

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO BASADO EN RASPBERRY PI PARA EL HOGAR

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

JEFFERSON JADIR MINALLA PEÑARRIETA

jefferson.minalla@epn.edu.ec

JORGE LUIS VALLEJO DURÁN

jorge.vallejo02@epn.edu.ec

DIRECTORA: ING. GABRIELA KATHERINE CEVALLOS SALAZAR, MSc.

gabriela.cevalloss@epn.edu.ec

CODIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ, MSc.

fabio.gonzalez@epn.edu.ec

Quito, febrero 2022

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Minalla Peñarrieta Jefferson Jadir y Vallejo Durán Jorge Luis como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



MSc. Gabriela Cevallos Salazar

DIRECTORA DEL PROYECTO



MSc. Fabio González González

CODIRECTOR DEL PROYECTO

DECLARACIÓN

Nosotros, Minalla Peñarrieta Jefferson Jadir con CI: 1724498967 y Vallejo Durán Jorge Luis con CI: 1726604448 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



Jefferson Jadir Minalla Peñarrieta



Jorge Luis Vallejo Durán

DEDICATORIA

Para mí es una gran satisfacción saber que después de invertir varios años de mi juventud en aprender lo que me apasiona, descubrí destrezas y habilidades que permitieron desarrollarme tanto personal como profesionalmente. Por tal razón, lleno de alegría dedico este proyecto a cada uno de mis seres queridos, quienes a lo largo de mi estancia en la Escuela Politécnica Nacional fueron de gran apoyo para seguir adelante.

A mis padres Rohnmel Minalla y Maryuri Peñarrieta por brindarme su cariño en cada una de las agotadoras y largas noches de estudio, donde su compañía y café eran más que suficiente para seguir adelante en busca de mis sueños. Mami y papi, simplemente gracias por ser mis padres.

A mis hermanas y abuelita por acompañarme en este arduo camino, por brindarme a cada instante su amor incondicional y sobre todo por confiar siempre en mí.

Jefferson

AGRADECIMIENTO

Al terminar esta etapa de mi vida, entre madrugadas, aglomeraciones en el transporte público para llegar a la universidad y largas jornadas de clases; siempre tuve la certeza de que todo el esfuerzo que ponía día a día en clases por educarme tendría su recompensa y no me equivoqué. Sin duda alguna, para estar en este punto de mi vida, no sólo intervino mi empeño en las aulas, también estuvo presente la ayuda de toda mi familia que, con lo poco que tienen me brindaron su ayuda económica y emocional.

Agradezco a Dios por darme la fortaleza y sabiduría necesaria para afrontar este fascinante reto de la vida universitaria; por permitirme coincidir con maestros, compañeros y amigos que, al igual que yo compartimos la pasión de la Electrónica y las Telecomunicaciones.

A mis queridos padres Rohnmel y Maryuri por criarme y forjar en mí principios y valores que siempre me acompañan y me permiten avanzar. Mami y papi, les agradezco de todo corazón por la oportunidad que me dieron de continuar con mis estudios; ya que, sin su esfuerzo y sacrificio diario por brindarme lo necesario para estudiar y estar bien, no sería posible todo lo que he conseguido hasta el momento. Quiero decirles que, este logro y los que están por venir son gracias a ustedes ya que, son el motor de mis sueños. ¡Los quiero mucho!

A mis hermanas Angie y Jeyri, quienes a pesar de tener nuestras diferencias como hermanos por situaciones de las cuales terminamos riendo, son mis mejores compañeras fuera de clases, brindándome su cariño para que juntos salgamos adelante sin importar los obstáculos que se interpongan en la travesía por alcanzar nuestras metas.

A mi abuelita Jacinta por motivarme a seguir adelante y dejarme en claro que, el conocimiento es el único recurso para enfrentarme a la vida laboral.

A mis compañeros y amigos que conseguí a largo de la carrera; con quienes compartimos alegrías y nos motivamos en los momentos tristes, con el fin de enfrentar juntos los retos que nos presenta la vida.

Jefferson

DEDICATORIA

El siguiente proyecto de titulación va dedicado para toda mi familia quienes con mucho afecto me han ayudado a conseguir una de las metas más importantes en mi vida, brindándome además todo tipo de apoyo, especialmente en los momentos difíciles que se fueron suscitando en los últimos años.

Jorge

AGRADECIMIENTO

Agradezco especialmente a mis padres por haberme brindado su apoyo incondicional en todo momento. Gracias a sus sabios y certeros consejos he podido superar las distintas adversidades que fueron apareciendo en mi camino, sin ellos no habría podido conseguir todo lo que he logrado hasta el momento.

De igual forma, doy gracias a los docentes de la Escuela de Formación de Tecnólogos quienes contribuyeron enormemente impartiendo sus conocimientos en los últimos años para de esta manera poder formarme como futuro profesional del Ecuador.

Finalmente, me gustaría agradecer a mis amigos por ser parte de una de las mejores etapas de mi vida. Les doy gracias por haberme apoyado y por haber compartido junto conmigo tanto los buenos como los malos momentos del día a día.

Jorge

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN.....	1
1.1	Objetivo general	2
1.2	Objetivos específicos.....	2
1.3	Fundamentos.....	2
	<i>Raspberry Pi 3</i> Modelo B+	2
	Entradas y salidas (GPIO) del miniordenador <i>Raspberry Pi 3</i>	3
	<i>Raspbian</i>	5
	<i>Wake on LAN (WoL)</i>	6
	SONOFF.....	6
	Módulo ESP32.....	6
	A4988	8
	Infraestructura LAMP	9
	VNC <i>Viewer</i>	10
2	METODOLOGÍA.....	11
2.1	Descripción de la metodología usada	11
3	RESULTADOS Y DISCUSIÓN	13
3.1	Determinación de la arquitectura y dispositivos apropiados para el desarrollo del sistema domótico.....	13
	Control remoto de la iluminación.....	14
	Control remoto de persiana	16
	Control remoto de ordenadores	19
	Control de acceso.....	20
3.2	Desarrollo del algoritmo de control para automatizar los distintos equipos e instalaciones del hogar.....	23
	Algoritmo para accionamiento remoto de computadoras	23
	Algoritmo para accionamiento remoto de la puerta.....	30
	Algoritmo para el control remoto de la persiana	33
	Algoritmo para el control remoto de la iluminación.....	35

3.3	Diseño de la interfaz de usuario para el manejo del sistema domótico	36
	Sistema de registro de usuarios.....	37
	Sistema de inicio de sesión	40
	Menú principal	43
3.4	Implementación del sistema domótico	47
	Implementación del sistema en las persianas.....	47
	Implementación del encendido y apagado de la luz.....	49
	Implementación del encendido y apagado de computadoras.....	52
	Implementación del control de acceso	53
	Presupuesto del proyecto	54
3.5	Verificación del funcionamiento del sistema domótico	55
	Verificación del encendido y apagado de computadores	55
	Verificación del control de acceso.....	57
	Verificación del control de la iluminación	58
4	CONCLUSIONES Y RECOMENDACIONES	60
4.1	Conclusiones.....	60
4.2	Recomendaciones.....	61
5	REFERENCIAS BIBLIOGRÁFICA.....	62
	ANEXOS.....	64
	Anexo 1: Certificado de Funcionamiento.....	i
	Anexo 2: Manual de usuario para el administrador de prototipo.....	ii
	Anexo 3: Manual de mantenimiento del sistema domótico.....	v
	Anexo 4: Algoritmos de programación	vi
	Sistema de registro e inicio de sesión.....	vi
	Index.php (Anexo 4.1).....	vi
	Database.php (Anexo 4.2)	vii
	Login.php (Anexo 4.3).....	viii
	Logout.php (Anexo 4.4)	ix
	Register.php (Anexo 4.5)	x

Estilos.css (Anexo 4.6)	xi
Header.php (Anexo 4.7).....	xii
MENU PRINCIPAL	xiii
Menu-principal.html (Anexo 4.8)	xiii
Styles.css (Anexo 4.9)	xv
COMPUTADORES	xviii
Wakeonlan.html (Anexo 4.10).....	xviii
Encender.php (Anexo 4.11)	xix
Apagar.php (Anexo 4.12).....	xx
On_all.php (Anexo 4.13).....	xx
Off_all.php (Anexo 4.14).....	xx
Ping.php (Anexo 4.15)	xxi
Styles.css (Anexo 4.16)	xxii
PUERTA.....	xxiii
Detección_facial.ino (Anexo 4.17)	xxiii
PERSIANA	xxv
Persiana.ino Anexo (4.18)	xxv
DIAGRAMAS DE FLUJO	xxx
Diagrama de interfaz Anexo (4.19)	xxx
Diagrama del menú Anexo (4.20)	xxxi
Diagrama de computadoras Anexo (4.21).....	xxxi
Diagrama de monitoreo de computadores Anexo (4.22)	xxxii
Diagrama de persianas Anexo (4.23).....	xxxii
Diagrama de luz Anexo (4.24)	xxxiii
Diagrama de puerta Anexo (4.25).....	xxxiv

ÍNDICE DE FIGURAS

Figura 1.1	Ubicación de pines en la Raspberry Pi.....	3
Figura 1.2	Distribución y enumeración de pines en la Raspberry Pi.....	4
Figura 1.3	Arquitectura general de una tarjeta Raspberry Pi 3B +.....	5
Figura 1.4	Raspberry Pi OS (Raspbian).....	5
Figura 1.5	Módulo Wi-Fi ESP32	7
Figura 1.6	Driver A4988	8
Figura 1.7	Servidor Web Apache	9
Figura 1.8	PHP MyAdmin.....	10
Figura 1.9	Conexión a VNC	11
Figura 3.1	Arquitectura centralizada.....	14
Figura 3.2	Diagrama de conexión para interruptor convencional.....	15
Figura 3.3	Diagrama de conexión para interruptor inteligente SONOFF.....	15
Figura 3.4	Interruptor inteligente SONOFF CH1.....	16
Figura 3.5	Motor paso a paso Nema 17	17
Figura 3.6	Módulo ESP32.....	17
Figura 3.7	Torque versus velocidad	18
Figura 3.8	Diagrama de conexión con ESP32 para el control de motores paso a paso	18
Figura 3.9	Controlador A4988	19
Figura 3.10	Conexión Wake on LAN.....	20
Figura 3.11	Módulo ESP32-CAM	21
Figura 3.12	Diagrama de conexión con ESP32-CAM para manejo de la puerta.....	22
Figura 3.13	Diagrama de conexión en proteus.....	22
Figura 3.14	Asignación de IP estática	23
Figura 3.15	Activar WOL en placa madre.....	24
Figura 3.16	Habilitar encendido con paquete mágico.....	24
Figura 3.17	Permitir encendido mediante paquete mágico.....	25
Figura 3.18	Desactivación de contraseña para uso compartido	26
Figura 3.19	Asignación de derechos de usuario.....	27
Figura 3.20	Configuración de reglas de entrada.....	27
Figura 3.21	Modificación en registro de Windows	28
Figura 3.22	Apagado de forma remota.....	29
Figura 3.23	Matriz de monitoreo de estado	30
Figura 3.24	Parámetros de compilación.....	31

Figura 3.25	Conexión ESP32 y RS-232	32
Figura 3.26	Conexión a TTL proteus	32
Figura 3.27	Dirección IP asignada al ESP32-CAM.....	33
Figura 3.28	Página principal.....	37
Figura 3.29	Estructura de datos	38
Figura 3.30	Registro en Web	39
Figura 3.31	Usuarios registrados	40
Figura 3.32	Inicio de sesión	42
Figura 3.33	Mensaje de bienvenida	43
Figura 3.34	Menú.....	43
Figura 3.35	Página Web para control de computadoras.....	44
Figura 3.36	Página Web para el control de persiana.....	45
Figura 3.37	Interfaz de encendido y apagado SONOFF.....	46
Figura 3.38	Página Web ESP32-CAM	46
Figura 3.39	URL para instalar la tarjeta ESP32.....	47
Figura 3.40	Selección de módulo ESP32 Dev Module	47
Figura 3.41	Persiana.....	48
Figura 3.42	Motor paso a paso Nema 17	48
Figura 3.43	Controlador A4988	49
Figura 3.44	Distribución de pines en placa SONOFF	50
Figura 3.45	Conexión de conversor de USB a TTL con placa de SONOFF	50
Figura 3.46	Instalación convencional de un foco en el hogar	51
Figura 3.47	Instalación del interruptor inteligente SONOFF	51
Figura 3.48	Interruptor SONOFF CH1	52
Figura 3.49	Cableado entre el computador y el router.....	53
Figura 3.50	Solenoides instalados	53
Figura 3.51	ESP32-CAM colocado en la parte externa	54
Figura 3.52	Verificación de registro.....	55
Figura 3.53	Verificación de encendido	56
Figura 3.54	Verificación de todas las computadoras encendidas	56
Figura 3.55	Verificación de apagado.....	57
Figura 3.56	Verificación de todas las computadoras apagadas.....	57
Figura 3.57	Verificación de seguridad de la puerta	58
Figura 3.58	Apagado remoto de interruptor SONOFF	58
Figura 3.59	Encendido remoto de interruptor SONOFF	59
Figura 3.60	Envío de peticiones al motor paso a paso	59

ÍNDICE DE TABLAS

Tabla 3.1 Costos de materiales y mano de obra de la implementación	54
---	----

RESUMEN

El propósito del presente proyecto es el diseño e implementación de un sistema domótico dirigido a los miembros de un hogar que se encuentren dentro de la vivienda, en el que a través de un dispositivo móvil u ordenador conectado a la red interna de la residencia puedan: controlar la iluminación, apagar y encender remotamente las computadoras, controlar persianas y controlar el acceso a la puerta de ingreso del domicilio.

El diseño e implementación del sistema domótico, comprende los fundamentos teóricos tanto de las características como de las funcionalidades que ofrecen cada uno de los dispositivos electrónicos usados en el proyecto. Además, se detalla paso a paso la configuración de estos a través de programación para su correspondiente integración con el sistema domótico global creado.

La implementación del sistema domótico está constituida principalmente por un miniordenador *Raspberry Pi 3* mismo que opera como la unidad central de procesamiento, encargado de recibir y enviar la información a los sensores distribuidos alrededor de la vivienda. Sobre la *Raspberry Pi* se encuentra montado un servidor *Web* a través del cual se ha creado una página *Web* la cual involucra un sistema de acceso o *login* mediante una base de datos empleando *MySQL* y un menú de navegación que permite redirigir a cuatro páginas para comandar cada una las tareas anteriormente mencionadas. Las páginas *Web* que conforman la interfaz de usuario han sido creadas a través de lenguajes de programación como: *PHP*, *C++* y el lenguaje de marcado *HTML*, permitiendo de esta manera ofrecer una interacción completa entre el usuario final y el sistema domótico.

Al culminar el proyecto, se verifica la funcionalidad del sistema, montado en varias partes de una vivienda permitiendo de esta manera cerciorarse del funcionamiento y del cumplimiento de los objetivos propuestos. Adicionalmente, se detallan las conclusiones y recomendaciones del prototipo domótico.

PALABRAS CLAVE: Sistema domótico, *Raspberry Pi*, *MySQL*, interfaz *Web*.

ABSTRACT

The purpose of this project is the design and implementation of a home automation system aimed at the members of a household, in which through a mobile device or computer connected to the internal network of the residence they can: control lighting, remotely turn computers on and off, control blinds and control access to the front door of the home.

The design and implementation of the home automation system includes the theoretical foundations of both the characteristics and the functionalities offered by each of the electronic devices used in the project. In addition, the configuration of these is detailed step by step through programming for their corresponding integration with the global home automation system created.

The implementation of the home automation system is mainly constituted by a Raspberry Pi 3 minicomputer that operates as the central processing unit, in charge of receiving and sending the information to the sensors distributed around the house. A Web server is mounted on the Raspberry Pi and shows a Web page that has been created and involves an access or login system through a database using MySQL and a navigation menu that allows redirecting to four pages which command each one of the above-mentioned tasks. The Web pages that make up the user interface have been created through programming languages such as: PHP, C ++ and the HTML markup language, thus allowing a complete interaction between the end user and the home automation system.

At the end of the project, the functionality of the system is verified, assembled in various parts of a house, thus making sure that it is working and meeting the proposed objectives. Additionally, the conclusions and recommendations of the home automation prototype are detailed.

KEYWORDS: *Home automation system, Raspberry Pi, MySQL, Web interface.*

1 INTRODUCCIÓN

En la actualidad, la domótica involucra una serie de tecnologías que controlan de manera inteligente diversos aparatos e instalaciones tradicionales como los que conforman un hogar, lugar de trabajo, entre otros [1]. Este sistema permite llevar a cabo una gestión más eficiente de la energía, disponer de mayor seguridad y dar una respuesta rápida a los requerimientos de los usuarios, otorgando una mayor comodidad y bienestar en sus espacios [2].

Raspberry Pi es ideal para desarrollar e implementar este tipo de tecnologías. Este dispositivo es un miniordenador que, a diferencia del resto de placas de desarrollo existentes en el mercado, es muy potente y económico para realizar tareas que requieran de un mayor grado de complejidad y creatividad [3]. Posee un poder significativo de cómputo en su tamaño reducido, pues permite llevar a cabo aplicaciones extraordinarias y posee la facilidad de conectarse a Internet. Por tal razón, esta placa posibilita el desarrollo de un sistema que comande al resto de los artefactos y equipos que se encuentren en un determinado lugar por medio de la red. Adicionalmente, esta minicomputadora es adaptable y ampliable en cuanto a sus prestaciones según las necesidades cambiantes que se requieran [3].

El uso continuo de los dispositivos eléctricos y electrónicos genera una problemática con respecto a la gestión de la energía necesaria para el funcionamiento de estos. Además, dedicar tiempo para manipularlos diariamente de forma manual puede llegar a resultar monótono, por lo que se busca automatizar estas actividades cotidianas para el ahorro de tiempo y energía. Consecuentemente, aspectos relacionados a la infraestructura tales como: manejo manual de iluminación, seguridad y el control automático de persianas son factores que serán tomados en cuenta para lograr un estado de confort en los miembros del hogar.

Por tal motivo, se pretende establecer un espacio de convivencia adecuado y eficiente que vaya acorde a las actividades de cada uno de los miembros que viven en el domicilio. Esto obliga a desarrollar un sistema compacto e inteligente que permita minimizar los efectos negativos de seguridad y eficiencia energética. Además, es pertinente considerar la comodidad del sitio donde los integrantes del hogar no solo descansan; también, en muchos casos realizan sus actividades laborales a distancia por motivos de pandemia.

1.1 Objetivo general

Implementar un sistema domótico basado en *Raspberry Pi* para el hogar.

1.2 Objetivos específicos

- Determinar la arquitectura y dispositivos apropiados para el desarrollo del sistema domótico.
- Desarrollar un algoritmo de control que permita automatizar los distintos equipos e instalaciones del hogar.
- Diseñar una interfaz de usuario para el manejo del sistema domótico.
- Implementar el sistema domótico.
- Verificar el funcionamiento del sistema domótico.

1.3 Fundamentos

Raspberry Pi 3 Modelo B+

El microcomputador *Raspberry Pi* es un ordenador de placa reducida en el que, a pesar de su pequeño tamaño, presenta todos los puertos que incorpora un computador de escritorio estándar. Las características que presenta este modelo en concreto son las siguientes [4] :

- Procesador *Broadcom* BCM2837B0 64-bit ARM Cortex-A53 Quad Core SoC @ 1.4 (GHz).
- 1 (GB) RAM LPDDR2 SDRAM.
- 4 x puertos USB2.0 con salida de hasta 1.2 (A).
- Puerto GPIO de 40 pines extendido.
- Salida de video/audio a través del conector de 3.5 (mm) de 4 polos, HDMI, cámara CSI o LCD sin procesar (DSI).
- Almacenamiento: microSD.
- Gigabit Ethernet sobre USB 2.0. Máximo Rendimiento 300 (Mbps).
- LAN inalámbrica 2.4 (GHz) y 5 (GHz) IEEE 802.11.b/g/n/ac, *Bluetooth* 4.2, BLE.
- H.264, MPEG-4 *decode* (1080p30); H.264 *encode* (1080p30); OpenGL ES 1.1, 2.0 *graphics*.
- Requerimientos de energía: 5 (V) @ 2.5 (A) vía fuente de alimentación micro USB.
- Soporta *Raspbian*, *Windows* 10 IoT Core, OpenELEC, OSMC, *Arch Linux*, entre otros.

- Dimensiones: 85 (mm) x 56 (mm) x 17 (mm).

Adicionalmente, la selección de esta versión entre toda la gama que posee este miniordenador abarca en su mayoría el nivel de procesamiento que tiene la Raspberry Pi 3B+ llegando incluso a una velocidad de procesamiento de 1.4 (GHz) permitiendo de esta manera levantar sistemas operativos con gran demanda de recursos y evitar inconvenientes al momento de desarrollar aplicaciones sobre este fascinante computador. Todos los recursos a nivel de hardware de esta versión de Raspberry Pi y la cantidad de puertos disponibles sobre la tarjeta facilitan el desarrollo de aplicaciones que habitualmente se llevan a cabo en ordenadores de escritorio. Además, este modelo del miniordenador en comparación al más actual como la versión 4, es más estable a nivel de aplicaciones y sistema operativo debido a su tiempo en el mercado.

Entradas y salidas (GPIO) del miniordenador *Raspberry Pi 3*

Una característica importante de la *Raspberry Pi* es la fila de pines GPIO (entrada/salida de propósito general) ubicada en la parte lateral derecha del miniordenador. La *Raspberry Pi 3B+* tal y como se muestra en la Figura 1.1, cuenta con 40 pines a lo largo de la placa, mismos que pueden ser programados para comandar dispositivos conectados directamente a los pines de este ordenador [5].



Figura 1.1 Ubicación de pines en la *Raspberry Pi*

Es fundamental comprender a cómo referirse a los pines del miniordenador al momento de realizar la programación. Existen dos maneras de hacer referencia a cada uno de ellos, tal y como se explica a continuación:

- **Numeración GPIO:** esta numeración es la que comprende el minordenador al momento de realizar la programación [5].
- **Numeración física:** es una manera sencilla de referirse a los pines donde simplemente se debe enumerar de izquierda a derecha de forma ordenada [5].

En la Figura 1.2, se muestra la distribución de cada uno de los pines en la *Raspberry Pi* a través del terminal en el sistema operativo *Raspbian*. Con ayuda del comando **pinout** se puede observar tanto la numeración física (recuadro azul) como la numeración GPIO (recuadros amarillos).

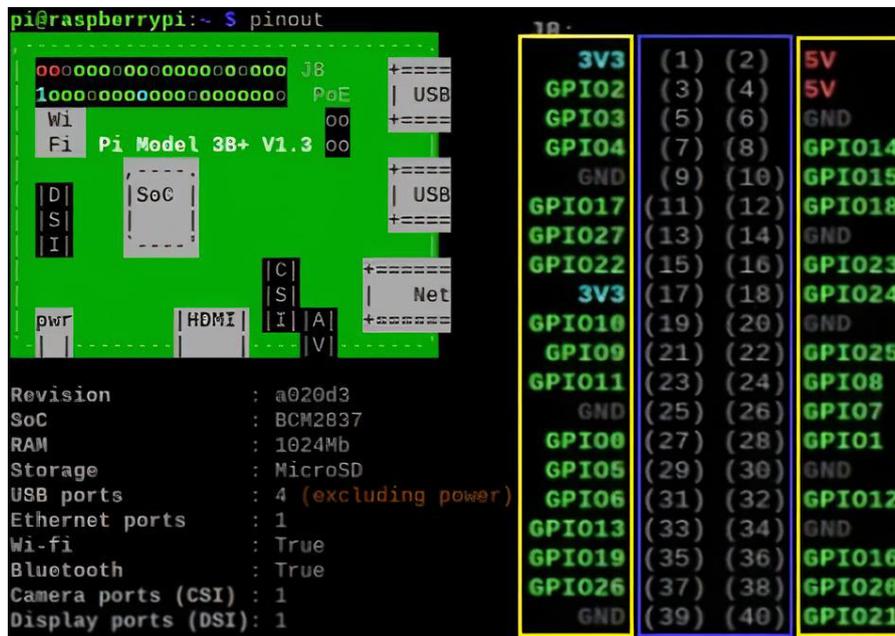


Figura 1.2 Distribución y enumeración de pines en la *Raspberry Pi*

Por otra parte, una de las redes soportadas por la *Raspberry Pi* y a su vez la principal para poder desarrollar el sistema domótico es la red inalámbrica, debido a que el miniordenador cuenta con un chip que controla a los módulos tanto de *Wi-Fi* como de *Bluetooth*. Dentro de las particularidades de esta placa se presenta la posibilidad de conectarse a redes de 5 (GHz). A continuación, en la Figura 1.3 se puede observar la arquitectura de la *Raspberry Pi* [6].

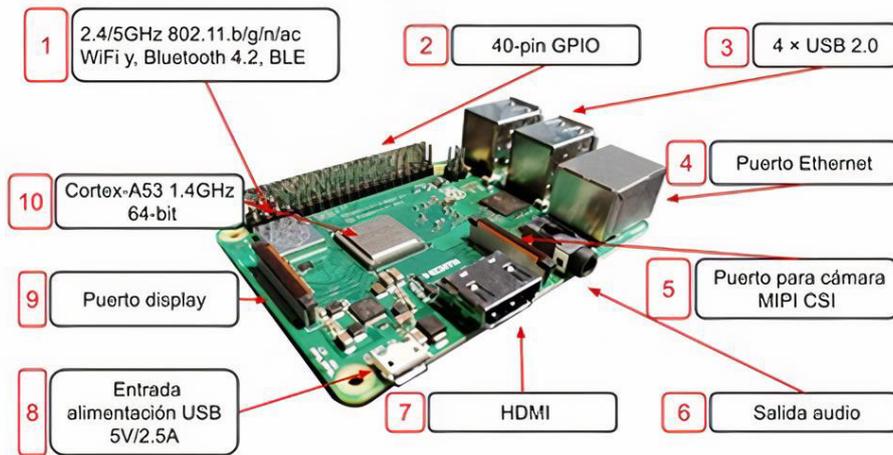


Figura 1.3 Arquitectura general de una tarjeta *Raspberry Pi 3B +* [6]

Raspbian

Raspbian es una distribución de *Linux* basada en Debian que por excelencia y casi por defecto es utilizada en *Raspberry Pi*. Este sistema operativo es completamente adecuado y optimizado para el *hardware* de esta tarjeta. Tiene un peso de 500 (MB) hasta 2 (GB) dependiendo de la versión a utilizar. Dentro de este sistema operativo se realizarán las diferentes configuraciones para poder conectar al miniordenador mediante *Wi-Fi* además de realizar la instalación de diferentes aplicaciones que permitirán convertirlo en un servidor *Web* para el desarrollo del proyecto [7]. En la Figura 1.4 se puede apreciar la interfaz de trabajo de esta distribución de *Linux*.

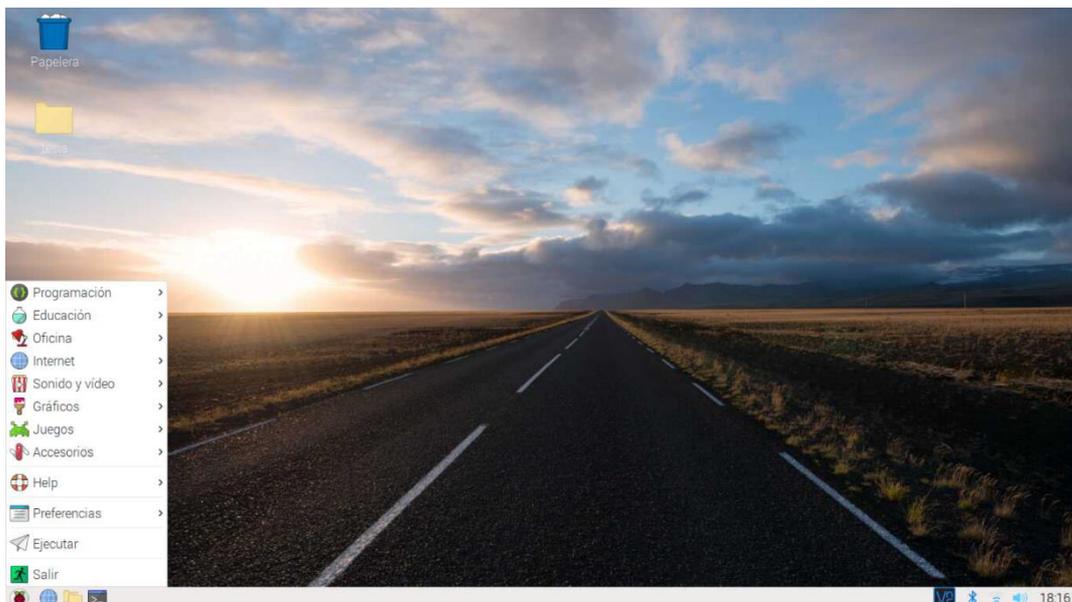


Figura 1.4 *Raspberry Pi OS (Raspbian)*

Wake on LAN (WoL)

Es un protocolo de redes de área local, el cual permite encender el computador de forma remota. Es importante considerar que este protocolo debe activarse desde la *BIOS* del computador cuando se desee usar y posteriormente activar determinados parámetros adicionales dentro del sistema operativo [8].

Cuando el PC se apaga, el adaptador de red todavía recibe energía y sigue escuchando en la red. Mediante esta tecnología *WoL*, el computador arranca gracias al uso de “paquetes mágicos”, los cuales están formados por una cadena de 6 bytes de valor 255 en lenguaje hexadecimal (FF FF FF FF FF FF) acompañada de 16 repeticiones de la dirección MAC del ordenador respectivo [8]. La mayoría de las aplicaciones de administración *WoL* transmiten el paquete como unidifusión UDP (*User Datagram Protocol*).

Por último, cuando la tarjeta de red del ordenador detecta este paquete y comprueba que la MAC es la suya, envía una señal eléctrica a la placa madre, dando paso a que el mismo arranque [8].

SONOFF

Es un *switch* inteligente controlado por la red inalámbrica *Wi-Fi* el cual permite encender o apagar aparatos electrónicos, los cuales normalmente son controlados de forma manual con algún tipo de interruptor común como: lámparas, televisores, calefactores, entre otros. Este dispositivo puede ser integrado a una interfaz *Web* o aplicación que posibilita accionar los aparatos eléctricos o electrónicos de forma remota desde un celular, *tablet* o computador [9].

Módulo ESP32

Es un *chip* de la familia SoC (*System on Chip*) que forma parte de un grupo de módulos de bajo precio y consumo energético. Su uso está orientado para el Internet de las cosas (*IoT*) a pesar de que puede ser aplicado en diversas áreas debido a su alto rendimiento. Ofrece capacidades inalámbricas tales como conectividad *Bluetooth* y *Wi-Fi*, admitiendo su control a distancia. Para su desarrollo cuenta con gran variedad de *software*, lenguajes de programación, *frameworks*, librerías, códigos y otros recursos. Sus características técnicas son las siguientes [10]:

- Voltaje de alimentación (USB): 5 (V) DC.
- Voltaje de entradas/salidas: 3.3 (V) DC.

- CPU principal: Tensilica Xtensa 32-(bit) LX6.
- Frecuencia de reloj: hasta 240 (Mhz).
- *Wi-Fi*: 802.11 b/g/n/e/i.
- *Bluetooth*: v4.2 BR/EDR y *Bluetooth* Low Energy (BLE).
- Memoria: 48 (KB) ROM, 520 (KB) SRAM 16 (KB) SRAM in RTC y QSPI Flash/SRAM.
- Pines Digitales GPIO: 32 (algunos pines solo como entrada).
- Conversor analógico Digital: Dos ADC de 12 (bits) tipo SAR, soporta mediciones en hasta 18 canales, algunos pines soportan un amplificador con ganancia programable.
- Chip USB-Serial: CP2102.
- Antena en PCB.

En la Figura 1.5, se observa la distribución de pines del módulo ESP32 en el que se indica tanto los pines de alimentación, GND (*ground*) y GPIO. El módulo puede ser programado con distintos lenguajes de programación tales como C/C++ y Python usando los entornos de desarrollo como el IDE de Arduino.

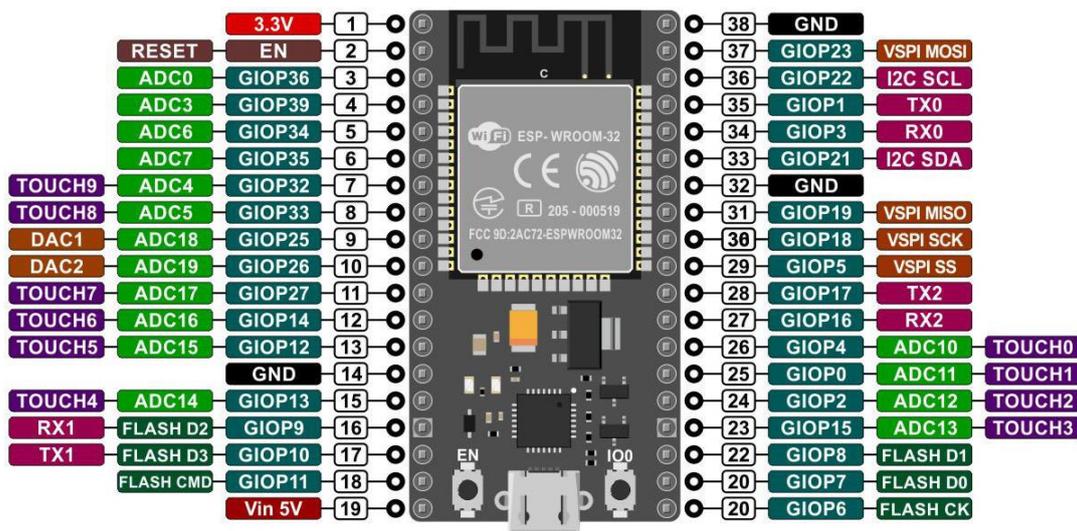


Figura 1.5 Módulo *Wi-Fi* ESP32 [13]

A4988

El controlador de motor paso a paso A4988 tiene una gran capacidad de salida de hasta 35 (V) y ± 1.5 (A) lo que le permite controlar un motor paso a paso bipolar con una corriente de salida de hasta 1.5 (A) por bobina, como el motor Nema 17 [11].

El controlador A4988 tiene un total de 16 pines. La distribución de los pines se muestra a continuación en la Figura 1.6.

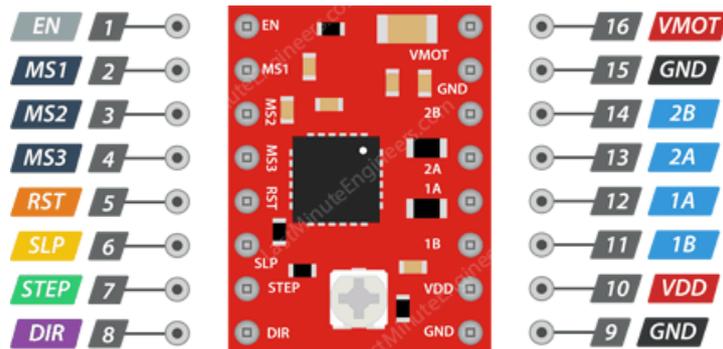


Figura 1.6 Driver A4988 [11]

La funcionalidad de cada uno de los pines del controlador A4988 son las siguientes:

- **VDD y GND:** Se utiliza para controlar los circuitos lógicos internos que pueden ser de 3 (V) a 5.5 (V).
- **VMOT y GND:** Suministra energía al motor paso a paso, puede ser de 8 (V) a 35 (V).
- **MS1. MS2 y MS3:** Posee tres entradas para seleccionar el tamaño de micro pasos (resolución).
- **STEP (PASO):** Controla los pasos del motor. Cada pulso alto enviado a este pin acciona el motor según el número de pasos establecido por el fabricante.
- **DIR (DIRECCIÓN):** Controla la dirección de giro del motor. Mientras el pin recibe un pulso alto el sentido de giro es horario mientras que un pulso bajo el motor gira en sentido antihorario.
- **EN (ENABLE):** Mientras el pin se encuentre en estado lógico bajo estará encendido el *chip*.
- **SLP (SLEEP):** Al enviar un pulso alto, el controlador se pone en modo de suspensión.
- **RST (RESET):** En estado lógico alto, se reestablece el inicio del número de pasos establecido en la programación del motor paso a paso.

Infraestructura LAMP

LAMP es un acrónimo de:

- *Linux*: el cual actúa como sistema operativo.
- *Apache*: mismo que es el servidor *Web*.
- *MySQL/MariaDB*: el cual es el gestor de bases de datos.
- *PHP/Perl/Python*: que se refiere al lenguaje de programación empleado.

El uso integral de todas las aplicaciones mencionadas anteriormente forma la infraestructura total y la base de operación del proyecto. Para la instalación de dichas aplicaciones en el sistema operativo *Raspbian* es necesario emplear el comando:

```
sudo apt-get update
```

Este comando actualiza el repositorio de archivos para poder descargar las versiones más recientes de cada *software*. Para instalar el servidor *Web* apache se emplea el siguiente comando:

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

Una vez realizada la instalación respectiva del servidor *Web* Apache, para verificar el correcto funcionamiento de este, se debe ingresar la dirección IP de la *Raspberry Pi* en el navegador y se debe verificar que aparezca la imagen mostrada en la Figura 1.7.

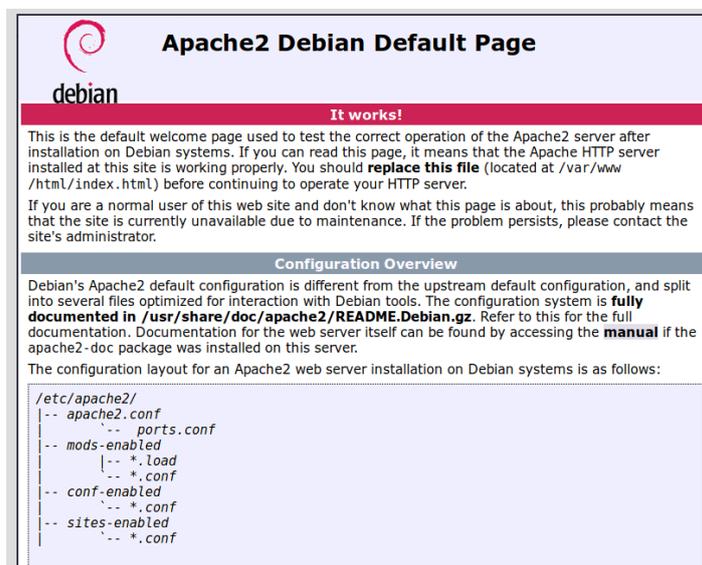


Figura 1.7 Servidor *Web* Apache

Para cambiar la página de inicio del servidor *Web*, se debe modificar el archivo `index.html` e ingresar a la siguiente dirección mediante el terminal de *Raspbian*:

```
sudo nano /var/www/html/index.html
```

En versiones anteriores, se utilizaba MySQL para el gestor de base de datos, sin embargo ya no es posible instalarlo en sistemas operativos *Raspbian* más recientes. Actualmente se instala MariaDB para este tipo de aplicación. El comando necesario es el siguiente:

```
sudo apt-get install mariadb-server mariadb-client
```

Una vez realizada la instalación y configuraciones necesarias de la base de datos, es necesario ingresar a la dirección IP de la *Raspberry* seguida de php-myadmin y aparecerá una ventana de inicio de sesión como se muestra en la Figura 1.8.



Figura 1.8 PHP MyAdmin

Finalmente, para la configuración del lenguaje de programación, que en este caso es PHP, se utiliza el siguiente comando:

```
sudo apt-get install php libapache2-mod-php
```

VNC Viewer

Es un *software* de tipo cliente servidor que controla la *Raspberry* de forma remota. Establece la comunicación entre dos dispositivos, de los cuales uno cumple la función de servidor, el cliente o los clientes observan el contenido del mismo. Con esta aplicación se crea la conexión remota para visualizar el escritorio de la máquina que

ejecuta el servidor, además de controlarla. En *Windows* no es posible conectar a un escritorio virtual, pero si el servidor se ejecuta en una PC con *Linux*, si puede conectarse a este.

Para controlar de forma remota la *Raspberry Pi* mediante un ordenador con *Windows*, se requiere ingresar la dirección IP de la misma, tal como se muestra en la Figura 1.9. Además es indispensable ingresar las credenciales de administrador y contraseña de la misma como se mencionó anteriormente. El puerto para la conexión remota es el 5900.

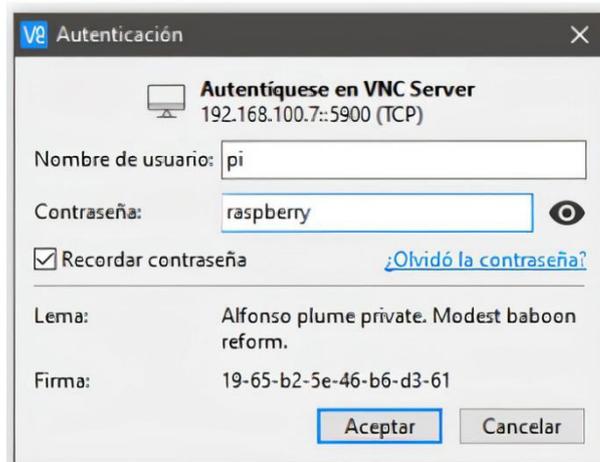


Figura 1.9 Conexión a VNC

2 METODOLOGÍA

Para la elaboración del proyecto integrador se utilizó un procedimiento de desarrollo enfocado en el usuario, a través del desarrollo paralelo del sistema domótico y la interfaz *Web* de control para los usuarios mediante el uso de la *Raspberry Pi*, que permitiera minimizar los esfuerzos de control de cada uno de los dispositivos que conforman el hogar y el manejo adecuado de los recursos energéticos que se tiene a disposición.

2.1 Descripción de la metodología usada

Los tipos de investigación utilizados fueron: aplicada y exploratoria. Aplicada debido a que al concluir el proyecto se tiene la implementación de un sistema domótico donde se evidencia el cumplimiento de los objetivos planteados en base a los conocimientos adquiridos durante la permanencia en la Escuela Politécnica Nacional - ESFOT.

Asimismo, se utilizó una investigación exploratoria puesto que, a pesar de encontrar varios proyectos domóticos en Internet, la mayoría de estos funcionan con aplicativos y dispositivos inteligentes diseñados para su correspondiente integración y muy pocos de

estos proyectos emplean *Raspberry Pi* para automatizar los diferentes dispositivos en el hogar.

Para comprender el manejo del *hardware* y *software* del minicomputador al igual que de los distintos sensores y actuadores utilizados en el desarrollo del proyecto, se utilizó el método de recolección de información ya que en base a documentos tecnológicos digitales se adquirió un mayor enfoque en cuanto a las funcionalidades y prestaciones que la *Raspberry Pi* puede ofrecer para realizar el prototipo domótico.

Una vez estudiadas las características del miniordenador se llevó a cabo una investigación para decidir tanto el sistema operativo como el tipo de servidor *Web* que será instalado sobre el miniordenador. Se eligió *Raspbian* como sistema operativo debido a que es una distribución oficial de *Raspberry Pi* y optimizada para ser usado en cualquier placa *Raspberry Pi*. Por otro lado, se hizo uso de Apache como servidor *Web*, el cual se caracteriza por ser gratuito, multiplataforma y de código abierto. Además, este puede ser instalado y configurado en el sistema operativo *Raspbian* sin ningún problema.

El minicomputador *Raspberry Pi* encargado de transmitir los datos a los diferentes dispositivos que forman parte de sistema domótico global es el controlador central de toda la estructura domótica montada sobre la vivienda, es decir, el resto de los componentes que forman parte del sistema se gestionan de forma remota mediante una interfaz web capaz de enviar y recibir información a cada uno de los dispositivos.

Con relación al diseño del entorno *Web* se utilizó HTML como lenguaje de marcado y los lenguajes de código abierto PHP y C++ para la ejecución de las diferentes tareas que se comandarán a través de la página *Web* en la vivienda. Cualquier dispositivo mediante un navegador *Web*, dentro de la red interna del hogar, puede controlar ya sea el encendido y apagado de la luz en la vivienda, el apagado remoto de computadoras dentro del hogar, el control de las persianas y el control de acceso en la puerta del domicilio.

A través de la página *Web* alojada en la *Raspberry Pi*, el sistema es capaz de controlar el encendido y apagado de la iluminación del hogar mediante un interruptor inteligente de la marca SONOFF, el cual trabaja con un *firmware* llamado *Tasmota*. Además, este permite integrar dicho control a la interfaz *Web* creada para el control del sistema domótico global.

Por otro lado, parte de todo el sistema también incluye el apagado y encendido remoto de computadoras que se encuentran operado dentro de la red interna de la vivienda. Para llevar a cabo dicho procedimiento se empleó conexiones cableadas desde la tarjeta de red de cada uno de los equipos hacia el dispositivo enrutador del domicilio; se aplicó algunas configuraciones dentro de las directivas locales de los computadores y todo este proceso se reflejó en la interfaz *Web* con la ayuda del lenguaje de programación PHP.

Adicionalmente, en lo que respecta al control de la persiana se utilizó un motor paso a paso Nema 17 y un módulo ESP32 mismo que brinda una solución de control mediante *Wi-Fi* a través de la programación en el IDE de Arduino. Este pequeño y potente chip permitió controlar a través de la red inalámbrica del hogar el motor para el cierre o apertura de la persiana. Para realizar la integración de dicho control con la interfaz *Web* del sistema domótico, se programó el módulo ESP32 con el lenguaje de programación C++.

Para el control de acceso a la puerta de ingreso, se dispuso de un equipo biométrico de reconocimiento facial colocado en la parte externa. Dicho sensor posee también un módulo ESP32 en el cual se realizó la respectiva programación en el lenguaje de alto nivel C++, con el fin de que se integre con el resto de los apartados de control de toda la interfaz *Web* del sistema domótico.

Finalmente, se efectuaron varias pruebas para cerciorarse del correcto funcionamiento entre cada uno de los dispositivos que forman parte del sistema inteligente y la interfaz *Web* desarrollada, con la finalidad de calibrar tiempos y corregir errores que se puedan presentar.

3 RESULTADOS Y DISCUSIÓN

3.1 Determinación de la arquitectura y dispositivos apropiados para el desarrollo del sistema domótico

El prototipo del sistema domótico dispone de una arquitectura centralizada en la que el controlador central es decir la *Raspberry Pi* es la encargada de transmitir los datos e información a los diferentes sensores y actuadores que forman parte del sistema. En otras palabras, el miniordenador comanda cada uno los actuadores que permiten el encendido y apagado de la luz, las computadoras, el control de la persiana y de la puerta

en la vivienda. Dicha estructura de arquitectura se puede observar a continuación en la Figura 3.1.

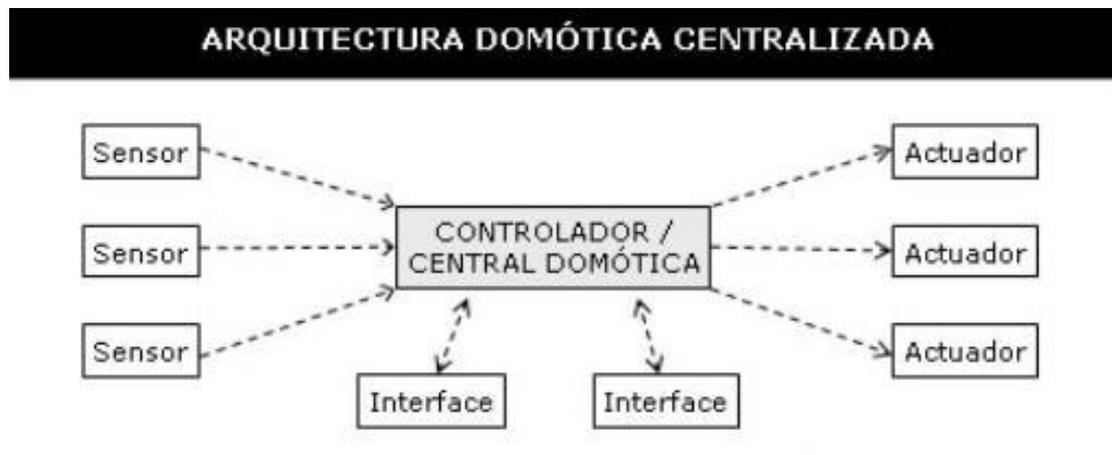


Figura 3.1 Arquitectura centralizada

Una vez comprendida la arquitectura de desarrollo del sistema domótico global, para poder automatizar los distintos procesos mencionados líneas atrás, se realizó una investigación exploratoria del funcionamiento de los distintos actuadores y sensores que permiten ser integrados tanto al control de la *Raspberry Pi* como a la interfaz *Web*.

Control remoto de la iluminación

Para el control del encendido y apagado de la luz se utilizó un interruptor inteligente de la marca *SONOFF* debido a que para su control remoto se añadió un cable hasta dicho interruptor. La instalación eléctrica convencional en la vivienda es la que se puede apreciar en la Figura 3.2, en la que el cable neutro se conecta a un extremo del foco y la fase es aquella que funcionará a modo de *switch* a través del interruptor tradicional para el encendido y apagado del foco.

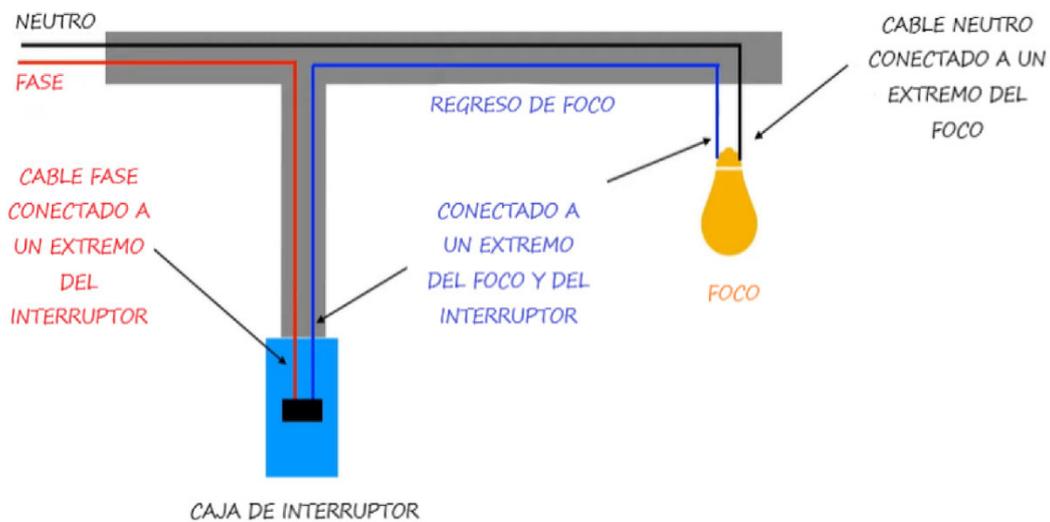


Figura 3.2 Diagrama de conexión para interruptor convencional

Ahora bien, para poder realizar la instalación del interruptor inteligente es necesario añadir un cable neutro desde uno de los extremos en el que se encuentra conectado el cable neutro en el foco hasta la caja del interruptor donde se colocará el dispositivo *SONOFF*. En la Figura 3.3 se muestra el diagrama de conexión para el interruptor inteligente.

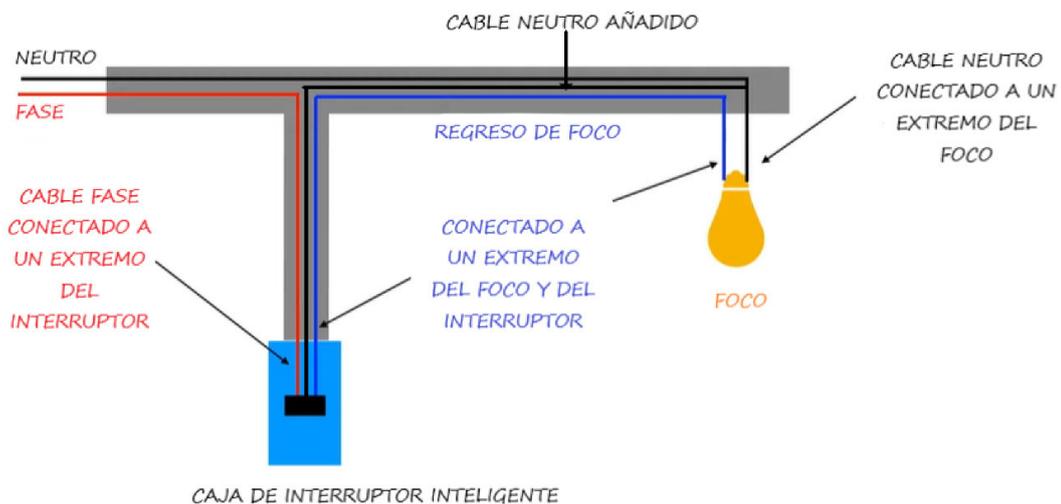


Figura 3.3 Diagrama de conexión para interruptor inteligente *SONOFF*

El modelo del interruptor inteligente elegido para la implementación del prototipo del sistema domótico es un *SONOFF* CH1 (ver Figura 3.4) es decir, un interruptor de tan solo un canal de conexión, el cual permite realizar el encendido y apagado de un foco

en el domicilio. Dentro de la gama extensa de este tipo de interruptores, existen modelos capaces de poseer hasta 4 canales en caso de desear automatizar lugares del hogar con varias bombillas de luz.



Figura 3.4 Interruptor inteligente *SONOFF* CH1

Este tipo de interruptores internamente en su circuitería permiten que el usuario pueda programar su funcionamiento con la ayuda de *software* que provee el fabricante y módulos que facilitan cargar el programa directamente a la placa.

Las especificaciones técnicas de funcionamiento de este modelo en concreto de interruptor inteligente son las siguientes:

- Rango de voltaje: 90-250 (V) AC 50/60 (Hz).
- Corriente máxima: 2 (A).
- Máximo de potencia: 400 (Vatios).
- Frecuencia: 2.4 (GHz).

Control remoto de persiana

Para el manejo de la persiana a través de la interfaz *Web*, se empleó el uso de un motor paso a paso Nema 17, ver Figura 3.5. El control remoto del motor se lo realiza con la ayuda de un módulo ESP32, ver Figura 3.6, el cual permite ser programado para controlar el número de vueltas que debe realizar el motor para abrir o cerrar la persiana. El motor Nema 17 tiene un torque de 3.2 (kg.cm), lo cual lo hace apto para la implementación tanto en el despliegue como de cierre de la persiana; esto debido a que las dimensiones y peso de esta es de 1(m) x 1(m) y 2.7 kg respectivamente.

El cálculo del torque (Ecuación 3.1) en los motores paso a paso depende principalmente de la velocidad a la cual trabajará el motor. A continuación, en la Figura 3.7 se muestra la relación de trabajo entre la velocidad y el torque proporcionado por la ficha técnica del fabricante. En dicha figura se puede apreciar que el motor puede obtener una fuerza máxima del alrededor de los 0.32 N.m (Newton.metro) es decir:

$$1 \text{ kilogramo-metro (kg.m)} = 9.80665 \text{ Newton-metro (N.m)}$$

Por lo tanto:

$$\frac{0.32 \text{ (N.m)} \times 1 \text{ (kg.m)}}{9.80665 \text{ (N.m)}} = 0.03263 \text{ (kg.m)} = 3.263 \text{ (kg.cm)}$$

Ecuación 3.1 Torque máximo NEMA 17

Es decir, la capacidad de movimiento del motor Nema 17 ante la persiana de 2.7 kg de peso se mantiene bajo las referencias correctas de trabajo.



Figura 3.5 Motor paso a paso Nema 17



Figura 3.6 Módulo ESP32

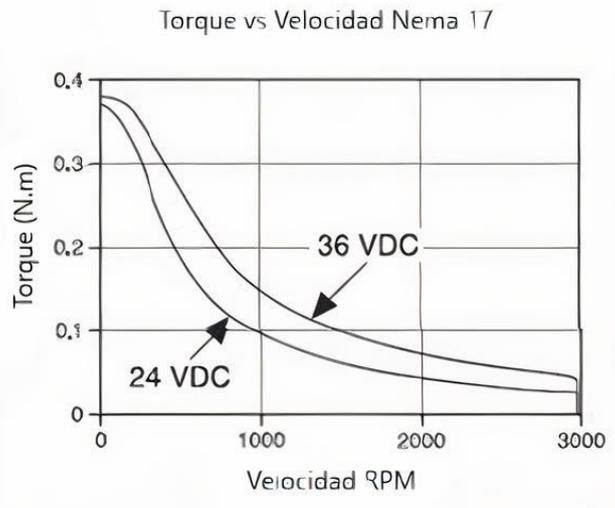


Figura 3.7 Torque versus velocidad

Este increíble módulo puede recibir órdenes a través de la red *Wi-Fi* del hogar, lo cual es muy útil en el desarrollo de temas enfocados a la domótica. A continuación, se muestra un esquema en la Figura 3.8 con la integración de 3 motores Nema el cual es el número máximo de motores a ser conectados sobre el módulo por motivos de corriente y tensión.

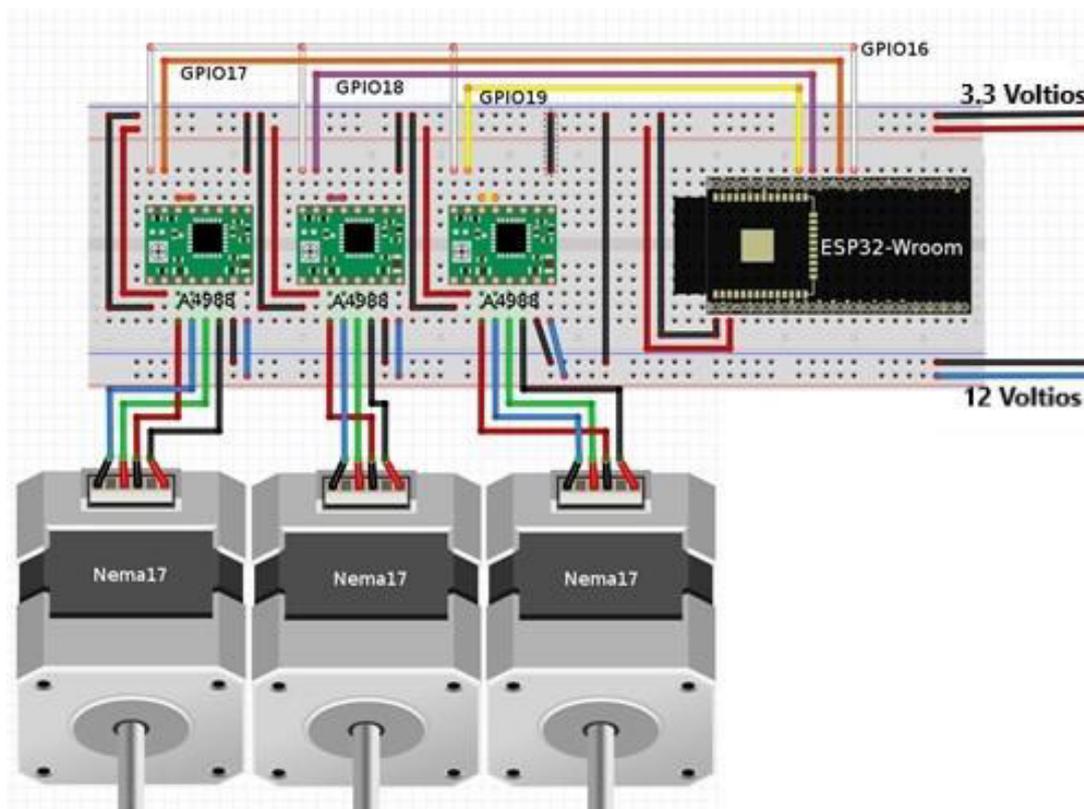


Figura 3.8 Diagrama de conexión con ESP32 para el control de motores paso a paso

Dentro de toda la circuitería mostrada anteriormente en la Figura 3.7, es necesario emplear el controlador A4988, ver Figura 3.9, mismo que permitirá suministrar la energía suficiente para poner en funcionamiento el motor paso a paso Nema 17 debido a que el módulo ESP32 solo opera con una tensión máxima de 3.3 (V) y el motor a una tensión de 12 (V). Pues bien, el uso del controlador A4988 no solo tiene que ver con temas relacionados a la tensión, sino que también permite indicarle al motor el número de vueltas a ejecutar y la dirección que este debe realizar con la ayuda de los pulsos enviados por el ESP32.



Figura 3.9 Controlador A4988

Entre las increíbles prestaciones del módulo ESP32 están que permite a través del lenguaje C++, programar tanto el diseño gráfico como la ejecución de las órdenes de giro del motor a través del IDE de Arduino, para posteriormente integrar el control de la persiana a la interfaz global del sistema domótico.

Control remoto de ordenadores

Por otro lado, en lo que respecta al manejo remoto de los ordenadores que se encuentran dentro de la vivienda, para lograr el encendido y apagado de cada uno de ellos es necesario que se encuentren conectados directamente, mediante cableado, al equipo enrutador del hogar para poder enviar las ordenes de encendido y apagado a través del protocolo *Wake on LAN* (WoL). Entre los requisitos para apagar remotamente los ordenadores están:

- Que se encuentren en la misma red de área local (LAN).
- Conocer el nombre de usuario y la contraseña de inicio de sesión.
- Conocer la dirección IP asignada al ordenador.
- En caso de que el sistema operativo sea inferior a *Windows 8* se debe ingresar a *regedit* y realizar cambios dentro del registro, mismos que serán explicados posteriormente.

Asimismo, para conseguir el encendido remoto de los ordenadores es necesario:

- Habilitar *Wake up on LAN* dentro de la BIOS de los ordenadores.
- En la ventana Administrador de dispositivos – Adaptadores de red – Propiedades – Administración de energía se deberá marcar todas las casillas y aceptar.
- Continuando en el apartado Propiedades, en Opciones Avanzadas es fundamental seleccionar *Wake on Magic Packet* asegurándonos de que este marcado la opción *Enabled*.

La respecta conexión desde el equipo enrutador hacía los computadores, se puede visualizar en la Figura 3.10.

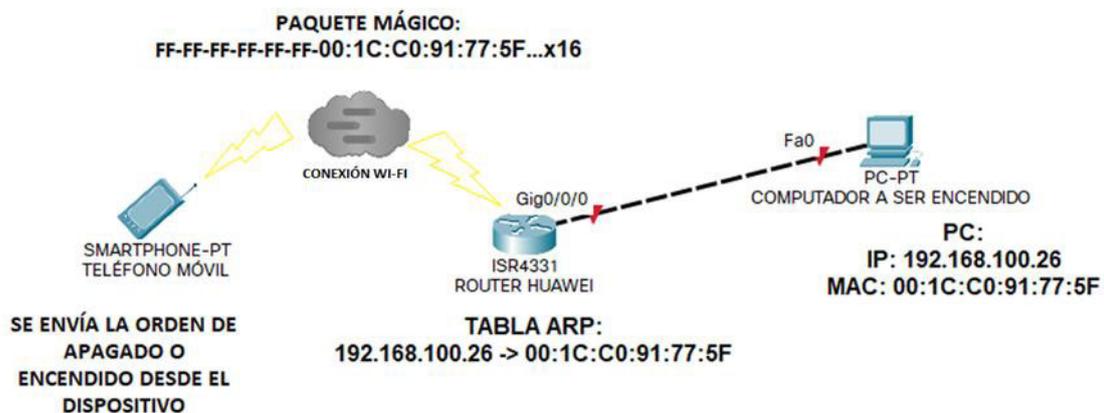


Figura 3.10 Conexión Wake on LAN

Ahora bien, para conseguir que todo esto funcione a través de la interfaz *Web* se deberá realizar la correspondiente programación en el lenguaje PHP, procedimiento que se explicará posteriormente.

Control de acceso

Por último, en relación con el control de acceso, del mismo modo en base a una investigación exploratoria y tomando en consideración las increíbles prestaciones del control remoto que brinda el módulo ESP32, se utilizó un módulo ESP32-CAM tal y como

se muestra en la Figura 3.11, para controlar la apertura o cierre de la puerta en el hogar por medio de la detección facial.



Figura 3.11 Módulo ESP32-CAM

De igual manera, dicho módulo permite enviar órdenes a una cerradura electrónica instalada en la puerta a través de la detección facial. Todo el control del sistema se lo puede realizar mediante una interfaz *Web* programada en C++ por medio del IDE de Arduino. Dentro de las principales características del módulo están:

- Voltaje de alimentación: 5 (V).
- *Wi-Fi*: 802.11b/g/n.
- Tipo de cámara: OV2640 2 (MP).
- Posee un CPU de 32 bits de doble núcleo de baja potencia.
- Frecuencia de trabajo: 240 (MHz).

En la Figura 3.12 y en la Figura 3.13 se observa el diagrama de conexión para el desarrollo del sistema biométrico que permitirá realizar la apertura de la puerta en el hogar. El circuito presenta dos secciones de voltaje de 12 (V) y 5 (V) respectivamente. La primera con mayor voltaje es la que permite el accionamiento del solenoide o cerradura electrónica por medio del módulo relé. La segunda sección corresponde a los 5 (V) de corriente continua obtenidos gracias al circuito integrado 7805, permitiendo la alimentación del módulo ESP32-CAM, el cual es el encargado de realizar la detección facial y además enviar la señal de activación al relé cuando corresponda.

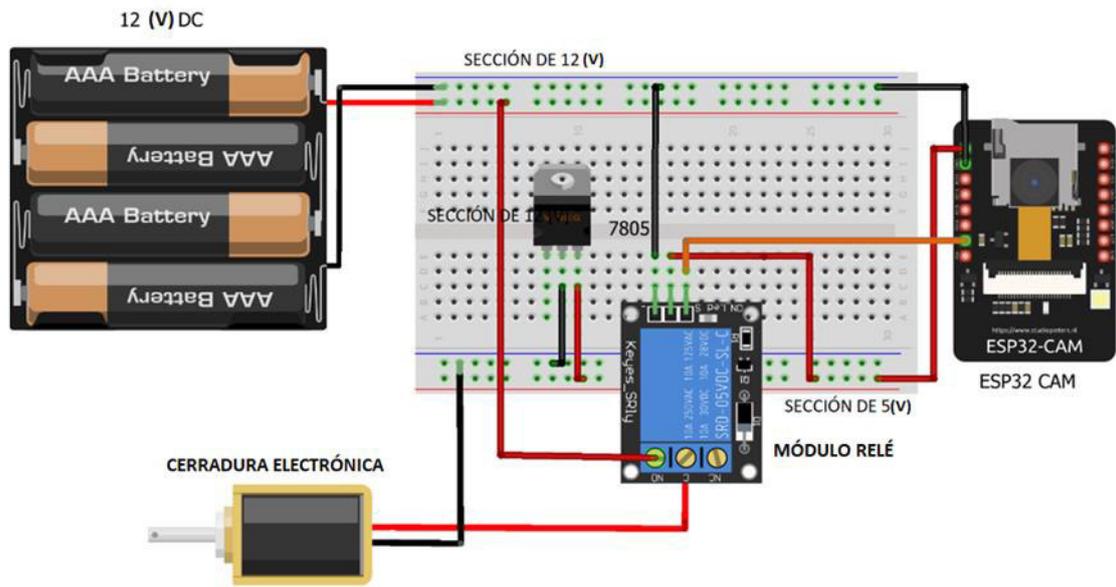


Figura 3.12 Diagrama de conexión con ESP32-CAM para manejo de la puerta

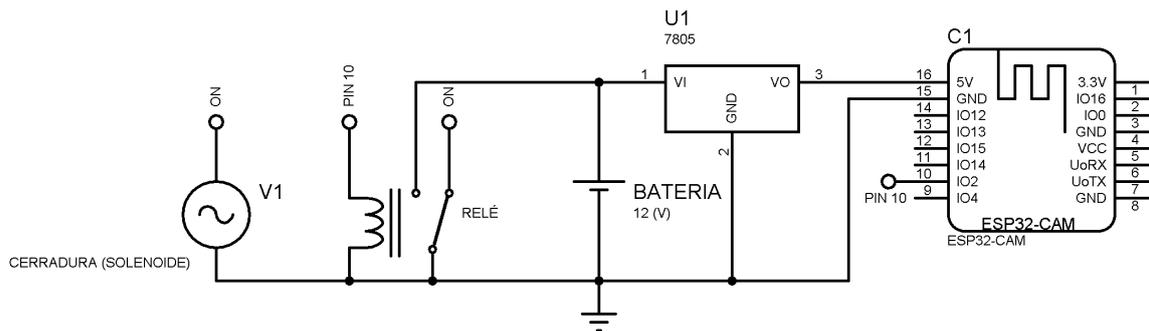


Figura 3.13 Diagrama de conexión en proteus

Finalmente, se procedió a configurar una IP estática para cada uno de los dispositivos que conforman la red local, estos dispositivos son: el minicomputador *Raspberry Pi* y los módulos ESP32. La asignación se realizó a través de su respectiva dirección física, tal y como se visualiza en la Figura 3.14.

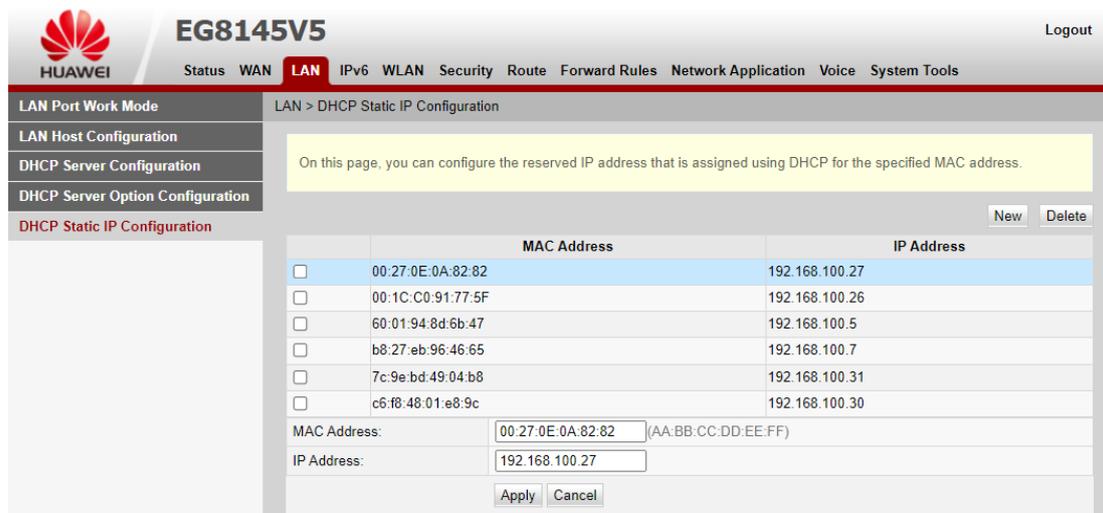


Figura 3.14 Asignación de IP estática

3.2 Desarrollo del algoritmo de control para automatizar los distintos equipos e instalaciones del hogar

Para lograr el funcionamiento completo de todo el sistema se desarrollaron diferentes algoritmos que finalmente son comandados bajo la misma interfaz *Web*. Cabe mencionar que se podrá ingresar, siempre y cuando se haya autenticado una cuenta creada en el sistema de registro. De esta manera, es posible iniciar sesión y posteriormente acceder a cada aplicación mediante el menú principal de la página.

Algoritmo para accionamiento remoto de computadoras

El algoritmo desarrollado se divide en dos secciones: la primera corresponde al encendido de ordenadores y la segunda sección concierne al apagado remoto de los mismos. Ambas funciones se encuentran disponibles en la misma sección del menú *Web* que corresponde al archivo `wakeonlan.html`, ver Anexo 4.10. Además, posibilita la verificación del estado de las computadoras en un apartado que se actualiza constantemente, de tal manera que refleja si estos dispositivos se encuentran encendidos o apagados.

Encendido

En primer lugar, para encender remotamente el computador, es necesario activar el protocolo *Wake On LAN* en la placa madre de este, ingresando desde la BIOS como se muestra en la Figura 3.15.

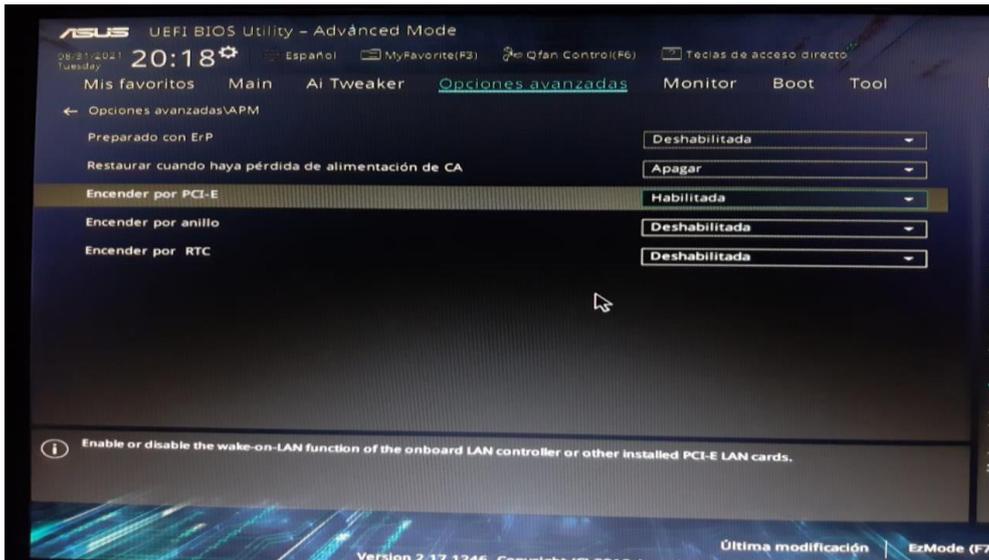


Figura 3.15 Activar WOL en placa madre

Posteriormente, se debe realizar las configuraciones respectivas en *Windows* para continuar con el procedimiento. Es necesario acceder al administrador de dispositivos e ingresar a las propiedades de la tarjeta de conexión *Ethernet* y activar el parámetro *wake on magic packet when the system is in the S0* como se muestra en la Figura 3.16. Cabe recalcar que dependiendo de cada tarjeta de red el nombre del parámetro a activar puede variar; sin embargo, siempre estará presente la palabra *Wake on LAN* que permitirá identificar la función respectiva.

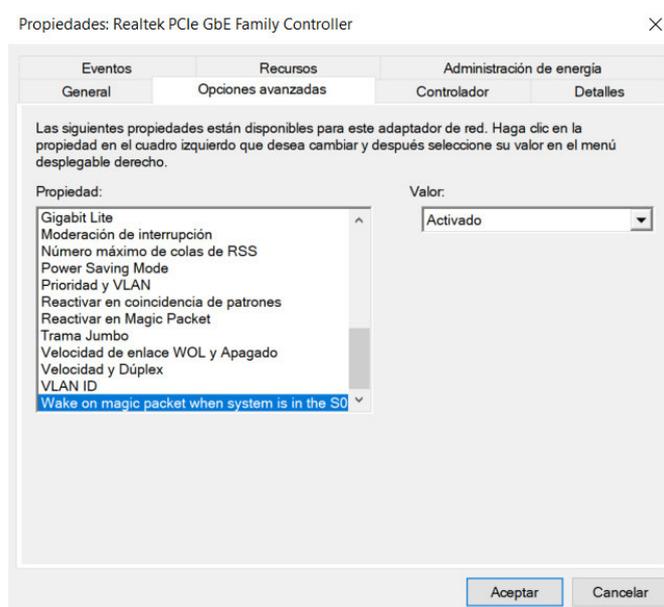


Figura 3.16 Habilitar encendido con paquete mágico

Adicionalmente, es necesario activar el parámetro que permite reactivar el equipo mediante un paquete mágico como se muestra en la Figura 3.17.

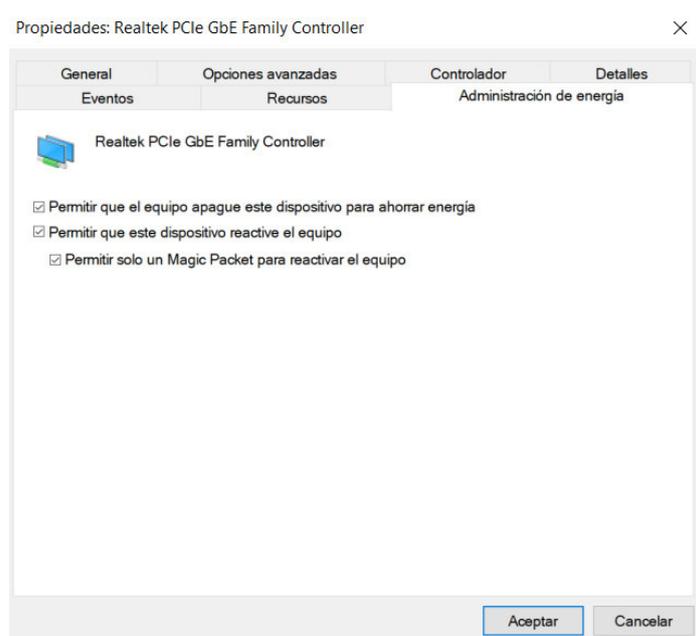


Figura 3.17 Permitir encendido mediante paquete mágico

Con estas propiedades de la tarjeta de red configuradas correctamente dentro del sistema operativo, es posible modificar el algoritmo de control y agregar la dirección MAC del dispositivo a encender. El lenguaje de programación encargado de realizar esta función es PHP, el cual ejecuta el comando de encendido como si fuera por medio del terminal del sistema operativo de la *Raspberry Pi*. Acompañado a la instrucción debe ir la dirección física del computador. Dicho comando es el siguiente:

```
exec("wakeonlan $mac");
```

Donde \$mac, es la variable que selecciona la dirección MAC correspondiente a la computadora seleccionada. Dicha variable corresponde a los datos de la dirección física descrita en el código principal (ver Anexo 4.11) y que son seleccionados mediante la etiqueta de hipertexto *select*.

Para obtener la dirección MAC del computador a ser encendido, es necesario ejecutar dentro del CMD el siguiente comando:

```
ipconfig -all
```

Al ejecutarlo aparecerá información variada de la red, por ejemplo: dirección IPv4, puerta de enlace, máscara de subred, entre otros. Es importante verificar que la dirección física que se obtenga sea del adaptador de *ethernet* ya que, si se toma la de otro adaptador como del inalámbrico, el computador no encenderá.

Adicionalmente, existe la opción de encender todos los computadores en caso de que sean varios y sea más factible prender a todos en una sola acción, en lugar de tener que seleccionar uno por uno. El archivo que permite realizar esta función es el denominado como `on_all.php`; presenta las siguientes líneas de código:

```
<?php
    echo "<font color='green'>j Todas las computadoras
        encendidas exitosamente!</font>";

    exec("wakeonlan 00:1C:C0:91:77:5F");

    exec("wakeonlan 00:27:0E:0A:82:82");

?>
```

Apagado

Ahora bien, para el apagado de computadoras es imprescindible que tanto el computador como la *Raspberry Pi* se encuentren en el mismo grupo de trabajo de red por lo cual se recomienda configurar como red privada o red doméstica en los ajustes de red del computador. Además, es necesario que dentro de las configuraciones de uso compartido se desactive la protección por contraseña, como se muestra en la Figura 3.18; caso contrario la *Raspberry PI* no podrá realizar la ejecución del comando.

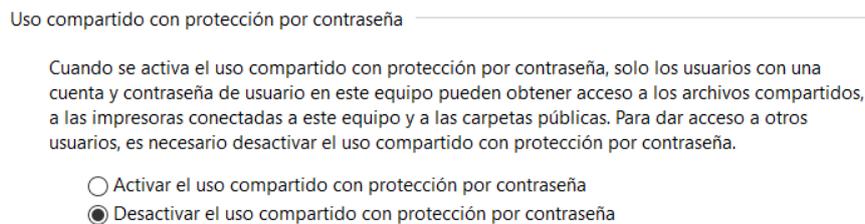


Figura 3.18 Desactivación de contraseña para uso compartido

Una vez configurado dicho parámetro, es necesario ingresar a las directivas de grupo local en herramientas de *Windows* y asignar los derechos de forzado de cierre remoto a *everyone*, como se indica en la Figura 3.19. Dependiendo de la versión de *Windows* en lugar de "everyone" se deberá colocar "todos" para que se pueda reconocer la instrucción. Mediante la opción "comprobar nombres" es posible identificar cuál de las dos opciones se debe colocar, es decir en inglés o en español.

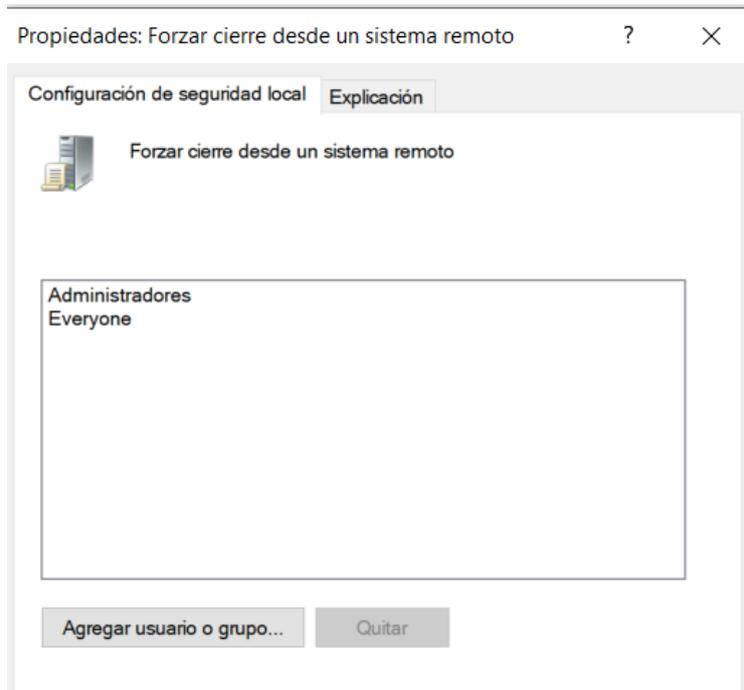


Figura 3.19 Asignación de derechos de usuario

Además, es necesario configurar el *firewall* de *Windows* para evitar problemas de comunicación. Es necesario dirigirse a la siguiente ruta:

Panel de control\Sistema y seguridad\Firewall de Windows Defender

Dentro de dicha ruta, se debe ingresar a la configuración avanzada para poder modificar las reglas de entrada del cortafuegos, tal como se muestra en la Figura 3.20. Los parámetros que deben estar habilitados para evitar problemas de comunicación son los siguientes:

- Archivos e impresoras compartidos (petición eco: ICMPv4 de entrada)
- Compartir archivos e impresoras (SMB de entrada)



Figura 3.20 Configuración de reglas de entrada

Posteriormente, se debe realizar cambios dentro del registro de *Windows* como administrador. Una vez ejecutado *regedit* es necesario ir a la siguiente ruta:

Equipo\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System

En esta dirección es necesario crear un valor DWORD de 32 bits con el nombre *LocalAccountTokenFilterPolicy*, asignándole además un valor hexadecimal de 1, tal como se indica en la Figura 3.21.

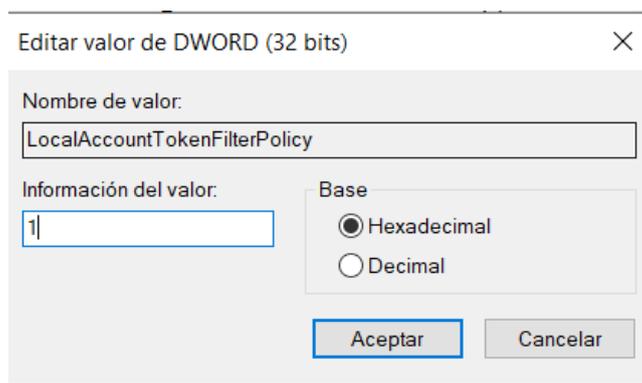


Figura 3.21 Modificación en registro de Windows

Una vez realizados los pasos anteriores, es necesario modificar el algoritmo de apagado remoto, configurando la dirección IP, usuario y contraseña de cada computador. El comando que ejecuta dicha orden es el siguiente:

```
shell_exec("net rpc shutdown $pc -f -t 10 -C APAGANDO");
```

Donde:

\$pc, contiene las variables de la dirección IP, usuario y contraseña de cada computador independiente, las cuales se encuentran especificadas en el código principal (ver Anexo 4.12) y que son seleccionados mediante la etiqueta de hipertexto *select*.

- *-f*, indica forzar el cierre.
- *-t 10*, indica el tiempo en segundos para que se apague el computador.
- *-C APAGANDO*, indica el mensaje que se reflejará en el computador a apagar.

De esta manera, la *Raspberry Pi* podrá apagar el computador remotamente mediante los comandos ejecutados por el lenguaje PHP. Al ejecutar esta acción aparece una ventana como la que se muestra en la Figura 3.22.

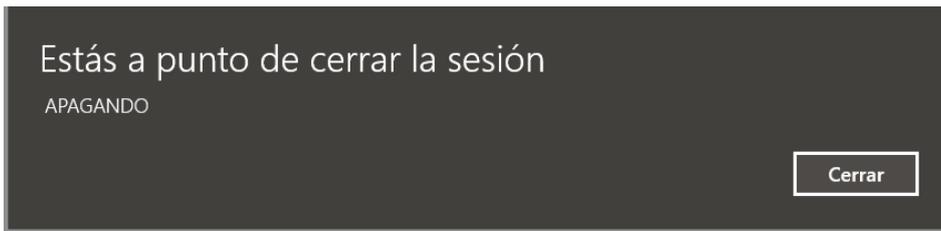


Figura 3.22 Apagado de forma remota

De igual manera que con el encendido, existe la opción de apagar todos los computadores en una sola acción en lugar de tener que seleccionar uno por uno. El archivo que permite realizar esta función es el denominado `off_all.php` y presenta las siguientes líneas de código:

```
<?php
    echo "<font color='red'>¡Todas las computadoras apagadas
        exitosamente!</font>";

    shell_exec("net rpc shutdown -l 192.168.100.26 -U jeff%jefferson -f -t 10 -C
        APAGANDO ");

    shell_exec("net rpc shutdown -l 192.168.100.27 -U jorge2%asdqw -f -t 10 -
        C APAGANDO");

?>
```

Monitoreo de estado

Para el monitoreo de cada una de las computadoras, se implementó un código de igual forma en lenguaje PHP, que permite visualizar el estado de estas. La función de la *Raspberry* es realizar varios *pings* de forma infinita hacia la dirección IP de la computadora mediante el siguiente comando:

```
exec("ping -c 1 $ip",$output,$status);
```

Donde:

- `$ip`, es la variable de dirección de red correspondiente a cada dispositivo
- `$output`, es la variable del nombre personalizable de cada computador o artefacto.
- `$status`, indica el estado de encendido y apagado

De esta manera, es posible observar a través de una matriz *Web* si cada computador se encuentra encendido o apagado, tal como se muestra en la Figura 3.23.

Dentro del código de programación para agregar un computador o dispositivo que tenga una dirección IP asignada por el dispositivo de enrutamiento se debe modificar la matriz de la siguiente forma:

```
$iplist = array(
    array("192.168.100.26","PC JEFFERSON"),
    array("192.168.100.27","PC JORGE"),
);
```

MONITOREO DE COMPUTADORAS			
#	IP	ESTADO	DESCRIPCION
0	192.168.100.26	Apagado	PC JEFFERSON
1	192.168.100.27	Apagado	PC JORGE

Figura 3.23 Matriz de monitoreo de estado

Para una visualización breve del cambio de estado se estableció que la página se recargue automáticamente cada 3 segundos:

```
echo header("refresh: 3");
```

El código completo de monitoreo de estado se visualiza en el Anexo 4.15.

Algoritmo para accionamiento remoto de la puerta

El dispositivo encargado de esta función es el ESP32-CAM. En primer lugar, fue necesario la programación en el IDE de Arduino ya que al ser un módulo de esta tarjeta debe ser codificado en su entorno de desarrollo. Posteriormente, es necesario definir el tipo de cámara y las variables globales para el desarrollo del algoritmo, como se observa a continuación, principalmente se define como: *CAMERA_MODEL_AI_THINKER* ya que es la que permite el reconocimiento facial.

```
#define CAMERA_MODEL_AI_THINKER //RECONOCIMIENTO FACIAL
```

```
#define Rele 2 //DEFINICION DE VARIABLE PARA EL PIN 2
```

```
#include "camera_pins.h"
```

Luego, se procedió a definir los parámetros de nombre de red y contraseña para que el ESP32 realice la conexión con el enrutador y de esta forma este le pueda brindar una dirección IP local.

```
const char* ssid = "NETLIFE - FMLA MINALLA";
```

```
const char* password = "D@nger2o21!";
```

Las demás líneas del algoritmo se pueden visualizar en el Anexo 4.17. Seguidamente, se procedió a compilar el programa por lo que para aquello es esencial que dentro del IDE de Arduino se mantenga configurado con los parámetros que se observan en la Figura 3.24, para de esta forma evitar errores de compilación y de memoria dentro de la tarjeta.

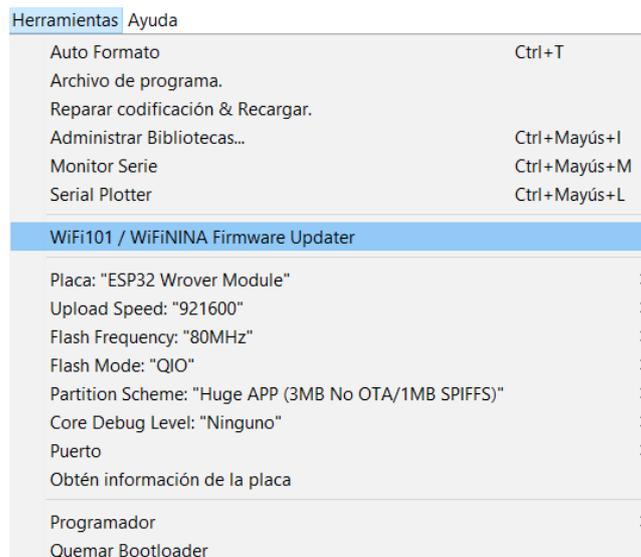


Figura 3.24 Parámetros de compilación

Para que el código pueda cargarse fue necesario usar un conversor USB a TTL. Los pines de tierra (GND) y IO0 del ESP32 deben encontrarse puenteados en un inicio. El pin UOT del ESP32 debe estar conectado al pin de recepción (Rx) del conversor. El pin UOR del ESP32 debe conectarse al pin de transmisión (Tx) del conversor. Finalmente, para la alimentación deben conectarse entre sí los pines de 5 (V) y GND de ambas tarjetas respectivamente. La conexión se visualiza de forma gráfica en la Figura 3.25 y la Figura 3.26.

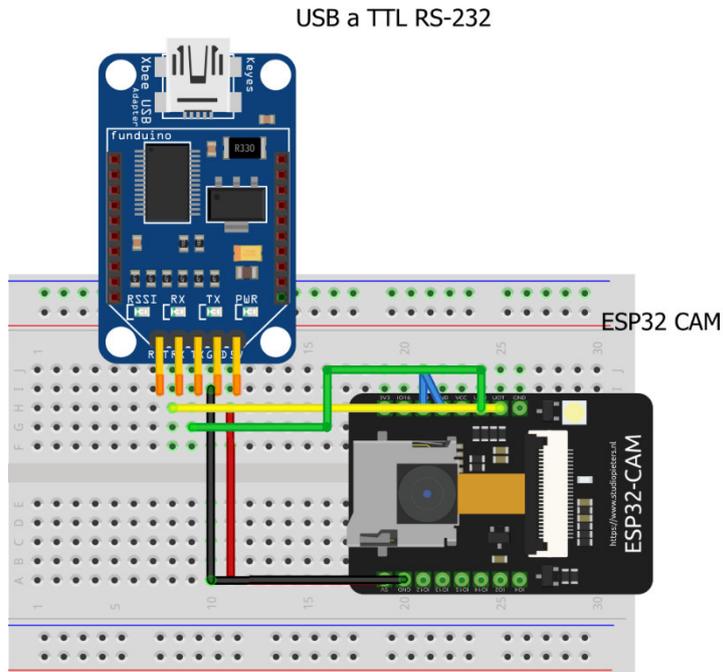


Figura 3.25 Conexión ESP32 y RS-232

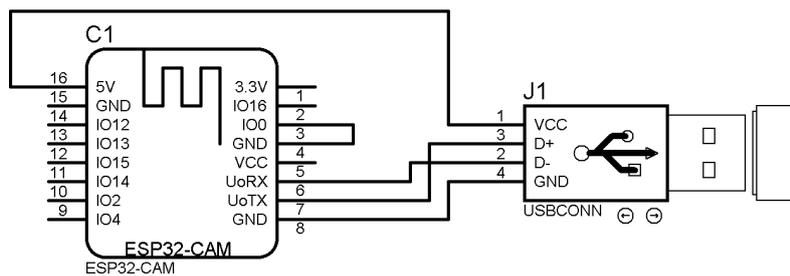
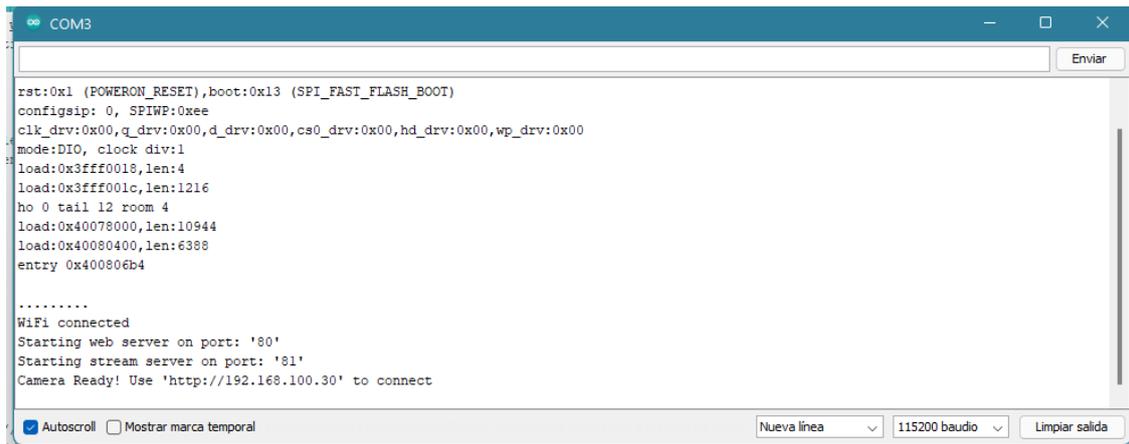


Figura 3.26 Conexión a TTL proteus

Una vez cargado el código en el módulo se procedió a retirar el puente entre los pines GND e IO0, para posteriormente presionar el botón de reinicio para que el monitor serial configurado en 115200 baudios pueda mostrar la dirección IP asignada, tal como se muestra en la Figura 3.27.



```
COM3
Enviar
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4

.....
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.100.30' to connect

Autoscroll [x] Mostrar marca temporal Nueva línea 115200 baudio Limpiar salida
```

Figura 3.27 Dirección IP asignada al ESP32-CAM

Algoritmo para el control remoto de la persiana

Para mostrar una interfaz *Web* que permita enviar las ordenes al ESP32 y seguidamente al motor paso a paso con el fin de abrir y cerrar la persiana, se procedió a habilitar un servidor *Web* sobre el módulo para lo cual es necesario ingresar las credenciales de acceso al *router*, tal y como se muestra a continuación:

```
WiFiServer server(80);

const char* ssid = "NETLIFE - FMLA MINALLA";

const char* password = "Danger2021";
```

Ahora, para poder inicializar la conexión con la red *Wi-Fi* se deberá realizar la siguiente programación, misma que permitirá autenticar si las credenciales tanto del nombre como de la contraseña de la red inalámbrica son correctas. Adicionalmente, se configura una IP estática para el desarrollo de la interfaz para lo cual se asignó la IP 192.168.100.31. Cabe recalcar que esta configuración se le debe realizar dentro del *void setup*. A continuación, se detallan las líneas de programación a seguir para poder iniciar el servidor.

```

        Serial.println();

        Serial.print("CONECTANDO A: ");

        Serial.println(ssid);

        WiFi.begin(ssid, password);

        while (WiFi.status() != WL_CONNECTED){

            delay(5000);

            Serial.println("ESPERANDO CONECTAR A LA
                RED WI-FI..."); }

        IPAddress ip(192,168,100,31);

        IPAddress gateway(192,168,100,1);

        IPAddress subnet(255,255,255,0);

        WiFi.config(ip, gateway, subnet);

        Serial.println("");

        Serial.println("RED WI-FI CONECTADA");

        Serial.println("DIRECCIÓN IP: ");

        Serial.println(WiFi.localIP());

        server.begin();

```

Una vez conectado a la red *Wi-Fi* local y asignada la correspondiente dirección IP estática, se procede a realizar la debida programación de la interfaz *Web* (ver Anexo 4.18) donde al presionar un botón desde el navegador, este enviará una petición al módulo ESP32 mismo que responderá con el envío de dos señales de pulso a través de los pines GIOP 16 y 17. El pin 16 se encargará de realizar el control de giro. Por lo tanto, con 1L realiza el giro hacia la derecha y con 0L invierte el giro, es decir hacia la izquierda. Por otro lado, el pin 16 controlará el número de giros que realizará el motor para enrollar o desplegar la persiana.

Como dato referencial dentro de la programación para el número de vueltas que deberá realizar el motor paso a paso, se tiene que el ángulo de paso del motor es de 1.8° es decir que, al enviar 200 pulsos, esto será $200 * 1.8^\circ = 360^\circ$ se estaría consiguiendo un giro completo de motor.

Algoritmo para el control remoto de la iluminación

Para poder *flashear* o reprogramar la configuración del interruptor *SONOFF* que viene de fábrica para ser integrado a *Home Assistant*, se debe cargar las nuevas líneas de código a través de un conversor de USB a TTL. Dicho método de carga se mostrará en líneas posteriores. El interruptor inteligente cuenta internamente con un ESP32 mismo que permite controlar de forma *Web* la iluminación del hogar. Entre las líneas de programación lo que se realiza es enviar datos al ESP32 para abrir y cerrar el *switch* interno con el que cuenta el interruptor y de esta manera impedir o permitir el paso de corriente hacia el foco.

```
void EmergencyReset(void) {  
    Serial.begin(115200);  
    Serial.write(0xA5);  
    Serial.write(0x5A);  
    delay(1);  
    if (Serial.available() == 2) {  
        if ((Serial.read() == 0xA5) && (Serial.read() == 0x5A)) {  
            SettingsErase(3);  
            do {  
                Serial.write(0xA5);  
                delay(1000);  
            } while (Serial.read() == 0xA5);  
            ESP_Restart(); } }  
    Serial.println();  
    Serial.flush();}
```

3.3 Diseño de la interfaz de usuario para el manejo del sistema domótico

La interfaz de usuario para el manejo del sistema domótico fue desarrollada a través de programación en HTML para elementos de diseño como botones, casillas, entre otros. Mientras que para los estilos de dichos elementos se utilizó CSS. Incluye un sistema de registro de usuarios que posteriormente tienen que pasar por el proceso de inicio de sesión para poder ingresar.

La página principal del sistema que se observa en la Figura 3.28 corresponde al archivo llamado index.php y presenta la interfaz en la cual aparece el nombre del proyecto y las opciones iniciales de registro o inicio de sesión según corresponda.

```
<h1> SISTEMA DOMÓTICO CON RASPBERRY  
PI - HOGARES</h1>
```

```
<a href="login.php">Iniciar sesión</a> ó <a  
href="register.php">Registrarse</a>
```

Para poder ingresar al sistema es necesario ingresar a un navegador *Web* de preferencia e ingresar la dirección:

```
http://192.168.100.7/home/
```

La dirección IP corresponde a la asignada por el enrutador a la *Raspberry*, mientras que /home es la ruta donde se encuentra el archivo index.php correspondiente a la página principal o de inicio.

Así mismo, los estilos de la página tales como: fondo o *background*, la posición, color, tipo y tamaño de fuente; están dados por la codificación en CSS.

```
body{margin: 0; padding: 8% 0;  
font-family: 'Lora', serif; text-align: center;  
background:url(../epnfondo.png); background  
repeat:no-repeat; background-attachment:fixed; }
```



Figura 3.28 Página principal

Sistema de registro de usuarios

Este sistema está conformado por principalmente dos archivos:

- database.php (Anexo 4.2)
- register.php (Anexo 4.5)

La página principal corresponde al archivo register.php, en la cual es posible realizar el proceso de registro. Dicha página se puede observar en la Figura 3.30. En esta sección *Web* se codifica en HTML con ayuda de la etiqueta *input* y el método *post*, los espacios en los cuales el usuario ingresa tanto el nombre como la contraseña para poder identificarse. Se hace referencia a las siguientes líneas de código:

```
<form action="register.php" method="post">  
  
  <input name="nombre" type="text" placeholder="Ingrese su nombre">  
  
  <input name="password" type="password" placeholder="Ingrese su contraseña">  
  
  <input type="submit" value="Enviar">  
  
</form>
```

Ahora bien, para poder registrar cada usuario, fue imprescindible la creación de la base de datos dentro de MariaDB.

El primer paso correspondió a seleccionar el nombre como el tipo de codificación de caracteres, los cuales fueron *login-base* y *utf-8* respectivamente.

Dentro *login-base* se creó una estructura de datos llamada usuarios la cual contiene tres variables:

- *int*: Corresponde al número de la lista de usuarios y es de tipo entero.
- *nombre*: Corresponde al nombre de usuario que va a ser registrado y es de tipo variable.
- *password*: Corresponde a la contraseña del usuario respectivo y es de tipo variable.

Dicha estructura o tabla se puede visualizar en la Figura 3.29.



Figura 3.29 Estructura de datos

Mediante el lenguaje PHP fue posible establecer la comunicación entre la página y la base de datos creada en *PHPmyadmin*. En el *script* denominado como *database.php* se observa dicha conexión con las siguientes líneas de programación:

```
$server = 'localhost';

$username = 'root';

$password = 'asdq';

$databse = 'login-base';

try {

$conn = new PDO("mysql:host=$server;dbname=$databse;",
                $username, $password);
```

Donde:

- `$server = 'localhost'`: Nombre del servidor, en este caso es el servidor local correspondiente a Apache hospedado en la *Raspberry*.
- `$username = 'root'`: Nombre del usuario creado en la aplicación *PHPmyadmin*.
- `$password = 'asdq'`: Contraseña del usuario en cuestión.
- `$database = 'login-base'`: Nombre de la base de datos donde se almacenarán los usuarios para el posterior inicio de sesión.
- `$conn`: Almacena la conexión de la base de datos, verificando que los datos de las variables previamente establecidas sean correctos.

Una vez realizada la conexión entre la base de datos y la página *Web* fue posible crear los usuarios en la pantalla de registro, tal y como se muestra en la Figura 3.30.

Página principal

Registrarse

¿Ya tienes cuenta? [Iniciar Sesión](#)

Figura 3.30 Registro en *Web*

El sistema arroja un mensaje de que el usuario ha sido creado correctamente o en caso de haber existido algún error de igual forma lo señala para poder asegurarse de que el proceso se ha cumplido correctamente. La programación es la siguiente:

```
catch (PDOException $e) {  
  
    die('Conexión Fallida: '.$e->getMessage());  
}
```

Para que las variables de usuario y contraseña ingresadas mediante el formulario de registro sean guardadas en la base de datos, se creó la variable “`$sql`” que ingresa los valores respectivos dentro de la tabla creada previamente llamada “usuarios” y los vincula mediante la función llamada *bindParam*. Además, la contraseña se encripta

mediante la función `PASSWORD_BCRYPT` para que no se pueda visualizar la contraseña de los usuarios en caso de que cualquier persona ingrese a la base de datos:

```
$sql = "INSERT INTO usuarios (nombre, password) VALUES (:nombre,
                                     :password)";

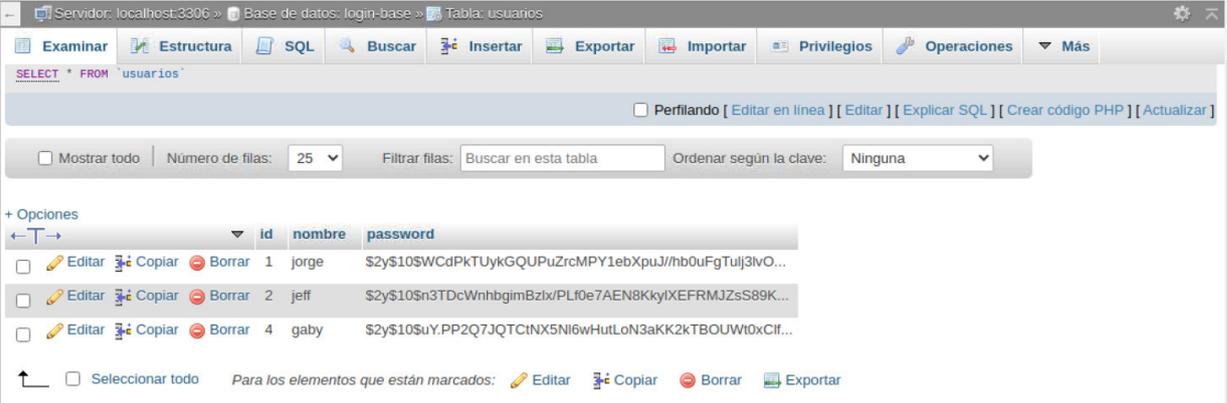
$stmt = $conn->prepare($sql);

$stmt->bindParam(':nombre',$_POST['nombre']);

$password = password_hash($_POST['password'], PASSWORD_BCRYPT);

$stmt->bindParam(':password',$password);
```

Finalmente, para observar de una forma más interna que el usuario ha sido creado, se ingresa a la base de datos y se visualiza el número de usuario, es decir su ID, ver Figura 3.31.



The screenshot shows a database management interface with a table named 'usuarios'. The table has three columns: 'id', 'nombre', and 'password'. There are three rows of data:

	id	nombre	password
<input type="checkbox"/>	1	jorge	\$2y\$10\$WCdPkTUykGQUPuZrcMPY1ebXpuJ//hb0uFgTulj3V0...
<input type="checkbox"/>	2	jeff	\$2y\$10\$n3TDcWnhbgimBzlx/PLf0e7AEN8KkyIXEFRMJZsS89K...
<input type="checkbox"/>	4	gaby	\$2y\$10\$uY.PP2Q7JQTCINX5Ni6wHutLoN3aKK2kTBOWt0xCif...

Figura 3.31 Usuarios registrados

Sistema de inicio de sesión

Ahora bien, para el inicio de sesión se trabaja en conjunto con los archivos de registro explicados previamente. Sin embargo, los scripts principales en este paso corresponden a:

- login.php (Anexo 4.3)
- logout.php (Anexo 4.4)

De igual forma que en el proceso de registro, para el inicio de sesión es necesaria la comunicación con la base de datos mediante el lenguaje PHP. Dentro del mismo se encuentra tanto el diseño que se puede observar en la Figura 3.32, como el algoritmo respectivo que permite el proceso de inicio de sesión. Algunas líneas importantes son:

```

require 'database.php';

if (!empty($_POST['nombre']) && !empty($_POST['password'])){

    $records = $conn->prepare('SELECT id, nombre,
password FROM usuarios WHERE nombre=:nombre');

```

Donde:

Require: Realiza una petición al archivo PHP que almacena y pregunta por los usuarios registrados.

\$records = \$conn->prepara: Selecciona el nombre y credenciales correspondientes a cada usuario registrado previamente en el proceso de registro.

En caso de que se ingrese un usuario no registrado o a su vez la contraseña introducida no sea correcta el sistema arrojará un mensaje indicando dicho error. Las líneas de código que permiten realizar esta autenticación son las siguientes:

```

if(count($results) > 0 && password_verify($_POST['password'], $results['password'])){

    $_SESSION['user_id'] = $results['id']; header('Location: /home'); } else{

    $message = 'Las credenciales no son correctas';

```

Donde:

- password_verify(\$_POST['password']): Valida si las credenciales ingresadas por el método POST en la interfaz *Web* corresponden a las almacenadas en la base de datos.
- \$_SESSION['user_id'] = \$results['id']; header('Location: /home'): Si los datos corresponden, se procede a redirigir a la interfaz inicial que se observa en la Figura 3.33, donde el usuario podrá ingresar al menú principal o cerrar sesión.
- \$message = 'Las credenciales no son correctas'. En caso de que no coincidan los datos se despliega el mensaje que lo indica.

Iniciar Sesión

¿No tienes cuenta? [Registrarse](#)

Figura 3.32 Inicio de sesión

Una vez que se hayan autenticado las credenciales de forma correcta, el sistema arrojará la interfaz de bienvenida que se muestra en la Figura 3.33, misma que identifica al usuario ingresado y posee un enlace que lo redirige al menú principal para el manejo remoto de cada uno de los dispositivos o a su vez presenta la opción de cerrar sesión y volver a la interfaz de inicio para volver a ingresar con un nuevo usuario y contraseña. La sección de código correspondiente es la siguiente:

```
<?php if(!empty($user)): ?>  
  
    <br>Bienvenido/a <?= $user['nombre'] ?> <br>Has  
    ingresado satisfactoriamente <div>  
  
    <a href="menu/menu-principal.html">Ingresar al menu del  
    sistema domótico</a> ó <a href="logout.php">Cerrar
```

En caso de que el usuario desee cerrar su sesión, el script de logout.php realiza su función mediante los siguientes comandos:

```
<?php  
  
    session_start();  
  
    session_unset();  
  
    session_destroy();  
  
    header('Location: /home');  
  
    ?>
```

Donde *session_destroy* realiza la función de culminar con la sesión presente y redirige a la ruta */home*, que como se explicó anteriormente corresponde a la página de inicio que solicita registro o inicio de sesión.

Página principal

Bienvenido/a jorge
Has ingresado satisfactoriamente
[Ingresar al menu del sistema domótico](#) ó [Cerrar Sesión](#)

Figura 3.33 Mensaje de bienvenida

Menú principal

Una vez que el usuario se encuentre registrado y haya iniciado sesión dentro del sistema, el siguiente punto es el menú principal que se puede observar en la Figura 3.34. El menú presenta los cuatro dispositivos a ser manejados remotamente, es decir: computadoras, persiana, luces y puerta. Este apartado al igual que las interfaces mencionadas anteriormente, fue codificado en HTML y sus estilos fueron diseñados en CSS.



Figura 3.34 Menú

Para que cada sección del menú pueda redireccionar a la página o dirección IP respectiva según corresponda se utiliza la etiqueta "href" la cual realiza esta acción al momento de seleccionar cada apartado.

Computadoras

```
<a href="wol/wakeonlan.html">  
  
<span class="text-item">COMPUTADORAS</span>  
<span class="down-item"></span>
```

Al ingresar a la sección de computadoras se despliega la página en HTML que permite el encendido y apagado remoto de dichos dispositivos, además del hipervínculo que permite ingresar al monitoreo, tal y como se visualiza en la Figura 3.35.

En esta página contiene esencialmente el elemento “*select*” que con ayuda del enlistado “*dropdown*” permite escoger la computadora a ser encendida o apagada. Así mismo, el elemento “*input*” de tipo “*submit*”, permiten la creación de botones que ejecutan las órdenes de encendido o apagado según corresponda.



Figura 3.35 Página Web para control de computadoras

Persiana

```
<a href=" http://192.168.100.31/">  
  
<span class="text-item">PERSIANAS</span>  
<span class="down-item"></span>
```

Dentro de la interfaz *Web* para el control de la persiana se cuenta con dos botones ubicados a los laterales de la página *Web* mismos que permiten abrir o cerrar la persiana, es decir; controlar el sentido de giro del motor paso a paso ya sea hacia la izquierda o derecha. En la Figura 3.36 se puede apreciar el diseño *Web* para el control remoto de la persiana. Cabe señalar que solo se implementó en una persiana.



CONTROL DE PERSIANA - HOGAR

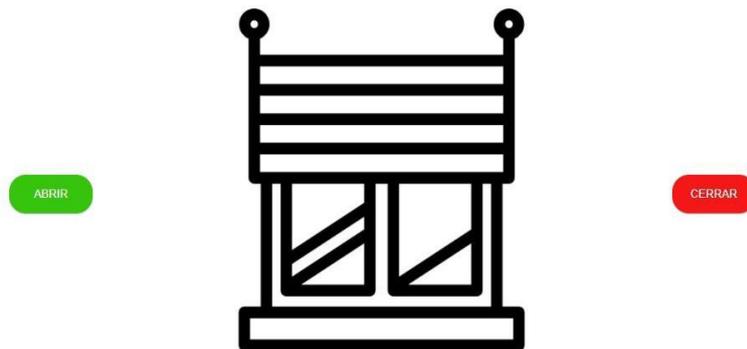


Figura 3.36 Página *Web* para el control de persiana

Iluminación

```
<a href="http://192.168.100.5/">  
  
<span class="text-item">LUCES</span>  
<span class="down-item"></span> </a>
```

Para el encendido y apagado remoto de la iluminación se cuenta con una interfaz sencilla en la que adicionalmente se puede ajustar horarios automáticos de encendido y apagado a cualquier hora del día. A continuación, en la Figura 3.37 Interfaz de encendido y apagado SONOFF se observa el diseño de la interfaz *Web*.

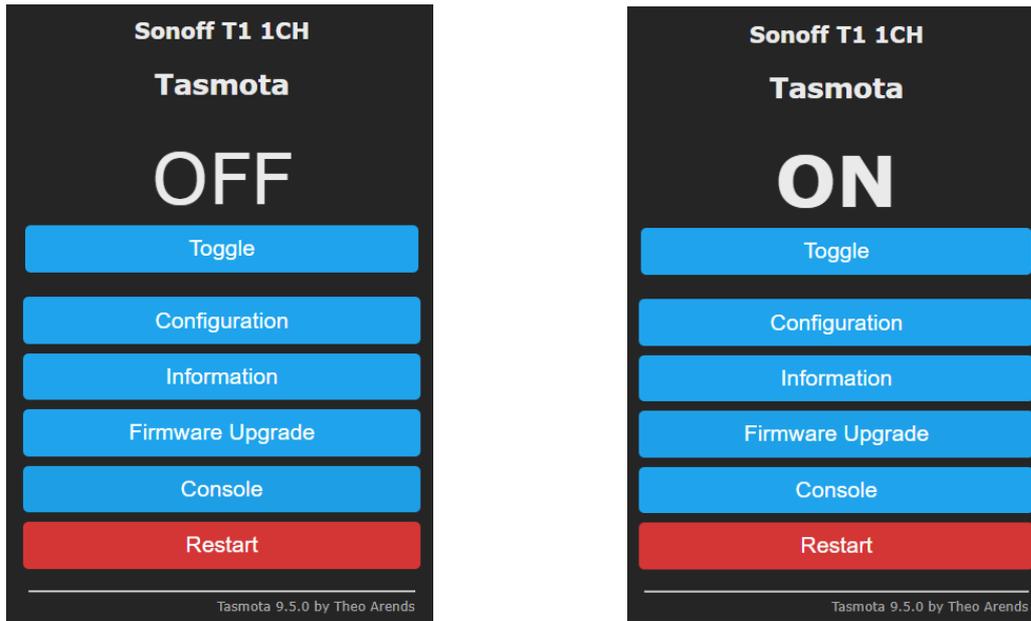


Figura 3.37 Interfaz de encendido y apagado SONOFF

Puerta

Al igual que en las páginas *Web* antecesoras, su codificación fue realizada en HTML desde la programación en el entorno de desarrollo de Arduino. La sección en cuestión se visualiza en la Figura 3.38 y muestra los parámetros que se pueden configurar en la cámara tales como: resolución de imagen, saturación, brillo, modo espejo, entre otros.

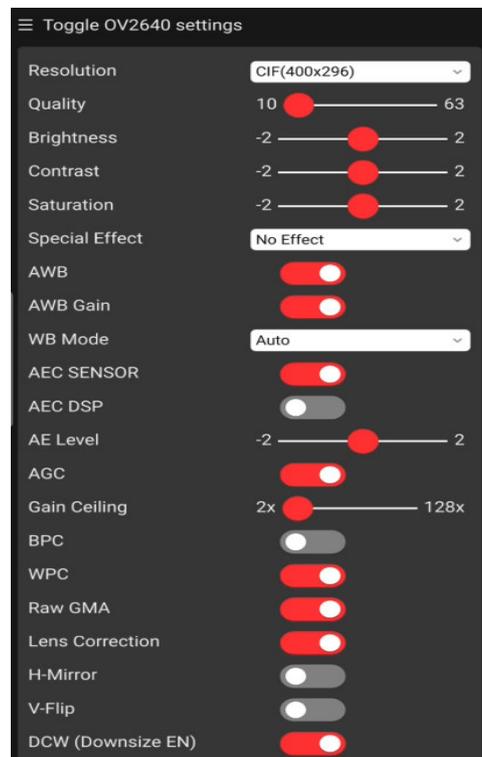


Figura 3.38 Página *Web* ESP32-CAM

El código completo del menú se observa en el Anexo 4.8.

3.4 Implementación del sistema domótico

En primera instancia, el miniordenador *Raspberry Pi* fue colocado dentro del dormitorio en el cual se encuentran la mayoría de los dispositivos a ser manejados mediante el sistema domótico, al ser la comunicación de forma inalámbrica mediante red wifi se debe considerar la cobertura que ofrece el enrutador.

Implementación del sistema en las persianas

Para la implementación del control remoto de la persiana se procedió a cargar el algoritmo previamente desarrollado sobre el módulo ESP32, tomando en consideración que dentro del apartado preferencias del IDE de Arduino se añadiera la siguiente URL misma que permite reconocer el dispositivo para cargar el código sobre el mismo. La URL https://dl.espressif.com/dl/package_esp32_index.json fue añadida en el apartado Gestor de URLs Adicionales de Tarjetas ubicado en el apartado Archivo - Preferencias tal y como se muestra en la Figura 3.39.

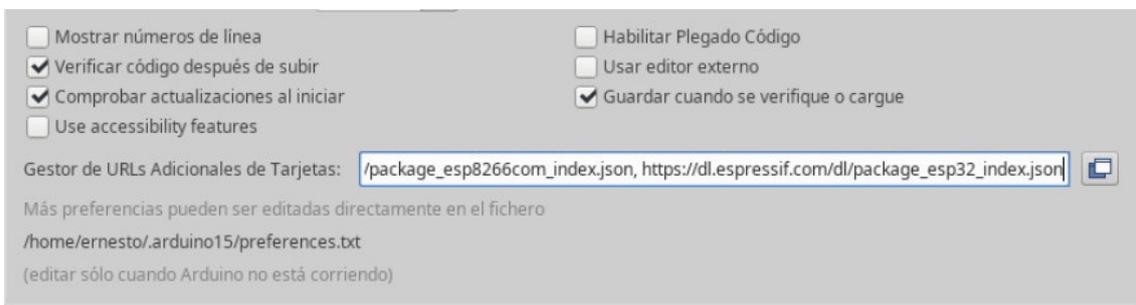


Figura 3.39 URL para instalar la tarjeta ESP32

Ya añadida la URL dentro del gestor de tarjetas del IDE de Arduino aparecerá el módulo ESP32 el cual fue instalado y debidamente seleccionado antes de cargar el código, ver Figura 3.40.

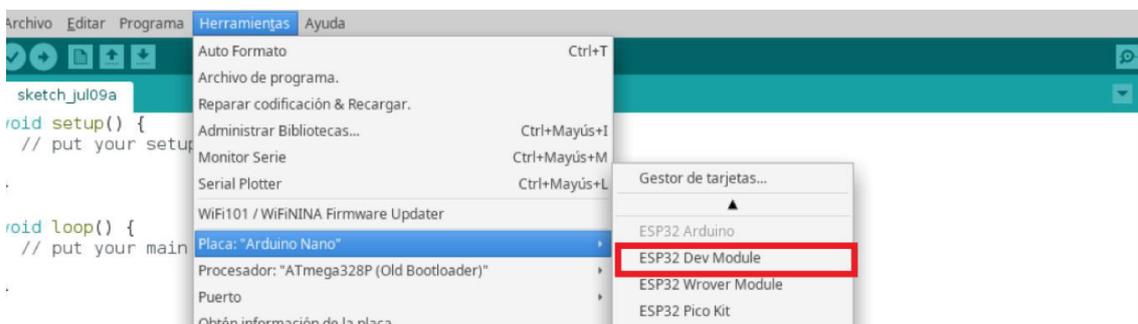


Figura 3.40 Selección de módulo ESP32 *Dev Module*

La persiana instalada en la vivienda es de 1 (m) x 1 (m), ver Figura 3.41, misma que funciona con la ayuda de un motor Nema 17 el cual se encuentra atornillado sobre la pared con un piñón, ver Figura 3.42, diseñado para encajar con la cadena de la persiana y de esta forma a través de la interfaz *Web* controlar el cierre o apertura de la misma. El motor paso a paso será controlado a través de un driver llamado A4988, el cual permite controlar el sentido de giro de dicho motor, se lo puede visualizar dentro de toda la configuración del circuito para el control de la persiana en la Figura 3.7.

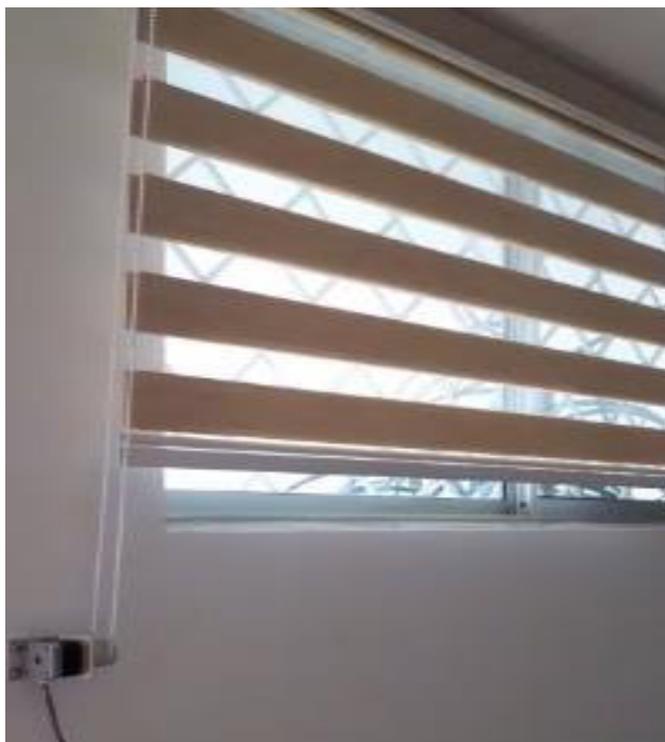


Figura 3.41 Persiana



Figura 3.42 Motor paso a paso Nema 17

Adicionalmente, el controlador A4988 (ver Figura 3.43) permite regular la corriente máxima enviada al motor paso a paso a través del *preset* regulador para evitar sobrecargas. La corriente máxima de trabajo para el motor Nema 17 es de 1.5 (A) y para su respectivo ajuste se procede a utilizar la Ecuación 3.2:

$$I_{max} = \frac{V_{ref}}{(8 * R_S)}$$

$$V_{ref} = I_{max} * 8 * R_S$$

$$V_{ref} = 1.5 (A) * 8 * 0.1(\Omega)$$

$$V_{ref} = 1.2 (V)$$

Ecuación 3.2 Establecimiento de la corriente máxima

Es decir, con la ayuda de un multímetro conectado su terminal positiva en el *preset* y el negativo en el pin 9 deberá estar regulado los 1.2 (V).

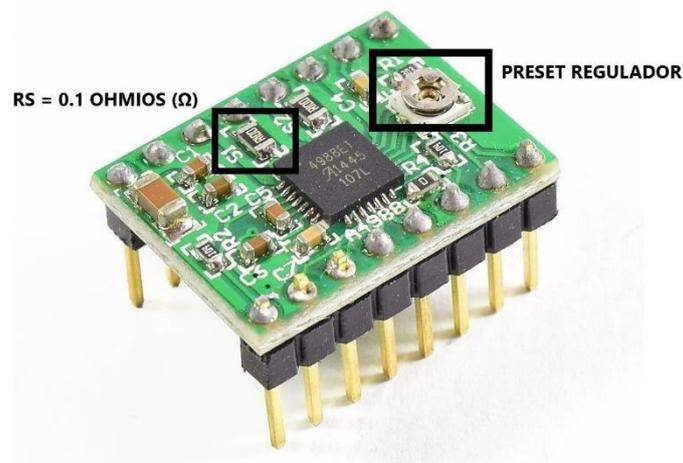


Figura 3.43 Controlador A4988

Implementación del encendido y apagado de la luz

Antes de instalar el interruptor inteligente SONOFF para el encendido y apagado de la iluminación, se debe cargar el algoritmo de programación sobre una de las dos tarjetas con las que cuenta el mismo. Aquella programación es a través de un *firmware* llamado Tasmota que permite integrar el interruptor inteligente a la interfaz *Web* creada para el control del sistema domótico en general. Para aquello, una vez desmontada las placas se deberá identificar cuatro pines de conexión los cuales se detallan en la Figura 3.44 .

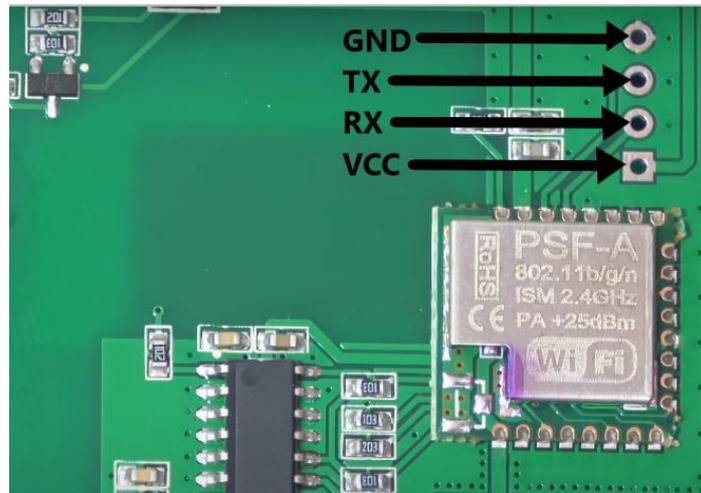


Figura 3.44 Distribución de pines en placa SONOFF

Una vez observado la distribución de pines, a través de un conversor de USB a TTL, se procedió cargar el código sobre la placa. Tomar en cuenta que tanto RX y TX en el conversor como en la placa deberán ir cruzados. En la Figura 3.45 se puede apreciar las conexiones realizadas antes de cargar el código.

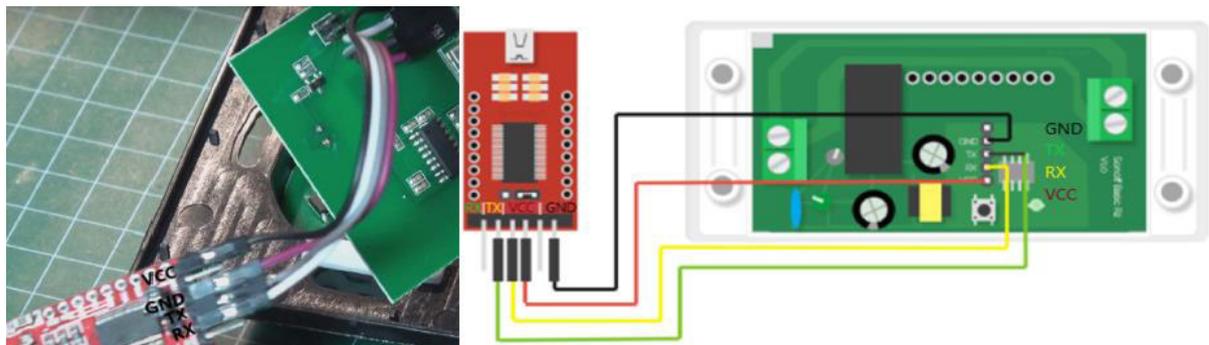


Figura 3.45 Conexión de conversor de USB a TTL con placa de SONOFF

Ya cargado el código sobre la placa, para realizar la implementación del interruptor inteligente SONOFF se procedió a retirar la caja de interruptor convencional permitiendo visualizar los dos únicos cables (fase y retorno del foco) existentes en la instalación, tal y como se indicó en la Figura 3.2. Dicha instalación se la pueda observar a continuación en la Figura 3.46.

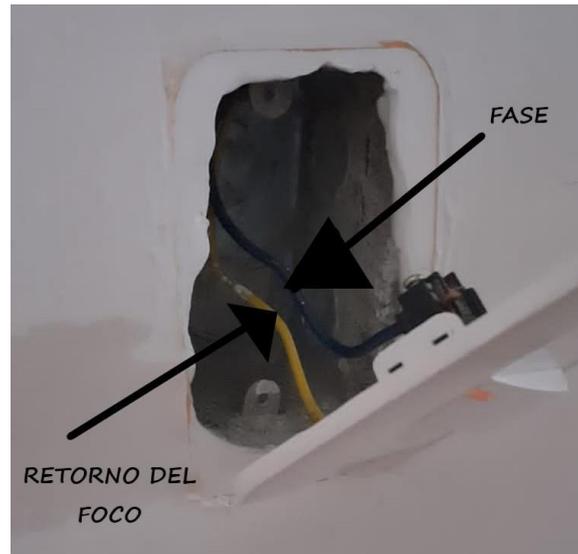


Figura 3.46 Instalación convencional de un foco en el hogar

En base a aquello, se procedió a realizar la instalación de un nuevo cable neutro (ver Figura 3.3) desde un extremo del foco de la vivienda hasta el punto de salida en la caja donde se colocará el interruptor inteligente. La instalación realizada se la puede observar en la Figura 3.47.

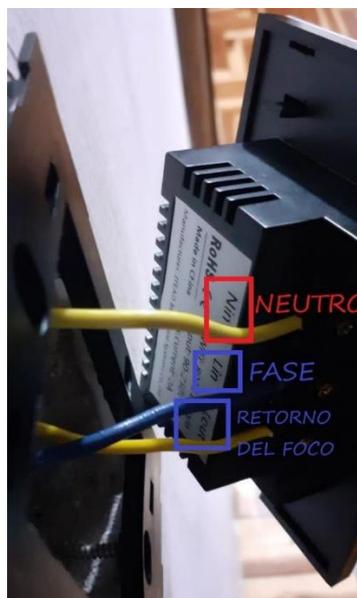


Figura 3.47 Instalación del interruptor inteligente SONOFF

Finalmente, luego de colocar y atornillar los tres cables necesarios para el funcionamiento del interruptor, se puede apreciar el interruptor empotrado en la pared del domicilio, ver Figura 3.48.



Figura 3.48 Interruptor SONOFF CH1

Implementación del encendido y apagado de computadoras

En la sección de implementación correspondiente al encendido y apagado remoto de las computadoras, se los hizo con dos ordenadores; lo que se realizó fue un enlace cableado de aproximadamente 20 metros desde la posición del equipo enrutador hasta el dormitorio donde se encuentra el primer computador, como se muestra en la Figura 3.49, el cable utilizado fue un par trenzado categoría 5e.

El segundo computador se colocó en otro lugar del hogar para que sea posible demostrar que el control remoto pueda darse desde cualquier posición, lo importante es que el equipo haya sido conectado mediante cable de red, para de esta manera activar el mismo mediante el paquete mágico que trabaja en conjunto con la tarjeta de red. Cabe recalcar que las instrucciones de encendido y apagado remoto se ejecutarán mediante el código desarrollado y explicado previamente en lenguaje PHP.



Figura 3.49 Cableado entre el computador y el *router*

Implementación del control de acceso

Para el apartado correspondiente a la puerta, se implementó el circuito y se colocó el solenoide o cerradura electrónica en la parte superior de la puerta, tal como se muestra en la Figura 3.50, de esta manera la puerta puede ser asegurada para no permitir el ingreso de personas no autorizadas.



Figura 3.50 Solenoide instalado

Por otra parte, la cámara que contiene el módulo ESP32-CAM se situó en la parte externa al domicilio (ver Figura 3.51) para que pueda realizar el reconocimiento facial. Se procedió a tomar muestras de las facciones de la persona que tiene acceso a la

interfaz mediante la opción denominada como “enroll face”. La cámara toma alrededor de 5 muestras del rostro de la persona a la cual se le va a permitir el ingreso para posteriormente guardar dichos datos en la memoria interna del dispositivo. Es importante que además se encuentren activadas las opciones de “Face detection” y “Face recognition”.



Figura 3.51 ESP32-CAM colocado en la parte externa

Presupuesto del proyecto

Los costos totales de materiales para la implementación se aprecian en la Tabla 3.1:

Tabla 3.1 Costos de materiales y mano de obra de la implementación

	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
Microprocesador	<i>Raspberry Pi</i>	1	70	70
	Cargador – Alimentación (5V)	1	5	5
	Tarjeta SD 32 GB	1	5	5
Computadores	Cable par trenzado cat 5e con conectores	2	6	12
Iluminación	Interruptor inteligente SONOFF	1	35	35
	Convertor USB a TTL	1	3	3
Persianas	Módulo ESP32	1	15	15
	Motor Nema 17	1	13	13
	Persiana tipo zebra	1	15	15
	Driver A4988	1	4.5	4.5
Puerta	ESP32-CAM	1	15	15
	Módulo relé para arduino	1	5	5
	Componentes electrónicos varios	1	4	4
Mano de obra	Gestión de instalación del sistema por parte de los dos integrantes del proyecto	2	250	500
			TOTAL	701.5

3.5 Verificación del funcionamiento del sistema domótico

En primer lugar, se verificó el funcionamiento del sistema de registro de usuarios e inicio de sesión. Se procedió a realizar los siguientes pasos:

1. Seleccionar la opción de registro e ingresar el nuevo usuario y contraseña.
2. El sistema arroja un mensaje que indica que se ha realizado el proceso con éxito por lo cual se procede a validar esta notificación en la base de datos, tal como se indica en la Figura 3.52.
3. Se procede a iniciar sesión al sistema domótico con las credenciales respectivas.
4. Se autentica el usuario y se da la opción de ser redirigido al menú principal o cerrar sesión.

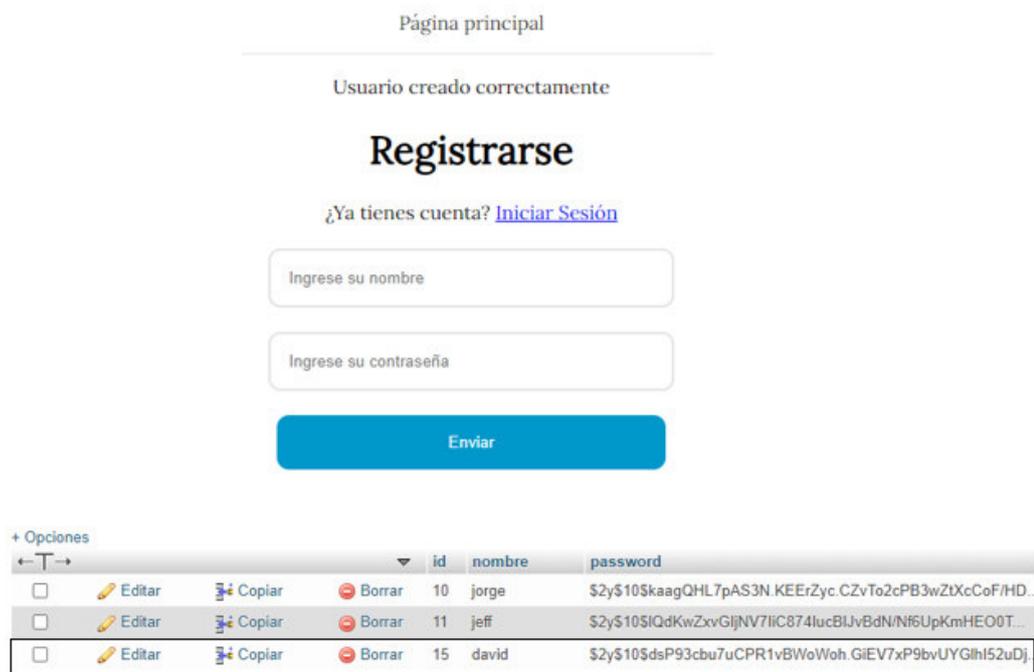


Figura 3.52 Verificación de registro

Una vez en el menú inicial se verificaron las cuatro diferentes aplicaciones del sistema y se procedió a probar cada una de ellas.

Verificación del encendido y apagado de computadores

Para verificar el funcionamiento del accionamiento remoto de computadores se trabajó junto con la matriz de estado que permite validar los cambios de cada dispositivo. La primera función en ser verificada fue la de encendido remoto, seleccionando uno de los dos computadores para posteriormente ejecutar el comando que permite a la *Raspberry*

PI enviar el paquete mágico que realiza la actividad previamente mencionada. Este proceso se detalla en la Figura 3.53.

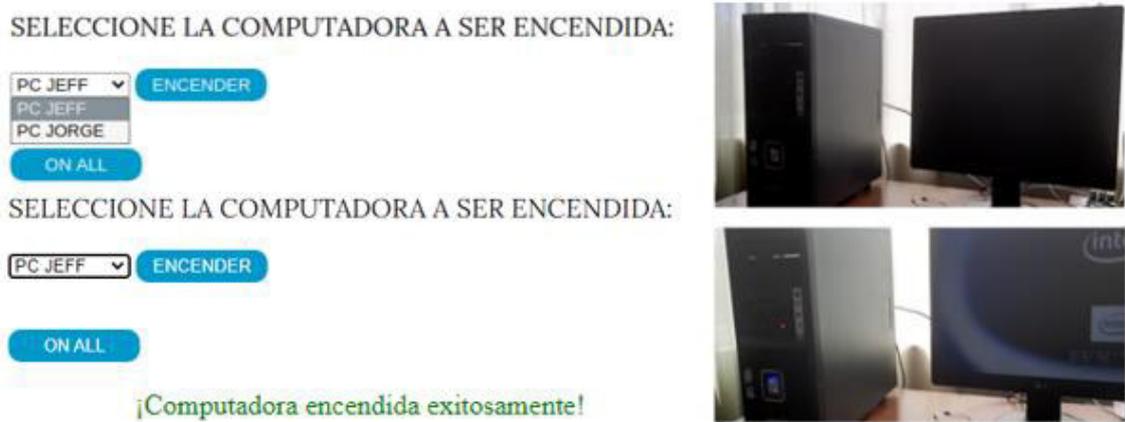


Figura 3.53 Verificación de encendido

Posteriormente se usó la opción llamada “ON_ALL” para encender el otro computador, ver Figura 3.54.



Figura 3.54 Verificación de todas las computadoras encendidas

Para el procedimiento de apagado de igual manera se detalla el proceso en la Figura 3.55.

SELECCIONE LA COMPUTADORA A SER APAGADA:

PC JEFF ▾ APAGAR
PC JEFF
PC JORGE
OFF ALL



SELECCIONE LA COMPUTADORA A SER APAGADA:

PC JORGE ▾ APAGAR
OFF ALL



¡Computadora apagada exitosamente!

Figura 3.55 Verificación de apagado

Finalmente se procedió a usar la función “*OF_ALL*” para apagar la computadora restante. Ver Figura 3.56.

SELECCIONE LA COMPUTADORA A SER APAGADA:

PC JEFF ▾ APAGAR
OFF ALL



¡Todas las computadoras apagadas exitosamente!



Figura 3.56 Verificación de todas las computadoras apagadas

Verificación del control de acceso

Ahora bien, para validar el estado del funcionamiento de la seguridad de la puerta, en primera instancia una persona que no corresponde al sistema se acercó al detector facial de la cámara para que se pueda visualizar la alerta de intruso y el solenoide no se active. De igual forma se pudo validar la activación del solenoide al momento de que una persona que tiene permitido el ingreso se acercó y el reconocimiento facial realizó su labor. Este proceso se visualiza en la Figura 3.57.



Figura 3.57 Verificación de seguridad de la puerta

Verificación del control de la iluminación

En relación con la verificación del funcionamiento de la interfaz *Web* para el control de la iluminación, se puede validar a través de las Figura 3.58 y Figura 3.55 que responde a las peticiones realizar desde la página *Web* hacia el interruptor inteligente.

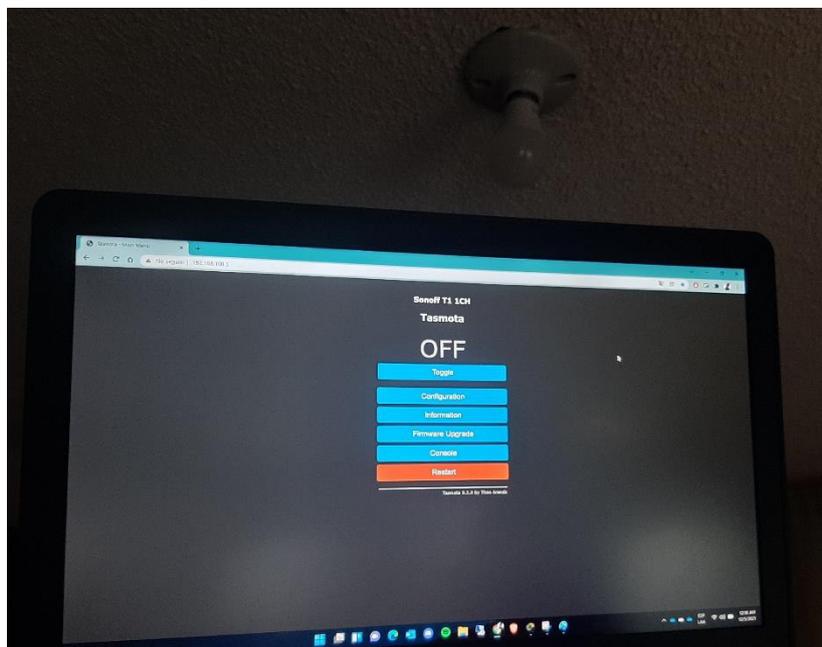


Figura 3.58 Apagado remoto de interruptor SONOFF

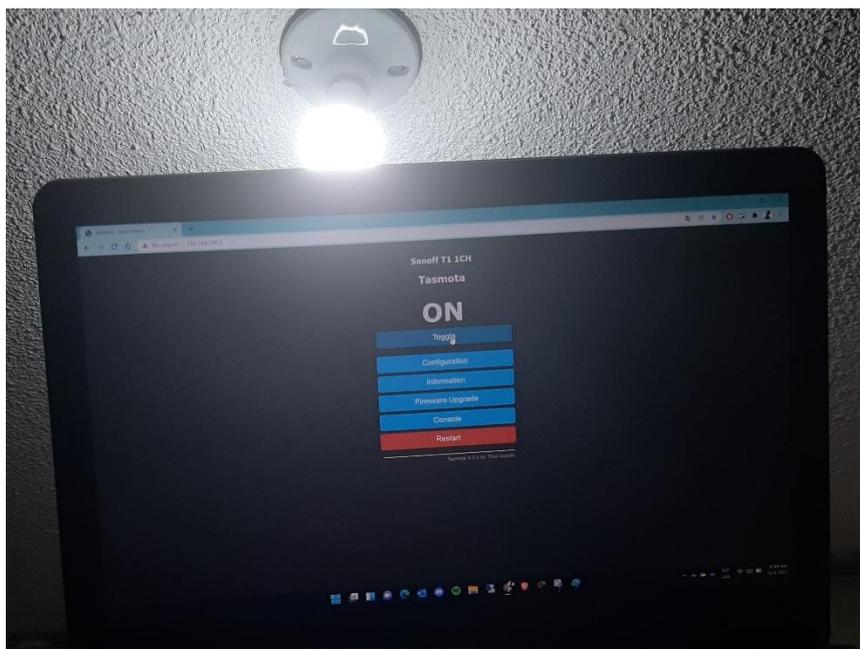


Figura 3.59 Encendido remoto de interruptor SONOFF

Finalmente, para lograr la automatización de la persiana a través de la *Web*, se puede validar a través de la terminal del IDE de Arduino, mediante el envío de las peticiones a través del método GET para el cierre y apertura de la persiana. Esto se muestra a continuación en la Figura 3.60. Finalmente, en el Anexo 2 se puede visualizar el Manual de Usuario para el administrador del prototipo, mientras que el Anexo 3 corresponde al Manual de mantenimiento del sistema domótico.

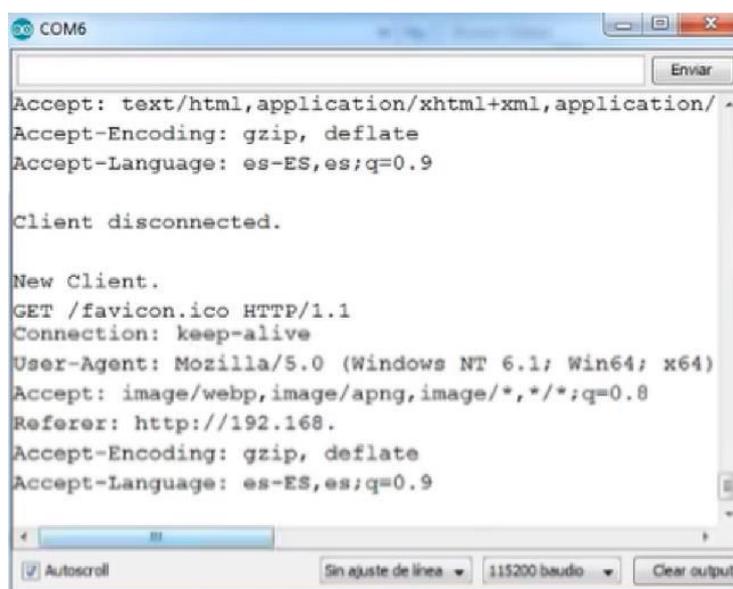


Figura 3.60 Envío de peticiones al motor paso a paso

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- En relación al bajo costo de la gama de miniordenadores *Raspberry Pi* en comparación al resto de equipos inteligentes utilizados en la actualidad, estos permiten desarrollar complejos sistemas domóticos sin tener que invertir grandes cantidades de dinero para automatizar el hogar.
- Para evitar enviar órdenes a través del protocolo SSH desde la distribución *Linux Raspbian* al minicomputador, el diseño de una interfaz *Web* para su control es la mejor opción para ser presentado al usuario final debido a que no todos poseen el conocimiento para ejecutar ordenes mediante una terminal de comandos.
- Diseñar una aplicación compatible con diversos sistemas operativos en el mercado resulta difícil, por lo que desarrollar páginas *Web* que necesiten de un navegador *Web* instalado en el dispositivo a enviar las ordenes, resulta más fácil y práctico tanto para el desarrollador como para el usuario final.
- El desarrollo de sistemas domóticos no se va a fijar a futuro en tan solo proporcionar confort y seguridad, sino que juntarán estos aspectos con servicios de eficiencia energética y ahorro en el hogar.
- El desarrollo de sistemas domóticos no solo está enfocado en automatizar procesos cotidianos dentro del hogar sino más bien busca poco a poco ofrecer esta tecnología en hospitales, escuelas, colegios e incluso en lugares públicos.
- La interfaz *Web* de control al encontrarse desarrollada sobre código abierto no necesita licencia por lo que, si se busca seguir incrementando las prestaciones sobre los diferentes lugares de la vivienda, se podrá realizar sin ningún inconveniente.
- Analizados los diferentes módulos para el desarrollo de sistemas domóticos, se determinó que el ESP32 además de contar con varios pines a los cuales a través de programación se los puede comandar, es posible enviar ordenes al dispositivo mediante la red *Wi-Fi* lo que facilita la automatización.
- La domótica como nueva tendencia tecnológica es una manera práctica y útil de monitorear y controlar remotamente cualquier artefacto del hogar, lo que permite generar una nueva oportunidad de negocio.
- Para que exista una comunicación correcta entre el computador y la *Raspberry Pi* al momento de generar las órdenes de apagado del ordenador, es necesario que este tenga un sistema operativo de *Windows 8* o superior para no generar

problemas con el apartado de herramientas administrativas y directivas de seguridad local. En caso de tener versiones inferiores, es necesario editar el registro de *Windows* agregando valores *DWORD*.

- El sistema en conjunto fue desarrollado como página *Web* para que cualquier dispositivo con acceso a un navegador *Web* pueda ingresar al mismo, ya que crear una aplicación que pueda ser compatible con diferentes sistemas operativos tales como *Android*, *iOS*, *Windows*, *Linux*; resulta mucho más complicado.
- El lenguaje *PHP* es óptimo para la ejecución de comandos en *Linux* ya que permite llamar al terminal y realizar las operaciones de este mediante las líneas de programación.

4.2 Recomendaciones

- Para evitar problemas de conexión con cada uno de módulos que forman parte del sistema domótico, asegurarse que la señal de *Wi-Fi* a través del *router* se irradie de manera correcta a cada uno de ellos ubicados alrededor de toda la vivienda.
- Para evitar obtener errores al momento de cargar el programa de sobre el módulo *ESP32 CAM*, es recomendable usar la alimentación de 5 (V) en lugar de la de 3.3 (V).
- Es recomendable desactivar la administración de energía del estado de vínculos dentro de los parámetros *PCI express* para evitar inconvenientes con el encendido remoto mediante el protocolo *WOL*.
- La página *Web* del sistema es *http* ya que al ser local sin acceso a Internet solo fue necesario el uso del puerto 80. Sin embargo, si se realiza un proyecto a mayor escala con una IP pública es necesario la apertura del puerto 443 por motivos de seguridad.
- Es recomendable usar el sistema operativo "*Raspbian*" para el manejo del miniordenador ya que es el oficial y por lo tanto es menos propenso a presentar errores de instalación o ejecución de aplicaciones.
- En el caso de añadir interruptores inteligentes *SONOFF* al hogar, es recomendable adquirir de varios canales, para poder comandar varios focos mediante un solo interruptor.
- Se debe desenergizar el hogar al momento de realizar la conexión electrónica del dispositivo *SONOFF* y asegurarse de realizar una correcta instalación del cable *NEUTRO* desde el foco hasta la caja donde se ubicará el dispositivo.

- El uso de motores paso a paso al instante de automatizar persianas es demasiado útil debido a su enorme torque, lo que permite controlar cualquier tamaño de aquellas, pero tomar a consideración que los tiempos de despliegue y cierre de las mismas no es inmediato.
- Asegurarse de trabajar con las corrientes precisas para poner en funcionamiento el motor paso a paso ya que corrientes superiores a las marcas por el *datasheet* del fabricante provocará que se dañe el motor.

5 REFERENCIAS BIBLIOGRÁFICA

- [1] Tecnología, «ROIBOS,» Domótica: nuevas tecnologías para ganar en confort y sostenibilidad, 6 Abril 2018. [En línea]. Available: <https://roibos.casa/domotica-nuevas-tecnologias-ganar-confort-sostenibilidad/>. [Último acceso: 09 Agosto 2021].
- [2] Ambientum, «Eficiencia Energética,» Eficiencia energética gracias a las viviendas inteligentes, 20 Febrero 2020. [En línea]. Available: <https://www.ambientum.com/ambientum/eficiencia-energetica/eficiencia-energetica-gracias-a-las-viviendas-inteligentes.asp>.
- [3] J. Velasco, «Hipertextual,» Ideas para usar Raspberry Pi en casa, 6 Septiembre 2016. [En línea]. Available: <https://hipertextual.com/2013/09/ideas-usar-raspberry-pi-casa>. [Último acceso: 09 Agosto 2021].
- [4] Vleeko, «SanDoRobotics,» 18 mayo 2019. [En línea]. Available: <https://sandorobotics.com/producto/dev-14643/>. [Último acceso: 24 Agosto 2021].
- [5] M. Bellan, «Programo ERGO SUM,» Primeros pasos con Pines GPIO en Raspberry Pi, 14 08 2018. [En línea]. Available: <https://www.programoergosum.es/tutoriales/introduccion-a-pines-gpio-en-raspbian/>.
- [6] D. R. E. Marmolejo, «Raspberry Pi 3B+,» Hetpro, 22 08 2019. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/raspberry-pi-3-b-plus/>.

- [7] A. Artiz, «Sistema Operativo de Raspberry Pi,» Raspberry Pi, 14 07 2020. [En línea]. Available: <https://www.programoergosum.es/tutoriales/introduccion-a-raspberry-pi/>.
- [8] WayBackMachine, «Wake on Lan,» 13 septiembre 2019. [En línea]. Available: <https://web.archive.org/web/20081217044017/http://gsd.di.uminho.pt/jpo/software/wakeonlan/mini-howto/wol-mini-howto-2.html>. [Último acceso: 24 Agosto 2021].
- [9] «Geek Factory,» [En línea]. Available: <https://www.geekfactory.mx/tienda/domotica/sonoff-basic-interruptor-wifi/>. [Último acceso: 10 Agosto 2021].
- [10] «AV Electronics,» [En línea]. Available: <https://avelectronics.cc/producto/tarjeta-de-desarrollo-esp32-wifi-bluetooth/>. [Último acceso: 10 Agosto 2021].
- [11] F. Ortiz, «Ingeniería, informática y diseño,» 14 09 2019. [En línea]. Available: <https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>.
- [12] Velasco, «Soft Zone,» 29 Mayo 2020. [En línea]. Available: <https://www.softzone.es/programas/linux/raspberry-pi-os/>. [Último acceso: 10 Agosto 2021].
- [13] «Digital Guide Ionos,» 03 Noviembre 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/configuracion/como-configurar-un-servidor-web-raspberry-pi-con-lamp/>. [Último acceso: 11 Agosto 2021].
- [14] F. Alvear, «Electronics,» Programar ESP32 con IDE Arduino, 15 08 2019. [En línea]. Available: <https://www.taloselectronics.com/blogs/tutoriales/programar-esp32-con-ide-arduino>.

ANEXOS

ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO



ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 3 de enero de 2022

CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, Gabriela Katherine Cevallos Salazar, docente a tiempo completo de la Escuela Politécnica Nacional y como directora de este trabajo de titulación, certifico que he constatado el correcto funcionamiento de la Implementación de un sistema domótico basado en Raspberry Pi para el hogar, mismo que fue implementado por los estudiantes Jefferson Minalla y Jorge Vallejo.

DIRECTOR

Ing. Gabriela Cevallos, Msc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 28. EXT: 2729
email: gabriela.cevalloss@epn.edu.ec

Quito-Ecuador

ANEXO 2: MANUAL DE USUARIO PARA EL ADMINISTRADOR DE PROTOTIPO

Para poder ingresar al sistema domótico es necesario ingresar a algún navegador WEB de preferencia mediante la dirección IP configurada por el administrador de forma estática o brindado de forma automática por el enrutador mediante DHCP. En específico en esta implementación se debe ingresar a la dirección 192.168.100.7/home; donde /home es la ruta que presenta la página inicial. Una vez dentro de la página se ofrece las opciones de crear una cuenta o registrarse.



Si se ingresa al apartado de registro se debe ingresar las credenciales respectivas para poder iniciar sesión posteriormente

Registrarse

¿Ya tienes cuenta? [Iniciar Sesión](#)

Una vez se haya creado la cuenta en la interfaz de registro, se ingresa a la opción de iniciar sesión ingresando las credenciales previamente creadas.

Iniciar Sesión

¿No tienes cuenta? [Registrarse](#)

Ahora bien, si el administrador desea ingresar a la base de datos para eliminar usuarios que ya no van ingresar al sistema, se debe dirigir a la dirección: 192.168.100.7/phpmyadmin; la cual redirige a dicha ruta. Se debe ingresar a la base de datos con las contraseñas:

Usuario: root

Contraseña: raspberry



Una vez dentro se procede a ingresar a la tabla denominada como usuarios en la cual se van a encontrar todos los clientes registrados del sistema. En este apartado es posible editar los nombres de usuario e incluso borrar a los que ya no pertenezcan para evitar intrusos. Además por motivos de seguridad la contraseña de cada usuario aparece como encriptada. Si se desea eliminar la encriptación se puede realizar en el código de programación del archivo register.php, específicamente en la línea siguiente:

```
$password = password_hash($_POST['password'], PASSWORD_BCRYPT
```

Se elimina la función *PASSWORD_BCRYPT* para realizar lo mencionado anteriormente.

Mostrar todo |
 Número de filas: 25 |
 Filtrar filas: |
 Sort by key: Ninguna

- Opciones

				id	nombre	password
<input type="checkbox"/>				10	jorge	\$2y\$10\$kaagQHL7pAS3N.KEErZyc.CZvTo2cPB3wZtXcCoF/HD...
<input type="checkbox"/>				11	jeff	\$2y\$10\$IQdKwZxvGijNV7IiC874IucBIJvBdN/Nf6UpKmHEO0T...
<input type="checkbox"/>				15	david	\$2y\$10\$dsP93cbu7uCPR1vBW0Woh.GiEV7xP9bvUYGhI52uDj...

Seleccionar todo |
 Para los elementos que están marcados:
 Editar |
 Copiar |
 Borrar |
 Exportar

ANEXO 3: MANUAL DE MANTENIMIENTO DEL SISTEMA DOMÓTICO

En caso de que el sistema domótico presente un funcionamiento inapropiado, es recomendable proceder a reiniciar el miniordenador Raspberry Pi a través del software VNC Viewer de tal manera que mediante el siguiente comando:

```
sudo shutdown -r now
```

Una vez reiniciado el minicomputador, el servicio web levantado a través de Apache al igual que la base de datos ejecutada mediante MariaDB se ejecutarán de manera automática permitiendo utilizar el sistema domótico con normalidad.

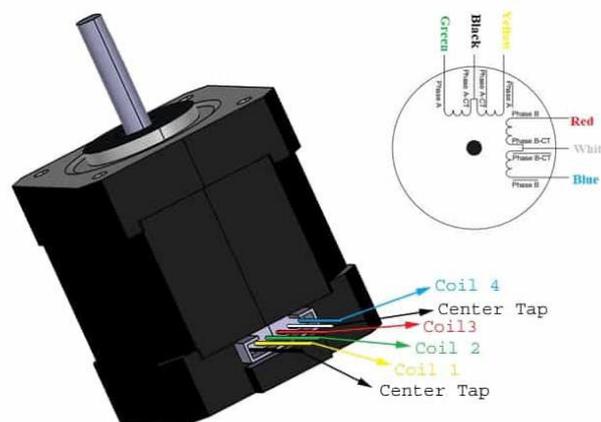
CONTROL DE TEMPERATURA

Por otro lado, en caso de que la Raspberry Pi presente altas elevaciones de temperatura se deberá instalar un disipador de calor al igual que la instalación de un ventilador. Para poder visualizar la temperatura del procesador y la frecuencia en hercios del miniordenador se deberá utilizar el siguiente comando:

```
-n 1 "vcgencmd measure_clock arm; vcgencmd measure_temp"
```

CARACTERÍSTICAS DE FUNCIONAMIENTO DE MOTOR PASO A PASO NEMA 17

Para un correcto funcionamiento del motor paso a paso Nema 17 se debe trabajar sobre el mismo a una tensión máxima de 5 (V). Un punto importante dentro del control para una correcta operación del motor es que ambas bobinas se encuentren funcionales por lo que es recomendable verificar a través de un multímetro la continuidad en cada una de ellas tal y como se muestra a continuación:



CARACTERÍSTICAS DE FUNCIONAMIENTO DE INTERRUPTOR INTELIGENTE SONOFF

- Rango de voltaje: 90-250 (V) AC 50/60 (Hz)
- Corriente máxima: 2 (A)
- Máximo de potencia: 400 (Vatios)
- Frecuencia: 2.4 (GHz)

En la Figura 3.43 se puede visualizar la respectiva conexión de cada uno de los conductores que permiten el correcto funcionamiento del interruptor inteligente. Obligatoriamente de debe contar en la instalación eléctrica con los siguientes cables: **NEUTRO, FASE, RETORNO DEL FOCO**. Para poder verificar el funcionamiento en los mismos, se puede hacer uso de un multímetro y validar a través del común neutro el funcionamiento de casa una de las fases.

ANEXO 4: ALGORITMOS DE PROGRAMACIÓN

SISTEMA DE REGISTRO E INICIO DE SESIÓN

Index.php (Anexo 4.1)

```
<?php
    session_start(); // Se inicia la sesión
    require 'database.php'; // LLama a la base de datos con nombre
database.php
    if (isset($_SESSION['user_id'])){ // Si existe la sesión está iniciada, consulta
valores de nombre y contraseña de la tabla llamada usuarios
        $records = $conn->prepare('SELECT id, nombre, password FROM usuarios
WHERE id =:id'); //Realiza lo indicado en la línea anterior
        $records->bindParam(':id', $_SESSION['user_id']); // Se vincula la sesión con
los datos respectivos
        $records->execute();
        $results = $records->fetch(PDO::FETCH_ASSOC); //Se guardan los
resultados de los datos obtenidos desde la base

        $user = null;

        if(count($results) > 0){
            $user = $results; //Se llena la variable en caso de que existan los
resultados
```

```

    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"> <!-- Se determina el tipo de codificación de
caracteres, el tipo de fuente y la ruta de los estilos en CSS -->
    <title>HOME</title>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="assets/css/estilos.css">
</head>
<body>

    <?php require 'partials/header.php' ?>

    <?php if(!empty($user)): //Se valida si no está vacía la variable de usuarios,
es decir si ya está iniciada la sesión ?>
        <br>Bienvenido/a <?= $user['nombre'] //En caso de que esté
logueado, se lee el nombre de usuario y se arroja el mensaje de bienvenida ?>
        <br>Has ingresado satisfactoriamente
        <div>
            <a href="menu/menu-principal.html">Ingresar al menu del sistema
domótico</a> ó <a href="logout.php">Cerrar Sesión</a>
        </div>

        <?php else: ?>

        <h1>PROTOTIPO DE SISTEMA DOMÓTICO CON RASPBERRY PI -
HOGARES</h1>
        <a href="login.php">Iniciar sesión</a> ó <a
href="register.php">Registrarse</a>
    <?php endif; ?>
</body>
</html>

```

Database.php (Anexo 4.2)

```
<?php
```

```

$server = 'localhost'; //Nombre del servidor, en este caso es el servidor local de la
raspberry
$username = 'root'; //Usuario de inicio de sesión en la base de datos
$password = 'asdq'; //Contraseña del usuario de la base de datos
$database = 'login-base'; //Nombre de la base de datos

try {
    $conn = new PDO("mysql:host=$server;dbname=$database;", $username,
$password); //Se intenta conectar a la base de datos con las credenciales respectivas
} catch (PDOException $e) {
    die('Conexión Fallida: ' . $e->getMessage()); //En caso de que no se establezca la
conexión se envía el mensaje de error
}

?>

```

Login.php (Anexo 4.3)

```

<?php

session_start(); //Se inicia la sesión
if (isset($_SESSION['user_id'])){ //Si la sesión está iniciada con las credenciales
correctas se procede a autenticar y redirigir a la página de bienvenida
    header('Location: /home');
}

require 'database.php'; //Se llama a la base al archivo php que requiere la base de
datos

if (!empty($_POST['nombre']) && !empty($_POST['password'])){ //Si no se encuentran
vacíos los espacios de usuario y contraseña
    $records = $conn->prepare('SELECT id, nombre, password FROM usuarios
WHERE nombre=:nombre'); //Se procede a validar los datos ingresados con los de la
tabla usuarios de la base de datos llamada login-base.
    $records->bindParam(':nombre', $_POST['nombre']); //vincula los datos
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);
    $message = "";

    if(count($results) > 0 && password_verify($_POST['password'],
$_SESSION['password'])){ //Valida las credenciales
        $_SESSION['user_id'] = $results['id'];
        header('Location: /home'); //En caso de que las credenciales
coincidan, dirige a la página de bienvenida
    }
}

```

```

    } else{
        $message = 'Las credenciales no son correctas'; //En caso de que
        las credenciales no coincidan se despliega el mensaje de error
    }
}
?>

```

```

<!DOCTYPE html>
<!-- En esta sección se configura el formulario de inicio de sesión, se configuran sus
elementos de ingreso de datos como casillas y los elementos de envío de datos como
botones, además se redirecciona a los diferentes archivos en caso de que se requiera
crear un nuevo usuario. El diseño de la página dirige los estilos de css. -->
<html>
<head>
    <title>LOGIN</title>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="assets/css/estilos.css">
</head>
<body>
    <?php require 'partials/header.php' ?>
    <h1>Iniciar Sesión</h1>
    <span>¿No tienes cuenta? <a href="register.php">Registrarse</a></span>

    <?php if (!empty($message)) : ?>
        <p><?= $message ?></p>
    <?php endif;?>

    <form action="login.php" method="post">
    <input type="text" name="nombre" placeholder="Ingrese su nombre">
    <input type="password" name="password" placeholder="Ingrese su
contraseña">
    <input type="submit" value="Enviar">
</form>
</body>
</html>

```

Logout.php (Anexo 4.4)

```

<?php
session_start(); //Se inicia la sesión

```

```
session_unset(); //Se quita la sesión
```

```
session_destroy(); //Se destruye la sesión
```

```
header('Location: /home'); //Se redirecciona a la página principal
```

```
?>
```

Register.php (Anexo 4.5)

```
<?php
```

```
require 'database.php'; //Se llama a la base de datos en MariaDB
```

```
$message = "";
```

```
if (!empty($_POST['nombre']) && !empty($_POST['password'])) {
```

```
    $sql = "INSERT INTO usuarios (nombre, password) VALUES (:nombre, :password)";
```

```
//Se llena los datos ingresados mediante el formulario de la página web en la tabla de usuarios dentro de la base de datos
```

```
    $stmt = $conn->prepare($sql); // Se prepara a la base de datos para asociar los
```

```
datos
```

```
    $stmt->bindParam(':nombre', $_POST['nombre']); //Se vincula el nombre del usuario
```

```
ingresado en la página web dentro de la tabla de usuarios
```

```
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT); //Se
```

```
encripta la contraseña
```

```
    $stmt->bindParam(':password', $password); // Se vincula la contraseña del usuario
```

```
ingresado en la página web dentro de la tabla de usuarios
```

```
    if ($stmt->execute()) {
```

```
        $message = 'Usuario creado correctamente'; //Si se crea el usuario correctamente se despliega el mensaje indicando
```

```
    } else {
```

```
        $message = 'Ha ocurrido un error al crear un nuevo usuario'; //Si no se pudo crear el usuario se despliega el mensaje indicando el inconveniente
```

```
    }
```

```
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<!-- En esta sección se configura el formulario de registro, se configuran sus elementos de ingreso de datos como casillas y los elementos de envío de datos como botones,
```

además se redirecciona a los diferentes archivos en caso de que se requiera iniciar sesión. El diseño de la página redirige los estilos de css -->

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Registrarse</title>
    <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="assets/css/estilos.css">
  </head>
  <body>

    <?php require 'partials/header.php' ?>

    <?php if(!empty($message)): ?>
      <p><?= $message ?></p>
    <?php endif; ?>

    <h1>Registrarse</h1>
    <span>¿Ya tienes cuenta? <a href="login.php">Iniciar Sesión</a></span>

    <form action="register.php" method="post">
      <input name="nombre" type="text" placeholder="Ingrese su nombre">
      <input name="password" type="password" placeholder="Ingrese su contraseña">
      <input type="submit" value="Enviar">
    </form>

  </body>
</html>
```

Estilos.css (Anexo 4.6)

```
body{
  margin: 0;
  padding: 8% 0;
  font-family: 'Lora', serif;
  text-align: center;
  background:url(../epnfondo.png);
  background-repeat:no-repeat;
  background-attachment:fixed;
}

input[type="text"], input[type="password"]{
  outline: none;
```

```

padding: 15px;
display: block;
width: 300px;
border-radius: 10px;
border: 2px solid #e5e5e5;
margin: 20px auto;
}

input[type="submit"]{
padding: 15px;
color: #fff;
background: #0098cb;
width: 320px;
margin: 20px auto;
margin-top: 0;
border: 0;
border-radius: 10px;
cursor: pointer;
}

input[type="submit"]:hover {
background-color: #00b8eb;
}

header{
border-bottom: 2px solid #eee;
padding:15px 0;
margin-bottom: 10px;
width: 100%;
text-align: center;
}

header a {
text-decoration: none;
color: #333;
}

```

Header.php (Anexo 4.7)

```

<header>
    <a href="/home">Página principal</a>
</header>

```

MENU PRINCIPAL

Menu-principal.html (Anexo 4.8)

```
<!DOCTYPE html>
<!-- Se configura el título de la página, el encabezado y el cuerpo. Se modifica el tipo
de letra desde google fonts, los estilos de igual manera se codifican en css -->
<html>
<head>
  <meta charset="UTF-8">
  <title>MENU PRINCIPAL</title>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <section class="title">
    <h1>MENU PRINCIPAL PROTOTIPO SISTEMA DOMÓTICO
PARA HOGARES</h1>
  </section>
  <nav class="navegacion">
    <ul class="menu">
      <li class="first-item">
        <!-- Se selecciona la ruta del archivo html
correspondiente a las computadoras, una vez se de click en este apartado se
redirecciona a dicho enlace. Además se selecciona la ruta del ícono de computador
para la parte del menú -->
        <a href="wol/wakeonlan.html">
          
          <span class="text-
item">COMPUTADORAS</span>
          <span class="down-item"></span>
        </a>
      </li>
      <li>
        <!-- Se apunta a la dirección correspondiente a
las persianas, una vez se de click en este apartado se redirecciona a dicho enlace.
Además se selecciona la ruta del ícono de persiana para la parte del menú -->
        <a href="http://192.168.100.31">
```

```

class="imagen">
item">PERSIANAS</span>
</li>
</a>
</li>
<li>
<!-- Se apunta a la dirección correspondiente a
las luces, una vez se de click en este apartado se redirecciona a dicho enlace.
Además se selecciona la ruta del ícono de luz para la parte del menú -->
<a href="http://192.168.100.5/">
class="imagen">
item">LUCES</span>
</li>
</a>
</li>
<li>
<!-- Se apunta a la dirección correspondiente a
la puerta, una vez se de click en este apartado se redirecciona a dicho enlace.
Además se selecciona la ruta del ícono de puerta para la parte del menú -->
<a href="http://192.168.100.30">
class="imagen">
item">PUERTA</span>
</li>
</a>
</li>
</ul>
</nav>
</body>
<center>
<footer>
<a href=" ../logout.php">Cerrar sesión y volver a la página
principal</a>
</footer>

```

`</center>`

`</html>`

Styles.css (Anexo 4.9)

```
*{
    margin: 0;
    padding: 0;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

body{
    font-family: 'Lora', serif;
}

a{
    text-decoration: none;
}

ul{
    list-style: none;
}

img{
    max-width: 100%;
}

.title{
    width: 100%;
    background: linear-gradient(180deg, #3639b1, #fd1d1d);
    height: 250px;
    display: flex;
    align-items: center;
    justify-content: center;
}

.title h1{
    color: #fff;
    font-size: 40px;
    font-weight: 200;
    text-align: center;
}
```

```
}
```

```
.navegacion{  
  width: 100%;  
  background: #1c1c1c;  
  display: flex;  
  justify-content: center;  
}
```

```
.menu{  
  width: 1000px;  
  display: flex;  
  flex-wrap: wrap;  
  height: 90px;  
}
```

```
.menu li{  
  flex-basis: 0;  
  flex-grow: 1;  
  border-right: 1px solid #363636;  
  position: relative;  
}
```

```
.menu .first-item{  
  border-left: 1px solid #363636;  
}
```

```
.menu li a{  
  display: flex;  
  flex-direction: column;  
  font-size: 18px;  
}
```

```
.menu .imagen{  
  position: absolute;  
  bottom: 0;  
  right: 30px;  
  width: 0%;  
  z-index: 20;  
  transition: all 300ms cubic-bezier(0.445, 0.050, 0.550, 0.950);  
  transition-timing-function: cubic-bezier(0.445, 0.050, 0.550, 0.950);  
}
```

```
.menu li a .text-item{  
  width: 100%;  
  height: 100%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  text-align: center;  
  color: #fff;  
  top: 0%;  
  position: absolute;  
  z-index: 10;  
  transition: all 500ms cubic-bezier(0.175, 0.885, 0.320, 1.275);  
  transition-timing-function: cubic-bezier(0.175, 0.885, 0.320, 1.275);  
}
```

```
}
```

```
.menu li a .down-item{  
  position: absolute;  
  top: 0%;  
  width: 100%;  
  height: 100%;  
  background: #1c1c1c;  
  transition: all 500ms cubic-bezier(0.215, 0.610, 0.355, 1.000);  
  transition-timing-function: cubic-bezier(0.215, 0.610, 0.355, 1.000);  
}
```

```
}
```

```
.menu li: hover .down-item, .menu li: hover .text-item{  
  top: 100%;  
  height: 115%;  
}
```

```
}
```

```
.menu li: hover .imagen{  
  width: 100%;  
  right: 0px;  
}
```

```
}
```

```
.cerrar{  
  text-align: center;  
  text-decoration: none;  
  padding: 15px;  
  display: inline-block;
```

```

width: 80px;
margin: 20px auto;
margin-top: 0;
border: 0;
border-radius: 20px;
cursor: pointer;
box-shadow: 0 1px 1px rgba(0,0,0,0.3)
}

```

COMPUTADORES

Wakeonlan.html (Anexo 4.10)

```

<html>
  <!-- Cabecera de la página web, se configuran los estilos con css además del tipo
de letra -->
  <head>
    <link rel="stylesheet" type="text/css" href="stylesheet.css">
    <title>CTRL REMOTO PC'S</title>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Lora&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="css/styles.css">
  </head>
  <body>
    <!-- Se modifican los estilos del cuerpo de la página web, se selecciona la ruta en
la cual se encuentran las imagenes para la decoración -->
    <div id=top>
      <h4 class="topwords" id="title">ENCENDIDO Y APAGADO REMOTO DE
COMPUTADORAS
      
      
    </h4>
    </div>
    <div id="main">
      <form action="encender.php" method="post">
        <!-- Se agrega la dirección MAC o física de cada computador a ser
encendido posteriormente, además se agrega el nombre que corresponda a cada host
-->
        <p>SELECCIONE LA COMPUTADORA A SER ENCENDIDA:</p>
        <select name="dropdown">
          <option value="00:1C:C0:91:77:5F">PC JEFF</option>

```

```

        <option value="00:27:0E:0A:82:82">PC JORGE</option>

</select>

    <input type="submit" value="ENCENDER">
</form>
    <!-- Al presionar el boton "ON ALL" se procede a ejecutar el script en php que
enciende a todas las computadoras en una sola acción -->
    <form action="on_all.php" method="post">
        <input type="submit" value="ON ALL">
    </form>
    <!-- Se agregan la dirección IP además del usuario y contraseña de los
computadores a ser apagados, de igual forma se configura el nombre de cada host -->
    <form action="apagar.php" method="post">
        <p>SELECCIONE LA COMPUTADORA A SER APAGADA:</p>
        <select name="dropdown">
            <option value="-I 192.168.100.26 -U jeff%jefferson">PC JEFF</option>
            <option value="-I 192.168.100.27 -U jorge2%asdqw">PC
JORGE</option>
        </select>

        <input type="submit" value="APAGAR">
    </form>
    <!-- Al presionar el boton "OFF ALL" se procede a ejecutar el script en php que
apaga a todas las computadoras en una sola acción -->
    <form action="off_all.php" method="post">
        <input type="submit" value="OFF ALL">
    </form>
    <!-- Enlace que hace referencia a la matriz de estado codificada en php que
permite identificar el estado de cada computador -->
    <a href="ping.php">COMPROBAR EL ESTADO DE LAS
COMPUTADORAS</a>
</div>
</body>
</html>

```

Encender.php (Anexo 4.11)

```

<?php
    $mac = $_POST["dropdown"]; //Toma el valor almacenado en la variable, es
decir la pc escogida por el usuario a ser encendida
    if ($mac) {

```

```

        echo "<font color='green'>¡Computadora encendida
exitosamente!</font>"; //Se envía un mensaje que avisa que la computadora ya se
encendió
        exec("wakeonlan $mac"); //Se ejecuta el wakeonlan en base a la
dirección MAC de la computadora seleccionada
    }
?>

```

Apagar.php (Anexo 4.12)

```

<?php
    $pc      = $_POST["dropdown"]; //Toma el valor almacenado en la variable,
es decir la pc escogida por el usuario a ser apagada
    if ($pc) {
        echo "<font color='red'>¡Computadora apagada
exitosamente!</font>"; //Se envía un mensaje que avisa que la computadora ya se
apagó
        shell_exec("net rpc shutdown $pc -f -t 10 -C APAGANDO PC
DESDE RASPBERRY PI"); //Se ejecuta el comando en base a los datos de la
computadora a ser apagada
    }
?>

```

On_all.php (Anexo 4.13)

```

<?php
    echo "<font color='green'>¡Todas las computadoras encendidas
exitosamente!</font>";
    exec("wakeonlan 00:1C:C0:91:77:5F");
    exec("wakeonlan 00:27:0E:0A:82:82");
?>

```

Off_all.php (Anexo 4.14)

```

<?php
    echo "<font color='red'>¡Todas las computadoras apagadas
exitosamente!</font>";
    shell_exec("net rpc shutdown -l 192.168.100.26 -U jeff%jefferson -f -
t 10 -C APAGANDO PC DESDE RASPBERRY PI");
    shell_exec("net rpc shutdown -l 192.168.100.27 -U jorge2%asdqw -f
-t 10 -C APAGANDO PC DESDE RASPBERRY PI");
?>

```

Ping.php (Anexo 4.15)

```
<?php
//DATOS PARA LA MATRIZ
$iplist = array //Matriz en la cual se llena los valores de dirección de red y nombre
de cada computador
(
    array("192.168.100.26","PC JEFFERSON"),
    array("192.168.100.27","PC JORGE"),
);
$i = count($iplist);
$results = [];
for ($j=0;$j<$i;$j++){
    $ip = $iplist[$j][0];
    $ping = exec("ping -c 1 $ip",$output,$status); //Se realizan pings a la dirección IP
para que refleje el estado de encendido o apagado en caso de que exista respuesta o
no
    $results[] = $status;
}
//MATRIZ DE MONITOREO - Se establece el tipo de letra además del estilo de la
tabla
echo '<font face=Courier New>';
echo "<table align=center border=1>
<th colspan=4> MONITOREO DE COMPUTADORAS </th>
<tr>
<td align=center width=20>#</td>
<td align=center width=100>IP</td>
<td align=center width=100>ESTADO</td>
<td align=center width=250>DESCRIPCION</td>
</tr>";
foreach ($results as $item => $k) {
    echo '<tr align=center>';
    echo '<td align=center>.$item.</td>';
    echo '<td align=center>.$iplist[$item][0].</td>';
    if ($results[$item]==0) {
        echo '<td align=center style=color:green>Encendido</td>';
    }
    else{
        echo '<td align=center style=color:red>Apagado</td>';
    }

    echo '<td>.$iplist[$item][1].</td>';
```

```

    echo '</tr>';
}
echo "</table>";
echo '</font>';
echo header("refresh: 3"); //La página web se actualiza cada 3 segundos para obtener
los valores de encendido o apagado de forma eficaz
?>

```

Styles.css (Anexo 4.16)

```

body{
    font-family: 'Lora', serif;
    background:url(power.png);
    background-size: 480px;
    background-position: 600px 70px;
    background-repeat:no-repeat;
    background-attachment:fixed;
}

.topwords{
    text-align: center;
    font-size: 35px;
}

p{
    font-size: 20px;
}

#main{
    width: 100%;
    height: 100px;
    padding: 25px;
}

input[type="submit"]{
    padding: 5px;
    color: #fff;
    background: #0098cb;
    width: 100px;
    margin: 20px auto;
    margin-top: 0;
    border: 0;
    border-radius: 10px;
}

```

```

    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #00b8eb;
}

```

PUERTA

Detección_facial.ino (Anexo 4.17)

```

//RECONOCIMIENTO FACIAL MEDIANTE ESP32 CAM

#include "esp_camera.h"
#include <WiFi.h>

//EN PRIMER LUGAR SE SELECCIONA LA CÁMARA, EN ESTE CASO SE ELEGI AI
//THINKER YA QUE ES LA QUE PERMITE RECONOCIMIENTO FACIAL
#define CAMERA_MODEL_WROVER_KIT
#define CAMERA_MODEL_ESP_EYE
#define CAMERA_MODEL_M5STACK_PSRAM
#define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER //RECONOCIMIENTO FACIAL
#define Rele 2 //DEFINICION DE VARIABLE PARA EL PIN 2
#include "camera_pins.h"

const char* ssid = "NETLIFE - FMLA MINALLA"; //SE ESTABLECE EL SSID DE LA
//RED A LA CUAL SE CONECTA EL ESP32
const char* password = "D@nger2o21!"; //SE ESCRIBE LA CONTRASEÑA
//PARA QUE SEA POSIBLE LA AUTENTICACIÓN

void startCameraServer(); //SE INICIA LA CONEXIÓN AL SERVIDOR DE LA
//CÁMARA

boolean recRostro = false; //SE INICIA CON EL RECONOCIMIENTO DESACTIVADO
boolean activarRele = false; //SE INICIA CON EL RELE APAGADO
long prevMillis = 0;
int interval = 5000;

void setup() {
    pinMode(Rele, OUTPUT); //SE SELECCIONA COMO SALIDA AL RELE
    digitalWrite(Rele, 0); //SE INICIA EN ESTADO BAJO AL RELE

    Serial.begin(115200); //SE ESTABLECE LA CANTIDAD DE BAUDIOS
    Serial.setDebugOutput(true);
    Serial.println(); //IMPRIME LOS MENSAJES EN EL MONITOR SERIAL

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
}

```

```

config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
//SE INICIA CON ESPECIFICACIONES ALTAS PARA PRE ASIGNAR BUFERES
MAS GRANDES
if (psramFound()) {
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

// INICIA LA CAMERA
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("El inicio de la cámara fallo, error: 0x%x", err); //EN CASO DE EXISTIR
ERRORES, LA CÁMARA LOS REFLEJA MEDIANTE EL MONITOR SERIAL
    return;
}

sensor_t * s = esp_camera_sensor_get();

if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); //SE CONFIGURA LA POSICIÓN VERTICAL DE LA CÁMARA,
ESTE PARAMETRO SE PUEDE MODIFICAR DE IGUAL MANERA EN LA INTERFAZ
    s->set_brightness(s, 1); //SE CONFIGURA EL BRILLO EN 1, ESTE PARAMETRO
SE PUEDE CONFIGURAR DE IGUAL MANERA POSTERIORMENTE EN LA
INTERFAZ
    s->set_saturation(s, -2); //SE CONFIGURA LA SATURACIÓN EN -2, ESTE
PARAMETRO SE PUEDE CONFIGURAR DE IGUAL MANERA POSTERIORMENTE
EN LA INTERFAZ
}
//SE ESTABLECE COMO INICIAL LA RESOLUCIÓN QVGA PARA UNA ÓPTIMA
CANTIDAD DE FRAMES
s->set_framesize(s, FRAMESIZE_QVGA);

```

```

#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password); //EL MODULO TOMA LAS CREDENCIALES PARA
  CONECTARSE A LA RED

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("RED INALÁMBRICA CONECTADA");

  startCameraServer();

  Serial.print("CÁMARA LISTA, USE LA IP 'http://'); //IMPRIME LA DIRECCIÓN IP
  LOCAL PARA CONECTARSE A LA INTERFAZ
  Serial.print(WiFi.localIP());
  Serial.println("PARA CONECTARSE");
}

void loop() {
  if (recRostro == true && activarRele == false) //SI EL ROSTRO CORRESPONDE Y
  EL RELE SE ENCUENTRA CERRADO
  {
    activarRele = true; //SE CAMBIA DE ESTADO A VERDADERO PARA ABRIR EL
    RELE
    digitalWrite(Rele, HIGH); //SE MANDA SEÑAL EN ALTO PARA QUE EL
    INTERRUPTOR SE ABRA
    prevMillis = millis();
  }
  if (activarRele == true && millis() - prevMillis > interval) //SI EL ROSTRO NO
  CORRESPONDE Y SE DETECTA INTRUSO
  {
    activarRele = false; //EL ESTADO SE MANTIENE EN FALSO Y NO SE ABRE EL
    RELE
    recRostro = false; //NO SE MANDA NINGUNA SEÑAL PARA ABRIR EL
    INTERRUPTOR
    digitalWrite(Rele, 0);
  }
}

```

PERSIANA

Persiana.ino Anexo (4.18)

```

#include <WiFi.h>                                     //LIBRERÍA WI-FI.

```

```

WiFiServer server(80); //INICIALIZACIÓN DEL SERVIDOR A
TARVÉS DEL PUERTO 80.

const char* ssid = "NETLIFE - FMLA MINALLA"; //NOMBRE DE LA RED WI-
FI.
const char* password = "D@nger2o21!"; //CONTRASEÑA DE LA RED WI-
FI.
String nuevo_dato; //VARIABLE PARA ALMACENAR LAS
SOLICITUDES DEL CLIENTE HTTP.

//-----CÓDIGO HTML-----
-----

String pagina = "<!DOCTYPE html>"
"<html>"
"<head>"
"<meta name='viewport' content='initial-scale=1.0'>"
"<meta charset='utf-8'>"
"<style>#map {height: 100%;}html, body {height: 100%;margin: 0;padding: 0;} </style>"
"</head>"
"<body>"
"<h1>CONTROL REMOTO DE PERSIANA</h1>"

"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonBlueLong.png'o
nmousedown=location.href='/dec1' >"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/nixiesmall1.png'>"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonOrangeLong.pn
g'onmousedown=location.href='/inc1' >"

"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonPinkLong.png'o
nmousedown=location.href='/dec2' >"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/nixiesmall2.png'>"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonGreenLong.png'
onmousedown=location.href='/inc2' >"

"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonTurqLong.png'o
nmousedown=location.href='/dec3' >"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/nixiesmall3.png'>"
"<input type=image style=width:33%;height:20%
src='http://myrobotlab.org/sites/default/files/users/user25images/ButtonPurpleLong.png'
onmousedown=location.href='/inc3' >"

"</body>"
"</html>";

//-----

```

```

void setup(){
  Serial.begin(115200);          //VELOCIDAD DE COMUNICACIÓN (BAUDIOS).
  pinMode(16, OUTPUT);          //GPIO 16 - SALIDA ; PIN PARA ESTABLECER
  SENTIDO DE GIRO DE LOS MOTORES PASO A PASO.
  pinMode(17, OUTPUT);          //GPIO 17 - SALIDA ; CONTROL DEL NÚMERO
  DE PASOS DEL MOTOR DE LA PERSIANA 1.
  pinMode(18, OUTPUT);          //GPIO 18 - SALIDA ; CONTROL DEL NÚMERO
  DE PASOS DEL MOTOR DE LA PERSIANA 2.
  pinMode(19, OUTPUT);          //GPIO 19 - SALIDA ; CONTROL DEL NÚMERO
  DE PASOS DEL MOTOR DE LA PERSIANA 3.

  //-----INICIALIZACIÓN DE CONEXIÓN Wi-Fi-----
  -----

  Serial.println();
  Serial.print("CONECTANDO A: ");
  Serial.println(ssid);          //SE MUESTRA EL NOMBRE DE LA RED A LA CUAL
  SE ESTA CONECTANDO EL ESP32.
  WiFi.begin(ssid, password);    //INICIO DE COMUNICACIÓN CON RED WI-FI
  LOCAL.

  while (WiFi.status() != WL_CONNECTED){ //BUCLE PARA COMPRIBAR SI EXISTE
  O NO CONECCIÓN CON LA RED WI-FI LOCAL.
    delay(5000);                  //RETARDO DE 5 [S] HASTA LOGRAR
  CONECTARSE A LA RED.
    Serial.println("ESPERANDO CONECTAR A LA RED WI-FI..."); //MENSAJE DE
  ESPERA HASTA LOGRAR LA CONECCIÓN.
  }

  IPAddress ip(192,168,100,31);    //CONFIGURACIÓN DE LA DIRECCIÓN IP
  ESTÁTICA DEL SERVIDOR WEB.
  IPAddress gateway(192,168,100,1);
  IPAddress subnet(255,255,255,0);
  WiFi.config(ip, gateway, subnet);

  Serial.println("");            //EN CASO DE EXISTIR CONECCIÓN CON LA RED
  LOCAL.
  Serial.println("RED WI-FI CONECTADA");
  Serial.println("DIRECCIÓN IP: ");
  Serial.println(WiFi.localIP()); //MUETSRA LA DIRECCIÓN IP DE LA RED
  LOCAL.

  server.begin();                //INICIO DEL SERVIDOR.
}

void loop(){
  WiFiClient client = server.available(); //EL ESP32 EMPIEZA A ESCUCHAR A
  LOS CLIENTES (NAVEGADOR WEB).

  if (client){                   //EN CASO DE EXISTIR UN NUEVO CLIENTE.
    Serial.println("NUEVO CLIENTE");
    String currentLine = "";

```

```

while (client.connected()){           //MIENTRAS EL CLIENTE SE ENCUENTRE
CONECTADO REALIZA EL BUCLE.
    if (client.available()){           //SI EXISTEN BYTES ENVIADOS POR EL
CLIENTE.
        char c = client.read();       //LEE EL BYTE ENVIADO.
        Serial.write(c);              //MUESTRA EL DATO A TRAVÉS DEL MONITOR
SERIAL.
        nuevo_dato += c;              //SE ALMACENA LA CABECERA DEL HTTP
REQUEST.
        if (c == '\n'){               //CADA QUE EL DATO ENVIADO SEA UN
CARACTER "SALTO DE LÍNEA"
            if (currentLine.length() == 0){ //SE TERMINA EL HTTP REQUEST Y SE
RESPONDE CON LAS SIGUIENTES LÍNEAS...
                client.println("HTTP/1.1 200 OK");
                client.println("Content-type:text/html");
                client.println("Connection: close");
                client.println();

//-----LÓGICA PARA EL CONTROL DE LOS MOTORES PASO A
PASO-----

                if (currentLine.endsWith("GET /right1")) {           //PETICIÓN PARA
CONTROL DEL MOTOR PASO A PASO (1).
                    digitalWrite(16, HIGH);                         //PULSO ALTO - ROTACIÓN
HACIA LA DERECHA DEL MOTOR PASO A PASO.
                    for (int i=0; i <= 3600; i++){
                        digitalWrite(17, HIGH);
                        delay(10);
                        digitalWrite(17,LOW);
                        delay(10);
                    }
                }
                if (currentLine.endsWith("GET /left1")) {           //PETICIÓN PARA CONTROL
DEL MOTOR PASO A PASO (1).
                    digitalWrite(16, LOW);                           //PULSO BAJO - ROTACIÓN
HACIA LA IZQUIERDA DEL MOTOR PASO A PASO.
                    for (int i=0; i <= 3600; i++){
                        digitalWrite(17, HIGH);
                        delay(10);
                        digitalWrite(17,LOW);
                        delay(10);
                    }
                }
                if (currentLine.endsWith("GET /right2")) {           //PETICIÓN PARA
CONTROL DEL MOTOR PASO A PASO (2).
                    digitalWrite(16, HIGH);                         //PULSO ALTO - ROTACIÓN
HACIA LA DERECHA DEL MOTOR PASO A PASO.
                    for (int i=0; i <= 3600; i++){
                        digitalWrite(18, HIGH);
                        delay(10);
                        digitalWrite(18,LOW);
                        delay(10);
                    }
                }
            }
        }
    }
}

```



```

nuevo_dato = "";
VARIABLE nuevo_dato.
client.stop();
Serial.println("CLIENTE DESCONECTADO");
Serial.println("");
}
}

```

//SE PROCEDE A LIMPIAR LA

//SE TERMINA LA CONEXIÓN

DIAGRAMAS DE FLUJO

Diagrama de interfaz Anexo (4.19)

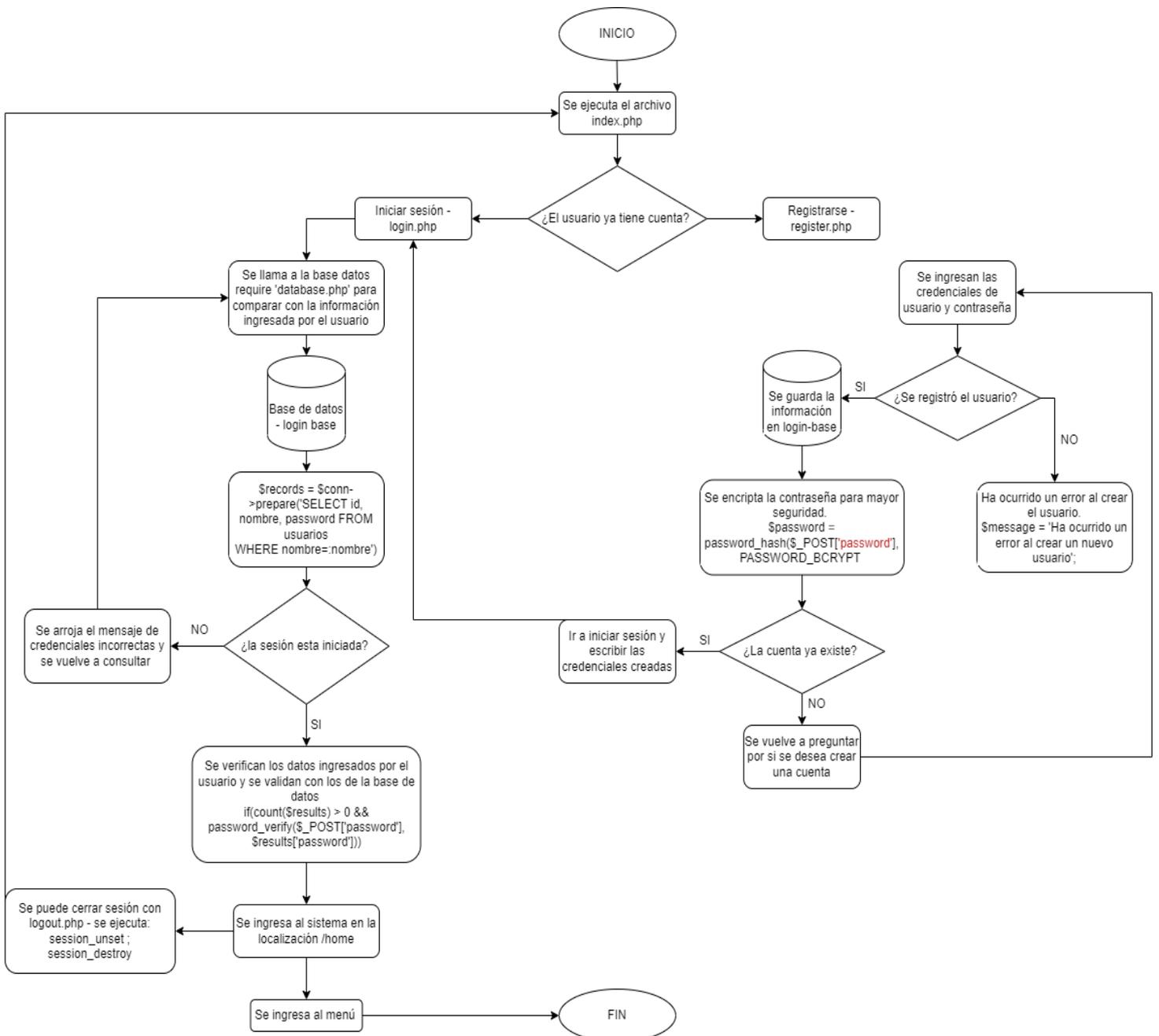


Diagrama del menú Anexo (4.20)

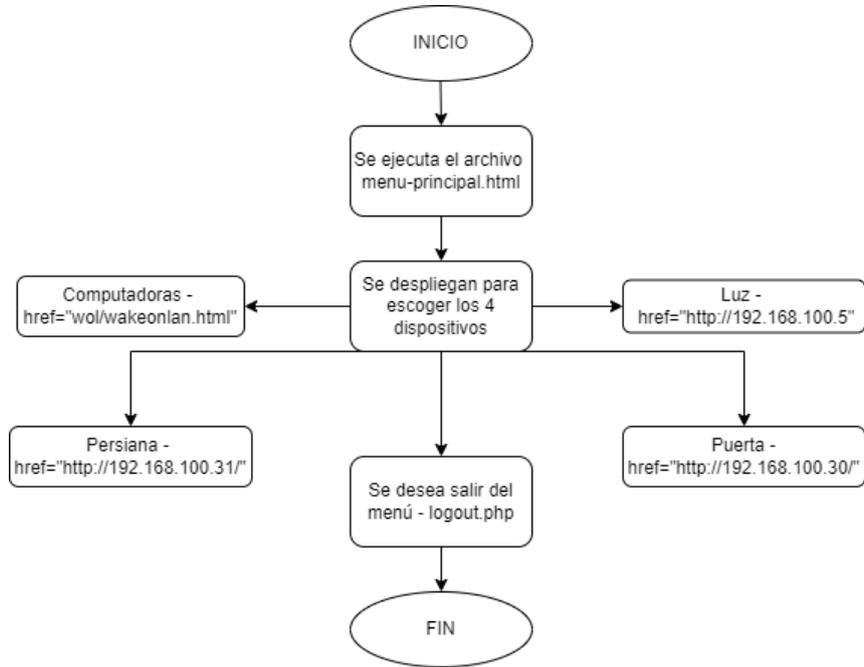


Diagrama de computadoras Anexo (4.21)

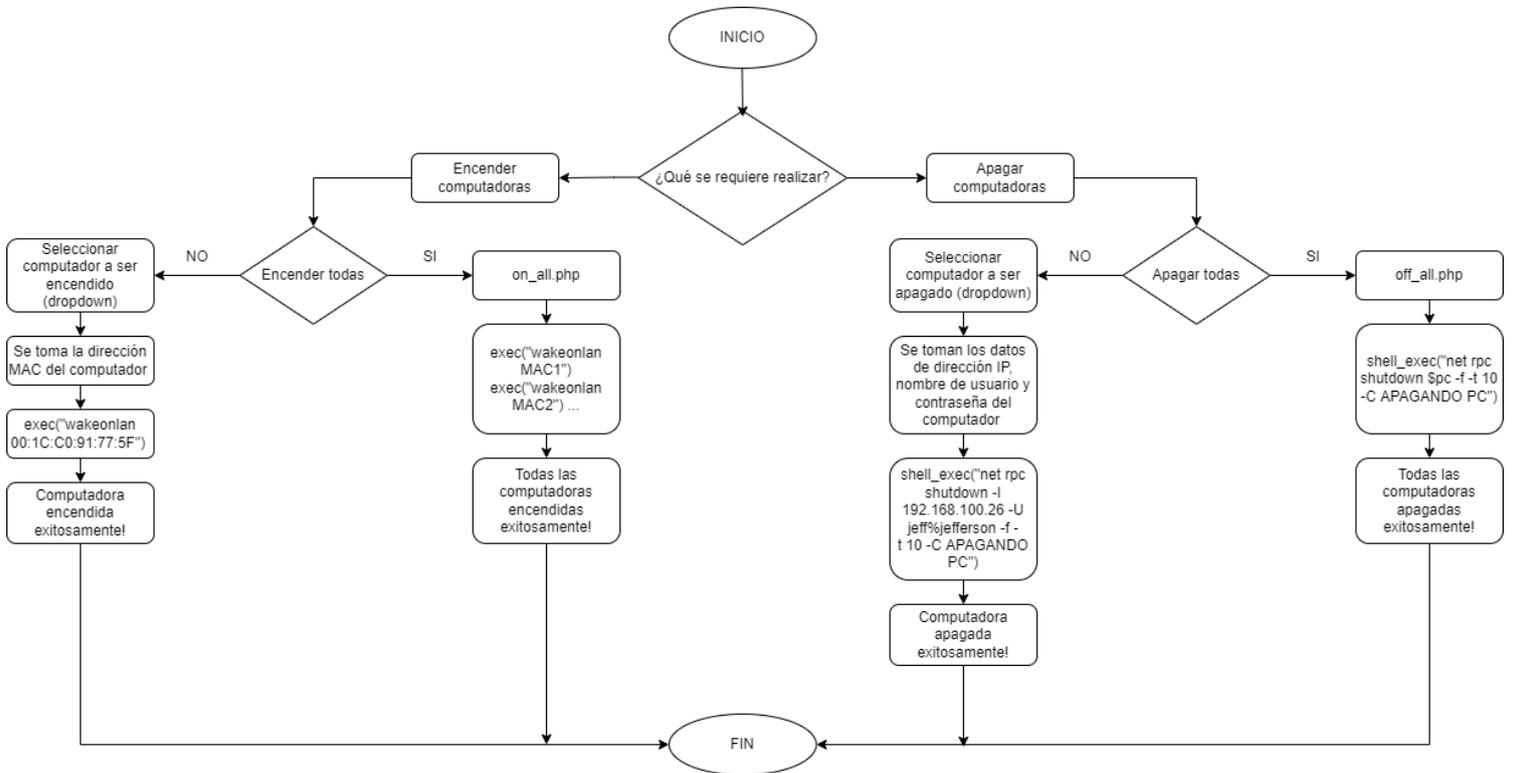


Diagrama de monitoreo de computadores Anexo (4.22)

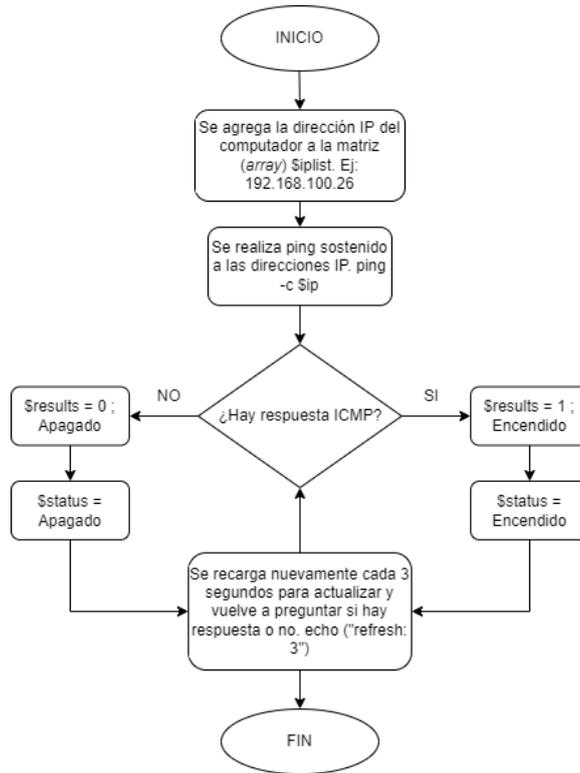


Diagrama de persianas Anexo (4.23)

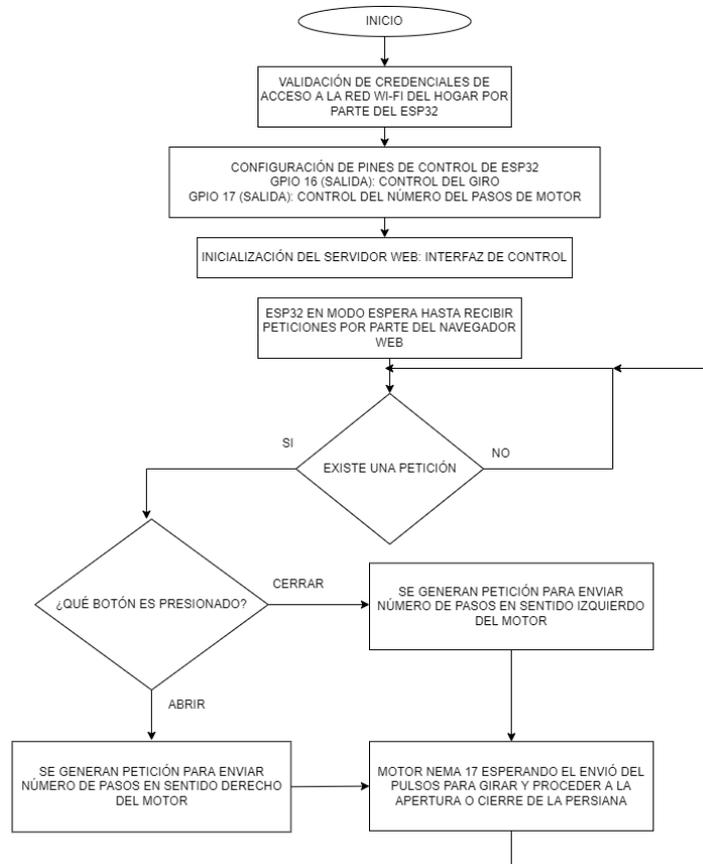


Diagrama de luz Anexo (4.24)

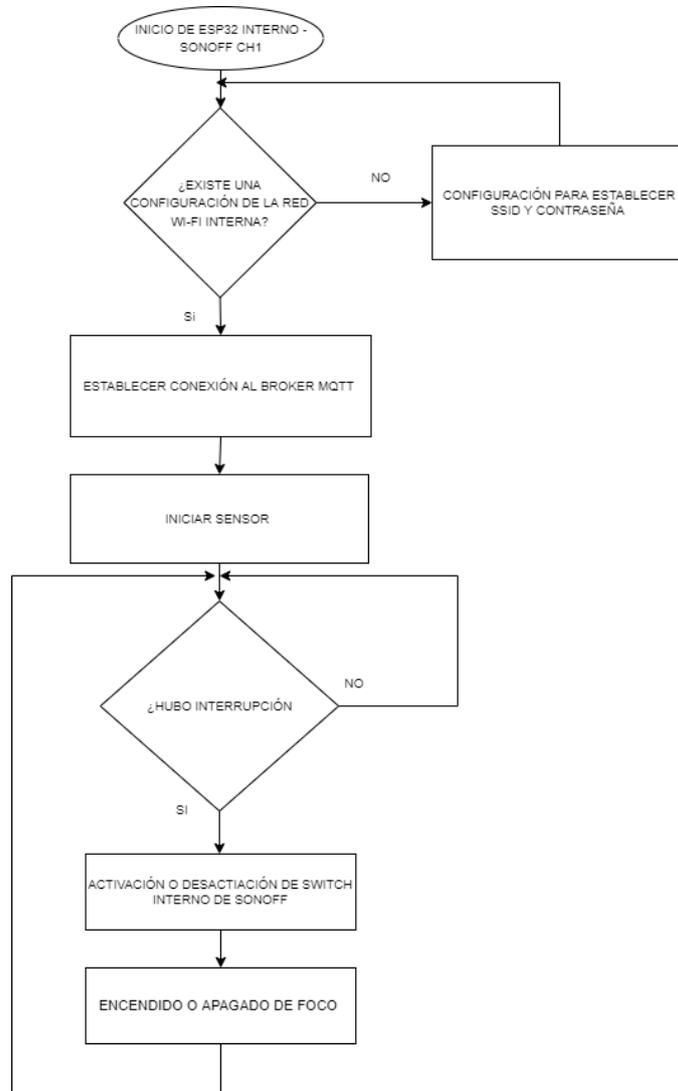


Diagrama de puerta Anexo (4.25)

