

**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**TESIS**

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DOMÓTICA  
DE BAJO COSTO PARA EL CONTROL REMOTO DE HOGARES  
BASADA EN IOT**

PARA OBTENER EL TÍTULO PROFESIONAL DE:

**INGENIERO ELECTRÓNICO**

**ELABORADO POR:**

DANIEL AUGUSTO GUEVARA AGÜERO

**ASESOR**

MSC. ING. JOSÉ AMBROSIO MACHUCA MINES

**LIMA – PERÚ**

**2020**



**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DOMÓTICA  
DE BAJO COSTO PARA EL CONTROL REMOTO DE HOGARES  
BASADA EN IOT**

**Dedicatoria:**

Este trabajo lo dedico con mucho amor a mis padres y mi hermana. También a todas las personas que me apoyaron y se alegraron cuando logré cumplir mis metas.

**Agradecimiento:**

En primer lugar agradezco a Dios, luego a mis padres (Ligia y Augusto), a mi hermana Diana y mi familia completa, que me alentaron y nunca dejaron de confiar en mí. También a mis amigos, por brindarme su apoyo y ánimos en la realización de esta tesis. Finalmente, un agradecimiento especial a la UNI, porque ahí aprendí a sobreponerme a las adversidades.

## RESUMEN

En la presente tesis se detalla el diseño e implementación de una plataforma domótica basada en IoT la cual permite controlar diversos artefactos comúnmente presentes en un hogar. Dicho control se podrá ejercer desde locaciones remotas empleando dispositivos electrónicos finales que cuenten con acceso a internet, tales como celulares, tablets, computadoras, etc.

El sistema ha sido diseñado priorizando el bajo costo y la baja complejidad de la interfaz web. Es en base a ello que, tras un adecuado análisis, se han escogido los elementos de hardware más económicos y eficaces compatibles con la realidad del mercado peruano. A su vez, se ha construido una interfaz intuitiva que no requiere de mayores conocimientos por parte del usuario para su manipulación.

Se ha evaluado el desempeño del sistema en una vivienda habitada por una familia de 4 personas con la finalidad de detectar errores y efectuar las acciones correctivas correspondientes. De esta manera, se busca conseguir un sistema estable y duradero en condiciones de trabajo reales.

## **ABSTRACT**

This paper details the design and implementation of a home automation platform based on IoT which allows controlling various devices commonly present at home. This control can be exercised from remote locations using electronic devices that have internet access, such as cellphones, tablets, computers, etc.

This system has been designed prioritizing the low cost and the low complexity of the web interface. It is on this basis that, after an adequate analysis, the most economical and efficient hardware elements compatible with the reality of the Peruvian market have been chosen. At the same time, it has been built an intuitive interface that doesn't require greater knowledge from the user to manipulate it.

The performance of the system has been evaluated in a house inhabited by a family of 4 in order to detect errors and carry out the corresponding corrective actions. In this way, the aim is to achieve a stable and durable system in real working conditions.

## ÍNDICE

<b>PRÓLOGO</b> .....	<b>5</b>
<b>CAPÍTULO I</b>	
<b>INTRODUCCIÓN</b> .....	<b>7</b>
1.1. Generalidades .....	7
1.2. Descripción de la Problemática .....	8
1.2.1. Formulación del Problema .....	9
1.3. Objetivos .....	9
1.3.1. Objetivo General .....	9
1.3.2. Objetivos Específicos .....	9
1.4. Hipótesis .....	10
1.4.1. Hipótesis Principal .....	10
1.4.2. Hipótesis Secundarias .....	10
<b>CAPÍTULO II</b>	
<b>FUNDAMENTO TEÓRICO</b> .....	<b>11</b>
2.1. Introducción .....	11
2.2. Domótica .....	11
2.3. IoT ( <i>Internet of Things</i> ) .....	12
2.4. Sistemas Embebidos (ES) .....	13
2.4.1. Arduino .....	14
2.4.2. Raspberry Pi .....	17
2.5. Relés o Relays .....	20
2.5.1. Módulo Relé 5VDC .....	21
2.6. Motores DC .....	22
2.6.1. Puente H con transistores .....	24
2.6.2. Circuito Integrado L293D .....	25
2.7. Módulos de comunicación inalámbrica – Módulo Wi-Fi ESP-01 .....	27
2.7.1. ESP8266EX .....	29
2.7.2. Actualizando el firmware del ESP-01 .....	30
2.8. Servidores Web .....	33



2.9. Base de Datos .....	34
<b>CAPÍTULO III</b>	
<b>DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>35</b>
3.1. Introducción .....	35
3.2. Arquitectura del sistema .....	35
3.3. Desarrollo de la interfaz web y la aplicación móvil .....	36
3.4. Configuración de los servicios en internet .....	39
3.5. Programación del controlador central (Raspberry Pi) .....	41
3.6. Diseño y programación de los microcontroladores y actuadores .....	44
3.6.1. Conexión entre el Arduino NANO y el módulo Wi-Fi ESP-01 .....	44
3.6.2. Diseño de circuito de control de las luminarias .....	45
3.6.3. Diseño de circuito de control de las persianas .....	48
3.6.4. Diseño de circuito de control de la terma eléctrica .....	49
<b>CAPÍTULO IV</b>	
<b>EVALUACIÓN Y ANÁLISIS DE RESULTADOS .....</b>	<b>51</b>
4.1. Introducción .....	51
4.2. Presentación ilustrativa de los resultados .....	51
4.3. Análisis de resultados .....	58
4.3.1. Complejidad en la implementación .....	58
4.3.2. Facilidad de uso de la plataforma .....	58
4.3.3. Bajo costo del sistema .....	61
<b>CONCLUSIONES .....</b>	<b>64</b>
<b>RECOMENDACIONES .....</b>	<b>65</b>
<b>ANEXOS .....</b>	<b>66</b>
<b>ANEXO A GLOSARIO DE TÉRMINOS E ÍNDICE DE FIGURAS Y TABLAS .....</b>	<b>67</b>
<b>ANEXO B DESCRIPCIÓN DEL PINADO DEL ARDUINO NANO .....</b>	<b>72</b>
<b>ANEXO C SCRIPTS EN HTML, PHP Y CSS PARA LA INTERFAZ WEB .....</b>	<b>73</b>
<b>ANEXO D SCRIPT EN PYTHON PARA EL CONTROLADOR CENTRAL .....</b>	<b>83</b>
<b>ANEXO E SCRIPT EN ARDUINO PARA EL CONTROL DE LUMINARIAS .....</b>	<b>89</b>
<b>ANEXO F SCRIPT EN ARDUINO PARA EL CONTROL DE PERSIANAS .....</b>	<b>91</b>
<b>ANEXO G SCRIPT EN ARDUINO PARA EL CONTROL DE LA TERMA .....</b>	<b>93</b>
<b>ANEXO H FORMATO DE ENCUESTA SOBRE LA INTERFAZ WEB .....</b>	<b>95</b>
<b>BIBLIOGRAFÍA .....</b>	<b>97</b>

## PRÓLOGO

En la actualidad, la gran mayoría de personas tiene en su vivienda un punto de acceso a internet; su influencia en los hogares es tal, que casi se ha vuelto una necesidad para realizar actividades cotidianas como comprar, vender, estudiar, trabajar, comunicarse, entretenerse, etc. Por tal motivo, últimamente se han intensificado los esfuerzos en automatizar todo tipo de trabajo, dentro de los cuales se encuentran los quehaceres del hogar, y controlarlo mediante dispositivos electrónicos a través de la red de internet. Este concepto de automatización de viviendas ha adoptado el nombre de domótica.

El propósito del presente trabajo consiste en implementar un sistema domótico que controle tres tipos de dispositivos domésticos usualmente presentes en una vivienda (luminarias, persianas y terma), ubicadas en tres habitaciones diferentes (sala, dormitorio y baño). Asimismo, se demuestra que la adquisición de dicho sistema, no representa un gasto económico fuerte para una familia con ingresos dentro del promedio; del mismo modo, la manipulación del sistema, no requiere de conocimientos especializados para un buen funcionamiento. La única limitación de este sistema es que el usuario no garantice una conexión estable a internet, lo que ocasionaría una mala comunicación y a su vez, un mal funcionamiento de los dispositivos.

Para desarrollar exitosamente este proyecto de tesis, se realizó un compendio con la información detallada extraída de la página web de los fabricantes de los dispositivos electrónicos que componen el sistema domótico. Se estudiaron a fondo los lenguajes de marcado de hipertexto y de programación, requeridos en el diseño de la interfaz web y en la comunicación con la base de datos. Además, se pusieron a prueba los códigos de los programas para un correcto funcionamiento de los dispositivos a controlar. Finalmente, para validar los aspectos subjetivos expuestos en las hipótesis del presente trabajo, se ha realizado una encuesta virtual a un grupo de personas que pudieron interactuar con la plataforma web y aprobaron el funcionamiento de la misma.

El presente trabajo consta de cuatro capítulos y siete anexos que buscan esclarecer cualquier duda en cuanto al diseño e implementación del sistema domótico. En el capítulo I se explica la problemática que se encontró y el camino recorrido hasta que surgió el

propósito de este proyecto de tesis. Asimismo, se detallan los objetivos a lograr e hipótesis que se demuestran a lo largo de este trabajo. En el capítulo II se brindan todos los conocimientos necesarios en los que se basa el diseño e implementación del sistema domótico completo (interfaz web, controladores, actuadores, servicios en la nube, etc.). En el capítulo III se detallan los pasos seguidos para diseñar e implementar una vivienda “inteligente” (es decir, automatizada). Finalmente, en el capítulo IV se muestran gráficamente los resultados obtenidos y el hardware del sistema domótico que se instaló en la vivienda para su automatización.

La empresa peruana PC Ingenieros E.I.R.L. ha facilitado mucho la realización del presente trabajo, brindando principalmente: el uso de sus servicios contratados en la nube (dominio, hosting y base de datos) y el financiamiento para la compra de los componentes electrónicos utilizados. Debido a ello, el autor expresa un especial agradecimiento.

## **CAPÍTULO I**

### **INTRODUCCIÓN**

#### **1.1. Generalidades**

El concepto de domótica ha existido por muchos años en el mundo, pero hace muy poco que ha tomado protagonismo entre las personas que buscan invertir un poco de dinero en remodelar su casa y añadirle ciertas comodidades, o en aquellos que buscan construir una casa automatizada desde cero. Y por mencionar algunas de las razones por las que no tuvo mucha acogida en el mercado, se encuentran las siguientes: requieren de una instalación muy complicada y de un ambiente prediseñado para albergar los equipos necesarios; asimismo, los precios de los equipos a utilizar son demasiado elevados, así como también lo es el costo de instalación y el mantenimiento de los mismos. [1]

En el Perú ha sido considerado un lujo (el automatizar las viviendas) desde el principio, ya que es una tecnología moderna (pero no complicada) que se encarga de la automatización de una vivienda. Adicionalmente, las empresas que se dedican a ello ([2], [3], [4], [5] y [6]) poseen tarifas que no son alcanzables más que para un público con ingresos que exceden al promedio en demasía. [1] Es debido a ello que el concepto de domótica en el Perú no es muy conocido y la tecnología menos empleada; sin embargo, después de incorporarse la tecnología IoT en el mercado, el concepto de domótica volvió a aparecer con mayor acogida, ya que los servicios ofrecían nuevos beneficios a los usuarios y los precios se reducían un poco, siendo aún inalcanzables para la mayoría de la población.

Actualmente, en el país existen algunos edificios que presentan la tecnología domótica en sus instalaciones, tales como: el hotel Westin, la nueva sede del Banco de la Nación, la Biblioteca Nacional, la clínica Delgado, la Torre de Interbank, etc. [7]

En mayo del 2019, los analistas afirmaban que la adopción de la tecnología IoT tardaría entre tres y cinco años en el sector empresarial, tras los cuales se adaptaría el uso de las tecnologías domótica e IoT y aumentaría el porcentaje de empresas (22% en el 2019) que se veían beneficiadas por ella (cabe mencionar que han pasado aproximadamente tres años desde que dichas tecnologías llegaron al Perú). [8]

## 1.2. Descripción de la Problemática

En el Perú, existe un porcentaje mínimo de viviendas que se encuentran automatizadas. El gran porcentaje que no lo está y no goza de los beneficios de esta tecnología, tiene como sus principales razones: la falta de conocimiento acerca del tema, los precios inalcanzables de los equipos, instalaciones y mantenimientos, y la complicada interacción que se da entre el usuario y la interfaz que controla el sistema domótico.

La principal causa del bajo consumo de viviendas “*smart*” en el territorio nacional, es la falta de conocimiento acerca de la tecnología y eso se debe principalmente a la poca difusión del concepto de domótica entre los mismos peruanos. Un número reducido de personas conocen sobre el tema y a su vez, no lo consideran una necesidad ni una inversión justificada, motivo por el cual, no encuentran objeto en compartir la información con sus allegados.

Como segunda causa determinante, se tiene que las tarifas proporcionadas por las empresas que se encuentran en el rubro, son exorbitantes. Por tomar un ejemplo publicado en el 2017, en la página web del diario oficial El Peruano: la empresa Smart House Perú ofrecía (en territorio nacional) una automatización básica en un departamento por un precio que oscilaba entre los 1,500 y 4,000 dólares americanos (USD). Asimismo, se indicó que la automatización de algunos ambientes de una vivienda podría costar entre los 6,000 y 8,000 soles (PEN), mientras que la automatización de todo el hogar podría llegar a costar más de 10,000 dólares americanos (USD). [1] Cabe mencionar que la Remuneración Mínima Vital (RMV) durante el año 2017 fue de S/.850.00 [9], lo que al cambio de ese entonces valdría un poco más de \$260 (USD). [10] Es por este motivo, que la domótica fue considerada un lujo entre los potenciales clientes.

La tercera y última causa mencionada en este trabajo, refiere a la complejidad presentada por las interfaces ofrecidas a los clientes, las cuales requieren de la lectura de un manual o de una capacitación previa para el correcto uso y aprovechamiento del sistema.

Ante esta realidad, surge la necesidad del desarrollo de una plataforma domótica de bajo costo, que se encuentre destinada al público de todos los sectores, ya sean personas naturales o jurídicas de clase económica baja o media. Asimismo, se necesita que dicha plataforma no presente dificultades en su instalación, mantenimiento y utilización, siendo estos dos últimos posibles de realizar por un usuario sin conocimientos especializados y de manera intuitiva.

### 1.2.1. Formulación del Problema

#### a) Problema General

El problema general consiste en conocer:

- ¿En qué medida el diseño e implementación de una plataforma domótica de bajo costo basada en IoT permite el control remoto de hogares?

#### b) Problemas Específicos

Los problemas específicos consisten en conocer:

- ¿Cómo se puede diseñar una interfaz web y una topología de red adecuada para el sistema propuesto en el presente trabajo?
- ¿Qué lenguajes de programación resultan útiles y óptimos para el desarrollo del sistema?
- ¿Qué se requiere para una correcta implementación del sistema?
- ¿Qué método de evaluación sería recomendable para encontrar posibles fallas en el sistema?

### 1.3. Objetivos

Los objetivos de este trabajo se dividen en dos grupos, los cuales se detallan a continuación.

#### 1.3.1. Objetivo General

El objetivo general de este trabajo consiste en:

- Diseñar e implementar una plataforma domótica de bajo costo basada en IoT para el control remoto de hogares.

#### 1.3.2. Objetivos Específicos

Los objetivos específicos de este trabajo consisten en:

- Diseñar una interfaz web intuitiva con íconos universales y de fácil acceso con lenguaje de marcado HTML y CSS, así como una topología de red con elementos de hardware confiables, duraderos y de bajo costo.
- Programar la interfaz web con lenguaje de programación PHP, JavaScript y MySQL; del mismo modo la topología de red con Python, C++ y comandos AT para el correcto funcionamiento del sistema.
- Implementar la interfaz web y la topología de red en un sistema único con los medios y protocolos de comunicación apropiados para una comunicación estable.
- Evaluar el sistema instalado en un ambiente real de trabajo durante un periodo de prueba adecuado y subsanar cualquier posible falla encontrada.

## **1.4. Hipótesis**

Al igual que los objetivos, las hipótesis de este trabajo también se dividen en dos grupos los cuales se detallan a continuación.

### **1.4.1. Hipótesis Principal**

Con el diseño e implementación de una plataforma domótica de bajo costo basada en IoT es posible realizar el control remoto de hogares.

### **1.4.2. Hipótesis Secundarias**

Las hipótesis específicas de este trabajo consisten en:

- Es posible diseñar una interfaz web intuitiva y de fácil acceso con lenguaje de marcado HTML y CSS, así como una topología de red con elementos de hardware confiables, duraderos y de bajo costo.
- Es factible programar la interfaz web con lenguaje de programación PHP, JavaScript y MySQL; del mismo modo la topología de red con Python, C++ y comandos AT.
- Es posible implementar la interfaz web y la topología de red en un sistema único con los medios y protocolos de comunicación apropiados para mantener una comunicación estable.
- Es factible evaluar el sistema domótico instalado en un ambiente real de trabajo durante un periodo de prueba y subsanar cualquier posible falla que se presente.

## CAPÍTULO II

### FUNDAMENTO TEÓRICO

#### 2.1. Introducción

En el segundo capítulo se definirán al detalle todos los conceptos necesarios para tener una base sólida que permitan entender en qué se basan el diseño e implementación del presente trabajo, tanto a nivel de hardware como de software.

#### 2.2. Domótica

La domótica engloba un conjunto de tecnologías destinadas al control y automatización de una vivienda, que le permite al usuario medir de una manera más eficiente su consumo de energía, además de aumentar la seguridad y comodidad de los residentes. Algunas de las funciones que cumplen los sistemas domóticos se aprecian en la Figura 2.1. Actualmente, el rápido avance tecnológico ha permitido que los servicios brindados sean de mejor calidad y menor costo. Asimismo, su utilización es más intuitiva, y tanto la instalación como la programación del sistema son altamente versátiles. [11]

A continuación se procederá a explicar de forma escueta los beneficios recientemente mencionados:

- **Ahorro de energía.**- La automatización de los equipos pretende eliminar casi en su totalidad los consumos de energía no deseados e inútiles, que son generados debido a la necesidad de apagar los equipos manualmente.
- **Accesibilidad.**- El usuario se podrá comunicar con los equipos mediante su celular y/o computadora, lo que permitirá que los adultos mayores y/o personas con discapacidad puedan acceder y controlar todos los dispositivos del hogar.
- **Seguridad.**- En vista que el usuario tendrá acceso a los dispositivos de la vivienda, ya sean cámaras y/o cerrojos, éste va a poder velar por la seguridad de su hogar o, en su defecto, programar algún sistema de alerta de intruso con la misma finalidad.
- **Comodidad o “Confort”.**- La facilidad que brinda el hecho de poder controlar los dispositivos de la vivienda desde su celular, aumenta la comodidad y genera bienestar en el usuario.



La factibilidad de estos beneficios, se basa en un estudio local del ambiente en el que se desea instalar el sistema domótico, a fin de escoger acertadamente el hardware y software a utilizar. Es por ello que se requieren estándares, los cuales brindan las normas y recomendaciones para garantizar la correcta funcionabilidad y la mayor eficiencia de los sistemas domóticos. Dentro de los estándares internacionales más conocidos y usados, se encuentran: el KNX [12], el LonWorks [13], X-10, BACnet, etc. [14]



**Figura 2.1** Imagen ilustrativa del concepto de Domótica  
(Fuente: Casas Digitales [15])

### 2.3. IoT (*Internet of Things*)

El “Internet de las Cosas” (traducido al español) se puede empezar a definir partiendo del concepto de Internet (la red de redes), que es la encargada de interconectar todos los dispositivos electrónicos a nivel mundial para el intercambio de información entre ellos. Luego, ampliamos la definición a una red de “cosas” (por usar el término del nombre) que no necesariamente se encuentren manipulados por un usuario en específico, sino que cuenten con acceso a internet y se mantengan a la espera de una orden a ejecutar. Los avances tecnológicos actuales han llegado tan lejos que la nueva meta es que todos los dispositivos se encuentren organizados y controlados por una computadora programable, siendo esta última, la encargada de verificar que todas las acciones se lleven a cabo, dejando a un lado, la intervención humana en el proceso. A esta interacción se le conoce como la interacción máquina a máquina o M2M (*Machine to Machine*). [16]

Dentro de la lista de “cosas” que formarán parte de esta red se encuentran los electrodomésticos más comunes como el refrigerador, el horno de microondas, la licuadora, la olla arrocera, la terma, etc. (Ver Figura 2.2) Asimismo, cabe mencionar la



prioridad la optimización energética y la del uso de la memoria, además de emplear los mejores sistemas de comunicación internos para garantizar una comunicación sin retardos ni interrupciones. [24]

Las microcomputadoras, que controlan el funcionamiento de los electrodomésticos, normalmente emplean sistemas embebidos para ello; mientras que las computadoras personales (PC, *Personal Computer*) los emplean para el control de los periféricos y algunas funciones menores.

Debido a que son diseñadas para trabajos específicos, los ES comúnmente presentan ciertas ventajas y beneficios que se mencionarán a continuación: [23]

- **Dimensiones compactas.**- Al trabajar con diseños innovadores y personalizados, los ES no requieren de ventiladores para su enfriamiento, ya que sólo se basan en principios termodinámicos, lo que reduce en gran magnitud el espacio innecesario y las dimensiones de los mismos.
- **Alta Confiabilidad.**- Los ES de uso industrial son diseñados para trabajar 24/7 a un ritmo exigente y a condiciones extremas, por lo que se garantiza su buen desempeño en la mayoría de situaciones.
- **Optimización energética.**- Ya que están diseñadas para trabajar sin descanso, los ES vienen con un firmware ligero que no consume mucha potencia en su procesamiento. Adicionalmente, el ahorro energético de su sistema de enfriamiento libre de ventiladores, da como resultado una gran eficiencia en cuanto al consumo de energía.

Hoy en día son muchas firmas las que trabajan en el diseño e implementación de los ES, siendo 2 de las más reconocidas: el Arduino y el Raspberry Pi, las cuales se describirán a continuación:

#### **2.4.1. Arduino**

Acorde a su página oficial, Arduino se define como “una plataforma electrónica abierta, apta para todo público, que está basada en hardware y software muy fáciles de utilizar.” [25] Es por ello que se obtiene una placa con un microcontrolador programable (removible en algunos casos y embebido en otros) que presenta varias entradas, entre analógicas y digitales, así como salidas digitales, fuentes internas para alimentar otros circuitos integrados que son adaptables y compatibles, entre otros beneficios. Para poder programar el microcontrolador Arduino (hardware) se utiliza el lenguaje de programación Arduino (basado en el lenguaje C) que se procesa en el software Arduino (IDE) siendo este último, un programa libre (gratuito) y descargable de su página oficial. Se ha hecho muy popular y

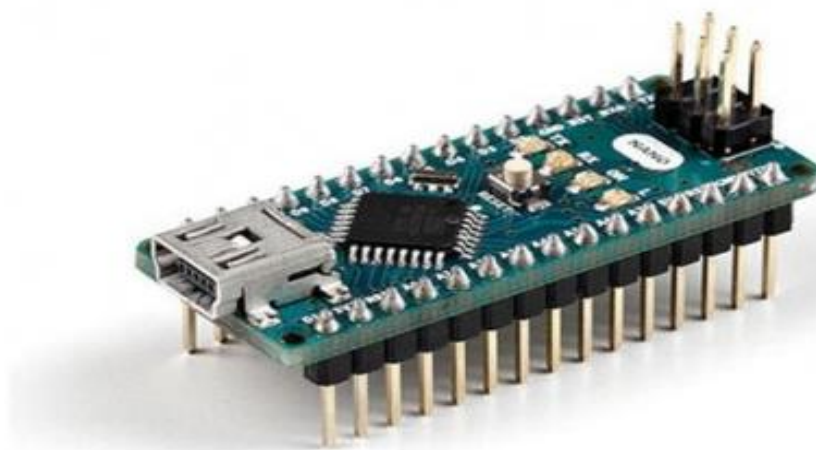
querido por la accesibilidad y adaptabilidad que presenta en ámbitos profesionales, académicos, de investigación, etc., además de poseer hardware de bajo costo y software ampliables y de código abierto, contar con miles de librerías libres y una comunidad global de usuarios abierta activa. [26]

La empresa Arduino pone a disposición múltiples versiones de la placa, cada una con características diferentes y destinadas a proyectos y trabajos diferentes. Por mencionar algunos de los modelos más usados, se tienen a los Arduinos UNO, MEGA y NANO. Siendo este último (ver Figura 2.3) el que se va a utilizar para el desarrollo de la presente tesis, por ello se procederá a mostrar algunas de sus especificaciones en la Tabla 2.1.

**Tabla 2.1** Especificaciones Técnicas del Arduino NANO

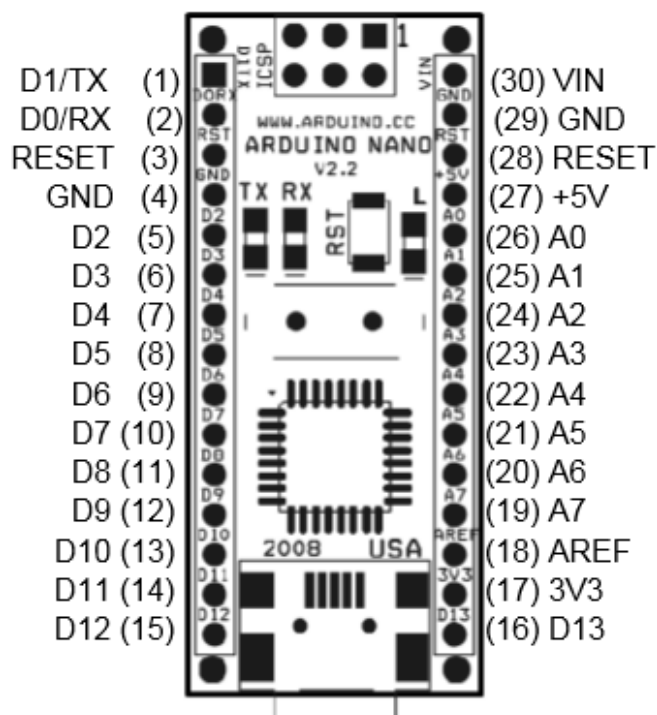
Microcontrolador	ATmega328
Arquitectura	AVR
Voltaje de Operación	5 V
Memoria Flash	32 KB (2 KB para el bootloader)
SRAM	2 KB
Velocidad del Reloj	16 MHz
Pines de Entrada Analógica	8
EEPROM	1 KB
Corriente DC por Pines I/O	40 mA
Voltaje de Entrada	7-12 V
Pines Digitales I/O	22 (6 son PWM)
Salida PWM	6
Consumo de Potencia	19 mA
Tamaño del PCB	18 x 45 mm
Peso	7 g
Código del Producto	A000005

(Fuente: Arduino [27])



**Figura 2.3** Arduino NANO  
(Fuente: Arduino [28])

Dentro de las especificaciones técnicas que son brindadas por los fabricantes de Arduino; y de los microcomponentes electrónicos como circuitos integrados, microprocesadores, microcontroladores, etc., se brinda también información sobre la distribución de los pines en la tarjeta (ver Figura 2.4 y Tabla 2.2) para darle un correcto uso y disminuir al máximo el riesgo de malograr la placa o parte de ella.



**Figura 2.4** Mapa de pines del Arduino NANO  
(Fuente: Arduino [29])

**Tabla 2.2** Descripción de pines de un Arduino NANO

Pin No	Nombre	Tipo	Descripción
1-2, 5-16	D0-D13	I/O	Entradas/Salidas digitales
3, 28	RESET	Input	Reset (Activo bajo)
4, 29	GND	PWR	Tierra
17	3V3	Output	Salida de 3.3V
18	AREF	Input	Referencia del ADC
19-26	A7-A0	Input	Entradas Analógicas
27	+5V	I/O	Salida/Entrada de alimentación de 5V
30	VIN	PWR	Alimentación

(Fuente: Arduino [29])

#### 2.4.2. Raspberry Pi

Como lo indica en su página web oficial, Raspberry Pi es “una minicomputadora del tamaño de una tarjeta de crédito que se puede conectar a los periféricos más comunes (monitor o TV vía HDMI, teclado, mouse, etc.), la cual es muy útil para aprender programación y para realizar proyectos básicos de electrónica. En comparación con una computadora desktop corriente, el Raspberry Pi puede realizar muchas de las mismas acciones, como la ofimática (hojas de cálculo, editores de texto, etc.), navegar por la red e incluso jugar videojuegos. La minicomputadora Raspberry Pi es la tercera marca de computadora más vendida en el mundo, usada por adultos y niños para fines personales, académicos, didácticos y comerciales.” [30]

Los ES fabricados por la fundación Raspberry Pi, también ofrecen una amplia variedad de modelos, cada uno diseñado y destinado para un público y propósito diferente. Además, se pueden encontrar dentro de un mismo modelo diferentes versiones del minicomputador, en la que solamente varía la intensidad (o la carga) de trabajo soportable por el mismo.

En el caso del software asociado a los dispositivos de Raspberry Pi, se tienen que mencionar 4 elementos esenciales:

##### a) Raspbian

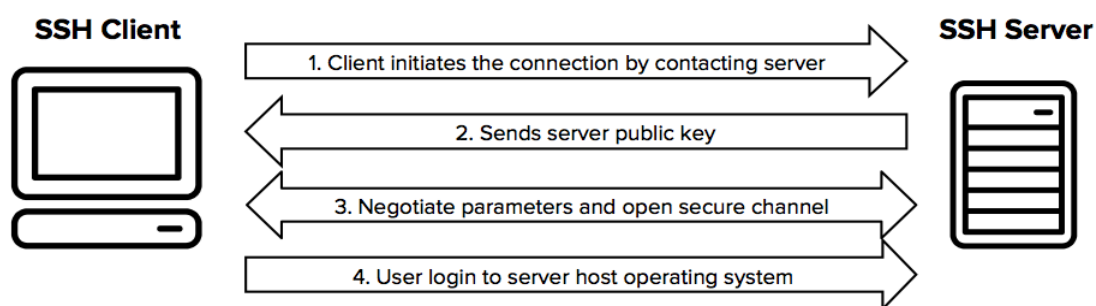
Raspbian es el sistema operativo (OS, *Operating System*) oficial de todos los modelos y versiones de Raspberry Pi. [31] Es un OS libre basado en el Debian y optimizado exclusivamente para el hardware Raspberry Pi. Mientras que un OS se encarga de configurar los programas básicos, aplicaciones y utilidades que un computador necesitan para arrancar, el Raspbian, también viene con más de 35,000 paquetes precompilados para alcanzar el mejor rendimiento posible del Raspberry Pi. [32]

## b) Python

De acuerdo a su página oficial, Python es “un lenguaje de programación interpretado, interactivo y orientado a objetos. Dentro de sus atributos incorpora a los módulos, excepciones, tipificación dinámica, tipos de data dinámica de muy alto nivel y clases. A su vez, combina una potencia notable con una sintaxis muy clara. Tiene interfaces para importar bibliotecas y realizar muchas llamadas al sistema, así como para trabajar con un sistema de ventanas. Es extensible en C o C++. También es aplicable como lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Y por último, Python es portátil, se puede ejecutar en diversas variantes de Unix, así como en Mac y en versiones de Windows 2000 en adelante.” [33] Cabe mencionar que gracias a lo último y a que es un software libre, se ha hecho muy popular y querido por el público.

## c) SSH (*Secure Shell*)

Como se indica en su página oficial, el SSH es un paquete de software que habilita la administración segura del sistema y transferencia de archivos sobre redes inseguras. [34] En otras palabras, permite que el usuario pueda acceder, desde un dispositivo con acceso a la red y una cuenta de cliente SSH, a una terminal del Raspberry Pi y así poder ejecutar diversos comandos remotamente. [35] Surgió como un reemplazo seguro de la comunicación Telnet y en lugar de realizar la comunicación sin algún tipo de cifrado, utiliza técnicas criptográficas para garantizar una comunicación segura y encriptada. [36] Se puede apreciar un diagrama con el funcionamiento del protocolo SSH en la Figura 2.5.



**Figura 2.5** Funcionamiento del Protocolo SSH  
(Fuente: SSH [34])

## d) VNC (*Virtual Network Computing*)

El VNC es un sistema para compartir gráficamente el control sobre una computadora, es decir, se le permite al usuario controlar remotamente la interfaz gráfica (pantalla) de un ordenador desde otro dispositivo, siempre y cuando el primero ejecute el software VNC Server (Servidor) y el último, el VNC Viewer (Veedor). Mediante esta conexión, se

transmiten las entradas generadas por el ratón, como los movimientos del cursor y por el teclado (ver ejemplo en la Figura 2.6). [37]



**Figura 2.6** Ejemplo de Interfaz gráfica del Raspberry Pi, vista desde un iPhone  
(Fuente: Raspberry Pi [37])

Tras mencionar las diversas características generales de las minicomputadoras Raspberry Pi, se ha visto conveniente para la elaboración de este trabajo la utilización del Raspberry Pi 2 Modelo B (ver Figura 2.5).



**Figura 2.7** Raspberry Pi 2 Modelo B  
(Fuente: Raspberry Pi [38])



Debido a ello, se mencionarán en la Tabla 2.3 algunas especificaciones técnicas que presenta el mencionado computador:

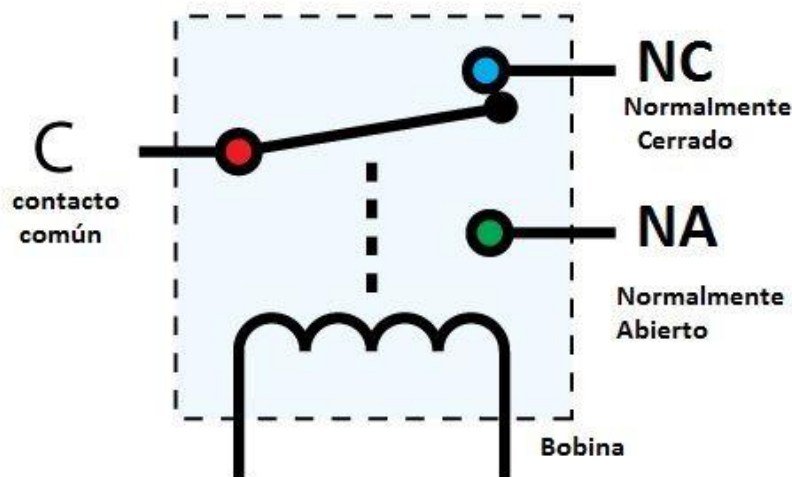
**Tabla 2.3** Especificaciones Técnicas del Raspberry Pi 2 Modelo B

Procesador	Quad Core ARM Cortex-A7@900MHz
Memoria RAM	1GB
Puertos Ethernet	1
Puertos USB	4
Pines GPIO	40
Puertos HDMI	1
Entradas multimedia	1 combinada de audio y video compuesto de 3.5mm
Interfaz de cámara (CSI)	1
Interfaz de pantalla (DSI)	1
Ranura para Micro SD	1
Tarjeta gráfica	1 núcleo gráfico VideoCore IV 3D

(Fuente: Raspberry Pi [38])

## 2.5. Relés o Relays

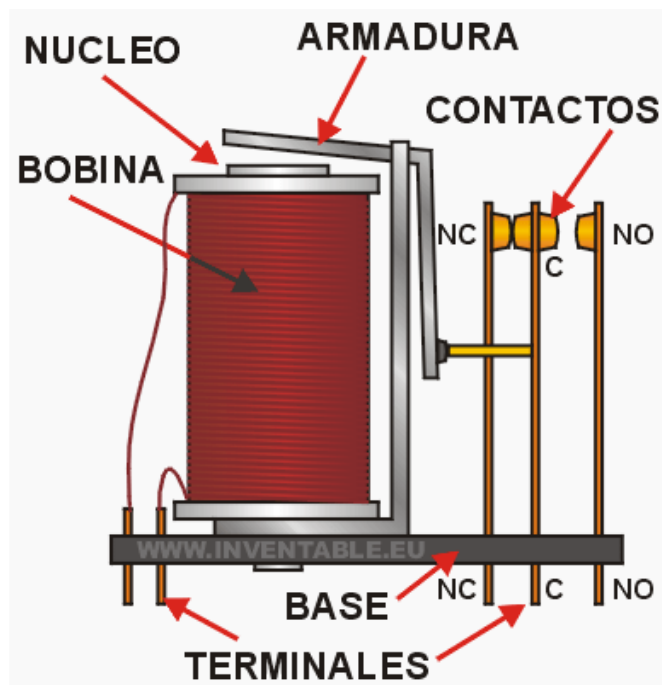
Un relé (o relay) es un dispositivo electromecánico muy utilizado que permite el control de una línea eléctrica de media o alta potencia, mediante un circuito electrónico de baja potencia. [39] Fue inventado en la primera mitad del siglo XIX por el físico estadounidense Joseph Henry, quien empleó una bobina y un electroimán (ver Figuras 2.8 y 2.9) para desarrollar un dispositivo que supere al interruptor convencional. [40]



**Figura 2.8** Estructura circuital de un relé  
(Fuente: ÁreaTecnología.com [41])

Dentro de las principales ventajas que posee, se pueden apreciar:

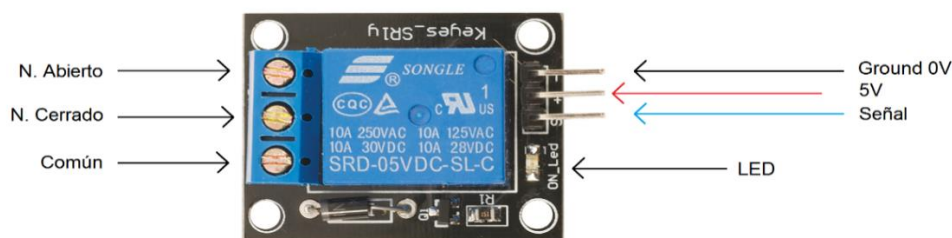
- Permite el control de un dispositivo a distancia y de manera segura.
- Posee dos circuitos completamente separados, lo que permite que la corriente del circuito de activación del relé sea muy pequeña; mientras que la corriente del segundo circuito puede llegar a ser muy grande.
- Con una misma señal de control se pueden activar varios relés casi en simultáneo.



**Figura 2.9** Estructura interna de un relé  
(Fuente: Inventable.eu [39])

### 2.5.1. Módulo Relé 5VDC

Debido a que el relé es un dispositivo muy útil en la electrónica, se han fabricado modelos de relés con características específicas y enfocadas a diversas situaciones. Como un ejemplo de ellas, se tienen a los que son compatibles con los sistemas embebidos basados en microcontroladores (como el Arduino, PIC, etc.) cuyas señales de entrada y salida son los estándares de los circuitos TTL (3.3V y 5V). Es por ello que se diseñaron los relés exclusivos con un voltaje de operación de 5V (ver Figura 2.10). En ellos, el circuito de activación requiere de una señal de 0V y 5V, para bloquear y permitir, respectivamente, el paso de la corriente de la línea de alta potencia.



**Figura 2.10** Módulo de relé para Arduino  
(Fuente: OpenLanuzza [42])

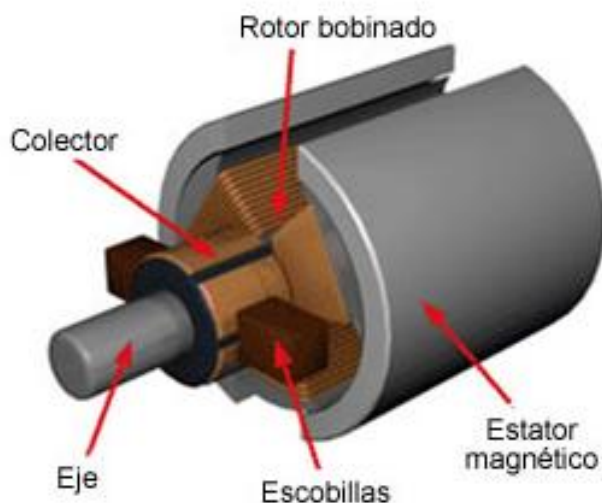
Como se puede apreciar en la Figura 2.10, el módulo cuenta con 3 pines en el lado derecho, los cuales deben ser conectados a tierra (“Ground 0V”), a una señal de 5V (“5V”) y a la señal de control que será la que active o desactive la bobina interna o interruptor (“Señal”). Asimismo, en el lado izquierdo se pueden observar los 3 bornes a controlar: Normalmente Abierto (NO), Normalmente Cerrado (NC) y Común (COM). Los 2 últimos se encuentran en circuito cerrado por defecto, mientras que el NO se encuentra aislado. Al mandar un “1” lógico (5V) por la señal de control, el COM se desconectará de NC y se conectará con NO (al mismo tiempo que se encenderá el LED indicador que viene con el módulo); al mandar un “0” lógico (0V) por la señal de control, el COM volverá a su posición inicial (y a su vez, el LED indicador se apagará). [42]

## 2.6. Motores DC

Un motor, según la RAE, “es una máquina destinada a generar energía mecánica a expensas de otra fuente de energía, donde dicha energía mecánica es capaz de realizar algún trabajo.” [43] De esto, se puede deducir que un motor térmico transforma la energía térmica en energía mecánica; un motor hidráulico, energía hidráulica en mecánica; y así existen muchos otros tipos de motores.

En el caso del motor DC, como es de esperarse, emplea electricidad de corriente continua para generar energía mecánica (movimiento mecánico rotatorio); su funcionamiento se basa principalmente en fuerzas magnéticas, y debido a ello, el motor dispone de ciertos mecanismos internos electrónicos o electromecánicos para administrarlas. Los elementos principales del motor son: el estator, el rotor y el colector o acumulador (ver Figura 2.11). El estator es el componente fijo que mayormente se encuentra compuesto de un electroimán, mientras que el rotor, también compuesto por un electroimán, es el componente giratorio de la máquina, anclado a un eje central en torno del cual rotará. En vista que las propiedades magnéticas de un imán o electroimán nos indican que los polos opuestos se atraen y polos iguales se repelen, se polarizan el estator y el rotor de manera tal que por un lado del rotor sea atraído y por el otro repelido, resultando así un movimiento

giratorio o rotatorio inicial. Sin embargo, este proceso culminaría rápidamente, de no ser por el colector, quien se encarga de conmutar las polaridades del estator (invirtiendo el sentido del flujo de corriente) para que el rotor continúe su proceso de rotación. De esta manera, el acumulador puede controlar la velocidad de giro del rotor, cuya variación será directamente proporcional al aumento o disminución que se pueda presentar en la intensidad de corriente con la que se alimenta al motor, es decir, a mayor corriente de alimentación, mayor energía mecánica podrá brindar el rotor y viceversa. [44] [45]



**Figura 2.11** Constitución de un motor DC  
(Fuente: Aficionados a la Mecánica [46])

Dentro de las principales características y ventajas que presenta el motor DC se tienen: [44] [45]

- Presenta un torque de arranque relativamente alto, lo que permite que rompa la inercia con mayor facilidad que otros motores.
- Es muy versátil, ya que puede ser utilizado en aplicaciones de alta potencia (procesos industriales) y de baja potencia (el presente trabajo).
- Puede trabajar a diferentes velocidades, sean altas o bajas, sólo dependen del colector para que realice las variaciones correspondientes.

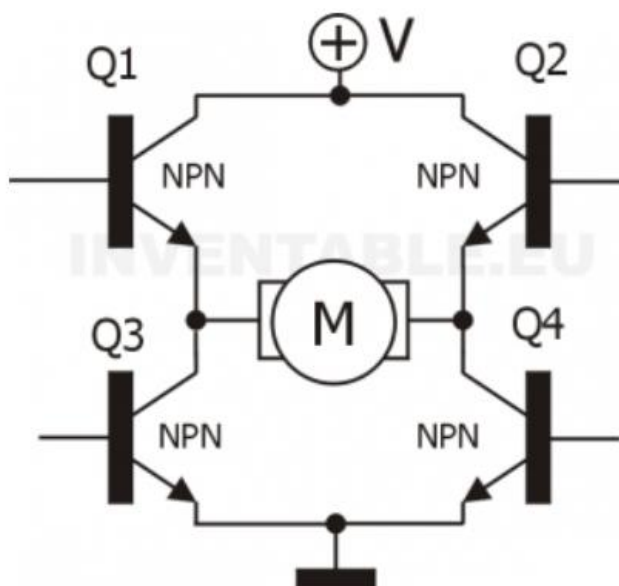
Dentro de las características recientemente descritas de los motores DC, se tiene la presencia de un gran torque de arranque, motivo suficiente para creer que el consumo de corriente es elevado también, como que lo es. Dicho esto, se puede entender que resulta imposible para un microcontrolador poder energizarlo lo suficiente como para que funcione. Es por ello, que se tiene la necesidad de recurrir a una fuente de alimentación y a un circuito externo que se encargue del control del giro del motor (ya sea en sentido horario o

antihorario), mediante la polarización del mismo. Como fuente externa se empleará un transformador de 220VAC~12VDC; mientras que, para el control del sentido de giro, se tienen dos opciones: Un puente H y el IC L293D. A continuación se describirán las opciones mencionadas:

### 2.6.1. Puente H con transistores

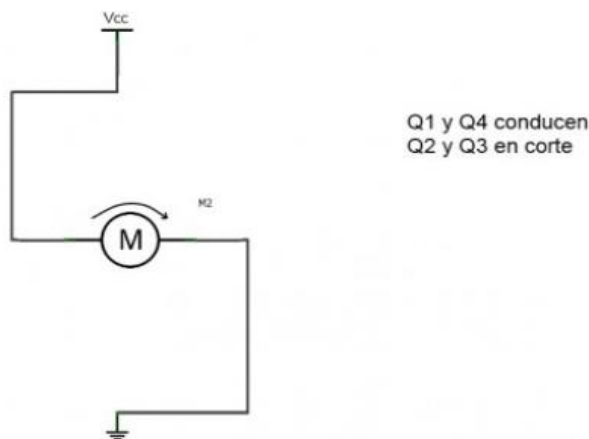
Un puente H (llamado así por la forma que presenta su configuración electrónica, véase la Figura 2.12) es utilizado para controlar motores DC de baja tensión y puede estar compuesto de diversos elementos, tales como: interruptores, transistores NPN, PNP, MOSFET, etc. En este inciso, se describirá el puente H con transistores NPN por ser los más comerciales y estar incluidos en la configuración del IC L293D.

Para esta ocasión, se hace uso de la función de interruptor que posee el transistor, en la que se alterna entre sus estados de corte y saturación (circuito abierto y cortocircuito, respectivamente), cerrando y permitiendo el paso de la corriente a través de ellos. Esto, sumado a la configuración del puente H, especialmente diseñado para que se logre polarizar al motor de las dos maneras posibles, permite conseguir el control sobre el sentido de giro del motor, simplemente con un correcto manejo de los transistores. Cabe mencionar que durante el diseño del puente H, se escogen los transistores, dependiendo únicamente de las características y especificaciones técnicas del motor, ya que ello determina la potencia a la cual trabajará el sistema.

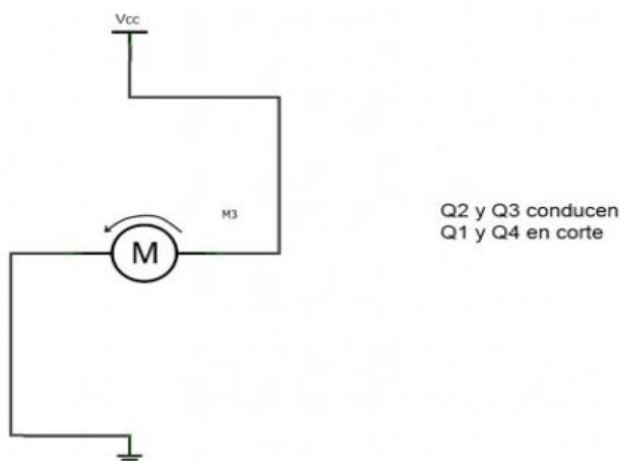


**Figura 2.12** Puente H con transistores NPN sin la lógica de control  
(Fuente: Inventable.eu [47])

A continuación, se mostrarán los dos casos de funcionamiento del puente H en las Figuras 2.13 y 2.14:



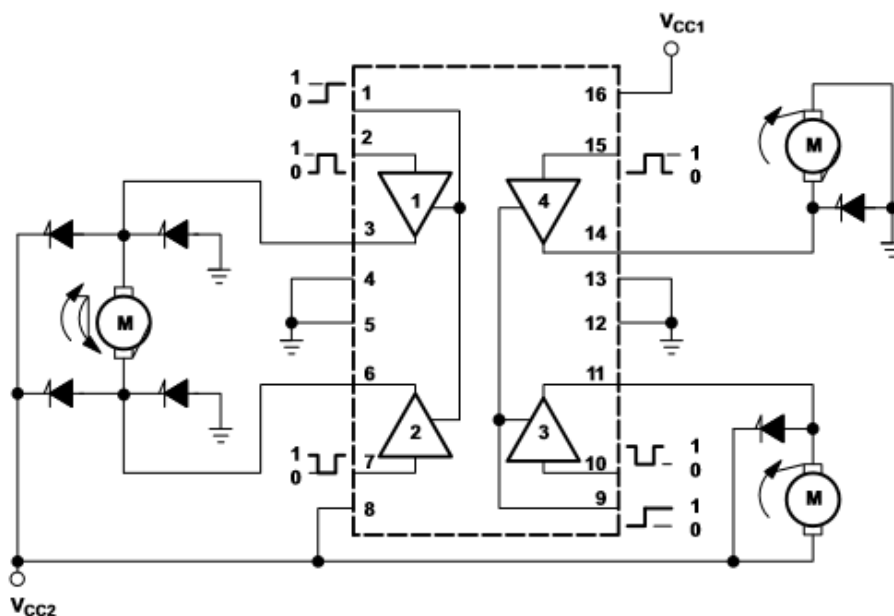
**Figura 2.13** Motor accionado con sentido de giro horario  
(Fuente: DIYMakers.es [48])



**Figura 2.14** Motor con sentido de giro anti horario  
(Fuente: DIYMakers.es [48])

### 2.6.2. Circuito Integrado L293D

El circuito integrado L293D cumple la misma función que el puente H con transistores anteriormente descrito, sin embargo, dentro de un mismo IC se puede controlar hasta 4 motores DC con un solo sentido de giro o 2 motores con ambos sentidos de giro, de 600mA cada uno (con picos máximos de 1.2A cada uno), todo esto debido a que posee internamente 4 mitades de puente H. Para conseguir ello, el L293D requiere de dos fuentes de alimentación: la primera fuente para la parte lógica del chip (por lo general 5V) y la segunda para la parte de potencia y control de motores (desde 4.5V a 36V, dependiendo del motor). [49] En la Figura 2.15 se aprecia la conexión del chip con tres motores DC.



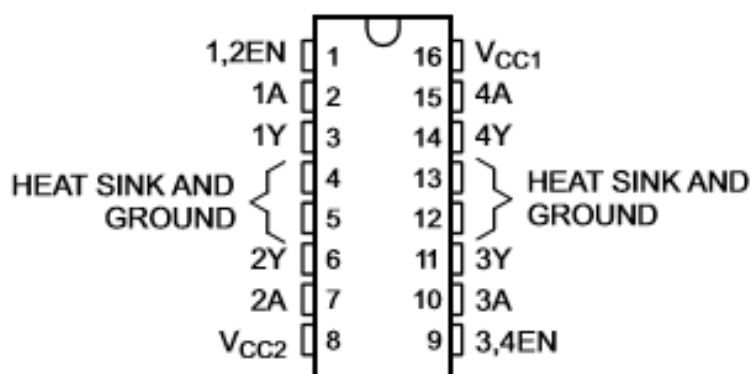
**Figura 2.15** Diagrama de conexión del IC L293D con 3 motores  
(Fuente: Texas Instrument [49])

Cabe mencionar que los diodos apreciados en la Figura 2.15, se encuentran de manera interna en el IC L293D. En la Figura 2.16 y la Tabla 2.4, se presentan la distribución y funciones de los pines del chip:

**Tabla 2.4** Descripción de los pines del L293D

Nº Pin	Tipo	Función
1	Entrada	Habilitadores de los canales 1 y 2 (Activo alto)
2, 7, 10, 15	Entrada	Entradas de los drivers
3, 6, 11, 14	Salida	Salidas de los drives
9	Entrada	Habilitadores de los canales 3 y 4 (Activo alto)
4, 5, 12, 13		Tierra
16		Fuente de 5V para el circuito lógico
8		Fuente de potencia entre 4.5V y 36V

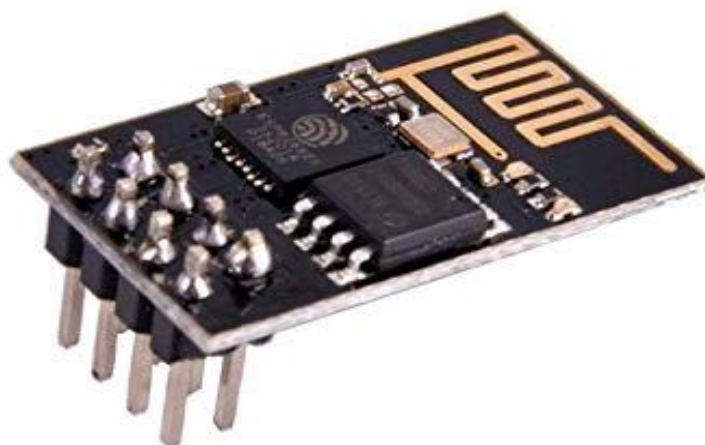
(Fuente: Texas Instrument [49])



**Figura 2.16** Distribución de pines del circuito integrado L293D  
(Fuente: Texas Instrument [49])

### 2.7. Módulos de comunicación inalámbrica – Módulo Wi-Fi ESP-01

Los módulos de comunicación inalámbrica son PCB (*Printed Circuit Board*) que poseen un circuito tal, que les permite comunicarse a otros dispositivos o módulos mediante algún canal inalámbrico (Bluetooth, ZigBee, Wi-Fi, etc.). Éstos pueden ser diseñados con un MCU (*Micro Controller Unit*) interno, a fin de poder controlar otros módulos y dispositivos, ejecutando programas previamente cargados en la memoria flash del mismo. Para cubrir las funciones de conexión requeridas en el presente trabajo, se ha visto conveniente el uso del módulo Wi-Fi ESP-01 (ver Figura 2.17), por ser un módulo capaz de comunicarse inalámbricamente y de controlar otros dispositivos con su microcontrolador interno, el circuito integrado ESP8266EX.



**Figura 2.17** Módulo Wi-Fi ESP-01  
(Fuente: Amazon.es [50])

En la Tabla 2.5 se muestran las especificaciones técnicas más resaltantes e importantes del módulo para este trabajo:

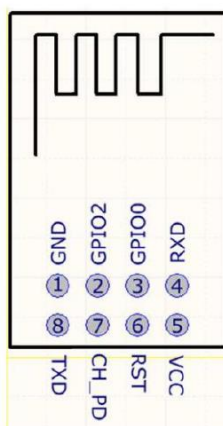


**Tabla 2.5** Especificaciones Técnicas del Módulo Wi-Fi ESP-01

Categoría	Ítems	Valores
Parámetros Wi-Fi	Protocolos Wi-Fi	802.11 b/g/n
	Rango de frecuencias	2.4GHz - 2.5GHz (2400M - 2483.5M)
Parámetros de Hardware	Bus periférico	UART/HSPI/I2C/I2S/Ir Control remoto
		GPIO/PWM
	Voltaje de operación	3.0 ~ 3.6V
	Corriente de operación	Valor promedio: 80mA
	Rango de T° de operación	-40° ~ 125°
	Rango de T° ambiente	Temperatura normal
	Tamaño del módulo	14.3mm*24.8mm*3mm
Interfaz externa	N/A	
Parámetros de Software	Modos Wi-Fi	Station/SoftAP/SoftAP+station
	Seguridad	WPA/WPA2
	Encriptación	WEP/TKIP/AES
	Actualización de firmware	UART Download / OAT (vía internet) / descarga e instalación vía host
	Desarrollo de software	Support Cloud Server Development / SDK for custom firmware development
	Protocolos de red	IPv4, TCP/UDP/HTTP/FTP
	Configuración de usuario	AT Instruction Set, Cloud Server, Android/iOS App

(Fuente: Datasheet del ESP-01 [51])

Asimismo, se grafica la distribución de pines en la Figura 2.18 y la descripción de los mismos en la tabla 2.6:

**Figura 2.18** Mapa de pines del módulo Wi-Fi ESP-01

(Fuente: Datasheet del ESP-01 [51])

**Tabla 2.6** Descripción de pines del Módulo WI-Fi ESP-01

Pin N°	Nombre del Pin	Función
1	GND	Tierra
2	GPIO2	Pin general de E/S
3	GPIO0	Pin general de E/S
4	RxD	UART0, Pin de recepción de data
5	VCC	Alimentación de 3.3V
6	RST	1.- Pin de reset externo, activo bajo 2.- MCU externo
7	CH_PD	Chip Enable Pin - Activo alto
8	TxD	UART0, Pin de transmisión de data

(Fuente: Datasheet del ESP-01 [51])

### 2.7.1. ESP8266EX

El ESP8266EX es un SoC producido por el fabricante chino Espressif Systems, diseñado especialmente para satisfacer las necesidades básicas y comunes en la industria del IoT: Consumo bajo y eficiente de energía, diseño compacto y rendimiento confiable. Es un SoC muy versátil, que puede trabajar de manera autónoma (debido al microprocesador y a la memoria flash, contenidos en el mismo SoC) o pertenecer a un microcontrolador mayor cumpliendo la función de comunicación vía Wi-Fi entre dispositivos. [52]

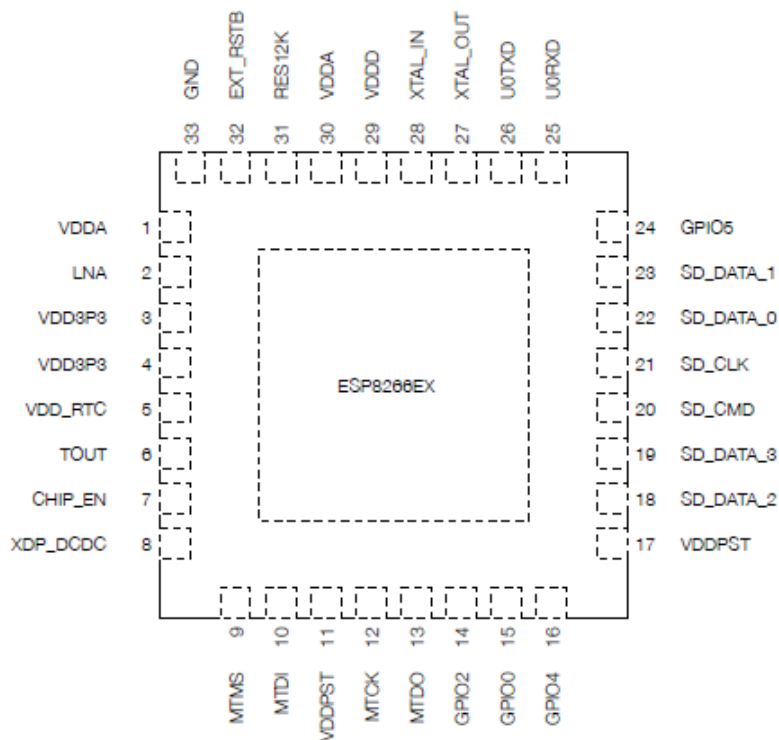
Se ha visto necesario para las expectativas de la presente tesis emplear únicamente sus características de conectividad Wi-Fi, las cuales se describen en la Tabla 2.7.

**Tabla 2.7** Características Wi-Fi del ESP8266EX

Certificación	Wi-Fi Alliance
Protocolos	802.11 b/g/n (HT20)
Rango de frecuencias	2.4GHz – 2.5GHz (2400MHz – 2483.5MHz)
Potencia de Tx	802.11 b: +20 dBm
	802.11 g: +17 dBm
	802.11 n: +14 dBm
Sensibilidad de Rx	802.11 b: -91 dbm (11Mbps)
	802.11 g: -75 dbm (54Mbps)
	802.11 n: -72 dbm (MCS7)
Antena	PCB Trace, Externa, Conector IPEX, Chip cerámico

(Fuente: Datasheet del ESP8266EX [52])

Asimismo, se procede a mostrar la distribución de pines del ESP8266EX en la Figura 2.19.



**Figura 2.19** Distribución de pines del ESP8266EX  
(Fuente: Datasheet del ESP8266EX [52])

### 2.7.2. Actualizando el firmware del ESP-01

El firmware es comúnmente definido como un tipo de programa inherente a los dispositivos electrónicos, no puede ser clasificado como software ni como hardware, sino como una combinación de ambos, ya que emplea un archivo ejecutable (.exe) e incluye un sistema integrado del dispositivo electrónico. Se almacena en la memoria ROM (*Read Only Memory*) y sirve para designar al software que se encuentra adherido al hardware, es decir, al grabado de fábrica. No se puede alterar fácilmente y cada vez que se enciende el hardware, éste arranca automáticamente. Finalmente, el firmware es capaz de almacenar perennemente sus instrucciones y no requieren de una fuente de alimentación. [53] [54]

Debido a que el módulo Wi-Fi ESP-01 trae como firmware de fábrica lo siguiente:

- AT Versión 1.3.0.0 (Publicado el 14/Jul/2016)
- SDK Versión 2.0.0 (Publicado el 14/Jul/2016)

Se realizó una actualización a fin de poder entablar una mejor comunicación con el módulo utilizando comandos actuales. Es por ello, que se instalaron versiones más modernas:

- AT Versión 1.6.0.0 (Publicado el 03/Feb/2018)
- SDK Versión 2.2.0 (Publicado el 03/Feb/2018)

Se requieren de dos elementos principales para realizar la actualización del mismo:

- **Flash Download Tools v3.6.8.**- Es el programa libre oficial que será el encargado de transcribir los firmwares e instalarlos en el módulo. Descargable en:

[https://www.espressif.com/sites/default/files/tools/flash\\_download\\_tools\\_v3.6.8.zip](https://www.espressif.com/sites/default/files/tools/flash_download_tools_v3.6.8.zip) [55]

- **ESP8266 AT bin v1.6.0.0.**- Es la carpeta que contiene los archivos necesarios para actualizar el firmware del módulo. Descargable en:

[https://www.espressif.com/sites/default/files/ap/esp8266\\_at\\_bin\\_v1.6.zip](https://www.espressif.com/sites/default/files/ap/esp8266_at_bin_v1.6.zip) [56]

A continuación se procede a conectar el módulo ESP-01 a un conversor USB a TTL (ver Figura 2.20), tal como se indica en la Tabla 2.8:



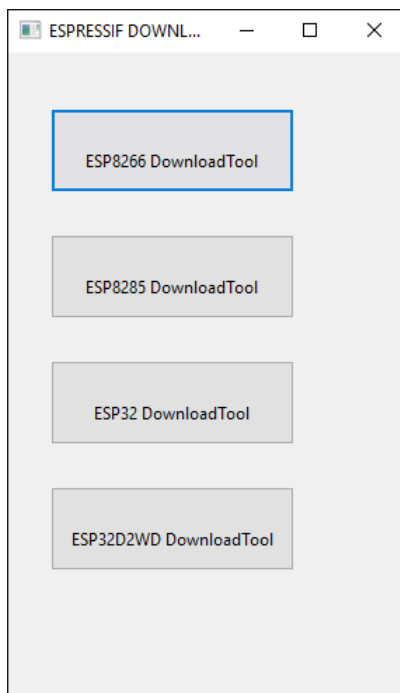
**Figura 2.20** Conversor USB a Serial TTL (Módulo PL2303)  
(Fuente: Naylamp Mechatronics [57])

**Tabla 2.8** Conexiones entre los módulos ESP-01 y PL2303

Pin N°	Pin del ESP-01:	Conectado al PL2303 en el pin:
1	GND	GND
2	GPIO2	No se conecta
3	GPIO0	GND
4	RxD	TxD
5	VCC	3.3V
6	RST	No se conecta
7	CH_PD	3.3V
8	TxD	RxD

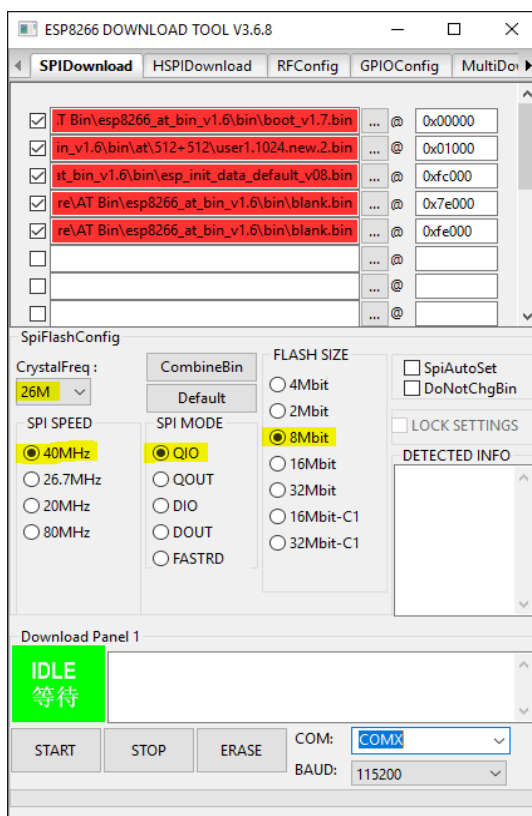
(Fuente: Datasheet del ESP-01 [51])

Después de conectarlo correctamente, se abre el programa Flash Download Tools y se escoge la opción “ESP8266 Download Tool”, como se aprecia en la Figura 2.21:



**Figura 2.21** Ventana inicial del Flash Download Tools  
(Fuente: Elaboración propia)

Tras ello, aparecerá una ventana, la cual se deberá configurar como lo indica la Tabla 2.9:



**Figura 2.22** Ventana de configuración del firmware y sus ubicaciones  
(Fuente: Elaboración propia)

**Tabla 2.9** Archivos del firmware y sus respectivas ubicaciones

Ubicación del archivo en la carpeta descargada	Ubicación de destino (@)
bin\boot_v1.7.bin	0x00000
bin\at\512+512\user1.1024.new.2.bin	0x01000
bin\esp_init_data_default.bin	0xfc000
bin\blank.bin	0x7e000
bin\blank.bin	0xfe000

(Fuente: Elaboración propia)

Cabe mencionar que se deben marcar las alternativas tal y como se resaltan en la Figura 2.22, verificar que el BAUD sea 115200 y que el puerto COM (resaltado en azul “COMX”) coincida con el del módulo PL2303.

Finalmente, dar clic en el botón de “START” y esperar a que se llene la barra horizontal inferior para poder cerrar el programa y retirar el conector USB.

## 2.8. Servidores Web

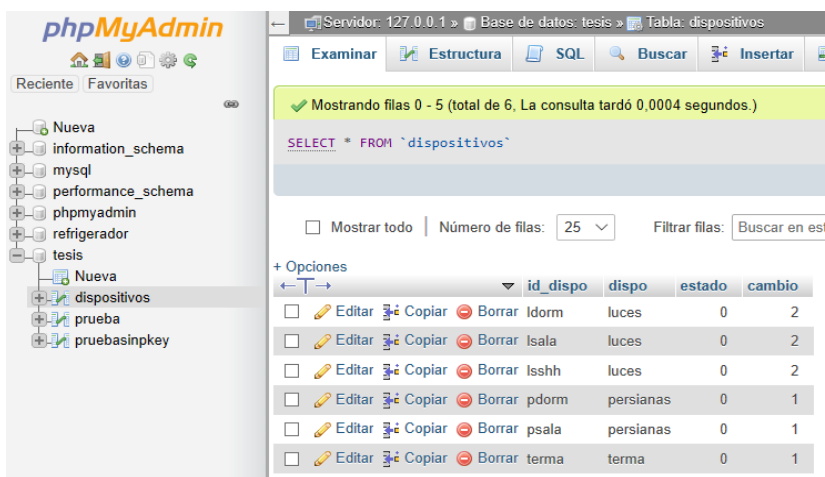
Los servidores web son un tipo de software destinado a la gestión de las aplicaciones web, envía la información de diversas páginas web a los usuarios que la solicitan, siempre y cuando tengan los permisos correspondientes. Su comportamiento se encuentra basado en la lógica del modelo cliente-servidor, en la cual, el cliente, mediante comandos realiza consultas o brinda órdenes al servidor, para que éste las responda o ejecute, según sea el caso. Toda interacción entre cliente y servidor es posible debido al uso del protocolo de comunicación HTTP (*HyperText Transfer Protocol*), así como de un Navegador Web que es el encargado de interpretar las respuestas enviadas por el servidor web (por ejemplo Google Chrome, Mozilla Firefox, Microsoft Edge, Safari etc.). Los servidores web son ejecutables, tanto de manera local como de manera global, estos últimos mediante el uso de internet. Cabe mencionar, que los servidores que se ejecutan en internet, regularmente requieren de un DNS (*Domain Name Server*) que se encarga de traducir las URL en la dirección IP correspondiente al servidor al que se quiere acceder. [58] [59]

Por mencionar algunos ejemplos de los servidores web más populares en la actualidad, se tienen a Apache, Microsoft IIS (*Internet Information Server*), Sun Java System Web Server, Nginx y Lighttpd. Sin embargo, para el presente trabajo se empleará el servidor web Apache, ya que es gratuito, de código abierto y ejecutable en casi cualquier plataforma, factores que contribuyen enormemente a alcanzar las metas propuestas en el primer capítulo. [58] Además, para realizar las pruebas de manera local, se utilizará otro programa, uno que

ayudará a nuestra PC a fungir de servidor web real, pero sin acceso a internet, es decir, que no tendrá un acceso de manera global. Ese programa es XAMPP, un software libre que se puede trabajar en casi todas las plataformas, brinda un servidor Apache, un sistema de gestión de bases de datos MySQL y puede interpretar los scripts en lenguaje PHP, lo que resulta ideal para la realización de pruebas de manera local. [60]

## 2.9. Base de Datos

Una base de datos es un conjunto de datos que se encuentran relacionados y/o tienen un vínculo que los une entre sí (de esto último depende que sean bases de datos relacionales o no relacionales). Asimismo, dicho grupo debe presentar un orden que facilite su futuro uso y a su vez, garantizar la seguridad y veracidad de sus datos (es decir, que no sean alterados). Hoy en día, se tienen muchas implementaciones de una base de datos como por ejemplo: MySQL, Oracle, Microsoft Access, entre otros. [61] Todos ellos brindan diferentes beneficios en cuanto a gestión de la información y de los elementos de las bases de datos, sin embargo, todos tienen en común las rutinas optimizadas para guardar, eliminar, consultar y gestionar la información. La necesidad de emplear estos programas especializados es la misma que se tiene al querer manejar varios millones de registros de forma optimizada. [62] Para mayor facilidad, se utilizará la base de datos MySQL, que es la base de datos de código abierto más popular en el mundo. [63] Además, como se mencionó anteriormente, se adapta satisfactoriamente con el programa XAMPP y su paquete de phpMyAdmin, que es el software que sirve de herramienta para que el usuario pueda interactuar gráficamente con la base de datos (ver Figura 2.23); así como también, haciendo uso de los manuales de MySQL que son fácilmente descargables de su página oficial.



**Figura 2.23** Interfaz gráfica de phpMyAdmin  
(Fuente: Elaboración propia)

## CAPÍTULO III

### DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

#### 3.1. Introducción

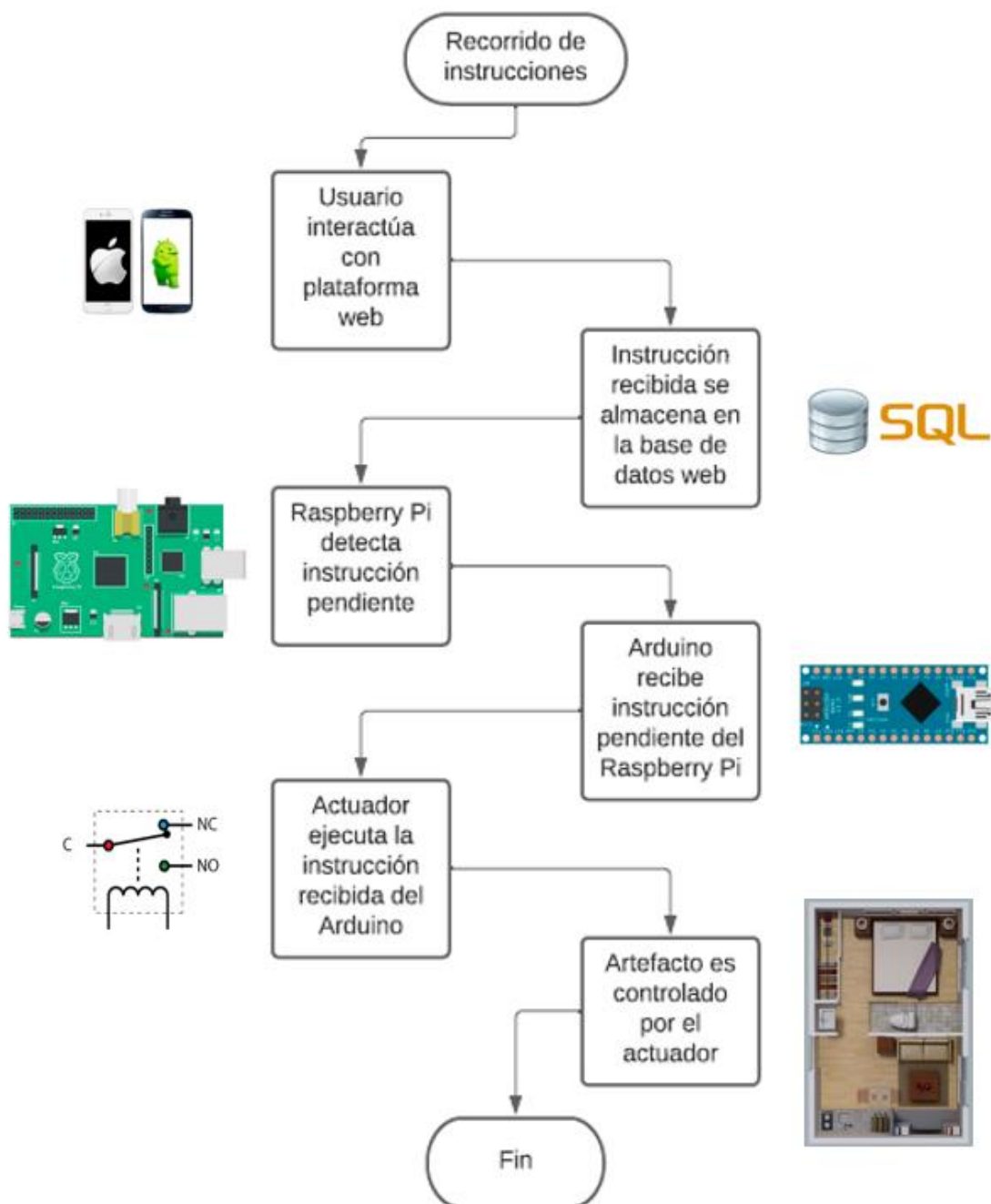
En el tercer capítulo se aplicarán los conceptos adquiridos en el capítulo anterior para entender los pasos seguidos en el diseño e implementación del sistema domótico. Se detallarán todos los aspectos del trabajo, tales como: la arquitectura del sistema, la programación de los microcontroladores (Arduinos), la lógica del servidor local (Raspberry Pi), la configuración del servidor web y la base de datos virtual (en la nube). Asimismo, se publicarán los códigos empleados en la interfaz web y el aplicativo móvil correspondiente.

#### 3.2. Arquitectura del sistema

La arquitectura empleada para conseguir el control remoto de un hogar desde cualquier dispositivo con acceso a internet, cuenta con los siguientes elementos:

- **Dispositivos finales del usuario.-** Son aquellos dispositivos como por ejemplo: computadoras, laptops, celulares, etc. mediante los cuales, el usuario podrá acceder a la interfaz web a fin de controlar sus dispositivos en casa. Cabe mencionar que el usuario únicamente requiere contar con acceso a internet, sin importar si se encuentra en la misma red del hogar o no.
- **Servidor web y base de datos.-** El servidor web será el que le permita al usuario acceder desde cualquier parte del mundo y tendrá constante interacción con la base de datos a fin de gestionar los estados de los dispositivos correcta y eficazmente.
- **Controlador central.-** Será el encargado de supervisar el estado de los dispositivos (información alojada en la base de datos) y ante cualquier orden emitida por el usuario, comunicarse inalámbricamente con los microcontroladores para indicarles cómo proceder.
- **Microcontroladores.-** Recibirán las órdenes del controlador central inalámbricamente e inmediatamente las ejecutarán enviando las señales correspondientes a los actuadores que tendrán a su cargo.
- **Actuadores.-** Serán los dispositivos que interactúen directamente con el artefacto a controlar. Conforme reciba las señales del microcontrolador, activará o desactivará dicho artefacto.





**Figura 3.1** Recorrido de las instrucciones emitidas por el usuario  
(Fuente: Elaboración propia)

Para comprender mejor el funcionamiento del sistema que se grafica en la Figura 3.1, se explicará el proceso siguiendo la ruta que realiza la orden emitida por el usuario a través de la interfaz web.

### 3.3. Desarrollo de la interfaz web y la aplicación móvil

En vista que la interfaz web y la aplicación móvil, serán los intermediarios entre el usuario y el sistema de control, tienen que estar diseñados de manera tal, que su uso sea lo más

intuitivo y sencillo posible. La función principal de estos elementos es la de recibir órdenes del usuario y modificar la columna correspondiente de la base de datos para que el controlador central pueda continuar con el proceso.

En el presente trabajo se ha desarrollado una página web en la que se visualiza la plataforma domótica y desde la cual, se pueden controlar todos los dispositivos mencionados en un principio (luminarias, terma y persianas), instalados en tres ambientes distintos (sala de estar, dormitorio y baño). La aplicación móvil tiene la única función de adaptar la página web al tamaño de pantalla de un dispositivo móvil con OS Android y mostrarlo sin mayor distorsión en sus dimensiones, facilitando, de esta forma, la accesibilidad a la página. A continuación se explicarán las pautas consideradas para el diseño y desarrollo de los mencionados intermediarios.

La interfaz web ha sido desarrollada utilizando dos lenguajes: HTML (*Hipertext Markup Language*) y PHP (*Hipertext Pre-Processor*). El primero, como sus siglas lo indican, es un lenguaje de marcado de hipertexto que básicamente nos permite desarrollar páginas web mediante el uso de etiquetas, que organizan la información a mostrar (sea texto, imagen, video, enlaces, etc.) y personalizarlas con diversos atributos (colores, tamaños, formas, fuentes, etc.); mientras que el segundo, es un lenguaje de programación especialmente creado para el desarrollo web y adaptable en cualquier script de HTML [64]; como sus siglas lo indican, pre-procesan el hipertexto y a diferencia del primero, el script en lenguaje PHP realiza una acción (o serie de acciones) con el hipertexto y no sólo lo organiza, en otras palabras, el script en PHP es un programa que le da dinamismo a la página web.

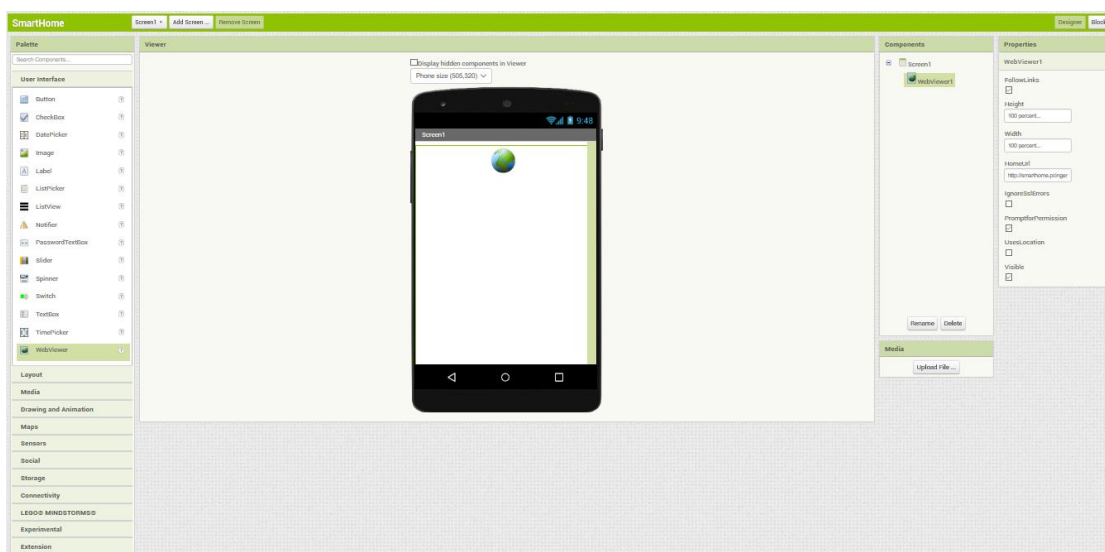
La interfaz web consta de seis ventanas:

1. **Ventana de acceso.**- Es la primera ventana, en ella se solicita un usuario y contraseña para poder acceder a la plataforma de control.
2. **Ventana de menú.**- Es la segunda ventana, en ella habrán dos botones: el primero será un atajo que permita apagar todos los dispositivos del hogar al presionarlo: luminarias, persianas y terma (útil cuando se quiere salir de la vivienda rápidamente); y el segundo será un botón de acceso al menú del hogar.
3. **Ventana del menú de la casa.**- Es la tercera ventana, en ella se encontrarán accesos directos a los menús de control de cada ambiente del hogar: sala, habitación y baño.
4. **Ventana de la sala.**- En ella se podrán apreciar los dispositivos a controlar dentro de la sala que para este caso son: las luminarias y las persianas.
5. **Ventana del dormitorio.**- En ella se podrán apreciar los dispositivos a controlar dentro del dormitorio que para este caso son: las luminarias y las persianas.

**6. Ventana del baño.-** En ella se podrán apreciar los dispositivos a controlar dentro del baño que para este caso son: las luminarias y la terma.

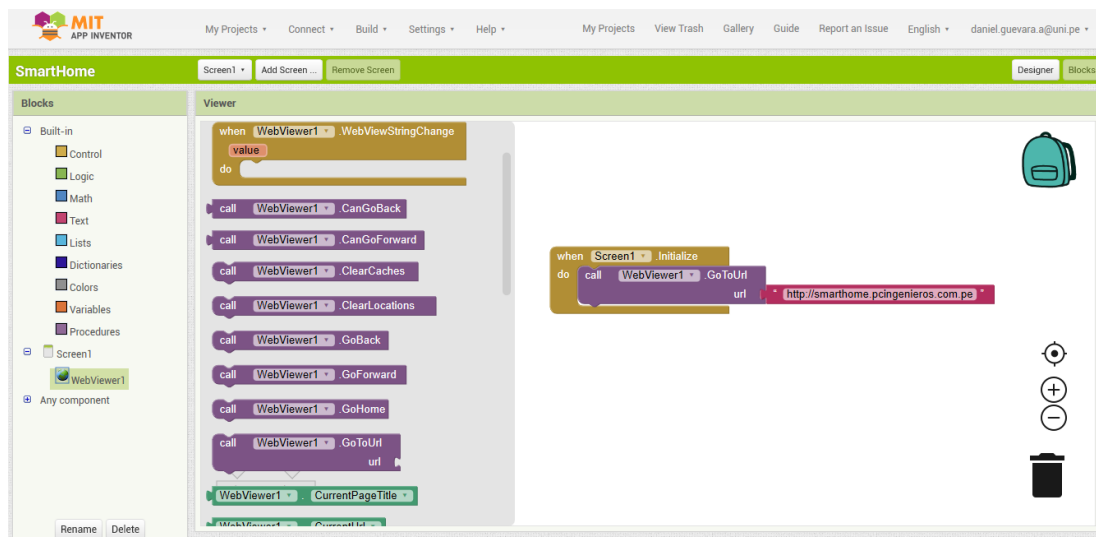
Los scripts de la página web y todas sus ventanas se podrán encontrar en la sección de ANEXOS (Anexo B).

En el caso de la aplicación móvil, se hizo uso del “*MIT App Inventor*”, una página web proporcionada por el MIT (*Massachusetts Institute of Technology*) que facilita la fabricación de aplicaciones móviles para los celulares con OS Android (archivos con terminación .apk). [65] Es una herramienta muy sencilla de utilizar, presenta una interfaz gráfica de diseño bastante intuitiva, la cual no requiere de mayores conocimientos previos para su entendimiento (véase Figura 3.2). Cabe mencionar que se le dio el nombre de “*SmartHome*” a la aplicación móvil, debido a que es la palabra que mejor define la finalidad de la presente tesis.



**Figura 3.2** Interfaz gráfica del *MIT App Inventor* en modo *Designer*  
(Fuente: MIT App Inventor [65])

Asimismo, el *MIT App Inventor* brinda una interfaz gráfica de bloques para la programación de la aplicación móvil, también sencilla de usar, como se puede apreciar en la Figura 3.3. En ella, se observa que al iniciarse la aplicación, se mostrará la página web en la que está almacenada la interfaz web desarrollada para el trabajo. Cabe mencionar, que el único componente utilizado fue el “*WebViewer*”.



**Figura 3.3** Interfaz gráfica del MIT App Inventor en modo *Blocks*  
(Fuente: MIT App Inventor [65])

### 3.4. Configuración de los servicios en internet

Los servicios en internet están conformados por dos elementos muy importantes: un servidor web y una base de datos. La función principal del primero es la de almacenar los scripts (en HTML y PHP) de la interfaz web para que los navegadores los reproduzcan y el usuario pueda interactuar con ella a nivel global. En cambio, el segundo almacenará la información sobre el estado actual de los dispositivos y los cambios solicitados por el usuario. Cabe recalcar que la base de datos se encontrará en constante interacción con el controlador central, a fin de tener una pronta respuesta frente a las órdenes recibidas.

Las órdenes se almacenarán en la columna “cambio” (véase Figura 3.4) de la base de datos, solamente pueden tomar dos valores: los números “1” y “2”. El número “1” se utilizará cuando se quiera encender un dispositivo; por el contrario, el número “2”, cuando se desee apagarlo. La última fila que corresponde al rótulo “todooff” (en la columna de “id\_dispo”), sólo podrá recibir el número “2”, debido a que es un botón de acceso rápido para apagar todos los dispositivos cuando se requiera salir del hogar; todas las demás filas podrán recibir ambos valores mencionados recientemente. La columna “estado” representará el estado actual de los dispositivos, siendo “1” el estado correspondiente a “DISPOSITIVO ENCENDIDO” y “0” a “DISPOSITIVO APAGADO”.

En esta ocasión, se ha hecho uso del servidor web (y dominio) que posee la empresa PC Ingenieros EIRL (pcingenieros.com.pe), quien contrató los servicios de hosting de la empresa estadounidense Bluehost. [66] La información sobre el servidor brindado es la siguiente:

**Tabla 3.1** Información del servidor web

Artículo	Detalle
Versión de cPanel	78.0 (build 46)
Versión de Apache	2.4.39
Versión de PHP	7.0.33
Versión de MySQL	5.6.41-.84.1
Sistema operativo utilizado	Linux
Dirección IP compartida	162.241.218.133

(Fuente: Bluehost [66])

Para poder configurar el servidor web, de modo que almacene los scripts necesarios para la interfaz web, se requiere ingresar a la sección de subdominios del cPanel (Panel de control) donde se creará un subdominio que será el utilizado para la interfaz web. Luego, se accederá al administrador de archivos para poder cargar los scripts correspondientes. En el caso de la base de datos, se tiene que acceder a la interfaz de PhpMyAdmin, que es la interfaz gráfica donde se puede visualizar y gestionar la base de datos MySQL.

Como detalles finales, se indicará lo siguiente:

- Se creó un subdominio “smarthome” dentro del dominio de “pcingenieros.com.pe”, siendo la URL asignada para el servidor del presente trabajo:

`https://smarthome.pcingenieros.com.pe/`

- Se creó una base de datos “pcingeni\_smarthome”, que tendrá de contenido la tabla “dispositivos”. En ella se almacenará toda la información referente a los dispositivos a controlar (véase Figura 3.4).

	id_dispo	estado	cambio
<input type="checkbox"/> Editar Copiar Borrar	ldorm	1	2
<input type="checkbox"/> Editar Copiar Borrar	lsala	0	0
<input type="checkbox"/> Editar Copiar Borrar	lssh	1	2
<input type="checkbox"/> Editar Copiar Borrar	pdorm	0	2
<input type="checkbox"/> Editar Copiar Borrar	psala	0	2
<input type="checkbox"/> Editar Copiar Borrar	terma	0	2
<input type="checkbox"/> Editar Copiar Borrar	todooff	0	0

**Figura 3.4** Tabla “dispositivos” de la base de datos “pcingeni\_smarthome”

(Fuente: Elaboración propia)

### 3.5. Programación del controlador central (Raspberry Pi)

En el presente trabajo, el controlador central representa, haciendo la analogía, lo que para el cuerpo humano viene a ser el cerebro, es decir, recibe información, la interpreta y analiza para luego tomar decisiones y realizar acciones o, en su defecto, encomendarlas a sus subordinados. La manera en que recibe información se da como respuesta a cada solicitud (*query*) que realiza el controlador a la base de datos.

Para cumplir las expectativas de la tesis, se ha visto conveniente implementar el controlador central en una minicomputadora Raspberry Pi 2 Modelo B (véase sección 2.4.2). Como se explicó en la sección indicada, el Raspberry Pi cuenta con un intérprete del lenguaje Python incluido en el sistema operativo Raspbian, Es por ello que para programarlo realizaremos un script en el lenguaje Python, dicho script contendrá un programa que se encargará de entablar una conexión casi constante con la base de datos y recopilará la información de la misma para realizar las acciones pertinentes.

Se procede a explicar brevemente las dos responsabilidades básicas y fundamentales que tendrá el controlador central, las cuales son:

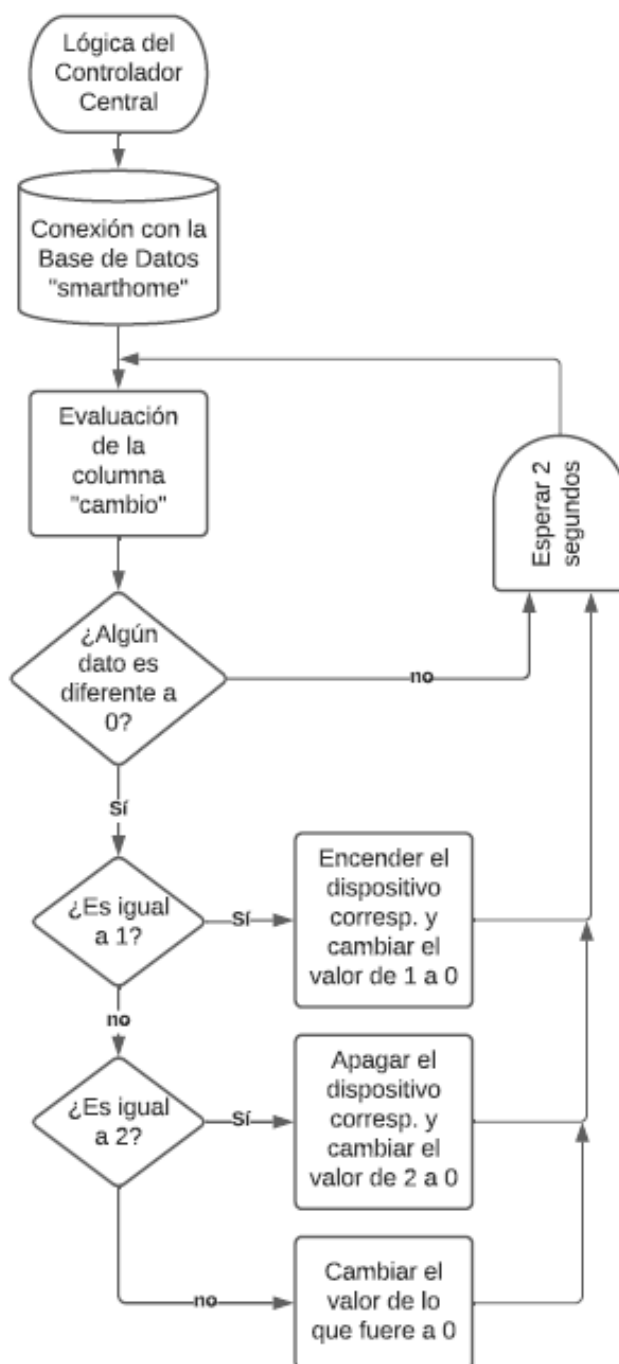
- **Conexión recurrente y periódica con la base de datos.-** El controlador central deberá mantener una conexión casi perenne con la base de datos, con la finalidad de poder notar, casi instantáneamente, cualquier variación en la misma.
- **Tomar acción ante un cambio en la base de datos.-** El controlador central, tras detectar una variación en la base de datos, deberá interpretar la misma y en base al programa, enviar la palabra clave correspondiente al microcontrolador.

Ahora se procederá a explicar detalladamente los criterios de programación tomados en cuenta para con el controlador central.

Para poder cumplir con las funciones mencionadas recientemente, el controlador central se verá en la necesidad de contar con tres librerías externas, las cuales se mencionarán y explicarán a continuación:

- **Librería `mysql.connector`.**- Esta librería se empleará para poder entablar conexión con una base de datos MySQL a través de un nombre de usuario y su respectiva contraseña. Además, si dicho usuario cuenta con los privilegios correspondientes, puede ingresar, modificar y recopilar la información almacenada.
- **Librería `time`.**- Esta librería se empleará para tener control de la frecuencia con la cual el Raspberry Pi entable conexión con la base de datos.
- **Librería `socket`.**- Esta librería se empleará para enviar las palabras clave a los microcontroladores mediante la red de Wi-Fi.

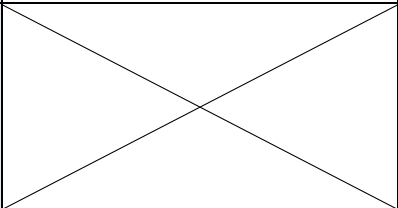
Finalmente, se presentará la lógica que seguirá el controlador central para realizar determinadas acciones en la Figura 3.5 y la Tabla 3.2.



**Figura 3.5** Diagrama de flujo de la lógica del controlador central  
(Fuente: Elaboración propia)

Es preciso aclarar que la lógica del controlador central es un bucle infinito, es decir, su ciclo se va a repetir siempre que éste se encuentre encendido.

**Tabla 3.2** Lógica del controlador central frente a las variaciones percibidas

Acción a tomar cuando:"	"cambio" = "1"	"cambio" = "2"
"id_dispo" = "ldorm"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de encendido a luces del dormitorio</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de apagado a luces del dormitorio</b>
"id_dispo" = "lsala"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de encendido a luces de la sala</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de apagado a luces de la sala</b>
"id_dispo" = "lssh"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de encendido a luces del baño</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de apagado a luces del baño</b>
"id_dispo" = "pdorm"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal para abrir las persianas del dormitorio</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal para cerrar las persianas del dormitorio</b>
"id_dispo" = "psala"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal para abrir las persianas de la sala</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal para cerrar las persianas de la sala</b>
"id_dispo" = "terma"	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "1"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de encendido a la terma</b>	<b>Modifica en su fila:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía señal de apagado a la terma</b>
"id_dispo" = "todooff"		<b>Modifica en todas las filas:</b> <ul style="list-style-type: none"> <li>➤ "estado" = "0"</li> <li>➤ "cambio" = "0"</li> </ul> <b>Envía la señal de apagado a todos los microcontroladores.</b>

(Fuente: Elaboración propia)

Para explicar el contenido de la Tabla 3.2, se recorrerá la ruta que sigue la orden emitida por el usuario. Se tiene que la columna "cambio" de la base de datos será modificada. Al percibir la variación, el controlador central modificará la columna "estado" y la columna



“cambio” (valga la redundancia); asimismo, enviará a el o los microcontroladores correspondientes, la palabra o palabras clave necesarias para cumplir con la orden del usuario. Cabe mencionar, que en el caso de la terma hay una situación especial; con la finalidad de disminuir las responsabilidades del usuario, se programó al controlador central para que la apague después de transcurrirse un lapso de tiempo prudente, como para que caliente el agua, pudiendo ser de segundos, minutos u horas, dependiendo de lo que decida el usuario. Para fines de practicidad al momento de demostrar el apagado automático de la terma, se configuró un lapso de 10 segundos, tras los cuales, de no ser apagada la terma manualmente por el usuario, será el Raspberry Pi el encargado de desactivarla. Se podrá observar el programa en la sección de ANEXOS (Anexo C)

### **3.6. Diseño y programación de los microcontroladores y actuadores**

Los microcontroladores y actuadores serán los que ejecuten físicamente la orden emitida por el controlador central, es debido a ello que se relacionarán directamente con los artefactos del hogar. Cabe mencionar que la interacción en este nivel será netamente alámbrica (cableada).

Tal como se pudo visualizar en la arquitectura del sistema (ver Figura 3.1), se busca controlar tres artefactos del hogar; para ello se hará uso principalmente del Arduino NANO, quien trabajará en equipo con el módulo relé NO 5VDC (véase secciones 2.4.1 y 2.5.1 respectivamente) o con el IC L293D (sección 2.6.2), presentes en tres diseños distintos según el artefacto. En esta sección se describirán los tres diseños correspondientes: el diseño de control de las luminarias de tres habitaciones distintas (sala de estar, dormitorio y baño), el diseño de control de las persianas instaladas en dos habitaciones diferentes (sala de estar y dormitorio) y el diseño de control de la terma eléctrica (baño).

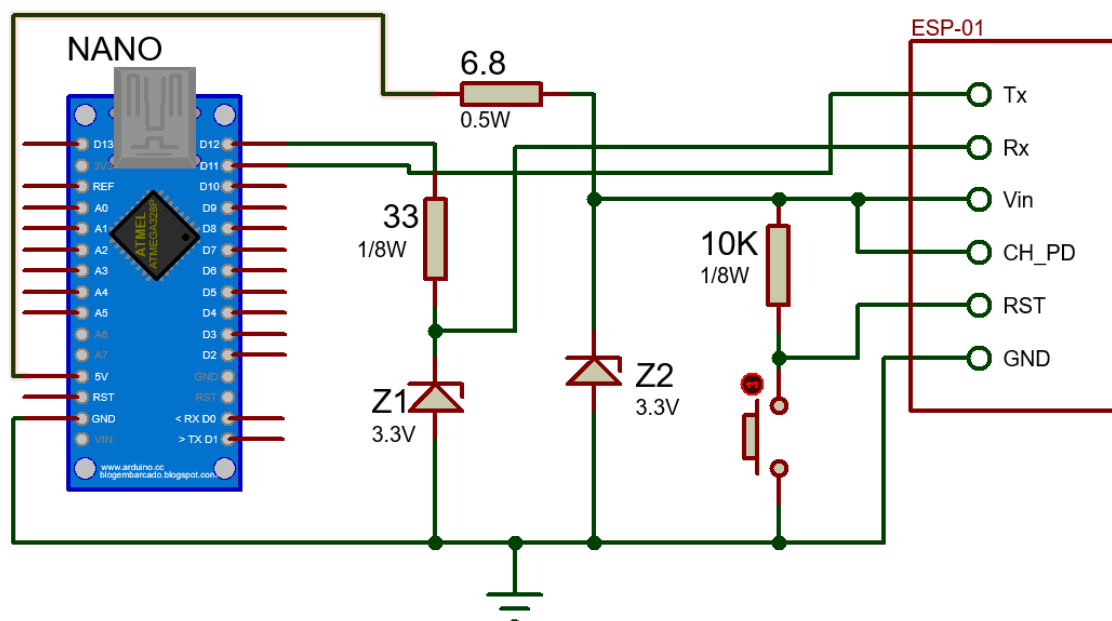
Como un detalle previo al diseño de los circuitos de control, se tiene la conexión entre el Arduino NANO y el módulo Wi-Fi ESP-01, la misma que permitirá la comunicación inalámbrica entre el Arduino NANO y el Raspberry Pi.

#### **3.6.1. Conexión entre el Arduino NANO y el módulo Wi-Fi ESP-01**

En vista que el Arduino NANO es un microcontrolador pensado para circuitos completamente alambrados, no cuenta con ninguna antena. Es debido a ello, que se requiere de un módulo externo, el módulo Wi-Fi ESP-01 en esta ocasión, para establecer una comunicación inalámbrica.

El módulo ESP-01 trabaja a 3.3V, a diferencia del Arduino NANO que lo hace a 5V. Es por tal motivo que establecer una conexión directa entre ambos, causaría que el módulo Wi-Fi se malograra. La solución planteada en esta tesis, es el diseño de un circuito mediador que

convierta toda señal de 5V en 3.3V y viceversa. Es así como se logra que ambos módulos compartan una misma fuente de alimentación y se comuniquen entre ellos. El circuito descrito se aprecia en la Figura 3.6.

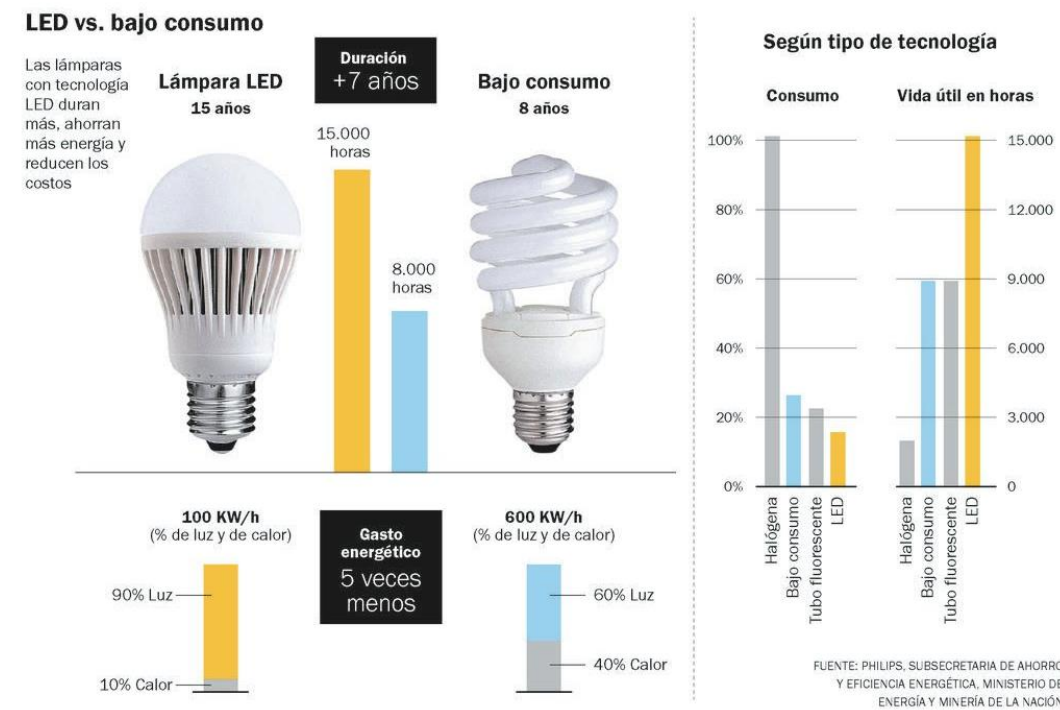


**Figura 3.6** Diagrama circuital de la conexión entre el Arduino NANO y el ESP-01  
(Fuente: Elaboración propia)

Cabe mencionar, que para hallar los valores de las resistencias se tuvieron que aplicar los conocimientos de diseño adquiridos en la universidad.

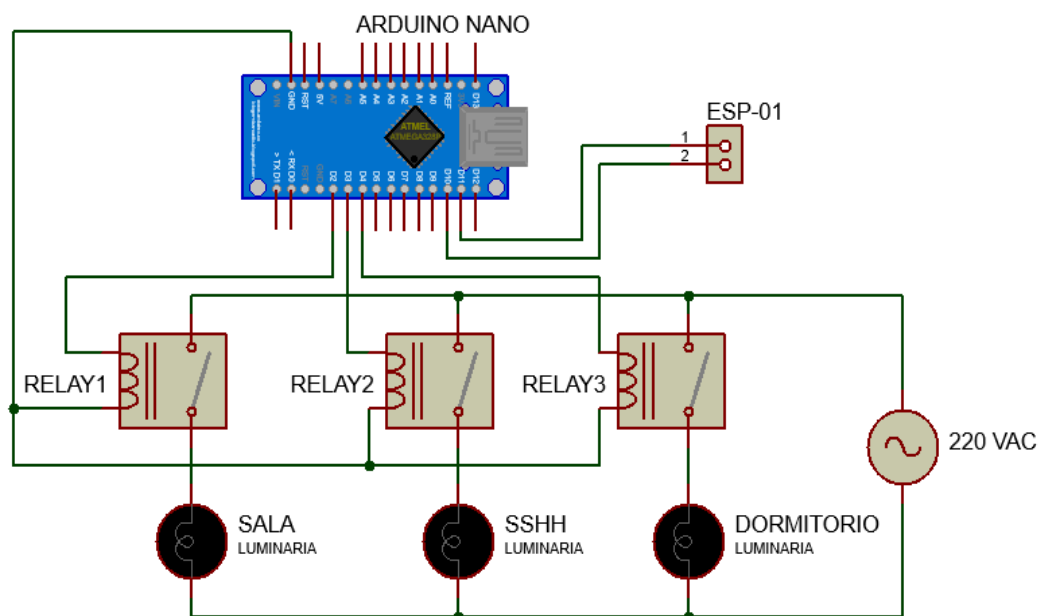
### 3.6.2. Diseño de circuito de control de las luminarias

Los tres ambientes escogidos, cuya iluminación será controlada por el usuario, son: la sala de estar, el baño y un dormitorio. Para lograr lo propuesto se hace uso de tres módulos relés 5VDC controlados por un único microcontrolador Arduino NANO. Para este caso se han considerado las luminarias del tipo fluorescente circular porque representan un consumo mayor de energía y se busca examinar el peor de los casos. Dichas luminarias consumen de 32W (una cantidad muy mayor si se compara con el consumo de las luminarias LED, ver Figura 3.7). Aún en el caso mencionado (con el uso de fluorescentes) el módulo relé a usar será útil, ya que la corriente no superará los 10A (corriente máxima soportada por el módulo relé 5VDC).



**Figura 3.7** Diferencias entre luminaria LED y de bajo consumo (ahorradores)  
(Fuente: Grupo La Provincia [67])

La Figura 3.8 muestra el diagrama circuital diseñado para el control de las luminarias.

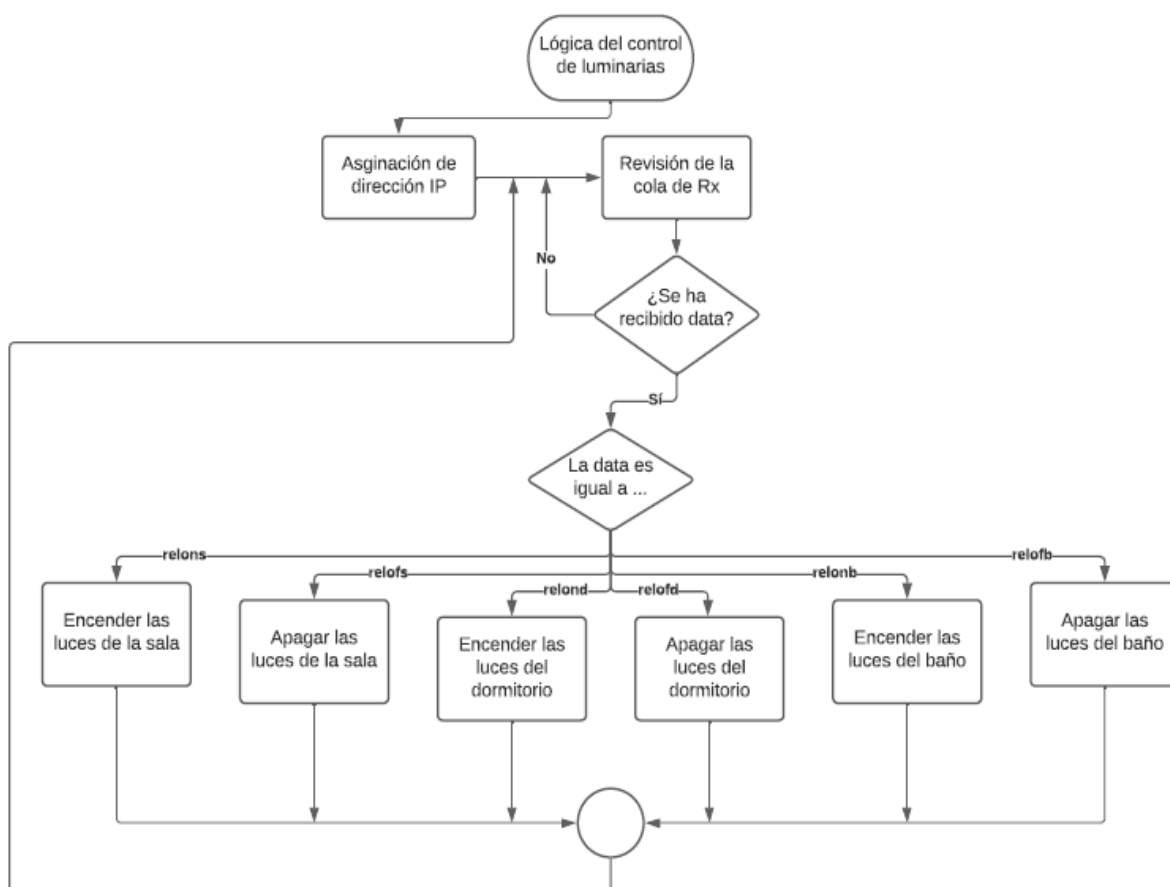


**Figura 3.8** Diagrama circuital para el control de las luminarias  
(Fuente: Elaboración propia)

Cabe aclarar que la conexión entre el Arduino NANO y el ESP-01 no es la mostrada en la Figura 3.8, sino la mostrada en la Figura 3.6. La conexión que se encuentra en la gráfica

sólo busca indicar que la transmisión y recepción de data por parte del Arduino NANO se dará a través de sus pines D11 y D12.

El funcionamiento del circuito graficado en la Figura 3.8 se da de la siguiente manera: el módulo ESP-01 recibirá una palabra clave proveniente vía Wi-Fi del Raspberry Pi y las retransmitirá al Arduino NANO para que las interprete y envíe las señales correspondientes a los actuadores (relés). La función de estos últimos será la de activar y/o desactivar las luminarias. La lógica para el control de las luminarias se presentan en la Figura 3.9 y las palabras claves y sus funciones se describen en la Tabla 3.3, mientras que el script en arduino utilizado se podrá visualizar en la sección ANEXOS (Anexo D):



**Figura 3.9** Diagrama de flujo de la lógica para el control de las luminarias  
(Fuente: Elaboración propia)

La secuencia lógica para el control de luminarias será la misma que se empleará para el control de persianas y el de la terma, motivo por el cual se obviarán los diagramas de flujo en los dos casos siguientes.

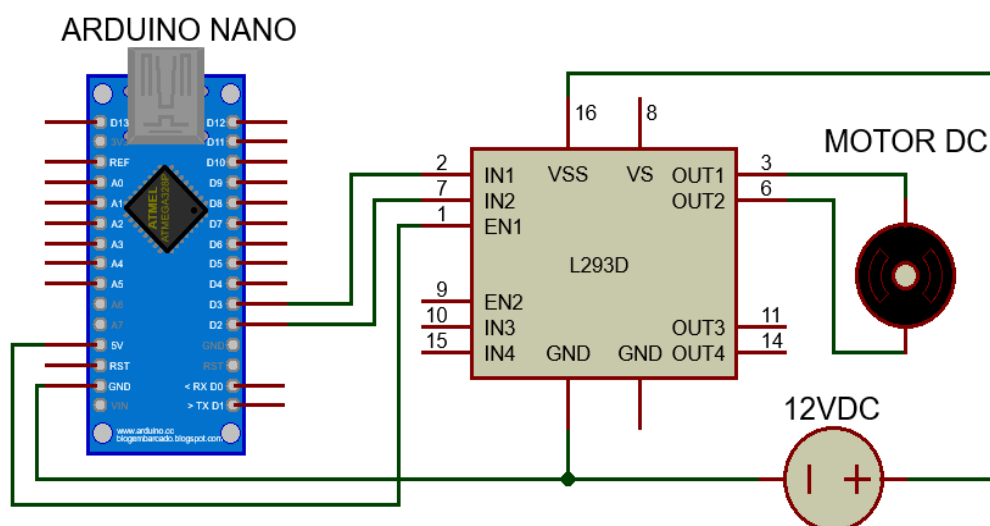
**Tabla 3.3** Comandos interpretados por el Arduino NANO

Palabra clave recibida	Función
“relons”	<b>ENCIENDE</b> la luminaria de la sala
“relofs”	<b>APAGA</b> la luminaria de la sala
“relonb”	<b>ENCIENDE</b> la luminaria del baño
“relofb”	<b>APAGA</b> la luminaria del baño
“relond”	<b>ENCIENDE</b> la luminaria del dormitorio
“relofd”	<b>APAGA</b> la luminaria del dormitorio
Cualquier otra palabra	No realiza acción alguna

(Fuente: Elaboración propia)

**3.6.3. Diseño de circuito de control de las persianas**

En esta subsección se explicará el diseño de control de las persianas, que comúnmente se encuentran instaladas en las ventanas y/o mamparas del hogar. Debido a que dichas persianas requieren de un usuario que enrolle la soguilla en un sentido para abrirlas y en el otro para cerrarlas, se hará uso de un motor DC controlado por un circuito integrado L293D. Con ello se podrá girar el motor en sentido horario o anti horario. El circuito integrado en mención se encarga de responder a las señales que son enviadas por el microcontrolador para accionar su giro en el sentido correspondiente. Para lograrlo necesita de dos fuentes de alimentación: una para la parte lógica y otra para la parte de potencia. En vista a ello, se le brinda una alimentación con el microcontrolador Arduino (5V) y otra con una fuente externa de 12V. Los pormenores de su funcionamiento se pueden volver a encontrar en la sección 2.6.2.

**Figura 3.10** Diagrama circuital para el control de las persianas

(Fuente: Elaboración propia)

Cabe aclarar que no se graficó el módulo Wi-Fi ESP-01 en la Figura 3.10 para no saturarla. Considerar que tiene la misma ubicación y configuración que en la Figura 3.8.

En esta ocasión, el Arduino NANO recibirá la orden desde el controlador central para abrir o cerrar la persiana y según la palabra clave recibida, enviará la señal correspondiente al L293D para que accione el motor DC en el sentido deseado. Las palabras claves y sus funciones se describen en la Tabla 3.4, mientras que el script en arduino utilizado se podrá visualizar en la sección ANEXOS (Anexo E):

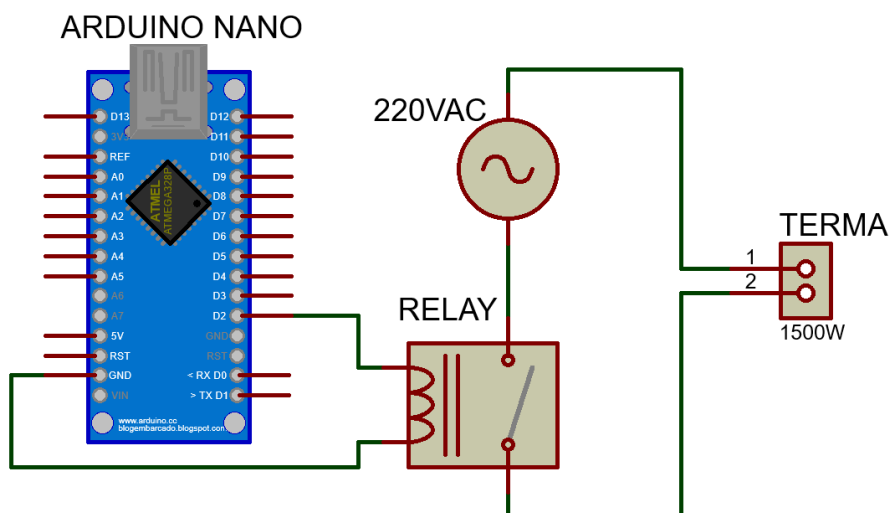
**Tabla 3.4** Comandos interpretados por el Arduino NANO

Palabra clave recibida	Función
“perons”	<b>ABRE</b> las persianas de la sala
“perofs”	<b>CIERRA</b> las persianas de la sala
“perond”	<b>ABRE</b> las persianas del dormitorio
“perofd”	<b>CIERRA</b> las persianas del dormitorio
Cualquier otra palabra	No realiza acción alguna

(Fuente: Elaboración propia)

#### 3.6.4. Diseño de circuito de control de la terma eléctrica

El control de la terma eléctrica requerirá de un microcontrolador Arduino NANO y un módulo relé 5VDC, como los que se usaron en el primer caso. Una terma eléctrica consume alrededor de 1500W, por lo tanto, la corriente no superará los 10A, que es el máximo soportado por el módulo relé a utilizar. Es así que se presenta a continuación el diagrama circuital de control de la terma eléctrica (ver Figura 3.11).



**Figura 3.11** Diagrama circuital para el control de la terma eléctrica  
(Fuente: Elaboración propia)

Es preciso aclarar que no se graficó el módulo Wi-Fi ESP-01 en la Figura 3.11 para no sobrecargar el diagrama. Considerar que tiene la misma ubicación y configuración que en la Figura 3.8.

Del diagrama circuital de la Figura 3.11 se puede visualizar que la terma estará encendida siempre y cuando el relé se lo permita. A su vez, dicho relé será controlado por el Arduino NANO, encargado de enviar la señal de activación o desactivación según se requiera; de esta manera, se tendrá un control manual sobre la terma. Adicionalmente, el Arduino está programado para apagar automáticamente la terma transcurrido un tiempo prudente para el calentamiento del agua. El periodo con el que se ha trabajado es de 10 segundos por su practicidad, sin embargo, se puede escoger el periodo al gusto del usuario. En estos casos, se busca que la automatización del hogar libere al usuario de ciertas responsabilidades, y a su vez, genere un ahorro en cuanto al consumo de energía. Las palabras claves y sus funciones se describen en la Tabla 3.5, mientras que el script en arduino utilizado se podrá visualizar en la sección ANEXOS (Anexo F):

**Tabla 3.5** Comandos interpretados por el Arduino NANO

Palabra clave recibida	Función
"termon"	<b>ENCIENDE</b> la terma
"termof"	<b>APAGA</b> la terma
Cualquier otra palabra	No realiza acción alguna

(Fuente: Elaboración propia)

## CAPÍTULO IV

### EVALUACIÓN Y ANÁLISIS DE RESULTADOS

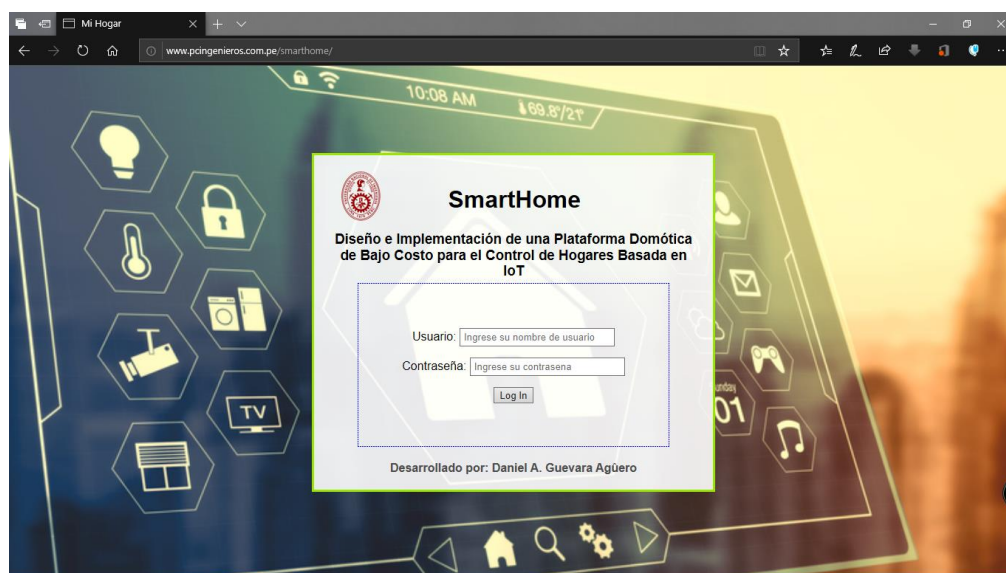
#### 4.1. Introducción

En este último capítulo se presentará el trabajo finalizado mediante capturas de pantalla de cada ventana de la interfaz web y fotos de cada elemento del sistema domótico; asimismo, se realizará una evaluación y análisis de los resultados obtenidos, como por ejemplo, la complejidad en la implementación, la facilidad de uso de la plataforma (mediante la realización de una encuesta) y el costo que representa la adquisición del sistema por parte de un usuario.

#### 4.2. Presentación ilustrativa de los resultados

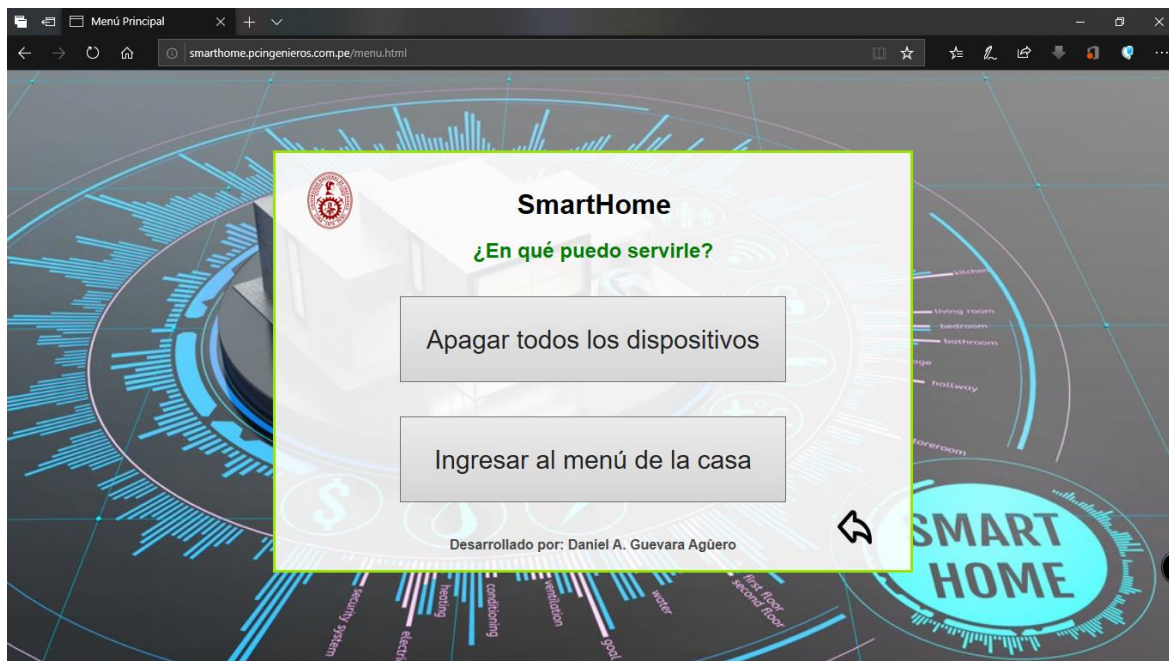
A continuación se presentarán resultados gráficos del trabajo finalizado. Se iniciará con capturas de pantalla de la interfaz web, seguido de los circuitos de control y finalmente, los actuadores con sus dispositivos a controlar. Asimismo, a lo largo de la sección se harán diversas acotaciones para aclarar detalles sobre cada elemento.

Empezando con la exhibición de imágenes, se tienen todas las ventanas de la plataforma web, comprendidas desde la Figura 4.1 hasta la Figura 4.7:

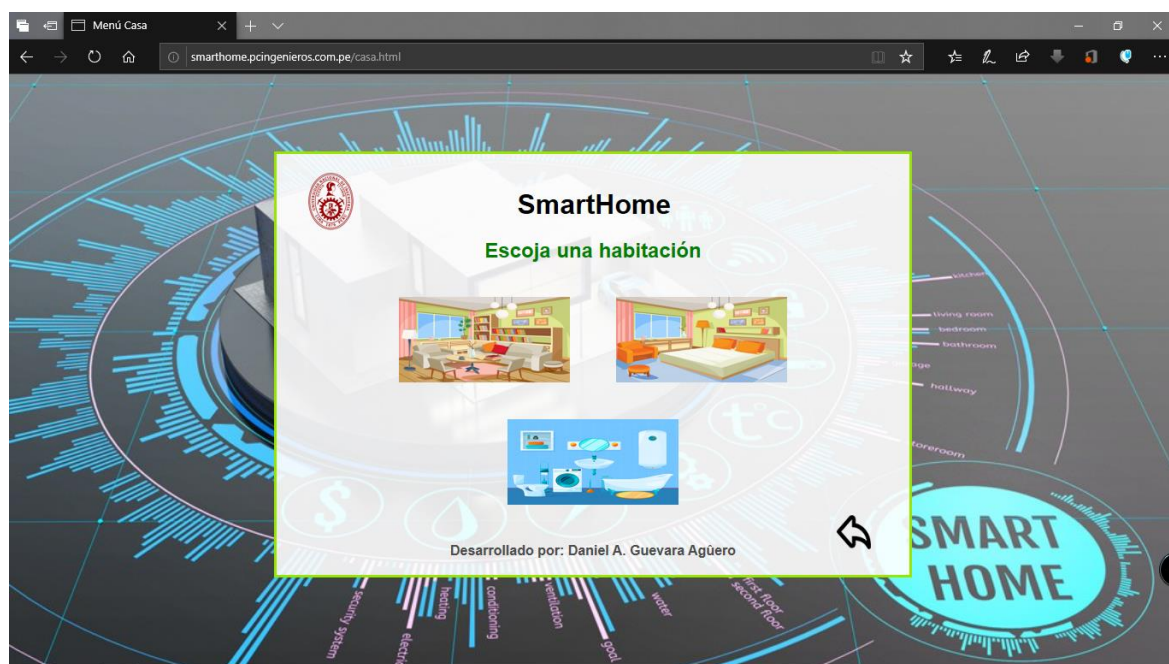


**Figura 4.1** Captura de pantalla de la página "index.php" de la interfaz web  
(Fuente: Elaboración propia)

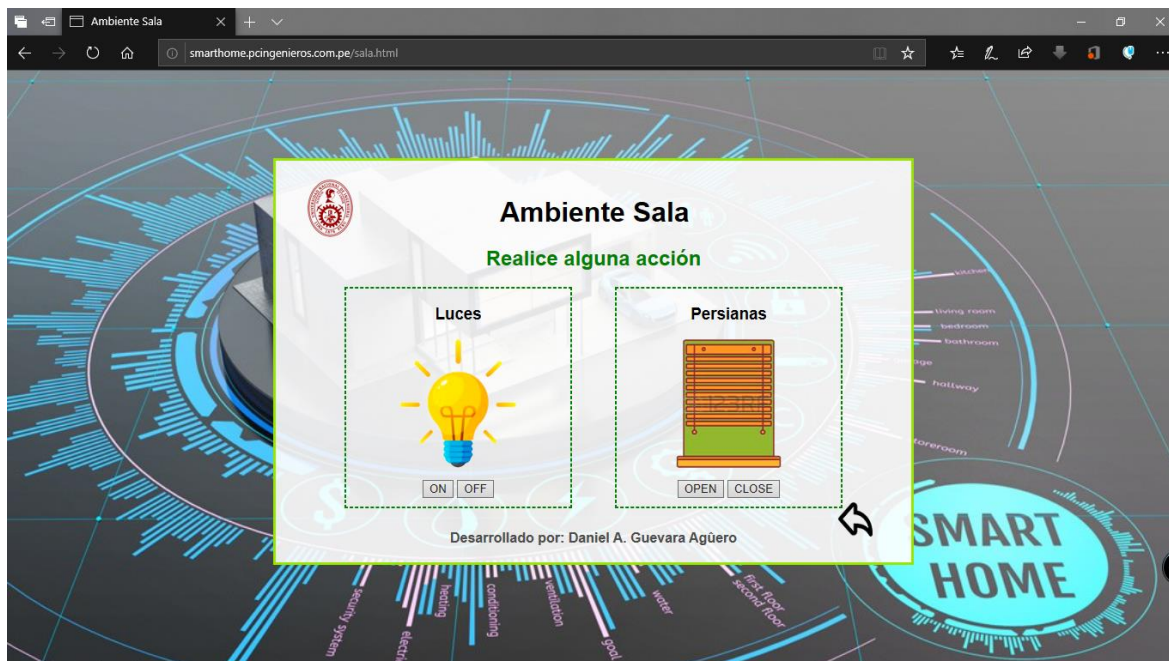




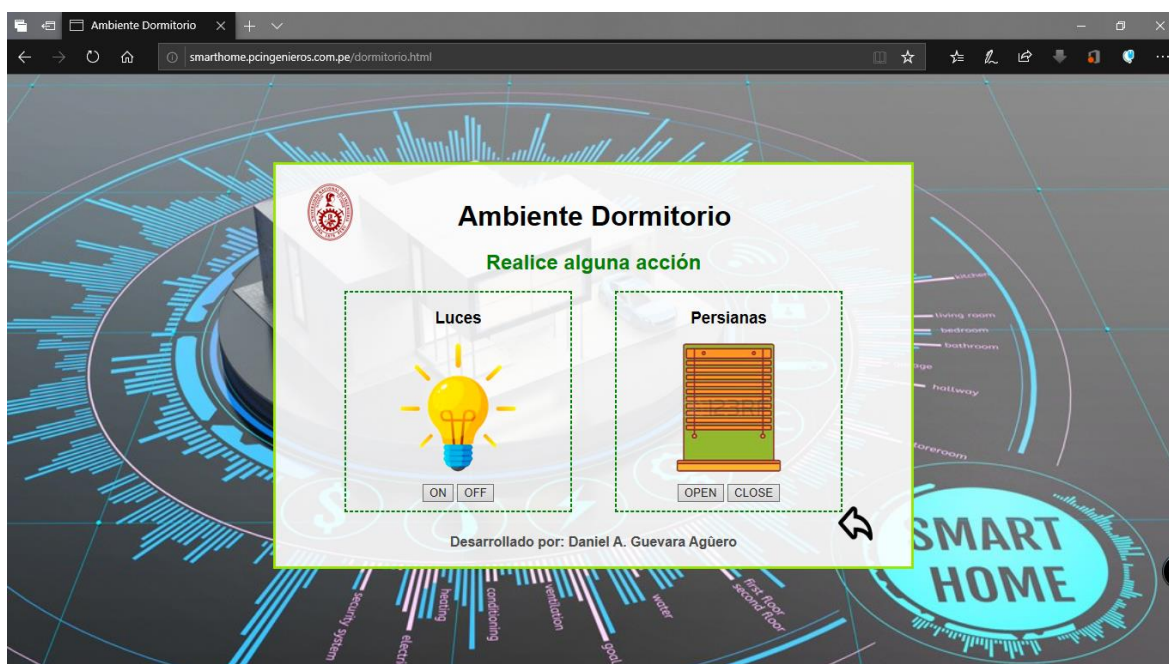
**Figura 4.2** Captura de pantalla de la página "menu.html" de la interfaz web  
(Fuente: Elaboración propia)



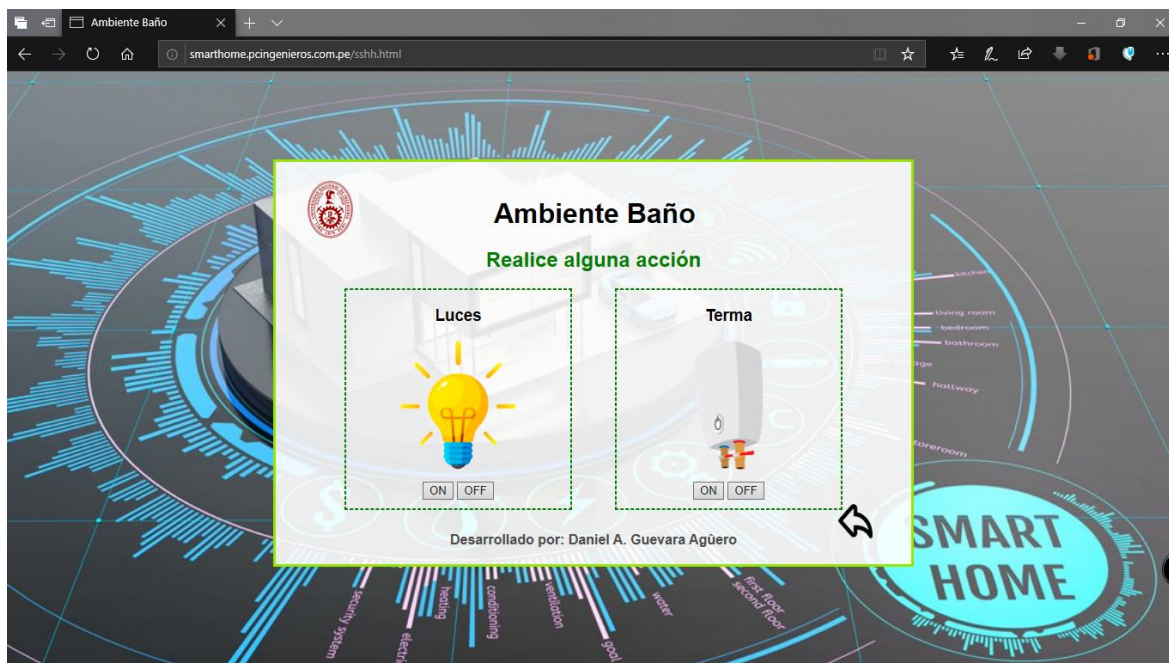
**Figura 4.3** Captura de pantalla de la página "casa.html" de la interfaz web  
(Fuente: Elaboración propia)



**Figura 4.4** Captura de pantalla de la página "sala.html" de la interfaz web  
(Fuente: Elaboración propia)



**Figura 4.5** Captura de pantalla de la página "dormitorio.html" de la interfaz web  
(Fuente: Elaboración propia)



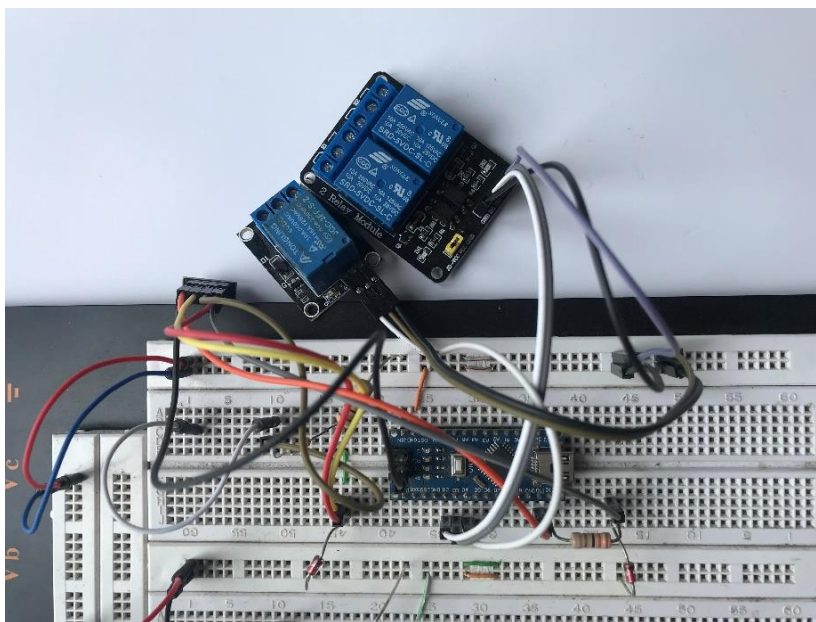
**Figura 4.6** Captura de pantalla de la página "ssh.html" de la interfaz web  
(Fuente: Elaboración propia)



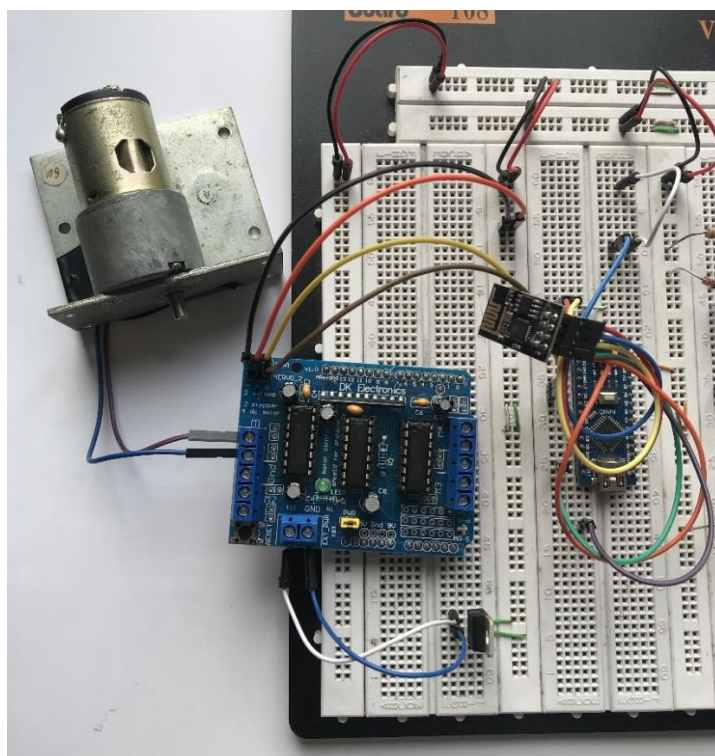
**Figura 4.7** Captura de pantalla de la página index.php vista desde un dispositivo móvil con sistema operativo Android  
(Fuente: Elaboración propia)

La Figura 4.7 muestra la captura de pantalla del aplicativo móvil, ejecutado desde un celular Motorola G4 con OS Android.

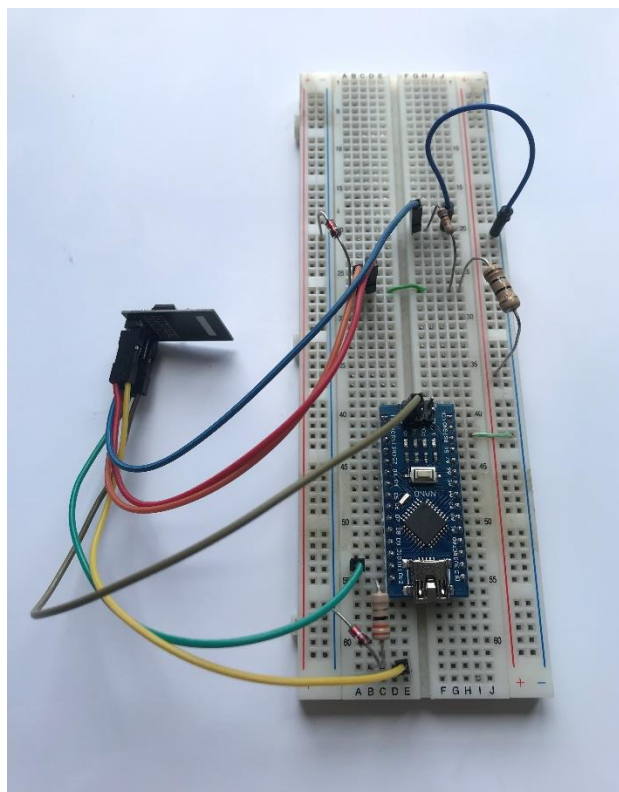
Continuando con las imágenes, desde la Figura 4.8 hasta la Figura 4.10, se mostrarán los circuitos de control utilizados en todo el sistema con sus respectivos actuadores:



**Figura 4.8** Circuito de control de luminarias (Arduino NANO + ESP-01 + 03 Relés)  
(Fuente: Elaboración propia)



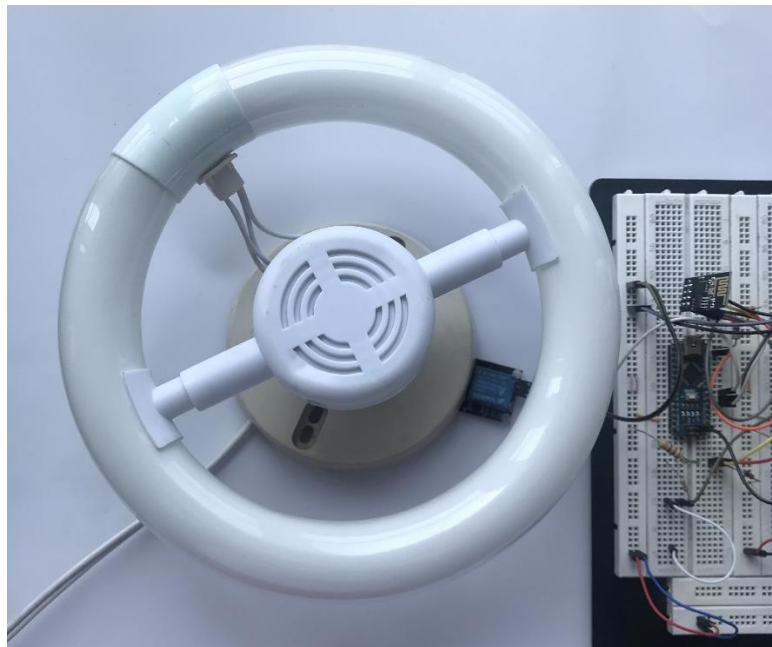
**Figura 4.9** Circuito de control de persianas (Arduino NANO + ESP-01 + Motor DC y controlador)  
(Fuente: Elaboración propia)



**Figura 4.10** Circuito de control de la terma (Arduino NANO + ESP-01)  
(Fuente: Elaboración propia)

La terma que se usa en el presente trabajo cuenta con un panel de control propio, motivo por el cual el circuito de control de la terma (Figura 4.10) no requiere de actuadores especiales como sí los requieren las persianas y las luminarias (motor DC y relés, respectivamente). Debido a ello, el circuito de control cuenta únicamente con el microcontrolador y su módulo de comunicación inalámbrico.

Finalmente, se mostrarán los dispositivos con sus respectivos circuitos de control instalados. Se ha hecho uso de una terma eléctrica de la marca Sole y de una luminaria del tipo fluorescente ahorrador, las cuales se aprecian en las Figura 4.11 y 4.12.



**Figura 4.11** Luminaria con su circuito de control instalado  
(Fuente: Elaboración propia)



**Figura 4.12** Terma con su circuito de control instalado  
(Fuente: Elaboración propia)

### **4.3. Análisis de resultados**

Esta sección está destinada al análisis de todos los resultados obtenidos a lo largo del trabajo; como se mencionó en la introducción de este capítulo, se verán tres aspectos principales del sistema: la complejidad en la implementación, la facilidad de uso de la plataforma y el bajo costo del mismo.

#### **4.3.1. Complejidad en la implementación**

Para analizar qué tan complejo resulta implementar el sistema trabajado, se tendrá que evaluar el mismo dividido en dos partes complementarias e independientes: el software y el hardware.

El software utilizado tanto en el desarrollo de la plataforma, como en la programación del hardware comprende lenguajes de programación de alto nivel, es decir, que no necesitan definir ciertos aspectos del hardware más complejos de los habituales (como los registros, direcciones de memoria, sincronización de *clocks*, etc.) para crear un programa.

Por otro lado, el hardware utilizado en la implementación del sistema, fue seleccionado cuidadosamente para que brinde todas las facilidades posibles al configurarlo. Es debido a ello que encontramos un Arduino NANO y un Raspberry Pi dentro de los elementos principales implementados. Ambos presentan una interfaz sencilla de utilizar y su lenguaje de programación, como se mencionó anteriormente, es de alto nivel y se puede aprender rápidamente los aspectos más básicos. Como detalle adicional se tiene también, que tanto el Arduino como el Raspberry Pi, son empleados como material didáctico en algunos colegios (sin conocimientos de electrónica) y universidades, lo que puede demostrar que no requiere de conocimientos especializados para su uso.

#### **4.3.2. Facilidad de uso de la plataforma**

En vista que la facilidad de uso de la interfaz diseñada es un aspecto subjetivo, se ha realizado una encuesta a un grupo de potenciales usuarios del sistema doméstico trabajado. Las preguntas realizadas se han enfocado en el conocimiento de los encuestados sobre el término “domótica”, sobre su interés en adquirir un sistema doméstico en sus hogares, sobre la facilidad de uso que ellos encuentran en la interfaz web y sobre el rango de precios que ellos consideren justo para el sistema descrito.

Para poder realizar la encuesta, se ha hecho uso de una plataforma muy conocida, Google Forms [68]. Gracias al uso de esta plataforma, se pudo difundir la encuesta sin mayores dificultades, se brindó una mayor seguridad en cuanto a la confidencialidad de las respuestas y se pudo obtener un análisis estadístico confiable con facilidad.

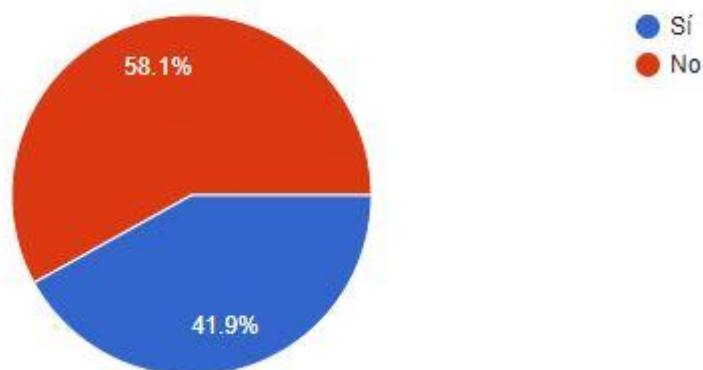
A continuación, los datos de la encuesta realizada:

- Población: 93 personas
- Perfil de los participantes: Hombres y mujeres sin un nivel de estudio específico
- Rango de las edades de los participantes: Entre 15 y 56 años
- Tipo de encuesta: Online
- Fecha de realización de la encuesta: Entre el 26 de octubre y el 3 de noviembre

Además, se presentan gráficos circulares que representan las respuestas de toda la población encuestada desde la Figura 4.13 a la Figura 4.16:

¿Conoce o alguna vez ha escuchado hablar del término "Domótica"?

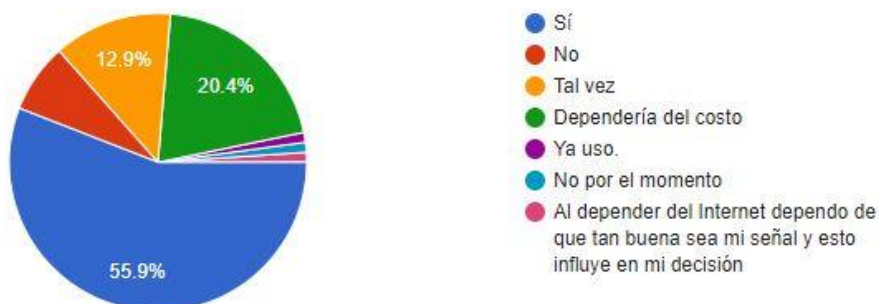
93 respuestas



**Figura 4.13** Conocimiento del término "Domótica"  
(Fuente: Google Forms [68])

¿Le interesaría adquirir un sistema que le permita controlar ciertos artefactos de su hogar (luminarias, persianas, terma, etc) desde cualquier parte del mundo?

93 respuestas

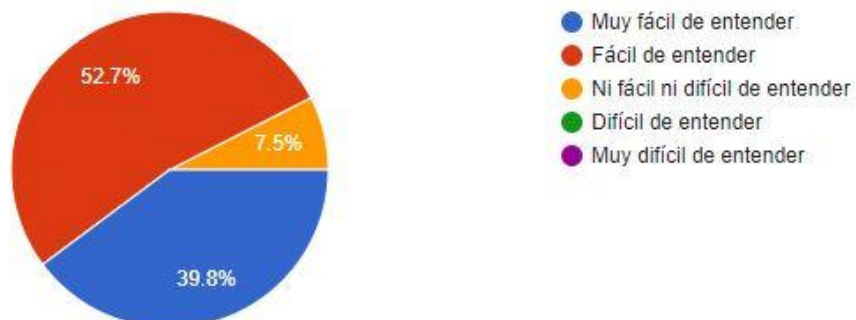


**Figura 4.14** Interés de adquisición del sistema  
(Fuente: Google Forms [68])



¿Qué tan entendible encuentra la siguiente interfaz web?

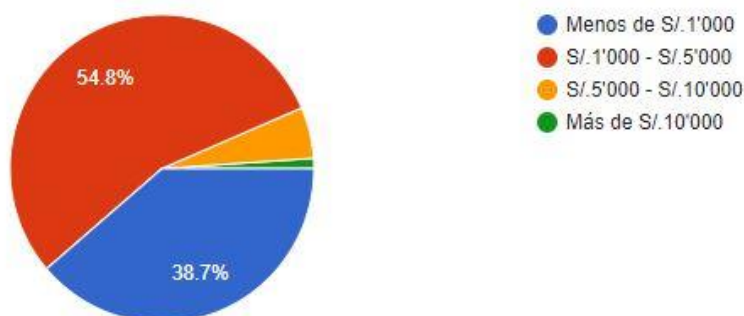
93 respuestas



**Figura 4.15** Facilidad de uso de la interfaz web  
(Fuente: Google Forms [68])

Cuánto considera usted que sería un precio justo y razonable por el sistema mencionado en la pregunta anterior. Por favor considere una vivienda con sala, comedor, cocina, baño y 1 dormitorio.

93 respuestas



**Figura 4.16** Precio justo del sistema domótico en una vivienda brevemente descrita  
(Fuente: Google Forms [68])

Tras visualizar las dos últimas figuras, se tiene certeza que la mayoría de personas encuentran la interfaz web fácil de usar (algunos la encuentran muy fácil). Asimismo, otra mayoría de personas consideran que un precio justo por el sistema descrito se encuentra en el rango entre S/.1'000 y S/.5'000 (algunos consideran que debería ser menor a S/.1'000), lo que comprueba que el sistema domótico desarrollado en la presente tesis cumple con dos de las características anunciadas inicialmente: bajo costo y facilidad de uso de la interfaz web. En el siguiente punto se verán los gastos necesarios para la

implementación del sistema y se podrá verificar si el costo del sistema domótico puede encontrarse dentro del rango de precio justo que determinó la mayoría de encuestados.

#### **4.3.3. Bajo costo del sistema**

En este punto se presentarán los gastos realizados para la implementación del sistema domótico; asimismo, se cotizará una instalación ficticia en 3 habitaciones (sala, baño y dormitorio) de un departamento promedio. De esta manera, podremos comparar y sacar conclusiones de forma más acertada.

Empezando con los gastos para la implementación del sistema, se tienen:

##### **a) Costos de los servicios almacenados en la nube**

Como parte de los servicios almacenados en la nube se tienen al servidor web y la base de datos virtual. Para la realización de este trabajo, dichos gastos fueron asumidos por la empresa PC Ingenieros E.I.R.L., quienes nos informaron que dicho valor era de aproximadamente S/.15.00 mensuales.

##### **b) Costos del controlador central**

Como se mencionó anteriormente, el controlador central está conformado principalmente por el Raspberry Pi y sus complementos para un buen funcionamiento.

Enlistando el total de elementos utilizados, se tiene lo siguiente:

- Raspberry Pi 2 Modelo B (S/.125.00)
- Cargador de 5V/1A (S/.15.00)
- Memoria MicroSD 16GB (S/.20.00)
- Antena WiFi (S/.25.00)

##### **c) Costos de los microcontroladores**

En el caso de los microcontroladores se encuentran los siguientes componentes:

- Arduino NANO (S/.18.00)
- ESP-01 (S/.15.00)
- Circuito de conexión (S/.5.00)
- Protoboard (S/.7.00)

##### **d) Costos de los actuadores**

Para los actuadores, los precios varían dependiendo de la habitación y aparatos a controlar, y los valores que pueden tener son los siguientes:

- Módulo Relé 5VDC NO (S/.3.50)
- Motor DC (S/.20.00)
- L293D (S/.4.00)
- Circuito de conexión (S/.2.50)

Finalmente, se realiza la cotización para la instalación del sistema domótico en un departamento de tamaño promedio. Los dispositivos y habitaciones a controlar serán los mismos que se han considerado durante todo el trabajo: las luminarias de 3 habitaciones, una terma y unas persianas.

Para visualizarla de una mejor manera, se presenta la Tabla 4.1:

**Tabla 4.1** Cotización de la implementación del sistema domótico

COMPONENTE	COSTO UNITARIO	CANTIDAD	SUBTOTAL
Servidor Web y Base de Datos	S/.15.00	1	S/.15.00
Raspberry Pi 2 Modelo B	S/.125.00	1	S/.125.00
Cargador de 5V/1 <sup>a</sup>	S/.15.00	2	S/.30.00
Memoria MicroSD 16GB	S/.20.00	1	S/.20.00
Antena WiFi	S/.25.00	1	S/.25.00
Arduino NANO	S/.18.00	3	S/.54.00
ESP-01	S/.15.00	3	S/.45.00
Circuito de conexión	S/.7.50	1	S/.7.50
Protoboard	S/.7.00	3	S/.21.00
Módulo Relé 5VDC NO	S/.3.50	5	S/.17.50
Motor DC	S/.20.00	1	S/.20.00
L293D	S/.4.00	1	S/.4.00
<b>TOTAL</b>			<b>S/.384.00</b>

(Fuente: Elaboración propia)

De la tabla se puede apreciar que los costos de los materiales son relativamente bajos, y luego de añadirle los costos de instalación, se sigue manteniendo en un rango de precio accesible para un hogar con ingresos que se encuentren en el promedio. Cabe mencionar que según la Figura 4.16, la mayoría de las personas consideran que el precio justo por el sistema domótico descrito se encuentra en el rango entre S/.1'000 y S/.5'000, lo que claramente es posible de cumplir con los datos encontrados en la Tabla 4.1.

Como último aspecto importante a considerar en la instalación del sistema domótico en una vivienda, se encuentra el tiempo de ejecución, éste comprende desde el reconocimiento del ambiente a automatizar hasta la verificación del correcto funcionamiento del sistema mencionado. Con la finalidad de brindar una explicación gráfica al respecto, se presenta la Tabla 4.2.

**Tabla 4.2** Cronograma de instalación del sistema domótico

Actividad	Día 1	Día 2	Día 3	Día 4	Día 5
Solicitud del sistema domótico					
Reconocimiento y Análisis del ambiente					
Compra de componentes					
Programación de los componentes					
Instalación de los componentes en la vivienda					
Verificación de un buen funcionamiento					

(Fuente: Elaboración propia)

Como se puede apreciar en la Tabla 4.2, se requiere menos de una semana para poder adquirir e instalar el sistema domótico. Cabe mencionar que el tiempo correspondiente a la “Instalación de los componentes en la vivienda” podría variar dependiendo de la cantidad de artefactos domésticos que se quieran automatizar.

## CONCLUSIONES

Tras finalizar la presente tesis, se obtuvieron las siguientes conclusiones:

- Se diseñó una plataforma intuitiva, entendible para todo tipo de usuario, que no requiere de instrucción previa para la manipulación de los artefactos domésticos vinculados.
- Se logró implementar un sistema domótico de bajo costo, confiable y escalable, que no representa un gasto considerable dentro de la economía de una familia de clase media.
- Se consiguió controlar los artefactos comunes de una vivienda, desde locaciones remotas mediante dispositivos que cuenten con acceso a internet.

## RECOMENDACIONES

El prototipo presentado en este proyecto no está diseñado para su comercialización, en vista que no se han considerado las normas técnicas al momento de su planificación. Para poder llegar a tal punto, se brindan las siguientes recomendaciones:

- La plataforma web y las comunicaciones inalámbricas del sistema no cuentan con los protocolos de seguridad mínimos para garantizar la invulnerabilidad del sistema domótico, lo que deja indefenso al usuario ante cualquier virus y/o hacker que intente controlar ilegalmente los artefactos de un hogar ajeno. Para conseguir un mejor sistema de protección, se requieren conocimientos muy especializados que se escapan del propósito de este trabajo.
- Las conexiones hechas en cada dispositivo de control, han sido realizadas en un protoboard, las cuales son útiles y suficientes para el propósito de la presente tesis. Para mejorar las conexiones, se recomienda diseñar PCBs especiales (hechas a la medida) para cada conexión entre elementos.
- El presente diseño no contempla los casos en los que haya pérdida de paquetes, por lo que de haber, puede presentar problemas y necesitar un reinicio de los componentes para retomar su buen funcionamiento. Se recomienda implementar en el programa un reinicio automático en caso de pérdida de paquetes en la información.

## **ANEXOS**

**ANEXO A GLOSARIO DE TÉRMINOS E ÍNDICE DE FIGURAS Y TABLAS****GLOSARIO DE TÉRMINOS**

A	Amperios
AC	Alternating Current
ADC	Analog-to-Digital Converter
AMQP	Advanced Message Queuing Protocol
CoAP	Constrained Application Protocol
CSI	Camera Serial Interface
CSS	Cascading Style Sheets
DC	Direct Current
DNS	Domain Name System
DSI	Display Serial Interface
ES	Embedded System
EUR	Euros
g	Gramos
GB	Gigabyte
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDMI	High-Definition Multimedia Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IC	Integrated Circuit
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
IoT	Internet de las cosas
IP	Internet Protocol
ISO	International Organization for Standardization
KB	Kilobyte



KNX	Abreviación de Konnex
LON	Local Operating Network
LTS	Long Term Support
M2M	Machine to Machine
Mbits/s	Megabits por segundo
MCU	Micro Controller Unit
MHz	Mega Hertz
MIT	Massachusetts Institute of Technology
mm	Milímetros
MQTT	Message Queue Telemetry Transport
NC	Normaly Close
NOOBS	New Out Of Box Software
NO	Normaly Open
OS	Operating System
PC	Personal Computer
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor
RAE	Real Academia Española
RAM	Random Access Memory
RMV	Remuneración Mínima Vital
ROM	Read Only Memory
SD	Secure Digital
SoC	System on Chip
SSH	Secure Shell
TV	Televisor
µP	Microprocesador
URL	Uniform Resource Locator
USB	Universal Serial Bus
USD	United States Dollar(s)
V	Voltios
Wi-Fi	Wireless Fidelity
VNC	Virtual Network Computing
XMPP	Extensible Messaging and Presence Protocol

## ÍNDICE DE FIGURAS

<b>Figura 2.1</b> Imagen ilustrativa del concepto de Domótica -----	12
<b>Figura 2.2</b> Imagen ilustrativa del concepto de IoT -----	13
<b>Figura 2.3</b> Arduino NANO -----	16
<b>Figura 2.4</b> Mapa de pines del Arduino NANO-----	16
<b>Figura 2.5</b> Funcionamiento del Protocolo SSH -----	18
<b>Figura 2.6</b> Ejemplo de Interfaz gráfica del Raspberry Pi, vista desde un iPhone -----	19
<b>Figura 2.7</b> Raspberry Pi 2 Modelo B-----	19
<b>Figura 2.8</b> Estructura circuital de un relé -----	20
<b>Figura 2.9</b> Estructura interna de un relé -----	21
<b>Figura 2.10</b> Módulo de relé para Arduino -----	22
<b>Figura 2.11</b> Constitución de un motor DC-----	23
<b>Figura 2.12</b> Puente H con transistores NPN sin la lógica de control -----	24
<b>Figura 2.13</b> Motor accionado con sentido de giro horario -----	25
<b>Figura 2.14</b> Motor con sentido de giro anti horario -----	25
<b>Figura 2.15</b> Diagrama de conexión del IC L293D con 3 motores -----	26
<b>Figura 2.16</b> Distribución de pines del circuito integrado L293D-----	27
<b>Figura 2.17</b> Módulo Wi-Fi ESP-01-----	27
<b>Figura 2.18</b> Mapa de pines del módulo Wi-Fi ESP-01-----	28
<b>Figura 2.19</b> Distribución de pines del ESP8266EX-----	30
<b>Figura 2.20</b> Conversor USB a Serial TTL (Módulo PL2303)-----	31
<b>Figura 2.21</b> Ventana inicial del Flash Download Tools -----	32
<b>Figura 2.22</b> Ventana de configuración del firmware y sus ubicaciones -----	32
<b>Figura 2.23</b> Interfaz gráfica de phpMyAdmin-----	34
<b>Figura 3.1</b> Recorrido de las instrucciones emitidas por el usuario -----	36
<b>Figura 3.2</b> Interfaz gráfica del <i>MIT App Inventor</i> en modo <i>Designer</i> -----	38
<b>Figura 3.3</b> Interfaz gráfica del <i>MIT App Inventor</i> en modo <i>Blocks</i> -----	39
<b>Figura 3.4</b> Tabla “dispositivos” de la base de datos “pcingeni_smarthome”-----	40
<b>Figura 3.5</b> Diagrama de flujo de la lógica del controlador central -----	42

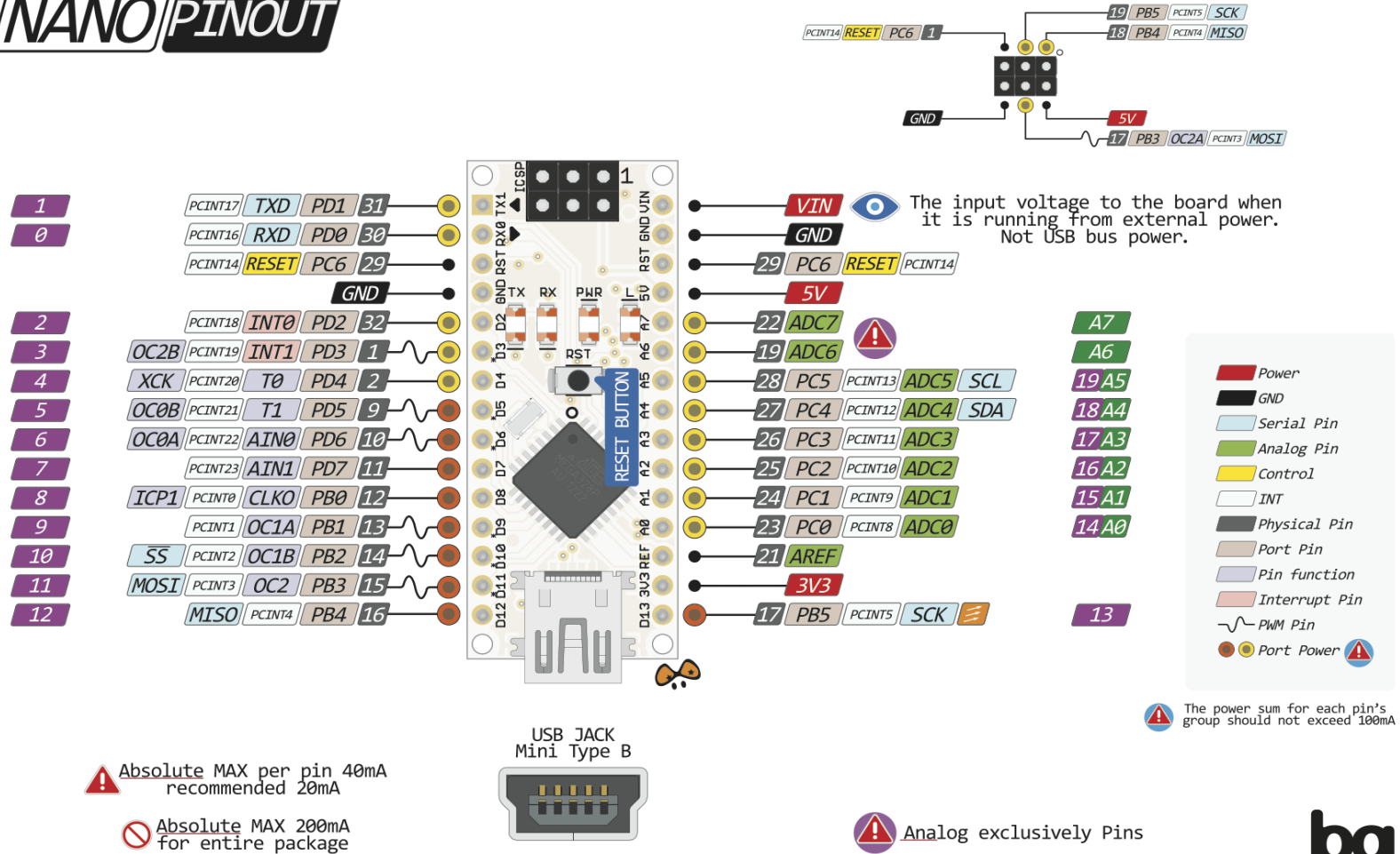
<b>Figura 3.6</b>	Diagrama circuital de la conexión entre el Arduino NANO y el ESP-01-----	45
<b>Figura 3.7</b>	Diferencias entre luminaria LED y de bajo consumo (ahorradores)-----	46
<b>Figura 3.8</b>	Diagrama circuital para el control de las luminarias -----	46
<b>Figura 3.9</b>	Diagrama de flujo de la lógica para el control de las luminarias -----	47
<b>Figura 3.10</b>	Diagrama circuital para el control de las persianas -----	48
<b>Figura 3.11</b>	Diagrama circuital para el control de la terma eléctrica -----	49
<b>Figura 4.1</b>	Captura de pantalla de la página "index.php" de la interfaz web-----	51
<b>Figura 4.2</b>	Captura de pantalla de la página "menu.html" de la interfaz web-----	52
<b>Figura 4.3</b>	Captura de pantalla de la página "casa.html" de la interfaz web-----	52
<b>Figura 4.4</b>	Captura de pantalla de la página "sala.html" de la interfaz web-----	53
<b>Figura 4.5</b>	Captura de pantalla de la página "dormitorio.html" de la interfaz web-----	53
<b>Figura 4.6</b>	Captura de pantalla de la página "ssh.html" de la interfaz web-----	54
<b>Figura 4.7</b>	Captura de pantalla de la página index.php vista desde un dispositivo móvil con sistema operativo Android-----	54
<b>Figura 4.8</b>	Circuito de control de luminarias (Arduino NANO + ESP-01 + 03 Relés) -----	55
<b>Figura 4.9</b>	Circuito de control de persianas (Arduino NANO + ESP-01 + Motor DC y controlador)-----	55
<b>Figura 4.10</b>	Circuito de control de la terma (Arduino NANO + ESP-01) -----	56
<b>Figura 4.11</b>	Luminaria con su circuito de control instalado -----	57
<b>Figura 4.12</b>	Terma con su circuito de control instalado -----	57
<b>Figura 4.13</b>	Conocimiento del término "Domótica" -----	59
<b>Figura 4.14</b>	Interés de adquisición del sistema -----	59
<b>Figura 4.15</b>	Facilidad de uso de la interfaz web-----	60
<b>Figura 4.16</b>	Precio justo del sistema domótico en una vivienda brevemente descrita-----	60

## ÍNDICE DE TABLAS

<b>Tabla 2.1</b>	Especificaciones Técnicas del Arduino NANO -----	15
<b>Tabla 2.2</b>	Descripción de pines de un Arduino NANO -----	17
<b>Tabla 2.3</b>	Especificaciones Técnicas del Raspberry Pi 2 Modelo B -----	20
<b>Tabla 2.4</b>	Descripción de los pines del L293D -----	26
<b>Tabla 2.5</b>	Especificaciones Técnicas del Módulo Wi-Fi ESP-01 -----	28
<b>Tabla 2.6</b>	Descripción de pines del Módulo WI-Fi ESP-01 -----	29
<b>Tabla 2.7</b>	Características Wi-Fi del ESP8266EX -----	29
<b>Tabla 2.8</b>	Conexiones entre los módulos ESP-01 y PL2303 -----	31
<b>Tabla 2.9</b>	Archivos del firmware y sus respectivas ubicaciones -----	33
<b>Tabla 3.1</b>	Información del servidor web -----	40
<b>Tabla 3.2</b>	Lógica del controlador central frente a las variaciones percibidas -----	43
<b>Tabla 3.3</b>	Comandos interpretados por el Arduino NANO -----	48
<b>Tabla 3.4</b>	Comandos interpretados por el Arduino NANO -----	49
<b>Tabla 3.5</b>	Comandos interpretados por el Arduino NANO -----	50
<b>Tabla 4.1</b>	Cotización de la implementación del sistema domótico -----	62
<b>Tabla 4.2</b>	Cronograma de instalación del sistema domótico -----	63

## ANEXO B DESCRIPCIÓN DEL PINADO DEL ARDUINO NANO

# NANO PINOUT



## ANEXO C SCRIPTS EN HTML, PHP Y CSS PARA LA INTERFAZ WEB

Nombre del archivo: index.php

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width:device-width">
<title>Mi Hogar</title>
<link rel="stylesheet" href="css/main.css">
</head>
<body>
<div id="contenedor">
<div class="login">

<h1>SmartHome</h1>
<h3>Diseño e Implementación de una Plataforma Domótica de Bajo
Costo para el Control de Hogares Basada en IoT</h3>
<form class="login" action="index.php" method="post">
<label for="usuario">Usuario: </label>
<input type="text" id="usuario" name="usuario"
placeholder="Ingrese su nombre de usuario" required><br>
<label for="contrasena">Contraseña: </label>
<input type="password" id="contrasena" name="contrasena"
placeholder="Ingrese su contraseña" required><br>
<input type="submit" name="login" value="Log In">
</form>
<?php
if (isset($_POST['usuario']) && isset($_POST['contrasena'])) {
$usuario = $_POST['usuario'];
$contrasena = $_POST['contrasena'];
if ($usuario == "admin" && $contrasena == "admin") {
header('Location:
http://smarhome.pcingenieros.com.pe/menu.html');
} else {
$denegado = "Usuario o contraseña incorrectos";
?>
<p style="color:red"><?=$denegado?></p>
<?=}}?>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
</div>
</div>
</body>
</html>

```

**Nombre del archivo: menú.html**

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width:device-width">
<title>Menú Principal</title>
<link rel="stylesheet" href="css/menu.css">
</head>
<body>
<div id="contenedor">
<div class="menu">

<h1>SmartHome</h1>
<h2>¿En qué puedo servirle?</h2>
<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="menu.html">
<input type="hidden" name="dispo" value="todooff">
<input type="submit" name="off" value="Apagar todo (luces,
persianas y terma)">
</form>
<a href="casa.html" target="_self"><input type="button"
name="menucasa" value="Ingresar al menú de la casa"></a>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
<a href="index.php" target="_self"></a>
</div>
</div>
</body>
</html>

```

**Nombre del archivo: casa.html**

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width:device-width">
<title>Menú Casa</title>
<link rel="stylesheet" href="css/casa.css">
</head>
<body>
<div id="contenedor">
<div class="casa">


```

```

<h1>SmartHome</h1>
<h2>Escoja una habitación</h2>
<a href="sala.html" target="_self"></a>
<a href="dormitorio.html" target="_self"></a>
<a href="ssh.html" target="_self"></a>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
<a href="menu.html" target="_self"></a>
</div>
</div>
</body>
</html>

```

**Nombre del archivo: sala.html**

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width:device-width">
<title>Ambiente Sala</title>
<link rel="stylesheet" href="css/ambiente.css">
</head>
<body>
<div id="contenedor">
<div class="ambiente">

<h1>Ambiente Sala</h1>
<h2>Realice alguna acción</h2>
<div class="controlables">
<div class="dispo1">
<h3>Luces</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="sala.html">
<input type="hidden" name="dispo" value="lsala">
<input type="submit" name="on" value="ON">
<input type="submit" name="off" value="OFF">
</form>
</div>
<div class="dispo2">

```



```

<h3>Persianas</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="sala.html">
<input type="hidden" name="dispo" value="psala">
<input type="submit" name="on" value="OPEN">
<input type="submit" name="off" value="CLOSE">
</form>
</div>
</div>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
<a href="casa.html" target="_self"></a>
</div>
</div>
</body>
</html>

```

#### Nombre del archivo: dormitorio.html

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width=device-width">
<title>Ambiente Dormitorio</title>
<link rel="stylesheet" href="css/ambiente.css">
</head>
<body>
<div id="contenedor">
<div class="ambiente">

<h1>Ambiente Dormitorio</h1>
<h2>Realice alguna acción</h2>
<div class="controlables">
<div class="dispo1">
<h3>Luces</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="dormitorio.html">
<input type="hidden" name="dispo" value="ldorm">
<input type="submit" name="on" value="ON">
<input type="submit" name="off" value="OFF">
</form>
</div>

```

```

<div class="dispo2">
<h3>Persianas</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="dormitorio.html">
<input type="hidden" name="dispo" value="pdorm">
<input type="submit" name="on" value="OPEN">
<input type="submit" name="off" value="CLOSE">
</form>
</div>
</div>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
<a href="casa.html" target="_self"></a>
</div>
</div>
</body>
</html>

```

#### Nombre del archivo: sshh.html

```

<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
<meta charset="utf-8" content="width:device-width">
<title>Ambiente Baño</title>
<link rel="stylesheet" href="css/ambiente.css">
</head>
<body>
<div id="contenedor">
<div class="ambiente">

<h1>Ambiente Baño</h1>
<h2>Realice alguna acción</h2>
<div class="controlables">
<div class="dispo1">
<h3>Luces</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="sshh.html">
<input type="hidden" name="dispo" value="lsshh">
<input type="submit" name="on" value="ON">
<input type="submit" name="off" value="OFF">
</form>

```

```

</div>
<div class="dispo2">
<h3>Terma</h3>

<form action="onoff.php" method="post">
<input type="hidden" name="ambiente" value="ssh.html">
<input type="hidden" name="dispo" value="terma">
<input type="submit" name="on" value="ON">
<input type="submit" name="off" value="OFF">
</form>
</div>
</div>
<h4>Desarrollado por: Daniel A. Guevara Agüero</h4>
<a href="casa.html" target="_self"></a>
</div>
</div>
</body>
</html>

```

#### Nombre del archivo: onoff.php

```

<?php
$server = '162.241.218.133';
$user = "pcingeni_daga";
$pw = "06071995";
$db = "pcingeni_smarthome";
$ambiente = $_POST['ambiente'];
$dispo = $_POST['dispo'];
$conexion = new mysqli($server,$user,$pw,$db);
if (!conexion){
die("Fallo la conexion".mysqli_connect_error());}
if(isset($_REQUEST['on'])){
$val = "1";
}else if(isset($_REQUEST['off'])){
$val = "2";}
$sqla = "UPDATE dispositivos SET cambio='$val' WHERE
id_dispo='$dispo'";
if($conexion->query($sqla)){
echo "Actualizado exitosamente";}else{
echo "Error actualizando".$conexion->error;}
$conexion->close();
header("Location:
http://smarthome.pcingenieros.com.pe/$ambiente");
?>

```

**Nombre del archivo: alloff.php**

```

<?php
$server = '162.241.218.133';
$user = "pcingeni_daga";
$pw = "06071995";
$db = "pcingeni_smarthome";
$ambiente = $_POST['ambiente'];
$dispo = $_POST['dispo'];
$conexion = new mysqli($server,$user,$pw,$db);
if (!conexion){
die("Fallo la conexion".mysqli_connect_error());}
if(isset($_REQUEST['loff'])){
$val = "2";}
$sqla = "UPDATE dispositivos SET cambio='$valor' WHERE
dispo='$dispo'";
if(
$conexion->query($sqla)){echo "Actualizado exitosamente";
}else{
echo "Error actualizando".$conexion->error;}
$conexion->close();
header
("Location:http://smarthome.pcingenieros.com.pe/$ambiente");
?>

```

**Nombre del archivo: main.css**

```

html, body {
    width: 100%; height: 100%;}
body {
    margin: 0; display: table;
    background-image: url(../img/background.jpg);
    background-position: center; background-size: cover;
    background-repeat: no-repeat;
    font-family: sans-serif;}
#contenedor {
    display: table-cell;
    text-align: center; vertical-align: middle;}
.login {
    width: 500px; min-height: 200px;
    background-color: rgba(255, 255, 255, 0.9);
    margin: 0 auto; padding: 20px;
    border: solid #99e600 3px;}
.login img {
    float: left; height: 70px; width: auto;
    margin-left: 15px;}

```

```

.login h1 {
    font-weight: bold;    margin-right: 68px;}
.login h3 {
    margin-top: 25px; margin-bottom: 5px;}
.login form {
    width: 400px; height: 100px; padding: 10px;
    background-color: transparent; border: dashed blue 1px;}
.login form label, input {
    margin-top: 15px;}
.login form input:first-of-type {
    margin-top: 50px;}
.login form input:not([type=submit]) {
    padding: 3px; width: 200px;}
.login h4 {
    color: #4d4d4d; margin-top: 20px; margin-bottom: 0;}

```

**Nombre del archivo: menu.css**

```

html, body {
    width: 100%; height: 100%;}
body {
    margin: 0; display: table;
    background-image: url(../img/smarthome.jpg);
    background-position: center; background-size: cover;
    background-repeat: no-repeat;
    font-family: sans-serif;}
#contenedor {
    display: table-cell;
    text-align: center; vertical-align: middle;}
.menu {
    width: 700px; min-height: 400px;
    background-color: rgba(255, 255, 255, 0.9);
    margin: 0 auto; padding: 20px;
    border: solid #99e600 3px;}
.menu img {
    float: left; height: 70px; width: auto; margin-left: 15px;}
.menu h1 {
    font-weight: bold; margin-right: 68px;}
.menu h2 {
    color: green;}
.menu form, a:not(:last-of-type) {
    display: inline-block; margin: 20px 25px;}
.menu form input, a input {
    width: 450px; height: 100px; font-size: 30px;}

```

```
.menu a img.volver {
    width: 40px; height: auto; float: right;
    position: relative; bottom: 50px; right: 25px;}
.menu h4 {
    color: #4d4d4d; margin-top: 20px; margin-bottom: 0;}
```

**Nombre del archivo: casa.css**

```
html, body {
    width: 100%; height: 100%;}
body {
    margin: 0; display: table;
    background-image: url(../img/smarthome.jpg);
    background-position: center; background-size: cover;
    background-repeat: no-repeat; font-family: sans-serif;}
#contenedor {
    display: table-cell;
    text-align: center; vertical-align: middle;}
.casa {
    width: 700px; min-height: 400px;
    background-color: rgba(255, 255, 255, 0.9);
    margin: 0 auto; padding: 20px; border: solid #99e600 3px;}
.casa img.UNI {
    float: left; height: 70px; width: auto; margin-left: 15px;}
.casa h1 {
    font-weight: bold; margin-right: 68px;}
.casa h2 {
    color: green;}
.casa a:not(:last-of-type) {
    display: inline-block; margin: 20px 25px;}
.casa a img{
    width: 200px; height: 100px; font-size: 30px;
    background-color: transparent;}
.casa a img.volver {
    width: 40px; height: auto; float: right;
    position: relative; bottom: 50px; right: 25px;}
.casa h4 {
    color: #4d4d4d; margin-top: 20px; margin-bottom: 0;}
```

**Nombre del archivo: ambiente.css**

```
html, body {
    width: 100%; height: 100%;}
body {
    margin: 0; display: table;
    background-image: url(../img/smarthome.jpg);
    background-position: center; background-size: cover;
    background-repeat: no-repeat; font-family: sans-serif;}
#contenedor {
    display: table-cell;
    text-align: center; vertical-align: middle;}
.ambiente {
    width: 700px; min-height: 400px;
    background-color: rgba(255, 255, 255, 0.9);
    margin: 0 auto; padding: 20px; border: solid #99e600 3px;}
.ambiente img.UNI {
    float: left; height: 70px; width: auto; margin-left: 15px;}
.ambiente h1 {
    font-weight: bold; margin-right: 68px;}
.ambiente h2 {
    color: green;}
.ambiente .controlables {
    width: 700px; min-width: 150px;}
.ambiente .controlables img {
    height: 150px; width: auto; margin-bottom: 10px;}
.ambiente .controlables input[type=submit] {
    display: inline; margin-bottom: 10px;}
.controlables .dispo1 {
    width: 261px; margin: 0 25px 25px 60px;
    min-height: 150px; float: left; border: dashed green 2px;}
.controlables .dispo2 {
    width: 261px; margin: 0 60px 25px 25px;
    min-height: 150px; float: right; border: dashed green 2px;}
.ambiente a img.volver {
    width: 40px; height: auto; float: right;
    position: relative; bottom: 50px; right: 25px;}
.ambiente h4 {
    color: #4d4d4d;
    margin-top: 20px; margin-bottom: 0;
    clear: both;}
```

## ANEXO D SCRIPT EN PYTHON PARA EL CONTROLADOR CENTRAL

Nombre del archivo: RaspberryPi.py

```

import time
import mysql.connector
import socket
mydb =
mysql.connector.connect(host="162.241.218.133",user="pcingeni_d
aga",passwd="06071995",port='3306', db="pcingeni_smarthome")
cur = mydb.cursor() # Creando el cursor para recorrer la db
cur.execute("UPDATE dispositivos SET cambio='0', estado='0'
WHERE 1") # Inicializando las variables
mydb.commit() # Hacer efectiva la escritura de datos
mydb.close() # Cerrando conexion con el servidor
tini = 0
tfin = 0
time.sleep(1.8)

while True:

##### CONEXIÓN A LA DB #####
mydb =
mysql.connector.connect(host="162.241.218.133",user="pcingeni_d
aga",passwd="06071995",port="3306",db="pcingeni_smarthome")
cur = mydb.cursor()
cur.execute("SELECT estado, cambio FROM dispositivos WHERE 1")
res = cur.fetchall() # res ahora contiene los datos sacados de
la db
print (res)
alloff = int(res[6][1])

##### LUCES #####
s1 = socket.socket()
host1 = "192.168.1.10"
port1 = 80

##### Lógica Luces Dormitorio #####
ldant = int(res[0][0])
lddes = int(res[0][1])
if lddes==1 and ldant==0:
s1.connect((host1,port1))
mensaje = 'relond\n'
time.sleep(0.25)
s1.send(mensaje . encode())
time.sleep(0.25)

```



```

sl.close()
print ("PRENDIENDO LDORMITORIO")
cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='ldorm'")
mydb.commit()
elif lddes==2 and ldant==1:
sl.connect((host1,port1))
mensaje = 'relofd\n'
time.sleep(0.25)
sl.send(mensaje . encode())
time.sleep(0.25)
sl.close()
print ("APAGANDO LDORMITORIO")
cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='ldorm'")
mydb.commit()
else
cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='ldorm'")
mydb.commit()

##### Lógica Luces Sala #####
lsant = int(res[1][0])
lsdes = int(res[1][1])
if lsdes==1 and lsant==0:
sl.connect((host1,port1))
mensaje = 'relons\n'
time.sleep(0.25)
sl.send(mensaje . encode())
time.sleep(0.25)
sl.close()
print ("PRENDIENDO LSALA")
cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='lsala'")
mydb.commit()
elif lsdes==2 and lsant==1:
sl.connect((host1,port1))
mensaje = 'relofs\n'
time.sleep(0.25)
sl.send(mensaje . encode())
time.sleep(0.25)
sl.close()
print ("APAGANDO LSALA")
cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='lsala'")

```

```

    mydb.commit()
else
    cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='lsala'")
    mydb.commit()

##### Lógica Luces SSHH #####
lbant = int(res[2][0])
lbdes = int(res[2][1])
if lbdes==1 and lbant==0:
    sl.connect((hostl,portl))
    mensaje = 'relonb\n'
    time.sleep(0.25)
    sl.send(mensaje . encode())
    time.sleep(0.25)
    sl.close()
    print ("PRENDIENDO LSSH")
    cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='lssh'")
    mydb.commit()
elif lbdes==2 and lbant==1:
    sl.connect((hostl,portl))
    mensaje = 'relofb\n'
    time.sleep(0.25)
    sl.send(mensaje . encode())
    time.sleep(0.25)
    sl.close()
    print ("APAGANDO LSSH")
    cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='lssh'")
    mydb.commit()
else
    cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='lssh'")
    mydb.commit()

##### PERSIANAS #####
sp = socket.socket()
hostp = "192.168.1.11"
portp = 80

##### Lógica Persianas Dormitorio #####
pdant = int(res[3][0])
pddes = int(res[3][1])
if pddes==1 and pdant==0:
    sp.connect((hostp,portp))

```

```

mensaje = 'relond\n'
time.sleep(0.25)
sp.send(mensaje . encode())
time.sleep(0.25)
sp.close()
print ("ABRIENDO PDORMITORIO")
cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='pdorm'")
mydb.commit()
elif pddes==2 and pdant==1:
    sp.connect((hostp,portp))
    mensaje = 'relofd\n'
    time.sleep(0.25)
    sp.send(mensaje . encode())
    time.sleep(0.25)
    sp.close()
    print ("CERRANDO PDORMITORIO")
    cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='pdorm'")
    mydb.commit()
else
    cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='pdorm'")
    mydb.commit()

##### Lógica Persianas Sala #####
psant = int(res[4][0])
psdes = int(res[4][1])
if psdes==1 and psant==0:
    sp.connect((hostp,portp))
    mensaje = 'relons\n'
    time.sleep(0.25)
    sp.send(mensaje . encode())
    time.sleep(0.25)
    sp.close()
    print ("ABRIENDO PSALA")
    cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='psala'")
    mydb.commit()
elif psdes==2 and psant==1:
    sp.connect((hostp,portp))
    mensaje = 'relofs\n'
    time.sleep(0.25)
    sp.send(mensaje . encode())
    time.sleep(0.25)
    sp.close()

```

```

    print ("CERRANDO PSALA")
    cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='psala'")
    mydb.commit()
else
    cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='psala'")
    mydb.commit()

##### TERMA #####
st = socket.socket()
hostt = "192.168.1.12"
portt = 80

##### Lógica TERMA #####
tant = int(res[5][0])
tdes = int(res[5][1])
ttrans = tfin - tini
if tdes==1 and tant ==0:
    st.connect((hostt,portt))
    mensaje = 'relon\n'
    time.sleep(0.25)
    st.send(mensaje . encode())
    time.sleep(0.25)
    st.close()
    print ("PRENDIENDO TERMA")
    cur.execute("UPDATE dispositivos SET estado='1', cambio='0'
WHERE id_dispo='terma' ")
    mydb.commit()
    tini = time.time()
elif tdes==2 and tant==1:
    st.connect((hostt,portt))
    mensaje = 'relof\n'
    time.sleep(0.25)
    st.send(mensaje . encode())
    time.sleep(0.25)
    st.close()
    print ("APAGANDO TERMA")
    cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='terma'")
    mydb.commit()
    tini = 0
    tfin = 0
elif tant==1:
    tfin = time.time()
else

```

```

    cur.execute("UPDATE dispositivos SET cambio='0' WHERE
id_dispo='terma'")
    mydb.commit()
    if ttrans > 10.0:
        st.connect((hostt,portt))
        mensaje = 'relof\n'
        time.sleep(0.25)
        st.send(mensaje . encode())
        time.sleep(0.25)
        st.close()
        print ("APAGANDO TERMA AUTOMATICAMENTE")
        cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE id_dispo='terma'")
        mydb.commit()
        tini = 0
        tfin = 0

##### APAGANDO TODO #####
    if alloff==2:
        sl.connect((hostl,portl))
        mensaje = 'reloft\n'
        time.sleep(0.25)
        sl.send(mensaje . encode())
        time.sleep(0.25)
        sl.close()
        sp.connect((hostp,portp))
        mensaje = 'reloft\n'
        time.sleep(0.25)
        sp.send(mensaje . encode())
        time.sleep(0.25)
        sp.close()
        st.connect((hostt,portt))
        mensaje = 'relof\n'
        time.sleep(0.25)
        st.send(mensaje . encode())
        time.sleep(0.25)
        st.close()
        print ("APAGANDO TERMA AUTOMATICAMENTE")
        cur.execute("UPDATE dispositivos SET estado='0', cambio='0'
WHERE 1")
        mydb.commit()

##### CERRANDO CONEXIÓN #####
    mydb.close()
    time.sleep(0.2)

```

## ANEXO E SCRIPT EN ARDUINO PARA EL CONTROL DE LUMINARIAS

Nombre del archivo: Luces.ino

```
#include <SoftwareSerial.h>
#define sala 2
#define dorm 3
#define sshh 4
#define RxD 11
#define TxD 12

SoftwareSerial ESP8266(RxD, TxD);
char data;
const byte numChars = 50;
char receivedChars[numChars];
boolean newData = false;

void setup(){
  ESP8266.flush(); delay(10);
  ESP8266.begin(9600); delay(10);
  Serial.begin(9600); delay(10);
  pinMode(sala,OUTPUT); digitalWrite(sala,LOW);
  pinMode(dorm,OUTPUT); digitalWrite(dorm,LOW);
  pinMode(sshh,OUTPUT); digitalWrite(sshh,LOW);}

void loop(){
  static byte i = 0;
  char endMarker = '\n';
  char rc;
  while (ESP8266.available() > 0 && newData == false) {
    rc = ESP8266.read();
    if (rc != endMarker) {
      receivedChars[i] = rc;
      i++;
      if (i >= numChars) {
        i = numChars - 1;
      }
    } else {
      receivedChars[i] = '\0'; // Termina el string
      for ( int n=0 ; n<=i; n++ ) {
        if (receivedChars[n] == 'r') {
          if (receivedChars[n+1] == 'e') {
            if (receivedChars[n+2] == 'l') {
              if (receivedChars[n+3] == 'o') {
                if (receivedChars[n+4] == 'n') {
                  switch (receivedChars[n+5]) {
```

```

        case 's':
            digitalWrite (sala,HIGH);
            break;
        case 'd':
            digitalWrite (dorm,HIGH);
            break;
        case 'b':
            digitalWrite (sshh,HIGH);
            break;}
    } else if (receivedChars[n+4] == 'f') {
        switch (receivedChars[n+5]) {
            case 's':
                digitalWrite (sala,LOW);
                break;
            case 'd':
                digitalWrite (dorm,LOW);
                break;
            case 'b':
                digitalWrite (sshh,LOW);
                break;
            case 't':
                digitalWrite (sala,LOW);
                digitalWrite (dorm,LOW);
                digitalWrite (sshh,LOW);
                break;}
        }
    }
}
i = 0;
newData = true;
}
}
if (newData == true) {
    Serial.print("This just in ...");
    Serial.println(receivedChars);
    newData = false;
}
if (Serial.available() > 0){
    ESP8266.write(Serial.read());
}
}

```

## ANEXO F SCRIPT EN ARDUINO PARA EL CONTROL DE PERSIANAS

Nombre del archivo: Persianas.ino

```
#include <SoftwareSerial.h>
#define opsala 2
#define clsala 3
#define opdorm 4
#define cldorm 5
#define RxD 11
#define TxD 12

SoftwareSerial ESP8266(RxD, TxD);
const byte numChars = 50;
char receivedChars[numChars];
char data;
boolean newData = false;

void setup(){
  ESP8266.flush(); delay(10);
  ESP8266.begin(9600); delay(10);
  Serial.begin(9600); delay(10);
  pinMode(opsala,OUTPUT); digitalWrite(opsala,LOW);
  pinMode(clsala,OUTPUT); digitalWrite(clsala,LOW);
  pinMode(opdorm,OUTPUT); digitalWrite(opdorm,LOW);
  pinMode(cldorm,OUTPUT); digitalWrite(cldorm,LOW);
}

void loop(){
  static byte i = 0;
  char endMarker = '\n';
  char rc;
  while (ESP8266.available() > 0 && newData == false) {
    rc = ESP8266.read();
    if (rc != endMarker) {
      receivedChars[i] = rc;
      i++;
      if (i >= numChars) {
        i = numChars - 1;
      }
    } else {
      receivedChars[i] = '\0'; // Termina el string
      for ( int n=0 ; n<=i; n++ ) {
        if (receivedChars[n] == 'r') {
          if (receivedChars[n+1] == 'e') {
            if (receivedChars[n+2] == 'l') {
```



```

        if (receivedChars[n+3] == 'o') {
            if (receivedChars[n+4] == 'n') {
                switch (receivedChars[n+5]) {
                    case 's':
                        opclpers(opsala);
                        break;
                    case 'd':
                        opclpers(opdorm);
                        break;}
            } else if (receivedChars[n+4] == 'f') {
                switch (receivedChars[n+5]) {
                    case 's':
                        opclpers(clsala);
                        break;
                    case 'd':
                        opclpers(cldorm);
                        break;
                    case 't':
                        opclpers(clsala);
                        opclpers(cldorm);
                        break;}
                }
            }
        }
    }
    i = 0;
    newData = true;
}
}
if (newData == true) {
    Serial.print("This just in ...");
    Serial.println(receivedChars);
    newData = false;
}
if (Serial.available() > 0){
    ESP8266.write(Serial.read());
}
}
void opclpers(int amb) {
    digitalWrite(amb,HIGH); delay(1500);
    digitalWrite(amb,LOW);
}
}

```

## ANEXO G SCRIPT EN ARDUINO PARA EL CONTROL DE LA TERMA

Nombre del archivo: Terma.ino

```
#include <SoftwareSerial.h>
#define terma 2
#define RxD 11
#define TxD 12

SoftwareSerial ESP8266(RxD, TxD);
const byte numChars = 50;
char receivedChars[numChars];
boolean newData = false;

void setup(){
  ESP8266.flush(); delay(10);
  ESP8266.begin(9600); delay(10);
  Serial.begin(9600); delay(10);
  pinMode(terma,OUTPUT); digitalWrite(terma,LOW);
}

void loop(){
  static byte i = 0;
  char endMarker = '\n';
  char rc;
  while (ESP8266.available() > 0 && newData == false) {
    rc = ESP8266.read();
    if (rc != endMarker) {
      receivedChars[i] = rc;
      i++;
      if (i >= numChars) {
        i = numChars - 1;
      }
    } else {
      receivedChars[i] = '\0'; // Termina el string
      for ( int n=0 ; n<=i; n++ ) {
        if (receivedChars[n] == 'r') {
          if (receivedChars[n+1] == 'e') {
            if (receivedChars[n+2] == 'l') {
              if (receivedChars[n+3] == 'o') {
                if (receivedChars[n+4] == 'n') {
                  digitalWrite(terma,HIGH);
                } else if (receivedChars[n+4] == 'f') {
                  digitalWrite(terma,LOW);
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```
        }
    }
}
}
i = 0;
newData = true;
}
}
if (newData == true) {
    Serial.print("This just in ...");
    Serial.println(receivedChars);
    newData = false;
}
if (Serial.available() > 0){
    ESP8266.write(Serial.read());
}
}
```

## ANEXO H FORMATO DE ENCUESTA SOBRE LA INTERFAZ WEB

## Control Remoto de Hogares (SmartHome)

Toda información ingresada en esta encuesta será utilizada para fines netamente académicos, asimismo, se garantiza la confidencialidad de la misma. Finalmente, le agradezco de antemano por su tiempo y apoyo brindado.

**\*Obligatorio**

Nombre y Apellido \*

Tu respuesta \_\_\_\_\_

Edad \*

Tu respuesta \_\_\_\_\_

¿Conoce o alguna vez ha escuchado hablar del término "Domótica"? \*

Sí

No

¿Le interesaría adquirir un sistema que le permita controlar ciertos artefactos de su hogar (luminarias, persianas, terma, etc) desde cualquier parte del mundo? \*

Usted necesitaría únicamente de un dispositivo electrónico con acceso a internet (celular, tablet, computadora, laptop, etc).

Sí

No

Tal vez

Dependería del costo

Otros: \_\_\_\_\_

Cuánto considera usted que sería un precio justo y razonable por el sistema mencionado en la pregunta anterior. Por favor considere una vivienda con sala, comedor, cocina, baño y 1 dormitorio.

- Menos de S/.1'000
- S/.1'000 - S/.5'000
- S/.5'000 - S/.10'000
- Más de S/.10'000

¿Qué tan entendible encuentra la siguiente interfaz web? \*



- Muy fácil de entender
- Fácil de entender
- Ni fácil ni difícil de entender
- Difícil de entender
- Muy difícil de entender

¿Tiene algún comentario o sugerencia?

Tu respuesta

Página 1 de 1

Enviar

Nunca envíes contraseñas a través de Formularios de Google.

Google no creó ni aprobó este contenido. [Denunciar abuso](#) - [Condiciones del Servicio](#) - [Política de Privacidad](#)

Google Formularios



## BIBLIOGRAFÍA

- [1] Sitio oficial del diario El Peruano, «Crecerá mercado de automatización para "Casas Inteligentes",» 27 Nov 2017. [En línea]. Available: <http://www.elperuano.pe/noticia-crecera-mercado-automatizacion-para-casas-inteligentes-61506.aspx>. [Último acceso: 05 Dic 2019].
- [2] Página Web de Smart House Perú, «Página de Inicio,» 2019. [En línea]. Available: <http://www.smarthouseperu.com/shp2/inicio/>. [Último acceso: 05 Dic 2019].
- [3] Página Web de Conauti SAC, «Página de Inicio,» 2019. [En línea]. Available: <https://conauti.com.pe/>. [Último acceso: 05 Dic 2019].
- [4] Página Web de LCN Perú, «Página de Inicio,» 2019. [En línea]. Available: <http://www.lcnperu.com/>. [Último acceso: 05 Dic 2019].
- [5] Página Web de Everis Group, «Página de Inicio,» 2019. [En línea]. Available: <https://www.everis.com/peru/es>. [Último acceso: 05 Dic 2019].
- [6] Página Web de Domótica, «Página de Inicio,» 2019. [En línea]. Available: <http://domotica.com.pe/>. [Último acceso: 05 Dic 2019].
- [7] Sitio oficial del diario Gestión, «Zona Lounge,» 06 Ago 2019. [En línea]. Available: <https://gestion.pe/especial/zonalounge/tecnologia/fotos-edificios-inteligentes-8-proyectos-ubicados-lima-noticia-1994498>. [Último acceso: 05 Dic 2019].
- [8] Sitio oficial del diario El Comercio, «El Internet de las Cosas ¿Cómo ha avanzado el Perú?,» 27 May 2019. [En línea]. Available: <https://elcomercio.pe/economia/dia-1/internet-cosas-avanzado-peru-noticia-638887-noticia/?ref=e.cr>. [Último acceso: 05 Dic 2019].
- [9] Sitio oficial del diario Gestión, «¿Cuál es el sueldo mínimo en el Perú y cómo ha evolucionado en el tiempo?,» 26 Jun 2019. [En línea]. Available: <https://gestion.pe/tu-dinero/finanzas-personales/sueldo-minimo-peru-aumento-nnda-nnlt-252048-noticia/>. [Último acceso: 06 Dic 2019].
- [10] Sitio oficial del diario El Peruano, «La Remuneración Mínima Vital,» 15 Feb 2017. [En línea]. Available: [http://www.elperuano.pe/noticia-la-remuneracion-minima-vital-51149.aspx%20\(accessed%20December%2030,%202017](http://www.elperuano.pe/noticia-la-remuneracion-minima-vital-51149.aspx%20(accessed%20December%2030,%202017). [Último acceso: 06 Dic 2019].

- [11] Página Web de la Asociación Española de Domótica e Inmótica - CEDOM, «¿Qué es Domótica?,» 2019. [En línea]. Available: <http://www.cedom.es/sobre-domotica/que-es-domotica>. [Último acceso: 20 Nov 2019].
- [12] Sitio oficial del Estándar KNX, «¿Qué es KNX?,» 2019. [En línea]. Available: <https://www.knx.org/knx-es/para-profesionales/Que-es-KNX/Una-breve-introduccion/index.php>. [Último acceso: 28 Nov 2019].
- [13] Sitio oficial de LonMark International, «What is the LonWorks Platform?,» 2019. [En línea]. Available: [http://www.lonmark.org/connection/what\\_is\\_lon](http://www.lonmark.org/connection/what_is_lon). [Último acceso: 28 Nov 2019].
- [14] Sitio oficial de la IECOR, «Estándares Internacionales de Domótica,» 2019. [En línea]. Available: <https://www.iecor.com/estandares-internacionales-de-domotica/>. [Último acceso: 28 Nov 2019].
- [15] Casas Digitales, «Instalar domóticas sin obras,» 23 Mar 2017. [En línea]. Available: <https://www.casasdigitales.com/instalar-domotica-sin-obras/>. [Último acceso: 28 Nov 2019].
- [16] Página Web de Deloitte Touche Tohmatsu Limited - DTTL, «IoT - Internet Of Things,» 2019. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/loT-internet-of-things.html>. [Último acceso: 20 Nov 2019].
- [17] Página Web de Investopedia, «Wearable Technology,» 25 Jun 2019. [En línea]. Available: <https://www.investopedia.com/terms/w/wearable-technology.asp>. [Último acceso: 02 Dic 2019].
- [18] Sitio Oficial del MQTT, «FAQ - What is MQTT?,» 2019. [En línea]. Available: <http://mqtt.org/faq>. [Último acceso: 02 Dic 2019].
- [19] Sitio oficial del CoAP, «RFC7252 Constrained Application Protocol,» 2016. [En línea]. Available: <http://coap.technology/>. [Último acceso: 02 Dic 2019].
- [20] Sitio oficial del AMQP, «AMQP is the Internet Protocol for Business Messaging,» 2019. [En línea]. Available: <https://www.amqp.org/about/what>. [Último acceso: 02 Dic 2019].
- [21] Sitio oficial del XMPP, «FAQ - What is XMPP?,» 2019. [En línea]. Available: <https://xmpp.org/about/faq.html>. [Último acceso: 02 Dic 2019].
- [22] ZIGURAT - Innovation & Technology Business School, «What do the next five years hold for the IoT?,» 2019. [En línea]. Available: <https://www.e-zigurat.com/innovation-school/blog/what-do-the-next-five-years-hold-for-the-iot/>. [Último acceso: 02 Dic 2019].
- [23] Lanner-América, «Sistemas Embebidos 101,» 15 Oct 2019. [En línea]. [Último acceso: 03 Dic 2019].

- [24] El Financiero, «¿Sabes qué es un Sistema Embebido?,» 15 May 2018. [En línea]. Available: <https://elfinanciero.com.mx/bajio/sabes-que-es-un-sistema-embebido>. [Último acceso: 03 Dic 2019].
- [25] Sitio oficial de Arduino, «What is Arduino?,» 2019. [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 20 Nov 2019].
- [26] Arduino, «¿Qué es Arduino?,» 2020. [En línea]. Available: <https://arduino.cl/que-es-arduino/>. [Último acceso: 09 Mar 2020].
- [27] Sitio oficial de Arduino, «Arduino NANO - Tech Specs,» 2019. [En línea]. Available: <https://store.arduino.cc/usa/arduino-nano>. [Último acceso: 09 Dic 2019].
- [28] Sitio oficial de Arduino, «Arduino NANO,» 2019. [En línea]. Available: <https://store.arduino.cc/usa/arduino-nano>. [Último acceso: 09 Dic 2019].
- [29] Sitio oficial de Arduino, «Arduino NANO - User Manual,» 2019. [En línea]. Available: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>. [Último acceso: 09 Dic 2019].
- [30] Sitio oficial de la Raspberry Pi Foundation, «FAQ - What is a Raspberry Pi?,» 2019. [En línea]. Available: <https://www.raspberrypi.org/documentation/faqs/#introduction>. [Último acceso: 20 Nov 2019].
- [31] Sitio oficial de la Raspberry Pi Foundation, «Raspbian,» 2019. [En línea]. Available: <https://www.raspberrypi.org/downloads/raspbian/>. [Último acceso: 10 Dic 2019].
- [32] Sitio oficial del OS Raspbian, «Raspbian,» 2019. [En línea]. Available: <https://www.raspbian.org/>. [Último acceso: 10 Dic 2019].
- [33] Sitio oficial de la Python Software Foundation, «FAQ - What is Python?,» 2019. [En línea]. Available: <https://docs.python.org/3/faq/general.html#what-is-python>. [Último acceso: 10 Dic 2019].
- [34] Sitio oficial de la Secure Shell, «SSH (Secure Shell),» 2019. [En línea]. Available: <https://www.ssh.com/ssh/>. [Último acceso: 10 Dic 2019].
- [35] Raspberry Shop, «Guía de Raspberry Pi para principiantes,» 2019. [En línea]. Available: <https://www.raspberrypi.es/guia-completa-raspberry-pi.php>. [Último acceso: 10 Dic 2019].
- [36] Página Web de Hostinger, «¿Cómo funciona el SSH?,» 17 Oct 2019. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-ssh>. [Último acceso: 10 Dic 2019].
- [37] Sitio oficial de la Raspberry Pi Foundation, «VNC (Virtual Network Computing),» 2019. [En línea]. Available: <https://www.raspberrypi.org/documentation/remote-access/vnc/README.md>. [Último acceso: 10 Dic 2019].
- [38] Sitio oficial de la Raspberry Pi Foundation, «Raspberry Pi 2 Modelo B,» 2019. [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Último acceso: 10 Dic 2019].



- [39] Inventable.eu, «Introducción a los Relés,» Creative Commons, 2020. [En línea]. Available: <https://www.inventable.eu/introduccion-a-los-reles/>. [Último acceso: 17 Ene 2020].
- [40] Ingeniería Mecafenix, «¿Qué es un relevador y cómo funciona?,» 17 Abr 2017. [En línea]. Available: <https://www.ingmecafenix.com/electricidad-industrial/relevador/>. [Último acceso: 17 Ene 2020].
- [41] Á. Tecnología.com, «Relés,» 2020. [En línea]. Available: <https://www.areatecnologia.com/electricidad/rele.html>. [Último acceso: 17 Ene 2020].
- [42] OpenLanuza, «Utilizando un relé en Arduino,» 17 Ene 2018. [En línea]. Available: <https://openlanuza.com/utilizando-un-rele-en-arduino/>. [Último acceso: 20 Ene 2020].
- [43] Real Academia Española (RAE), «Diccionario de la lengua española,» 2019. [En línea]. Available: <https://dle.rae.es/?w=motor>. [Último acceso: 06 Mar 2020].
- [44] Harmonic Drive SE, «Motor DC: El pionero de los electromotores,» 2020. [En línea]. Available: <https://harmonicdrive.de/es/glosario/motor-dc-el-pionero-de-los-electromotores>. [Último acceso: 06 Mar 2020].
- [45] Compañía Levantina de Reductores (CLR), «Motores de corriente continua y alterna: estudio y selección,» 2020. [En línea]. Available: <https://clr.es/blog/es/motores-corriente-continua-alterna-seleccion/>. [Último acceso: 06 Mar 2020].
- [46] Aficionados a la Mecánica, «Coche Eléctrico,» 2017. [En línea]. Available: [http://www.aficionadosalamecanica.com/coche-electrico\\_control.htm](http://www.aficionadosalamecanica.com/coche-electrico_control.htm). [Último acceso: 06 Mar 2020].
- [47] Inventable.eu, «¿Cómo funciona un puente para motores de corriente continua?,» 26 May 2017. [En línea]. Available: <https://www.inventable.eu/2017/05/26/funciona-puente-motores-corriente-continua/>. [Último acceso: 09 Mar 2020].
- [48] DIYMakers.es, «Control de velocidad y sentido de Motor DC,» 26 Nov 2013. [En línea]. Available: <http://diymakers.es/control-velocidad-y-sentido-de-motor-dc/>. [Último acceso: 09 Mar 2020].
- [49] Texas Instrument, «L293x Quadruple Half-H Drives,» Ene 2016. [En línea]. Available: <http://www.ti.com/lit/ds/symlink/l293.pdf>. [Último acceso: 09 Mar 2020].
- [50] Amazon.es, «Compra de módulo Wi-Fi ESP01,» 2020. [En línea]. Available: <https://www.amazon.es/HiLetgo-ESP8266-Transceptor-inalámbrico-Compatible/dp/B00O9DSZBA>. [Último acceso: 20 Ene 2020].
- [51] AI-Thinker, «ESP-01 Wi-Fi Module,» 2015.
- [52] Espressif Systems, «ESP8266EX Datasheet,» 2019.
- [53] Samsung, «Firmware,» 13 May 2018. [En línea]. Available: <https://www.samsung.com/latin/support/tv-audio-video/firmware/>. [Último acceso: 21 Ene 2020].

- [54] LG, «¿Qué es el Firmware?,» 25 Oct 2016. [En línea]. Available: <https://www.lgblog.cl/lgtv/tecnologia-tendencias/que-es-el-firmware/>. [Último acceso: 21 Ene 2020].
- [55] Espressif Systems, «Tools,» 13 Dic 2019. [En línea]. Available: [https://www.espressif.com/en/support/download/other-tools?keys=&field\\_type\\_tid%5B%5D=14](https://www.espressif.com/en/support/download/other-tools?keys=&field_type_tid%5B%5D=14). [Último acceso: 22 Ene 2020].
- [56] Espressif Systems, «AT Bin - ESP8266EX,» 07 Feb 2018. [En línea]. Available: [https://www.espressif.com/en/support/download/at?keys=&field\\_type\\_tid%5B%5D=14](https://www.espressif.com/en/support/download/at?keys=&field_type_tid%5B%5D=14). [Último acceso: 22 Ene 2020].
- [57] Naylamp Mechatronics, «Módulo PL2303 Conversor USB a Serial TTL,» 2020. [En línea]. Available: <https://naylampmechatronics.com/conversores-ttl/40-modulo-pl2303-conversor-usb-a-serial-ttl.html>. [Último acceso: 22 Ene 2020].
- [58] EcuRed, «Servidor Web,» 2020. [En línea]. Available: [https://www.ecured.cu/Servidor\\_Web](https://www.ecured.cu/Servidor_Web). [Último acceso: 27 Ene 2020].
- [59] Servicios Informáticos Hostname Ltda., «Servidor Web,» 26 Feb 2014. [En línea]. Available: <https://www.hostname.cl/blog/servidor-web>. [Último acceso: 27 Ene 2020].
- [60] Sitio oficial de Apache Friends, «¿Qué es XAMPP?,» 2020. [En línea]. Available: <https://www.apachefriends.org/es/index.html>. [Último acceso: 09 Mar 2020].
- [61] EcuRed, «Bases de datos,» 2020. [En línea]. Available: [https://www.ecured.cu/Bases\\_de\\_datos#V.C3.A9ase\\_tambi.C3.A9n](https://www.ecured.cu/Bases_de_datos#V.C3.A9ase_tambi.C3.A9n). [Último acceso: 27 Ene 2020].
- [62] Platzi.com, «Bases de datos desde cero,» 2020. [En línea]. Available: <https://platzi.com/base-de-datos/>. [Último acceso: 09 Mar 2020].
- [63] Sitio oficial de MySQL, «MySQL Editions,» 2020. [En línea]. Available: <https://www.mysql.com/products/>. [Último acceso: 09 Mar 2020].
- [64] The PHP Group, «¿Qué es PHP?,» 2020. [En línea]. Available: <https://www.php.net/manual/es/intro-what-is.php>. [Último acceso: 11 Mar 2020].
- [65] Massachusetts Institute of Technology, «MIT App Inventor,» 2020. [En línea]. Available: [ai2.appinventor.mit.edu/](http://ai2.appinventor.mit.edu/). [Último acceso: 17 Mar 2020].
- [66] Bluehost, «The Best Web Hosting,» 2020. [En línea]. Available: <https://www.bluehost.com/>. [Último acceso: 16 Mar 2020].
- [67] Grupo La Provincia, «Para ahorrar, ¿Lámparas LED o de bajo consumo?,» 30 May 2018. [En línea]. Available: <https://www.grupolaprovincia.com/economia/para-ahorrar-lamparas-led-o-de-bajo-consumo-174835>. [Último acceso: 03 Feb 2020].
- [68] Google, «Formularios de Google: crea y analiza encuestas de forma gratuita,» 2020. [En línea]. Available: [https://www.google.com/intl/es-419\\_pe/forms/about/](https://www.google.com/intl/es-419_pe/forms/about/). [Último acceso: 2020].