



**Universidad Nacional de Ingeniería**  
**Facultad de Electrotecnia y Computación**  
**Ingeniería Electrónica**

*Proyecto Monográfico*

**“Implementación de un Prototipo de Sistema de Control de Acceso y Luminaria,  
controlado por Voz para Personas con Discapacidad Físico-Motora y Personas  
Ciegas”**

Autor:

Br. Mario Ernesto Goussen Sequeira

Carnet:

2015-0165U

Tutor:

Ing. Marco Antonio Munguía Mena

**Managua, Nicaragua 2022**



## **Dedicatoria**

Dedico este trabajo con gran amor y admiración a toda mi familia, en especial a mi Madre Amada por el apoyo incondicional, por siempre impulsarme a ser mejor, por ser mi motivación para culminar con éxito mi carrera, por su sacrificio constante de Madre en todos estos años de estudios y hasta el día de hoy para ver crecer a sus hijos como profesionales y personas de bien.

## **Agradecimientos**

A Dios por haberme dado salud y vida para permitirme cumplir esta meta con éxito.

A mis maestros y tutores "Ing. Carlos Alberto Ortega Huembes" Q.D.E.P quien fue el primer tutor que me ayudo en la redacción del documento y me brindo sus conocimientos en este campo para desarrollar mi proyecto. De igual manera al "Ing. Marco Antonio Munguía Mena" quien me ha ofrecido sabios consejos y ha sabido encaminarme al camino correcto para culminar este proyecto de manera exitosa.

A mi familia por el apoyo, comprensión y guía para lograr cumplir esta meta en mi vida.

## Resumen Del Tema

En este proyecto monográfico se llevó a cabo la fabricación de un prototipo de sistema de control de acceso y luminaria, controlado por medio de voz para personas con discapacidades físico-motoras y personas ciegas. El desarrollo de este proyecto se ha realizado por medio de cuatro etapas.

La primera etapa se ha realizado la recopilación de información de las distintas tecnologías necesarias para garantizar el correcto funcionamiento de todo el sistema, en la cual se ha realizado una breve descripción de cada herramienta y explicando el motivo por los cuales cada una de ellas son necesarias.

La segunda etapa comprende todo el diseño del dispositivo representado con un diagrama de bloque de todo el sistema incluyendo la programación de la tarjeta de desarrollo y las configuraciones de la plataforma en la nube.

La tercera etapa está conformada por toda la fabricación del sistema de acondicionamiento para la tarjeta de desarrollo y todas las pruebas correspondientes para verificar el funcionamiento de todo el sistema de control.

La cuarta etapa se elaboró todo el presupuesto que fue usado para la fabricación del dispositivo además de realizar una comparativa tecnológica con otras soluciones similares en el mercado actualmente.

# Contenido

Introducción.....	1
Objetivos.....	2
I. Objetivos General.....	2
II. Objetivos Específicos .....	2
Justificación.....	3
Primer capítulo: Marco Teórico (Descripción del Hardware y Software) .....	4
Servicios .....	4
Plataforma y Tecnologías en la Nube.....	5
Hardware .....	7
Segundo Capitulo: Estructura Del Software .....	9
Diagrama De Bloque .....	9
Diagrama de Flujo.....	10
Tercer capítulo: Fabricación y Pruebas del Hardware .....	20
Simulación del Circuito de Acondicionamiento .....	20
Fabricación del Dispositivo .....	22
Pruebas Finales.....	38
Cuarto capítulo: Comparativa Tecnológica y Económica.....	44
Presupuesto Del Prototipo.....	44
Comparativa Del Prototipo.....	47
Conclusiones.....	55
Recomendaciones .....	56
Bibliografía.....	57
Anexo .....	1
1. Configuración de la Plataforma.....	1
2. Configuración de Raspberry pi 3 y 3 B+.....	7
3. Configuración de ESP-01 y Nodemcu v2.....	12

## Tabla De Ilustraciones

Figura 1: Diagrama De Bloque.....	9
Figura 2: Flujograma Del ESP-01. ....	11
Figura 3: Flujograma Para Raspberry Pi y NodeMCU V2. ....	12
Figura 4: Flujograma Del Servidor. ....	13
Figura 5: Flujograma Del Home y Login.....	14
Figura 6: Flujograma Del Registro. ....	15
Figura 7: Flujograma De Configuración. ....	16
Figura 8: Flujograma Del Password. ....	17
Figura 9: Flujograma Del Control.....	18
Figura 10: Flujograma De Data. ....	19
Figura 11: Circuito De Acondicionamiento. ....	21
Figura 12: Circuito De Acondicionamiento Para Dos GPIO. ....	21
Figura 13: Circuito Para Raspberry Pi. ....	23
Figura 14: Diseño de la PCB Para Raspberry Pi. ....	24
Figura 15: Impresión De La PCB.....	25
Figura 16: Diseño En 3D de la PCB Para Raspberry Pi.....	25
Figura 17: Baquelita Virgen. ....	26
Figura 18: PCB Con El Circuito.....	26
Figura 19: Circuito En Cobre. ....	27
Figura 20: Componentes Electrónicos.....	28
Figura 21: Conector Tipo C13, Capuchones Y Jack Rj45. ....	28
Figura 22: Soldadura De La Terminada. ....	28
Figura 23: Baquelita Terminada. ....	28
Figura 24: Parte Interior De La Caja. ....	29
Figura 25: Parte Exterior De La Caja.....	29
Figura 26: Prototipo Finalizado. ....	29
Figura 27: Circuito Para ESP-01. ....	30
Figura 28: Diseño de la PCB Para ESP-01.....	31
Figura 29: PCB En 3D. ....	32
Figura 30: Impresión De La PCB ESP-01. ....	32
Figura 31: Baquelita Virgen. ....	33
Figura 32: Circuito En La Baquelita.....	33
Figura 33: Circuito En Cobre. ....	34
Figura 34: Baquelita Perforada.....	34
Figura 35: Componentes Electrónicos.....	34
Figura 36: Conectores. ....	34
Figura 37: PCB Terminada De ESP-01. ....	35
Figura 38: Parte Exterior De La Caja Del ESP-01.....	35
Figura 39: Parte Interior De La Caja Del ESP-01. ....	35
Figura 40: Prototipo Del ESP-01 Terminado. ....	36
Figura 41: Circuito Para NodeMCU V2. ....	37
Figura 42: Diseño De La PCB Para NodeMCU V2.....	37
Figura 43: Pruebas Del Prototipo Completo. ....	38
Figura 44: Sistema De Control. ....	39
Figura 45: Activación Por Google Assistant. ....	39
Figura 46: Plataforma En La Nube.....	40
Figura 47: Sistema De Control Activado. ....	40
Figura 48: Apagar Lampara 1.....	41
Figura 49: Dispositivos Apagado. ....	41
Figura 50: Lámparas Apagadas.....	41
Figura 51: Activación De Lámpara 2.....	42
Figura 52: Activación De Lámpara 3.....	42
Figura 53: Activación De Todas Las Lámparas.....	43
Figura 54: Características Del Sonoff Basic.....	48
Figura 55: Interior Del Sonoff.....	48
Figura 56: 2-CH Triac HAT Para Raspberry Pi. ....	49
Figura 57: Enchufe inteligente NHP-D610. ....	50

Figura 58: Comparativa Entre ESP-01 Vs Sonoff Basic.....	51
Figura 59: Comparativa Entre Raspberry Pi vs 2-CH Triac Hat .....	52
Figura 60: Comparativa Entre Raspberry Pi vs Yale .....	53
Figura 61: Comparativa Entre NodeMCU V2 vs NHP-D610 .....	54
Figura 62: Login .....	1
Figura 63: Registro.....	1
Figura 64: Verificar .....	2
Figura 65: Home .....	2
Figura 66: configuración.....	2
Figura 67: Notificación.....	3
Figura 68: Agregar Dispositivo .....	3
Figura 69: Agregar Dispositivo Ejemplo 1.....	4
Figura 70: Home Dispositivo 1 .....	5
Figura 71: Agregar Dispositivo Ejemplo 2.....	5
Figura 72: Home Dos dispositivos.....	6
Figura 73: Eliminar Dispositivo .....	6
Figura 74: Logo Raspberry Pi .....	7
Figura 75: Pin del GPIO .....	7
Figura 76: GPIO de la Raspberry Pi .....	8
Figura 77: Autenticación.....	9
Figura 78: Control de GPIO .....	9
Figura 79: Pines del GPIO.....	9
Figura 80: Ejemplo de Autenticación .....	10
Figura 81: Control de Dos Salidas .....	10
Figura 82: Consola de Comandos .....	10
Figura 83: Ejecutar Programa.....	11
Figura 84: Programa Ejecutado .....	11
Figura 85: Nodemcu v2.....	12
Figura 86: ESP-01.....	12
Figura 87: SHA-1 .....	12
Figura 88: Seguridad SSL .....	12
Figura 89: SSL Info .....	13
Figura 90: Certificado SSL .....	13
Figura 91: Pines de Control.....	13
Figura 92: Configuración de Pines .....	13
Figura 93: Contraseña del WIFI .....	13
Figura 94: Autenticación al Servidor.....	14
Figura 95: Control de Pines.....	14
Figura 96: Control de Dos Pines.....	14

# Introducción

La revolución digital que se dio en las últimas décadas del siglo pasado y han tenido su máximo apogeo en los primeros años del nuevo siglo han traído con ella nuevas tecnologías que facilitan la vida de las personas, pero este avance no ha llegado a todos por igual. Las personas con discapacidad físico-motora y personas ciegas o con visión reducida, han quedado casi en la mayoría de los casos por fuera o en segundo plano.

En el 2009 la población de Nicaragua estaba conformada de 5.7 millones de habitantes [1]. La discapacidad en Nicaragua alcanzaba un 2.5% de la población total del cual predominan las personas con discapacidad físico motora con un 36.9% y en menor medida las discapacidades visuales un 12.7% según el estudio realizado el mismo año por [2]. Sin embargo, estas personas se ven afectada debido a la falta de accesibilidad para poder movilizarse adecuadamente, además de complicarse con el uso tradicional de interruptores, cerradura y olvidar también el apagado de luces antes de acostarse, especialmente para el caso de parejas ciegas que no poseen a una persona que les ayude y su única guía es un mapa mental.

Viendo de esta manera el factor de seguridad y uso eficiente de la energía eléctrica, existen soluciones de dispositivos inteligente que pueden ayudar a facilitar la vida de estas personas, poseyendo en la mayoría de los casos estos terminales IoT un costo elevado para muchas familias. Donde el índice de pobreza alcanzaba un 24.9% de la población en 2016 [3].

Debido a la existencia de esta problemática se ha tomado inspiración para basar este proyecto monográfico con la finalidad de generar una posible solución, creando una plataforma de control de iluminaria y seguridad utilizando la voz por medio del lenguaje natural para dictar comandos de control la cual le permitirá una fácil manipulación de sus dispositivos del hogar, creando un ambiente accesible para las personas con discapacidades físicas-motoras y personas ciegas<sup>1</sup>.

---

<sup>1</sup> De acuerdo con “Organización de ciegos maricela Toledo” y “uso de lenguaje inclusivo persona en situación de discapacidad” el termino correcto a utilizar es “personas ciegas”.

<https://www.cultura.gob.cl/wpcontent/uploads/2017/01/guia-recomendaciones-lenguaje-inclusivo-discapacidad.pdf>



# Objetivos

## I. Objetivos General

- Implementar un Prototipo de Sistema de Control de Acceso y Luminaria, controlado con Voz para Personas con Discapacidad Físico-Motora y Personas Ciegas.

## II. Objetivos Específicos

- Describir los Software y Hardware necesarios para el funcionamiento del prototipo.
- Desarrollar una aplicación basada en Python para el sistema de control de acceso y luminaria.
- Diseñar el sistema de control electrónico y efectuar las pruebas de fabricación del prototipo.
- Determinar los costos de las diferentes tarjetas desarrolladas.

## Justificación

Las personas discapacitadas necesitan elementos para poder tener accesibilidad en el entorno en que ellos se encuentren. Abarcando a las personas ciegas, sufren al desconocimiento de un nuevo entorno que no sea familiar para ellos dificultándose la movilización además de no poder describir una gran cantidad de objetos en ese ambiente pero con el tiempo desarrolla habilidad que le permite movilizarse de forma más natural, poseen problema en el uso cotidiano de abrir o cerrar una puerta y también de no poder identificar el tipo de llave que deben de usar para poder acceder a su hogar o una habitación en concreto y más cuando poseen una gran cantidad de llaves puede con llevar a pérdidas de estas mismas. En Organización de Ciegos de Nicaragua Maricela Toledo declararon que es un problema frecuente el olvidar apagar las luces de sus casas haciendo que este problema sea un mayor gasto energético en sus facturas y esto se convertirá en un mayor gasto monetario para ellos [4].

Las personas con discapacidad física-motora es un campo muy amplio en el cual existen personas que pueden hacer el uso correcto de manipulación de cerraduras tradicionales al no presentar problemas en sus manos, pero pueden presentar problemas en las piernas que le dificulte la movilización en por la habitación además que se debe contemplar aquellas personas que tiene dificultades físicas en sus manos y la manipulación de cerraduras muy rudimentarias pueden llegar a convertirse en toda una odisea cada día.

Este proyecto otorga una solución de bajo costo de poder convertir todo el sistema de iluminaria existente de su hogar en dispositivos fácil control para las personas ciegas y con discapacidad física-motora haciendo el uso de la voz con ayuda de sus dispositivos móviles para poder controlar el acceso a la casa y las demás puertas que puede existir en su hogar, esta será controlada por una placa de desarrollo la cual esta se encargará de controlar cada uno de sus dispositivos configurados, además de presentar una interfaz completamente accesible y diseñada para las personas ciegas para que puedan realizar un seguimiento de sus dispositivos el cual le ayudara a recordar si han dejado una lámpara encendida o incluso una cerradura desactivada.

## **Primer capítulo: Marco Teórico (Descripción del Hardware y Software)**

En este primer capítulo se abarcarán las diferentes herramientas y tecnologías tanto a nivel de hardware como de software que se utilizaron para hacer el diseño del proyecto, además de realizar una explicación de la importancia que tiene cada uno y el rol que juegan para conseguir el correcto funcionamiento de todo el ecosistema.

### **Servicios**

#### **Google Assistant**

Google Assistant es un asistente virtual desarrollado con inteligencia artificial por Google. Los usuarios interactúan principalmente haciendo el uso de la voz natural, el asistente puede buscar cosas en internet, programar eventos y alarmas, ajustar la configuración de hardware en el dispositivo del usuario y mostrar información del usuario [5]. Esta herramienta está disponible en la mayoría de los dispositivos móviles que posee el sistema operativo Android en el mercado actual y en diferentes bocinas inteligentes por lo que el uso de esta herramienta está disponible y alcance de muchos usuarios en la actualidad.

Esta herramienta permite interactuar con los aparatos del hogar configurado en la plataforma con las frases que se desea utilizar para activar o desactivar un equipo, haciendo el uso del dispositivo móvil o una bocina inteligente que integre Google Assistant por medio de la voz natural facilitándole el uso a las personas con discapacidades físicas-motoras y las personas ciegas. En la actualidad existen otros asistentes virtuales como Alexa de Amazon o Bixby de Samsung, pero en este caso se utilizará Google Assistant.

#### **Amazon Web Services (AWS)**

En el mercado mundial actualmente existe diferentes empresas que brindan soluciones en la nube, algunas que se pueden encontrar son Google Cloud y Microsoft Azure. Pero en este caso se trabajó con AWS que es la propuesta por Amazon de la cual se va a necesitar para alojar la plataforma en la nube brindándonos todos los beneficios que conlleva, algunos de estos son: recursos escalables, ahorro de dinero en infraestructura, alta disponibilidad, alto rendimiento, mayor seguridad, etc [6]. Logrando de esta manera conseguir que la plataforma pueda ser accesible desde cualquier lugar y que el usuario podrá acceder a la página web para poder llevar el control de sus dispositivos.

## Duck DNS

Duck DNS es un servicio gratuito DNS con subdominio “duckdns.org” [7]. se utilizará un servidor DNS para que la plataforma en la nube tenga un nombre de dominio donde el usuario pueda acceder sin necesidad de utilizar la IP pública del servidor y facilitándole recordar la URL de la página web, además que Google Assistant pide como requerimiento tener una IP pública asociada con un nombre de dominio para poder realizar la comunicación de lo contrario la API de Google no funciona.

## Certbot

Es una herramienta de código abierto la cual permite utilizar automáticamente los certificados “Let’s Encrypt” [8]. Esto es necesario porque la plataforma en la nube inicialmente trabaja por medio de HTTP, como no es una comunicación cifrada y segura, es necesario obtener un certificado SSL para lograr cifrar la comunicación, es posible crear un certificado como comúnmente suele realizarse, pero esto causa otro problema debido que el certificado generado no está validado por ninguna autoridad de certificación los navegadores dan una alerta diciendo que el sitio web no es seguro y para solucionar ese problema se utiliza “Certbot”, instalará un bot que administrará el certificado que pedirá obligatoriamente asociar el certificado con un nombre de dominio en este caso se utiliza el nombre creado con duckdns una vez configurado genera un certificado de “Let’s Encrypt” que es una autoridad de certificación gratuita [9]. Por consiguiente, la página web consta de un certificado SSL que cualquier navegador al momento de entrar podrán validar que el sitio es seguro por lo tanto no saldrá ninguna alerta de seguridad, para mantener la seguridad estos certificados tienen fecha de expiración, pero el bot se encargará automáticamente de renovarlo manteniendo la plataforma en nube cifrada, además tener un certificado SSL es necesario para poder utilizar la API de Google Assistant.

## Plataforma y Tecnologías en la Nube

Se desarrolló una plataforma en la nube que es la encargada de administrar y gestionar las solicitudes dadas por el usuario a través del asistente de voz e interpretarlas para enviar la información a los actuadores y lograr el control de la activación o desactivación de los dispositivos creando una percepción al usuario de una fácil gestión de su hogar, además de crear una interfaz web accesible para las personas ciegas contando con audio para facilitar la navegación de la plataforma además de tener la capacidad de informar por medio de audio el estado actual de sus dispositivos.

## Apache

Para garantizar el correcto funcionamiento de la plataforma en la nube es necesario tener instalado un servidor web que es un software el cual se encarga de procesar la aplicación web. Su trabajo es establecer una conexión entre el servidor físico y los navegadores de los visitantes al sitio web para lograr este objetivo extrae el contenido del servidor físico en cada solicitud de usuario y lo envía [10]. Para lograr realizar todo este trabajo se usó apache pues se encargará de administrar las solicitudes de los usuarios a través del protocolo HTTP, apache es responsable de garantizar una comunicación fluida y segura entre las dos máquinas.

## Flask

Apache simplemente es un administrador de las consultas clientes-servidor, pero este necesita una aplicación a la cual debe consultar y será la aplicación web que se desarrolló. Existen muchos lenguajes de programación para desarrollarla, aunque en este caso se hablará de un framework web llamado Flask [11]. Este framework permite desarrollar una aplicación web en el lenguaje Python de una forma sencilla ya que consta de diferentes herramientas, así permitiendo trabajar con las consultas del cliente, URL y poder utilizar un motor de plantillas web donde se puede manipular la página para que sea dinámica visualmente. Flask es totalmente compatible con apache debido que se basa en WSGI<sup>1</sup> que es una interfaz en común entre el servidor web y aplicaciones web. En este caso el servidor web sería apache y la aplicación web será Flask donde se programó toda la administración y control de los dispositivos del hogar.

## Mysql

En la administración de usuarios, configuración y dispositivos, siempre es necesario llevar un registro por ende se debe almacenar toda esa información para utilizarla posteriormente y para lograr controlar toda esta información de los usuarios es necesario utilizar un administrador de base de datos ha siendo uso de Mysql [12]. Se necesita Mysql porque la aplicación web almacenara las credenciales de cada usuario tanto como los dispositivos del hogar en la base de datos para no perder información y cada vez que el sistema de control le haga una consulta al servidor web este hará una consulta a la aplicación Flask la cual debe realizar una consulta Sql a la base de datos y para manejar esta consulta Sql donde puede ser búsqueda de información, inserción o eliminación, Mysql se encarga del correcto funcionamiento volviendo la información requerida a la aplicación Flask donde esta le devolverá una respuesta al usuario.

---

<sup>1</sup> WSGI son las siglas de Web Server Gateway Interface.

## Javascript

Existen acciones o eventos que a nivel de servidor no es posible de llevar a cabo por esta razón se debe buscar otras técnicas para lograr ejecutar acciones en específico que sean a nivel de cliente, funciones que se ejecuten en el navegador y para lograr esto se utilizó JavaScript que es un lenguaje de programación que te permite implementar funciones complejas en páginas web [13]. Esto permite crear contenido de actualización dinámica de manera que da la posibilidad de mostrar el estado de los dispositivos actual sin necesidad de volver a recargar toda la página web y solo se actualizará las imágenes de los dispositivos. Otra función es el uso de multimedia pero enfocado en reproducir audio, cuando el cliente pase el cursor por un texto en específico reproduciendo un audio para notificar la acción que está realizando.

## Hardware

Para que el trabajo de estos servicios se lleve a cabo se necesita un hardware que se encargue del control, la administración de los dispositivos y además que seas capaz de conectarse a internet mediante de redes inalámbricas. Existen una gran diversidad de soluciones de hardware que puede satisfacer en dependencia de las necesidades del usuario, se muestran tres posibles soluciones de módulos o placas de desarrollo que cumple con los requisitos básico para desactivar o activar los dispositivos.

### ESP-01

Es una solución para ciertos casos que no se necesite controlar más de dos dispositivos o solo se desee controlar un dispositivo de forma aislada debido que solo posee dos pines GPIO y se desee conseguir un diseño lo más pequeño posible, esta tarjeta posee un módulo wifi integrado por lo tanto es capaz de conectarse al servidor y a la vez controlar los dispositivos. Es una alternativa perfecta para una solución en un tamaño muy reducido y a muy bajo costo.

### NodeMCU V2

La segunda la solución en caso que se necesite más pines GPIO para controlar más dispositivos es el NodeMCU del cual existe dos variantes la versión V3 y la V2. La diferencia principal entre las versiones son el tamaño de la tarjeta debido que poseen la misma cantidad de GPIO con un total de nueve pines

para administrar nueve dispositivos diferentes. Se utilizará NodeMCU V2 debido que es la versión más pequeña y se desea crear un diseño pequeño. Esta tarjeta al igual que ESP-01 trae integrado una antena WIFI de esta manera se conectará directamente al servidor y como tiene más capacidad de pines es posible crear un diseño un poco más grande, pero con el beneficio de poner administrar muchos más dispositivos.

### **Raspberry pie B+ 3**

Existen casos que se desea controlar una gran cantidad de dispositivos a la vez para lograr esto se necesita una Raspberry Pi que posee un total de 28 pines GPIO esta es una de las alternativas más costosas, pero permite administrar más dispositivos a la vez, para este caso como la tarjeta es muy grande el diseño sería por medio de cables para activar y desactivar que se conectaría con el sistema de control que estaría en pequeñas cajas independiente que solo tendrá en su interior el sistema de acondicionamiento, esto permite agregar la cantidad de caja que se vaya a necesitar de forma independiente.

### **Sistema de acondicionamiento**

Esta última parte es necesaria para conseguir el objetivo final de controlar cerraduras y sistema de iluminación convencional del hogar haciendo uso de la voz, debido que la tarjeta Raspberry Pi, ESP-01 y NodeMCU V2 no son capaces por sí solo de activar dispositivos que necesite más 3.3V o 5V y 16mA de corriente directa, es necesario crear un circuito de acondicionamiento que le permita controlar voltaje de 120V y con un mínimo de 20A en corriente alternar que son los parámetros promedios que suelen consumir los equipos de un hogar convencional. Para lograr que las tarjetas de desarrollo consigan controlar estos parámetros de voltaje y corriente AC se necesita un intermediario que separe el sistema de control que trabaja en corriente DC proveniente del GPIO de las placas y el sistema de potencia que se conectará con el Triac que trabaja en corriente AC proveniente de la red de energía comercial del hogar, como intermediario se utiliza un optoacoplador para garantizar la protección de las placas de desarrollo. Por cada nodo de control que se necesite fabricar debe repetirse este mismo proceso.

## Segundo Capitulo: Estructura Del Software

En este capítulo se describe todo el proceso del diseño del dispositivo mediante un diagrama de bloques, donde cada bloque es una fase para lograr conseguir el prototipo final. Comprendiendo de la estructura del sistema en general hasta un análisis del mismo de forma más específica haciendo referencia a la programación que se llevó a cabo tanto para crear la plataforma en la nube como la programación de las tarjetas de desarrollo y concluyendo con el diseño del circuito electrónico de control.

### Diagrama De Bloque

En la Figura 1 se puede observar el diagrama de bloque de todo el sistema en general. El sistema da inicio en la parte llamada “**Bocinas/Smartphone**”, cuando el usuario en su dispositivo con la aplicación de Google Assistant ejecute la orden de activación que ha configurado, un ejemplo es “**Encender Lámpara 1**” esta frase Google Assistant la convierte en un texto que se envían dos parámetros al servidor que el primero será “**Acción**” de esta manera se obtendrá si se desea “**Encender**” o “**Apagar**” y el segundo parámetro es “**Identificador**” aquí dirá el dispositivo que será afectado por esa acción. Una vez que el servidor obtenga esta información de Google Assistant le responde con un mensaje este lo recibe y lo convierte en audio para indicarle al cliente que la orden ha sido ejecutada por ejemplo “**Lampara 1 Encendida**”.



Figura 1: Diagrama De Bloque.



La segunda parte tiene el nombre de “**Plataforma en la nube**” en la cual todo se desarrolla dentro del servidor con los dos parámetros recibidos llamados “**Acción**” y “**Identificador**” este realiza una búsqueda en la base de datos utilizando el identificador del dispositivo, una vez que lo encuentra este da la orden de ejecutar una actualización del estado esto quiere decir que si estaba apagado y la acción es “**Encender**” actualiza la base de datos a encendido.

En la tercera parte corresponde al “**sistema de control**”. El estado de “**lámpara 1**” ha sido modificado por lo tanto la tarjeta de desarrollo tiene que estar enviando una solicitud POST cada cierto tiempo para verificar si los estados de los dispositivos han sufrido un cambio, la placa envía un mensaje al servidor y este responde con todos los parámetros de cada dispositivo que estén registrado, una vez recibiendo esta información inicia a verificar uno a uno el “**Estado**” y el “**Identificador**” si ha detectado un cambio del estado entra en una condición en la cual dependido de la situación da la orden de enviar “**1**” o un “**0**” al GPIO correspondiente y de esta forma lograr cumplir con la orden ejecutada por el usuario.

## Diagrama de Flujo

En esta sección se explica en más detalle la programación que se llevó a cabo para lograr el objetivo que todo funcione correctamente. Para dar una explicación grafica de la estructura interna del funcionamiento del servidor y de cómo administra los GPIO las tarjetas de desarrollo se hace uso de diagrama de flujos representando cada proceso interno de cada uno.

### Flujograma del ESP-01

En la Figura 2 está el flujograma correspondiente a la tarjeta ESP-01. El primer proceso que debe cumplir es conectarse al servidor sino lo consigue manda una alerta de error con el siguiente mensaje “**[HTTPS] Unable to connect**” en el dado caso que logre conectar al servidor se configuraran las entradas y salidas digitales de la placa de desarrollo en el caso del ESP-01 solo puede administrar 2 dispositivos a la vez por lo tanto solo tendrá dos salidas, el programa empieza a leer las dos variable entrante proveniente del servidor que son “**ID**” que contiene una lista de los “**Identificador**” de cada dispositivo y “**Es**” que almacena el estado que se debe encontrar cada uno actualmente, haciendo lectura de estos datos se toma la acción a realizar con cada uno de los posibles casos, si “**ID**” es igual a “**Lámpara1**” y su estado “**Es**” es igual a “**1**” significa que se debe encender el dispositivo llamado “**Lámpara 1**”, en caso contrario que “**Es**” sea igual a “**0**” se debe mandar apagar. Para hacer estas acciones posibles se debe dar la orden de activar el GPIO correspondiente y se evalúa con “**True**” en caso que sea “**Es**” igual a “**1**” y “**False**” si “**Es**” igual a “**0**”, una vez activado se espera 7000ms que corresponden a 7s una vez completada la espera se vuelve al

inicio y se conecta con el servidor para realizar la misma consulta de nuevo para concluir en un bucle infinito con el objetivo que cada 7 segundo realicé una solicitud y saber si alguno de los dispositivo ha cambiado de estado. Leer la sesión de anexo para visualizar el código de programación del ESP-01 y realizar las modificaciones correspondientes para su correcto funcionamiento.

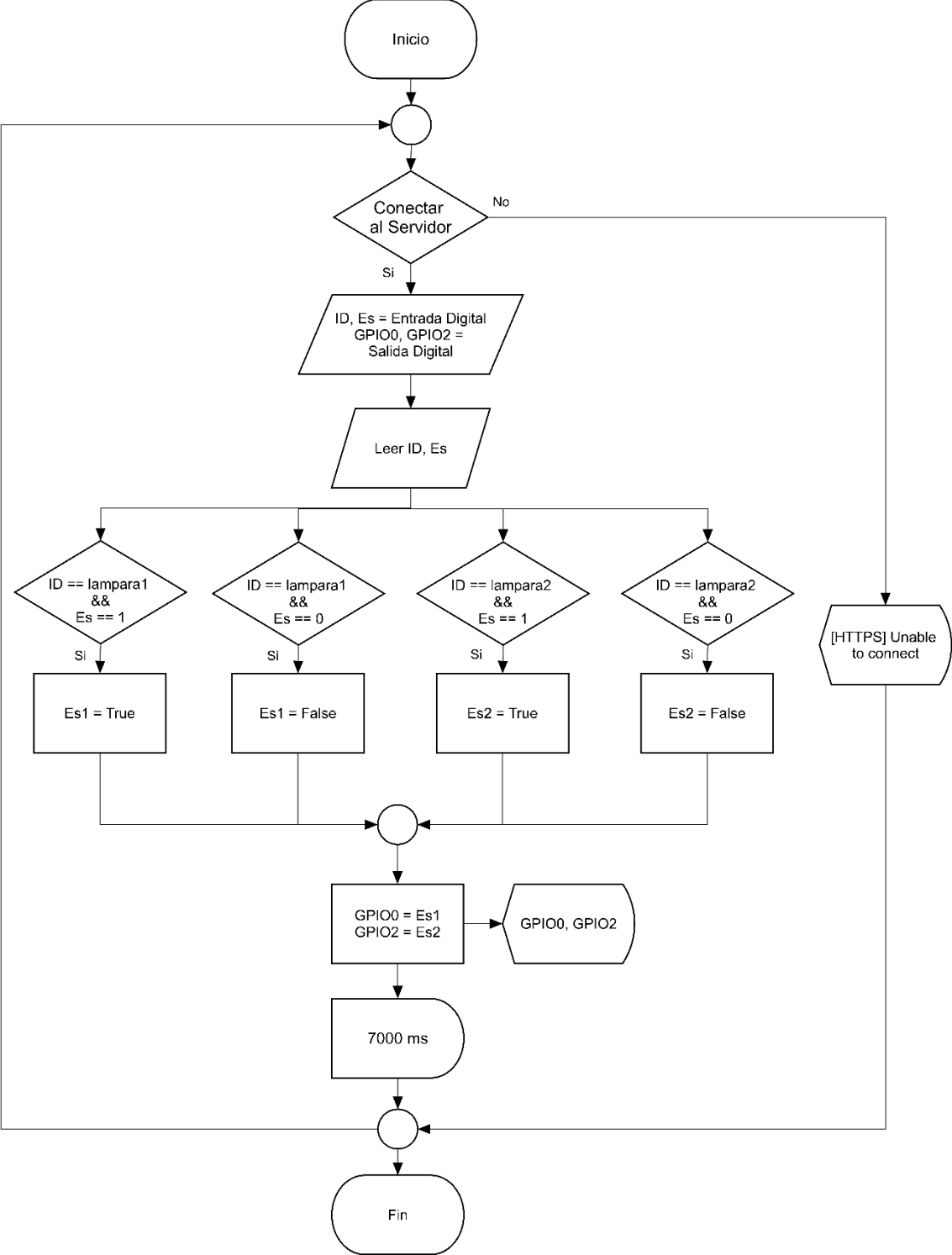


Figura 2: Flujograma Del ESP-01.

## Flujograma del NodeMCU y la Raspberry Pi.

Las tarjetas Raspberry Pi y el NodeMCU v2 al tener más capacidad de administrar dispositivos por contar con GPIO extras el flujograma necesita sufrir un cambio para poner representará esta mayor capacidad y se puede observar en la Figura 3. Para mayor información sobre el código de programación utilizado visitar el anexo número 2 y 3.

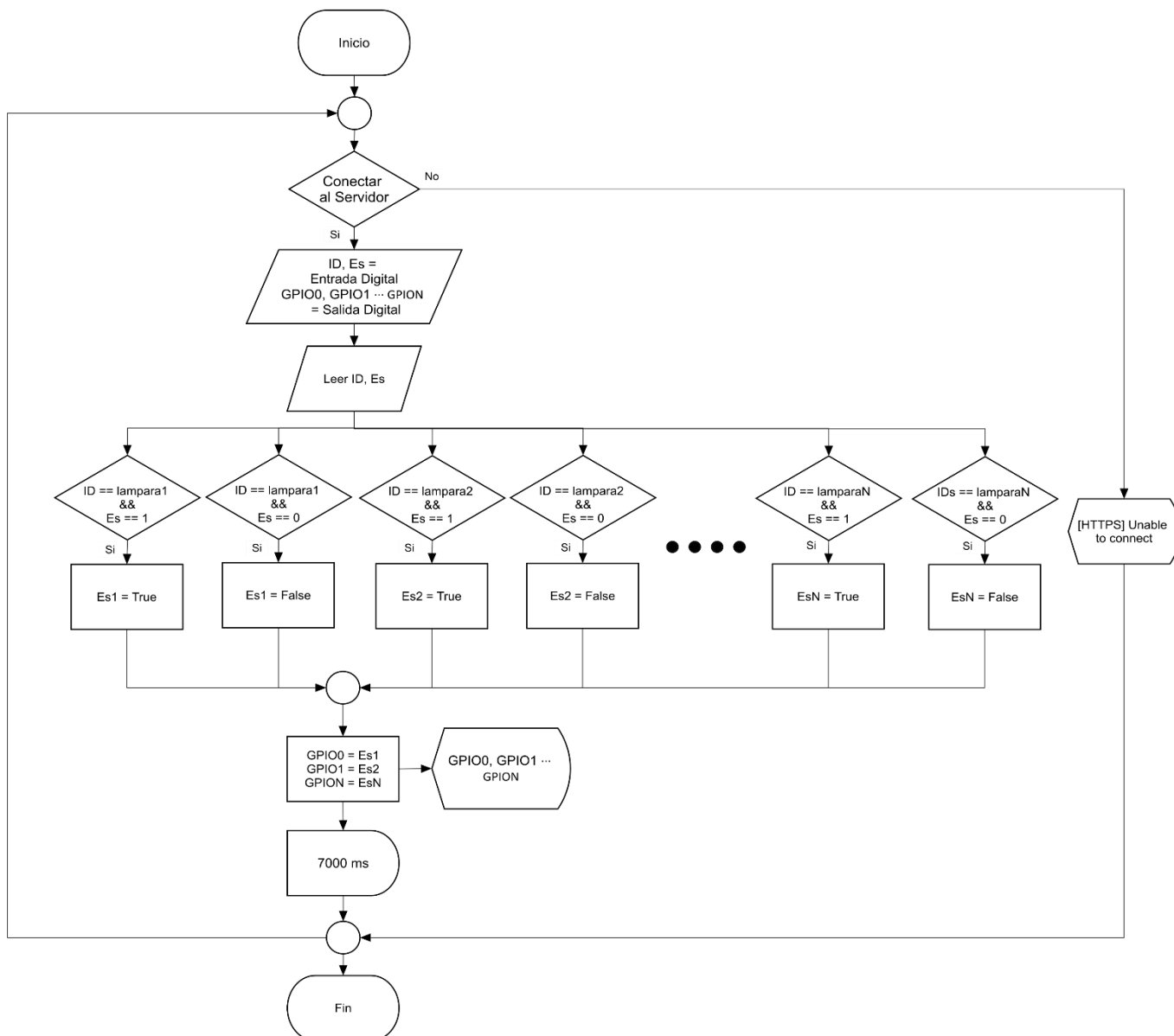


Figura 3: Flujograma Para Raspberry Pi y NodeMCU V2.

La principal diferencia con el diagrama anterior es la representación de los N posibles GPIO disponibles porque cada una de estas placas tiene más de dos pines GPIO por lo tanto las posibles condiciones también tienen a ser capaz de admitir N cantidad de dispositivos con N GPIO de salida correspondiente cada uno. El resto de los procesos son el mismo que se describió en el diagrama anterior en la Figura 2.

# Flujograma de la Plataforma

En la Figura 4 está presente el flujograma del servidor, para la explicación del mismo se dividirá en diferentes partes que para abarcar de forma más concreta cada uno de sus procesos.

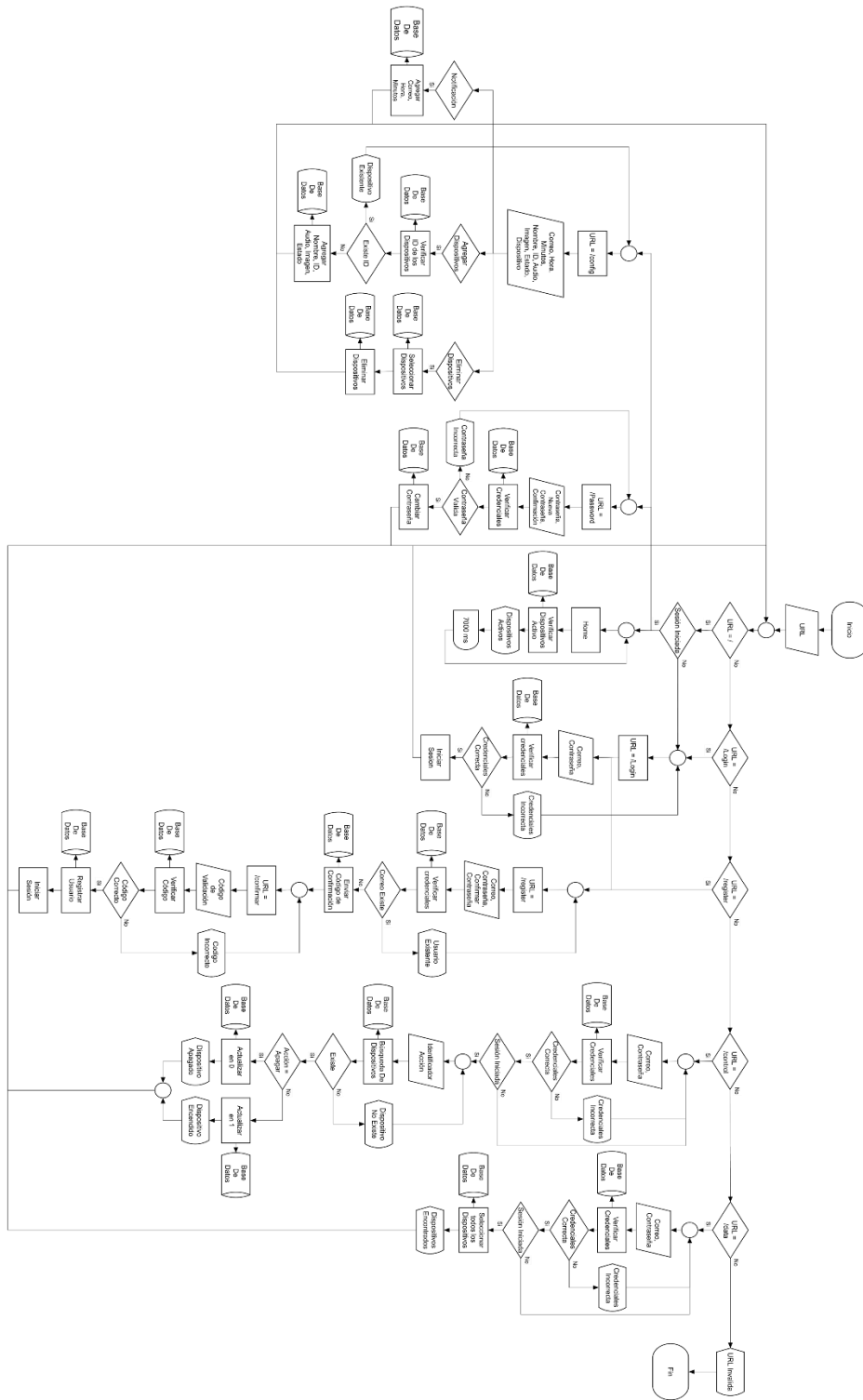


Figura 4: Flujograma Del Servidor

## Flujograma del Home y Login

En la Figura 5 se muestra el inicio del flujograma el primer proceso del programa es recibir la URL del usuario y basada en esta tomar una decisión, si la URL es igual a la raíz de la página el servidor muestra el inicio de esta, pero antes tiene que verificar si el usuario ha iniciado sesión correctamente en el dado caso que el usuario ya tiene una sesión iniciada se envía a la página de inicio **"HOME"** y va hacer una consulta en la base de datos para verificar los dispositivos activos actualmente y mostrarlo en pantalla después de eso va realizar una espera de 7000ms y después volver a realizar la misma consulta en la página **"Home"** de esta manera siempre está mostrando los estados actuales de los dispositivos. En el dado caso que la URL del usuario no sea igual a la raíz o sea la misma, pero todavía no ha iniciado sesión en ese caso el servidor va redirigir a la página de **"Login"** donde el usuario ingresa las credenciales de su cuenta de las cuales se obtiene dos variables que son **"Correo"** y **"Contraseña"**. El servidor realiza una solicitud en la base de datos para validar que las credenciales son correcta y se toma una decisión donde si no es correcto se vuelve al inicio de **"Login"** mostrado un mensaje de error el cual dirá **"credenciales incorrectas"** permitiendo al usuario volver a intentar iniciar sesión, en caso que las credenciales sean correctas se iniciará sesión y el servidor va redirigir a la URL raíz para ir a la página **"Home"** y realizar el proceso anteriormente descrito.

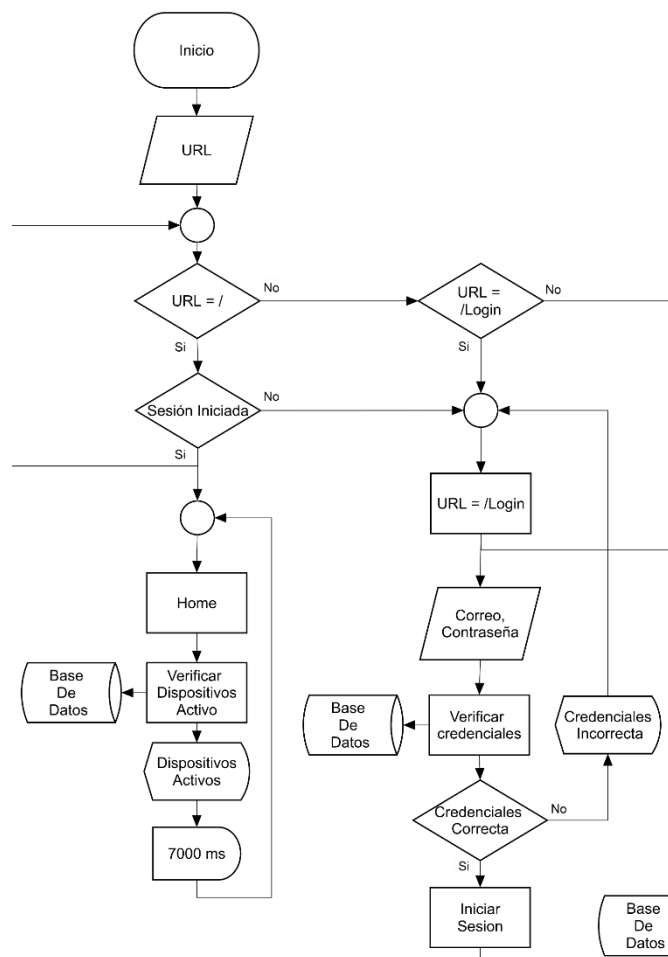


Figura 5: Flujograma Del Home y Login

## Flujograma del Registro

La Figura 6 describe del proceso de cuando la URL es igual a “/register” el servidor enviará a la página de registro donde el usuario podrá crear su cuenta, de manera que se obtienen tres variables que son “**Correo**”, “**Contraseña**” y “**Confirmar Contraseña**”, con los cuales el servidor efectúa una consulta en la base de datos para verificar si el correo ya está registrado en la página en el dado caso que si existe muestra un mensaje que dirá “**Usuario Existente**” y redirija al inicio de la página registro, en dado caso que el usuario no exista enviará un correo con un código de verificación, se guardara en la base de datos para que posteriormente se pueda validar que el código que ingrese el usuario sea correcto.

La URL “/confirmar” se obtiene una variable que es el código de verificación que el usuario ingresa en la página, se realiza una consulta en la base de datos para validar que el código sea el esperado y en dado caso que sea erróneo envía el mensaje “**código incorrecto**”, si el código es correcto se registra el usuario en la base de datos y de forma automática iniciar la sesión y enviarlo al inicio “/”.

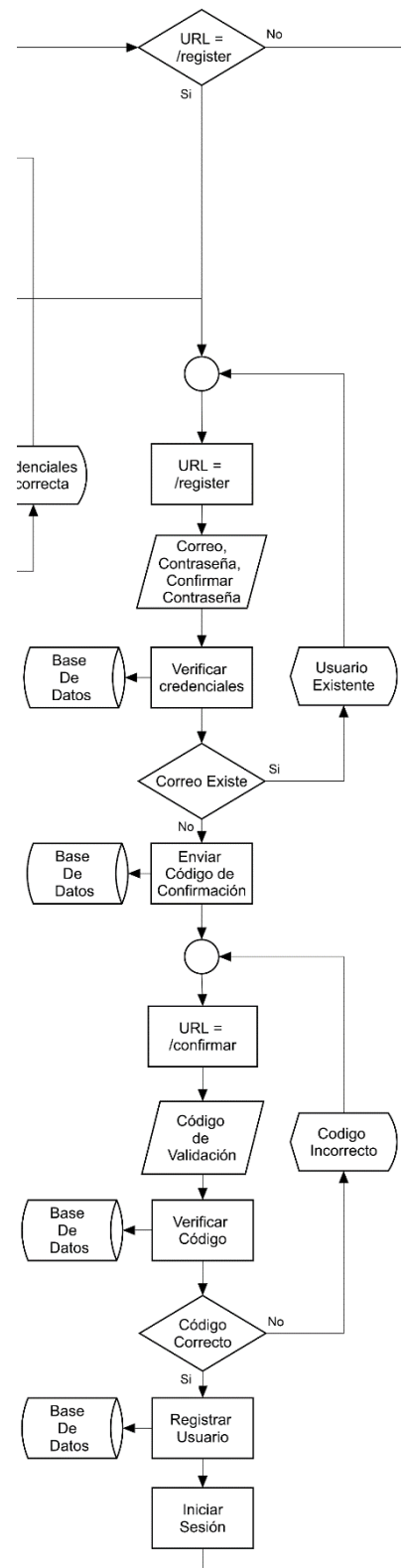


Figura 6: Flujograma Del Registro

## Flujograma de Configuración

Una vez que el usuario ha logrado iniciar sesión podrá acceder a la URL **“/config”** en la Figura 7 se puede visualizar. La función de esta URL es para poder realizar la configuración de sus dispositivos teniendo las opciones de **“Notificación”**, **“Agregar Dispositivos”** y **“Eliminar Dispositivos”**. El flujograma inicia definiendo las variables que se van obtener para lograr las configuraciones, cuando la opción sea **“Notificación”** el proceso que se hará durante esta opción es de agregar a la base de datos el correo donde desea recibir el mensaje de alerta, la hora y minutos para enviar el mensaje de recordatorio. dado el caso que la opción sea **“Agregar Dispositivos”** lo primero que se debe hacer es verificar si el ID que está ingresando el usuario no está repetido en la base de datos, en caso que el ID exista se redirija al usuario a configuraciones nuevamente con el mensaje de alerta **“Dispositivo Existente”**, si el ID no está repetido se agrega los parámetros del nuevo dispositivo de los cuales son **“Nombre”**, **“ID”**, **“Audio”**, **“Imagen”** y **“Estado”**, todos estos valores se deben ingresar en la base de datos. La opción **“Eliminar Dispositivos”** permite borrar de la base de datos todos aquellos dispositivos agregados previamente para hacerlo se presenta al usuario una lista donde podrá seleccionar el dispositivo que desea eliminar, se envía una solicitud Sql a la base de datos para eliminar el dispositivo seleccionado. Una vez que las configuraciones hayan terminado el programa enviara al usuario a la página de inicio.

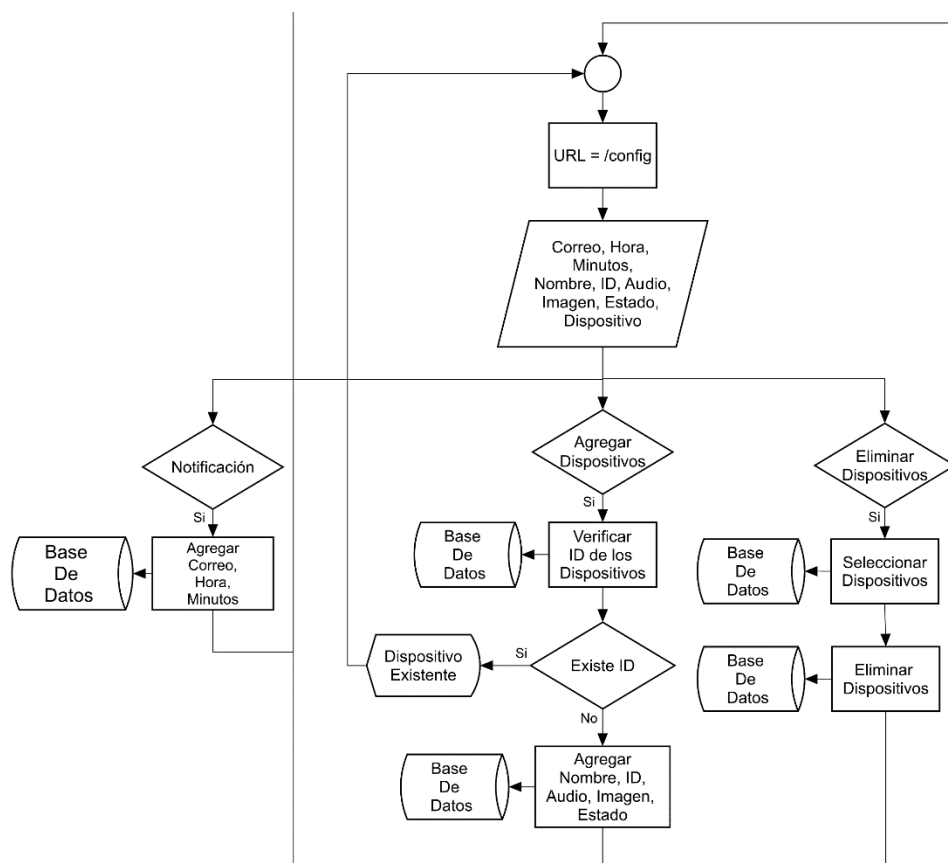


Figura 7: Flujograma De Configuración.

## Flujograma del Password

Existe otra URL que el usuario solo puede acceder cuando inicie la sesión y es “/Password” tal como se puede visualizar el flujograma en la Figura 8. El objetivo de esta función es de otorgar al usuario la facilidad de poder cambiar la contraseña actual, lo primero es configurar tres parámetros que serán: “**Contraseña**” es la contraseña con la cual actualmente se inició sesión, “**Nueva Contraseña**” donde se escribirá la nueva contraseña para remplazar la actual y “**Confirmación**” se debe de ingresar la nueva contraseña para validar que no ha sido escrita mal, se efectúa una consulta a la base de datos para verificar que ha escrito correctamente la contraseña actual si no es igual se enviara un mensaje de error diciendo “**Contraseña Invalida**” por lo tanto el usuario será enviado de vuelta a “/Password” y en dado caso que sea correcta la contraseña se manda una consulta a la base de datos para actualizar el parámetro de contraseña cambiándolo por la nueva contraseña, una vez que termine la consulta se redirigirá al usuario a la página de inicio.

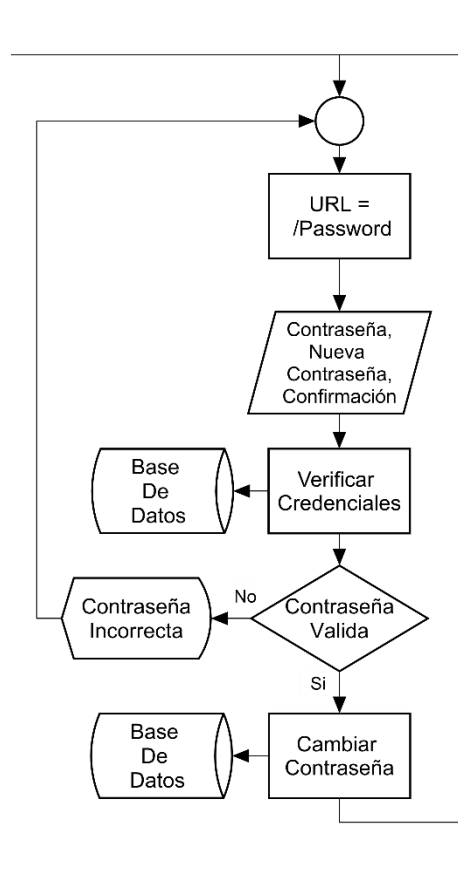


Figura 8: Flujograma Del Password.



## Flujograma del Control

En la Figura 9 se tiene presente los pasos a seguir cuando la URL es “/control”. Esta URL es diferente a las demás porque es la URL que usará Google Assistant para comunicarse con el servidor. El primer paso presente en el flujograma son las variables de credenciales debido que esta URL está destinada para hablar con Google Assistant se tiene que crear un inicio sesión dedicado para esta, la descripción de los procesos de iniciar sesión son los mismo de los procesos explicado anteriormente en “/login”.

Una vez que Google Assistant ha logrado iniciar sesión recibe dos variables que son “Identificador” donde está el nombre del dispositivo y “Acción” que es la orden que se debe efectuar, por ejemplo, puede ser “Apagar” o “Encender”. Se realiza una búsqueda en la base de datos para verificar que ese dispositivo existe o no. Si no existe se envía el mensaje “Dispositivo no Existe” para que Google Assistant le diga al usuario que se ha equivocado, en cambio que exista el dispositivo el servidor tomará una decisión en base a la variable “Acción”. Si es “Apagar” se debe actualizar la base de datos con el valor “0” y enviar el mensaje notificando que “Dispositivo Apagado” en caso contrario significa que se tiene que encender el dispositivo, se actualiza la base de datos a “1” respondiendo con el mensaje “Dispositivos Encendidos”. Una vez terminada las modificaciones la comunicación acaba.

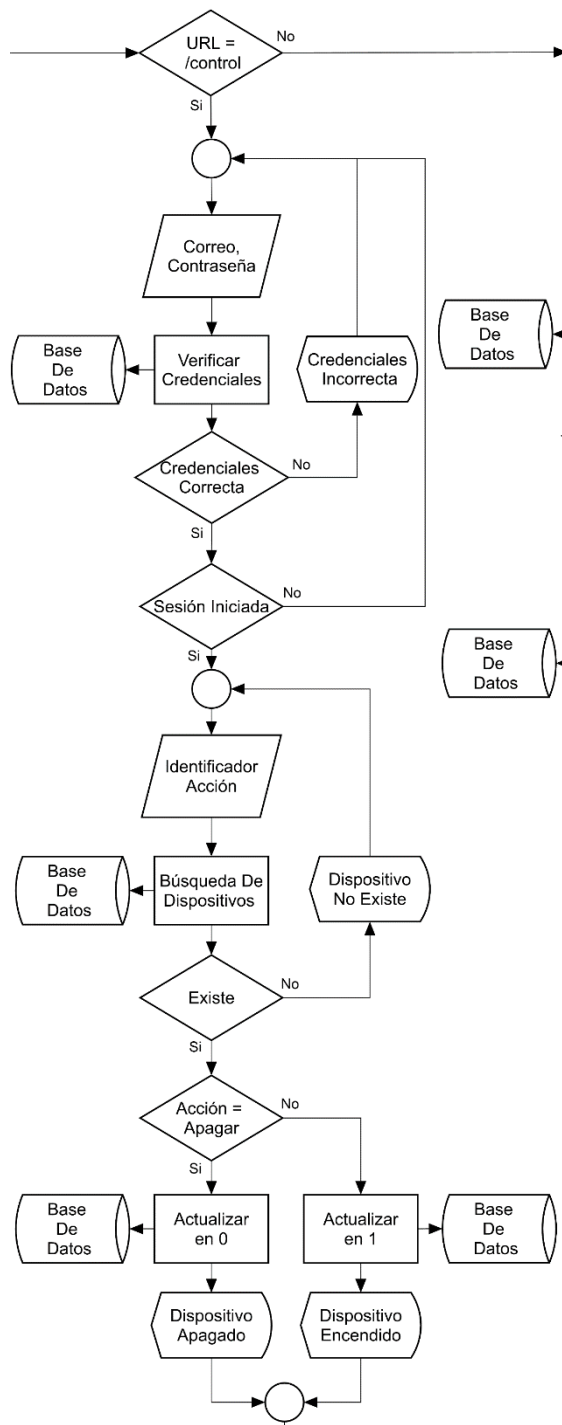


Figura 9: Flujograma Del Control

## Flujograma de Data

En la Figura 10 se presenta la URL “/data” y cuando la URL no ha sido ninguna de las anteriores explicadas, en ese caso muestra un mensaje que dirá “URL Invalida” y será el fin de todo el proceso. La URL “/data” es similar a la URL “/control” porque es una URL dedicada para que las tarjetas de desarrollo se puedan comunicar a través de esta al servidor. Debido que es una URL dedicada se debe integrar otro inicio sesión el mismo que hemos explicado anteriormente en la URL “/login”. Una vez iniciada la sesión del usuario realiza una consulta a la base de datos en la cual se manda a seleccionar todos los dispositivos registrados y mostrando un mensaje que contendrá “Dispositivos Encontrados”, de manera que las tarjetas de desarrollo van a recibir un listado con dos parámetros, “*identificador*” que es una lista con los nombres de los dispositivos y “*Estado*” que estará evaluado con “0” en el caso que deba apagarse o “1” si debemos encenderlo.

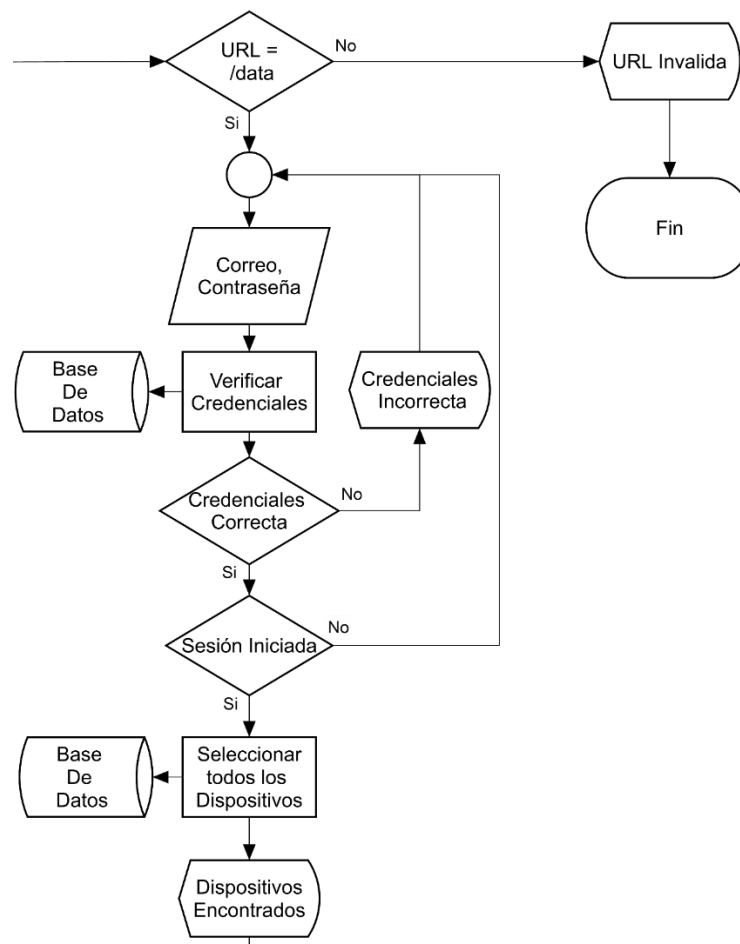


Figura 10: Flujograma De Data.

## Tercer capítulo: Fabricación y Pruebas del Hardware

Este tercer capítulo se trata de todo el tema relacionado con el proceso de fabricación del dispositivo, cada paso que se realizó para llegar al prototipo final de cada una de las tarjetas de desarrollo y las pruebas necesaria para verificar el correcto funcionamiento de todo el sistema completo. Todo este capítulo está reflejado en el diagrama de bloque mostrado en el capítulo anterior con el nombre correspondiente a “**sistema de control**”.

### Simulación del Circuito de Acondicionamiento

Un sistema de acondicionamiento es una parte esencial del prototipo debido que dentro de este existe una parte de control que trabaja con voltaje DC la cual ira conectada directamente a la placa de control y de forma opuesta existe la parte de potencia a la cual trabaja con voltaje AC de 120V, si no existirá un sistema de acondicionamiento que divida estas dos partes la tarjeta de control se quemaría por el alto voltaje por ese motivo una parte indispensable en el sistema.

La Figura 11 presenta el circuito de acondicionamiento en la simulación de manera que se puede dividir en dos partes para su comprensión las cuales son “**Sistema De Control**” y “**Sistema De Potencia**”. El sistema de control será la parte que ira directamente conectada a la tarjeta de desarrollo la cual está representada por un “**LogicState**” en la imagen, que ira conectado a un transistor para activar o desactivar el MOC3021 de esta forma la tarjeta de desarrollo podrá controlar los dispositivos del usuario estando totalmente aislada de los 120V AC de la corriente comercial evitando daños a la placa. El sistema de potencia es la parte que está directamente conectada a los dispositivos del usuario es decir que esta parte se trabaja con 120V AC y para lograr activar o desactivar los productos del usuario, se necesita trabajar con esta corriente por eso se utilizó un Triac que garantiza el poder abrir o cerrar el circuito para que funcione el dispositivo cuando se desee.

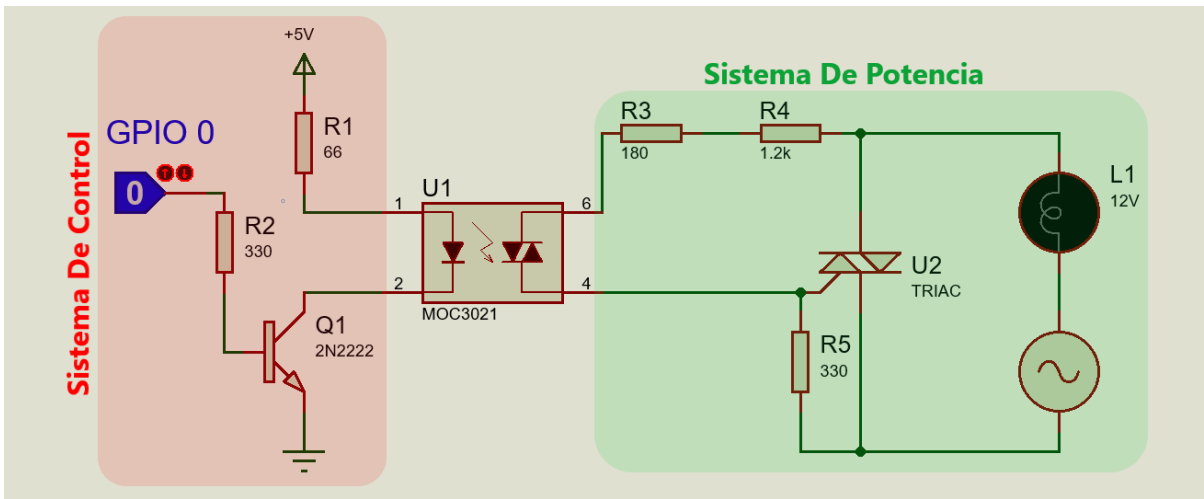


Figura 11: Circuito De Acondicionamiento.

En la Figura 12. esta el mismo circuito anteriormente explicado solo que se tendrán dos circuitos en paralelo con el objetivo que cada uno del circuito sea para cada GPIO de la tarjeta de desarrollo en el caso del ESP-01 solo tendremos dos GPIO en esta circunstancia por lo tanto el circuito de la imagen sería perfecto para esta tarjeta pero para el caso de la Raspberry Pi y del Nodemcu v2 que tienen más GPIO se tendría que agregar más circuito colocado en paralelo uno del otro para lograr controlar más dispositivos y para sacar más provecho de todos los GPIO que estas tarjeta poseen.

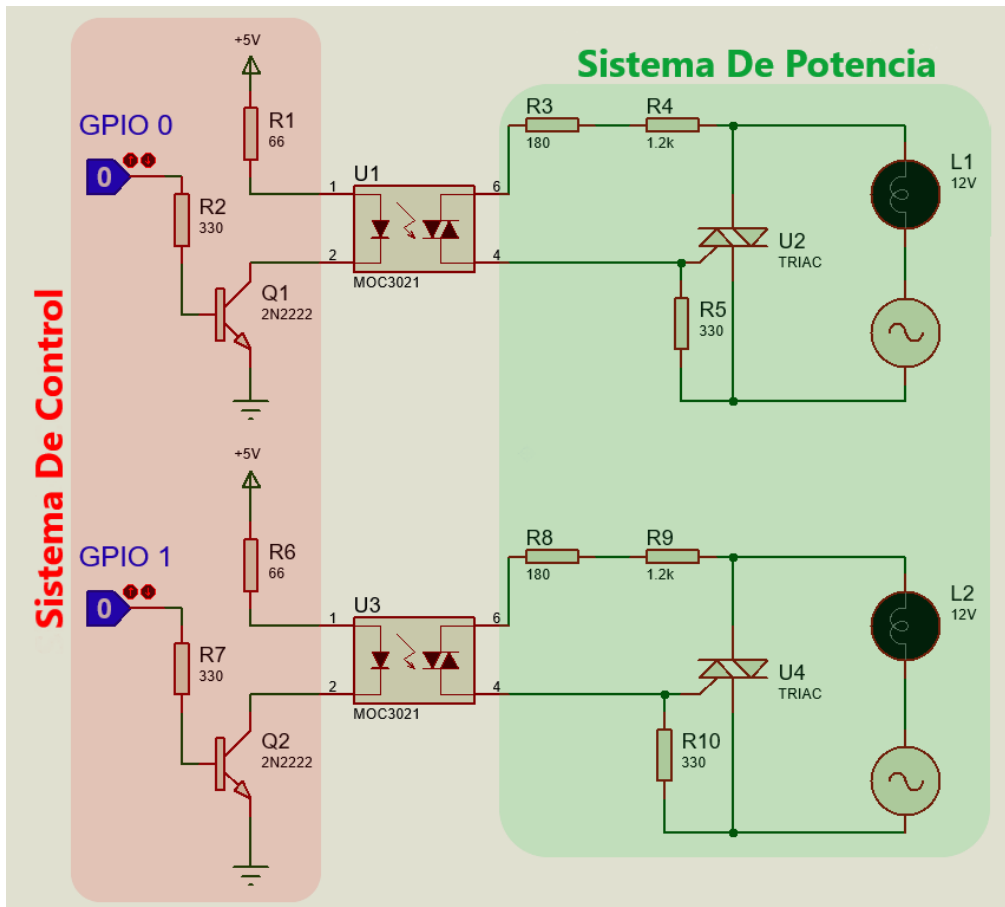


Figura 12: Circuito De Acondicionamiento Para Dos GPIO.

## Fabricación del Dispositivo

En esta sección se hablará en detalle paso a paso toda la elaboración de cada uno de los prototipos para cada tarjeta de desarrollo, dando inicio con el diseño del circuito modificado para adecuarse con las conexiones eléctricas comerciales, el diseño de la PCB, todo el proceso de la elaboración de la PCB en baquelita y el montaje de los componentes electrónicos en la placa hasta con concluir con el ensamblaje final en su caja.

### Raspberry Pi

En la figura 13. esta plateado el diseño del circuito para el prototipo a seguir de la Raspberry Pi, este prototipo es capaz de controlar tres dispositivos, la Raspberry Pi posee la capacidad de controlar más de tres, pero con fines demostrativo se ha decidido que para esta presentación solo se utilizaron tres debido que el único cambio significativo que sufre el circuito si se deseara agregar más solo sería copiar todo un módulo del circuito de acondicionamiento y conectarlo a los demás de forma paralela. El sistema de control será alimentado por los propios pines 5V que proporciona la propia tarjeta, esto con la finalidad de reducir costo y tamaño en una fuente de alimentación.

Debido que la Raspberry Pi es la encargada de alimentar el circuito por medio de cable es necesario calcular la distancia máxima que se podrá realizar las conexiones, para calcular eso se utilizó de la fórmula de la caída de voltaje que es la siguiente ecuación:  $VD = 2 \times K \times I \times L/CM$ , donde K = “constante de corriente continua”, I = “corriente”, L = “longitud en pies” y CM = “área de sección transversal” según [14]. Despejando la ecuación se tiene como resultado que la ecuación para calcular la distancia máxima es:

$$L = VD \times CM / 2 \times K \times I \quad \text{Ecuación 1}$$

El valor de  $VD = 1.65V$ , el  $CM = 0.509\text{kcmil}$  que es el valor del área transversal del AWG 23 que corresponde a un Cat6, para calcular el valor del área se utilizó el valor según mostraba en la tabla de [15], pero para convertir kcmil a milésimas circulares se tiene que multiplicar  $\text{kcmil} \times 1000$  entonces  $CM = 509$ ,  $K = 12.9$  para un conductor de cobre y  $I = 0.50\text{mA}$  que es la corriente que puede entregar la Raspberry Pi, una vez sustituyendo todos los valores se tiene como resultado:

$$L = 1.65V \times 509 / 2 \times 12.9 \times 0.50\text{mA}$$
$$L = 651\text{pies}$$

$$L = 198m$$

La distancia máxima de conexión con una caída del voltaje del 50% es 198 metros, la cual es una distancia bastante considerable para realizar cualquier instalación.

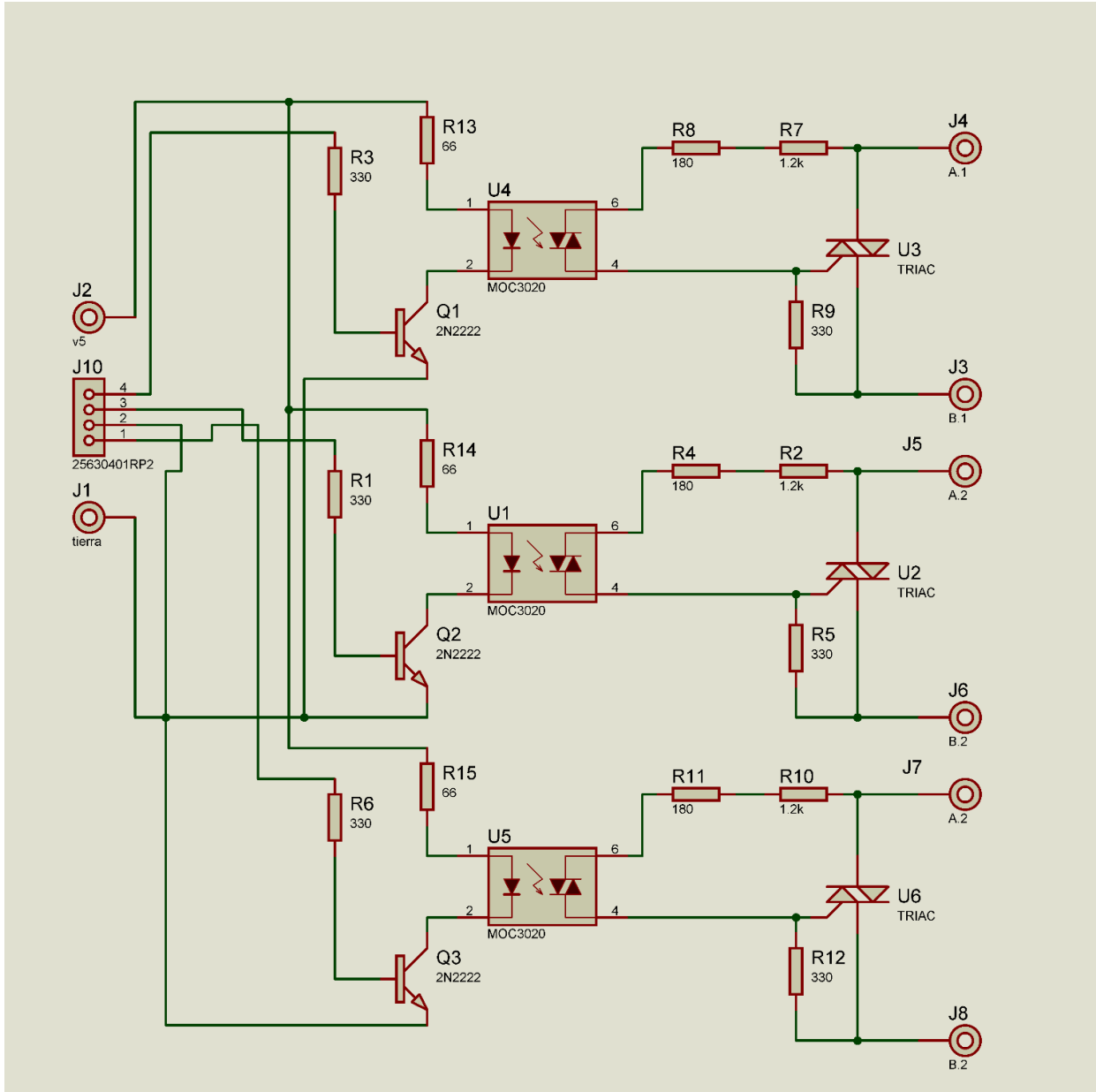


Figura 13: Circuito Para Raspberry Pi.

En la figura 14. está presente el diseño de la PCB creado a partir del circuito de la figura 13. para la Raspberry Pi, la cual se puede apreciar la instalación final que tendrá cada dispositivo sobre la placa en donde toda la parte de control se encuentra colocada en la parte superior y toda la parte de potencia se encuentra ubicada en la parte inferior esto con la finalidad de mantener lo más alejado posible toda la parte de control de la parte de potencia y el tamaño final que tendrá toda la placa será de 8 cm de ancho x 6 cm de alto.

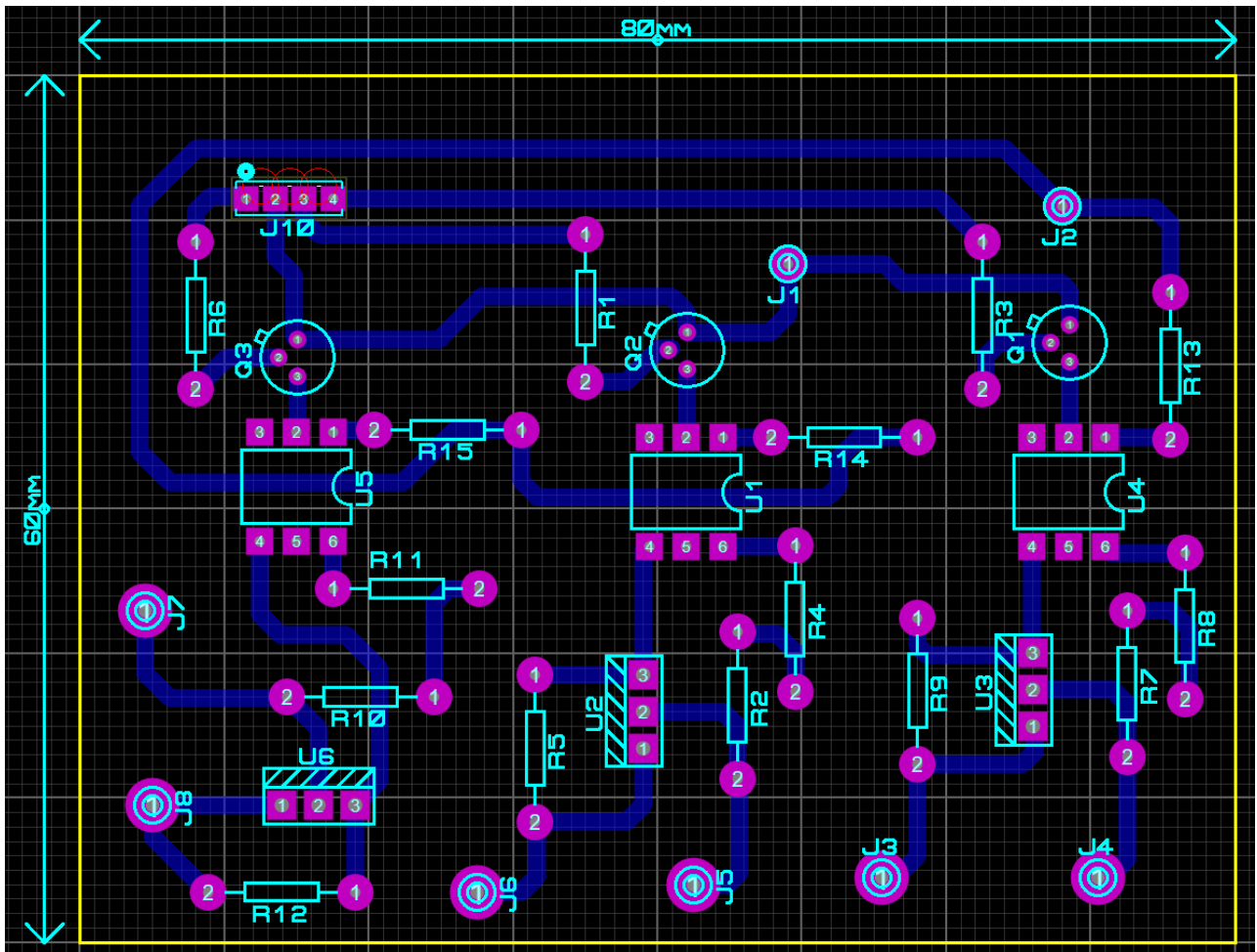


Figura 14: Diseño de la PCB Para Raspberry Pi.

Una vez concluido el diseño de la PCB se da inicio con el proceso de fabricación en físico, el primer paso es imprimir el circuito PCB con una impresora láser en papel satinado como se puede observar en la figura 15. Debe ser con este tipo de impresora y ese tipo de papel para que al momento de planchar la PCB en la baquelita la tinta logre adherirse sin problema en cobre.

En la figura 16. se puede observar cómo sería el diseño de la PCB con todos los componentes electrónicos montados sobre ella con el uso de un modelo 3D.

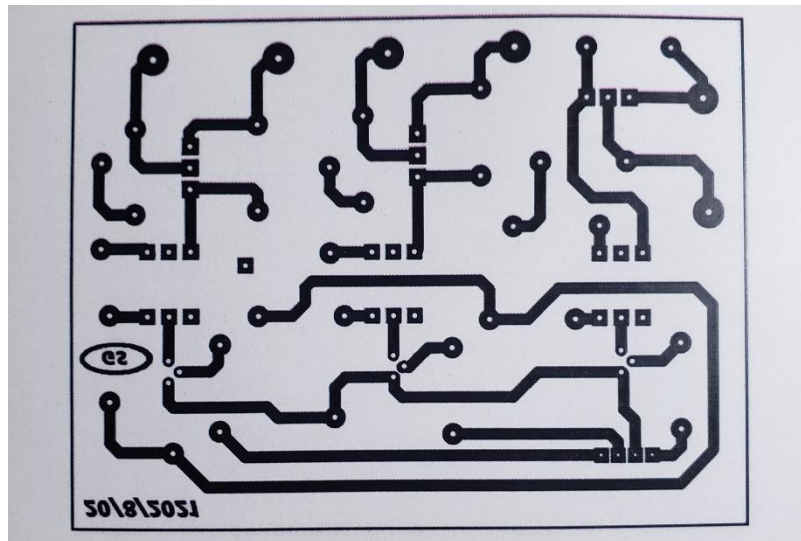


Figura 15: Impresión De La PCB.

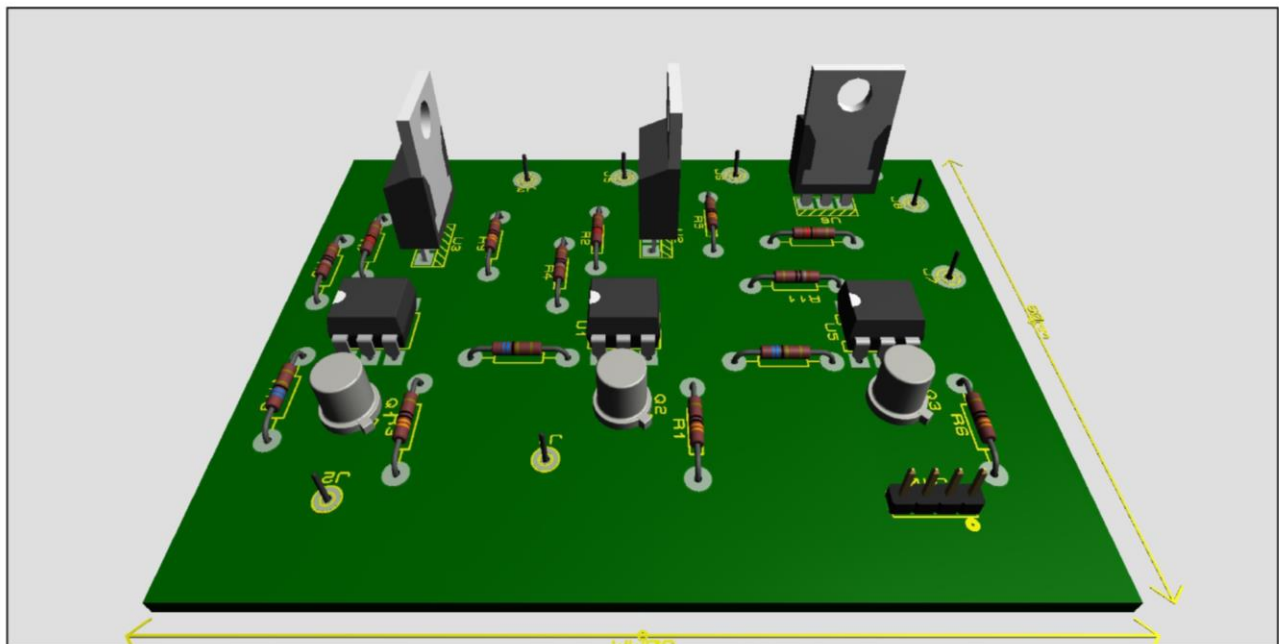


Figura 16: Diseño En 3D de la PCB Para Raspberry Pi.



El siguiente paso a llevar a cabo fue el planchado del circuito en la baquelita, primeramente, se tiene que elegir una placa que sea del tamaño del circuito y que sea una baquelita del tipo virgen sin perforaciones tal como en la figura 17. Luego de tener la placa con la cual se va a trabajar se tiene que poner la hoja de papel satinado sobre la baquelita y luego poner la plancha caliente sobre ella de forma que lentamente se debe frotar con el objetivo que todo el papel se caliente por igual de forma que la tinta logre pegarse sobre el cobre, concluido ese paso se metió la baquelita con el papel satinado en agua para retirar lentamente el papel y verificar que la tinta se adherido por completo como es el caso de la figura 18.



Figura 17: Baquelita Virgen.

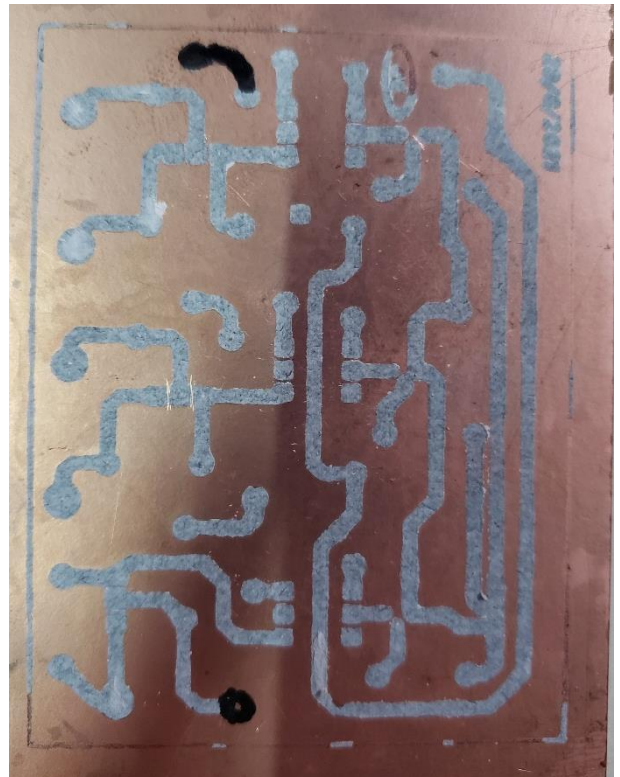
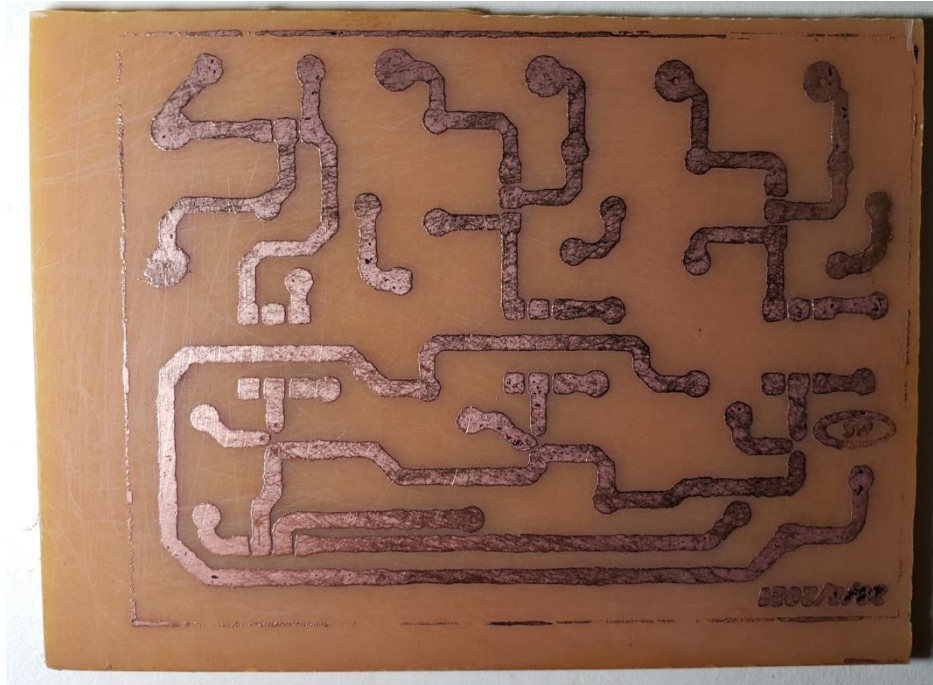


Figura 18: PCB Con El Circuito.

Ahora bien, para concluir el primer paso de crear el circuito en cobre se debe tomar la baquelita que está en la figura 18. Y exponerla ante una mezcla de ácido muriático con agua oxigenada, para realizar esta mezcla debe seguir la siguiente regla, por cada unidad de ácido muriático debe agregarse dos unidades de agua oxigenada. Se tiene que exponer la baquelita ante el ácido con el objetivo que este remueva todas las partes de cobre que no va a necesitar dejando de esta manera el circuito formado por el cobre restante debido que el ácido no logra eliminar las parte cubiertas por la tinta tal como se observa en la figura 19. Ahora que tiene el circuito en cobre se debe a proceder a realizar las perforaciones de dónde irán los componentes electrónicos con ayuda de un taladro y una broca muy pequeña.



*Figura 19: Circuito En Cobre.*

Continuando con el diseño del prototipo se realizó la compra de los componentes electrónicos basado en la simulación realizada en la figura 13. En la cual se adquirió tres unidades de Triac del modelo BT139X con capacidad de 120V/16A, tres unidades de MOC3021, tres transistores PN2222A, dos terminales de bloques de 3 pines cada una y todas las resistencias con las unidades correspondiente a la simulación para garantizar que todo funcione correctamente. Se puede observar todos los componentes en la figura 20. Aparte de los componentes que estaban representados en la simulación era necesario utilizar unos extras para realizar un correcto sistema de conexiones eléctricas en los cuales están: los capuchones siendo utilizado para los empalmes entre los cables, un conector hembra del tipo C13 para realizar con facilidad la conexión al sistema eléctrico comercial y un conector Jack Rj45 que fue utilizado para conectar el GPIO de la Raspberry Pi con el circuito de control de forma segura, tal como está representado en la figura 21.

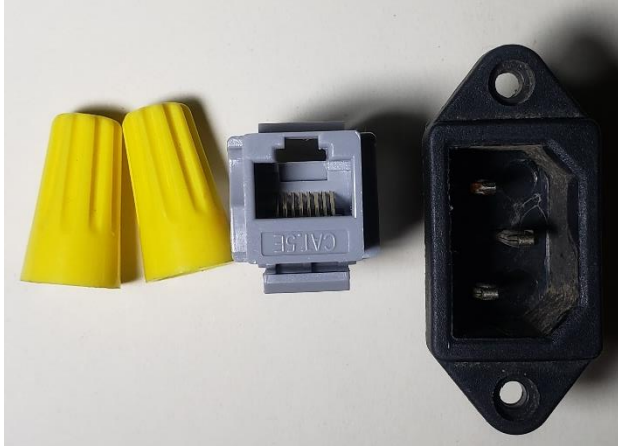


Figura 21: Conector Tipo C13, Capuchones Y Jack Rj45.

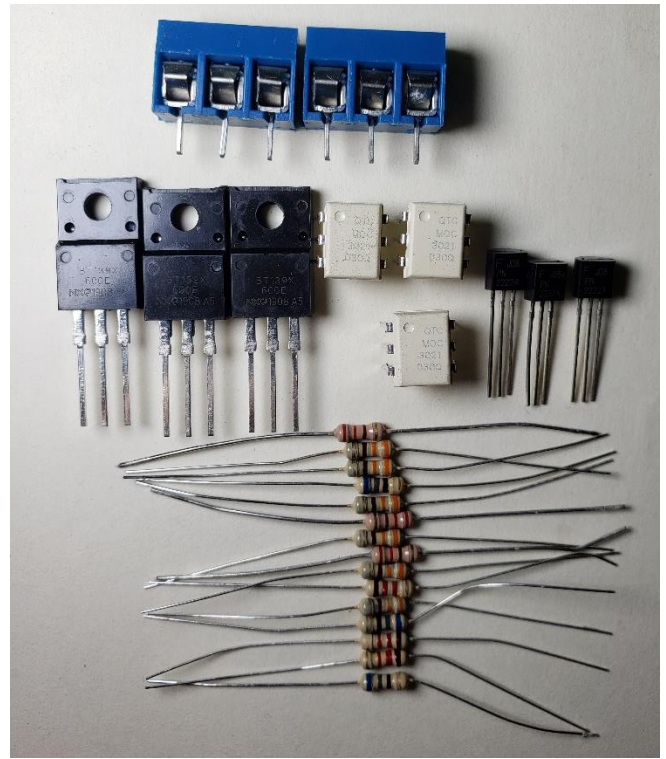


Figura 20: Componentes Electrónicos.

El siguiente paso a seguir fue realizar el montaje de todos los componentes electrónico en la baquelita recién creada, se debe colocar cada componente en su lugar correspondiente como esta presentado en la figura 16. Ya que muestra la posición de cada uno con una vista área por lo tanto lo se puede usar como guía para componentes complicados de colocar correctamente como son los Triac y los Moc. Una que cada pieza este donde corresponde se procede a soldar en la baquelita hasta que todos estén perfectamente colocados tal como en la figura 22 y la figura 23 con una perspectiva superior.

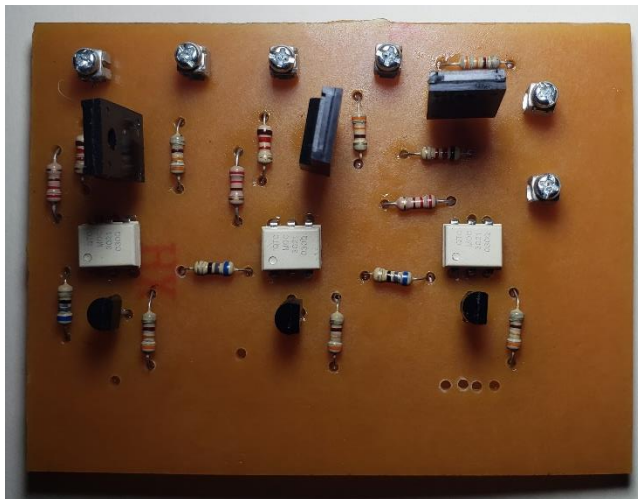


Figura 23: Baquelita Terminada.



Figura 22: Soldadura De La Terminada.

En la última etapa del prototipo está el montaje final en el cual se ha hecho elección de la caja en la cual será integrada la tarjeta como se podrá observar en su forma exterior en la figura 25 y su parte interior en la figura 24. Se ha elegido este tipo de caja ya que el material con el que está hecha es plástico por lo tanto evita cualquier cortocircuito que se pueda ocasionar por accidente además que estas cajas poseen una certificación anti-salpicaduras y anti-polvo IP55 por lo tanto mantiene el circuito lo más aislado posible ayudando a su preservación.

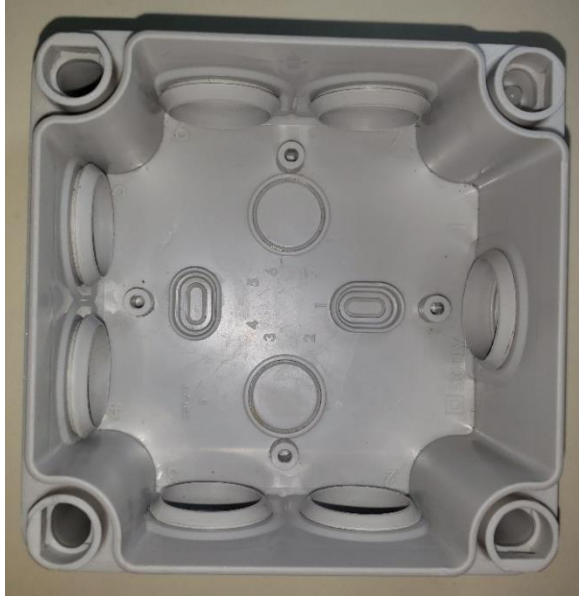


Figura 24: Parte Interior De La Caja



Figura 25: Parte Exterior De La Caja.

Una vez terminada toda la instalación de la tarjeta dentro de la caja con todas las conexiones exteriores en su respectivo lugar se verá como en la figura 26.

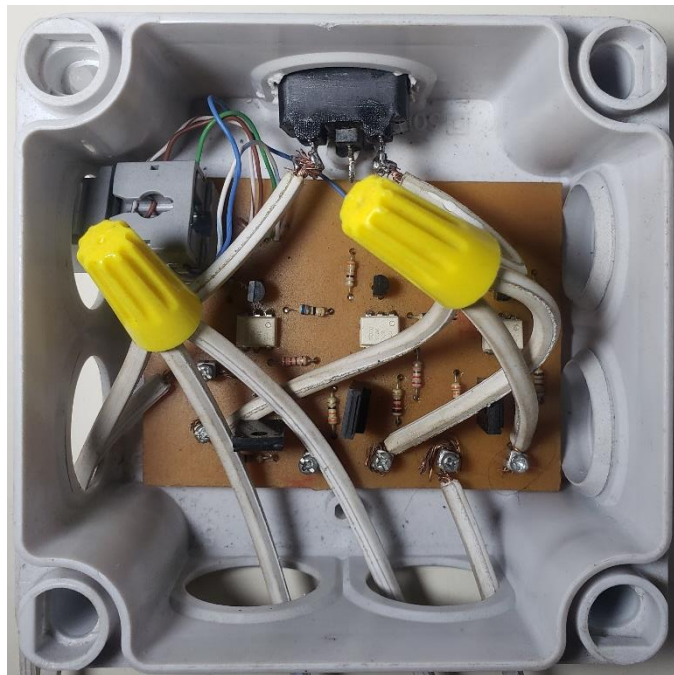


Figura 26: Prototipo Finalizado.

## ESP-01

Esta sección abarca el proceso de elaboración del prototipo de la tarjeta ESP-01 en su gran parte el proceso es similar al previo visto pero cambian pequeños detalles en el diseño y capacidad de control de la misma, en la figura 27. esta presente la simulación del circuito a seguir en este prototipo en primera instancia la diferencia notable con respecto al diseño de la Raspberry Pi es la capacidad de control, este modelo solo será capaz de controlar dos Triac debido que el modelo ESP-01 solo posee dos pines GPIO de control por el tamaño reducido de la placa, otra diferencia con el anterior prototipo es que la tarjeta de control esta alimentada directamente desde el circuito por lo tanto debemos agregar tres resistencias para realizar la caída de voltaje debido que el ESP-01 se alimenta con 3.3V y con una resistencia de 10K conectada al pin de **CH\_PD**<sup>1</sup>, haciendo caso omiso a esas dos diferencia el resto de los componentes son los mismo en ambos prototipos.

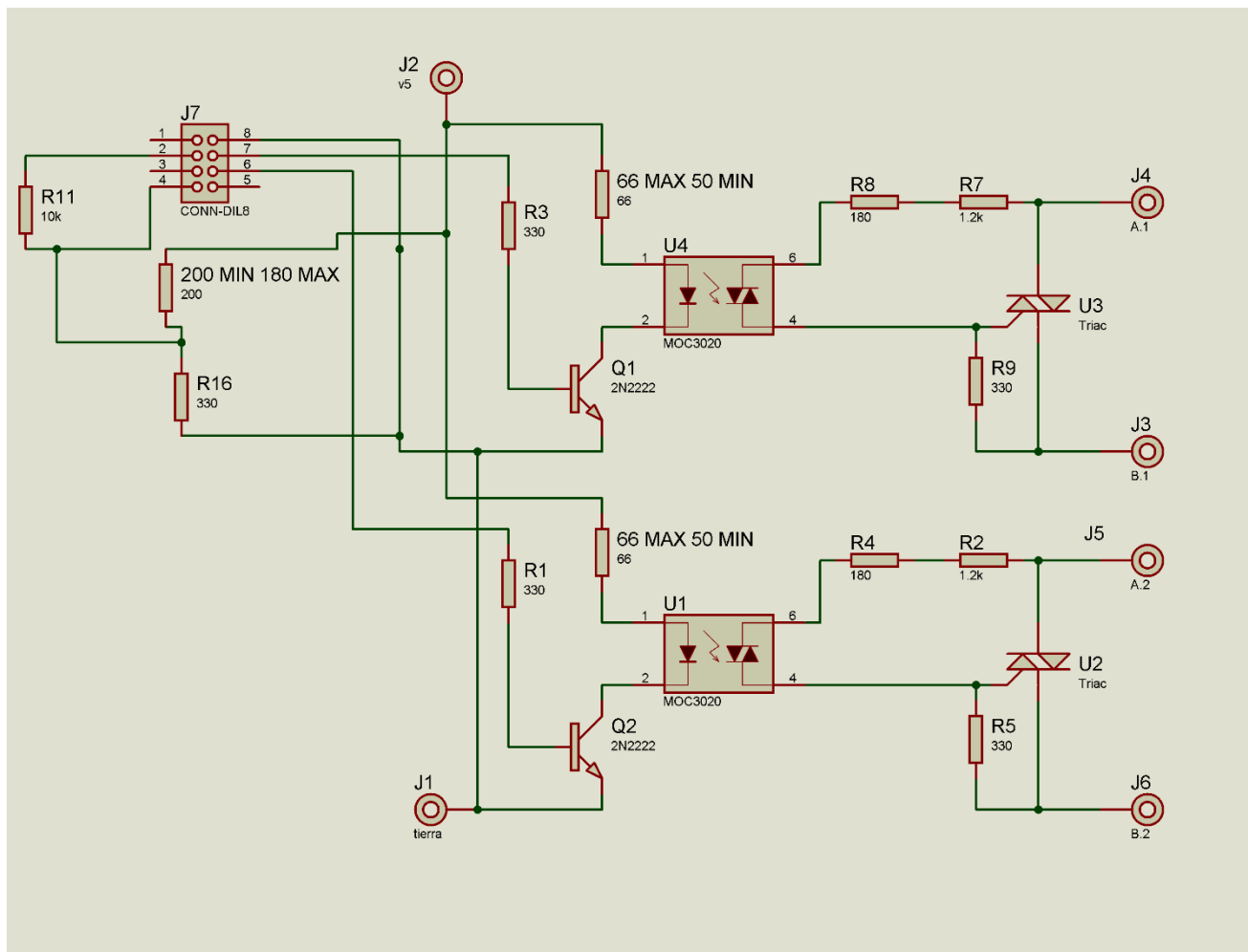


Figura 27: Circuito Para ESP-01.

<sup>1</sup> Chip Power-Down

Continuando con el trabajo en la fabricación en PCB el siguiente paso es el diseño que tendrá está en físico tal como se puede apreciar en la figura 28. Este diseño a diferencia del anterior es un poco más compacto con unas medidas de 6cm x 6cm siempre conservando la separación de la parte de control ubicada en el lado superior de la parte de potencia ubicada al lado opuesto, también posee los 8 pines por si desea instalar directamente el módulo ESP-01 a la placa o simplemente conectarlo por medio de cable.

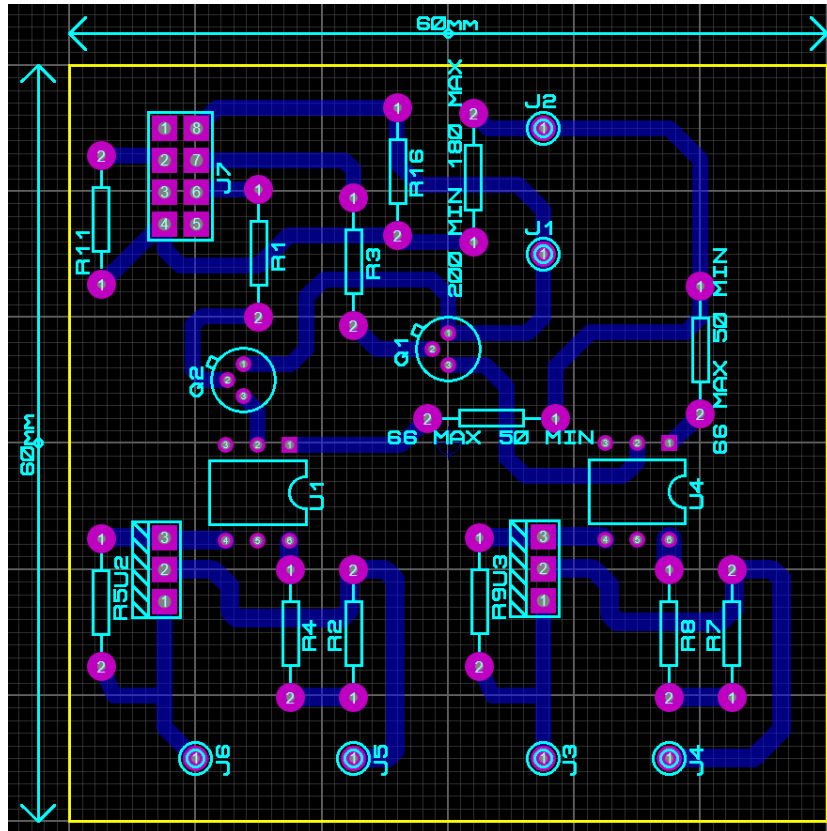


Figura 28: Diseño de la PCB Para ESP-01.

La figura 29. permite dar una idea de cómo se verá la placa en un modelo 3D para poder observar la ubicación de cada componente por si lo seamos modificar.

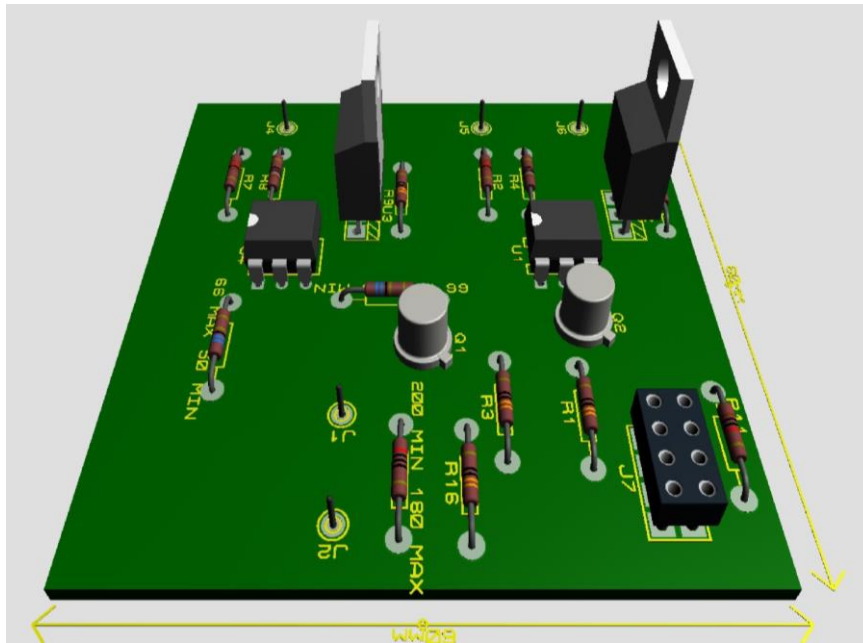


Figura 29: PCB En 3D

Una vez terminado el diseño que tiene la PCB se procede a los pasos de fabricación que serán los mismos pasos que el prototipo anterior, lo primero que debe hacer es imprimir en una impresora láser con papel satinado el circuito como en la figura 30. procede a colocar la hoja de papel satinada sobre la baquelita que debe tener un tamaño similar al circuito para que sea más fácil de planchar, la baquelita que se utilizó se puede apreciarla en la figura 31. Una vez finalizado el planchado del circuito se tiene que meter la placa en agua para remover el papel y verificar que toda la tinta haya logrado pegarse correctamente en el cobre debe quedar como en la figura 32.

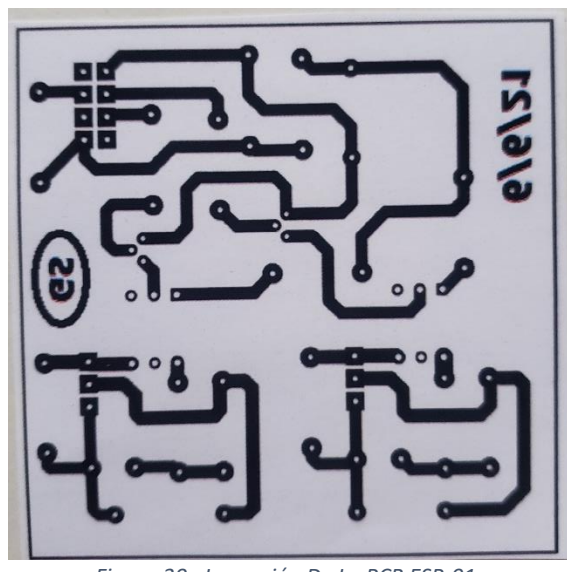
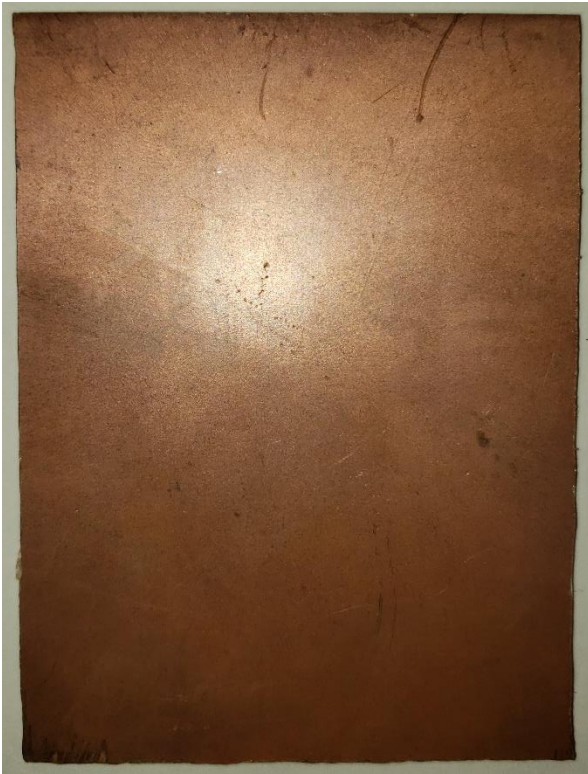


Figura 30: Impresión De La PCB ESP-01.



*Figura 31: Baquelita Virgen.*



*Figura 32: Circuito En La Baquelita.*

El siguiente paso es introducir la baquelita de la figura 32. en ácido muriático con agua oxigenada con el mismo objetivo de la vez anterior que es remover las partes de cobre que no se necesitan y solo debe quedar el circuito del prototipo en cobre, una vez expuesto la placa al ácido debe quedar algo similar a la figura 33. Debe comprobar que el ácido no se comiera cobre del circuito y si todo es correcto continua la parte de perforación de la tarjeta, con el uso de un taladro se perfora con cuidado donde irán soldado los componentes electrónicos tal como aparece en la figura 34.



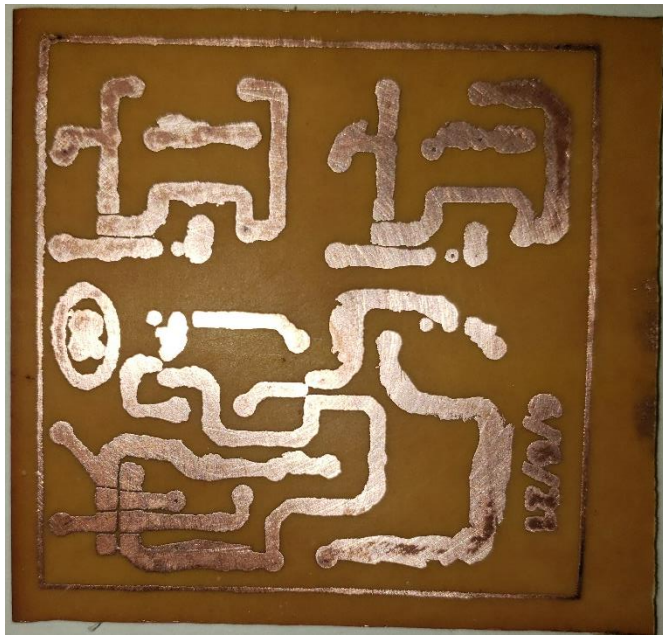


Figura 33: Circuito En Cobre.



Figura 34: Baquelita Perforada

En el apartado de componentes electrónico y conectores este prototipo poseen los mismos modelos que el prototipo anterior a excepción que incluyen tres resistencias más para el voltaje de alimentación del ESP-01 y debido que este prototipo lleva integrada la tarjeta no es necesario el uso del conector Jack Rj45, todos los componentes se pueden apreciar en la figura 35 y 36.

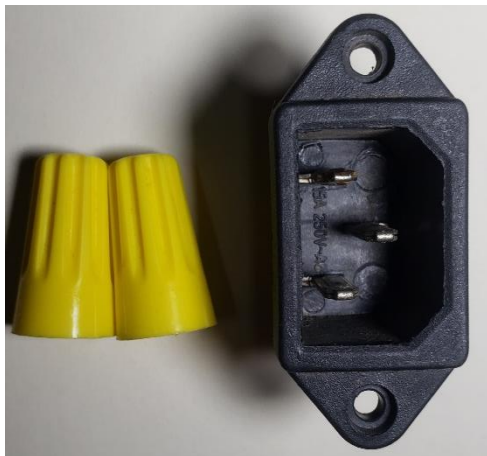


Figura 36: Conectores.

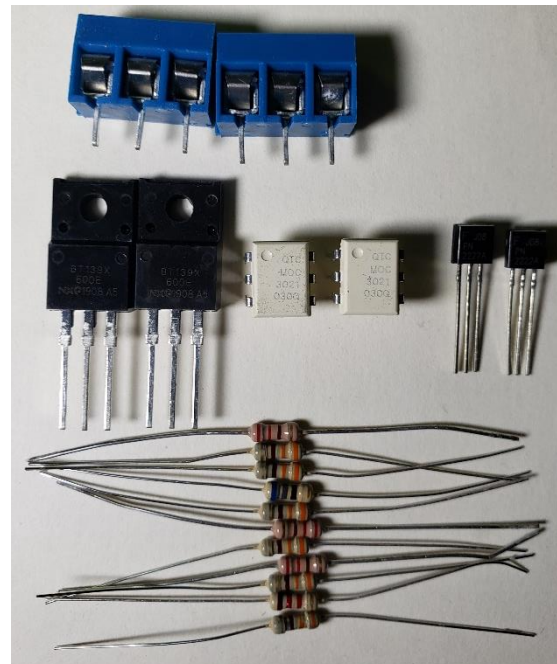


Figura 35: Componentes Electrónicos

En esta parte se procede a soldar todos los componentes electrónicos en la placa y verificar que no exista ningún tipo de cortocircuito de entre cada componente para evitar cualquier tipo de daño, en la figura 37 se puede visualizar todos los componentes soldados. Después de terminar toda la parte de soldar se realiza la instalación de la placa dentro de la caja protectora, para este prototipo se utilizó el mismo tipo de caja del anterior, pero con un tamaño reducido midiendo 9cm x 9cm a diferencia de la anterior que media 11cm x 11cm observar el interior de la caja en la figura 39 y su exterior en la figura 38.



Figura 37: PCB Terminada De ESP-01

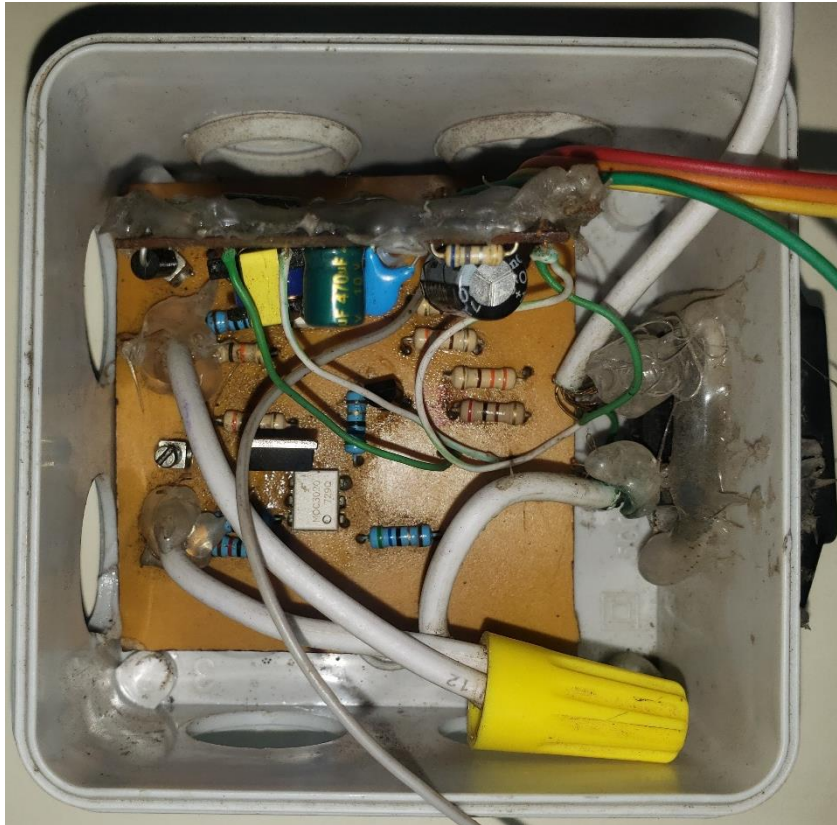


Figura 38: Parte Exterior De La Caja Del ESP-01



Figura 39: Parte Interior De La Caja Del ESP-01

En la última parte de este prototipo está presente la instalación final de la placa con todos los componentes electrónico en ella, la cual ha sido incrustada en la caja protectora como lo puede observar en la figura 40. Este prototipo necesita una fuente de poder de 5V/2A para que el ESP-01 funcione, para reducir costo y tamaño se utilizó la fuente de un cargador de teléfono.



*Figura 40: Prototipo Del ESP-01 Terminado*

## **NodeMCU V2**

En esta sección habla del prototipo NodeMCU V2, pero a diferencia de los otros dos prototipos que se fabricaron en físico esta versión solo constara de su diseño del circuito, la razón por la cual se ha decidido tomar esta decisión es debido que las diferencias entre diseños y conceptos es muy mínima respecto a los anteriores debido que un NodeMCU después de todo es un ESP-01 con más capacidades de control, el circuito electrónico creado para el NodeMCU V2 se puede visualizar en la figura 41. Al realizar una comparativa entre el circuito del ESP-01 de la figura 27. La diferencia notable es que la versión del NodeMCU no necesita tres resistencias que eran las utilizada para realizar la caída de voltaje de 5V a 3.3V ya que el ESP-01 trabaja con 3.3V pero en la versión del NodeMCU no la necesita porque trabaja con 5V que a su vez alimenta el circuito de control, aunque el diseño del circuito del NodeMCU muestre solo dos Triac al igual que la versión ES-01 esto no significa que su capacidad de control máxima sea de dos a como sucedía con el ESP-01, se pueden agregar en paralelo más circuito de acondicionamiento

para controlar más dispositivo pero en el diseño solo se han mostrado dos para realizar la explicación de los mínimos cambios que existen entre cada diseño.

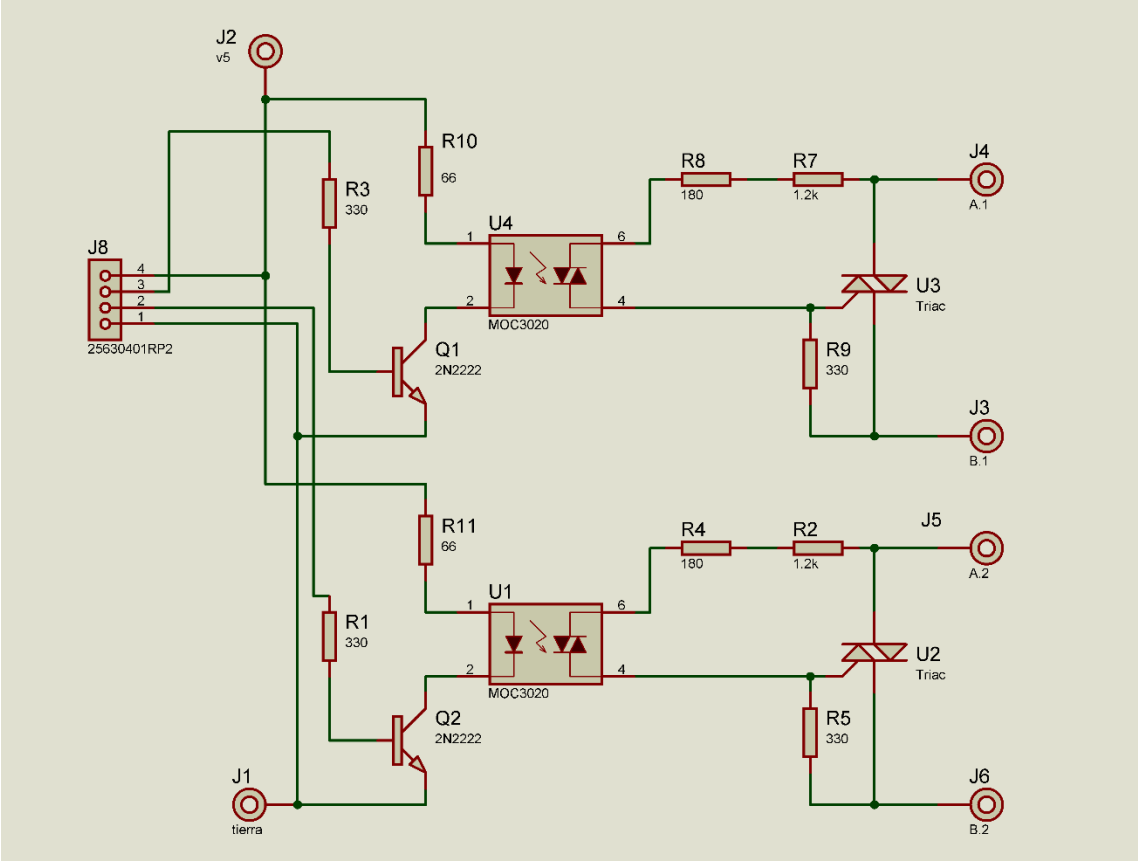


Figura 41: Circuito Para NodeMCU V2.

Por último, en la figura 42. esta el diseño del PCB que tendría este prototipo, es muy similar al prototipo del ESP-01 la única diferencia es que no tienen las tres resistencias.

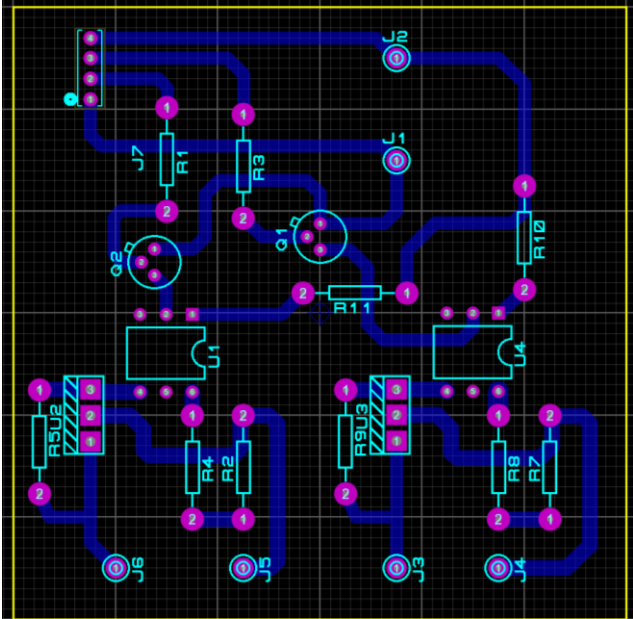


Figura 42: Diseño De La PCB Para NodeMCU V2

## Pruebas Finales

En esta sesión se presenta las pruebas realizadas para garantizar que la activación y desactivación de cada uno de los dispositivos sea correcta, además de validar que la comunicación de la Raspberry Pi con la plataforma en la nube este enviando y recibiendo los datos de forma adecuada, por motivo de una demostración grafica más fácil de visualizar para el documento las pruebas se realizaron con tres lámparas sin incluir una cerradura, pero el sistema funciona igual. En la figura 43 está el prototipo completo integrado con la Raspberry Pi.



*Figura 43: Pruebas Del Prototipo Completo*

En la figura 44 se puede apreciar a detalle las conexiones realizadas, en la parte de la izquierda esta la Raspberry Pi la cual se comunicará con la plataforma de la nube para saber que dispositivos deben estar activos o desactivados, debido que la Raspberry Pi necesita de un medio para controlar el prototipo lo hará por medio de un cable CAT5 que ira directamente al prototipo logrando de esta forma activar los Triac correspondiente para cada dispositivo.

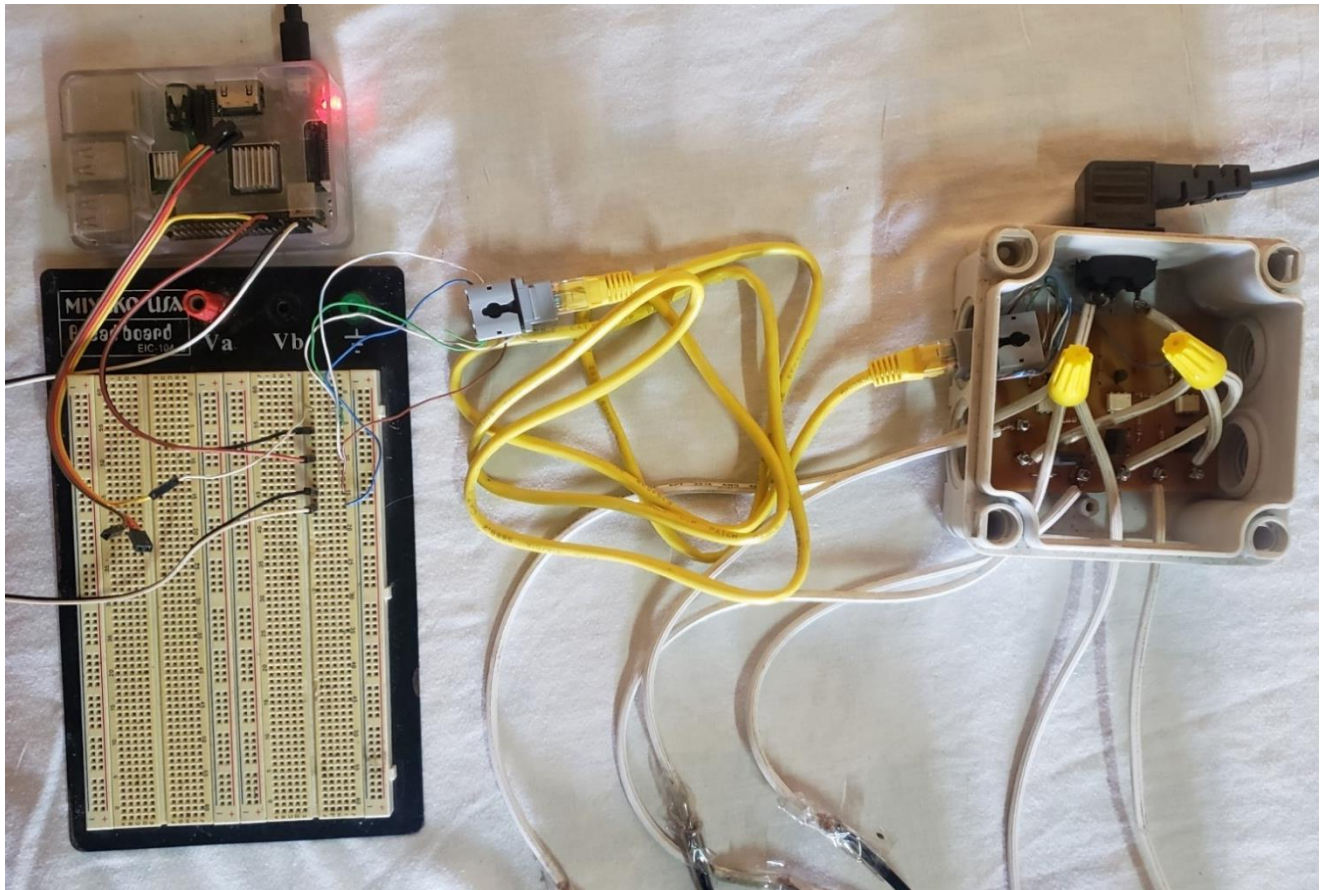


Figura 44: Sistema De Control

Las pruebas a realizadas se hicieron conforme al diagrama de bloque en la figura 1. Lo primero que se debe hacer es ejecutar el comando de activación por voz, para activar la primera lámpara el comando es “**Encender lámpara 1**” y para desactivar es “**Apagar lámpara 1**”. El primer paso activar lámpara 1 tal como en la figura 45. Este mensaje lo recibe la plataforma en la nube y el dispositivo cambia de imagen a un foco encendido en la figura 46. Posteriormente la tarjeta Raspberry Pi envía una solicitud POST al servidor para solicitar la información del dispositivo y de esa forma dar la orden de cambiar el estado del Triac con la finalidad de encender el foco 1 como se aprecia en la figura 47.

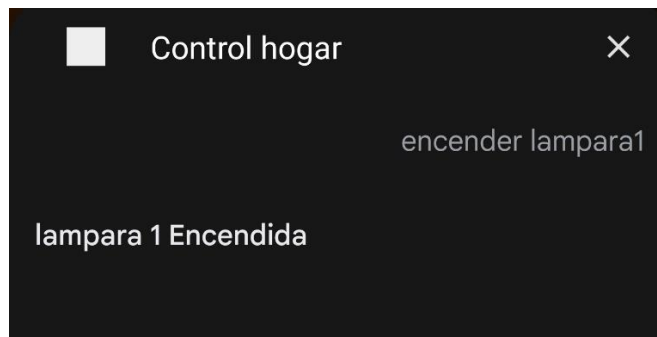


Figura 45: Activación Por Google Assistant

Dispositivos Actuales:

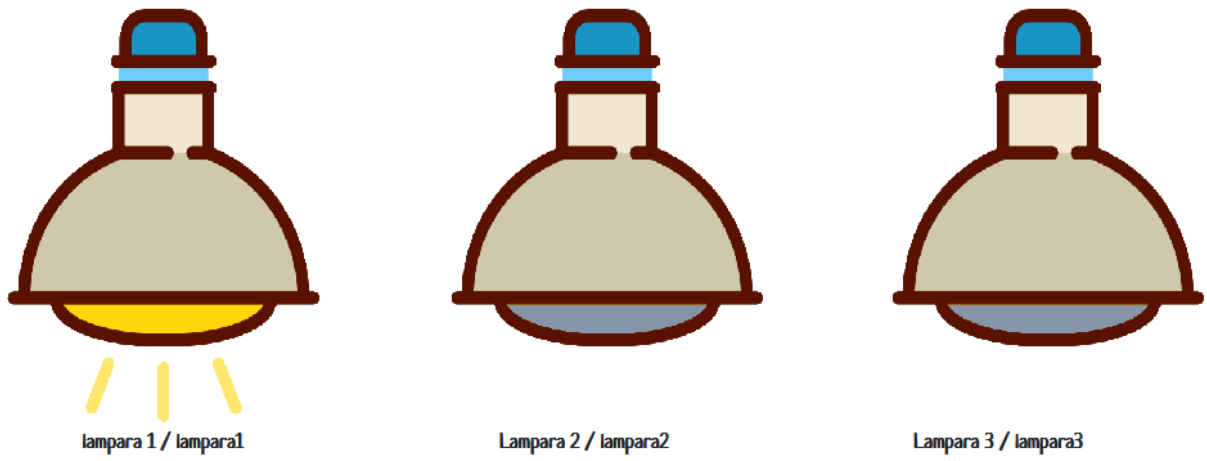


Figura 46: Plataforma En La Nube



Figura 47: Sistema De Control Activado

En esta parte se comprueba que la desactivación de los dispositivos funcione de forma correcta por lo tanto el primer paso es dar la orden a Google Assistant de apagar con el siguiente comando de voz **“Apagar lámpara 1”** como en la figura 48. Esta información es enviada al servidor y desactiva el foco cambiándolo por una lámpara apagada tal como se aprecia en la figura 49, posteriormente la Raspberry Pi envía una solicitud POST al servidor para actualizar el estado de todos los dispositivos y cuando lea la información de **“lámpara 1”** desactiva el GPIO correspondiente para apagarla por completo como se visualiza en la figura 50.

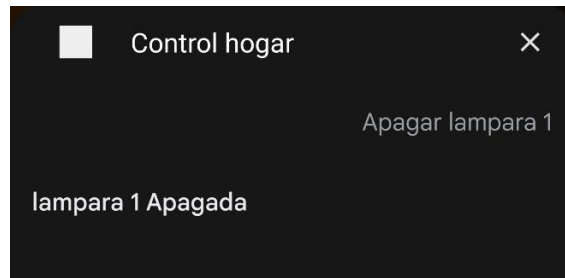


Figura 48: Apagar Lámpara 1

Dispositivos Actuales:

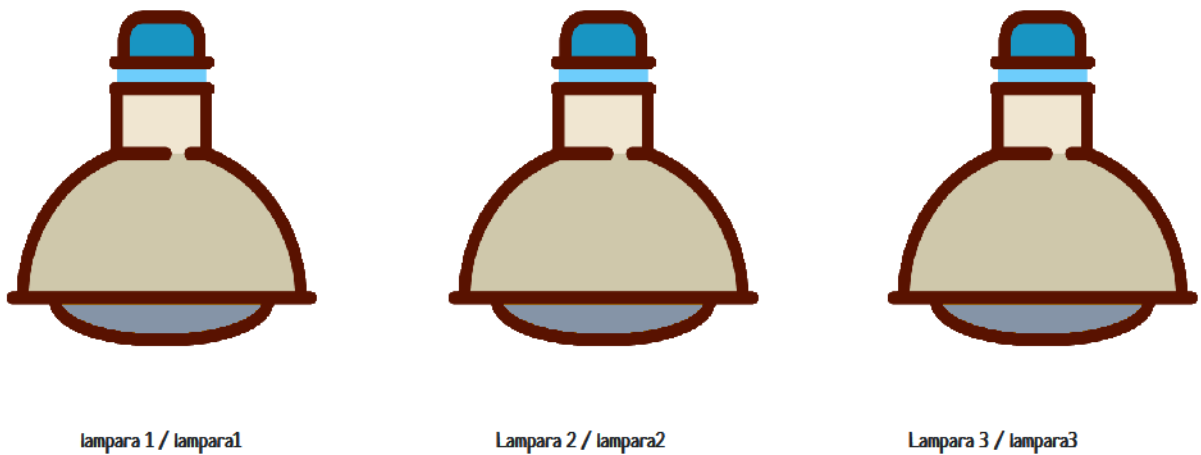


Figura 49: Dispositivos Apagado



Figura 50: Lámparas Apagadas



Para verificar que la activación de todos los Triac del sistema funcione correctamente de forma independiente se repite la prueba realizada anteriormente, pero activando una a una cada lámpara y validar que no existe ningún tipo de corto circuito que active alguna lámpara no deseada, también esta prueba sirve para saber si la comunicación de la Raspberry Pi con la plataforma en la nube se está dando de forma correcta activando los GPIO correctos en las siguientes imágenes se muestran la activación de lámpara 2 en la figura 51 y lámpara 3 en la figura 52. La activación de estas dos lámparas son los mismos pasos mostrado anteriormente con lámpara 1.



*Figura 51: Activación De Lámpara 2*



*Figura 52: Activación De Lámpara 3*

Se realiza una última prueba activando todas las lámparas al mismo tiempo para comprobar que no existe ningún tipo de corto circuito o algún tipo de interferencia que afecte al sistema, como se puede visualizar en figura 53.



*Figura 53: Activación De Todas Las Lámparas*

Tras finalizar todas las pruebas presentada anteriormente se puede concluir que el prototipo trabaja de forma correcta y no existe ningún tipo de fallo en el diseño o en la fabricación del mismo, debido que es capaz de activar o desactivar los dispositivo de acuerdo a la orden solicitada, también se ha comprobado que el aislamiento de la parte de potencia y la parte de control del circuito está funcionando de forma adecuada de lo contrario el alto voltaje de la parte de potencia habría dañado la Raspberry Pi.

## Cuarto capítulo: Comparativa Tecnológica y Económica

En este último capítulo se habla de la tecnología, la capacidad y el costo económico que tiene cada uno de los prototipos que fueron realizado en el proyecto con otras soluciones del mercado actual que tengan ciertas similitudes en cuanto característica y precios. Todo esto para brindarle al usuario un punto de vista más amplio respecto a cuál producto le es más conveniente adquirir para solucionar un problema en particular que tenga.

### Presupuesto Del Prototipo

Se hablará del costo económico llevado a cabo en la fabricación de cada uno de los prototipos, esto con la finalidad de explicar de dónde surge el costo total que tiene cada uno en el momento de realizar la comparativa con otro producto del mercado actual y analizar en detalle el gasto de los componentes electrónicos utilizados, todos los precios mostrados en las siguientes tablas fueron los precios apagar en el momento de la adquisición de cada pieza por lo tanto puede tender a cambiar.

En la primera tabla 1. Está presente el presupuesto del prototipo creado con la Raspberry Pi que tiene la capacidad de controlar tres dispositivos, pero eso no significa que sea el límite debido que la propia Raspberry Pi es capaz de controlar a un total de 28, el costo total en la fabricación ha sido de 5005 córdobas.

Tabla 1: Presupuesto Raspberry Pi

Nombre	Cantidad	Precio Unitario	Precio Total
Moc 3021	3	30	90
Triac BT139X	3	45	135
Resistencia	15	2	30
Transistor PN2222A	3	8	24
Baquelita	1	60	60
Caja	1	175	175
Ácido Muriático	1 L	64	64
Capuchones	10	3.5	35
Estaño	3 ft	9	27
Cable Eléctrico	2 m	15	30
Silicón	3	3	9

<b>Nombre</b>	<b>Cantidad</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
Jack Rj45	2	58	116
Conector hembra C13	1	25	25
Cable Cat5E	5 m	15	75
Conector Rj45	2	10	20
Terminal De Bloque	2	15	30
Impresión laser	1	10	10
Agua oxigenada	2	20	40
<b>Total Del Circuito</b>			<b>C\$995</b>
<b>Set Raspberry Pi 3 B+</b>	1	2810	2810
<b>Mano De Obra</b>			1200
<b>Total</b>			<b>C\$5005</b>

En la tabla 2. Esta el presupuesto del prototipo basado en el ESP-01 capaz de controlar 2 dispositivos al mismo tiempo que es lo máximo que permite la tarjeta, el costo de este modelo ha sido de 2154 córdobas.

*Tabla 2: Presupuesto ESP-01*

<b>Nombre</b>	<b>Cantidad</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
Moc 3021	2	30	60
Triac BT139X	2	45	47
Resistencia	13	2	26
Transistor PN2222A	2	8	16
Baquelita	1	60	60
Caja	1	80	80
Ácido Muriático	1 L	64	64
Capuchones	10	3.5	35
Estaño	3 ft	9	27
Cable Eléctrico	2 m	15	30
Silicón	3	3	9

Nombre	Cantidad	Precio Unitario	Precio Total
Conector hembra C13	1	25	25
Terminal De Bloque	2	15	30
Impresión laser	1	10	10
Fuente De Poder	1	150	150
<b>Total Del Circuito</b>			<b>C\$669</b>
<b>Esp-01</b>	1	285	285
<b>Mano De Obra</b>			1200
<b>Total</b>			<b>C\$2154</b>

En esta última tabla se creó un aproximado del gasto en la creación del prototipo con el modelo NodeMCU V2 con una capacidad de controlar tres dispositivos, pero a diferencia del ESP-01 esta versión tiene una capacidad máxima para controlar de ocho, el aproximado total sería de 2519 córdobas.

*Tabla 3: Presupuesto NodeMCU V2*

Nombre	Cantidad	Precio Unitario	Precio Total
Moc 3021	3	30	90
Triac BT139X	3	45	135
Resistencia	15	2	30
Transistor PN2222A	3	8	24
Baquelita	1	60	60
Caja	1	175	175
Ácido Muriático	1 L	64	64
Capuchones	10	3.5	35
Estaño	3 ft	9	27
Cable Eléctrico	2 m	15	30
Silicón	3	3	9
Agua oxigenada	2	20	40
Conector hembra C13	1	25	25
Fuente De Poder	1	150	150

Nombre	Cantidad	Precio Unitario	Precio Total
Terminal De Bloque	2	15	30
Impresión laser	1	10	10
<b>Total Del Circuito</b>			<b>C\$934</b>
<b>NodeMCU V2</b>	1	385	385
<b>Mano De Obra</b>			1200
<b>Total</b>			<b>C\$2519</b>

## Comparativa Del Prototipo

Actualmente en el mercado existen muchas soluciones similares de automatización de diferentes categorías, capacidad y precios. En este apartado se habla de unas cuantas soluciones comerciales actuales para analizar que ofrecen y realizar una comparativa tanto tecnológica como económica entre ellas con los distintos prototipos.

### Sonoff Basic

La primera solución a presentar es el **Sonoff Basic**, el cual tiene la capacidad de ser controlado tanto por voz haciendo uso de Google Assistant como de Alexa de Amazon y una App propia de la empresa, además posee conectividad a redes Wi-Fi y tiene una capacidad máxima de carga de 10A/2200W [16], el precio comercial de este producto es de unos 13 dólares en córdobas sería un equivalente aproximado de 481, en la figura 54. se puede observar el diseño y unas características extra.

Aunque esta solución tiene un precio mucho menor que cualquiera de los tres diferentes prototipos posee tres factores en contra, el Sonoff Basic solo es capaz de controlar un dispositivo por lo tanto implicaría realizar otra compra completa para controlar más de uno, su capacidad máxima de carga de es 10A/2200W lo cual es mucho menor capacidad que los Triac BT139X que posee cualquiera de los prototipos debido que los Triac tiene una capacidad de carga máxima de 16A/800V esto le permite al usuario controlar dispositivos que tengan una mayor demanda eléctrica y por ultimo una desventaja a nivel de diseño en la cual se observa en la figura 55, el Sonoff utiliza un Relay para controlar la carga por lo cual esto es un problema a futuro debido que los Relay se le suelen quebrar la pequeña lámina metálica de activación cosa que no suele suceder con los Triac porque estos no utilizan un método electromecánico para activarse por lo cual también le permite cambiar de estados mucho más rápido que un Relay.

El modelo del Relay que utiliza el Sonoff es el **W11-1A2STE** con una capacidad de 10A 250VAC/30VDC, es inferior por mucho al Triac BT139X, pero por esa misma razón tiene un precio mucho menor.



Figura 54: Características Del Sonoff Basic



Figura 55: Interior Del Sonoff

## 2-CH Triac HAT Para Raspberry Pi

Esta solución presenta un módulo especial para Raspberry Pi el cual permite controlar dos canales haciendo uso de Triac con una capacidad de carga máxima 20A/600V cada uno, posee un diseño de conexión GPIO estándar Raspberry Pi 40 pines, personalizada para placas de la serie Raspberry Pi, admite control por I2C y UART además de tener la capacidad de funcionar como un dimmer para controlar la intensidad de las luces [17], el diseño está presente en la figura 56.

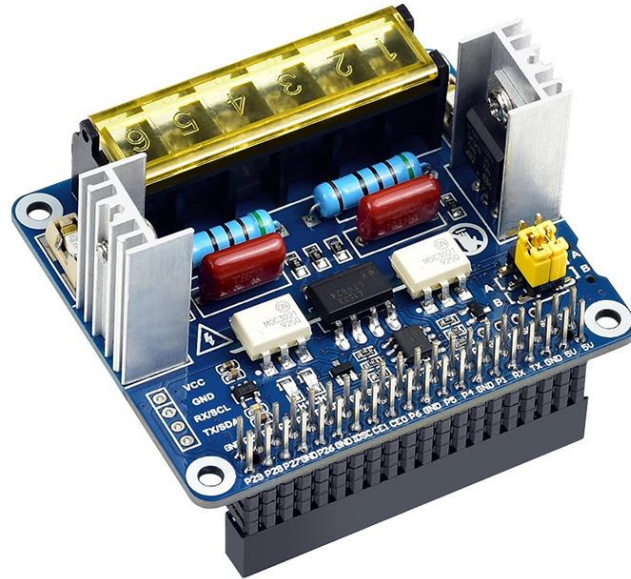


Figura 56: 2-CH Triac HAT Para Raspberry Pi

El costo aproximado que tiene esta placa es de 20 dólares el aproximado en córdobas sería de 740, esta solución tiene una gran similitud con cualquiera de los prototipos debido que utiliza Triac para realizar el control teniendo un Triac de 20A/600V que posee una mayor capacidad que el Triac BT139X que es de 16A/800V, pero una desventaja es su diseño exclusivo para Raspberry Pi que lo convierte un poco incómodo para trabajar con otras placas por deben sacar las líneas de comunicación y alimentación por medio de cables debido que la distribución de los pines son para Raspberry Pi por lo tanto no se pueden conectar directamente a otro modelo de placa además que por cada módulo que se desee agregar deben sacar nuevamente la línea de comunicación como la de alimentación provocando un caos de alambres, aunque cada prototipo del proyecto monográfico está diseñado para una placa especial el circuito base esta creado para ser utilizado con cualquier modelo de placa y a diferencia de esta solución los prototipo realizan el control por medio del GPIO.



## Enchufe inteligente NHP-D610

La tercera propuesta comercial está enfocada especialmente en el uso del hogar, es el enchufe inteligente **NHP-D610**, esta solución permite de manera sencilla convertir los tomacorriente en dispositivos inteligente haciendo uso de la aplicación de Nexxt, el usuario podrá encender y apagar sus electrodoméstico de forma remota desde cualquier lugar, además de admitir el control por voz mediante Google Assistant y Alexa de Amazon, es capaz de controlar dos tomacorrientes con una capacidad máxima de carga 15A/125V y dos puertos USB integrados de 5V/2A, además posee la característica de funcionar como protector de sobretensión con 200J [18], el precio aproximado de este producto es de 30 dólares, en la figura 57 se puede observar el dispositivo.



Figura 57: Enchufe inteligente NHP-D610

Esta solución a nivel económico tiene un costo menor y un diseño más comercial en comparativa de los prototipos, pero al tener un diseño tan comercial reduce las posibilidades de que dispositivos pueden ser utilizado debido que no se puede utilizar para controlar lámparas o bombillos comerciales ya que las terminales de conexiones que posee son tomas corriente así que los únicos dispositivos que se puede conectar debe poseer este tipo de conexión, en diferencia de los prototipos que tienen un diseño general que permite al usuario agregar cualquier tipo de conexión eléctrica que desee bien puede ser un toma corriente o incluso para conectar un bombillo. La capacidad máxima de carga que posee es las mismas características que los prototipos por lo tanto en esa característica son similares pero esta solución no brinda información sobre qué tipo de actuador utiliza para controlar por lo tanto puede generar desconfianza porque en un sistema de control es importante saber qué tipo de tecnología está utilizando.

## Resultados

En el transcurso de todo este protocolo se ha hablado de las características, diseño, costo económico y otras soluciones comerciales que poseen unos conceptos similares a los prototipos, en este apartado se presenta la comparativa definitiva de cada una de las propuestas comerciales con los prototipos que le corresponde ya sea por su rango de precio o incluso por sus características.

El **Sonoff Basic** es la solución comercial más básica de todas y la de menor precio, por lo tanto, el prototipo que le corresponde la comparativa es el modelo ESP-01 en la figura 58. se tiene una breve comparativa entre ambas con las principales características que son cuatros, el prototipo Es-01 cuesta el doble que el Sonoff pero tiene la doble capacidad de control además que utiliza un Triac el cual es mejor que un Relay porque no utiliza tecnología electromecánica por ende tiene una vida más larga, además que el Relay del Sonoff posee una capacidad de carga máxima inferior en comparativa con el Triac del Esp-01 por lo tanto si cuesta más pero tiene mayor capacidad de control, una vida útil más larga y puede controlar un dispositivo más, por esta razón es más rentable pagar un poco más por un producto que va durar más.

	 ESP-01	 SONOFF BASIC
<b>CAPACIDAD DE CONTROL</b>	<b>2</b>	<b>1</b>
<b>TIPO DE ACTUADOR</b>	<b>TRIAC</b>	<b>RELAY</b>
<b>CARGA MÁXIMA</b>	<b>16A/800V</b>	<b>10A/250V</b>
<b>PRECIO</b>	<b>C\$2154</b>	<b>C\$481</b>

Figura 58: Comparativa Entre ESP-01 Vs Sonoff Basic

El **2-CH Triac HAT** es un módulo dedicado exclusivamente a Raspberry Pi por lo tanto el prototipo a comparar será el modelo para Raspberry Pi debido que ambos utilizan la misma tarjeta de desarrollo y de esta forma al momento de la comparación de presupuesto sea lo más acorde posible, a diferencia de la anterior comparación se han agregado dos parámetros extras porque ambas soluciones se comunican de forma diferente, esto es muy importante de saberlo por si el usuario desea utilizar más de un módulo de cualquiera de las dos tarjetas, porque modulo comercial realiza el control por medio de la comunicación I2C/UART esto es ideal para reducir el uso de pines GPIO pero el problema ya se cuando el usuario quiera controlar otro modulo del mismo modelo, porque este módulo está fabricado para ser integrado encima de la Raspberry Pi por lo tanto para instalar otro módulo de este mismo modelo es necesario hacer uso de cable para sacar la líneas de alimentación y comunicación además de tener que realizar modificaciones física al módulo para poder instalarlo en una caja de forma independiente y todo esas modificación son gastos extras que se agregan de los cuales no están contemplado en la compra, la capacidad de control del módulo comercial es de dos dispositivos con una capacidad máxima de 20A/600V mientras el prototipo es capaz de controlar a tres pero a una menor capacidad máxima de 16A/800V, la elección en este apartado es del usuario por si desea controlar más dispositivo sin importar la capacidad máxima y en cuestión de precio no se ha incluido el precio de la tarjeta de desarrollo porque en ambos caso se necesita hacer la compra de la misma por ende solo se incluyó el gasto del circuito en sí. En la figura 59 está un resumen de la comparativa.

	 <b>RASPBERRY PI</b>	 <b>2-CH TRIAC HAT</b>
<b>CAPACIDAD DE CONTROL</b>	<b>3</b>	<b>2</b>
<b>TIPO DE ACTUADOR</b>	<b>TRIAC</b>	<b>TRIAC</b>
<b>CARGA MÁXIMA</b>	<b>16A/800V</b>	<b>20A/600V</b>
<b>I2C/UART</b>	<b>✗</b>	<b>✓</b>
<b>ON/OFF GPIO</b>	<b>✓</b>	<b>✗</b>
<b>PRECIO</b>	<b>C\$2195</b>	<b>C\$740</b>

Figura 59: Comparativa Entre Raspberry Pi vs 2-CH Triac Hat

En el mercador actual existen muy pocas cerraduras electrónicas que se puedan controlar por medio de la voz haciendo uso de los asistentes virtuales por lo tanto para esta comparativa se utilizó la cerradura **Yale RB-YRD540-WV-605** [19]. Es capaz de ser usada como las cerraduras tradicionales por medio de llaves, pero además posee características extras como el control por medio de un panel táctil donde el usuario ingresa la clave y como extra tiene la capacidad de ser controlada por medio de Google assistant. Para realizar esta comparativa se utilizó el prototipo con la Raspberry Pi debido que es el modelo más caro de crear con el valor de 7,105 córdobas ya incluyendo la cerradura, el precio en esta comparativa es muy importante porque la cerradura **Yale RB-YRD540-WV-605** tiene un valor de 10,150 córdobas es casi el doble del precio del prototipo, en la figura 60 se puede apreciar una comparativa entre el prototipo creado y la cerradura Yale entre sus principales diferencias.

	 <b>RASPBERRY PI</b>	 <b>GOOGLE NEST X YALE</b>
<b>CAPACIDAD DE CONTROL</b>	<b>3</b>	<b>1</b>
<b>TIPO DE ACTUADOR</b>	<b>ELECTROMECAÁNICO</b>	<b>ELECTROMECAÁNICO</b>
<b>CONTROL POR VOZ</b>	✓	✓
<b>PANEL TÁCTIL</b>	✗	✓
<b>PRECIO</b>	<b>C\$7105</b>	<b>C\$10150</b>

Figura 60: Comparativa Entre Raspberry Pi vs Yale

En la última propuesta corresponde al enchufe inteligente **NHP-D610**, esta solución se puede comparar con el ESP-01 o con el prototipo **NodeMCU V2**, pero por cuestión de costo de cada uno se realizó la comparativa con NodeMCU V2 ya que tiene un rango de costo muy similar, una característica diferente en esta comparativa es el tipo de conector que posee, la propuesta comercial tiene un tipo de conector Nema 5-15R que es el tomacorriente convencional de esta región esto tiene un inconveniente debido que no se puede utilizar para controlar bombilla o lámparas convencionales en cambio el prototipo posee una conexión general para que el usuario sea libre de utilizar el tipo de conector que quiera utilizar, en capacidad de carga máxima de ambas propuestas son muy similares aunque el NHP-D610 está un poco al límite además de no presentar información sobre qué tipo de actuador utiliza, otra diferencia notable el NodeMCU controla tres dispositivos en lugar de dos, el prototipo tiene un costo un poco mayor pero tiene la ventaja de controlar más dispositivos, utiliza Triac información que no brinda el fabricante y libertad de elección para las terminales eléctricas, observar en la figura 61.

	 <b>NODEMCU V2</b>	 <b>NHP-D610</b>
<b>CAPACIDAD DE CONTROL</b>	<b>3</b>	<b>2</b>
<b>TIPO DE ACTUADOR</b>	<b>TRIAC</b>	<b>-</b>
<b>CARGA MÁXIMA</b>	<b>16A/800V</b>	<b>15A/125V</b>
<b>TIPO DE CONECTOR</b>	<b>GENERAL</b>	<b>NEMA 5-15R</b>
<b>PRECIO</b>	<b>C\$2519</b>	<b>C\$1110</b>

Figura 61: Comparativa Entre NodeMCU V2 vs NHP-D610

## Conclusiones

Ha sido un largo camino para lograr obtener los resultados deseados y propuesto en el documento del prototipo, tanto en el apartado de la fabricación como el apartado narrativo del mismo. Pero finalmente se han cumplido cada uno de los objetivos que fueron propuesto en su inicio, se ha desarrollado la recopilación de información sobre las diferentes tecnologías de hardware y los distintos lenguajes de programación necesarios para el desarrollo del prototipo, explicando la importancia y el papel que juega cada una de las herramientas. Así mismo se efectuó toda la lógica de programación de todo el prototipo para lograr obtener que todo el sistema de control se active y desactive conforme las ordenes de ejecución del usuario en el momento que lo desee.

Una vez acabada la fase de programación del prototipo se da inicio a la creación y fabricación del diseño de control electrónico se ha realizado mediante simulación por medio del programa Proteus para garantizar el funcionamiento del circuito, además de ser utilizado para crear el diseño de cada una de las PCB para crear un tamaño exacto de las tarjetas y ubicar de forma correcta cada uno de los dispositivos electrónicos respecto al lugar que tendría dentro de su caja protectora.

Posteriormente se procedió a elaborar el presupuesto para las diferentes tarjetas de desarrollo para luego realizar una comparativa tanto tecnológicas como económicas entre diferentes tipos de tarjetas y las diferentes soluciones comerciales de la actualidad, mostrando cada una de las principales características de los productos como eran: capacidad de control, carga máxima, tipo de actuador y el precio del producto. Con la finalidad de que el usuario tenga un punto de vista más amplio respecto a cuál solución le conviene más a largo plazo.

## Recomendaciones

- En caso que el usuario desee controlar un dispositivo con mayor carga eléctrica se recomienda cambiar el modelo del Triac por uno de mayor capacidad y aumentar el ancho de la pista de la PCB recubriéndolas con una capa extra de estaño.
- Se recomienda cambiar el modelo del MOC por uno que tenga integrado la detección de cruce por cero, esto es para garantizar la desactivación de forma más exacta y precisa del Triac.
- En la fabricación de las PCB se recomienda verificar que cada uno de los pines mantenga una distancia considerable debido que el circuito es muy sensible a la estática al momento del montaje produciendo errores en la activación.
- Se recomienda integrar la opción en la plataforma en la nube de utilizar IFTT para configurar Google Assistant de una forma mucho más fácil, aunque la desventaja es limitar las frases de activación.
- Se recomienda integrar el uso de diferentes tipos de actuadores como pueden ser electroválvulas, duchas eléctricas, tomas corrientes convencionales, control de temperatura, portones eléctricos entre otros. De los cuales el usuario pueda utilizar desde un punto de vista de ocio y comodidad.
- Se sugiere utilizar una bocina inteligente que integre Google Assistant como la Nest Mini o la Nest Audio para tener una mayor comodidad y facilidad a la hora de activar los dispositivos.

## Bibliografía

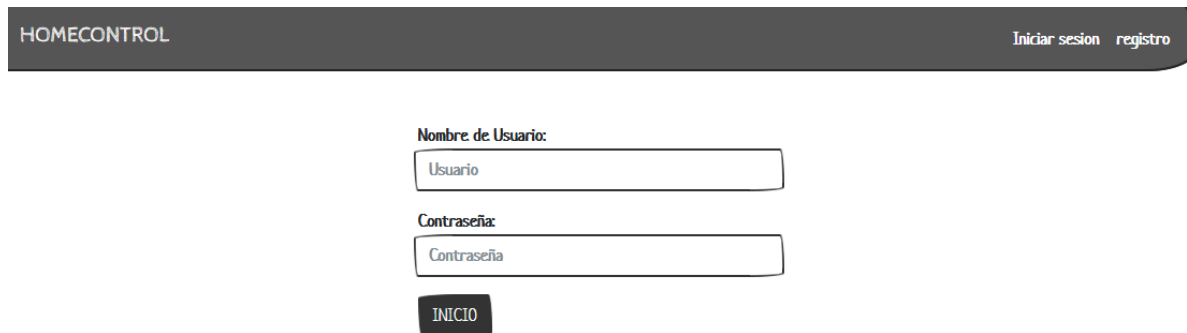
- [1] BM, «Banco Mundial,» 2009. [En línea]. Available: <https://datos.bancomundial.org/pais/nicaragua>. [Último acceso: 15 Septiembre 2020].
- [2] JICA, «ESTUDIO DIAGNÓSTICO DEL SECTOR DE LAS PERSONAS CON DISCAPACIDAD EN LA REPÚBLICA DE NICARAGUA,» 2014. [En línea]. Available: [https://www.jica.go.jp/nicaragua/espanol/office/others/c8h0vm000001q4bc-att/ESTUDIO\\_DISCAPACIDAD.pdf](https://www.jica.go.jp/nicaragua/espanol/office/others/c8h0vm000001q4bc-att/ESTUDIO_DISCAPACIDAD.pdf).
- [3] BM, «Banco Mundial,» 2016. [En línea]. Available: <https://datos.bancomundial.org/pais/nicaragua>.
- [4] OCNMT, Interviewee, [Entrevista]. 18 Octubre 2019.
- [5] Google, «Hey Google,» [En línea]. Available: <https://assistant.google.com/>.
- [6] Amazon, «AWS,» [En línea]. Available: <https://aws.amazon.com/es/about-aws/>.
- [7] Duckdns, «Duckdns,» [En línea]. Available: <https://www.duckdns.org/about.jsp>.
- [8] Electronic Frontier Foundation, «certbot,» [En línea]. Available: <https://certbot.eff.org/pages/about>.
- [9] Letsencrypt, «Letsencrypt,» [En línea]. Available: <https://letsencrypt.org/about/>.
- [10] G. B, «hostinger,» 29 Abril 2021. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-apache/>.
- [11] Pythonbasics, «Pythonbasics,» [En línea]. Available: <https://pythonbasics.org/what-is-flask-python/>.
- [12] Mysql, «Mysql,» [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
- [13] contributors, MDN, «mozilla,» 2 junio 2021. [En línea]. Available: [https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript).
- [14] Armo\_Electronics, «armoelectronics,» 6 Junio 2020. [En línea]. Available: <https://www.armoelectronics.com/calculadora-de-caida-de-voltaje/>.
- [15] Deltaeu, «Deltaeu,» [En línea]. Available: [https://shopdelta.eu/valor-de-awg\\_l6\\_aid938.html](https://shopdelta.eu/valor-de-awg_l6_aid938.html).
- [16] Sonoff, «Sonoff,» [En línea]. Available: <https://sonoff.tech/product/diy-smart-switch/basicr2/>.
- [17] Waveshare, «Waveshare,» [En línea]. Available: <https://www.waveshare.com/2-ch-triac-hat.htm>.
- [18] Connectivity, Nexxt Solutions, «Nexxt Solutions,» [En línea]. Available: <https://www.nexxtsolutions.com/conectividad/interna-productos/NHP-D610-es>.
- [19] Google, «Amazon,» [En línea]. Available: <https://www.amazon.com/Nest-Yale-Polished-Brass-Connect/dp/B07BH5QBWN>.



## Anexo

### 1. Configuración de la Plataforma

Primer paso ir a “[asistentehogarc1.duckdns.org](http://asistentehogarc1.duckdns.org)” mostrara:



HOMECONTROL Iniciar sesion registro

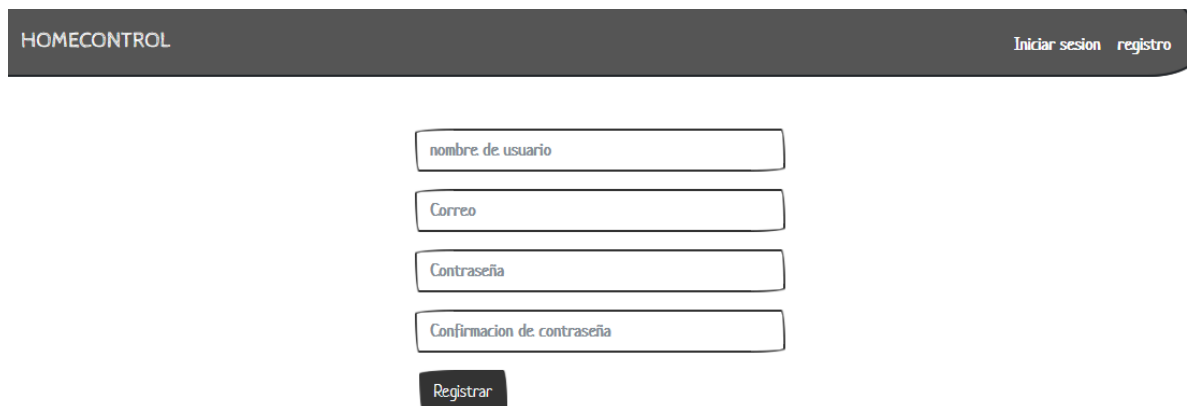
Nombre de Usuario:  
Usuario

Contraseña:  
Contraseña

INICIO

Figura 62: Login

Dar click al botón que dice “**registro**”:



HOMECONTROL Iniciar sesion registro

nombre de usuario

Correo

Contraseña

Confirmacion de contraseña

Registrar

Figura 63: Registro

Se debe llenar los campos y dar en el botón “**Registrar**”, el campo llamado correo puede ser llenado con cualquier correo electrónico, no es necesario que sea el mismo correo que se ha configurado con google assistant, una vez terminado de llenar los campos y dar click en “**Registrar**” mostrará lo siguiente:

Figura 64: Verificar

Debe introducir el código que se ha enviado al correo con el cual creo la cuenta para terminar el registro, una vez rellanado el campo llamado código damos click en “**Registrar**”:

Bienvenido Persona  
Dispositivos Actuales:

Figura 65: Home

Mostrará el panel de control vacío debido que no se tiene dispositivos agregados, para agregar los dos dispositivos que se configuraron en Google assistant “**lámpara 1**” y “**lámpara 2**” dar click al botón llamado “**configuración**”:

**Notificaciones**

Hora  Minutos

**Agregar Dispositivo**

Nombre  Identificador

Audio  Imagen

Estado

**Eliminar Dispositivo**

Dispositivos

Figura 66: configuración

En la parte izquierda se encontrar la parte de notificaciones en la cual están los siguientes campos:

### Notificaciones

:

Figura 67: Notificación

**Correo:** se puede cambiar el correo al cual se desee recibir los mensajes de alerta, deforma predeterminada los mensajes serán enviados al correo con el cual se ha creado la cuenta, cambiar el correo de notificaciones no significa cambiar el correo con el cual fue creado la cuenta en la plataforma.

**Hora y minutos:** en estos campos se pondrá la hora y los minutos que quiera recibir el mensaje de alerta, el formato de la hora es de 24 hora por lo que si quiera poner una alerta las 10:00 P.M. seria 22:00.

En la parte derecha esta “**agregar dispositivo**” y “**Eliminar Dispositivos**”:

### Agregar Dispositivo

### Eliminar Dispositivo

Figura 68: Agregar Dispositivo

En la parte de “**agregar dispositivos**” se encuentra los siguientes campos:

**Nombre:** en el cual se debe escribir el nombre que será utilizado por el asistente de google para confirmar que acción se ha efectuado correctamente.

**Identificador:** en este parámetro se pondrá el identificador del dispositivo, el cual debe ser el mismo identificador que se ha registrado previamente en google assistant ejemplo para “**lámpara 1**” el identificador que ha puesto fue “**lampara1**” por lo cual ese será el identificador que pondremos aquí.

**Audio:** en este campo los valores están predeterminado para elegir, se puede encontrar diferentes audios para lámparas y cerraduras con su número designado, este audio se reproducirá en el panel de control de la plataforma por lo cual hay que tener en cuenta de no repetir la selección al momento de agregar nuevos dispositivos para no causar confusión.

**Imagen:** en este campo están valores predeterminado para elegir al igual que en audio, aquí se podrá seleccionar una imagen para el dispositivo que se quiere agregar, en las seleccione se encuentra “**lampon**” que es una lámpara encendida, “**lampoff**” que se refiere a una lámpara pagada, “**cerron**” es un candado cerrado y “**cerroff**” es un candado abierto.

**Estado:** aquí estará dos opciones las cuales son “**Encendido**” y “**Apagado**” debe poner el estado que se encuentra actualmente el dispositivo, si la cerradura está abierta debe poner el estado de encendido.

En la parte “**Eliminar Dispositivos**” se encontrará los dispositivos actuales, aparecerán automáticamente si sea eliminar uno solo debe seleccionarlo y dar click en el botón “**Aceptar**”

En el siguiente ejemplo se agregarán los dispositivos “**lámpara 1**” y “**lámpara 2**” que se han configurado previamente en google assistant:

**Agregar Dispositivo**

lampara 1	lampara1
lamp1off	lampoff
Apagado	

*Figura 69: Agregar Dispositivo Ejemplo 1*

Una vez llenado los campos dar click en “**aceptar**” y mostrara el panel de control:

Bienvenido Persona

Dispositivos Actuales:



lampara 1 / lampara1

Figura 70: Home Dispositivo 1

Se puede observar que el dispositivo ha sido agregado correctamente y en la parte inferior esta “**nombre / identificador**” ahora volver a configuración y agregar al segundo dispositivo:

### Agregar Dispositivo

lampara 2	lampara2
lamp2on	lampon
Encendido	

Figura 71: Agregar Dispositivo Ejemplo 2

Los identificadores nunca se pueden repetir una vez agregado los campos dar en “**Aceptar**”:

Bienvenido Persona

Dispositivos Actuales:

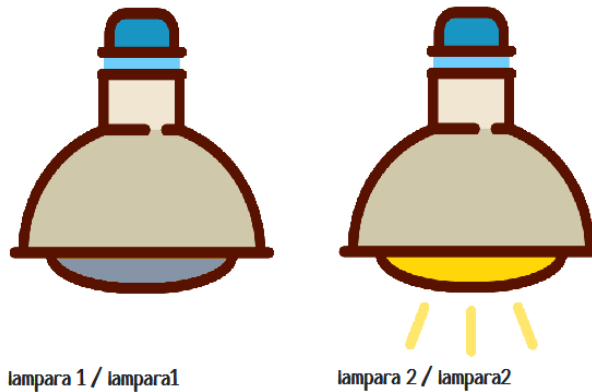


Figura 72: Home Dos dispositivos

Observar que ambos dispositivos se agregaron correctamente, volver a configuración y dar a la parte de **“Eliminar dispositivos”**:

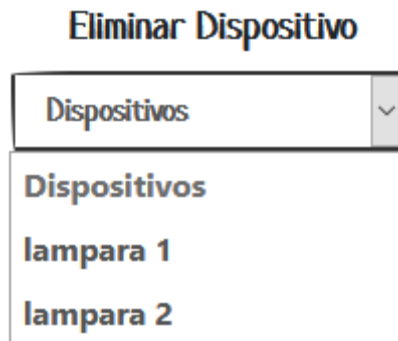


Figura 73: Eliminar Dispositivo

Se puede observar que los nombres de los dos dispositivos que se agregaron aparecen, seleccioné uno y dar click en **“Aceptar”** se eliminara automáticamente.

## 2. Configuración de Raspberry pi 3 y 3 B+

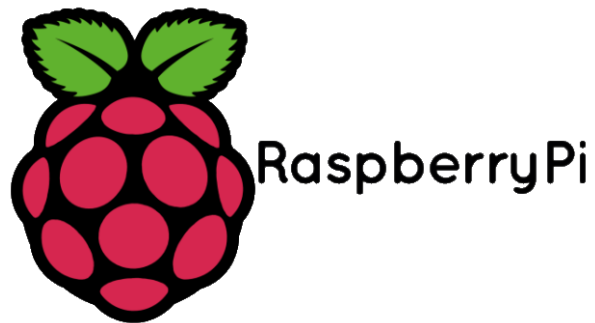


Figura 74: Logo Raspberry Pi

Descargue el archivo “*raspberry.py*” en el cual encontrara lo siguientes puntos importantes:

Configuración de pines:

```
13 #configuracion de pines
14 GPIO.setup(32,GPIO.OUT)
15 #GPIO.setup(pin,GPIO.OUT)
16 #GPIO.setup(pin,GPIO.OUT)
17 #GPIO.setup(pin,GPIO.OUT)
```

Figura 75: Pin del GPIO

En esta parte se pondrán los números de los pines que desea controlar, puede borrar el “#” de las líneas 15, 16 y 17 para agregar más pines y cambiando “*pin*” por el número correspondiente. Puede copiar “*GPIO.setup(pin, GPIO.OUT)*” para agregar indeterminados pines que desee controlar. Para conocer el número de cada pin del GPIO puede consultar la siguiente tabla:

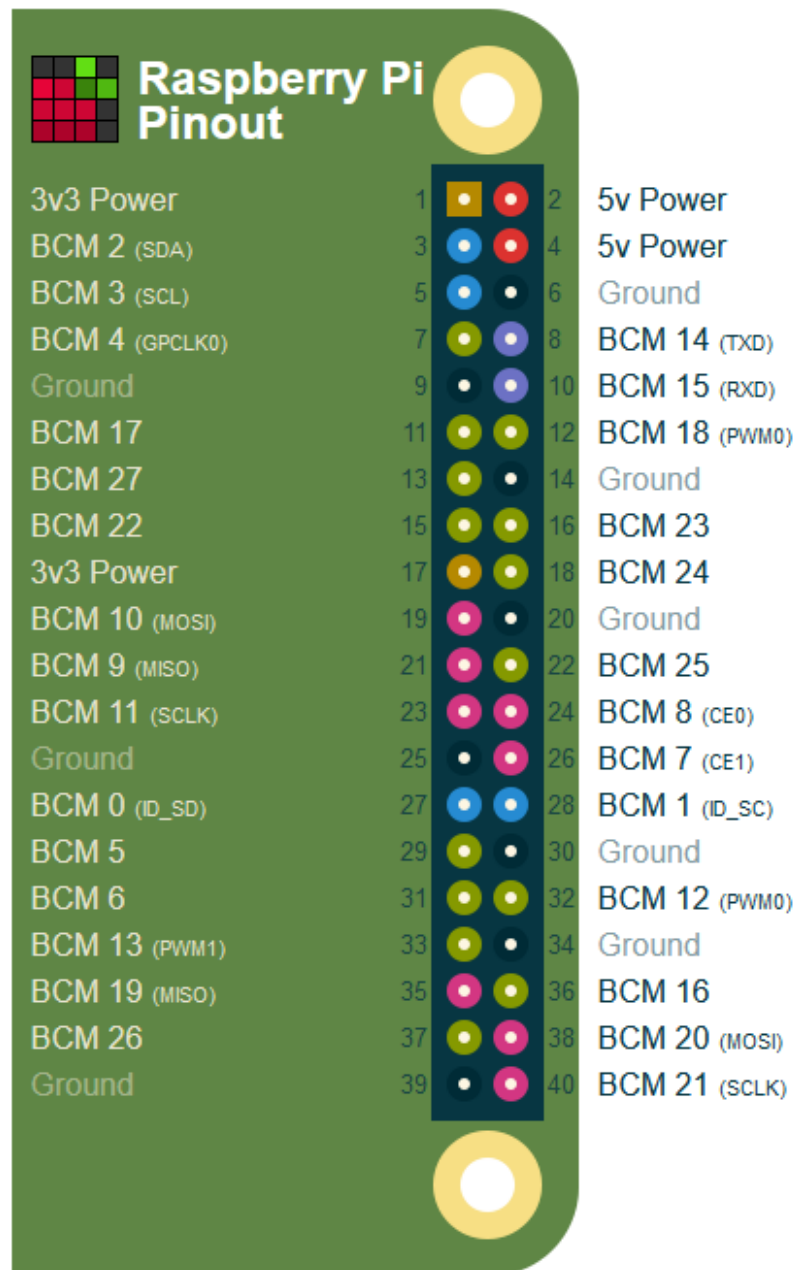


Figura 76: GPIO de la Raspberry Pi

Puede encontrar esta imagen en: <https://pinout.xyz/#> . si desea conocer más sobre el GPIO viste: <https://www.raspberrypi.org/documentation/usage/gpio/>.



La siguiente parte del código a modificar será:

```
24 #agregar credenciales de la cuenta creada en la plataforma relleno las casillas
25 #"usuario" y "password" conservando las comillas
26 auth2= HTTPBasicAuth('usuario','password')
```

Figura 77: Autenticación

“**usuario**” y “**password**” tendrá que sustituirlos con el correo y contraseña de la cuenta que ha creado en la plataforma “**Homecontrol**”.

La última parte del código a modificar es:

```
39
40 #para agregar mas dispositivos puede copiar este "if" y pegar la copia debajo cambiandole
41 #el identificador y agregando un nuevo pin de control en el GPIO
42 if str(i[0]) == "identificador" and str(i[1]) == "1":
43     GPIO.output(32,True)
44
45 elif str(i[0]) == "identificador" and str(i[1]) == "0":
46     GPIO.output(32,False)
47
```

Figura 78: Control de GPIO

Tiene que sustituir “**Identificador**” en la línea 42 y 45, con el ID del dispositivo que ha agregado en la plataforma previamente, además de modificar el número del pin si así lo desea, esta línea de código solo sirve para controlar 1 dispositivo para controlar más puede copiar las líneas de código de la 42 hasta la 46 y pegarla debajo de esta, conservando las tabulaciones y cambiando el “**Identificador**” con el ID del dispositivo a controlar y también el número de pin(los números de los pines no pueden ser los mismo para distintos dispositivos).

Para concluir se dará el ejemplo de configurar el control de “**lámpara 1**” y “**lámpara 2**”:

Primero configurar lo pines:

```
13 #configuracion de pines
14 GPIO.setup(40,GPIO.OUT)
15 GPIO.setup(36,GPIO.OUT)
16 #GPIO.setup(pin,GPIO.OUT)
17 #GPIO.setup(pin,GPIO.OUT)
```

Figura 79: Pines del GPIO

Agregar una línea de configuración de GPIO y se le pondrá el número pin que deseamos utilizar, es importante mirar el número del pin en el diagrama para realizar una buena conexión.

Segundo agregar la cuenta:

```
24 #agregar credenciales de la cuenta creada en la plataforma relleno las casillas
25 #"usuario" y "password" conservando las comillas
26 auth2= HTTPBasicAuth('persona','12345')
```

Figura 80: Ejemplo de Autenticación

Tercero agregar un dispositivo más para controlar copiando las líneas de código desde la 42 hasta la 46 y pegándolas debajo:

```
42         if str(i[0]) == "lampara1" and str(i[1]) == "1":
43             GPIO.output(40,True)
44
45         elif str(i[0]) == "lampara1" and str(i[1]) == "0":
46             GPIO.output(40,False)
47
48
49         if str(i[0]) == "lampara2" and str(i[1]) == "1":
50             GPIO.output(36,True)
51
52         elif str(i[0]) == "lampara2" and str(i[1]) == "0":
53             GPIO.output(36,False)
54
```

Figura 81: Control de Dos Salidas

Guardar el archivo, abrir la terminal donde este guardado el archivo:

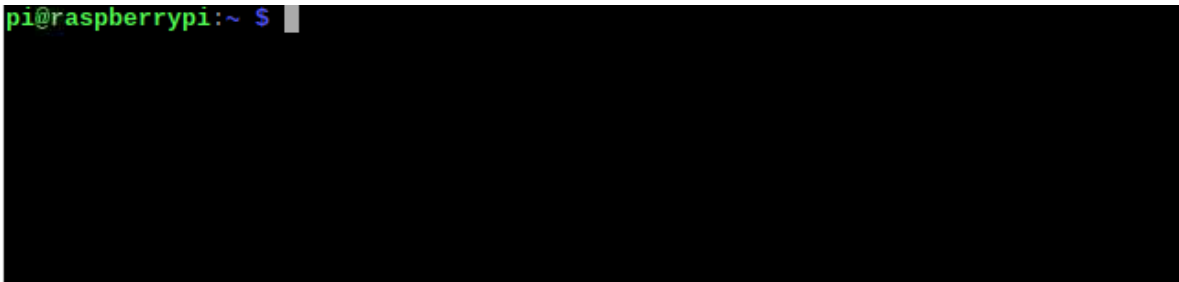


Figura 82: Consola de Comandos

Para entrar en un directorio específico escribir en la terminal "cd" más el directorio que de seamos entrar, ejemplo: **cd Download**.

y ejecutar el programa con “**Python3 nombredelarchivo.py**”:

```
pi@raspberrypi:~ $ python3 raspberry_1.0.py
```

*Figura 83: Ejecutar Programa*

Y dar enter en cual debe aparecer el siguiente mensaje:

```
pi@raspberrypi:~ $ python3 raspberry_1.0.py  
Run  
█
```

*Figura 84: Programa Ejecutado*

“**Run**” significa que se está ejecutando sin problemas.

### 3. Configuración de ESP-01 y Nodemcu v2

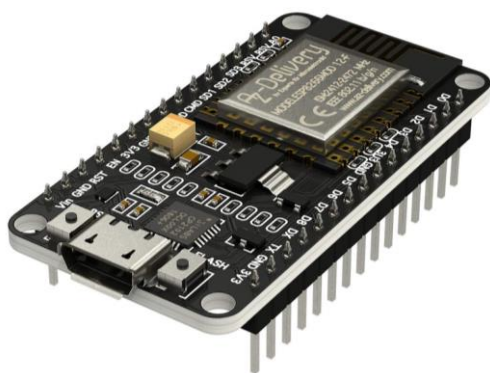


Figura 85: Nodemcu v2

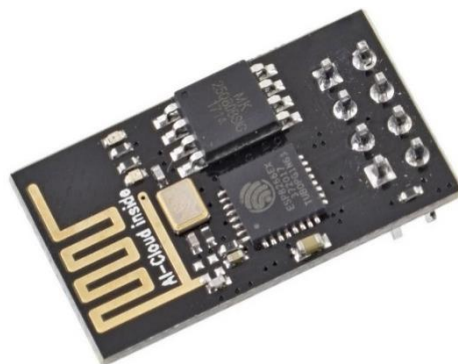


Figura 86: ESP-01


Descargue el archivo “**esp01.ino**” en el cual encontrara lo siguientes puntos importantes:

La primera variable a cambiar del código original se encuentra en la línea 19 en esta línea tendrá que poner el “**SHA-1**” del certificado web de la página en su formato hexadecimal agregando el “**0x**” a cada dígito del certificado tal como se muestra en la siguiente imagen:

```
18  
19 const uint8_t fingerprint[20] = {0x8E, 0x27, 0xE8, 0xA2, 0x82, 0xC2, 0xB9, 0x32, 0xEF, 0x0D, 0xAE, 0x37,  
20
```

Figura 87: SHA-1

Este código debe ser actualizado cada 3 meses debido que el certificado de la página web suele renovarse cada periodo de tiempo por tema de seguridad, para obtener el “**SHA-1**” de la página actualizado tiene que seguir los siguientes pasos:

Lo primero debe acceder al sitio web de la plataforma “<https://asistentehogarc1.duckdns.org>” en la parte superior de la barra de navegación dar click en el símbolo del candado. 

Este desplegara las siguientes opciones:

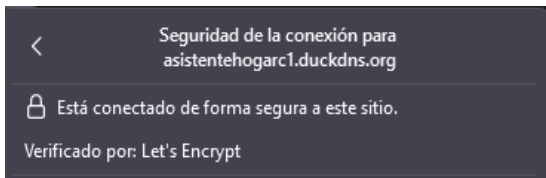


Figura 88: Seguridad SSL

Debe dar click en “**Más información**”, posteriormente dar click a la opción que dice “**ver certificado**”

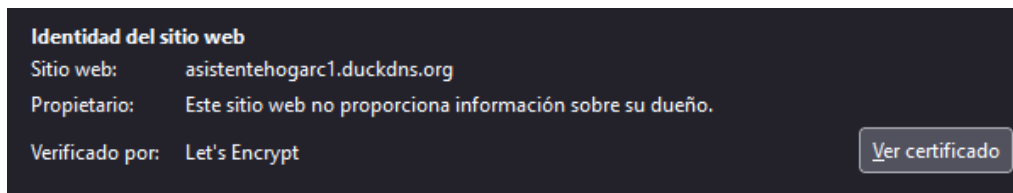


Figura 89: SSL Info

Al entrar en esta opción mostrara toda la información sobre el certificado de seguridad y debe ir donde dice "**Huellas digitales**" en este apartado está el "**SHA-1**" de la página el cual copiaremos y pegaremos en el código.

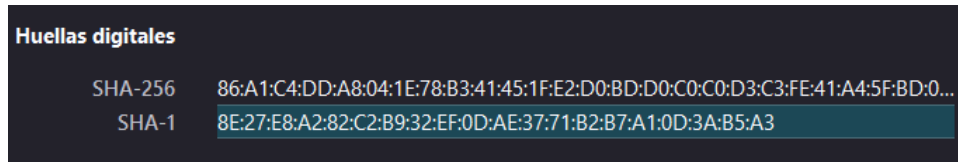


Figura 90: Certificado SSL

Configuración de pines para el "**ESP-01**":

```
28 pinMode(0, OUTPUT);
29 pinMode(2, OUTPUT);
30
```

Figura 91: Pines de Control

Para el "**ESP-01**" este apartado no debe ser modificado debido que esta tarjeta solo posee 2 pines en cambio para el "**Nodemcu V2**" debe seguir la siguiente configuración:

```
28 pinMode(D0, OUTPUT);
29 pinMode(D1, OUTPUT);
30
```

Figura 92: Configuración de Pines

El nombre de los pines debe agregar "**D + Número de pin**" de lo contrario mostrara un error y para agregar más pines deben copiar la línea 28 y pegarla debajo de la línea 29 modificando el número del nuevo pin a controlar.

En la línea 44 debe agregar al nombre de la red wifi y la contraseña para que el dispositivo sea capaz de conectarse a internet:

```
42
43 Wifi.mode(WIFI_STA);
44 WifiMulti.addAP("Nombre", "Password");
45
```

Figura 93: Contraseña del WIFI

En la línea 63 debe sustituir los campos llamado "**Usuario**" y "**Password**" por la cuenta creada en la plataforma de esta manera podrá iniciar sesión para recibir toda la información de los dispositivos.

```
61   if (https.begin(*client, "https://asistentehogarcl.duckdns.org/data")) {
62     //la autorizacion poner el usuario y password
63     https.setAuthorization("Usuario", "Password");
64
```

Figura 94: Autenticación al Servidor

La última parte a modificar corresponde a la parte del control de los dispositivos, en la cual debe sustituir "**Identificador**" por el nombre del dispositivo el cual tiene que ser escrito exactamente como fue configurado en la plataforma.

```
102
103     if(idef=="Identificador" && val=="1"){
104         digitalWrite(0,HIGH);
105     }else if (idef=="Identificador" && val=="0"){
106         digitalWrite(0,LOW);
107     }
108
109
110
111
112
```

Figura 95: Control de Pines

Un ejemplo para controlar dos dispositivos al mismo tiempo lo puede encontrar en la siguiente imagen en la cual lo único que se realizó fue copiar la línea 103 hasta la 111 y pegarla en la parte de abajo, modificando el nombre de los dispositivos y el número del pin que controlara cada uno, este mismo proceso debe repetirse para agregar un tercer o un cuarto dispositivo siempre recordando cambiar el número de pin.

```
102
103     if(idef=="lampara1" && val=="1"){
104         digitalWrite(0,HIGH);
105     }else if (idef=="lampara1" && val=="0"){
106         digitalWrite(0,LOW);
107     }
108
109
110
111
112
113
114     if(idef=="cerradura1" && val=="1"){
115         digitalWrite(2,LOW);
116     }else if (idef=="cerradura1" && val=="0"){
117         digitalWrite(2,HIGH);
118     }
119
120
121
122
123
```

Figura 96: Control de Dos Pines.