permite configurar el idioma utilizado por el teclado.
- **Advanced options:** permite modificar opciones de SSH. Es recomendable activar esta opción ya que nos permite manejar el microcomputador Raspberry de manera remota sin la necesidad de utilizar elementos como pantallas, teclados ni mouses.
- **Interfacing option:** permite realizar la configuración del puerto ethernet y de la red Wifi.

Una vez finalizada la configuración el microcomputador *Raspberry Pi 3*, éste se reiniciará y luego pasará directamente a la interfaz gráfica, caso contrario es necesario escribir en un terminal el comando *startx*, el cual forzará al microcomputador *Raspberry Pi 3* a inicializarse con una interfaz gráfica.

Antes de comenzar a instalar el servidor Web, es necesario asignar una dirección IP fija al microcomputador *Raspberry Pi 3*.

- **Configuración de la Raspberry con una dirección IP fija**

La versión *Raspberry Pi 3* tiene un controlador Wifi integrado en la placa, por lo cual facilita la conexión a Internet. Para éste proyecto es necesario cambiar la configuración de la tarjeta de red ya que por defecto el microcomputador *Raspberry Pi 3* se conecta por DHCP, es decir el microcomputador Raspberry funcionará con una dirección IP asignada por el router inalámbrico, esta dirección IP no será fija cambiará constantemente lo que causa dificultad al momento de realizar la programación ya que no se sabrá la dirección IP que asignará el router Wifi al microcomputador *Raspberry Pi 3*. Para evitar éste problema, la solución es configurar la dirección IP del microcomputador *Raspberry Pi 3* de manera estática. A continuación, se detalla los pasos para dicho proceso:

Abrir una terminal y escribir el comando: `sudo nano /etc/network/interfaces` el cual nos dirige al documento de las configuraciones de red de la Raspberry. Enseguida aparecerá un documento editable como el que se muestra en la figura 1.5.

```
interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet static
    address 192.168.1.19
    netmask 255.255.255.192
    gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.1.20
    netmask 255.255.255.192
    gateway 192.168.1.1
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Fig. 1.5 Fichero interfaces modificado con las direcciones IP estáticas

El documento tiene que ser modificado con los siguientes datos:

Address: contiene la dirección IP estática que se quiere asignar al microcomputador.

Netmask: incluir aquí la máscara de subred (normalmente no tiene que ser cambiado).

Gateway: contiene la información de la puerta de enlace, se puede obtener esta información en cualquier PC conectado a la red, escribiendo `ifconfig` en la consola de Linux o `ipconfig` en la consola de *Windows*.

La modificación se tiene que realizar en la interfaz `Wlan0` cuando la conexión sea inalámbrica y en la interfaz `Eth0` cuando la conexión sea cableada. En éste proyecto se configuraron en las dos interfaces, lo cual permite trabajar independientemente de la existencia o no de una red Wifi.

También se debe configurar la contraseña de la red inalámbrica modificando el siguiente archivo: `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` desde un terminal e incluir la siguiente información (ver figura 1.6):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
    ssid="MARCO AKD"
    psk="puntodatos"
    key_mgmt=WPA-PSK
}
```

Nombre de la red inalámbrica

Contraseña

Fig. 1.6 Configuración de la red inalámbrica

Finalmente se debe reiniciar el microcomputador *Raspberry Pi 3*. Al inicializarse se habrá realizado el cambio en la tarjeta de red. Con esta configuración el microcomputador *Raspberry Pi 3* siempre que inicie tendrá la misma dirección IP. Para éste proyecto se configuró con la dirección IP fija 192.168.1.20/26 para cuando se conecte inalámbricamente y 192.168.2.19/26 en el caso de conectarse con cable por el puerto ethernet.

Una vez configurada la dirección IP estática, el segundo paso es la creación de las páginas Web, pero antes de empezar con el diseño de las mismas, es necesario instalar un servidor Web en el microcomputador *Raspberry Pi 3*. En el anexo A-7 se presenta el concepto de servidores.

La realización de tareas como: controlar la iluminación de la vivienda, verificar el estado de ventanas y puertas, activar una alarma de seguridad, pueden ser ejecutadas de manera remota con la ayuda de SSH, pero esta interfaz de consola es complicada de utilizar y mucho más si el usuario no tiene conocimientos de esta herramienta. Además, es muy molesto ya que se tendrían que escribir largos comandos en la consola es por esta razón que para éste proyecto se necesita tener una interfaz gráfica. Esta interfaz es implementada a través de páginas web.

Otra posible opción para cumplir con las tareas antes mencionadas es crear una aplicación para que se ejecute en cada sistema operativo existente (*Windows, MAC, Linux, Android, etc.*). Esto sería demasiado complicado ya que se requiere conocer varios lenguajes de programación, además también se tendría que desarrollar una aplicación que se ejecute en el microcomputador *Raspberry Pi 3*. Por tanto, optar por esa opción sería complicado y tomaría demasiado tiempo su implementación. Es por éste motivo que en éste proyecto se decidió utilizar un sitio Web el cual tiene que ser accedido usando un navegador Web, permitiendo así tener compatibilidad con cualquier dispositivo que cuente con un navegador. Además, solamente se necesita conocer 4 idiomas; HTML (contenido de la página Web), CSS (estilo de la página), *Php* (comunicación con el servidor) y *JavaScript* (integración con el usuario). Es importante destacar que el sitio web fue diseñado y optimizado para trabajar con Google Chrome Versión 62.0.3202.94.

A continuación, se detalla el proceso para la instalación del servidor Apache (Anexo A-8) en el microcomputador *Raspberry Pi 3* con la finalidad de alojar las páginas Web que se creen.

- **Configuración e instalación del servidor Web**

Ingresar a un terminal y digitar el siguiente comando: `sudo apt-get update`, el cual se va a encargar de actualizar *Raspbian* en caso de existir actualizaciones disponibles.

A continuación, ejecutar el comando: `sudo apt-get install apache2 php5 libapache2-mod-php5` inmediatamente comenzará a instalarse Apache. En el caso que el terminal muestre un error al ejecutarse el comando, puede deberse a que la Raspberry no tiene programas para descomprimir o instalar paquetes. Ejecutar el comando `sudo dpkg --configure -a` para solucionar el inconveniente.

Una vez instalado Apache se debe comprobar que el servidor se haya instalado de manera correcta, esto se logra ingresando la dirección IP de microcomputador *Raspberry Pi 3* en un navegador, si el servidor se instaló correctamente debe mostrar en la pantalla una página Web como la que se indica en la figura 1.7 caso contrario se debe volver a instalar el servidor.

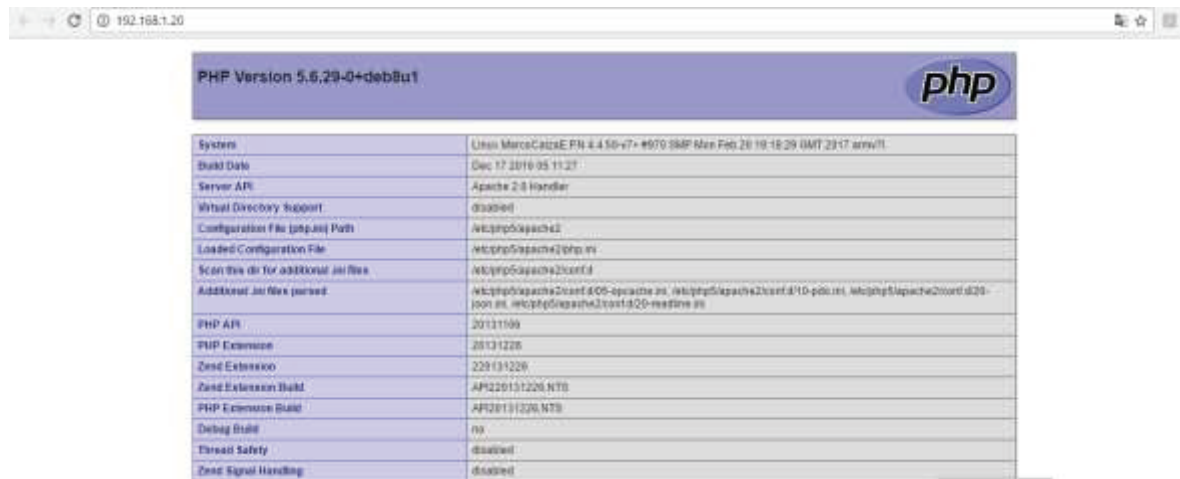


Fig. 1.7 Página web inicial de Apache

Finalmente, ya instalado apache hay que subir el archivo `index.php`, el cual es el script de la página Web que se mostrará por primera vez cuando se ingrese a dicha página. Este script está creado en HTML y *Php*.

Para subir el archivo `index` se utilizó el programa WinScp (Anexo A-6d), el cual brinda una interfaz gráfica que facilita la transferencia segura de archivos entre dos sistemas

informáticos. El archivo debe ser colocado en la carpeta VAR dentro de la cual existe la carpeta llamada WWW, finalmente al abrir la carpeta se encontrará la carpeta HTML en la cual se debe colocar el archivo index.php como se muestra en la figura 1.8.

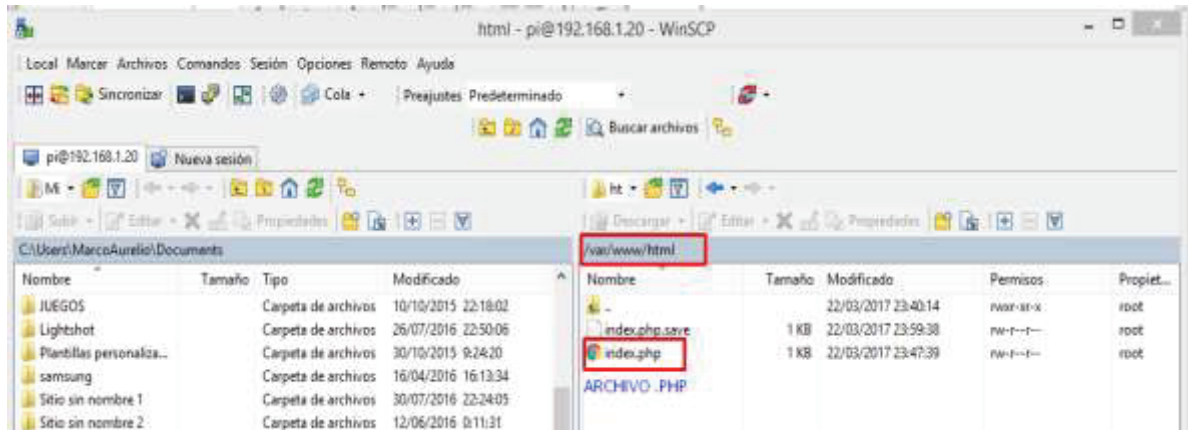


Fig. 1.8 Ubicación del archivo index.php utilizando WinScp

Una vez que se tenga instalado el sistema operativo el siguiente paso es la instalación de las librerías que se necesitan para el funcionamiento correcto del proyecto. Para el manejo de los pines de entradas y salidas de propósito general (GPIO) existen varias librerías que se pueden utilizar. A continuación, se nombran dos librerías que fueron ocupadas en el desarrollo del proyecto [8]:

- **WiringPi:** esta librería permite al usuario controlar los puertos GPIO desde un terminal, trabaja en lenguaje C.
- **Rpi.GPIO:** esta es una librería que trabaja en *Python* y no es adecuada para aplicaciones que requieran ser ejecutadas en tiempo real o que requieran respuestas de una manera rápida.

WiringPi incluye una biblioteca para el manejo de los puertos GPIO, estos comandos pueden ser usados dentro de scripts para manipular los pines GPIO (como entradas o salidas).

Para usar comandos GPIO no es necesario tener permisos root es decir no es necesario utilizar la función sudo ni ingresar como administrador, en el anexo A-9 se muestra varios comandos de la librería WiringPi.

• Instalación de la librería WiringPi

Primero descargar el archivo Snapshot [9] (ver la figura 1.9 cuadro rojo) que se encuentra disponible en la página Web : <https://git.drogon.net/?p=wiringPi;a=summary>

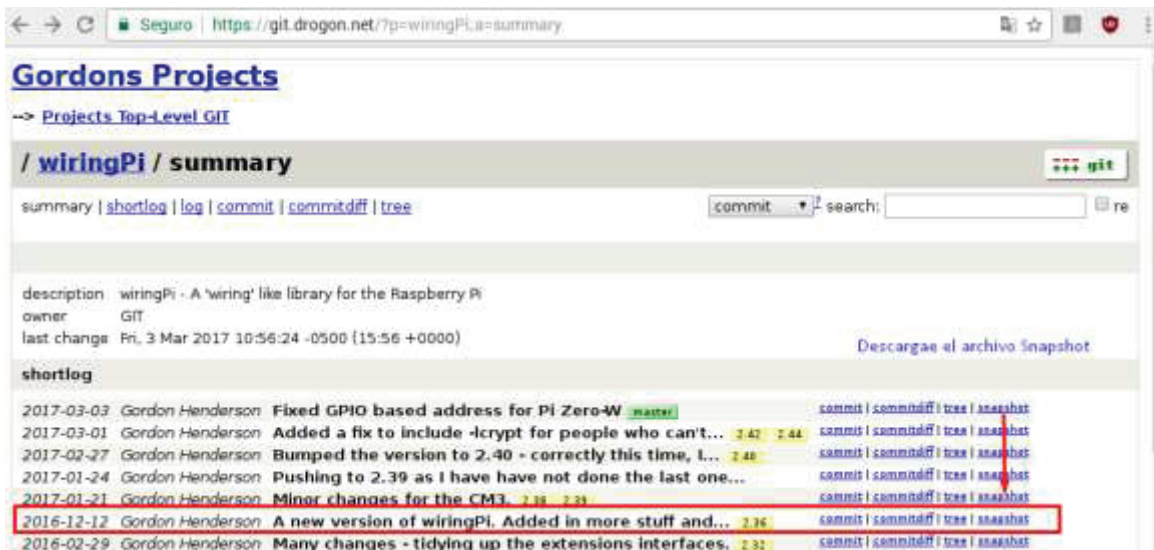


Fig. 1.9 Página para descargar WiringPi

A continuación, abrir un terminal y ubicarse en la dirección donde se descargó el archivo .tar.gz y escribir los siguientes comandos:

- **`cd /home/pi/Downloads`** (conduce a la ubicación de la descarga).
- **`tar xzf wiringPi-b1dfc18.tar.gz`** (se procede a descomprimir el archivo WiringPi).
- **`cd wiringPi-b1dfc18`** (éste comando permite entrar al documento).
- **`./build`** (para iniciar la instalación de WiringPi).

Una vez finalizada la instalación del servidor Apache y de las librerías necesarias se procede con el diseño de los diferentes sistemas.

2. METODOLOGÍA

Para el desarrollo del proyecto integrador se empleó principalmente dos tipos de investigación: aplicada y exploratoria. En la mayor parte del desarrollo se utilizó una investigación aplicada ya que al finalizar el proyecto se tiene como objetivo presentar un proyecto práctico donde se comprobará el cumplimiento de los objetivos planteados, para lo cual se utilizaron varios conocimientos adquiridos durante la permanencia en la Escuela Politécnica Nacional.

También se utilizó un tipo de investigación exploratoria ya que a pesar de existir un sinnúmero de proyectos domóticos, existen muy pocos sistemas que funcionan con microcomputadores *Raspberry Pi 3* por ser una versión nueva de Raspberry, para lo cual se examinó las características que trae el microcomputador *Raspberry Pi 3* y se tomó como guía varios proyectos desarrollados con versiones antiguas de *Raspberry Pi*.

Para poder entender el funcionamiento tanto del software, hardware y dispositivos eléctricos se utilizó el método de recolección de información. Se realizó el estudio de varias tesis y documentos digitales, así como tutoriales disponibles en el Internet.

Como primer punto se realizó un estudio de las características y mejoras que trae el microcomputador *Raspberry Pi 3* con respecto a sus versiones anteriores, los puntos a considerarse fueron su tamaño, tipo de procesador, alimentación, velocidades entre otros aspectos, los cuales ayudaron a determinar que el microcomputador *Raspberry Pi 3* tiene las características básicas requeridas para desarrollar un sistema domótico.

Después de estudiar las características del microcomputador *Raspberry Pi 3* se realizó un estudio para determinar la clase de servidor que será instalado. Se escogió Apache para montar un servidor Web, el cual es un software Open Source y con licencia GPL, que permita alojar la página Web que mostrará el estado de los focos distribuidos en la casa, además el estado de puertas y ventanas, y que también permita activar y desactivar el sistema de seguridad.

Se desarrolló el diseño de la página Web en *Php* para lo cual se utilizó el software *Dreamweaver*, mismo que permite la creación de páginas Web de una manera más amigable para el usuario.

Se verificó que es necesario crear una interfaz que permita la interacción entre la página Web y los puertos de entrada y salida del microcomputador *Raspberry Pi 3*, la cual se

desarrolló utilizando programación en lenguaje *JavaScript* y *Python* para la creación de varios scripts.

Finalmente, se realizaron varias pruebas en el protoboard para comprobar el funcionamiento del proyecto integrador con la finalidad de calibrar tiempos y corregir errores que se puedan presentar. Para una mejor visualización del prototipo se implementó una maqueta para lo cual fue necesario crear una placa para la transferencia del circuito previamente diseñado en el protoboard.

2.1 Funcionamiento del proyecto

A través de una red inalámbrica WLAN, se permitirá al ordenador del usuario acceder al microcomputador *Raspberry Pi 3*, el cual procesa las peticiones del usuario. Cualquier dispositivo con un navegador Web podrá controlar el estado de los focos, conocer el estado de puertas y ventanas y activar o desactivar el sistema de seguridad.

El sistema es capaz de controlar por medio de una página Web alojada en el microcomputador *Raspberry Pi 3*, 6 lámparas que están distribuidas en una maqueta, la cual simula la infraestructura de una vivienda de un piso que cuenta con 3 habitaciones y una sala. Además, se creó un sistema que ayuda a conocer el estado de ventanas y puertas, para ello se utilizó 5 sensores magnéticos. Adicionalmente se implementó un sistema de seguridad el mismo que se activa al detectar la presencia de intrusos, dicho sistema funciona de la siguiente manera: cuando se detecte un intruso el sistema tomará una foto cada tres segundos por tres ocasiones, estas fotos son enviadas a Twitter, además se activa una alarma sonora por un tiempo programado. Este sistema utiliza como elemento principal un sensor detector de movimiento (PIR).

Resumiendo, cuando el usuario ingrese a un navegador en su dispositivo móvil u ordenador de escritorio y digite la dirección IP del microcomputador *Raspberry Pi 3* (siempre y cuando esté dentro de la misma red WLAN), se cargará una página donde se solicita ingresar un usuario y contraseña, si la contraseña es correcta se mostrará una nueva página Web donde se podrá escoger cualquiera de las siguientes opciones:

- Control del sistema de iluminación.
- Control del estado de ventanas y puerta.
- Control del sistema de seguridad.

Si se selecciona la opción de control de luces aparecerá una nueva página Web en donde se muestran los estados de todos los focos de la casa, (será posible activar el cambio de estado de los focos); en el caso de seleccionar la opción para verificar el estado de ventanas y puertas se abrirá una nueva página donde se muestra el estado de las ventanas y de la puerta (Abierto / Cerrado). Finalmente, al escoger la opción de seguridad se mostrará una página Web con las opciones para activar o desactivar la alarma y también se podrá revisar el estado de la alarma.

3. RESULTADOS Y DISCUSIÓN

3.1 Descripción general del sistema

De acuerdo con los objetivos nombrados en la introducción, se diseñó el proyecto el cual cuenta con cuatro factores importantes para su correcto funcionamiento éstos son: el usuario, el sistema embebido, los dispositivos conectados en las entradas y salidas y el modem de Internet. En la figura 3.1 se muestra el esquema básico de funcionamiento del prototipo, en donde el modem de Internet es el encargado de autenticar al microcomputador *Raspberry Pi 3* para permitirle ingresar a la red, de esta manera el usuario puede ingresar a la página web almacenada en un servidor por medio de una dirección IP estática configurada en el microcomputador *Raspberry Pi 3*. Dicho servidor se encuentra instalado en el sistema embebido, una vez recibidas las peticiones el sistema embebido procesa las peticiones y realiza acciones que se verán reflejadas en los dispositivos conectados en los puertos de salida del sistema embebido.

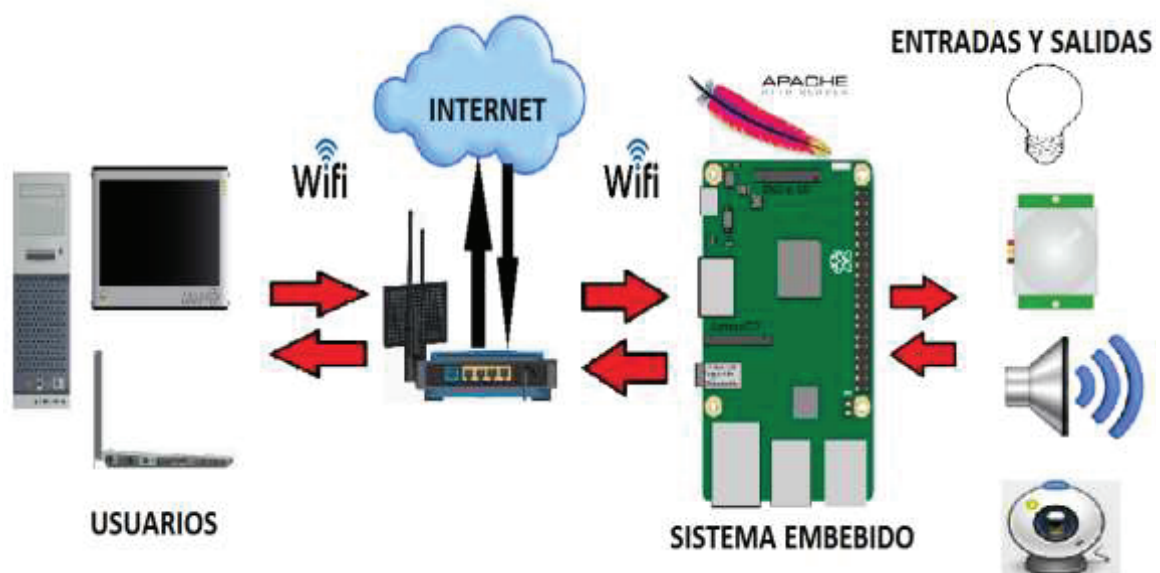


Fig. 3.1 Diagrama general del sistema doméstico

Una vez entendido el diagrama del sistema domótico se comienza con la implementación, el primer punto a tratar es el sistema embebido a utilizarse para lo cual, se analizó las versiones 1, 2 y 3 del microcomputador *Raspberry Pi*, para determinar la versión óptima a ser utilizada en el sistema domótico de la vivienda.

Para el desarrollo del control domótico se debe escoger la versión del microcomputador *Raspberry Pi* para la implementación de éste proyecto, por esta razón a continuación se presentan los factores que se tomaron en cuenta para la selección del microcomputador *Raspberry Pi 3*. En el anexo A-2 se detallan características de los diferentes modelos de placas *Raspberry Pi*.

3.2 Diseño de las conexiones del microcomputador Raspberry Pi 3

El microcomputador *Raspberry Pi 3* es el encargado de recibir las instrucciones y poner en marcha cualquier instrucción que sea solicitada por el usuario a través de la página Web, las tareas que se pueden realizar son:

- Control de iluminación de la vivienda.
- Control del estado de ventanas y puertas.
- Control del sistema de seguridad de la vivienda.

En la figura 3.2 se observa los pines de la Raspberry asignados a las diferentes funciones.

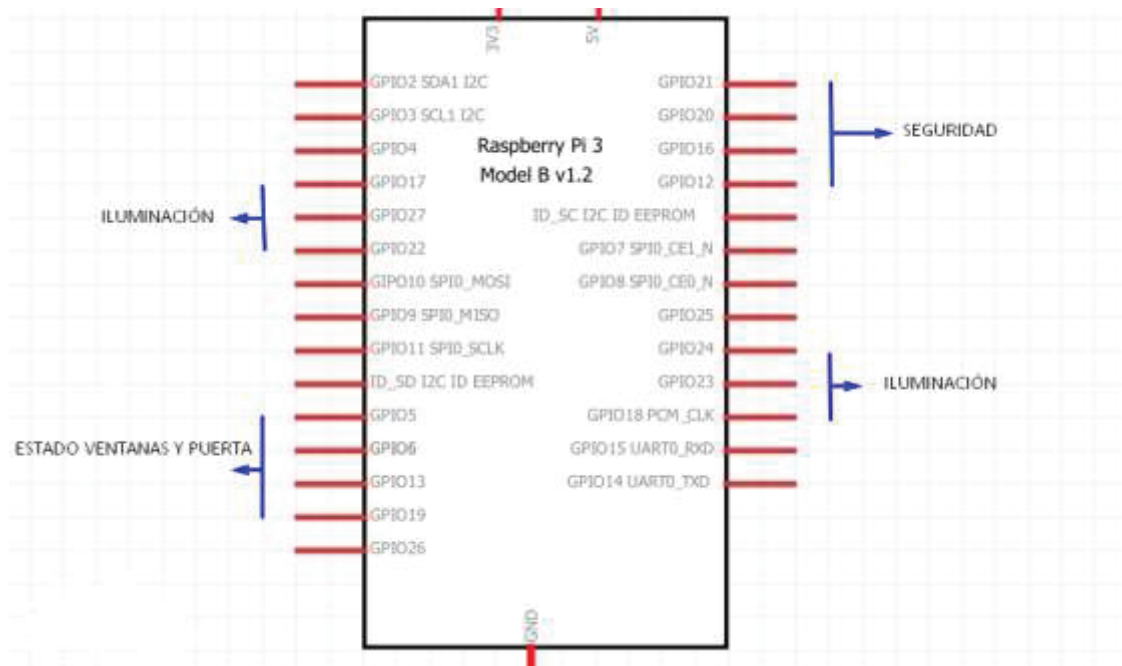


Fig. 3.2 Distribución de los pines acorde a las tareas a realizar

Una característica que debe cumplir un sistema domótico es la de brindar un alto nivel de seguridad, por lo cual se procedió a diseñar un sistema de acceso o login utilizando *Php*, HTML en el lado del servidor y un gestor de bases *MySQL* (en el anexo A-10 se muestra un pequeño resumen de los lenguajes de programación que se usó para éste proyecto).

A continuación, se muestran los pasos para el desarrollo de la página de acceso:

3.3 Página de acceso

El sistema de login está formado especialmente por tres archivos éstos son:

- index.php (Anexo A-11.a)
- página.php (Anexo A-11.b)
- usuario.sql (Anexo A-12.c)

Se utilizó HTML en el script index.php, el mismo que es el encargado del diseño de la página Web, mientras que el código en *Php* es el encargado de realizar la comunicación con la base de datos donde se encuentran creados los usuarios con sus respectivas contraseñas. En la figura 3.3 se observa la interfaz gráfica que se presenta al usuario.



Fig. 3.3 Página de login

Una vez validada la contraseña si esta es correcta se ejecuta el script página.php, el mismo que contiene la segunda página Web (ver figura 3.4) donde se puede escoger la acción que se desee realizar en el sistema domótico.

La página Web se realizó usando el programa *Dreamweaver* (Anexo A-6.b), esta página cuenta con una parte de programación en *Php* que es la encargada de la conexión con la base de datos.



Fig. 3.4 Página principal sistema domótico

Para el uso de la base de datos se instaló un gestor de bases de datos como *MySQL*.

- **Instalación y configuración de MySQL y PHPMyAdmin**

Una vez instalado Apache y *php*, se procede a instalar *MySQL* y *PHPMyAdmin*. El primer paso es activar la interfaz loopback, ya que si no lo hacemos dará un error al instalar *MySQL*, para esto ejecutar el comando:

```
sudo ifup lo
```

Luego se procede a instalar *MySQL* y *PHPMyAdmin*, escribiendo el siguiente comando:

```
sudo apt-get install mysql-server mysql-client php5-mysql phpmyadmin
```

Al instalarse aparecerá una pantalla de configuración preguntando la contraseña que se desea usar, esta contraseña será solicitada al momento de crear un nuevo usuario. Tras esto se accede al archivo *php.ini* introduciendo el siguiente comando:

```
sudo nano /etc/php5/apache2/php.ini
```

Dentro del archivo se debe dirigir hasta la línea "Dynamics Extensions" e ingresar lo siguiente:

```
extension=mysql.so
```

El archivo editado debe quedar de la siguiente manera (ver figura 3.5):


```
#####
; Dynamic Extensions ;
#####

; If you wish to have an extension loaded automatically, use the following
; syntax:
;
; extension=module_name.extension
;
; For example, on Windows:
;
; extension=msql.dll
;
; ... or under UNIX:
;
; extension=msql.so
;
; ... or with a path:
;
; extension=/path/to/extension/msql.so
;
; If you only provide the name of the extension, PHP will look for it in its
; default extension directory.
;
```

Fig. 3.5 Archivo php.ini instalación MySQL

Tras esto escribir el siguiente comando en un terminal:

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
```

Posteriormente ingresar al archivo apache2.conf (nano /etc/apache2/apache2.conf) y al final del archivo colocar la siguiente línea:

```
include /etc/phpmyadmin/apache.conf
```

El archivo editado quedará de la siguiente manera (ver figura 3.6):

```
GNU nano 2.9.6 Fichero: /etc/apache2/apache2.conf

requests.

Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
Use mod_remoteip instead.

logFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost$
logFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
logFormat "%h %l %u %t \"%r\" %>s %O" common
logFormat "%{Referer}i -> %U" referer
logFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
includeOptional conf-enabled/*.conf

# Include the virtual host configurations:
includeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
include /etc/phpmyadmin/apache.conf
```

Fig. 3.6 Archivo apache.config Instalación MySQL

Finalmente reiniciar el servidor Apache con el siguiente comando:

`/etc/init.d/apache2 restart`

Para verificar si se instaló correctamente ingresar en el navegador la dirección: `http://ipraspberry/phpmyadmin` y deberá mostrar la siguiente página (ver figura 3.7), caso contrario repetir el procedimiento.

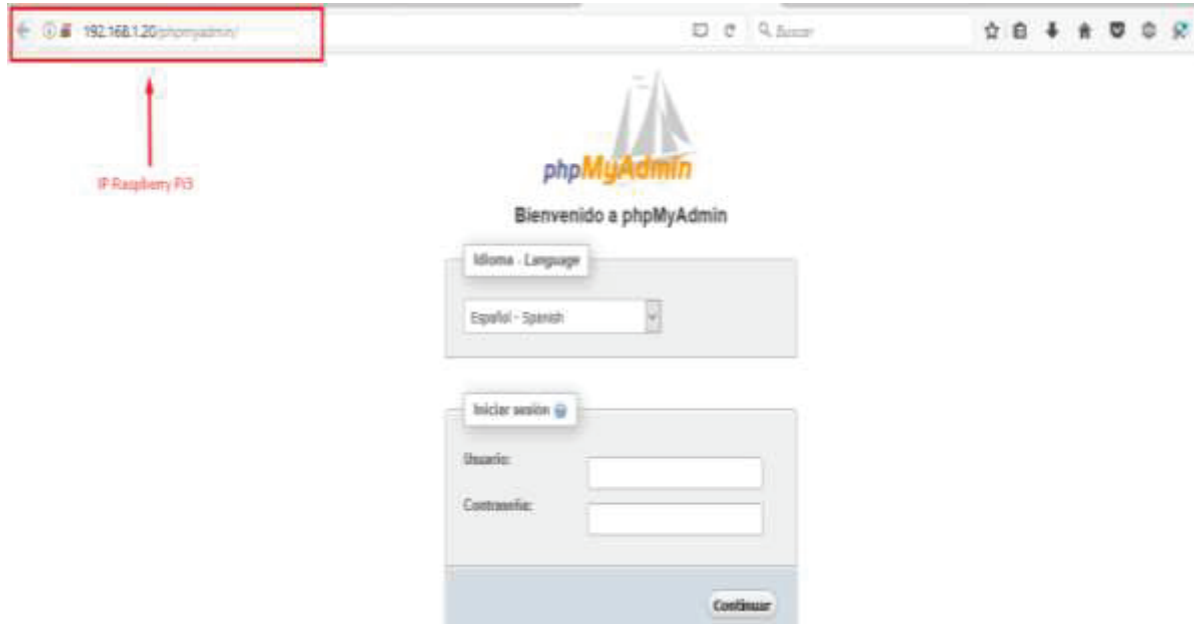


Fig. 3.7 Página principal MySQL

Para indicar con mayor facilidad la estructura del sistema domótico se le dividió en tres sub temas estos son:

- Control de iluminación de la vivienda. (Apartado 3.4)
- Control del estado de ventanas y puertas. (Apartado 3.5)
- Sistema de seguridad de la vivienda. (Apartado 3.6)

En cada subtema se indica el diagrama electrónico, el diseño de la interfaz, y su respectivo diagrama de flujo.

Hay que tener en cuenta que para éste proyecto se utilizaron varios archivos llamados `index.php` y `gpio.php`, los cuales son totalmente diferentes. Se realiza esta aclaración para que no exista confusión al momento de realizar el proyecto.

3.4 Diseño del control de iluminación

El sistema de iluminación está controlado mediante la página Web desde la cual se podrá encender y apagar las bombillas distribuidas en la vivienda, además es posible conocer el estado de la iluminación. El circuito electrónico implementado se observa en la figura 3.8.

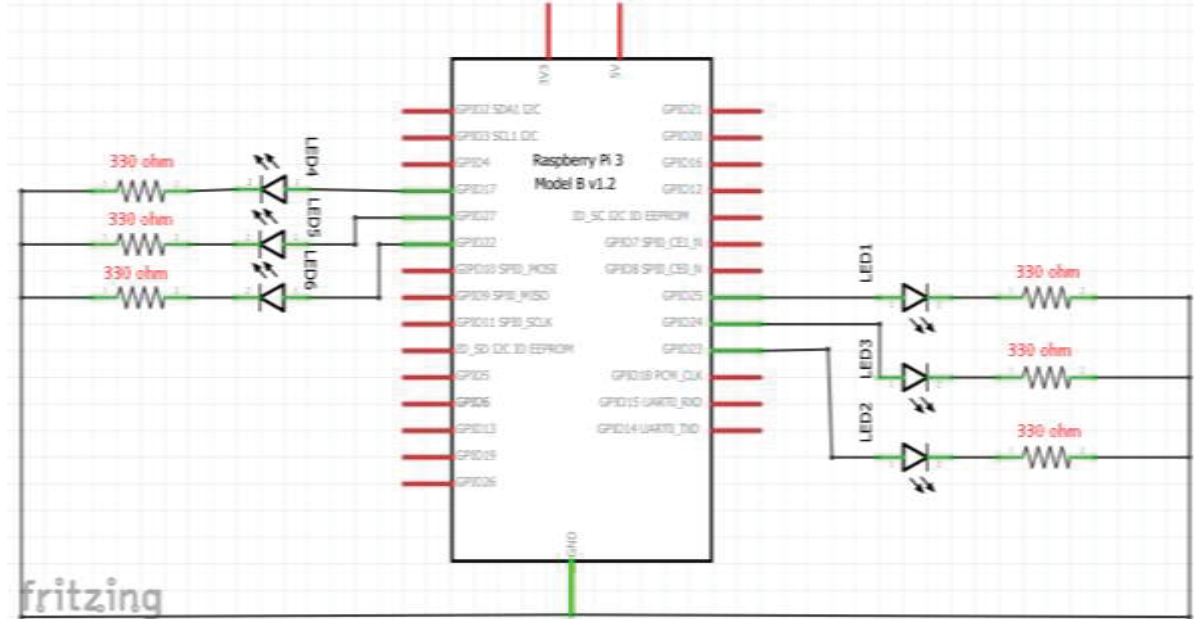


Fig. 3.8 Circuito electrónico para el control de iluminación

Una vez comprobado el funcionamiento del circuito electrónico en corriente directa (DC) se realizó el acoplamiento a corriente alterna (AC), para lo cual se utilizó un módulo de relés de 8 canales. Estos módulos son construidos con 8 opto acopladores para aislar la fuente de 5 voltios que utiliza el relé, de la fuente de 120 voltios que utilizan las lámparas, evitando de esta manera señales de ruido en el circuito.

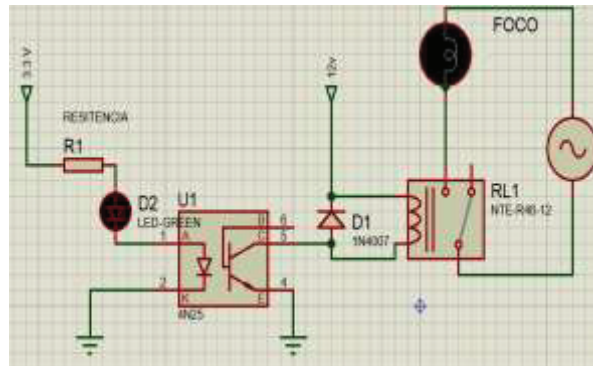


Fig. 3.9 Circuito de acoplamiento AC-DC

En la figura 3.9 se muestra el diagrama electrónico interno de un canal del relé, para mayor facilidad y comodidad se utilizó un módulo de 8 canales. Las características del módulo se presentan en el anexo C-4.

- **Diseño de la interfaz del control de iluminación**

Se pueden controlar los puertos de la *Raspberry PI* con sencillos scripts realizados en *Php*, pero no se tendría una interacción total con el usuario, es decir el usuario no tendría como elegir si quiere prender o apagar la iluminación por lo tanto es necesario crear una interfaz.

La página Web es la interfaz que permite al usuario comunicarse con el sistema domótico, esta página Web permite prender o apagar las bombillas cuando el usuario pulse un botón digital.

Se utilizó *JavaScript* para la interacción entre el usuario y la página Web. Al ser *JavaScript* un lenguaje de programación del lado del usuario éste estará ejecutándose constantemente en el navegador de Internet, gracias a esto se logra que la página Web actualice la información sin la necesidad de hacerlo manualmente para mostrar el estado de las lámparas instaladas (prendido/apagado).

Para el funcionamiento del control de iluminación se procedió a diseñar tres scripts, los cuales trabajan en conjunto. En la figura 3.10 se observa un diagrama de bloques del funcionamiento e interacción de los 3 scripts (*Index.php*, *gpio.php*, *script.js*).

En el anexo A-12.d se muestra el diagrama de flujo del sistema de iluminación.

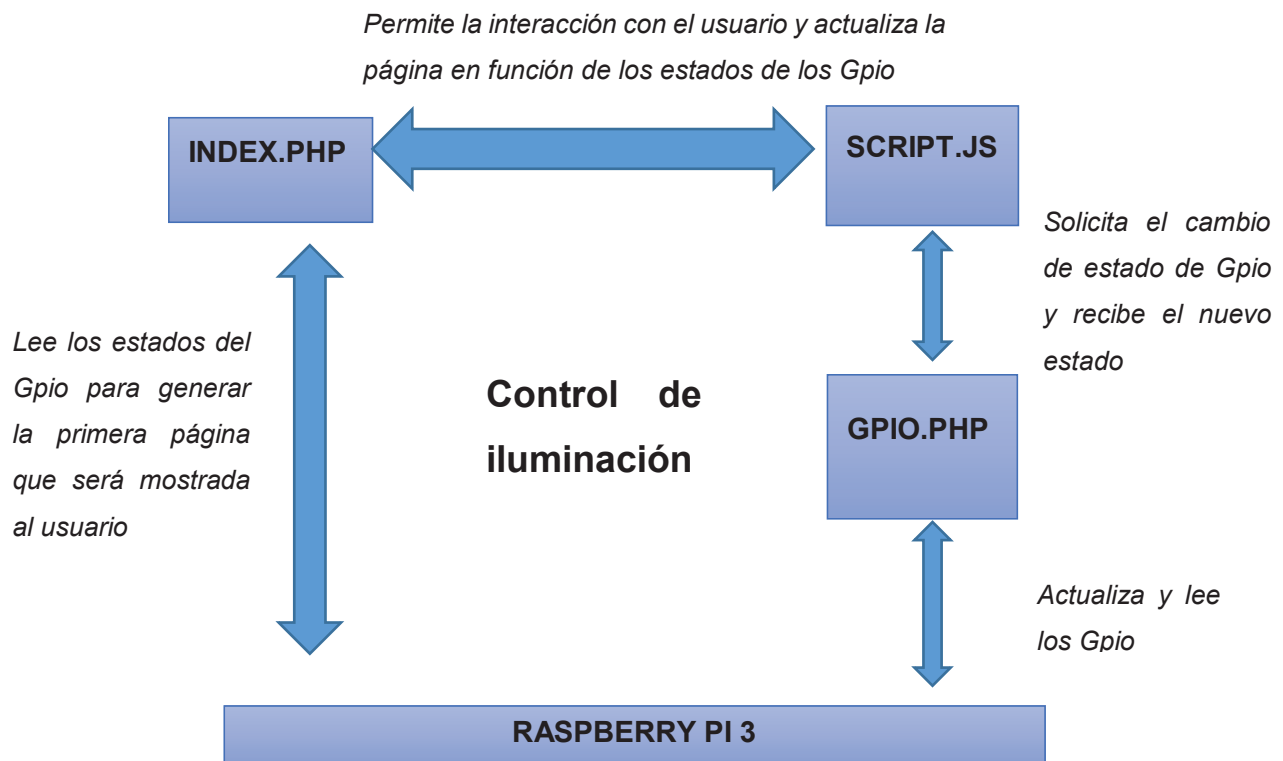


Fig. 3.10 Diagrama de funcionamiento del control de iluminación

A continuación, se presenta la descripción de cada uno de los scripts:

INDEX.PHP

El archivo index es fundamental cuando se desea crear un sitio Web, es decir éste será el encargado de indicar la primera página que se mostrará al usuario. Cuando el usuario ingrese un URL en la barra de direcciones del navegador lo primero que busca es el archivo index para mostrar su contenido (imágenes, texto, archivos, etc.).

El archivo Index.php del presente proyecto es el encargado de mostrar 12 imágenes que serán las encargadas de simular los botones para encender o apagar la iluminación de la casa, en la figura 3.11 se puede observar 6 imágenes, las mismas que cambian de estado cuando se presione sobre una de ellas. Todo esto permite al usuario conocer qué lugar de la casa se encuentra con iluminación encendida.

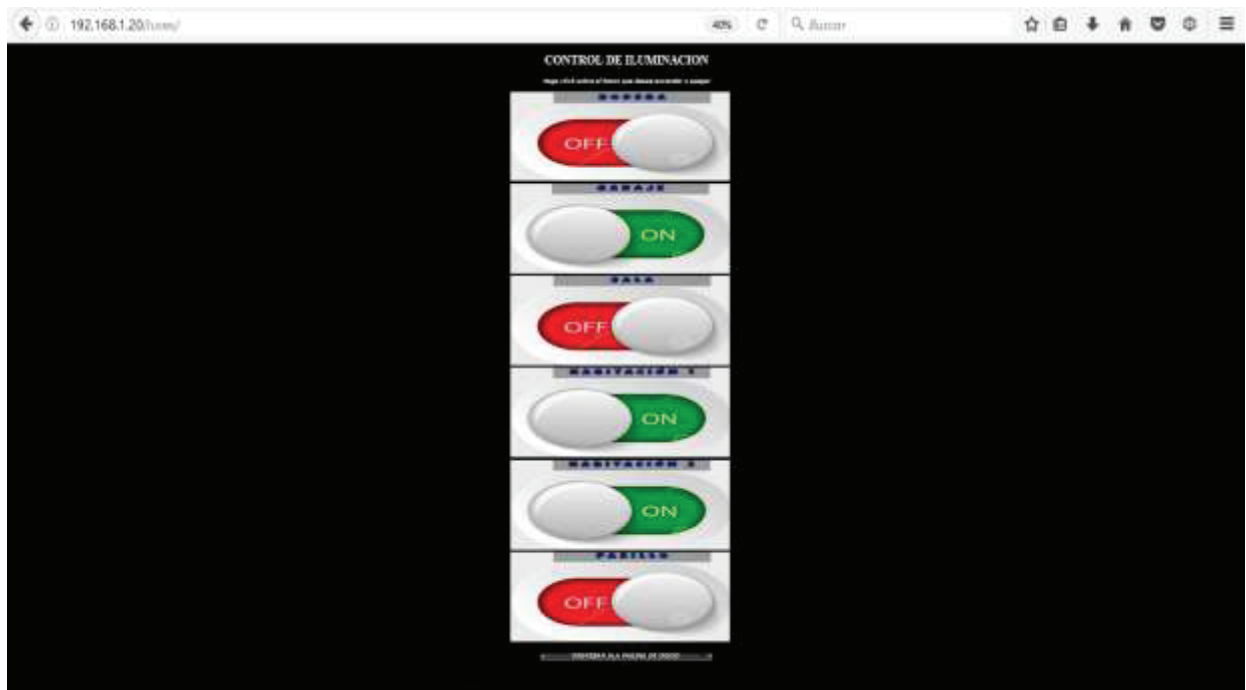


Fig. 3.11 Página Web del control de iluminación de la vivienda

En el código se resalta dos comandos importantes que se explican a continuación y en el anexo A-12.a se muestra el código completo implementado.

➤ **Onclick**

Este atributo se utiliza en páginas Web que incluyen código *JavaScript* para realizar una acción dinámica sobre algún elemento de la página Web. En éste caso se utilizó para detectar cuando el usuario presione sobre la imagen que simula un botón de encendido o apagado para ejecutar una tarea (encender o apagar la iluminación de un área seleccionada) [10].

➤ **Img src=**

Especifica la dirección o ubicación de la imagen que se va a mostrar al usuario a través de una página Web [10]. Se utilizó para indicar la ubicación de los botones de cada uno de los espacios de la vivienda tanto de encendido como de apagado, esta dirección es utilizada por el navegador para mostrar la imagen al usuario en el momento que se cargue la página. El fragmento del código donde se utilizó estos comandos es:

```
echo("<img id='button_".$.i."' src='data/img/red/red_".$.i.".jpg' onclick='cambio_pin (\"$.i.\");'/>");
```

GPIO.PHP

El archivo gpio.php es el encargado de ejecutar una acción sobre el microcomputador *Raspberry Pi 3*, es decir es el encargado de activar y desactivar los puertos de propósito general GPIO. Además de configurar los pines del microcomputador *Raspberry Pi 3* en modo de lectura (entrada) o escritura (salida) cuando reciba la orden proveniente del archivo script.js.

En el código gpio.php se destacan dos comandos que se explican a continuación y en el anexo A-12. b se puede observar el código completo implementado.

➤ **isset ()**

Se utiliza para comprobar si una variable existe y no es nula, Este comando es utilizado para recibir la variable [pic] del archivo script.js

➤ **Strip_tags ()**

La función strip_tags() es una de las más importantes en el lenguaje *PHP* se utiliza para eliminar las etiquetas HTML y *PHP* de un texto y convertirlas en cadenas simples, utilizado para extraer datos de otros archivos sin las etiquetas [11].

El fragmento del código donde se utilizó estos comandos es:

```
if (isset ( $_GET["pic"] )) {  
    $pic = strip_tags ( $_GET["pic"]);
```

SCRIPT.JS

El archivo Script.js es el encargado de mantener la página Web actualizada y también de recibir las ordenes provenientes del usuario, las cuales serán enviadas al archivo gpio.php para su ejecución.

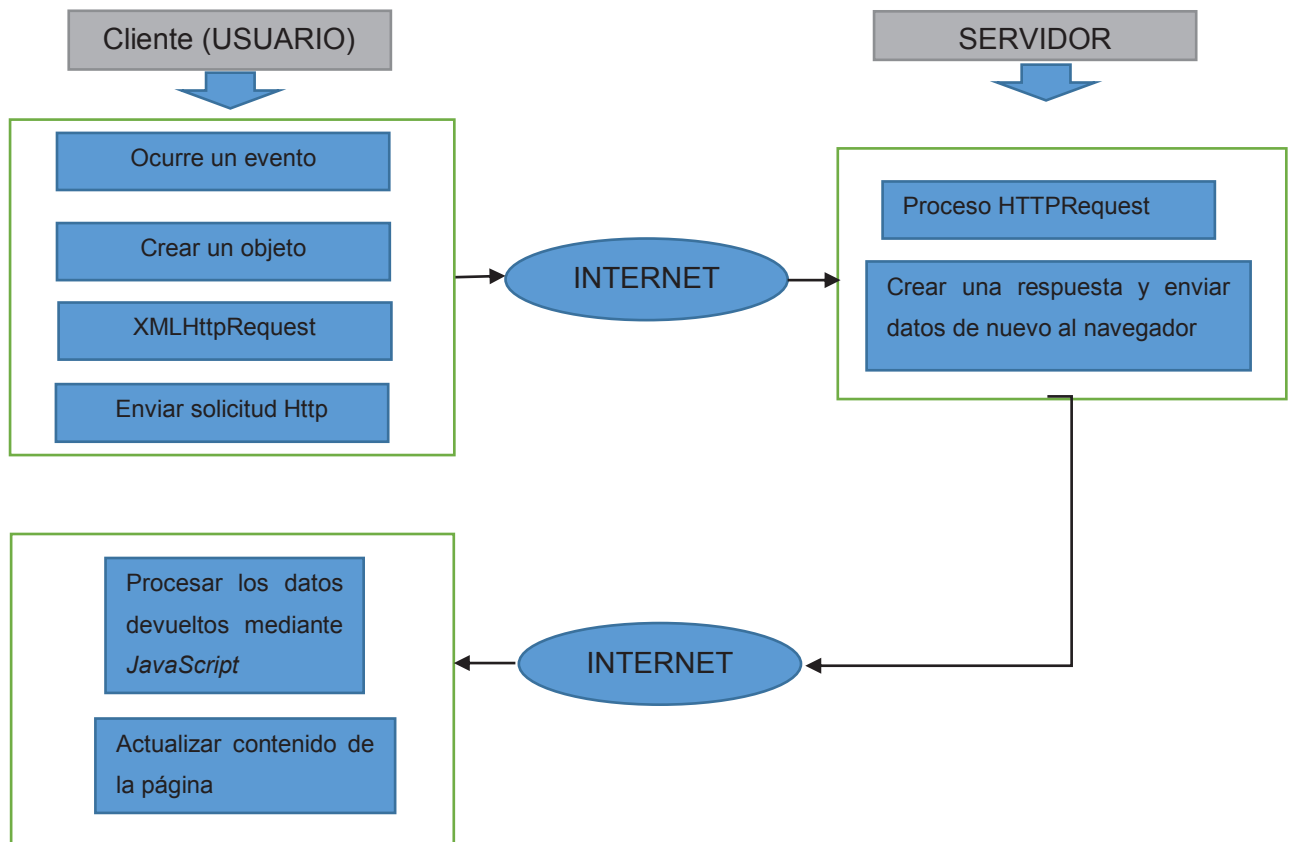
En el código se resalta dos comandos importantes que se explican a continuación y en el anexo A-12.c se muestra el código completo implementado.

➤ XMLHttpRequest

El objeto XMLHttpRequest es utilizado para solicitar datos desde un servidor, se podría decir que equivale a abrir un canal de comunicación con el servidor. Este objeto viene incorporado en los navegadores modernos y permite realizar las siguientes tareas:

- Actualizar la página Web sin la necesidad de actualizar manualmente la página.
- Solicitar datos desde un servidor.
- Recibir datos de un servidor.
- Enviar datos a un servidor.

El funcionamiento es el siguiente [12]:



1. Ocurre un evento en una página Web (realizar un clic sobre una imagen para encender un foco).
2. EL objeto XMLHttpRequest es creado por *JavaScript*.
3. El objeto XMLHttpRequest envía una solicitud a un servidor Web.
4. El servidor recibe y procesa la solicitud.
5. El servidor envía una respuesta a la página Web mediante el Internet.
6. La respuesta es leída por *JavaScript*.
7. La acción solicitada es ejecutada por *JavaScript* (prende un foco).

La sintaxis común en *JavaScript* para XMLHttpRequest es la siguiente:

```

1 var request = new XMLHttpRequest();
2   request.open( "GET" , "gpio.php?pic=" + pic, true);
3   request.send(null);
4   request.onreadystatechange = function () {
5     if (request.readyState == 4 && request.status == 200) {
6

```

A continuación, se explica cada una de las líneas

Línea número 1

Se crea un objeto XMLHttpRequest denominado request.

Línea número 2

Especifica el tipo de solicitud que se enviará al servidor, esta puede ser OPEN o SEND.

```
request.open( "GET" , "gpio.php?pic=" + pic, true);
```

Tabla 3.1 Partes del objeto open y send

METODO	DESCRIPCION
OPEN (MÉTODO, URL, ASYNC)	MÉTODO: Tipo de solicitud GET o POST url: Ubicación del archivo en el servidor async: True: asincrónico / false: sincrónico
SEND ()	Envía la solicitud al servidor utilizando el método GET

A continuación, se muestra diferencias entre la función GET y POST

Función GET y POST

```
request.open( "GET" , "gpio.php?pic=" + pic, true);
```

El método GET es más simple y rápido que el método POST [12]; pero se usan peticiones POST en los siguientes casos:

- Cuando se necesita actualizar un archivo o una base de datos en un servidor.
- Cuando se necesita enviar una gran cantidad de datos al servidor (la petición POST no tiene una limitación en el tamaño del archivo).
- POST es más seguro que GET.

Línea número 3

Indica que no se está enviando nada en el cuerpo de la solicitud HTTP, ya que se está proporcionando los datos en la URL, pero si se utiliza POST los datos deben colocarse en el cuerpo y ya no es necesario usar una URL.

Línea número 4

Indica que será ejecutada la función cuando cambie la propiedad readyState del objeto. La función Onreadystatechange será la encargada de definir dicha función.

Línea número 5

Maneja dos parámetros: readyState y status.

➤ ReadyState:

Contiene el valor numérico entero que representa la situación de intercambio de datos, si "request.readyState==4" indica que se ha recibido la información solicitada del servidor.

ReadyState puede tomar los siguientes valores:

0: No inicializado: indica que no se ha abierto la conexión con el servidor (No se ha llamado a Open).

1: Conexión con servidor establecida: indica que se ha abierto la conexión, pero no se ha enviado la petición (no se ha llamado a Send).

2: Recibida petición en servidor: indica que el servidor ya ha recibido la petición (se ha llamado a Send).

3: Enviando información: indica que está enviando la información por parte del servidor, pero todavía no se ha completado la recepción.

4: Completado: Indica que se ha recibido la información del servidor y está lista para operar con ella.

➤ **Status**

Es un código numérico enviado por el servidor que indica el tipo de respuesta dada a la petición, puede tomar los siguientes valores:

200: respuesta correcta.

404: no encontrado.

500: error interno del servidor.

Si se tiene la función de la siguiente manera “request. ReadyState == 4 && request.status == 200” esto indica que se ha completado el intercambio de información y que el servidor ha devuelto una respuesta correcta [12].

Para que la página Web pueda mostrar las imágenes se debe crear varias carpetas donde estén guardadas tanto las imágenes utilizadas para el diseño de la página Web como las imágenes utilizadas para mostrar el estado de la iluminación. Además de los de los tres scripts antes mencionados, estas carpetas deben ser guardadas en el servidor Web instalado en el microcomputador *Raspberry Pi* 3. Es decir, para su correcto funcionamiento los archivos creados deben estar guardados en el servidor Apache siguiendo el orden que se muestra a continuación:




/var/www/html/luces

Nombre	Tamaño	Modificado	Permisos	Propiet...
..		10/7/2017 22:21:56	rwxr-xr-x	pi
data		8/4/2017 17:46:46	rwxr-xr-x	pi
gpio.php	1 KB	29/4/2017 20:29:19	rw-r--r--	pi
index.php	4 KB	10/7/2017 21:46:16	rw-r--r--	pi
script.js	3 KB	29/4/2017 20:29:33	rw-r--r--	pi











/var/www/html/luces/data

Nombre	Tamaño	Modificado	Permisos	Propiet...
.		13/04/2017 9:39:58	rwxr-xr-x	pi
img		08/04/2017 17:46:50	rwxr-xr-x	pi


/var/www/html/luces/data/img

Nombre	Tamaño	Modificado	Permisos	Propiet...
		08/04/2017 17:46:46	rwxr-xr-x	pi
 red		08/04/2017 17:46:51	rwxr-xr-x	pi
 green		08/04/2017 17:46:50	rwxr-xr-x	pi

/var/www/html/luces/data/img/red

Nombre	Tamaño	Modificado	Permisos	Propiet...
		08/04/2017 17:46:50	rwxr-xr-x	pi
 Thumbs.db	157 KB	26/07/2016 20:09:36	rw-r--r--	pi
 red_7.jpg	43 KB	12/10/2013 0:08:00	rw-r--r--	pi
 red_6.jpg	44 KB	12/10/2013 0:23:00	rw-r--r--	pi
 red_5.jpg	22 KB	26/07/2016 20:09:35	rw-r--r--	pi
 red_4.jpg	24 KB	26/07/2016 20:08:42	rw-r--r--	pi
 red_3.jpg	23 KB	26/07/2016 19:57:23	rw-r--r--	pi
 red_2.jpg	21 KB	26/07/2016 19:55:55	rw-r--r--	pi
 red_1.jpg	22 KB	25/07/2016 23:19:35	rw-r--r--	pi
 red_0.jpg	23 KB	25/07/2016 23:17:47	rw-r--r--	pi

/var/www/html/luces/data/img/green

Nombre	Tamaño	Modificado	Permisos	Propiet...
 ..		8/4/2017 17:46:50	rwxr-xr-x	pi
 green_0.jpg	21 KB	25/7/2016 23:22:44	rw-r--r--	pi
 green_1.jpg	22 KB	25/7/2016 23:24:21	rw-r--r--	pi
 green_2.jpg	20 KB	26/7/2016 19:50:08	rw-r--r--	pi
 green_3.jpg	23 KB	26/7/2016 19:51:01	rw-r--r--	pi
 green_4.jpg	22 KB	26/7/2016 19:52:47	rw-r--r--	pi
 green_5.jpg	21 KB	26/7/2016 19:54:00	rw-r--r--	pi
 green_6.jpg	38 KB	11/10/2013 23:59:12	rw-r--r--	pi
 green_7.jpg	37 KB	11/10/2013 23:59:30	rw-r--r--	pi
 Thumbs.db	166 KB	26/7/2016 19:54:01	rw-r--r--	pi

Cuando el sistema domótico se encienda todas las lámparas se prenderán por el transcurso de 1 minuto, tiempo que servirá para poder verificar que todas las lámparas y otros elementos eléctricos estén funcionando correctamente (que no haya lámparas quemadas), después de dicho tiempo el sistema domótico estará listo para ser usado.

3.5 Diseño del control de ventanas y puerta

El sistema de control de estado de ventanas y puertas consta de una página Web desde la cual se puede verificar el estado de las cuatro ventanas y de la puerta principal (abierto o cerrado). El circuito implementado se puede observar en la figura 3.12.

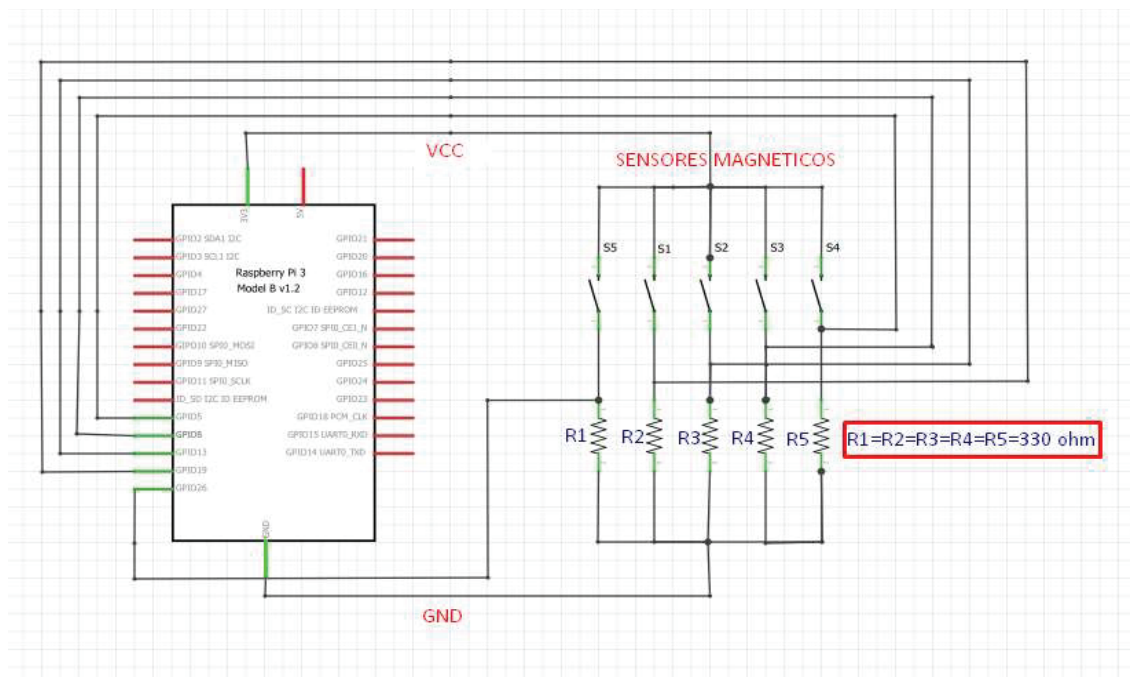


Fig. 3.12 Circuito implementado para el control de ventanas

El sensor magnético (Anexo C-2) es el principal elemento del sistema, el cual es el encargado de sensor constantemente el estado de la puerta o ventana. Además, será el encargado de enviar un nivel lógico al microcomputador *Raspberry Pi 3* para su interpretación.

El funcionamiento del control de ventanas y puertas es similar al del control de iluminación, se utilizó un archivo `Index.php` el cual es el encargado de la programación y diseño de la página Web. Además, lleva incrustado un script en *php* el cual cumple con la función de leer el estado de los sensores magnéticos.

Para la representación de los estados de las ventanas se utilizó 10 imágenes, 5 que muestran el estado abierto y 5 que muestran el estado cerrado, las cuales serán visualizadas por el usuario al momento de abrir la página Web. Es decir, cuando una ventana esté abierta mostrará la imagen que indica que la ventana se encuentra abierta con letra de color azul, caso contrario mostrará la imagen en color rojo que indica que la ventana se encuentra cerrada.

Para editar las imágenes se utilizó el programa *LIGHT RESIZER* (Anexo A-6.c), el cual permite editar y modificar imágenes que fueron utilizadas en la página web variando su tamaño y peso. Todas las imágenes utilizadas tienen una configuración idéntica a la que se muestra en la figura 3.13 ya que sin esta configuración la página Web no tendría una buena presentación.

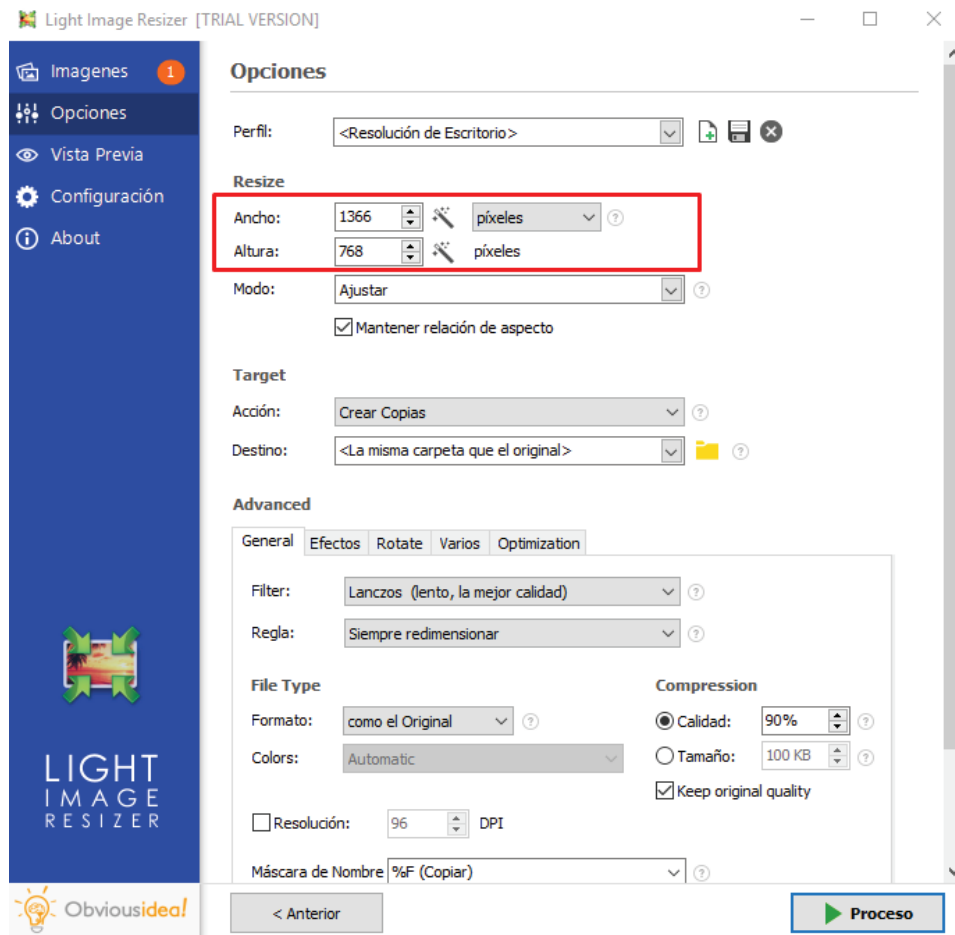


Fig. 3.13 Creación de imágenes para la página Web

El diagrama de flujo del funcionamiento se adjunta en el anexo A-13.b

- **DISEÑO DE LA INTERFAZ PARA EL CONTROL DE VENTANAS Y PUERTA**

Se ha diseñado un script realizado en *Php* para que el microcomputador pueda detectar si en sus pines existe un 1 lógico o 0 lógico, si tiene un 1 lógico el microcomputador Raspberry asumirá que la ventana se encuentra abierta, caso contrario indicará que la ventana se encuentra cerrada. Esto fue implementado en una página web para que esta sea la interfaz gráfica que se muestre al usuario.

El archivo `index.php` es el primer script que despliega el servidor para mostrar la página Web. Como se muestra en el diagrama de bloques (ver figura 3.14) existe una interacción directa entre el archivo `Index.php` y el microcomputador Raspberry. Cuando el usuario desee verificar el estado de las ventanas y de la puerta principal se ejecutará el archivo `Index.php`, éste indica al microcomputador *Raspberry Pi 3* que sus puertos GPIO debe configurarse en modo de entrada para luego preguntar el estado de los mismos. Este estado sirve para seleccionar qué imagen se tiene que mostrar en la página Web.



Fig. 3.14 Diagrama de bloques sistema de control de ventanas

INDEX.PHP (Anexo A-13)

Para el proyecto de control de estado de ventanas se creó un archivo index.php cuya programación se la realizó en lenguaje *Php* y el diseño se lo realizó en HTML. Como se indicó anteriormente hay que tener en cuenta que el archivo index es diferente al index del control de iluminación y al del control de seguridad, llevan el mismo nombre, pero sus contenidos son diferentes.

El archivo Index.php es el responsable de mostrar 10 imágenes que serán las encargadas de indicar al usuario el estado de las ventanas y de la puerta. En la figura 3.15 se puede verificar la página principal con las imágenes que se mostrarán al usuario.



Fig. 3.15 Página principal del sistema de control de ventanas




Para que la página Web se pueda actualizar de manera automática sin la necesidad de presionar manualmente sobre el botón actualizar del navegador, se utilizó un comando el mismo que actualiza la página Web cada cierto tiempo. Para el proyecto se utilizó un tiempo de 1000ms para cada actualización, el comando es el siguiente:

```
<script type="text/javascript">
setInterval( "window.location.reload()" ,1000);
</scrint>
```

Para que la página Web pueda mostrar las imágenes se deben crear varias carpetas donde estén guardadas tanto las imágenes utilizadas para el diseño de la página Web como las imágenes utilizadas para mostrar el estado de las ventanas, estas carpetas están guardadas en el servidor Web instalado en el microcomputador *Raspberry Pi 3*.

Para su correcto funcionamiento los archivos deben ser guardados en las carpetas que se detallan a continuación en el siguiente orden:







/var/www/html/puertas

 ..		Directorio superior	17/05/2017 21:12:32
 data		Carpeta de archivos	17/05/2017 20:56:18
 index.php	2 KB	Archivo PHP	17/05/2017 21:12:32

/var/www/html/puertas/data

 img		10/7/2017 21:42	Carpeta de archivos
---	--	-----------------	---------------------







/var/www/html/puertas/data/img

 ..		Directorio superior	17/05/2017 21:00:31
 abierto		Carpeta de archivos	17/05/2017 21:00:09
 cerrado		Carpeta de archivos	17/05/2017 21:01:13
 imagen1.jpg	10 KB	Imagen JPEG	31/07/2016 0:01:51
 imagen2.jpg	9 KB	Imagen JPEG	31/07/2016 0:02:22
 Thumbs.db	90 KB	Data Base File	01/08/2016 22:25:46

/var/www/html/puertas/data/img/abierto

 ..		Directorio superior	17/05/2017 21:00:09
 abierto_21.jpg	125 KB	Imagen JPEG	01/08/2016 21:00:47
 abierto_22.jpg	126 KB	Imagen JPEG	01/08/2016 21:04:40
 abierto_23.jpg	118 KB	Imagen JPEG	01/08/2016 21:10:00
 abierto_24.jpg	125 KB	Imagen JPEG	01/08/2016 21:18:59
 abierto_25.jpg	121 KB	Imagen JPEG	01/08/2016 21:13:14

/var/www/html/puertas/data/img/cerrado

 ..		Directorio superior	17/05/2017 21:01:13
 cerrado_21.jpg	106 KB	Imagen JPEG	01/08/2016 20:59:21
 cerrado_22.jpg	105 KB	Imagen JPEG	01/08/2016 21:02:31
 cerrado_23.jpg	97 KB	Imagen JPEG	01/08/2016 21:06:28
 cerrado_24.jpg	105 KB	Imagen JPEG	01/08/2016 21:16:02
 cerrado_25.jpg	102 KB	Imagen JPEG	01/08/2016 21:11:52

3.6 Diseño del sistema de control de seguridad

El sistema de seguridad del hogar está diseñado para cuando el usuario esté fuera de casa y desee tener una máxima protección de su hogar. Este sistema de seguridad está controlado mediante la página Web desde la cual se podrá activar y desactivar el sistema de control de seguridad. El circuito implementado se observa en la figura 3.16.

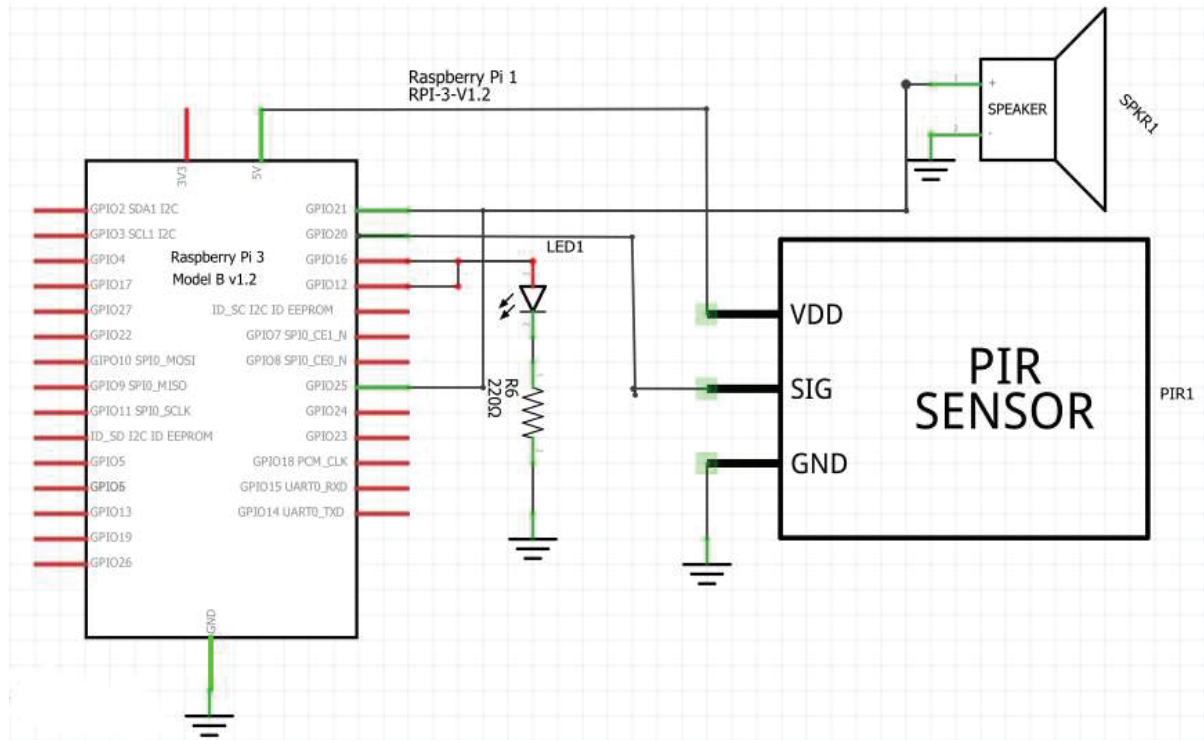


Fig. 3.16 Circuito implementado del sistema de seguridad

El sensor PIR (Anexo A-3) es el elemento principal, el cual es encargado de sensar cualquier movimiento que ocurra dentro de la casa. Este sensor está instalado en un lugar donde se obtiene la mayor visibilidad (pasillo de la casa). Cuando el sensor PIR detecte la presencia de algún intruso, enviará una señal al microcomputador *Raspberry Pi 3*, el mismo que procesará la información y realizará los siguientes pasos:

- Capta 3 imágenes a través de la cámara Web (Marca ZENSE –AW13) para enseguida guardarlos en una carpeta del microcomputador *Raspberry Pi 3*.
- Envía las imágenes guardadas a una cuenta de Twitter (previamente configurado).
- Emite una advertencia sonora a través de un parlante ubicado en el domicilio.
- Enciende una sirena (alarma).

Para el desarrollo del proyecto se utilizaron varios programas que deben ser instalados y que se detallan a continuación:

➤ **FSWEBCAM**

Este software es usado para capturar imágenes desde la *Webcam* conectada al puerto USB del microcomputador *Raspberry Pi 3* cuando el sensor PIR detecte un movimiento. *Fswebcam* es una aplicación fácil de usar, la cual puede ejecutarse desde la línea de comandos o desde un script bash en cualquier momento [13]. Además, se instaló la librería *Fswebcam* que permite ubicar títulos, subtítulos y también permite mostrar la hora en la que fue capturada la imagen en el pie de la fotografía.

Su instalación es simple solamente se debe ejecutar la siguiente línea en un terminal.

```
apt-get install fswebcam
```

Después de descargarse la librería ya está lista para ser usada. También se debe tomar en cuenta que no todas las cámaras Web son compatibles con la Raspberry, en éste caso se utilizó una Web cámara Zense (CAM-ZEN-ATW13) la cual funcionó de manera correcta.

El script para captar la imagen es el siguiente:

```
sudo fswebcam -r 320x240 -S 5 --title 'ALARMA EPN' --subtitle 'MOVIMIENTO  
DETECTADO' --info 'Monitor: Pasillo' --save foto_alarma.jpg
```

Donde:

-r: es la resolución de la imagen (ancho por altura).

-S: determina el número de frameworks (imagen) que salta para capturar una imagen estable y con buena iluminación.

--title, --subtitle, --info: son los diferentes campos del texto colocados en el pie de la fotografía.

--save: indica la ruta y nombre de archivo donde guarda la fotografía.

Tras estas configuraciones la aplicación capta imágenes como la que se puede ver en la figura 3.17.



Fig. 3.17 Imagen captada desde la Raspberry con Fswecam

➤ **TWYTHON**

Este software es utilizado para enviar las fotos almacenadas en el microcomputador *Raspberry Pi 3* a una cuenta de Twitter, usando un tweet a través de un script de *Python*. Para tener una comunicación entre la Raspberry y Twitter se necesita crear una aplicación en la API de Twitter. La API de Twitter es un programa diseñado para que programas externos interactúen con Twitter [14]. Debido a que Twython es una librería de *Python* para la API de Twitter, necesariamente se tiene que instalar *Python* en el microcomputador *Raspberry Pi 3*.

Para poder utilizar éste método de interacción entre el microcomputador Raspberry y *Python* es necesario registrarse en una cuenta. Además, que se debe crear una APP en Twitter, los pasos son muy sencillos y se explican a continuación:

1. Ingresar al link <https://apps.twitter.com/>
2. Ir al botón "Create New App"
3. Llenar los campos solicitados con la siguiente información:
 - **Name:** *Raspberry Pi Domótica*
 - **Description:** Permite revisar imágenes de la persona que ingresó a la casa (o cualquier información que describa la aplicación).
 - **Webzine:** <https://geekytheory.com> (se puede ubicar cualquier página web).
 - Los campos **Callback URL**, **Privacy Policy URL**, **Terms of Service URL**, no son necesarios de llenar pueden quedar vacíos.

- Una vez creada la App dirigirse a la pestaña **“permissions”** y escoger la opción **“Read, Write and Access direct messages”**
- Dirigirse a la pestaña **Keys and Access Tokens** y presionamos sobre **“Create My Access Token”** después se muestra una imagen como la de la figura 3.18, copiar los códigos alfanuméricos de **consumer key**, **consumer secret**, **Access token**, **Access token secret**, pues estos son los campos necesarios para ingresar en el script de *Python*.

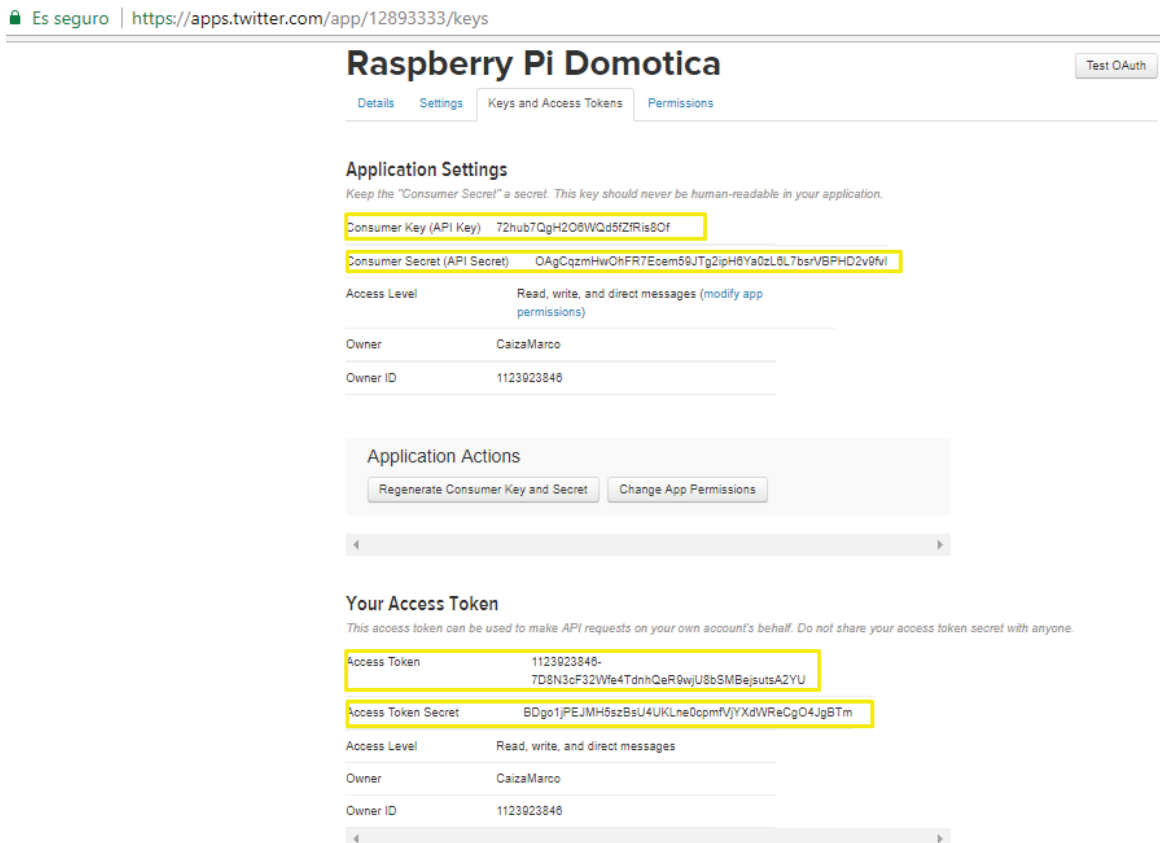


Fig. 3.18 Códigos necesarios para utilizar en python

Una vez que se tengan los datos necesarios del lado de Twitter, se procede con la instalación de la librería de TWYTHON para lo cual se necesita que el microcomputador *Raspberry Pi 3* esté actualizado y que tenga acceso a Internet. Se debe colocar en un terminal los siguientes comandos para culminar con la instalación:

```
Sudo apt-get update
Sudo apt-get upgrade
Sudo apt-get install python-setuptools
Sudo easy_install pip
Sudo pip instalar Twython
```

Después de la instalación se necesita ejecutar un script para probar su correcto funcionamiento. El script que se muestra a continuación envía una foto a Twitter (la foto enviada debe estar guardada en el microcomputador *Raspberry Pi 3*).

```
#!/usr/bin/python

# -*- coding: utf-8 -*-

from twython import Twython

CONSUMER_KEY = '72hub7QgH206WQd5fZfRis80f'

CONSUMER_SECRET =
'OAgCqzmHwOhFR7Ecem59JTg2ipH6Ya0zL6L7bsrVBPHD2v9fv1'

ACCESS_KEY = '1123923846-7D8N3cF32Wfe4TdnhQeR9wjU8bSMBejsutsA2YU'

ACCESS_SECRET = 'BDgo1jPEJMH5szBsU4UKLne0cpmfVjYXdWReCg04JgBTm'

api = Twython (CONSUMER_KEY,CONSUMER_SECRET,ACCESS_KEY,
ACCESS_SECRET)

photo = open ('foto_alarma.jpg', 'rb')

api.update_status_with_media(status='Intruso detectectado por la
alarma Raspberry ', media=photo)
```

➤ ESPEAK

Una vez que el sensor PIR detectó un movimiento, el sistema de seguridad emitirá un mensaje de advertencia por los altavoces para lo cual es necesario utilizar la librería *Espeak*, la cual es un programa para convertir texto en voz y que es capaz de ser manejada desde una consola de texto. Además, se puede controlar la velocidad y el tono de voz que se desee, pero su desventaja principal es que reproduce una voz muy robótica.

Para su uso se tiene que configurar de la siguiente manera (el texto a transformarse es el que se encuentra entre comillas):

```
espeak 'ALarma desactivada'
```

Si se reproduce el tono con esta configuración se puede notar que existe una pronunciación que no es correcta, esto ocurre ya que la aplicación viene por defecto en idioma inglés. Por

lo cual se tiene que indicar en qué idioma se desea reproducir el texto para esto se tiene que ubicar `-v` seguido de las iniciales del idioma deseado. En este caso quedaría de la siguiente manera:

```
espeak -ves 'Alarma desactivada'
```

Existe una infinidad de voces disponibles para reproducir el mensaje que pueden ser escogidas, si se desea escoger una voz masculina se tiene que colocar después del parámetro del idioma `+m1,+m2` o `+m3`; si se desea ocupar voces femeninas se puede utilizar `+f1,+f2,+f3`. Adicional, se puede controlar la velocidad de la voz, la configuración va después de `-s`, por ejemplo, en la siguiente línea de comando se configuró una velocidad de 150 palabras por minuto y con voz masculina.

```
espeak -ves+m3-s150 'Alarma desactivada'
```

El código para instalar Espeak desde un terminal es el siguiente:

```
sudo apt-get install espeak
```

La instalación comenzará de manera automática, una vez finalizado estará listo para ser usado [15].

➤ **LIBRERÍA RPI.GPIO**

Para que *Python* pueda interactuar con los puertos GPIO es necesario instalar la librería RPI.GPIO, la cual maneja la numeración de los pines de una manera diferente a las que maneja la librería WiringPi (misma que se utilizó para el control de la iluminación y control de puertas).

Su instalación es fácil y sencilla, el primer paso es abrir un terminal para comprobar que se tenga instalado *Python* y además que se encuentre actualizada su versión, esto se logra con el comando.

```
Sudo Python
```

Dentro de la consola comprobar la versión de *python*, se tiene que utilizar la versión 0.5.4 o mayor, para validar la versión instalada se puede utilizar las siguientes líneas de comando:

```
import RPi.GPIO  
rpi.gpio.version
```

Una vez comprobada la versión se procede a instalar la librería colocando la siguiente línea de configuración en el terminal.

```
wget 'http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.6.3.tar.gz'
```

Luego se procede a descomprimir el archivo utilizando el siguiente comando:

```
tar zxvf RPi.GPIO-0.6.3.tar.gz
```

Ingresa al directorio donde se instaló.

```
cd RPi.GPIO-0.5.4/
```

A continuación, instalar el paquete Python-dev introduciendo el siguiente comando:

```
sudo apt-get install python-dev
```

Finalmente ejecutar el archivo setup.py

```
sudo python setup.py install
```

DISEÑO DE LA INTERFAZ DEL CONTROL DE SEGURIDAD

Para el funcionamiento del sistema de seguridad se procedió a diseñar tres scripts los cuales trabajan en conjunto. En la figura 3.19 se puede observar un diagrama de bloques de la interacción de los tres scripts.

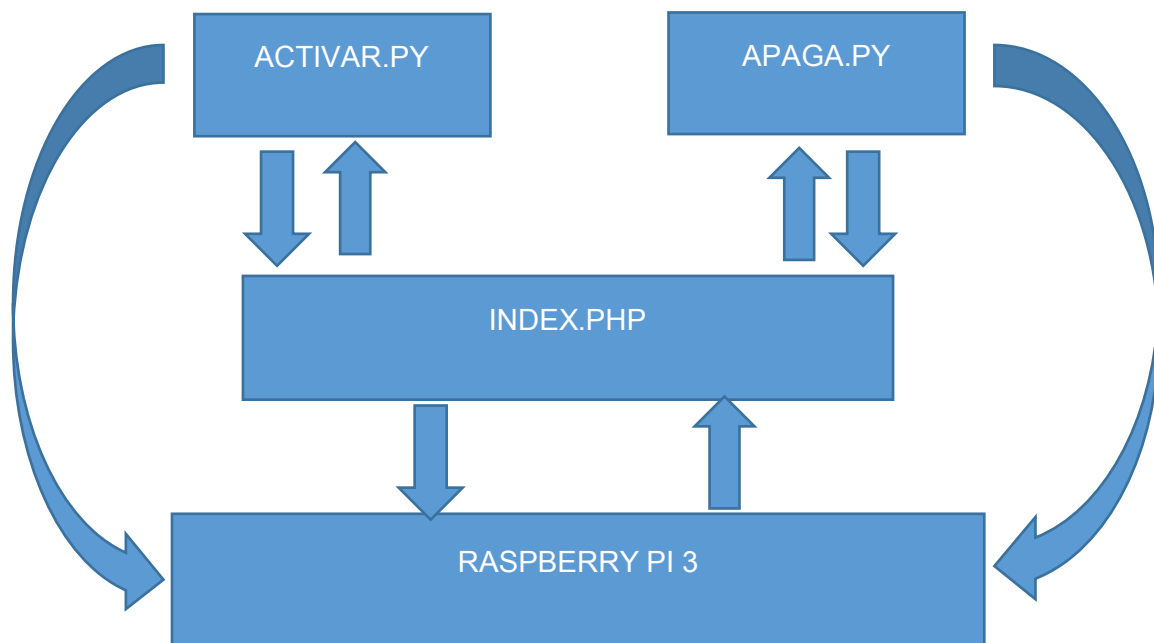


Fig. 3.19 Diagrama de bloques del sistema de seguridad

Se puede controlar los puertos del microcomputador *Raspberry Pi 3* con sencillos scripts realizados en *Python*, los mismos que serán activados y desactivados desde una página Web, la cual se desarrolló con la ayuda de *Dreamweaver* en lenguaje *Php*.

Los scripts creados son:

- Index.php
- Activar.py
- Apaga.py

➤ **INDEX.PHP (Anexo A-14.a)**

El archivo index es fundamental ya que es el encargado de mostrar la página Web al usuario. Esta página se mostrará en el momento que el cliente escoja la opción “Salir de casa” de la página Web principal.

➤ **ACTIVAR.PY (Anexo A-14.b)**

El script activar.py fue desarrollado en *Python* para sensar los sensores magnéticos de la puerta, las ventanas y el pasillo y según el estado de cada uno de ellos el microcomputador *Raspberry Pi 3* tomará decisiones como tomar fotos, enviar tweets, emitir mensajes sonoros, etc.

En el anexo A-14.d se muestra el diagrama de flujo del funcionamiento del sistema de seguridad.

➤ **APAGO.PY (Anexo A-14.c)**

Este script cumple la función de cancelar cualquier script que se encuentre en ejecución, es decir entrará en funcionamiento el momento que el usuario desee desactivar el sistema de seguridad. Por lo tanto, permite desactivar en cualquier momento el sistema.

Para lograr éste objetivo se usó el comando pkill, el cual permite matar o cancelar un programa simplemente especificando el nombre del programa y también se colocó la línea de configuración espeak para que se emita un mensaje sonoro que indique que la alarma fue desactivada.

Una vez que los programas requeridos han sido instalados y que los scripts han sido cerrados se procede a ejecutar las pruebas de funcionamiento para luego instalar el prototipo en una maqueta.

3.7 Implementación de la maqueta y pruebas de funcionamiento

En éste apartado se presenta la placa de circuito impreso (PBC) utilizado por el módulo esclavo, el cual es el encargado de ejecutar la acción que determine el microcomputador *Raspberry Pi 3*. El módulo esclavo contiene los siguientes elementos: lámparas, sensores magnéticos, un sensor PIR y una sirena.

La placa de circuito impreso se lo realizó utilizando la herramienta Ares del software Proteus. El circuito consta de una sola placa la cual tiene como principal función conectar los diferentes dispositivos y elementos electrónicos con los pines GPIO del microcomputador *Raspberry Pi 3*.

En la figura 3.20 se puede observar el circuito esquemático en Proteus, mientras que en la figura 3.21 se muestra el circuito PCB creado en Ares listo para su impresión. El siguiente paso es el montaje de los elementos sobre la placa.

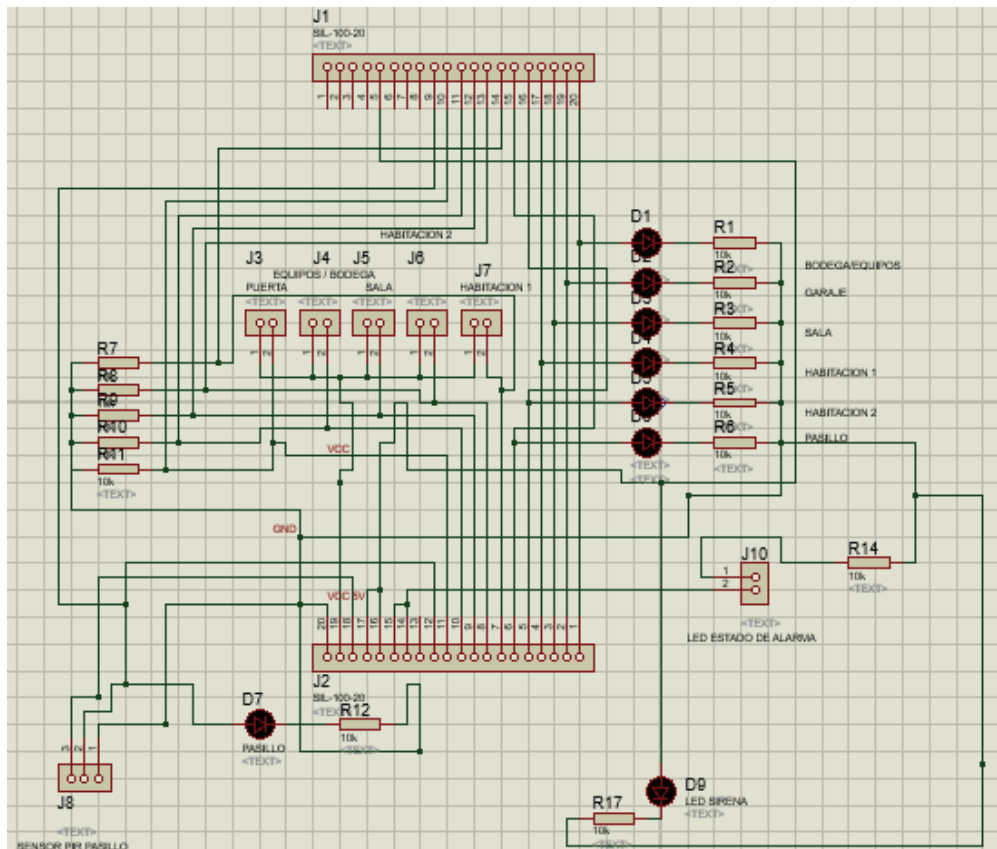


Fig. 3.20 Circuito esquemático diseñado en Proteus

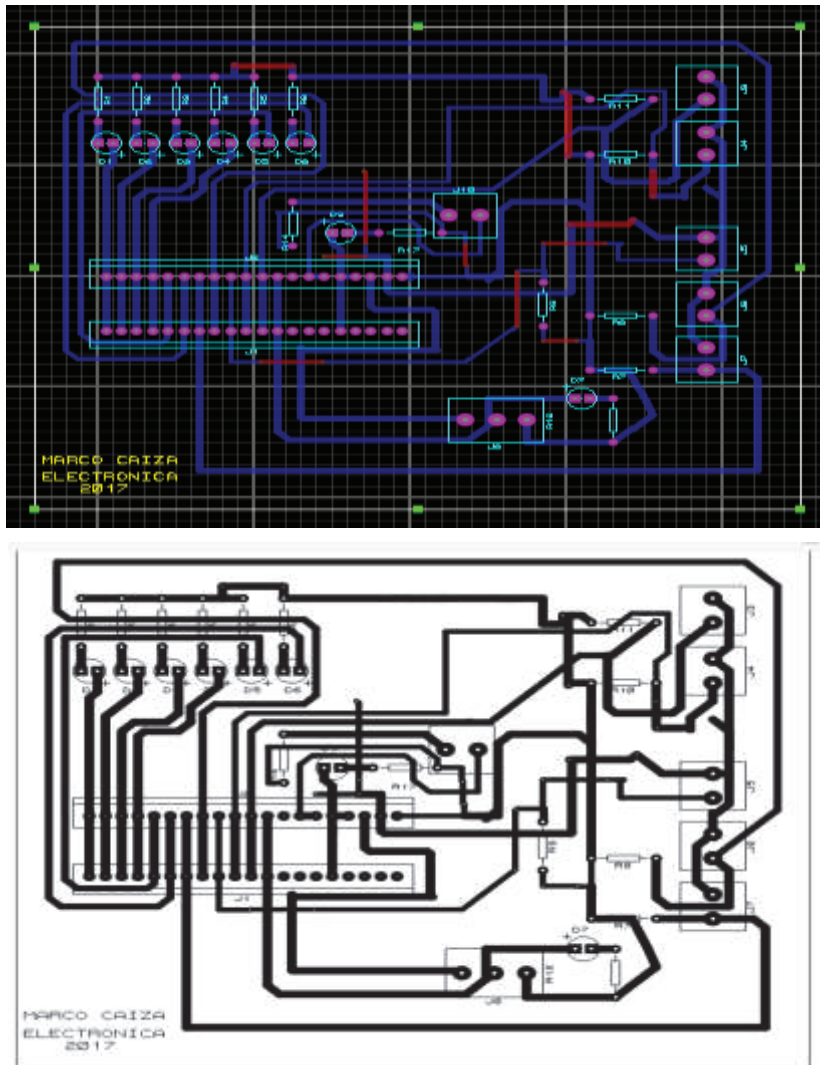


Fig. 3.21 Circuito impreso diseñado en Ares

Una vez montados los elementos en la placa el prototipo está listo para ser usado. El primer paso es comprobar la conectividad del microcomputador *Raspberry Pi 3* dentro de la red LAN para lo cual basta con realizar un ping extendido a la dirección IP asignada al microcomputador *Raspberry Pi 3*.

Ping [dirección IP] -t

```
C:\Users\Equipo>ping 192.168.1.20 -t

haciendo ping a 192.168.1.20 con 32 bytes de datos:
Respuesta desde 192.168.1.20: bytes=32 tiempo=441ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=19ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=14ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=22ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=15ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=15ms TTL=64
Respuesta desde 192.168.1.20: bytes=32 tiempo=19ms TTL=64
```

Fig. 3.22 Ping extendido

Después de comprobar la conectividad se procede con la verificación de la visualización correcta de la página Web almacenada en el microcomputador *Raspberry* mediante la cual

se tendrá acceso a la página de login. Para esto se debe ingresar a un navegador de Internet (de preferencia *Google Chrome*) e ingresar en la barra de direcciones la dirección IP previamente configurada en el microcomputador *Raspberry Pi 3*. En éste caso es 192.168.1.20.

En la figura 3.23 se muestra la página de login y de bienvenida.

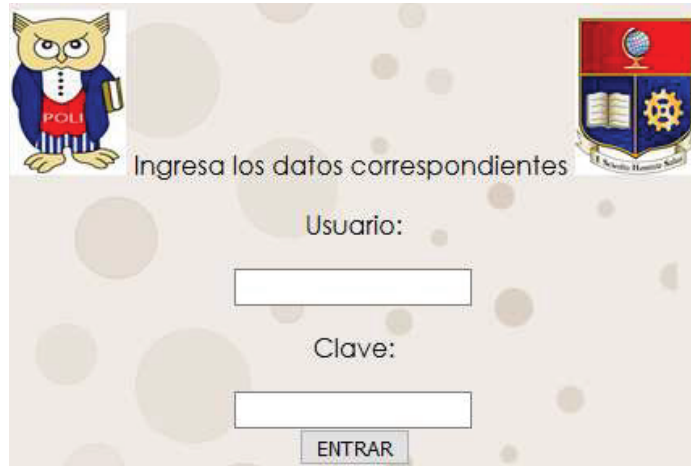


Fig. 3.23 Página principal del proyecto de titulación

En el anexo B se muestra un manual para crear los usuarios y contraseñas necesarias para poder ingresar al sistema domótico.

Si la contraseña es correcta se desplegará la página Web que se indica en la figura 3.24, la misma que muestra tres opciones que el usuario puede escoger. Se puede observar una opción para el control de iluminación, otra para verificar el estado de ventanas y finalmente una opción para ingresar al control de seguridad.



Fig. 3.24 Página Web principal

➤ **Pruebas del control de iluminación**

A través del control de iluminación del prototipo es posible encender y apagar las lámparas distribuidas en la maqueta. Además, se puede observar el estado de la mismas. En la figura 3.25 se verifica que el foco de la habitación uno se encuentra encendido mientras que el foco de la sala y habitación dos se encuentran apagados.

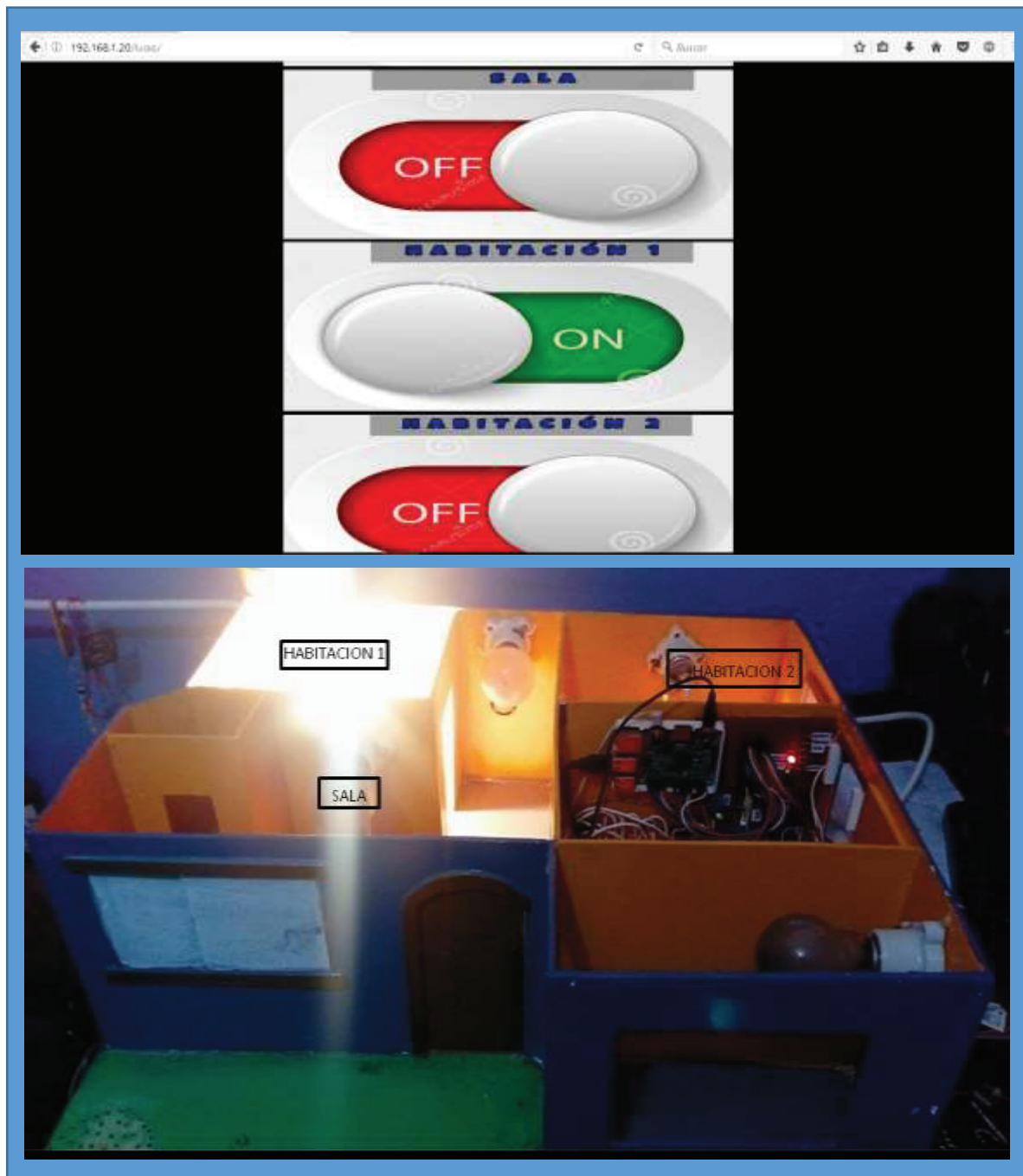


Fig. 3.25 Ejemplo del funcionamiento del sistema de iluminación

➤ **Pruebas del control de ventanas y puerta**

Una vez escogida la opción MOSTRAR ESTADO DE PUERTAS Y VENTANAS de la página principal se mostrará el estado de las mismas. En la figura 3.26 se puede observar que la ventana del cuarto de equipos y la puerta principal se encuentran abiertas (imagen roja), mientras que las demás ventanas de color azul se encuentran cerradas.



Fig. 3.26 Ejemplo del funcionamiento del estado de ventanas y de la puerta

➤ **Pruebas de funcionamiento del control de seguridad**

Una vez seleccionada la opción SALIR DE CASA se mostrará la página Web de la figura 3.27, la misma que consta de 4 botones que se detalla a continuación:

1. Botón regresar a la página principal.
2. Botón para activar el control de seguridad.
3. Botón para desactivar el control de seguridad.
4. Botón para verificar el estado de la alarma (si se encuentra activada y si detectó la presencia de intrusos) presenta las siguientes condiciones

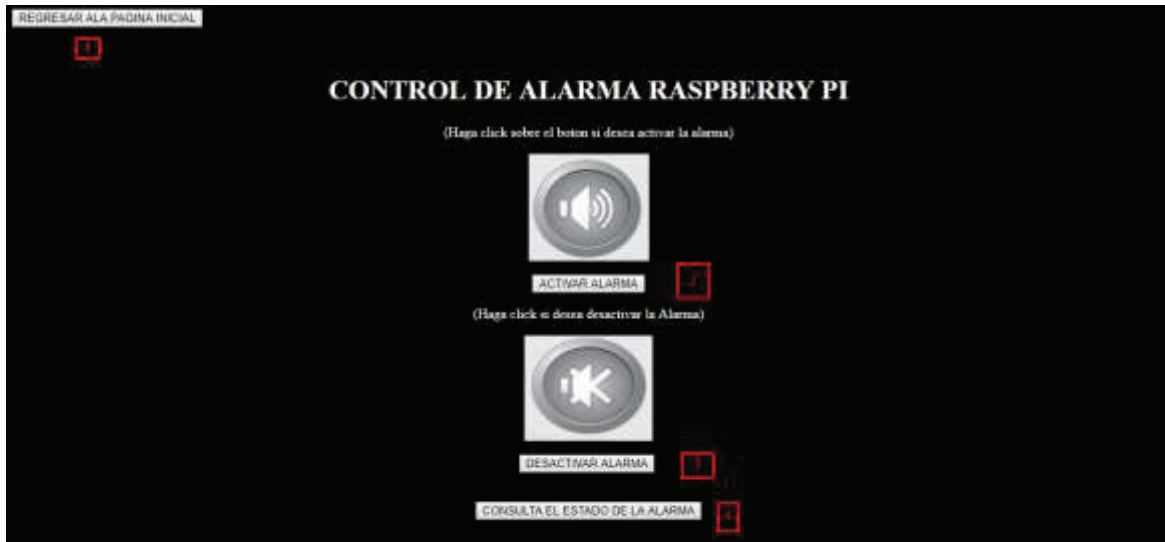


Fig. 3.27 Página Web sistema de seguridad

Los botones 2 y 3 tienen la función de activar y desactivar el control de seguridad.

El botón 4 cumple la función de verificar el estado del control de seguridad, al presionar el botón nos muestra una página Web la cual cambia las imágenes dependiendo del estado del control de seguridad y presenta las siguientes condiciones:

Si la alarma se encuentra activa y además detectó la presencia de intrusos, la página Web que se visualizará se muestra en la figura 3.28.



Fig. 3.28 Alarma activa con detección de intrusos

Si la alarma se encuentra desactivada y no ha detectado la presencia de intrusos, la página Web desplegada es la que se puede observar en la figura 3.29.



Fig. 3.29 Alarma desactivada sin detección de intrusos

Si la alarma se encuentra activa y no detectó la presencia de intrusos, la página Web mostrada es la que se puede observar en la figura 3.30.

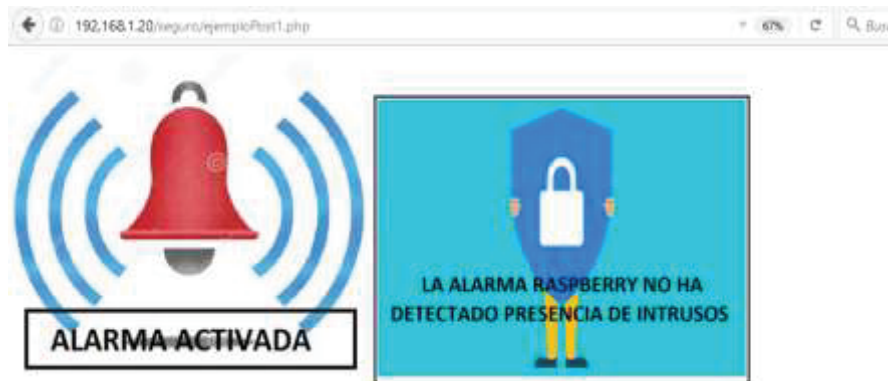


Fig. 3.30 Alarma activa sin detección de intrusos

3.8 Diseño y ensamblaje de la maqueta

Para representar la vivienda se construyó una maqueta, la cual representa una vivienda simple de un piso, y consta de dos habitaciones, un cuarto de equipos, una sala, un garaje y un baño. Esta fue diseñada para ajustarse a los requerimientos del sistema domótico (figura 3.31) y permitió instalar el control de iluminación, el control de ventanas y puerta y el sistema de seguridad.

La maqueta tiene como medidas 60cm por 52cm, las medidas completas se detallan en la figura 3.32.

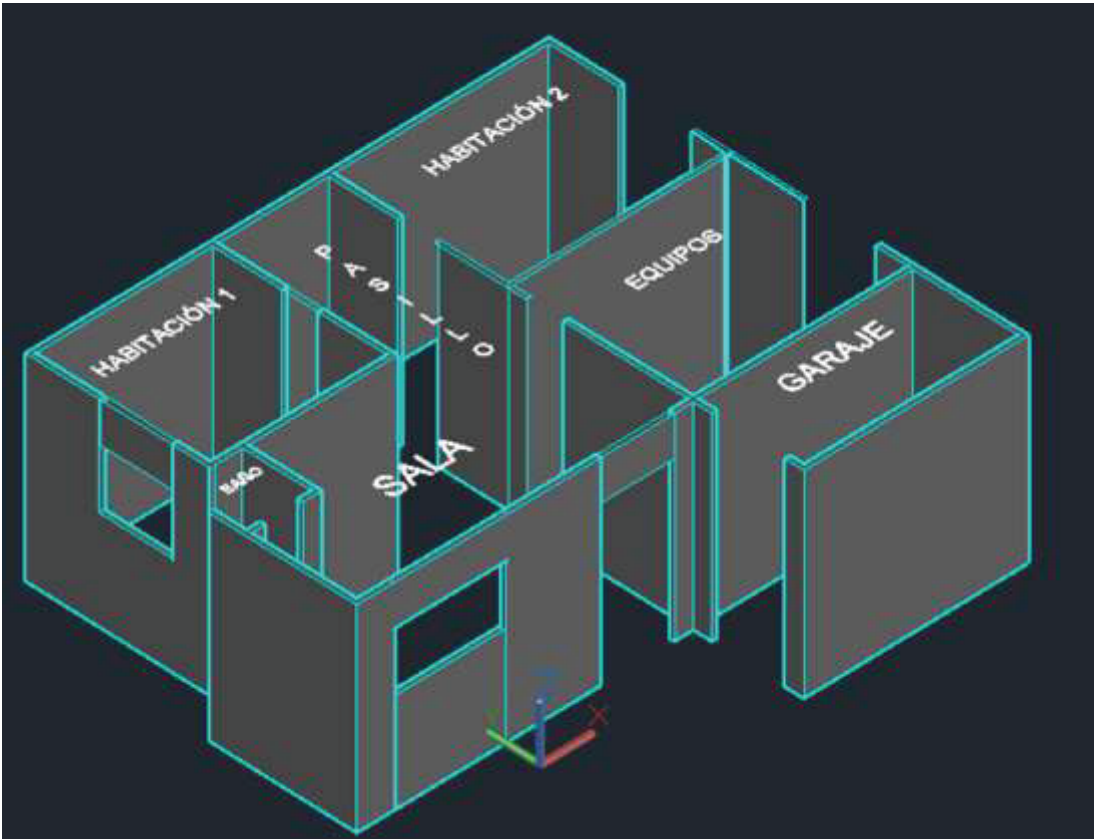


Fig. 3.31 Plano de la maqueta en 3D

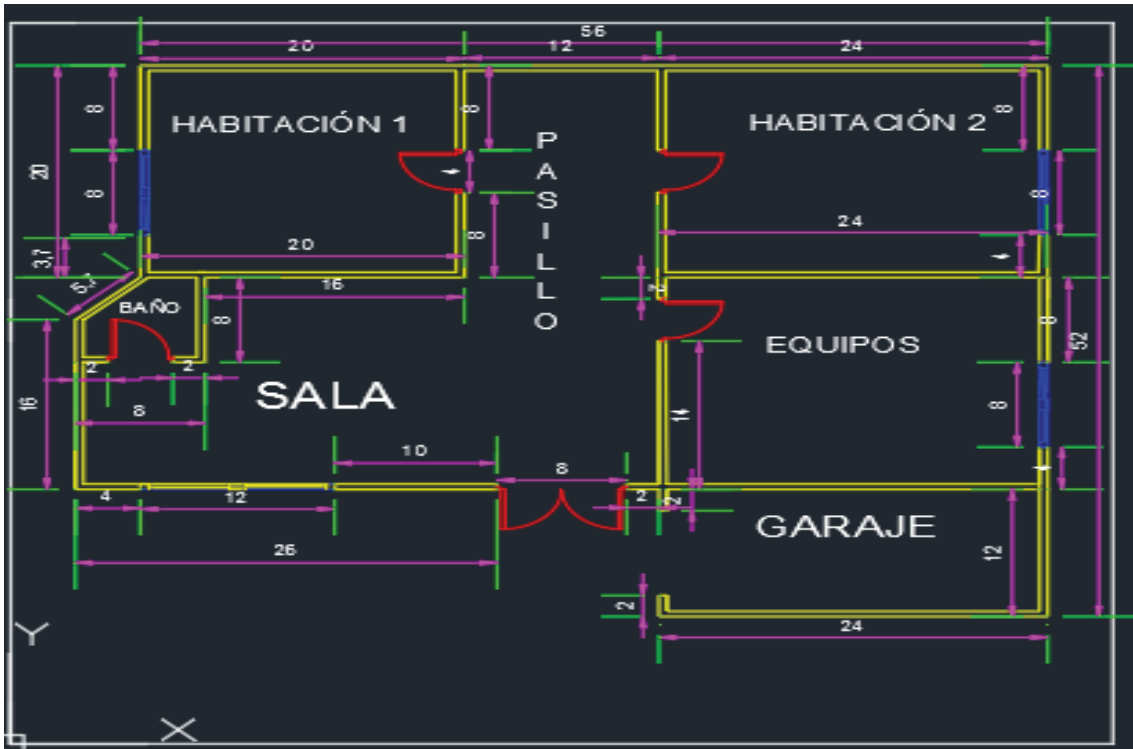


Fig. 3.32 Plano de la maqueta

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- La gama de microcomputadores *Raspberry Pi* cumple un mismo fin que es el de automatizar las cosas. La principal ventaja, por la cual es aceptado por una gran comunidad de usuarios es su asequible precio en comparación con otros microcomputadores.
- Se determinó que el microcomputador *Raspberry Pi 3* es óptimo para la construcción del sistema domótico, en primer lugar, por contar con un procesador mucho más rápido y potente a comparación de sus versiones anteriores. Además, el rendimiento es un 50 por ciento superior al microcomputador *Raspberry Pi2*. Otra ventaja que trae el microcomputador *Raspberry Pi 3*, es que funciona como fuente para los demás dispositivos conectados a ella proporcionando dos tipos de fuente 3.3 VDC y 5VDC; con esto se logra un consumo reducido de energía eléctrica.
- Existe una variedad de sistemas operativos que son capaces de trabajar con Raspberry; sin embargo, se analizó que para éste proyecto es necesario trabajar con *Raspbian* el mismo que está catalogado como el sistema operativo oficial de *Raspberry Pi* además de ser un sistema operativo para trabajo en comparación con sus competidores que son usados para tareas de entretenimiento.
- El sistema domótico puede ser controlado de manera remota con la ayuda de SSH (Secure SHell), pero esta interfaz de consola es complicada de utilizar y más si el usuario no tiene conocimientos de esta herramienta, por lo cual se implementó una página web para el control del sistema domótico.
- Crear una aplicación domótica para que se ejecute en cada sistema operativo (*Windows, MAC, Linux, Android, etc.*) es difícil y toma demasiado tiempo la implementación. Es por éste motivo que en el presente proyecto se diseñó páginas web accesibles desde cualquier dispositivo que cuente con un navegador dentro de la vivienda.
- El diseño de la página Web se realizó utilizando el programa *Dreamweaver*. Se comprobó que es un programa muy útil, pero al ser un programa tan avanzado su manejo puede resultar complicado.

- Para Implementar una interfaz entre la página Web, el microcomputador, los sensores y los actuadores se desarrolló varios scripts, utilizando principalmente dos lenguajes de programación como son *Php* y *Java script* permitiendo con esto una integración entre todos los elementos.
- Para la comunicación entre la página Web y el servidor se utilizó varios tipos de programación dependiendo de lo que se desea realizar, es así que para el control del sistema de seguridad se optó por utilizar el lenguaje de programación *Python* ya que es necesario utilizar un lenguaje de programación que se ejecute aún cuando haya otros scripts ejecutándose en el mismo microcomputador Raspberry.
- Para el control de iluminación, el control de estado de ventanas y la activación del sistema de seguridad se presenta varias interfaces gráficas amigables y fáciles de usar, mismas que fueron almacenadas en el microcomputador *Raspberry Pi 3*. Se puede acceder a ella desde cualquier dispositivo con conexión a Internet. Para ello el microcomputador *Raspberry Pi 3* debe estar conectado a una red de Internet de forma alámbrica o inalámbrica.
- El sistema de iluminación depende de *Java script* ya que éste lenguaje de programación que permite actualizar la página Web de una manera automática, mientras tanto que para el sistema de control de ventanas y puertas se utilizó *Php* que permite utilizar la función *Windows reload* para actualizar la página Web cada determinado tiempo.
- Para tener una mayor seguridad del sistema domótico se optó por tener una página de seguridad para de esta manera dotar de una mayor seguridad, por lo cual se desarrolló una base de datos donde se integraron tres aplicaciones *Php*, *MySQL* y *PHPMYAdmin*, las cuales permitieron crear una base de datos que contiene los usuarios y contraseñas de las personas que están autorizados a utilizar el sistema domótico.
- El diseño del proyecto sirve como una base o guía para personas que deseen conocer el funcionamiento del microcomputador *Raspberry Pi 3* y que estén interesadas en crear un proyecto domótico y no cuenten con conocimientos especializados.

- El administrador de la base de datos debe tener conocimientos básicos de *PHPMysqlAdmin* ya que será la persona responsable de asignar el usuario y contraseña a las personas que deseen utilizar el sistema domótico.

Recomendaciones:

- Para prevenir fallos de operación del sistema domóticos a causa de interrupciones eléctricas, se recomienda instalar una fuente de energía alternativa, de ese modo se asegura que el prototipo domótico no deje de funcionar en caso de alguna interrupción eléctrica.
- Se debe leer las especificaciones técnicas de cada dispositivo utilizado en el proyecto antes de trabajar con los mismos, con el fin de evitar daños en su utilización.
- Se recomienda sacar una copia de la tarjeta SD (backup) de cada archivo que se cree en el caso que la tarjeta de memoria principal presente fallas, el software que se puede usar es SDformatter.
- Se puede utilizar una base de datos para almacenar información proveniente de la Raspberry y mostrar en una página Web o guardar un historial de datos o hechos que sucedan en domicilio todo esto se puede lograr si se tiene un nivel alto de conocimiento en *MySQL*.
- Antes de conectar elementos eléctricos en los pines de la Raspberry se tiene que verificar si la configuración del pin gpio es pull-up o pull-down, si no se verifica dicha configuración la Raspberry puede sufrir daños.
- Se puede agregar más focos o ventanas en el domicilio, pero se recomienda utilizar un expansor de puertos GPIO Raspberry ya que se necesitaría más pines disponibles.
- Para la implementación del sistema en la vida real se recomienda ubicar más cámaras de una buena resolución en diferentes partes del domicilio. También se recomienda contratar un dominio para que la página Web de control del sistema

domótico pueda ser vista desde cualquier parte del mundo ya que el presente proyecto se realizó como prueba para una red WLAN.

BIBLIOGRAFÍA

- [1] STEFAN JUNESTRAND; XAVIER PASSARET; DANIEL VAZQUEZ, DOMOTICA Y HOGAR DIGITAL, MAGALLANES, MADRID: THOMSON EDICIONES SPAIN, 2005.
- [2] CEDATOS, «LA INSEGURIDAD EN EL ECUADOR,» Abril 2011. [En línea]. Available: http://www.cedatos.com.ec/detalles_noticia.php?Id=86. [Último acceso: 15 Diciembre 2016].
- [3] KUO BENJAMIN C, SISTEMAS DE CONTROL AUTOMATICO, SEPTIMA EDICION ed., PEARSON PRENTICE HALL, p. 19.
- [4] JOSE RAPALLINI- SEBASTIÁN ALEJANDRO- JUAN MANUE-HÉCTOR HUGO MAZZEO, «DOMOTICA-DESARROLLO DEL SISTEMA DE CONTROL PARA UN EDIFICIO INTELIGENTE,» Departamento de Ingeniería en Sistemas de Información Unlversida Tecnologica de la Plata, [En línea]. Available: http://uea2013.frbb.utn.edu.ar/wp-content/uploads/TE_2.pdf. [Último acceso: 12 Diciembre 2016].
- [5] RASPBERRY PI FOUNDATION, «RASPBERRY PI,» UK RESTERED CHARITY, [En línea]. Available: <https://www.raspberrypi.org/products/model-a/>. [Último acceso: 09 FEBRERO 2017].
- [6] J. A. PASCUAL, «RASPBERRY PI :¿QUÉ MODELO ME COMPRO,» COMPUTER HOY.COM, 02 FEBRERO 2015. [En línea]. Available: <http://computerhoy.com/noticias/hardware/raspberry-pi-que-modelo-me-compro-23811>. [Último acceso: 07 FEBRERO 2017].
- [7] LIGNUX, «SISTEMAS OPERATIVOS EN LA RASPBERRY PI,» 24 MAYO 2015. [En línea]. Available: <https://lignux.com/sistemas-operativos-en-la-raspberry-pi/>. [Último acceso: 18 FEBRERO 2017].
- [8] VICTOR, «MIS EXPERIENCIAS CON RASPBERRY,» 25 02 2013. [En línea]. Available: <http://muyraspi.blogspot.com/2013/02/programar-gpio.html>. [Último acceso: 31 03 2017].

- [9] WIRING PI, «GPIO INTERFACE LIBRARY FOR THE RASPBERRY PI,» 2017. [En línea]. Available: <http://wiringpi.com/the-gpio-utility/>. [Último acceso: 09 ABRIL 2017].
- [10] W3SCHOOLS.COM, «ONCLICK EVENTO,» 2017. [En línea]. Available: https://www.w3schools.com/jsref/event_onclick.asp. [Último acceso: 03 Mayo 2017].
- [11] DESAROLLOWEB.COM, «ELIMINAR ETIQUETAS HTML Y PHP DE UN STRING,» 2010. [En línea]. Available: <https://desarollobweb.com/faq/eliminar-etiquetas-html-php-string.html>. [Último acceso: 03 MAYO 2017].
- [12] W3SCHOOLS.COM, «EL OBJETO XMLHttpRequest,» 2017. [En línea]. Available: https://www.w3schools.com/xml/dom_httprequest.asp. [Último acceso: 03 MAYO 2017].
- [13] S. L. Bua, «RASPBERRY PI - WEBCAM STREAMING,» SLB LABS, [En línea]. Available: <http://www.slblabs.com/2012/09/26/rpi-webcam-stream/>. [Último acceso: 24 JUNIO 2017].
- [14] INSTRUCTABLES, «RASPBERRY PI TWITTERBOT,» [En línea]. Available: <http://www.instructables.com/id/Raspberry-Pi-Twitterbot/>. [Último acceso: 25 junio 2017].
- [15] KACHARREANDO, «COVERTIR TEXTO A VOZ,» 26 ABRIL 2015. [En línea]. Available: <http://www.kacharreando.com/ubuntu/texto-voz-ubuntu/>. [Último acceso: 29 JUNIO 2017].
- [16] J. M. HUIDROBO, «LA DOMOTICA COMO SOLUCION DEL FUTURO,» ABRIL 2007. [En línea]. Available: <https://www.fenercom.com/pdf/publicaciones/la-domotica-como-solucion-de-futuro-fenercom.pdf>. [Último acceso: 07 ENERO 2017].
- [17] HBA, «SISTEMAS DE INFORMACIÓN,» 23 MARZO 2015. [En línea]. Available: <http://domotica-ubiobio.blogspot.com/2015/03/comparacion-sistema-centralizado-vs.html>. [Último acceso: 11 ENERO 2017].
- [18] ALARMTECH, «CATALAGO DE PRODUCTO CONTACTOS MAGNETICOS,» 2012. [En línea]. Available: http://www.alarmtech.se/images/MC_ES.pdf. [Último acceso: 28 ENERO 2016].
- [19] MICRONICA, «SENSORES ACTUADORES Y ELEMENTOS DEL SISTEMA DE CONTROL,» [En línea]. Available:

http://www.micronica.es/files/pdfs/SIHD/SIHD_Sens_Actu_EC.pdf. [Último acceso: 30 ENERO 2017].

- [20] ISAAC PE, «COMPARATIVA Y ANALISIS :RASPERRY PI VS COMPETENCIA,» 12 AGOSTO 2014. [En línea]. Available: <https://www.raspberrypi.org/products/model-a/>. [Último acceso: 09 FEBRERO 2017].
- [21] ADAFRUIT, «PIR MOTION SENSOR,» [En línea]. Available: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor?view=all>. [Último acceso: 24 ENERO 2017].
- [22] AFICIONADOS A LA MECANICA, «SENSORES EN EL AUTOMOVIL,» 2014. [En línea]. Available: <http://www.aficionadosalamecanica.net/sensores.htm>. [Último acceso: 24 ENERO 2017].
- [23] ELECTROBIOMEDICAL, «SENSOR DE MOVIMIENTO INFRARROJO PIR,» [En línea]. Available: <http://www.electrobiomedical.com.co/download/datasheet/SEN0013.pdf>. [Último acceso: 25 ENERO 2017].
- [24] MARTIN, «WIMSCP FREE SFTP,SCP AND FTP CLIENT WINDOWS,» 17 SEPTIEMBRE 2014. [En línea]. Available: <https://winscp.net/eng/docs/lang:es>. [Último acceso: 06 MARZO 2017].
- [25] A. RUBI, «ADOBE DREAMWEAVER, CARACTERISTICAS, VENTAJAS Y DESVENTAJAS,» 15 AGOSTO 2015. [En línea]. Available: <https://prezi.com/00bd1yffddga/adobe-dreamweaver-caracteristicas-ventajas-desventajas/>. [Último acceso: 07 MARZO 2017].
- [26] OBVIOUSIDEA, «LIGHT IMAGE RESIZER,» [En línea]. Available: <http://www.obviousidea.com/es/windows-software/light-image-resizer/>. [Último acceso: 16 MAYO 2017].
- [27] J. M. Zolezzi, «QUE SON LOS SERVIDORES WEB Y PORQUE SON NECESARIOS?,» 20 SEPTIEMBRE 2010. [En línea]. Available: <https://www.duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios>. [Último acceso: 20 FEBRERO 2017].

- [28] I. s. A. S. Ramos, «ANÁLISIS DE LOS REQUERIMIENTOS TECNOLÓGICOS PARA LA IMPLEMENTACIÓN DE SERVIDORES WEB SEGUROS,» 11 DICIEMBRE 2002. [En línea]. Available: <http://www.monografias.com/trabajos12/rete/rete.shtml>. [Último acceso: 20 FEBRERO 2017].
- [29] NECRAFT, «FEBRUARY 2016 WEB SERVER SURVEY,» 22 FEBRERO 2016. [En línea]. Available: <https://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html>. [Último acceso: 22 FEBRERO 2017].
- [30] openSUSE, «APACHE,» [En línea]. Available: <https://es.opensuse.org/Apache>. [Último acceso: 22 FEBRERO 2017].
- [31] ECURED (CONOCIMIENTOS CON TODOS Y PARA TODOS), «LENGUAJE DE PROGRAMACIÓN,» [En línea]. Available: https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n. [Último acceso: 2017 FEBRERO 27].
- [32] A. D. L. TORRE, «LENGUAJES DEL LADO DEL SERVIDOR O CLIENTE,» 2006. [En línea]. Available: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html. [Último acceso: 03 Marzo 2017].
- [33] COPYRIGHT THE PHP GROUP, «¿QUE ES PHP?,» [En línea]. Available: <http://php.net/manual/es/intro-what-is.php>. [Último acceso: 03 MARZO 2017].
- [34] M. A. ALVAREZ, «QUE ES PYTHON,» 19 NOVIEMBRE 2003. [En línea]. Available: <https://desarrolloweb.com/articulos/1325.php>. [Último acceso: 03 MARZO 2017].
- [35] I. I. E. D. M. DIGITAL, «QUE ES MYSQL: DEFINICIÓN,» [En línea]. Available: <https://iiemd.com/mysql/que-es-mysql>. [Último acceso: 06 AGOSTO 2017].
- [36] ALEGGSA.COM.AR, «DEFINICIÓN DE PHPMYADMIN,» [En línea]. Available: <http://www.alegsa.com.ar/Dic/phpmyadmin.php>. [Último acceso: 06 AGOSTO 2017].

ANEXOS

ANEXO A: Conceptos necesarios

ANEXO B: Manual de usuario

ANEXO C: Manual de mantenimiento (Características técnicas de los dispositivos y programas utilizados en el prototipo)

ANEXO D: Maqueta del prototipo.

ANEXO E: Costos de implementación.

ANEXOS

ANEXO A: Conceptos necesarios

A-1. Domótica [16]

Hace varios años ya se había pensado en soluciones para desarrollar sistemas que puedan integrar todos los equipos domésticos, esta integración tecnológica se denomina domótica. Sin embargo, domótica se define de la siguiente manera: aquella tecnología en donde existen equipos automatizados los cuales tienen la capacidad de comunicarse interactivamente entre ellos a través de un bus doméstico que los integra [1]. En la figura 4.1 se puede observar como los diferentes sistemas presentes en un domicilio se unen a un solo bus de comunicación para lograr con éste la integración de servicios.



Fig. 4.1 Definición teórica de domótica mediante la disponibilidad de un bus doméstico

La domótica, dicho en pocas palabras, es la instalación e integración de varios dispositivos eléctricos electrónicos o informáticos instalados en una vivienda, los cuales permiten la automatización de las actividades diarias permitiendo tener un control local y remoto sobre todos los dispositivos instalados en la vivienda.

La domótica se encuentra en constante desarrollo tecnológico y por lo tanto debe cumplir con las siguientes características:

Integración: debe funcionar bajo el control de un ordenador central, de esta manera el usuario no debe estar pendiente de los diversos equipos autónomos instalados en la vivienda.

Interrelación: debe ofrecer la capacidad de relacionar diferentes elementos y obtener una variedad de decisiones, es decir un sistema domótico debe tener la capacidad de trabajar entre dispositivos de diferentes fabricantes o de diferente funcionalidad.

Facilidad de uso: debe garantizar el fácil uso del sistema; por ejemplo, tan solo con mirar la pantalla del ordenador central, el usuario se enterará de los parámetros de su vivienda

como la temperatura, el estado de puertas y de ventanas, el estado de la iluminación; y tan solo con presionar sobre un display se podrá controlar el estado de los mismos.

Control remoto: el control de una vivienda domótica tiene que ser remoto para que pueda ser controlada desde cualquier parte del mundo mediante otro computador y así poder simular presencia en su hogar. Esto es de mucha utilidad cuando el usuario viaja frecuentemente.

La arquitectura de un sistema domótico define la forma en que los elementos instalados en la vivienda se conectan al sistema de control, pudiendo ser una arquitectura centralizada o distribuida. Para el proyecto se utilizó la arquitectura centralizada ya que todos los dispositivos eléctricos y electrónicos se conectan a una sola unidad central (Microcomputador Raspberry Pi 3)

- **Arquitectura centralizada:** se caracteriza por tener una sola unidad central, la cual recibe toda la información de las entradas (sensores, luces, etc.), las procesa y envía a las salidas para realizar las tareas correspondientes [17].

En el presente proyecto se utilizó la arquitectura centralizada ya que la unidad central es el microcomputador *Raspberry Pi 3*, en la figura 4.2 se puede observar la estructura de una arquitectura centralizada.

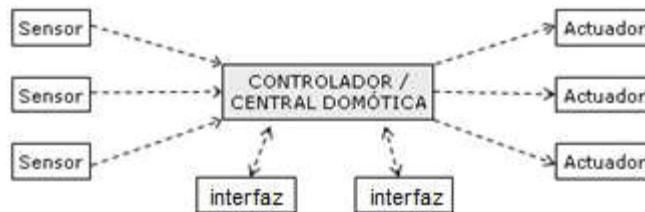


Fig. 4.2 Arquitectura domótica centralizada

Los sistemas domóticos están formados por uno o varios elementos, se puede clasificar de la siguiente manera:

Sensores: es un elemento con capacidad de captar una variación de una variable reconociendo todo su entorno. Esta variable puede ser física o química la cual es transformada en una variable eléctrica. En el actual proyecto se utilizó el sensor PIR en el anexo C-1 se muestra sus características.

Otro sensor utilizado es el sensor magnético (Anexo C-2) ya que son muy prácticos al momento de realizar instalaciones residenciales evitando dañar la estética del lugar. Este sensor está formado de un imán y un interruptor magnético, el sensor funciona como un

interruptor normalmente cerrado mientras hay campo eléctrico; cuando la puerta o ventana se abre el circuito también[18].

Controladores (Raspberry): la información obtenida por medio de los sensores sirve para que el controlador pueda tomar decisiones de realizar o no alguna acción que se va a reflejar por medio de los actuadores. [19]

Medio de transmisión: son elementos que permiten a los usuarios entregar y recibir información del sistema de control, es decir son los elementos que permiten la interrelación entre el usuario y el sistema [19].

Actuadores: son los dispositivos que siguen las órdenes del sistema de control, realizan las acciones que repercuten en el mundo real por ejemplo un foco, un motor, etc [19].

A-2. Versiones del microcomputador Raspberry Pi

RASPBERRY PI MODELO A [5]

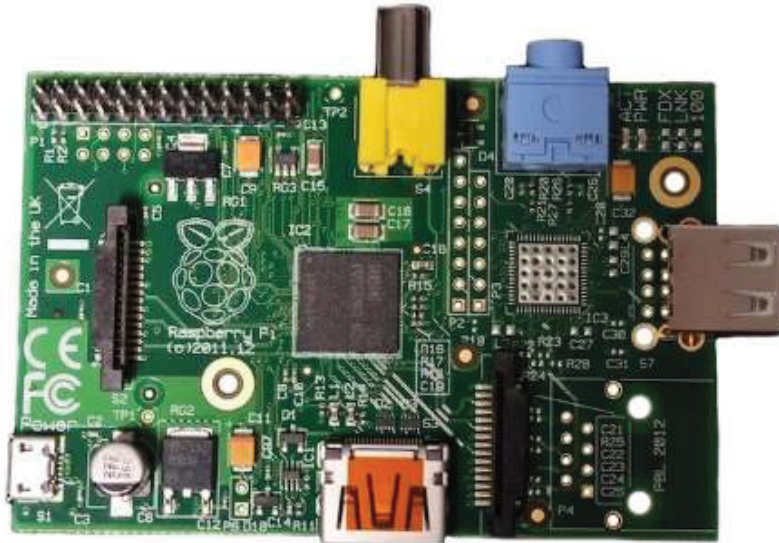


Fig. 4.3 Raspberry Pi A

En la figura 4.3 se puede observar la versión *Raspberry Pi A*, es la tarjeta más básica con tan solo 256 MB de RAM, un puerto USB y sin ningún puerto ethernet [20].

Este modelo se caracterizó por consumir menos energía en comparación con los demás modelos por ende es muy empleado en proyectos de robótica donde se requiera un consumo de energía bajo y en proyectos donde se requiera que la tarjeta tenga un peso mínimo.

RASPBERRY PI MODELO A + [5]

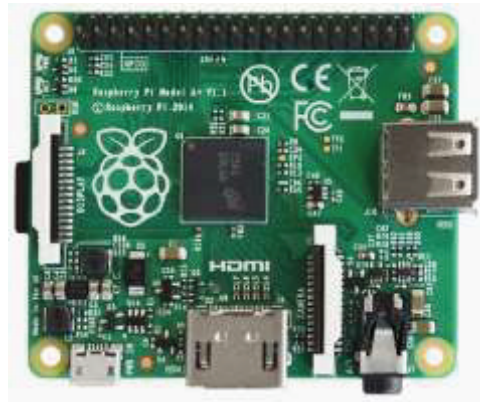


Fig. 4.4 Raspberry Pi A +

La versión que se muestra en la figura 4.4 es la versión que reemplaza a la *Raspberry Pi A*. Las mejoras con respecto a la versión A son las siguientes:

- La versión A+ cuenta con 40 pines GPIO (entradas y salidas de propósito general), a diferencia de la versión A que contaba solo con 26 puertos GPIO.
- El adaptador micro SD fue cambiado por un puerto micro SD Push- push, el cual ayuda a extraer de una manera mucho más fácil la tarjeta Micro SD.
- Se redujo el consumo de energía entre 0.5 w a 1w.
- Mejoró la calidad del audio.
- Se redujo su tamaño y se aumentó cuatro agujeros para que se pueda sujetar cuando se requiera realizar un montaje.

La versión A+ es utilizada en proyectos donde se requiera una instalación en una superficie fija con un consumo de potencia bajo. Y donde no se necesite puertos ethernet ni múltiples puertos USB [20].

RASPBERRY PI MODELO B [5]

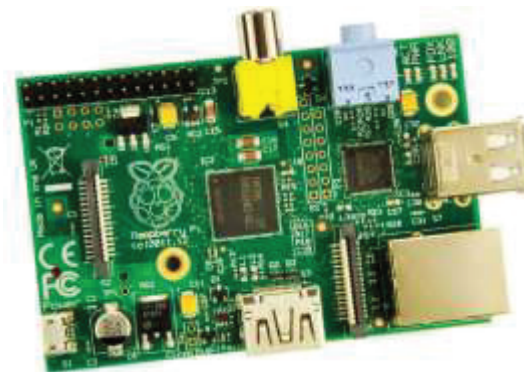


Fig. 4.5 Raspberry Pi Modelo B

Esta versión se muestra en la figura 4.5, el modelo es muy similar al modelo A, se diferencia en lo siguiente:

- Posee una memoria RAM de 512 MB.
- Posee dos puertos USB.
- Posee un puerto ethernet de 100 Mbps.
- Presenta un consumo de energía alto [20].

RASPBERRY PI MODELO B+ [5]



Fig. 4.6 Raspberry Pi Modelo B+

El modelo Raspberry B+ es el que se muestra en la figura 4.6, presenta las siguientes mejoras con respecto al modelo B:

- Cuenta con 4 puertos USB 2.0, a diferencia del modelo B que cuenta con 2 puertos USB.
- Se reemplaza el adaptador Micro SD por una versión de micro-empuje push-push.
- Reducción del consumo de energía entre 0.5W y 1W.

RASPBERRY PI 2 MODELO B [5]



Fig. 4.7 Raspberry Pi2 modelo B

Después de tener tanta competencia la compañía Raspberry necesitaba lanzar al mercado un nuevo modelo, el cual le permita competir en el mercado. Es así que se lanza al mercado la nueva versión la cual tiene como novedad 1 GB de RAM y un CPU ARM Cortex-A7 de cuatro núcleos de 900MHz. En esta tarjeta puede utilizar otros sistemas operativos más complejos como por ejemplo *Windows 10* o *Ubuntu*. En la figura 4.7 se puede observar que esta versión cuenta con una interfaz para cámara CSI, así como una interfaz para pantalla DSI [20].

RASPBERRY PI 3 MODELO B [5]



Fig. 4.8 Raspberry Pi 3 Modelo B

El modelo *Raspberry Pi 3* reemplazó a la versión *Pi2* en febrero del 2016, es la última versión lanzada al mercado por la compañía Raspberry. En la figura 4.8 se puede observar el microcomputador que se utilizara para el presente proyecto, la misma que presenta nuevas funciones como son:

- Un CPU ARMv8 quad-core de 64 bits a 1.2 GHz
- Una tarjeta LAN inalámbrica 802.11n 8
- Un módulo Bluetooth baja energía (BLE) [20]

A-3. Sensor de movimiento PIR

Los sensores PIR son utilizados para detectar movimiento, por lo general son utilizados para detectar si un humano se ha movido dentro del alcance que cubre éste tipo de sensores. Su principio se basa en la detección de la radiación infrarroja. Todos los seres vivos desprenden calor, lo mismo ocurre con las máquinas. Este calor se emite en forma de radiación infrarroja que podemos detectar con elementos adecuados como son los sensores PIR [21] [22].



Fig. 4.9 Sensor de movimiento PIR

Funcionamiento del sensor PIR

El sensor PIR cuenta con dos ranuras como se puede apreciar en la figura 4.10. Cada una de las ranuras está construida de un material especial que es sensible a la radiación infrarroja. Las dos ranuras cuentan con un lente que permite ver hasta una cierta distancia. Cuando el sensor se encuentra activo, ambos lentes detectan la misma radiación infrarroja. Cuando un cuerpo caliente (persona, animal, máquina) es detectado por el primer lente, éste provoca un cambio diferencial positivo entre los dos lentes; cuando el cuerpo caliente sale del área de detección, sucede lo contrario por lo que el sensor genera un cambio diferencial negativo. Estos cambios de positivo a negativo permiten detectar presencia, esto se puede apreciar en la figura 4.11.

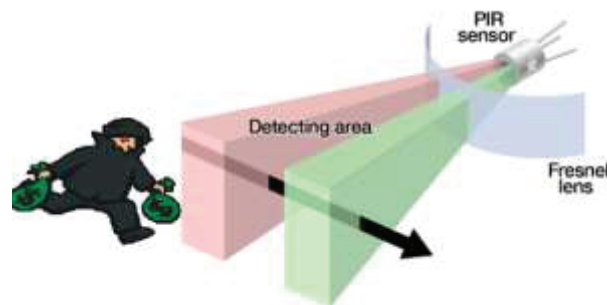


Fig. 4.10 Lentes del sensor PIR

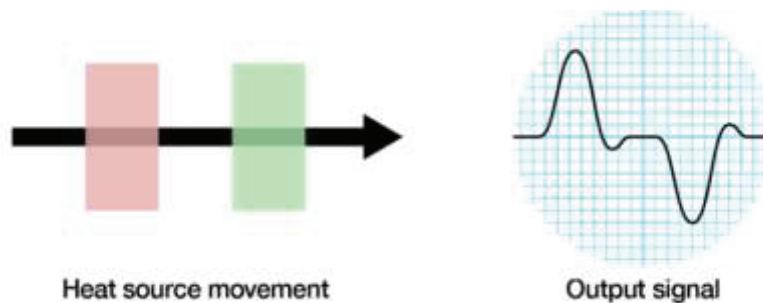


Fig. 4.11 Cambio diferencial producido por la detección de un cuerpo caliente

Esquema de conexiones

El sensor PIR cuenta con tres terminales; el pin 1 y 3 se utilizan para la alimentación como se verifica en la figura 4.12, el pin 2 es la salida de detección del movimiento la misma que es compatible con cualquier microcontrolador [23].

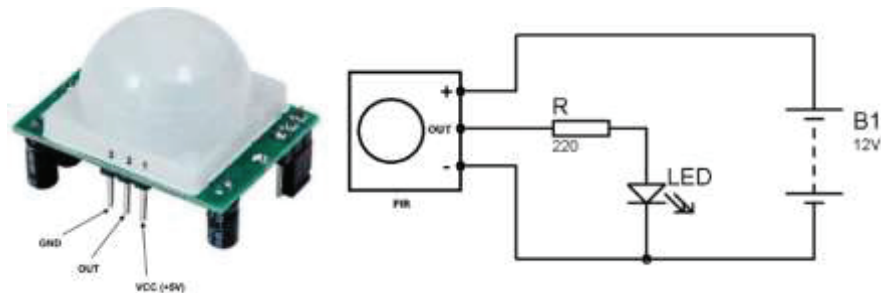


Fig. 4.12 Diagrama de pines del sensor PIR

A-4. Entradas y salidas de propósito general (GPIO) del microcomputador Raspberry Pi 3

Los pines GPIO son la interfaz entre el microcomputador Raspberry y el mundo exterior, los mismos que se pueden configurar como entradas o salidas. Al configurarse como entradas, se puede conectar interruptores o sensores, los cuales emitan señales analógicas o digitales que servirán para que el microcomputador pueda interpretar y realizar alguna acción. Mientras que si se configura los Gpio como salidas, se puede conectar leds o zumbadores. Sin embargo, si se usa un circuito especial, se puede acoplar elementos que funcionen a 110 voltios como por ejemplo bombillas, ventiladores, motores etc.

A-5. Sistemas operativos compatibles con Raspberry Pi

RASPBIAN [7]: es uno de los sistemas operativos más utilizados y conocidos que utiliza el microcomputador Raspberry el cual es basado en Debian (Linux). Este fue creado específicamente para el hardware *Raspberry Pi*, ofrece un escritorio completo y un entorno gráfico amigable con el usuario. Un sistema operativo es un conjunto de programas básicos que hacen que Raspberry funcione; sin embargo, *Raspbian* ofrece 35.000 paquetes en formato agradable.

PIDORA [7]: éste sistema operativo ofrece una experiencia más cercana y real a los ordenadores actuales; sin embargo, trae una desventaja en su rendimiento ya que a comparación con *Raspbian*, *Pidora* presenta un rendimiento muy bajo haciendo que una

página Web o algún archivo se demore en cargar varios minutos más, es compatible con el microcomputador *Raspberry Pi 2* y *3*.

ARCH GNU/LINUX ARM [7]: es un sistema operativo de software libre, proporciona una estructura de base ligera que permite moldear el sistema de acuerdo a las necesidades. El sistema operativo no cuenta con una interfaz gráfica por lo que no es recomendado para principiantes.

OPEN ELEC [7]: viene de la abreviatura Open Embedded Linux Entertainment Center. Es un software de licencia libre, ideal para proyectos de multimedia como reproductores de música o convertir la Raspberry en un cine en casa. Su principal ventaja es que está diseñado para consumir pocos recursos y arrancar rápido. Está diseñado para proyectos de multimedia y entretenimiento, por lo cual para el presente proyecto no sería conveniente utilizar dicho sistema operativo.

A-6. Descripción de los paquetes de software

Para montar la programación en el servidor, crear la página Web e instalar el sistema operativo, se hizo uso de paquetes de software que se detallan a continuación mismos que fueron utilizados en la implementación del sistema domótico:

a) Win32 Disk Imager

Esta herramienta permite sacar un respaldo del sistema operativo *Raspbian*, es una aplicación que funciona en *Windows* que permite crear copias de seguridad de la tarjeta de memoria SD, es decir permite sacar respaldos de la información. Cuenta con una interfaz gráfica, por lo cual se vuelve muy amigable con el usuario. En la figura 4.13 se muestra la interfaz gráfica.

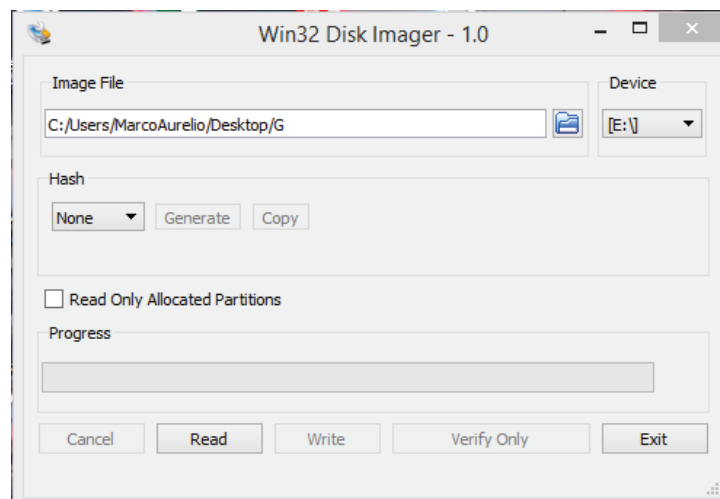


Fig. 4.13 Entorno de trabajo win32 Disk Imager

b) Dreamweaver

Es una aplicación de Adobe que tiene la finalidad de crear, diseñar o construir sitios Web. Este es uno de los programas más utilizados por los diseñadores de páginas Web por su sencillo manejo [25]. Además, sobresale de otros editores de páginas Web como Microsoft Expression Web y BlueGriffon ya que *Dreamweaver* tiene un gran poder de amplificación y personalización. Esta aplicación es multiplataforma soportada por *Windows*, *Linux* y *Mac*. Se utilizó esta herramienta ya que es de fácil uso; se puede crear páginas Web sin la necesidad de escribir líneas de código. En la figura 4.14 se puede ver la interfaz gráfica de *Dreamweaver*.

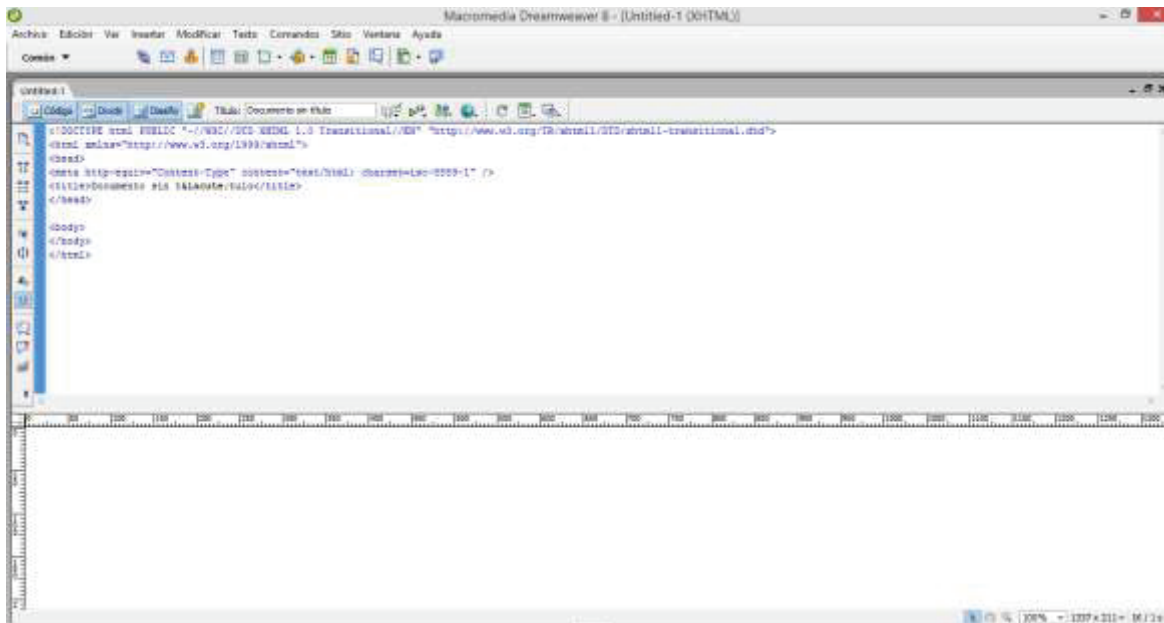


Fig. 4.14 Entorno de trabajo Dreamweaver

c) Light Resizer

Es un software utilizado para modificar el tamaño de imágenes, es de fácil instalación y está disponible para *Windows* [26]. Este programa se utilizó para crear y editar imágenes que serán cargadas en una página Web.

d) WinSCP

Es un cliente SFTP gráfico de software libre con una interfaz interactiva que funciona en *Windows*. La función principal es facilitar la transferencia segura de archivos entre dos sistemas informáticos [24]. En éste caso se utilizó para transferir los archivos de una manera gráfica hacia el servidor que se encuentra almacenado en el microcomputador *Raspberry Pi 3*

Tan solo con arrastrar el archivo hacia la pantalla correspondiente al servidor o viceversa, el archivo será transferido. Esto se puede verificar en la figura 4.15, en el lado izquierdo se tiene los documentos y archivos de la PC y en el lado derecho se encuentra las carpetas del servidor.

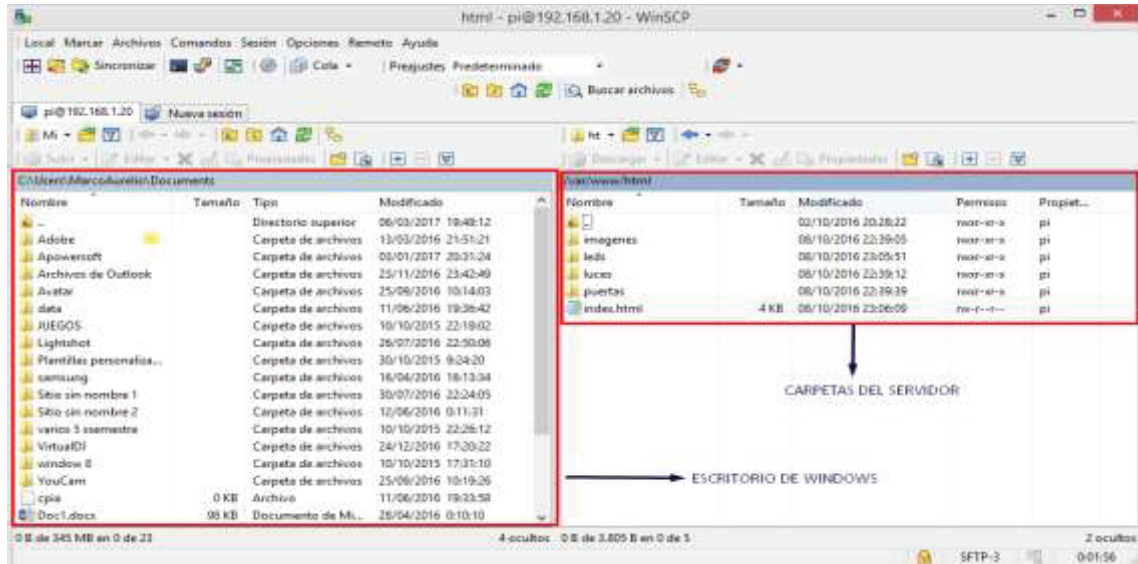


Fig. 4.15 Entorno de transferencia de archivos WinScp

A-7. Servidor Web

La tarea de los servidores Web es la de alojar sitios o aplicaciones, las cuales son requeridas o solicitadas por clientes por medio de un navegador, el cual se comunica con el servidor Web utilizando un protocolo HTTP (Hypertext Transfer Protocol). Por medio del protocolo HTTP se obtiene conexiones bidireccionales, unidireccionales síncronas y asíncronas entre el servidor y el cliente [27].

En conclusión, un servidor consta de un intérprete HTTP el cual se mantiene a la espera de algún requerimiento por parte del usuario para poder ejecutarlo y responder con el contenido o requerimiento solicitado. El cliente, una vez recibido el código requerido, lo interpreta y lo muestra en la pantalla [28].

En la figura 4.16 se muestra un diagrama básico de un servidor Web en la configuración cliente servidor, en el cual el cliente solicita una página al servidor y éste devuelve la misma en formato HTML.

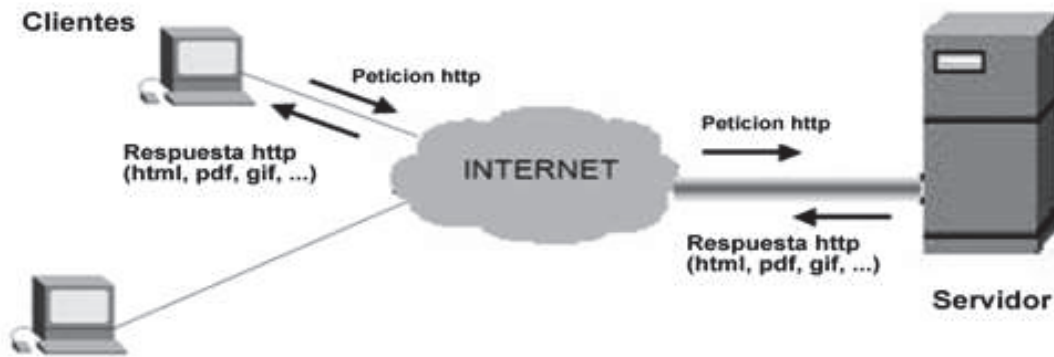


Fig. 4.16 Modelo cliente servidor

La encuesta realizada en junio del 2017 por Netcraft (compañía de análisis de servidores Web), muestra que existe una preferencia por utilizar cuatro tipos de servidores en los que se encuentran Apache Microsoft y Nginx. En la tabla 4.1 se muestra un cuadro donde se verifica el nivel de aceptación y utilización de los servidores [29].

Tabla 4.1 Porcentaje de utilización de servidores

Desarrollador	ene-16	Por ciento	feb-16	Por ciento	Cambio
Apache	405.443	40,54%	402.473	40,25%	-0,3
Nginx	290.038	29,00%	290.977	29,10%	0,09
Microsoft	99.947	9,99%	99.255	9,93%	-0,07
Google	16.545	1,65%	17.008	1,70%	0,05

A-8. Apache

Es un el servidor de software libre. En la tabla 4.1 se puede observar que Apache es el servidor que más se utiliza en Internet con un 40.25% de aceptación, según la encuesta realiza por Netcraft en julio del 2017. Este fue desarrollado para suministrar un servidor seguro, eficiente y extensible que proporciona servicios HTTP en sincronía con los estándares HTTP actuales. Su objetivo es servir o entregar páginas Web a los clientes Web o navegadores que lo solicitan; a continuación, se muestra dos características importantes de un servidor Apache [30].

- Rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Se puede ejecutar en varios sistemas operativos.

A-9. Comandos de la librería WiringPi [9]

COMANDO GPIO	FUNCION
Gpio -v	Permite conocer la versión de la librería que se encuentra instalado.
Gpio -g	Permite utilizar la numeración BCM, la cual es diferente a la numeración GPIO. Ejemplo: <code>gpio -g write 17 1</code>
Gpio -p	Funciona cuando el microcomputador Raspberry está acoplado a un piface (Placa de expansión para <i>Raspberry Pi</i>) que permite conectar motores, actuadores y sensores de forma fácil sin la necesidad de circuitos extras.
Gpio -x	Hace que el programa GPIO inicialice un módulo de expansión adicional.
<code>gpio mode <pin>in/out/pwm/clock/up/down/tri</code>	Establece el modo en el que trabajará el pin modo entrada o salida, modo pwm, modo reloj; además puede ser configurado como pull up o pull Down.
<code>gpio write <pin> 0/1</code>	Establece un pin de salida en alto (1) o bajo (0).
Gpio pwm <pin> <valor>	Para establecer un valor de modulación por ancho de pulsos, éste valor se encuentra entre 0-1023.
<code>gpio read <pin></code>	Lee y muestra el valor lógico del pin indicado 0 (bajo) o 1 (alto).
<code>gpio awrite <pin> <valor></code>	Realiza una lectura analógica del pin dado.
<code>gpio aread <pin></code>	Lee el valor analógico en el pin dado.
Gpio readall	Imprime una tabla con la numeración de los puertos (WiringPi, BCM_GPIO y números de los pines físicos).
<code>gpio allreadall</code>	Lee el estado de todos los pines (low/high)

<code>gpio wfi <pin> rising/falling/both</code>	Espera hasta que un GPIO cambie de estado
---	---

A-10. Lenguajes de programación

El lenguaje de programación se puede decir que es un idioma artificial diseñado para definir una secuencia de instrucciones que serán llevadas a cabo por máquinas como las computadoras o tarjetas inteligentes, pueden usarse para crear programas que controlen el comportamiento lógico y físico de máquinas [31].

El lenguaje en el lado del servidor se ejecuta en el servidor Web. Antes de enviar la página a través del Internet, estas páginas son capaces acceder a bases de datos, conexiones de red, etc. Los lenguajes de programación del lado del servidor más usados son: ASP, JSP, PERL Y Php.

En cuanto a los lenguajes de programación del lado del cliente, son independientes del servidor y permiten que la página pueda ser albergada en cualquier sitio por su precio cómodo ya que los servidores que aceptan la programación por el lado del servidor son más costosos. Los códigos más usados son Java y *JavaScript*, mismos que son incluidos en el código HTML.

Los lenguajes de programación que se usaron para el desarrollo del proyecto son:

Html [32]

Lenguaje de marcas de hipertexto (HTML), éste lenguaje de programación es el más básico en la creación de páginas Web. Se ejecuta en el lado del cliente, indica al navegador donde colocar cada texto, cada imagen o cada video. Funciona a base de etiquetas, es decir cada etiqueta tiene funciones específicas y diferentes como por ejemplo ubicar palabras en negrilla o justificar un párrafo; por lo general cada etiqueta tiene una etiqueta de cierre.

JavaScript [32]

Es un lenguaje de programación que funciona del lado del cliente ya que el navegador es el que soporta la carga de procesamiento. Se basa en crear efectos especiales dentro de las páginas Web. *JavaScript* tiene la ventaja de que puede ser incorporado en cualquier página Web sin la necesidad de instalar programas adicionales. Además, es soportado por la mayoría de los navegadores como Internet Explorer, Opera, Mozilla Firefox entre otros. La sintaxis de la programación es similar a la usada en Java y C.

Php

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, es rápido y tiene una gran librería de funciones. Es un lenguaje amigable y muy popular para principiantes, pero también presenta características avanzadas para programadores profesionales [33].

Sus características principales son:

- Fácil aprendizaje.
- Abundante información y documentación en su página oficial.
- Rápido.
- Lenguaje multiplataforma (*Linux, Windows, etc.*)
- Capacidad para manejar bases de datos, MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.

Python

Es un lenguaje de scripting de multiplataforma gratuito orientado a objetos; es capaz de desarrollar cualquier tipo de programa. Durante los últimos años este lenguaje ha sido muy utilizado por contar con una gran librería y documentación con información técnica muy útil al momento de desarrollar programas. Además, es sencillo y cuenta con mayor rapidez al momento de crear programas ya que se puede omitir de 3 a 5 líneas de código en comparación con Java o C [34] [29].

Algunas empresas que utilizan *Python* son Google, Walt Disney, La NASA, Yahoo!

MySQL

Es un servidor de base de datos muy conocido, desarrollado bajo la licencia dual comercial GPL/Licencia. Es un sistema de administración de bases de datos muy robusto y estable; además de manejar un sistema muy seguro con distintos niveles de seguridad. Al ser una plataforma muy conocida, existe mucha información en la web en la creación de base de datos por lo cual es muy simple y amigable para el uso del usuario. Una de las principales ventajas que trae es que puede funcionar en cualquier plataforma (*Windows, Linux, Mac*) [35].

MySQL se utilizó para crear una base de datos en la cual se guarda un usuario con su respectiva contraseña, misma que servirá para identificarse y poder acceder al sistema doméstico. Más adelante se explica el proceso de instalación y utilización.

PhpMyAdmin

Su principal función es manejar la administración de *MySQL* a través de una página Web, la cual fue desarrollada en *Php*. Permite eliminar, añadir o alterar bases de datos y se puede instalar en cualquier sistema operativo [36].

A-11. Sistema de login (autenticación)

A-11.a index.php

```
<?php
session_start();
$valido=true;
if(isset($_POST['entrar'])){
    /*Entra solo si se presiona el botón entrar*/
    //datos de acceso
    $host= "localhost";
    $usuario= "root";
    $contra= "raspberry";
    $db="usuario";
    //establecer la conexion
    $testconec= mysql_pconnect($host,$usuario,$contra) or die ("No se puede conectar con la base de
datos");
    mysql_select_db($db) or die ("No se encuentra la base de datos especificada usuario")
    $nombre=$_POST['usuario'];
    $contrasena=$_POST['contra'];
    $consulta="SELECT id, nombre,contrasena FROM user where nombre='$nombre' AND
contrasena='$contrasena'";
    $result=mysql_query($consulta) or die (mysql_error());
    $filasn= mysql_num_rows($result);
    if ($filasn<=0 || isset($_GET['nologin'])) {
        $valido=false;
    }else{
        $rowsresult=mysql_fetch_array($result);
        $_SESSION['idusuario']= $rowsresult['id'];
        $valido=true;
        //guardamos en sesion el nombre del usuario
        $_SESSION["usuario"]=$nombre;
        header("location:pagina.php?login=true");
    }
}
?>

<!DOCTYPE html>
```

```

<html>
<head>
  <title>LOGEO</title>
  <link href="css/estilo.css" rel="stylesheet" type="text/css">
  <style type="text/css">
<!--
.Estilo2 {color: #400080}
.Estilo3 {
  color: #000080;
  font-weight: bold;
}
.Estilo4 {color: #000080}
-->
  </style>
</head>

<body>
  <p align="center"><strong><u>ESCUELA POLITÉCNICA NACIONAL</u></strong></p>
  <p align="center"><u><strong>ESCUELA DE FORMACION DE TECNOLOGOS</strong></u></p>
  <p align="center" class="Estilo2"><span class="Estilo4"><strong>PROYECTO PREVIO A LA
OBTENCION DEL TITULO DE </strong><strong>TECNOLOGO EN ELECTRONICA Y
TELECOMUNICACION</strong></span><span class="Estilo2"><strong>ES</strong></span><br>
</p>
  <div id="form">
    <p><strong></strong> Ingrese los datos correspondientes </p>

    <form action="index.php" method=
      "post">
      <p>Usuario:</p>
      <input name="usuario" type="text"><br>
      <p>Contrase<input type="password"><br>
      <input name="entrar" type="submit" value="ENTRAR">
      <?php if ($valido==false) {
        echo '<p>Datos incorrectos <br/><a href="index.php">Intente de nuevo</a></p>';
      }?>
    </form>
  </div>
  <p align="center" class="Estilo3">TEMA: IMPLEMENTACION DE UN PROTOTIPO DOMOTICO PARA
SEGURIDAD Y CONTROL DE ILUMINACION DE UNA VIVIENDA UTILIZANDO UN MICROCOMPUTADOR
RASPBERRY PI 3</p>
  <p align="center">&nbsp;</p>

```

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

A-11.b página.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>ESCUELA POLITECNICA NACIONAL</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">
<STYLE type=text/css>
BODY {
    FONT-SIZE: 12px; FONT-FAMILY: Geneva, Arial, Helvetica, san-serif
}
TD {
    FONT-SIZE: 12px; FONT-FAMILY: Geneva, Arial, Helvetica, san-serif
}
.welcome {
    FONT-WEIGHT: normal; FONT-SIZE: 18px; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif
; color: #0000FF}
.heading { FONT-WEIGHT: bold; FONT-SIZE: 12px; FONT-FAMILY: Geneva, Arial, Helvetica, san-serif; color:
#333399 }
.Estilo1 {FONT-WEIGHT: bold; FONT-SIZE: 28px; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; color:
#333399; }
</STYLE>
</HEAD>
<BODY link=#333399 bgColor=#ffffff>
<TABLE cellSpacing=2 cellPadding=1 width=670 align=center border=0>
<TBODY>
<TR>
<TD height=36>
<P align=center class=Estilo1>BIENVENIDO AL CONTROL DOMOTICO </P>
</TR>
<TR>
<TD height=17><BLOCKQUOTE>&nbsp;</BLOCKQUOTE>
</TD>
</TR><tr>
<td><table width="10">
<tr>
<td width="10" class="heading">&nbsp;</td>
</tr>
</table>
<p align="center" class="heading">PRESIONE SOBRE LA ACCION QUE DESEA REALIZAR</p></td>
</tr>
```



```

<tr>
  <td><HR align=left SIZE=1></td>
</tr>
<tr>
  <td><table width="692">
    <tr>
      <td width="223" class="heading" align=center >
        <p>&nbsp;</p>
        <p class="welcome"><A href="http://192.168.1.20/luces/" class="welcome"> MOSTRAR ESTADO DE
ILUMINACION</A></p>
        <p>&nbsp;</p></td>

      <TD width="223" class="heading" align=center>
        <div align="center"><A href="http://192.168.1.20/puertas/" class="welcome">MOSTRAR ESTADO DE
PUERTAS Y VENTANAS</a></div></TD>

      <td width="224" class="heading" align=center>
        <A href="http://192.168.1.20/seguro/" class="welcome">SALIR DE CASA</a></td>
    </tr>
    <tr>
      <td>
        <p>
        </td>
      <td>
        
        </td>
      <td>
        <p> </p>
        </td>
    </tr>
  </table></td>
</tr>
<tr>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
</tr>
</TBODY>
</TABLE>
<P>&nbsp;</P>
</BODY>
</HTML>

```

A-11.c usuario.sql

```
-- phpMyAdmin SQL Dump
-- version 4.2.12deb2+deb8u2
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 10-07-2017 a las 21:30:46
-- Versión del servidor: 5.5.55-0+deb8u1
-- Versión de PHP: 5.6.30-0+deb8u1

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
-- Base de datos: `usuario`
--
-- Estructura de tabla para la tabla `user`
--
CREATE TABLE IF NOT EXISTS `user` (
  `id` int(2) NOT NULL,
  `nombre` varchar(30) NOT NULL,
  `contrasena` varchar(30) NOT NULL
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
--
-- Volcado de datos para la tabla `user`
--
INSERT INTO `user` (`id`, `nombre`, `contrasena`) VALUES
(3, 'leonardo', 'raspberry'),
(2, 'pi', 'raspberry'),
(1, 'monica', 'raspberry');
--
-- Índices para tablas volcadas
--
-- Índices de la tabla `user`
--
ALTER TABLE `user`
  ADD PRIMARY KEY (`id`);
--
-- AUTO_INCREMENT de las tablas volcadas
--
-- AUTO_INCREMENT de la tabla `user`
ALTER TABLE `user`
  MODIFY `id` int(2) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=5;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

A-12. Códigos relevantes utilizados en la programación del sistema de iluminación.

A-12.a index.php

Código completo

```
<!DOCTYPE html> <!--indicamos al navegador que utilizaremos el estándar HTML5-->

<html> <!-- Inicio del documento HTML-->
  <head> <!--Inicio de la cabecera HTML-->
    <meta charset="utf-8" /> <!--Codificación de caracteres Unicodigo-->
    <title>Raspberry Pi Gpio</title> <!-- Título de la página web-->

    <style type="text/css"> <!--inicio para la definición de los estilos */

    #centrar { <!--parámetros para centra la página web*/
width:640px;
height:640px;
margin:auto;
    }

    .Estilo1 { <!-- parámetros del modelo del titulo*/
color: #FFFFFF;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
font-style: italic;
    }

    .Estilo2 {color: #FFFFFF; font-weight: bold;} <!-- parámetros del modelo del subtítulo*/

    .Estilo3 {color: #FFFFFF;font-weight: bold;} <!-- color del botob atras*/
  </style> <!-- Final de la definición de los estilos-->

</head> <!-- Final de la cabecera HTML-->

<body style="background-color: black;"> <!--define el fondo de la página en color negro-->
```

```

<div id="centrar">
  <h1 align="center"><span class="Estilo2">CONTROL DE ILUMINACION</span></h1>
  <p align="center" class="Estilo1">Haga click sobre el botón que desee encender u apagar</p>
  <p>

<?php //inicio de la Programacion Php
  // $val_array = array(0,0,0,0,0,0);
  for ( $i= 0; $i<6; $i++) { //Este for es el encargado de generar el estado de los focos cuando
la página se inicie por primera vez

  system("gpio mode ".$i." out"); // establece los pines en modo de salida
  exec ("gpio read ".$i, $estado_foco[$i], $return ); //lee el estado de los puertos
  }
  //realiza el bucle para evaluar los estados de los pines gpio
  $i =0;
  for ( $i = 0; $i < 6; $i++) {
  // si la lectura del puerto (estado del foco) es 0 hago lo siguiente
  if ( $estado_foco[$i][0] == 0 ) {
    echo ("<img id='button_.$i.'" src='data/img/red/red_.$i.'.jpg' onclick='cambio_pin
( ".$i." );'/>"); // al ejecutar el onclick se ejecuta la función cambio pin en java script
  }
  // si la lectura del puerto (estado del foco) es 1 hago lo siguiente:
  if ( $estado_foco[$i][0] == 1 ) {
    echo ("<img id='button_.$i.'" src='data/img/green/green_.$i.'.jpg' onclick='cambio_pin
( ".$i." );'/>"); // al ejecutar el onclick se ejecuta la función cambio pin en java script
  }
  }
  }
  ?>
  <script src="script.js"></script> <!-- Llamo al script.js para que SE actualice la página web-
->
  </p>
  <blockquote>
  <p align="center" class="Estilo2"><a href="http://192.168.1.20/pagina.php"
class="Estilo1">&lt;-----REGRESAR ALA PAGINA DE INICIO----- --&gt; </A></p>
  <!--BOTON ATRAS-->
  </blockquote>
</div>
</body>
</html> // final del archivo HTML

```

A-12.b gpio.php

```
<?php
// Esta página es solicitada por el JavaScript, actualiza el estado del pin y luego lo imprime
// Obtención y uso de valores
if (isset ( $_GET["pic"] )) { // solicita el valor de pic a javascript
    $pic = strip_tags ($_GET["pic"]);
    // prueba si el valor es un número
    if ( (is_numeric($pic)) && ($pic <= 5) && ($pic >= 0) ) {

        // ajusta el modo del gpio a la salida
        system("gpio mode ".$pic." out");
        // leyendo el estado de pin
        exec ("gpio read ".$pic, $status, $return );
        // configura el gpio a alto / bajo
        if ($status[0] == "0" ) { $cambio[0] = "1"; }
        else if ($status[0] == "1" ) { $cambio[0] = "0"; }
        system("gpio write ".$pic." ".$cambio[0] );
        // leer el estado de pin
        exec ("gpio read ".$pic, $cambio1, $return );
        // imprime al cliente la respuesta
        echo($cambio1[0])
    }
    else { echo ("fail"); }
}
// error de impresión si no se pueden usar valores
else { echo ("fail"); }
?>
```

A-12.c script.js

Código completo

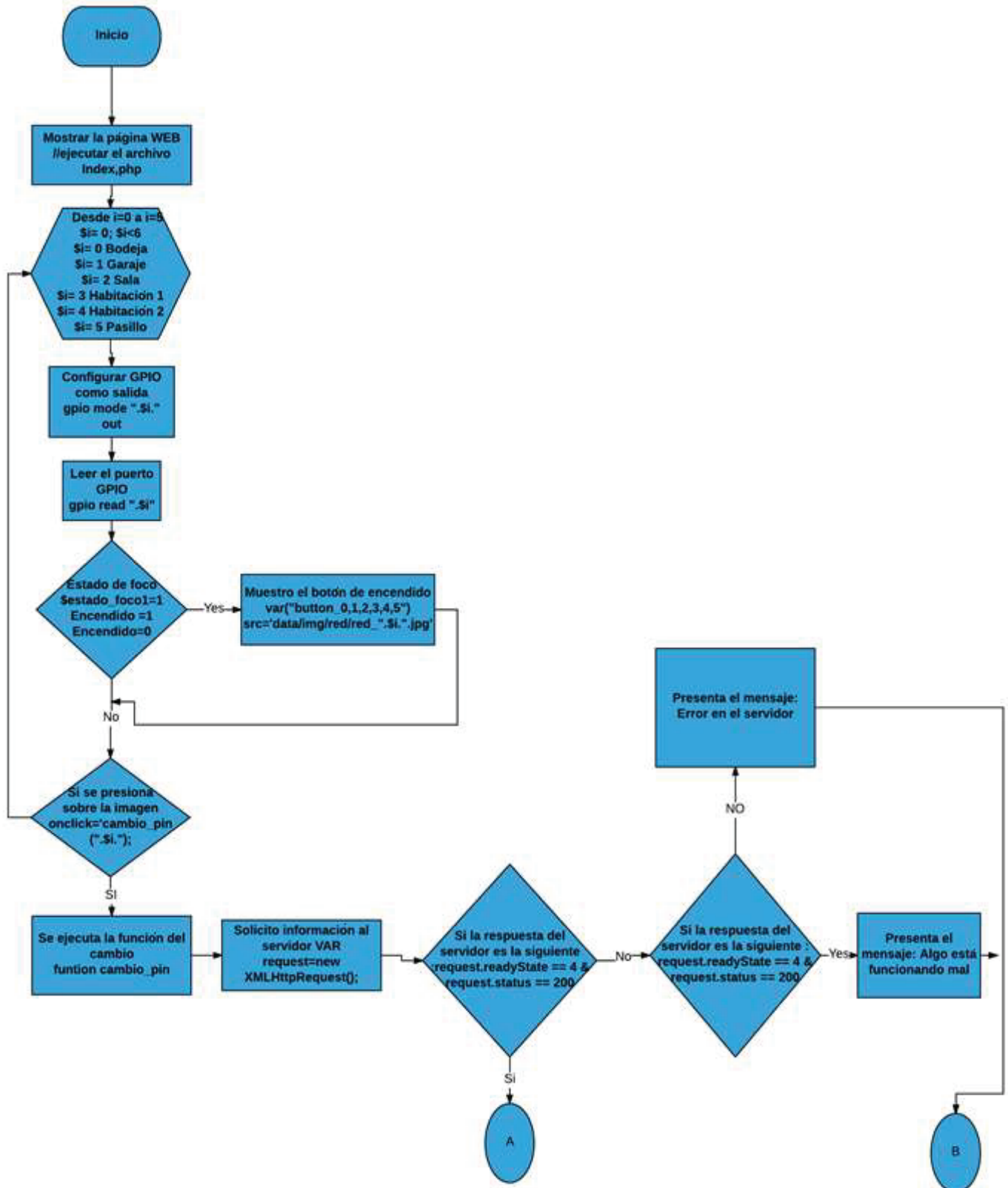
```
// JavaScript, utiliza imágenes como botones, envía y recibe valores desde y hacia el Rpi
// especifica un id a cada boton
var button_0 = document.getElementById("button_0");
var button_1 = document.getElementById("button_1");
var button_2 = document.getElementById("button_2");
var button_3 = document.getElementById("button_3");
var button_4 = document.getElementById("button_4");
var button_5 = document.getElementById("button_5");
// Crear una matriz para facilitar el acceso más adelante
var Buttons = [ button_0, button_1, button_2, button_3, button_4, button_5];
// Definimos la función (change_pin) que se encuentra en el index.php.
```

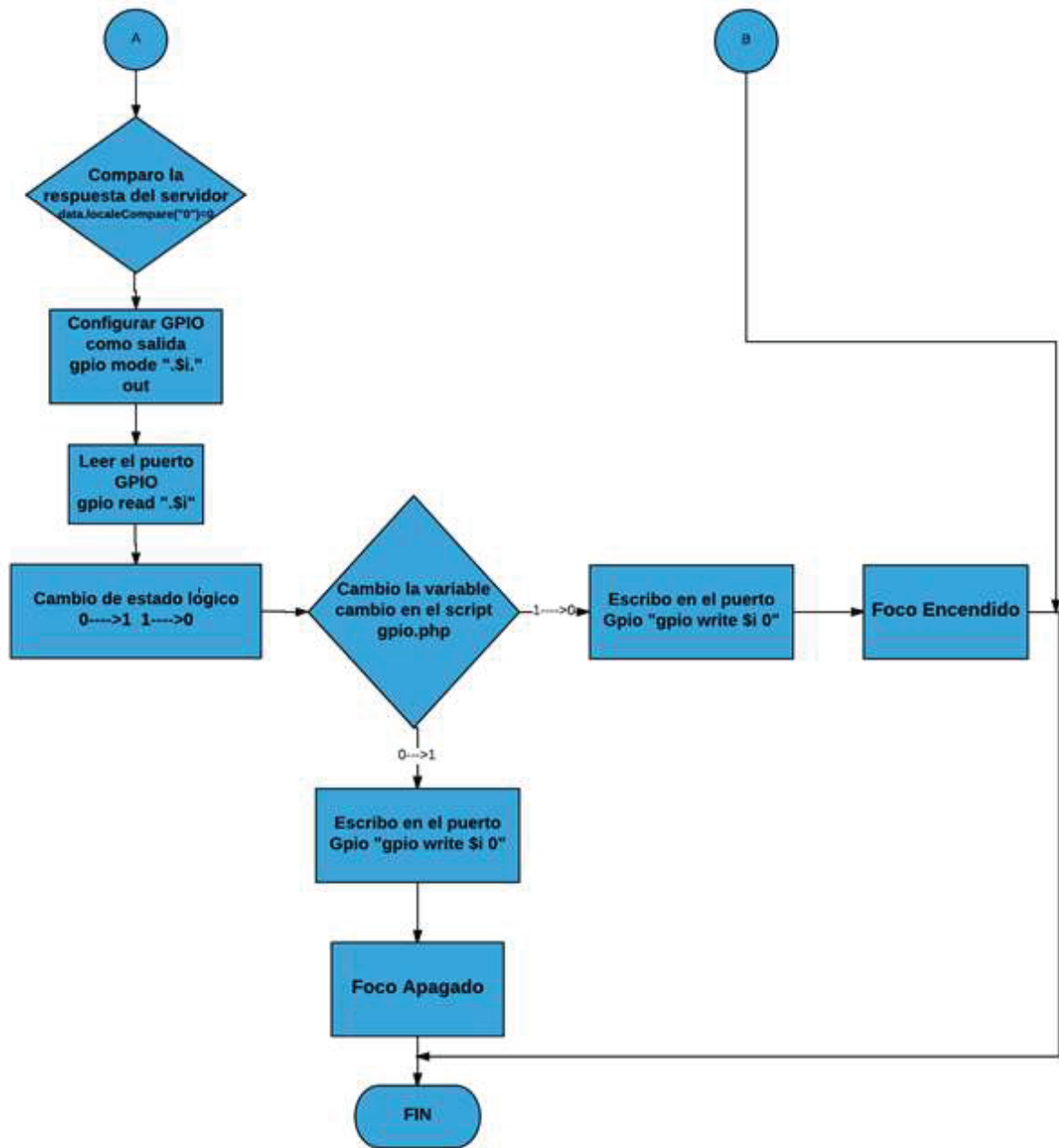
```

function cambio_pin ( pic ) {
var data = 0;
// Enviar el número de pic a gpio.php para cambios // Esta es la solicitud http
// se utiliza para solicitar y recibir datos desde un servidor web //crea un objeto XMLHttpRequest
var request = new XMLHttpRequest();
request.open( "GET" , "gpio.php?pic=" + pic, true); // indica que la solicitud debe manejarse de forma
asincrónica
request.send(null);
// recibir información
request.onreadystatechange = function () { //especifica una función que se ejecutara cada vez que cambie
el estado del objeto XMLHttpRequest //La propiedad onreadystatechange define una función que se ejecutará
cuando el readyState cambie.
if (request.readyState == 4 && request.status == 200) { //Cuando la propiedad readyState es 4 y la
propiedad status es 200, la respuesta está lista: //La propiedad readyState contiene el estado del
XMLHttpRequest.
data = request.responseText; //La propiedad responseText devuelve la respuesta del servidor como
una cadena de texto. //La cadena de texto se puede utilizar para actualizar una página web:
// actualizar la imagen de índice
if ( !(data.localeCompare("0")) ){
Buttons[pic].src = "data/img/red/red_"+pic+".jpg";
}
else if ( !(data.localeCompare("1")) ) {
Buttons[pic].src = "data/img/green/green_"+pic+".jpg";
}
else if ( !(data.localeCompare("fail"))) {
alert ("Algo esta mal! ");
return ("fail");
}
else {
alert ("Algo esta mal! ");
return ("fail");
}
}
// prueba si falla
else if (request.readyState == 4 && request.status == 500) {
alert ("error en el servidor");
return ("fail");
}
else if (request.readyState == 4 && request.status != 200 && request.status != 500 ) {
alert ("ALGO ESTA FUNCIONADO MAL");
return ("fail"); }
}
return 0;
}

```

A-12.d Diagrama de flujo del sistema de iluminación





A-13. Códigos relevantes utilizados en la programación del sistema de control de ventanas y puertas

A-13.a index.php

```

<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript">
      setInterval( "window.location.reload()", 1000);
    </script>
  </head>
</html>
  
```



```

</script>
<title>CONTROL DE VENTANAS Y PUERTAS</title>
</head>

<!-- On/Off button's picture -->
<body onLoad="redireccionar()">
<table width="1233" height="350" border="6" align="center" bordercolor="#000099" bgcolor="#FFFFFF">
<tr>
<th bordercolor="#000000" scope="col"><h1 align="center"><strong> ESTADOS</strong></h1>
<div>
<h1><strong><a href="javascript:location.reload()" class="Estilo2">ACTUALIZAR

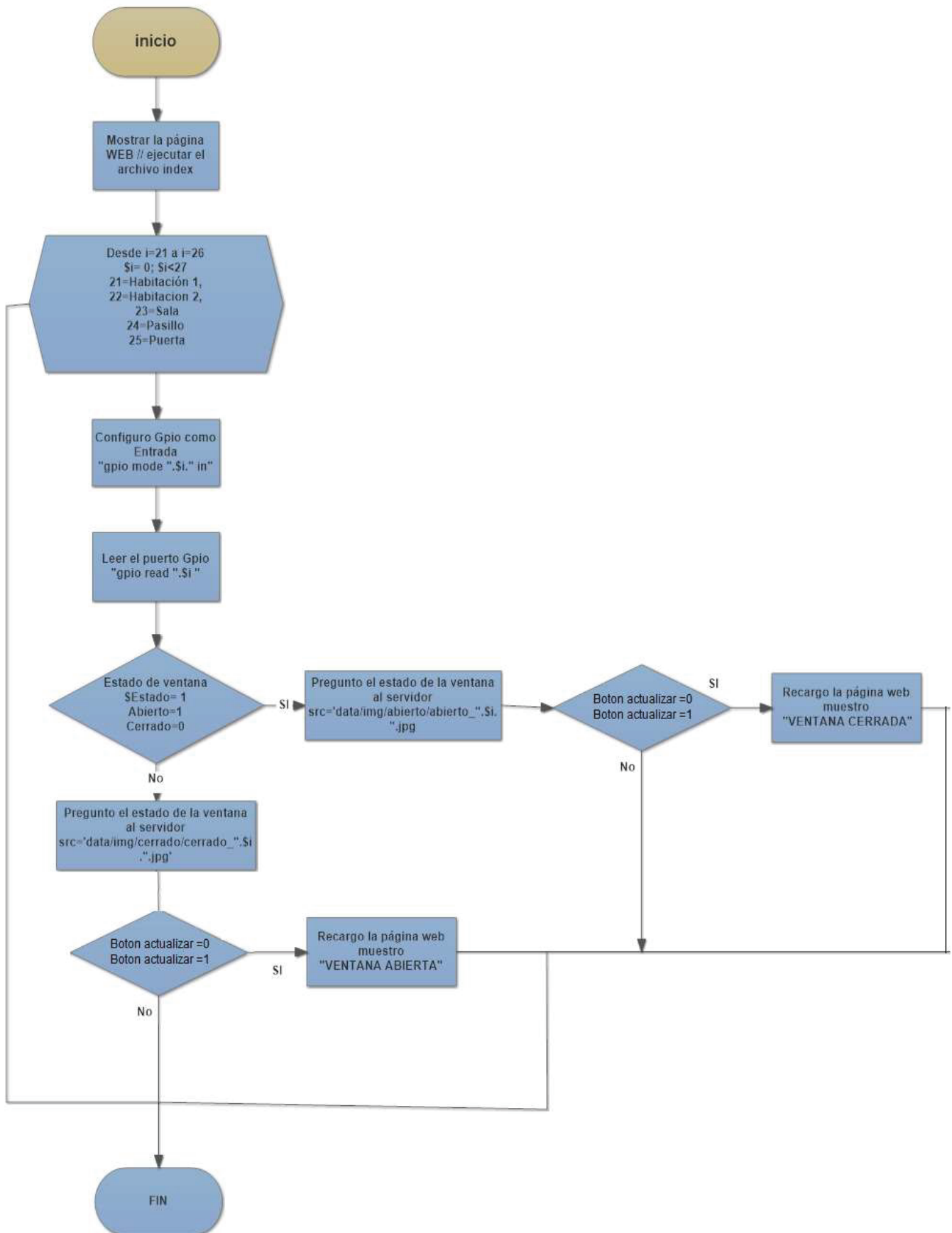
</a>
VENTANAS Y PUERTAS</strong> <span
class="Estilo2"><a href="http://192.168.1.20/pagina.php" class="Estilo2"> ---INICIO---</a></span></h1>
</div></th>
<p align="center" class="Estilo2">&nbsp;</p>
<!--BOTON ATRAS-->
</tr>
<tr>
<th height="126" scope="row">

<?php

$val_array = array(0,0,0,0,0);
//this php script generate the first page in function of the file
for ( $i= 21; $i<26; $i++) {
//set the pin's mode to output and read them
system("gpio mode ".$i." in ");
exec ("gpio read ".$i, $estado[$i], $return );
}
//for loop to read the value
for ($i = 21; $i < 26; $i++) {
//if CERRADO
if ($estado[$i][0] == 0 ) {
echo ("<img src='data/img/cerrado/cerrado_".$i.".jpg'/>");
}
//if ABIERTO
if ($estado[$i][0] == 1 ) {
echo ("<img src='data/img/abierto/abierto_".$i.".jpg'/>");
}
}
?>

```

A-13.b Diagrama de flujo del control de ventanas y puerta



A-14. Códigos relevantes utilizados en la programación del sistema de seguridad

Comandos relevantes utilizados en la programación del sistema de seguridad

A-14.a index.php

```
<html>
<head><title>prueba form</title>

<input type="button" onclick=" location.href='http://192.168.1.20' " value="regresar ala pagina inicial"
name="boton" />
<input type="button" onclick=" location.href='http://192.168.1.20:8081' " value="ver la camara en
vivo" name="boton" />
<style type="text/css">
<!--
.estilo1 {color: #ffffff}
-->
</style>
<body style="background-color: black;">
<script type="text/javascript">
<!--
function mm_swapimgrestore() { //v3.0
    var i,x,a=document.mm_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.osrc;i++) x.src=x.osrc;
}

function mm_preloadimages() { //v3.0
    var d=document; if(d.images){ if(!d.mm_p) d.mm_p=new array();
    var i,j=d.mm_p.length,a=mm_preloadimages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.mm_p[i]=new image; d.mm_p[i].src=a[i];}}
}

function mm_findobj(n, d) { //v4.01
    var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=mm_findobj(n,d.layers[i].document);
    if(!x && d.getDocumentById) x=d.getDocumentById(n); return x;
}
}
```

```

function mm_swapimage() { //v3.0
    var i,j=0,x,a=mm_swapimage.arguments; document.mm_sr=new array; for(i=0;i<(a.length-2);i+=3)
        if ((x=mm_findobj(a[i]))!=null){document.mm_sr[j++]=x; if(!x.osrc) x.osrc=x.src; x.src=a[i+2];}
    }
    //-->
</script>
<form action="index.php" method="post">
    <p class="estilo3">&nbsp; </p>
    <h1 align="center" class="estilo3 estilo1"><strong>control de alarma raspberry pi </strong></h1>
    <p align="center" class="estilo1">(haga click sobre el boton si desea activar la alarma) </p>
    <p align="center"><a href="../../../documents/sitio sin nombre 2/boton 1.png"></a><a href="#"
onmouseout="mm_swapimgrestore()"
onmouseover="mm_swapimage('image1','../../../documents/sitio sin nombre 2/boton
2.png',1)"></a></p>
    <p align="center">
        <input type="submit" name="activar" value ="activar alarma">
    </p>
    <p align="center"><span class="estilo1">(haga click si desea desactivar la alarma) </span></p>
    <p align="center"></p>
    <p align="center">
        <input type="submit" name="desactivar" value ="desactivar alarma">
    <br>
    <br>
</form>
</form>

<div align="center">
    <?php
    exec("sudo gpio mode 26 in");
    exec ("sudo gpio read 26", $rreturn );

    if ($_post[desactivar]) {
        exec("sudo python /var/www/html/seguro/apaga.py");
    }
    if ($_post[activar]) {

```

```
exec("sudo python /var/www/html/seguro/activar.py");
```

```
}
```

```
?>
```

```
<form name="formulariodatos" method="post" action="ejemplopost1.php">  
<input name="submit" type="submit" value="consulta el estado de la alarma" />  
</div>  
</body>  
</html>
```

A-14.b activar.php

```
import RPi.GPIO as GPIO # llama a la librería para Python  
import time          # sirve para igualar la hora  
import os  
from twython import Twython# libreria  
# define los pines que son de entrada y salida  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup (32, GPIO.IN) # lee el puerto 36  
GPIO.setup (36, GPIO.OUT) #led de activacion (led encendido=alarma activada)  
GPIO.setup (38, GPIO.IN) #sensor Pir pasillo  
GPIO.setup (22, GPIO.IN) #leer el puerto 40  
GPIO.setup (40, GPIO.OUT) #alarma  
GPIO.setup (29, GPIO.IN) #Magnetico Ventana  
GPIO.setup (31, GPIO.IN) #Magnetico Ventana  
GPIO.setup (33, GPIO.IN) #Magnetico Ventana  
GPIO.setup (35, GPIO.IN) #Magnetico Ventana  
GPIO.setup (37, GPIO.IN) #Magnetico Puerta  
  
# Datos de la cuenta de twitter de @raspbobot  
  
CONSUMER_KEY = '72hub7QgH2O6WQd5fZfRis8Of'  
CONSUMER_SECRET = 'OAgCqzmHwOhFR7Ecem59JTg2ipH6Ya0zL6L7bsrVBPHD2v9fvI'  
ACCESS_KEY = '1123923846-7D8N3cF32Wfe4TdnhQeR9wjU8bSMBejsutsA2YU'  
ACCESS_SECRET = 'BDgo1jPEJMH5szBsU4UKLne0cpmfVjYXdWReCgO4JgBTm'  
api = Twython(CONSUMER_KEY,CONSUMER_SECRET,ACCESS_KEY,ACCESS_SECRET) #se accede a  
twitter  
  
def sirena(tiempo):  
    GPIO.output(40, True)  
    time.sleep(tiempo) #descansar suspender  
    GPIO.output(40, False)
```

```

def desarmrAlarma():
    print "alarma desarmada"
    time.sleep(.1)

def parpadeo(numVeces):
    GPIO.output(36, True)
    while numVeces > 0:
        GPIO.output(36, True)
        time.sleep(.5)
        GPIO.output(36, False)
        time.sleep(.5)
        numVeces -=1

def enciendeLed():
    GPIO.output(36,False)

def apagarLed():
    GPIO.output(36,True)

def alarmaActivada():
    GPIO.output(40,True)

def movimientoDetectado():
    return GPIO.input(38)

def ventanahabitacion1():
    return not GPIO.input(29)

def ventanahabitacion2():
    return not GPIO.input(31)

def ventanasala():
    return not GPIO.input(33)

def ventanacuartodeequipos():
    return not GPIO.input(35)

def puerta():
    return not GPIO.input(37)

if movimientoDetectado():
    sirena (5)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar , porfavor desaloje la casa' | aplay")

```

```

elif ventanahabitacion1():
    sirena (2)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar Ventana habitacion 1 abierta' | aplay")

elif ventanahabitacion2():
    sirena (2)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar Ventana habitacion 2 abierta' | aplay")

elif ventanasala():
    sirena (2)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar Ventana sala abierta' | aplay")

elif ventanacuartodeequipos():
    sirena (2)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar Ventana cuarto de equipos abierta' |
aplay")

elif puerta():
    sirena (10)
    os.system (" espeak -ves+m3-s150 'no se puede alarmar Ventana cuarto de equipos abierta' |
aplay")
else:
    os.system (" espeak -ves+m3-s150 'La alarma raspberry se encuentra activada' | aplay")
    while True:
        apagarLed()
        if movimientoDetectado()== True or ventanahabitacion1()== True or
ventanahabitacion2()== True or ventanasala()== True or ventanacuartodeequipos()== True or puerta()==
True:
#que la camara tome fotos
        #ejecuta un mensaje sonoro por el altavoz
        os.system (" espeak -ves+m3-s150 'Intruso detectado , alerta , alerta, usted esta siendo
filmado' | aplay")
        sirena (5)
        os.system ( "sudo fswebcam -r 320x240 -S 5 --title 'ALARMA EPN' --subtitle
'MOVIMIENTO DETECTADO' --info 'Monitor: Pasillo' --save foto_alarma.jpg")
        photo = open('foto_alarma.jpg', 'rb')
        api.update_status_with_media(status='Intruso detectado por la alarma Raspberry ',
media=photo)
        sirena (5)
        os.system ( "sudo fswebcam -r 320x240 -S 5 --title 'ALARMA EPN' --subtitle
'MOVIMIENTO DETECTADO' --info 'Monitor: Pasillo' --save foto_alarma.jpg")
        photo = open('foto_alarma.jpg', 'rb')
        api.update_status_with_media(status='Intruso detectado por la alarma Raspberry ',
media=photo)

```

```

sirena (5)
os.system ( "sudo fswebcam -r 320x240 -S 5 --title 'ALARMA EPN' --subtitle
'MOVIMIENTO DETECTADO' --info 'Monitor: Pasillo' --save foto_alarma.jpg")
photo = open('foto_alarma.jpg', 'rb')
api.update_status_with_media(status='Intruso detectado por la alarma Raspberry ',
media=photo)

alarmaActivada()

```

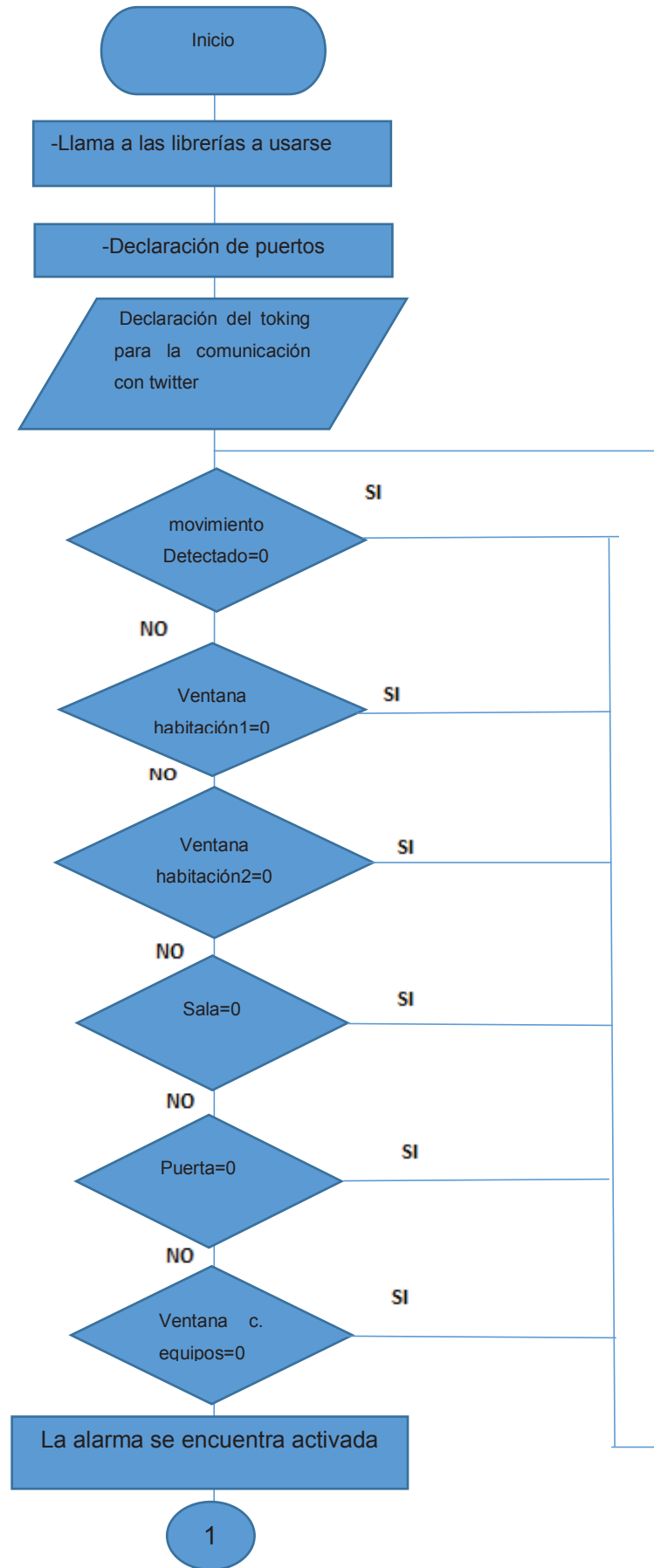
A-14.c apaga.py

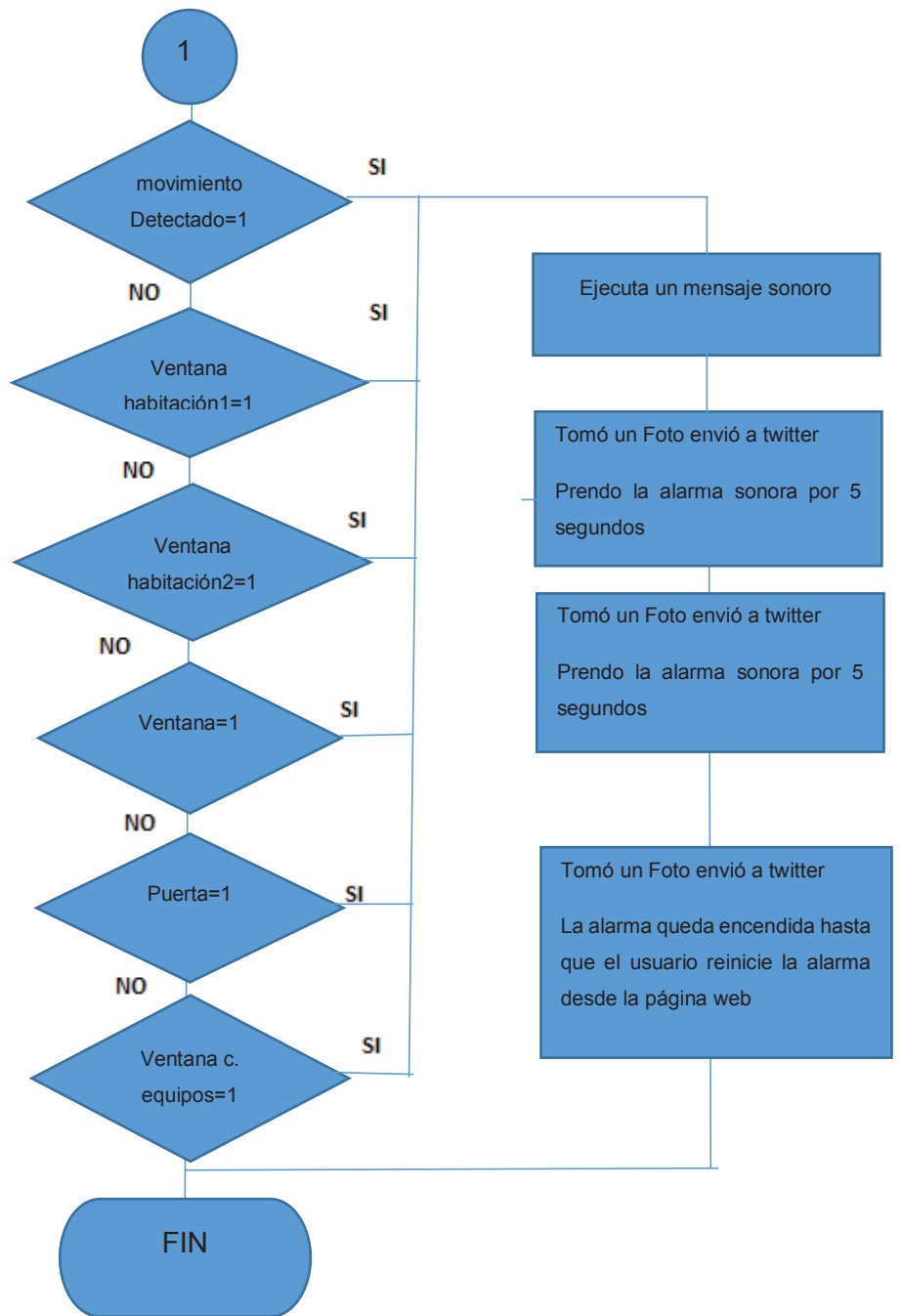
```

import time
#!/usr/bin/env/ python
import RPi.GPIO as GPIO
import sys
import os
os.system (" sudo pkill -9 -f activar.py ")
GPIO.setmode(GPIO.BOARD)
GPIO.setup (40, GPIO.OUT)
GPIO.setup (36, GPIO.OUT) #alarma
GPIO.output(40, False)
GPIO.output(36, False)
os.system (" espeak -ves+m3-s150 'Alarma desactivada' | aplay")

```


A-14.d Diagrama de flujo script activa.php





ANEXO B: Manual de usuario para el administrador del prototipo

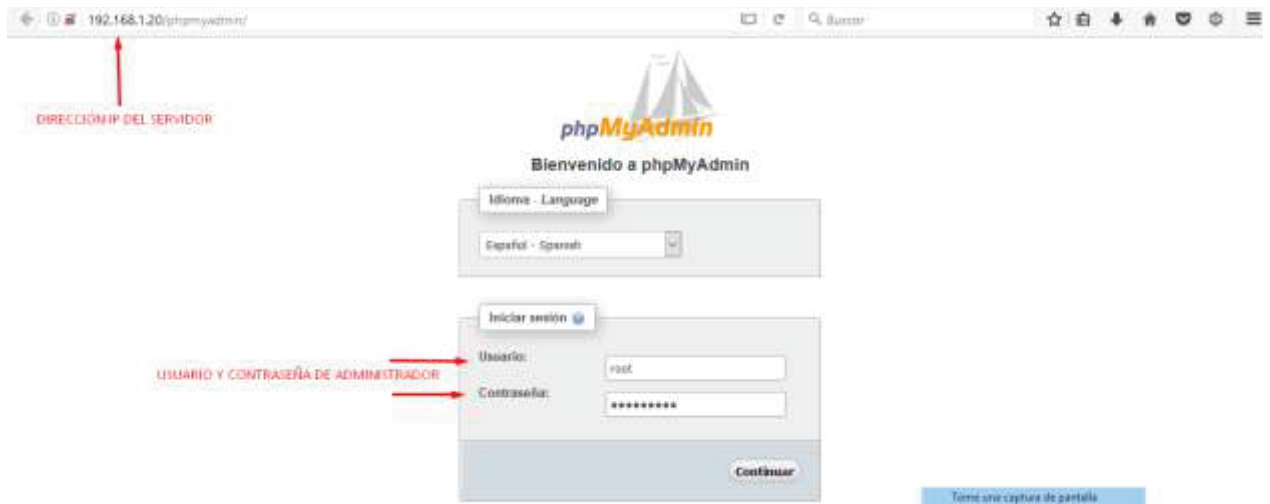
Para poder utilizar el sistema domótico es necesario abrir un navegador Web e ingresar la dirección IP que el administrador configuró en el microcomputador *Raspberry Pi 3*. En el apartado 1.6 de éste proyecto se puede verificar cómo cambiar o configurar la dirección IP estática en el microcomputador. Este prototipo se encuentra configurado con la dirección 192.168.1.20, misma que solo se puede abrir en una red WLAN. Una vez dentro de la página se debe ingresar un usuario y una contraseña.



Creación del usuario y contraseña para nuevos usuarios

El sistema puede tener varios usuarios; estos usuarios al tener autorización de acceso pueden tener el control del sistema domótico. Para poder utilizar el sistema es necesario crear los usuarios en la base de datos MySQL siguiendo los siguientes pasos:

- ✓ Ingresar al link <http://192.168.1.20/phpmyadmin/> (en caso de tener una diferente dirección Ip configurada, se debe de modificar el link).

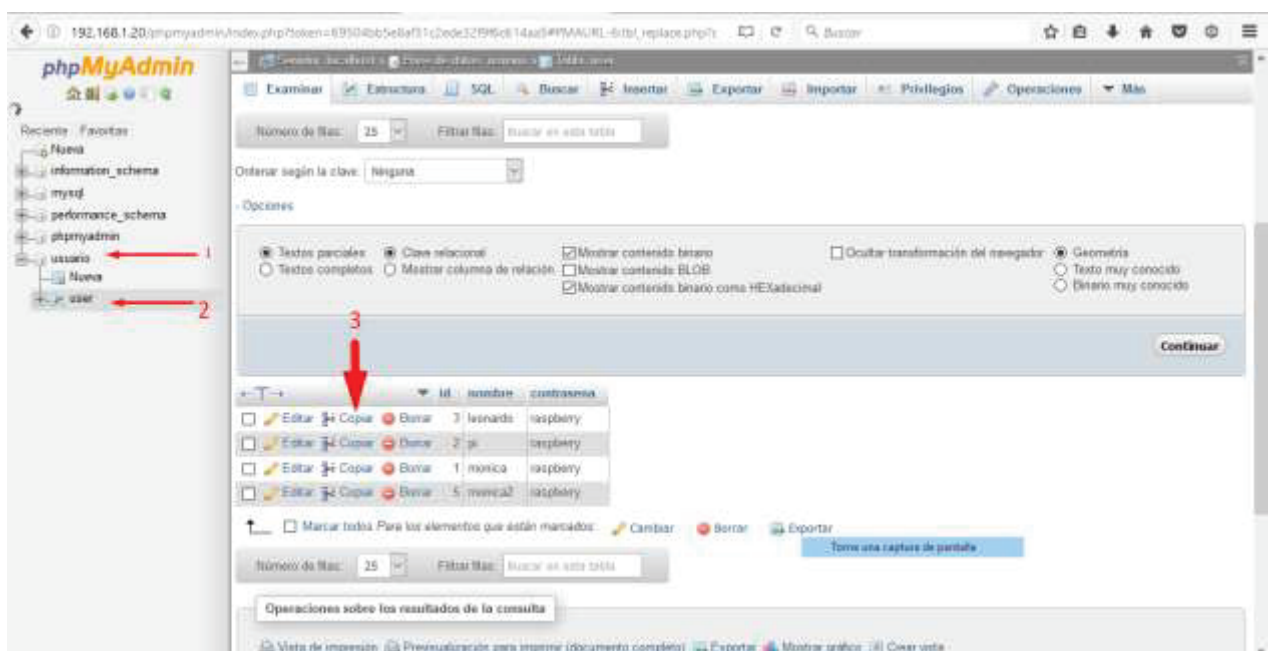


- ✓ Una vez ingresado al link pedirá un usuario y contraseña esta contraseña es la solicitada por MySQL al momento de su instalación, el presente proyecto tiene configurada la siguiente contraseña:

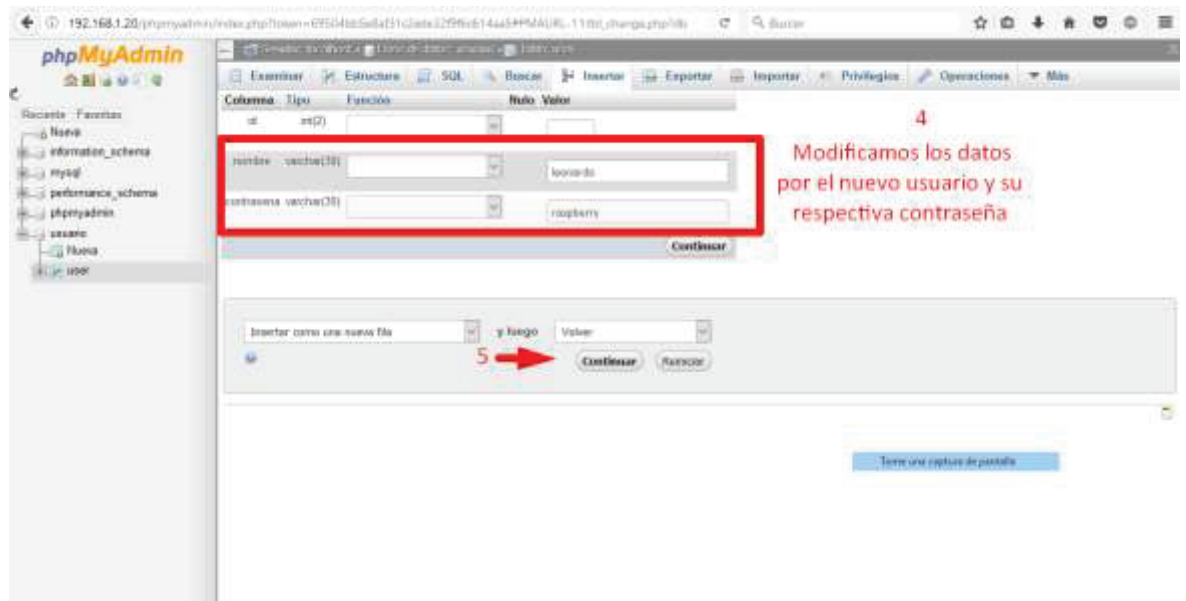
Usuario: root

Contraseña: Raspberry

- ✓ Una vez aceptada la contraseña mostrará la siguiente pantalla y se procede con los siguientes pasos:



1. Escoger la opción usuario.
2. Escoger la opción user.
3. Seleccionar “copiar” para crear otro usuario, se abrirá la siguiente imagen



4. Borrar los datos y crear el nuevo usuario.
5. Presionar continuar y finalmente comprobar en la tabla si se creó el usuario.

				id	nombre	contraseña			
<input type="checkbox"/>		Editar		Copiar		Borrar	2	pi	raspberry
<input type="checkbox"/>		Editar		Copiar		Borrar	1	monica	raspberry
<input type="checkbox"/>		Editar		Copiar		Borrar	7	leonardo	raspberry
<input type="checkbox"/>		Editar		Copiar		Borrar	6	leandro	raspberry

6. Finalmente se puede acceder al sistema domótico con el usuario y clave configurado.

ANEXO C: Manual de mantenimiento (Características de los dispositivos y programas utilizados en el prototipo).

En caso de que el prototipo presentara inconvenientes en su funcionamiento, es recomendable reiniciar el microcomputador Raspberry Pi3, esto se logra abriendo un terminal y escribiendo el siguiente comando:

```
sudo shutdown -r now
```

A continuación, se presenta las características de los dispositivos utilizados para que sirvan de guía en caso de que sea necesario su reemplazo.

C-1. Características del sensor PIR [23]

Tabla 4.2 Especificaciones técnicas sensor PIR

SENSOR PIR	
Voltaje de funcionamiento	DC 4.5V-20V
Corriente en reposo	60 uA
Salida de voltaje	Alta / Baja de nivel de señal: 3,3 V de salida TTL
Distancia de detección	3m-7m Ajustable
Angulo de detección	<110 °
Tiempo de retardo	5 A 200 Segundos
Temperatura de trabajo	-20° --- 80°
Peso de producto	6g

C-2. Características del sensor magnético [18]

Tabla 4.3 Especificaciones técnicas sensor magnético

SENSOR MAGNETICO	
Medidas	63.5mm x 13mm x 13mm
Corriente	300mA
Distancia de funcionamiento	menor a 15 a 25 mm
Potencia	3w
Tipo de interruptor	Normalmente abierto /Normalmente cerrado
Forma	Rectangular cubierto de plástico
Conexión	Cable

C-3. Pines del microcomputador *Raspberry Pi 3*

Tabla 4.4 Distribución de pines del microcomputador *Raspberry Pi 3*

Pines Raspberry Pi3			
PIN	FUNCIÓN	PIN	FUNCIÓN
1	3.3V	21	BCM 9 - MISO
2	SDA.1	22	BCM 25
3	SCL.1	23	BCM 11 - SCLK
4	GPIO.7	24	BCM 8 - CE0
5	0V	25	5
6	GPIO 0	26	5V
7	GPIO 2	27	0V
8	GPIO 3	28	BCM 1 - ID_SC
9	3.3.V	29	TXD
10	MOSI	30	RXD
11	MISO	31	GPIO.1
12	SCLK	32	CE0
13	0V	33	CE1
14	SDA.0	34	SCL.0
15	GPIO.21	35	0V
16	GPIO.22	36	GPIO.26
17	GPIO.23	37	0V
18	GPIO.24	38	GPIO.27
19	GPIO.25	39	GPIO.28
20	0V	40	GPIO.29

C-4. Características del módulo de relés 8 canales

Tabla 4.5 Características del módulo relés

Caractrísticas del Modulo Rele	
Voltaje entrada	5v
Voltaje de control	3.3 V ~ 9 V
Voltaje de salida	250 VCA o 30 VDC.
Corriente a la salida:	10 Amp
Tamaño	138mm x 56mm x 18.5mm
Peso	120g
# numero de canales	8

ANEXO D: Maqueta del prototipo

A continuación, se presentan varias fotos de la implementación del prototipo:



Fig. 4.17 Prototipo Domótico

ANEXO E : Costo de implementación

➤ Costos Directos

Son los costos empleados en materia prima.

Tabla 4.6 Costos directos

	Cantidad	Descripción	Costo unitario	Costo total (\$)
Microprocesador	1	Microcomputador <i>Raspberry Pi 3</i>	70	70 (Ecuador)
	1	Modulo relé	14	14
Actuadores	6	Focos	4	24
	5	Sensor Magnético	2.50	12.5
	1	Sirena	5	5
	1	Parlantes	12	12
	1	Elementos electrónicos	10	10
	1	Webcam	7	7
	1	Alarma sonora	12	12
	1	Baquelita	3	3
Montaje	1	Clavos, Tornillos, Boquillas	20	20
	3	Tabla triplex	14	14
	40	Pintura	5	15
	10	Cables para conexión 5v	0.10	4
	1	Cable para conexión 110v	0.40	4
	1	Extensión 110v	4.50	4.50
			Total 1	231

➤ Costos indirectos

Es el resultado de la mano de obra, diseño e ingeniería.

Tabla 4.7 Costos indirectos

Cantidad	Descripción	Costo unitario	Costo total (\$)
1	Elaboración maqueta	15	15
120	Diseño e ingeniería (horas)	10	1,200
		Total 2	1,215

➤ Costo Total

Tabla 4.8 Costo total de implementación

Componente	Costo(\$)
Costos directos	231
Costos indirectos	1,215
Total 1	1,446

La mano de obra termina siendo la más costosa ya que se consideró el tiempo de diseño para el cálculo de costos. Para un nuevo proyecto similar el precio de fabricación sería sumamente económico alrededor de los 300 dólares.

➤ **Costo de kits domóticos disponibles en el mercado**

Nombre del producto	Costo(\$)
Domótico Broadlink Hogares Kit Casa Inteligente	130
Kit De Alarma Ezviz Domótica Casa Inteligente	220
Vera Plus Casa Inteligente, Domótica, Iluminación	250

Fuente: Mercado Libre Ecuador

- **Nota:** Los precios mostrados no incluyen instalación ni tampoco accesorios extras.