ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO INALÁMBRICO CONTROLADO POR COMANDOS DE VOZ PARA EL HOGAR

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

Kevin David Clavijo Guañuna
kevin.clavijo@epn.edu.ec

Kevin Fabricio Tipanguano Chicaiza
kevin.tipanguano@epn.edu.ec

DIRECTOR: ING. LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.
leandro.pazmino@epn.edu.ec

CODIRECTOR: ING. MÓNICA VINUEZA RHOR, MSC.
monica.vinueza@epn.edu.ec

Quito, diciembre de 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Sres. Clavijo Guañuna Kevin David y Tipanguano Chicaiza Kevin Fabricio como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:

Ing. Leandro Pazmiño Ortiz, MSc.	**Ing. Mónica Vinueza Rhor, MSc.**
DIRECTOR DEL PROYECTO	CODIRECTORA DEL PROYECTO

DECLARACIÓN

Nosotros Kevin David Clavijo Guañuna con CI: 1722867304 y Kevin Fabricio Tipanguano Chicaiza con CI: 1725545105, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

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

Kevin Clavijo

Kevin David Clavijo Guañuna

Kevin Fabricio Tipanguano Chicaiza

DEDICATORIA

A mis padres Julio y Carmita por haberme forjado como la persona que soy en la actualidad, pues sin su apoyo incondicional, paciencia y sacrificio no habría logrado conseguir muchos de mis logros incluido este.

A mis hermanos Andy y Lenin quienes me motivaron constantemente para alcanzar mis metas.

KEVIN DAVID CLAVIJO GUAÑUNA

AGRADECIMIENTO

Agradezco de todo corazón a mis padres y hermanos, quienes han sido parte fundamental en la trayectoria de toda mi vida y los principales promotores de mis sueños, por su confianza, por creer en mí, por sus consejos y valores y por anhelar siempre lo mejor para mí.

Gracias a Dios, por brindarme la oportunidad de estudiar en la Escuela Politécnica Nacional, también porque cada día bendice mi vida con la hermosa oportunidad de estar y disfrutar a lado de las personas que más aprecio.

A mis compañeros y amigos de prepo de la “ESFOT 7 - 8” y todas las personas que estuvieron desde el comienzo de mi carrera. Gracias por brindarme su amistad, su respeto y compañía, sin ustedes nada hubiera sido igual.

A mis docentes, por haberme transmitido sus conocimientos, valores y formar mi vida profesional.

De manera especial agradezco a mis primos y amigos Jhony, Christopher, Michael, Christopher, Josselyn, Naty, Bryan, Xavier, a todos, gracias por los buenos momentos que hemos compartido, por sus palabras de aliento y apoyo en los momentos difíciles, por brindarme sus consejos, compañía y ánimo durante mi trayectoria en la universidad.

KEVIN DAVID CLAVIJO GUAÑUNA

DEDICATORIA

A mis padres por ser quienes me supieron guiar durante este largo camino, por enseñarme que con sacrificio y responsabilidad se puede conseguir lo que uno se proponga.

A mis hermanos por ayudarme en los momentos más difíciles.

A todas las personas que estuvieron presentes en los momentos buenos y malos, en especial a esa persona que desde el cielo me impulso a llegar a los más alto de mi carrea profesional.

KEVIN FABRICIO TIPANGUANO CHICAIZA

AGRADECIMIENTO

Agradezco a mis padres Gladys Chicaiza y Jaime Tipanguano por ayudarme a culminar mis estudios, por inculcarme la responsabilidad y el esfuerzo que me permitieron culminar mis estudios.

A mis hermanos que siempre creyeron en mí y me brindaron su ayuda incondicional.

A la Escuela Politécnica Nacional, a mis profesores y compañeros por haberme impartido todos sus conocimientos para mi formación profesional.

KEVIN FABRICIO TIPANGUANO CHICAIZA

ÍNDICE DE CONTENIDOS

1 Introducción 1
1.1 Objetivo general 1
1.2 Objetivos específicos 1
1.3 Fundamentos 2
Amazon Alexa 2
Blynk 3
Hardware 3
Arduino UNO 4
WeMos D1 mini 4
Arduino Nano 5
Arduino Leonardo 5
Arduino Mega 2560 6
2 Metodología 7
2.1 Descripción de la metodología usada 7
3 Resultados y Discusión 8
3.1 Identificación de los requerimientos del prototipo 8
3.2 Selección del *hardware* adecuado 9
Adquisición de materiales 11
Resistencias 11
Diodos 12
Transistor 12
LEDS 13
Relés 13
Baquelita 14
Borneras 14
Servomotor 15
Sirena 15

Ventilador 16
Luminarias 16
3.3 Interfaz humano-máquina 17
Desarrollo del programa de control 24
Comandos de voz en *Voiceflow* 24
Envió de datos al servidor *Blynk* en *WeMos* D1 mini 25
Lectura del código Arduino 30
3.4 Implementación del sistema 31
Ubicación de elementos 33
Diseño de los circuitos electrónicos 35
Circuito de control 35
Módulo relé 37
Fabricación de los circuitos impresos 38
3.5 Prueba y análisis de resultados 41
Prueba del circuito electrónico 41
Pruebas de comunicación 43
Prueba de funcionamiento del sistema 44
3.6 Manual de Uso y Mantenimiento 48
4 Conclusiones y Recomendaciones 50
4.1 Conclusiones 50
4.2 Recomendaciones 51
5 Referencias Bibliográficas 53
ANEXOS 56
Anexo 1: Certificado de Funcionamiento i
Anexo 2: Planos y Esquemas iii
ANEXO 3: Diagramas de flujo i
ANEXO 4: Comandos de voz i

ÍNDICE DE FIGURAS

Figura 1.1 Activación por voz de una habilidad con el servicio Alexa 2
Figura 1.2 Funcionamiento de *Blynk* IoT 3
Figura 1.3 Microcontrolador Arduino UNO 4
Figura 1.4 *WeMos* D1 mini 5
Figura 1.5 Microcontrolador Arduino Nano 5
Figura 1.6 Microcontrolador Arduino Leonardo 6
Figura 1.7 Microcontrolador Arduino Mega 7
Figura 3.1 Amazon *Echo dot* tercera generación 9
Figura 3.2 Resistencias de 220, 330 y 1k (Ω). 12
Figura 3.3 Diodo 1N4001 12
Figura 3.4 Transistores 2N2222A. 13
Figura 3.5 Diodo *LED* 13
Figura 3.6 Relé 5 (V_{DC}) 14
Figura 3.7 Baquelita. 14
Figura 3.8 Borneras. 15
Figura 3.9 Servomotor SG90. 15
Figura 3.10 Sirena 16
Figura 3.11 Ventilador. 16
Figura 3.12 Foco Led y boquilla. 17
Figura 3.13 Interfaz *web Blynk*. 18
Figura 3.14 Crear una plantilla en *Blynk*. 18
Figura 3.15 Configuración de pin virtual. 19
Figura 3.16 Configuración de botones. 19
Figura 3.17 Panel *web* completo. 20
Figura 3.18 Configuración de la aplicación *Blynk*. 21
Figura 3.19 Espacio de trabajo de la aplicación *Blynk*. 21
Figura 3.20 Configuración de los botones de la planta baja. 22
Figura 3.21 Espacio de trabajo de la aplicación *Blynk*. 22
Figura 3.22 Configuración de los botones de la planta alta. 23
Figura 3.23 Interfaz móvil 23
Figura 3.24 Solicitud *GET* de *Voiceflow*. 24

Figura 3.25 Parámetros para escribir el valor de un pin. 25
Figura 3.26 Constantes y variables. 25
Figura 3.27 Flujo de bloques en *Voiceflow*. 26
Figura 3.28 Configuración de las credenciales *Wi-Fi*. 27
Figura 3.29 Configuración *Wi-Fi*. 28
Figura 3.30 Establecer conexión con la red. 28
Figura 3.31 Función "*BLYNK*_WRITE". 29
Figura 3.32 Bloque condicional. 29
Figura 3.33 Diagrama de flujo del programa en *WeMos* D1 mini. 30
Figura 3.34 Iniciar comunicación serial. 31
Figura 3.35 Función "*Serial.available()*". 31
Figura 3.36 Planta Baja - Maquetado 32
Figura 3.37 Planta Alta - Maquetado 32
Figura 3.38 Esquema para el control de los dispositivos del hogar. 33
Figura 3.39 Ubicación de dispositivos – Planta Baja 34
Figura 3.40 Ubicación de dispositivos – Planta Alta 34
Figura 3.41 Diagrama esquemático (placa de control). 36
Figura 3.42 Diseño de la PCB en Ares para el circuito de control. 36
Figura 3.43 Diagrama esquemático (módulo relé). 37
Figura 3.44 Diseño de la PCB en Ares para el módulo relé. 38
Figura 3.45 Método de planchado de la placa de control. 39
Figura 3.46 Método de planchado del módulo relé. 39
Figura 3.47 PCB para el circuito de control. 39
Figura 3.48 PCB para el módulo relé. 40
Figura 3.49 Vista frontal y posterior de la placa de control. 40
Figura 3.50 Vista frontal y posterior del módulo relé. 41
Figura 3.51 Módulo relé 16 canales planta baja. 42
Figura 3.52 Módulo relé 16 canales planta alta. 42
Figura 3.53 Módulo relé 16 canales. 42
Figura 3.54 Comunicación *WeMos* y Arduino UNO planta baja. 43
Figura 3.55 Comunicación *WeMos* y Arduino UNO planta alta. 44
Figura 3.56 Sistema de control para la planta domótica. 44
Figura 3.57 Puerta de garaje abierta. 45
Figura 3.58 Control del sistema de iluminación de la planta baja. 45
Figura 3.59 Control del sistema de ventilación. 46
Figura 3.60 Sistema de iluminación planta alta de la vivienda. 46

Figura 3.61 Sistema de calefacción. 47
Figura 3.62 Código QR para el video del manual de uso. 48
Figura 3.63 Código QR para el video del manual de mantenimiento. 49

ÍNDICE DE TABLAS

Tabla 3.1 Tabla de selección de hardware 10
Tabla 3.2 Características de las placas ESP8266 11
Tabla 3.3 Simbología de los dispositivos 35
Tabla 3.4 Resultados de las pruebas de los comandos de voz. 48
Tabla 3.5 Resultados de las pruebas de la interfaz *web* y móvil 48
Tabla 5.1 Comandos de voz para activar los dispositivos de la planta baja y alta ii
Tabla 5.2 Comandos de voz que controlan los dispositivos de la planta baja ii
Tabla 5.3 Comandos de voz que controlan los dispositivos del piso 2. iii

RESUMEN

El presente trabajo de titulación corresponde al desarrollo de un programa en Arduino IDE, el cual permite al usuario controlar varios dispositivos de una vivienda mediante comandos de voz a través de Alexa y la plataforma *Blynk*.

En este documento se encuentran detalladas cada una de las cinco secciones en las que fue dividido el proyecto.

La primera sección describe la introducción, el objetivo general y los objetivos específicos, que incluye una corta explicación de las plataformas Alexa y *Blynk*

La segunda sección detalla la metodología empleada, acorde con los objetivos planteados, así mismo, se muestra el diseño de la maqueta de una vivienda en la cual el sistema fue implementado.

La tercera sección detalla las herramientas y componentes electrónicos que fueron empleados, además, describe el desarrollo de la interfaz humano-máquina.

La cuarta sección describe la disposición de los dispositivos electrónicos en la maqueta, además del procedimiento de construcción de las dos placas PCB: una para la etapa de control y otra para la etapa de potencia.

Por último, en la quinta sección se exponen las pruebas de funcionamiento del sistema, conclusiones y recomendaciones obtenidas a lo largo del desarrollo del proyecto.

PALABRAS CLAVE: Arduino, Alexa, automatización, *Blynk.*

ABSTRACT

This degree work corresponds to the development of a program in Arduino IDE, which allows the user to control various devices in a house by voice commands through Alexa and the Blynk platform.

This document details each of the five sections in which the project was divided.

The first section describes the introduction, the general objective and the specific objectives, and also includes a short explanation of the Alexa and Blynk platforms.

The second section details the methodology used, according to the objectives, and shows the design of a house mockup in which the system was implemented.

The third section details the tools and electronic components that were used and describes the development of the human-machine interface.

The fourth section describes the distribution of the electronic devices in the model, as well as the construction procedure of the two PCB boards: one for the control stage and the other for the power stage.

Finally, the fifth section presents the conclusions and recommendations obtained throughout the development of the project.

***KEYWORDS*:** Arduino, Alexa, *automation*, *Blynk.*

1 INTRODUCCIÓN

En la actualidad, el desarrollo de nuevas tecnologías permiten el control y la automatización de diversos dispositivos de la vivienda de forma remota a través de dispositivos inteligentes, flexibles y económicos, ofreciendo soluciones a las necesidades que surgen con los cambios sociales y las nuevas tendencias de estilo de vida [1].

Estos dispositivos han adquirido una gran importancia en el ámbito educativo, empresarial y/o personal, permitiendo que la interacción del usuario con distintos dispositivos de la vivienda o de la industria requieran de procesos automatizados empleando otros mecanismos de activación como la voz humana, mejorando así la calidad, la funcionalidad y el costo de bienes y servicios.

En el presente proyecto se utilizó la plataforma Arduino debido a que es una plataforma de hardware libre y fácil de utilizar mediante una interfaz sencilla, diseñada para facilitar el uso de la electrónica en diversos proyectos tanto educativos como empresariales.

Adicionalmente, se desarrolló una maqueta la cual tiene como objetivo representar una casa inteligente, mediante la aplicación de la domótica en conjunto con Arduino y Alexa se controlan varios componentes a través de interacciones por comandos de voz.

1.1 Objetivo general

Implementar un sistema domótico inalámbrico controlado por comandos de voz para el hogar.

1.2 Objetivos específicos

- Identificar los requerimientos necesarios para la implementación del prototipo.
- Analizar el hardware adecuado según los requerimientos necesarios para el prototipo.
- Desarrollar la interfaz humano-máquina para el sistema.
- Implementar el prototipo del sistema.
- Realizar pruebas de funcionamiento del prototipo.

1.3 Fundamentos

Amazon Alexa

Alexa es un asistente virtual controlado por voz que está basado en la nube de Amazon, este servicio está impulsado especialmente en los altavoces inteligentes Amazon Echo; Sin embargo, también está disponible para dispositivos de terceros. Además, cuenta con un marco de desarrollo de *software* que permite crear experiencias más personalizadas para los consumidores, llamadas habilidades o *Skills* [2]. Estas habilidades se comportan como aplicaciones para Alexa.

Cuenta con una interfaz de voz interactiva, para que el usuario pueda acceder al contenido de una habilidad, se debe pedir que Alexa invoque dicha habilidad. Además, Alexa siempre se encuentra lista para invocar nuevas habilidades o controlar dispositivos asociados a la nube de Amazon.

Alexa se comunica con la habilidad a través de una interfaz de programación de aplicaciones (*API)* la cual permite que sus productos y servicios se comuniquen con otros, es decir que exista una comunicación entre dos aplicaciones sin la necesidad de saber cómo están implementados [3].

Cuando la habilidad es invocada esta se va a comunicar con la plataforma *Voiceflow* en la cual se van a configurar los comandos de voz necesarios para controlar los dispositivos del presente proyecto. Después, se conectará a la nube de *Blynk* a través de una *API* la cual permite leer y escribir fácilmente valores hacia la aplicación móvil y los microcontroladores consiguiendo de esta manera controlar los dispositivos empleados en el proyecto.

. En la Figura 1.1 se muestra el proceso de invocar una habilidad con el servicio Alexa.

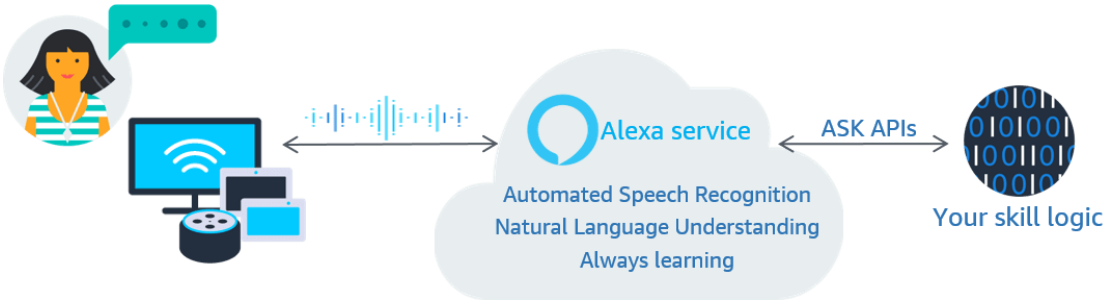


Figura 1.1 Activación por voz de una habilidad con el servicio Alexa [3].

Blynk

Blynk es una plataforma que proporciona infraestructura basada en la nube para el desarrollo, implementación y administración de forma remota de prototipos para el Internet de las Cosas (IoT) [4]. Cuenta con servicios multiplataforma (compatible con iOS, Android y *web*), lo que facilita el análisis de datos, el control de dispositivos de forma remota desde cualquier parte del mundo en tiempo real. La plataforma *Blynk* ofrece cuatro productos principales: Consola, *Apps*, *Edgent* y *Cloud*. Siendo este el corazón de la plataforma *Blynk* IoT [5].

En la Figura 1.2 se muestra el funcionamiento de la nube de *Blynk*, la cual es responsable de unir a través de una conexión a Internet todos los componentes de la plataforma (aplicación móvil, interfaz *web* y microcontroladores). Para facilitar el control y monitoreo de dispositivos de manera remota.

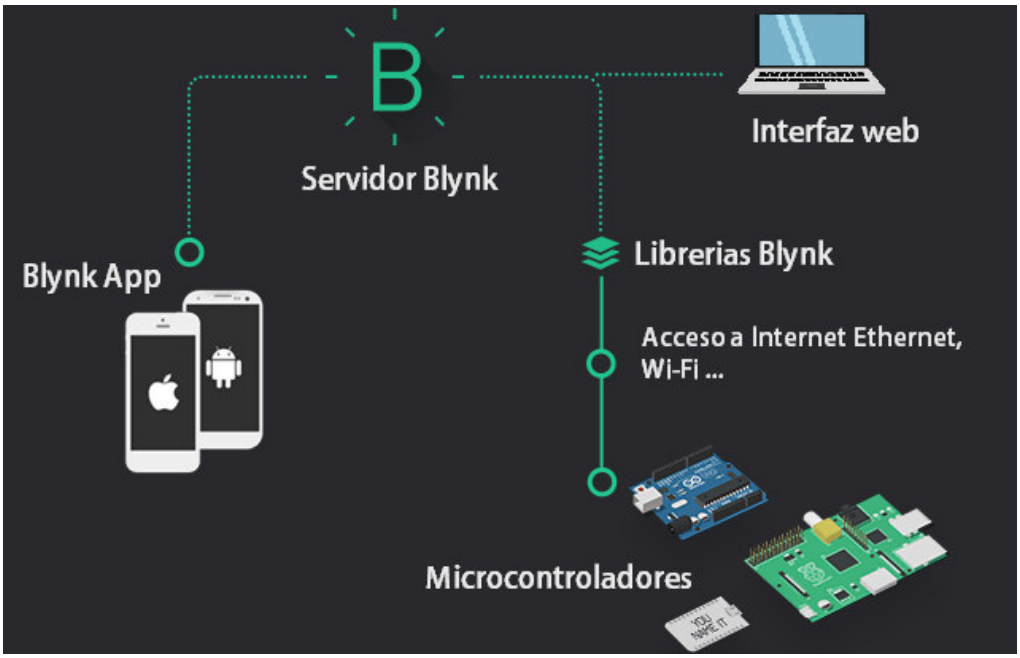


Figura 1.2 Funcionamiento de *Blynk* IoT [6].

Hardware

Arduino es una plataforma de desarrollo de código abierto u *open source*, la cual está basada en una placa de *hardware* libre que incorpora un microcontrolador reprogramable. Convirtiéndose así en una plataforma con un *software y hardware* flexible y fácil de utilizar [7]. Además, cuenta con un entorno de desarrollo integrado, llamado Arduino IDE, el que se encarga de editar, compilar y depurar códigos para el desarrollo de proyectos de baja y muy alta complejidad en cualquier placa Arduino [8].

Arduino UNO

La placa Arduino UNO (ver Figura 1.3) cuenta con un microcontrolador ATmega328p, tiene 14 pines de entrada / salida digital, 6 entradas analógicas. También cuenta con una conexión USB y conector de alimentación tipo CC. Alguna de las características más relevantes de este microcontrolador son [9]:

- Voltaje de funcionamiento 5 (V_{DC}).
- Voltaje de entrada 7-12 (V_{DC}).
- Pines E / S digitales 14.
- Pines de entrada analógica 6.
- Cuenta con puertos para transmitir y recibir datos UART, I2C Y SPI.
- Memoria *flash* 32 KB.
- Velocidad 16 MHz.

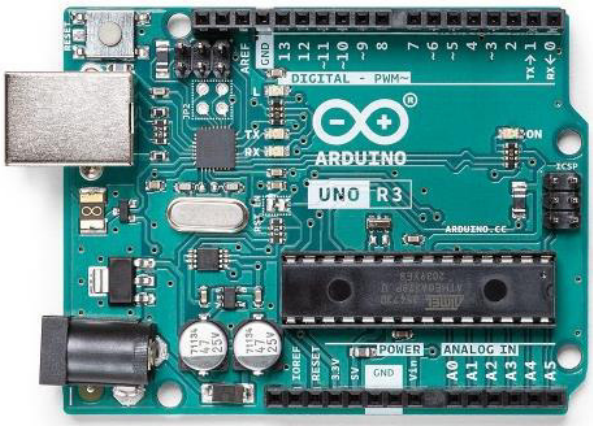


Figura 1.3 Microcontrolador Arduino UNO [9].

WeMos D1 mini

Es una tarjeta de desarrollo que está orientada especialmente al Internet de las cosas (IoT), cuenta con un microcontrolador ESP8266 (ver Figura 1.4), el cual posee una conectividad *Wi-Fi* para ser empleado en el desarrollo de diferentes aplicaciones [10]. La ventaja más importante es la utilización del IDE de Arduino, ya que permite incluir librerías para programar el módulo de forma más sencilla; sin embargo, también es compatible con comandos *Python*, AT [11]. Las características importantes para el desarrollo de este proyecto son [12]:

- Voltaje de funcionamiento 3,3 (V_{DC}).
- Tiene 9 pines de entrada y salida digital, y 1 entrada analógica.
- Memoria *flash* de 4 MB.
- Antena interna incorporada.

- Cuenta con pines para transmitir y recibir datos mediante el protocolo UART.

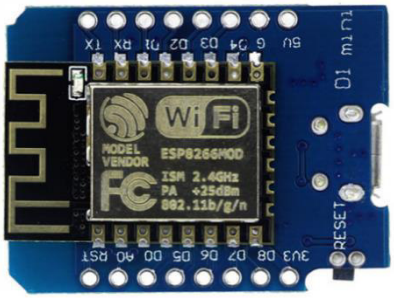


Figura 1.4 *WeMos* D1 mini [13].

A continuación, se realiza un breve resumen de las características proporcionadas por otras placas Arduino diferentes al modelo UNO.

Arduino Nano

El Arduino Nano (Figura 1.5) es una placa pequeña, completa y de gran compatibilidad con las placas de prueba que están basadas en el microcontrolador ATmega328. Carece de un conector de alimentación tipo CC, pero se puede alimentar por medio de un cable micro-USB tipo B. Alguna de las especificaciones más relevantes de esta placa son [13]:

- Voltaje de funcionamiento 5 (V_{DC}).
- Voltaje de entrada 7-12 (V_{DC}).
- Chip USB CH340.
- Cuenta con 14 entradas y salidas digitales, y con 8 entradas analógicas.
- Velocidad 16 MHz.
- Memoria *Flash* 32 KB.

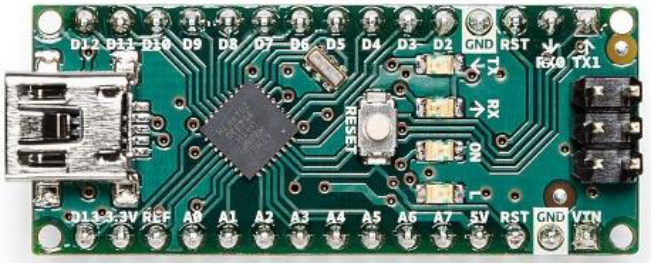


Figura 1.5 Microcontrolador Arduino Nano [13].

Arduino Leonardo

La placa de Arduino Leonardo (Figura 1.6) es otra opción popular para el desarrollo de proyectos, ya que es muy similar a la placa Arduino UNO. La ventaja que ofrece esta placa es la eliminación de un circuito adicional para la comunicación USB, ya que es

reconocido directamente como un teclado o ratón gracias al chip de conexión USB integrado dentro del microcontrolador. Las características más importantes de esta placa son [14]:

- Microcontrolador ATmega32u4.
- Voltaje de funcionamiento 5 (V_{DC}).
- Voltaje de entrada 7-12 (V_{DC}).
- Tiene 20 entradas y salidas digitales y 12 entradas analógicas.
- Memoria *flash* 32 KB.
- Velocidad 16 MHz.

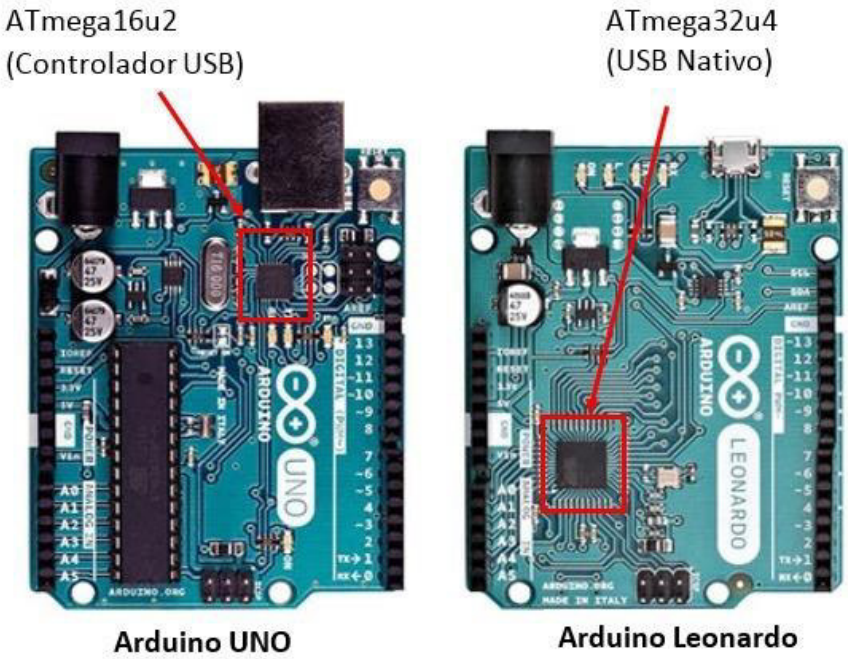


Figura 1.6 Microcontrolador Arduino Leonardo [14].

Arduino Mega 2560

La placa de Arduino Mega 2560 (ver Figura 1.7) está basada en un microcontrolador ATmega2560. Cuenta con 54 pines de entrada y salidas digitales, de los cuales 14 pueden ser utilizadas como salidas analógicas. Además, cuenta con 16 entradas analógicas. En general, esta placa es utilizada en proyectos donde el modelo Uno no cumple con los requerimientos necesarios (entradas, salidas digitales o analógicas), y el requerimiento de más memoria de cualquier tipo: *flash*, SRAM o EEPROM. Algunas de las características importantes de esta placa son [15]:

- Voltaje de funcionamiento 5 (V_{DC}).
- Voltaje de entrada 7-12 (V_{DC}).

- Tiene 54 entradas y salidas digitales, y cuenta con 14 salidas analógicas
- Cuenta con 16 entradas analógicas
- Memoria *flash* 128 - 256 KB
- Velocidad 16 MHz

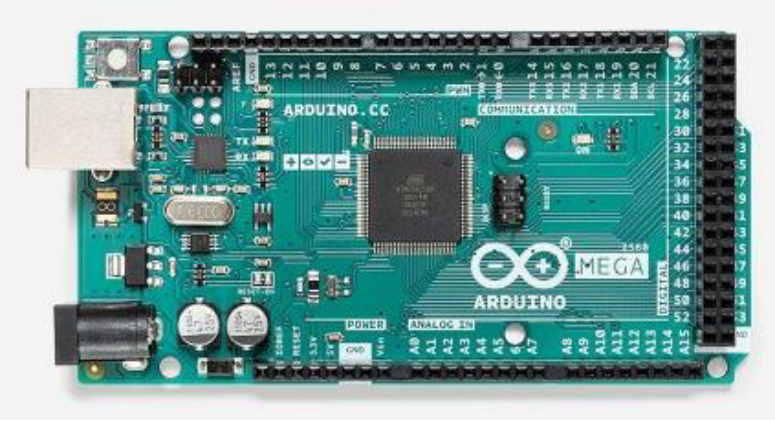


Figura 1.7 Microcontrolador Arduino Mega [15].

2 METODOLOGÍA

2.1 Descripción de la metodología usada

A continuación, se incluye un resumen de la metodología planteada en el plan de titulación, la misma que fue ejecutada para la realización del presente proyecto de titulación:

- En el desarrollo del proyecto se realizó el diseño, modelado y maquetado de una vivienda la cual está dividida en 7 secciones y un garaje. Toda la estructura de la maqueta se encuentra fabricada en madera MDF de 12 (mm) y presenta las siguientes dimensiones: 1 (m) de ancho, 1,5 (m) de largo y 1 (m) de alto. Los planos de la vivienda fueron realizados en el programa de diseño AutoCAD.
- Se determinaron los dispositivos que serán controlados, los cuales son: focos para el sistema de iluminación, ventiladores, servomotor, alarma y un calefactor. Adicionalmente, se realizó una investigación de los diferentes tipos de Arduino para seleccionar el que presente los recursos suficientes para la implementación, en este caso se seleccionaron a Arduino UNO y *WeMos* D1 mini.
- Para el desarrollo de los programas que permiten controlar mediante comandos de voz los dispositivos en la maqueta se utilizó el programa Arduino IDE y las plataformas *Voiceflow* y *Blynk*.

- Se realizó un diagrama en el cual se visualiza la ubicación de los dispositivos electrónicos en la maqueta. Además, se adquirieron los elementos necesarios para realizar el diseño y construcción de dos circuitos impresos (PCB) utilizados para la etapa de control y etapa de potencia.
- Una vez concluido el proyecto, se procedió a realizar las respectivas pruebas de funcionamiento: comunicación entre Arduino UNO, *WeMos* D1 mini y Alexa, funcionamiento de la aplicación *web* y móvil.

Finalmente, se realizó un video en el cual incluye el manual de usuario. En dicho manual se explica cómo establecer la conexión entre Alexa y Arduino para controlar los dispositivos mediante comandos de voz. Además, se visualizará el estado de los dispositivos en la interfaz *web* y móvil.

3 RESULTADOS Y DISCUSIÓN

En el presente proyecto se modeló y diseñó una maqueta de una vivienda para ser controlada mediante comandos de voz empleando Alexa, Arduino UNO, *WeMos* D1 mini, *Voiceflow* y la plataforma *web Blynk*. A continuación, se describen los requerimientos, dispositivos electrónicos, creación interfaz humano-máquina, desarrollo del programa de control, implementación del sistema, pruebas de funcionamiento y manual de uso y mantenimiento.

3.1 Identificación de los requerimientos del prototipo

La finalidad del proyecto fue construir un prototipo de una vivienda domótica que será controlada a través de comandos de voz, mediante la implementación de un asistente de voz de Alexa y de las placas Arduino y ESP8266.

El prototipo fue diseñado en dos fases, lo que permite verificar el funcionamiento de cada fase por separado.

La primera parte orientada al diseño de la etapa de control con las placas Arduino UNO y *WeMos* D1, esta etapa fue diseñada para cumplir con los siguientes requerimientos:

- Conocer cuántos dispositivos se van a implementar en la vivienda y evitar así sobredimensionar las salidas que se utilizarán en el proyecto. Además, debe ser capaz de alimentar a la placa del módulo relé.
- Se encarga de la recepción y del envió de datos al servidor *Blynk* cada vez que un dispositivo de la vivienda es accionado. Además, debe establecer una

comunicación entre los microcontroladores para controlar las salidas digitales del Arduino.

La segunda parte está conformada por el diseño de la etapa de potencia en la cual se construyó un módulo relé de 16 canales, el cual debe cumplir con los siguientes requerimientos:

- Debe ser capaz de controlar los dispositivos de la vivienda que funcionan con un voltaje alto a través de la señal que proviene de la etapa de control.
- Mostrar un identificador para cada canal del módulo relé, para así mostrar al usuario los canales que se encuentran activados.

Para que los dispositivos puedan ser controlados a través de comandos de voz se requiere de un dispositivo que sea capaz de procesar las instrucciones que brinda el usuario. Para esto se utilizó un asistente de voz inteligente Alexa *Echo dot* tercera generación (ver Figura 3.1), el cual mediante su servicio en la nube permite establecer una comunicación con la etapa de control a través de la plataforma *Voiceflow*.

Figura 3.1 Amazon *Echo dot* tercera generación [16].

Adicionalmente, el sistema debe estar conectado siempre a Internet, ya que el prototipo controla los dispositivos del hogar por comandos de voz a través de una conexión *Wi-Fi*.

3.2 Selección del *hardware* adecuado

Después de investigar algunos tipos de Arduino, se realizó una tabla comparativa de los aspectos más importantes (ver Tabla 3.1). En este caso se seleccionó Arduino UNO por las siguientes características:

- Tiene 14 pines digitales (6 salidas PWM), 6 pines analógicos y 2 pines de TX Y RX para la comunicación, los cuales son suficientes para controlar los 16 dispositivos distribuidos en la maqueta.
- Esta placa de Arduino posee una gran ventaja frente a los demás modelos, ya que es el único Arduino que no posee su microcontrolador (Atmega328P) soldado en la placa, es decir es desmontable y se lo puede remplazar fácilmente en el caso de que se averíe, mientras que en los otros modelos si se daña el microcontrolador se debería reemplazar toda la placa lo que generaría gastos económicos.
- Otro factor que se tomó en cuenta es que tiene mayor soporte, debido que la mayoría de las librerías y ejemplos de diversas aplicaciones se realiza basándose en este modelo por lo que es menos probable que aparezcan errores, pues en la actualidad es el Arduino más utilizado.

Tabla 3.1 Tabla de selección de hardware [15] [13] [9]

Especificaciones	Tipo de Arduino			
	Uno	**Nano**	**Leonardo**	**Mega 2560**
Procesador	ATmega328P	ATmega328P	ATmega32u4	ATmega2560
Velocidad reloj	16 MHz	16 MHz	16 MHz	16 MHz
Memoria RAM	2 KB	2 KB	2.5 KB	8 KB
Memoria EEPROM	1 KB	1 KB	1 KB	4 KB
Memoria *Flash*	32 KB	16 – 32 KB	32 KB	128 – 256 KB
Voltaje Funcionamiento / Entrada	5 (V_{DC}) / 7 – 12 (V_{DC})	5 (V_{DC}) / 7 – 12 (V_{DC})	5 (V_{DC}) / 7 – 12 (V_{DC})	5 (V_{DC}) / 7 – 12 (V_{DC})
Entradas / Salidas Digitales	14	14	20	54
Entradas / Salidas Analógicas	6	8	12	16
Pines PWM	6	6	7	15
Precio	9.99 USD	7.99 USD	10.99 USD	19.99 USD

En base a la Tabla 3.2 se obtiene que la tarjeta de desarrollo que se empleó en el presente proyecto junto con Arduino UNO es el *WeMos* D1 mini (ESP8266) debido a los siguientes aspectos:

- Posee un entorno de desarrollo amigable, fácil de utilizar y que es compatible con Arduino IDE. Además, cuenta con la capacidad de manejar completamente el protocolo TCP/IP ya que integra una conexión *Wi-Fi*.
- Puede ser implementado en espacios reducidos ya que viene preensamblada y lista para usar, permitiendo mejorar la disposición dependiendo de los requerimientos físicos del proyecto.
- Esta placa posee 12 pines, de los cuales se utilizaron los pines de TX y RX para establecer la comunicación con la tarjeta Arduino. Por lo que los dos pines de comunicación son suficientes para ser implementado en el proyecto.

Tabla 3.2 Características de las placas ESP8266 [17].

Especificaciones	ESP-01	*WeMos* D1 mini	NodeMCU
Microcontrolador	Esp8266EX	ESP8266EX	ESP8266EX
Velocidad reloj	80 MHz	80 – 160 MHz	80 – 160 MHz
Memoria *flash*	1 MB	4 MB	4 MB
Alimentación	2.5 -3.6 (V_{DC})	3.3 (V_{DC})	3.3 – 5 (V_{DC})
Protocolo *Wi-Fi*	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Número de pines	8	12	18
Señales de salida	Digital	Analógica / Digital	Analógica / Digital
Precio	5 USD	7.50 USD	8 USD
Dimensiones	25*15*3 mm	35*26*12 mm	49*26*12 mm

Adquisición de materiales

En base al cumplimiento de los requerimientos, una vez determinados los tipos de materiales para el desarrollo del módulo, se procedió a adquirirlos.

Resistencias

Son parte del módulo relé, se emplearon en total 32 resistencias: 16 resistencias de entrada de 1 (kΩ) y las otras 16 resistencias para los *LEDS* con un valor de 330 y 220 (Ω) (ver Figura 3.2).

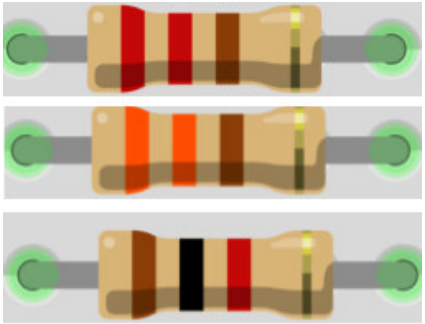


Figura 3.2 Resistencias de 220, 330 y 1k (Ω).

Diodos

Se utilizaron 16 diodos 1N4001 (ver Figura 3.3) que son rectificadores de propósito general, tiene la función de conducir la corriente en una sola dirección. Se emplearon en paralelo de la bobina del relé para prevenir un retorno de corriente y un incremento de tensión en los extremos cuando la bobina de los relés se desenergice.

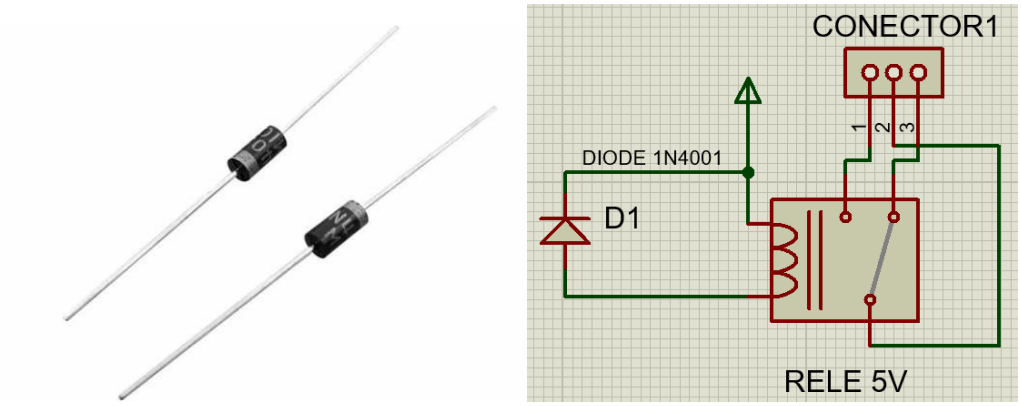


Figura 3.3 Diodo 1N4001 [18].

Transistor

Para el módulo relé de 16 canales se implementó transistores 2N2222A (ver Figura 3.4), son dispositivos que pueden amplificar señales bajas y funcionan como circuitos de conmutación que permiten el control de los relés. Se implementó estos transistores ya que pueden ser accionados desde la placa de control. Además, el rango de la corriente del colector I_C (600 mA) es suficiente para soportar la corriente que atraviesa la bobina del relé I_B (71,4 mA) [19] [20].

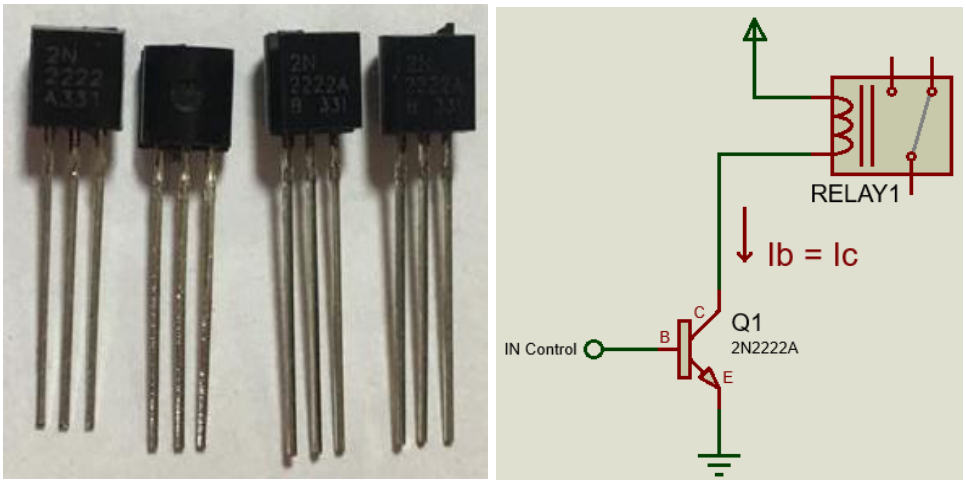


Figura 3.4 Transistores 2N2222A.

LEDS

Los *LEDs* de alto brillo (ver Figura 3.5) son componentes del módulo relé los cuales son independientes y sirven como un indicador de estado, se encienden cuando un canal del módulo relé está activo.

Figura 3.5 Diodo *LED* [21].

Relés

Los relés (ver Figura 3.6) son componentes importantes, ya que permiten cerrar o abrir circuitos electrónicos independientes desde la placa de control. En este caso se implementaron 16 relés SONGLE SRD debido a que necesitan 5 (V_{DC}) para su funcionamiento, lo que los hacen compatibles con el voltaje de los microcontroladores para ser alimentados. Además, soportan voltajes hasta 220 (V_{AC}) suficientes para el desarrollo del proyecto.

Figura 3.6 Relé 5 (V_{DC}) [22].

Baquelita

Para obtener una mejor distribución de los componentes electrónicos y evitar que las conexiones por medio de cables produzcan interferencias, se optó por realizar dos circuitos impresos (PCB) en baquelitas de fibra de vidrio (ver Figura 3.7). El primer circuito impreso conecta al Arduino UNO con el *WeMos* D1 mini, mientras que el segundo circuito impreso aloja los componentes del módulo relé.

Figura 3.7 Baquelita.

Borneras

Para transportar las señales de control y las salidas de los relés se emplearon borneras (ver Figura 3.8). Las cuales permiten alimentar a los terminales y evitar falsos contactos, garantizando un funcionamiento óptimo.

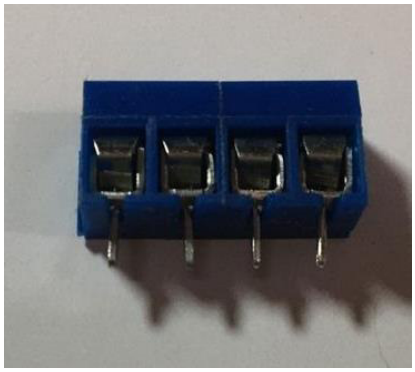


Figura 3.8 Borneras.

Servomotor

Se empleó un servomotor modelo SG90 (ver Figura 3.9), el cual permite controlar su rotación en 180 grados para el accionamiento de la puerta de garaje de la vivienda. También cuenta con dimensiones reducidas, lo que resulta ideal para ser implementado en el presente proyecto. Además, sus requerimientos de energía son bajos y es compatible con microcontroladores que funcionen a 5 (V_{DC}) permitiéndole ser alimentado directamente desde el circuito de control.

Figura 3.9 Servomotor SG90.

Sirena

Para el sistema de seguridad se optó por implementar una sirena zumbadora (ver Figura 3.10), la cual advierte de una situación de pánico o anormal para el usuario. Cumple con la función de reducir el tiempo de ejecución frente a un problema reduciendo así futuras pérdidas.

Figura 3.10 Sirena

Ventilador

En la Figura 3.11 se muestra el ventilador que se usó para simular el aire acondicionado de la planta baja y planta alta de la vivienda, el cual permite renovar y enfriar el aire en una estancia de la vivienda. Además, sus dimensiones 80 x 80 x 25 (mm) permiten que este dispositivo pueda ser implantado en espacios reducidos.

Figura 3.11 Ventilador.

Luminarias

Para el sistema de iluminación de la vivienda se emplearon focos *LEDs* con sus respectivas boquillas (ver Figura 3.12), las que deben ser alimentadas desde la placa del módulo relé con un voltaje 110 (V_{AC}). Estos elementos se encargan de iluminar las zonas interiores y exteriores de la vivienda.

Figura 3.12 Foco Led y boquilla.

3.3 Interfaz humano-máquina

Para el desarrollo de la interfaz humano-máquina se optó por utilizar la plataforma *web Blynk*. Está plataforma brinda una mayor versatilidad, ya que no está sujeta a ningún hardware en específico. Además, puede ser manejado por personas sin ningún tipo de experiencia en la creación de proyectos de IoT.

Por otra parte, el panel *web* permite configurar servicios, administrar dispositivos, personas, empresa y datos. Convirtiéndose así en una interfaz *web* para la creación de proyectos IoT aplicables a la vivienda como también a la industria. Estas son algunas de las características que impulsaron la utilización de la interfaz *web Blynk*. En la Figura 3.13 se muestra la interfaz *web Blynk*.

Para acceder a la Interfaz de usuario se debe crear una cuenta en la plataforma *Blynk* IoT. Una vez se haya iniciado sesión, se accede al panel principal de la interfaz, que está conformado por 3 áreas importantes menú, barra lateral y vista principal del panel como se aprecia en la Figura 3.13.

En la Figura 3.14 se muestra la creación de la plantilla que contiene los dispositivos que serán controlados a través de los microcontroladores, se debe asignar un nombre, seleccionar el tipo de conexión que se llevará a cabo y el microcontrolador que se utilizará.

En la Figura 3.15 se muestra la configuración de los pines virtuales que sirven para el control de los dispositivos de la vivienda. Por otra parte, se debe añadir el nombre con el cual va a responder el pin virtual, el tipo de dato y el valor mínimo y máximo de activación.

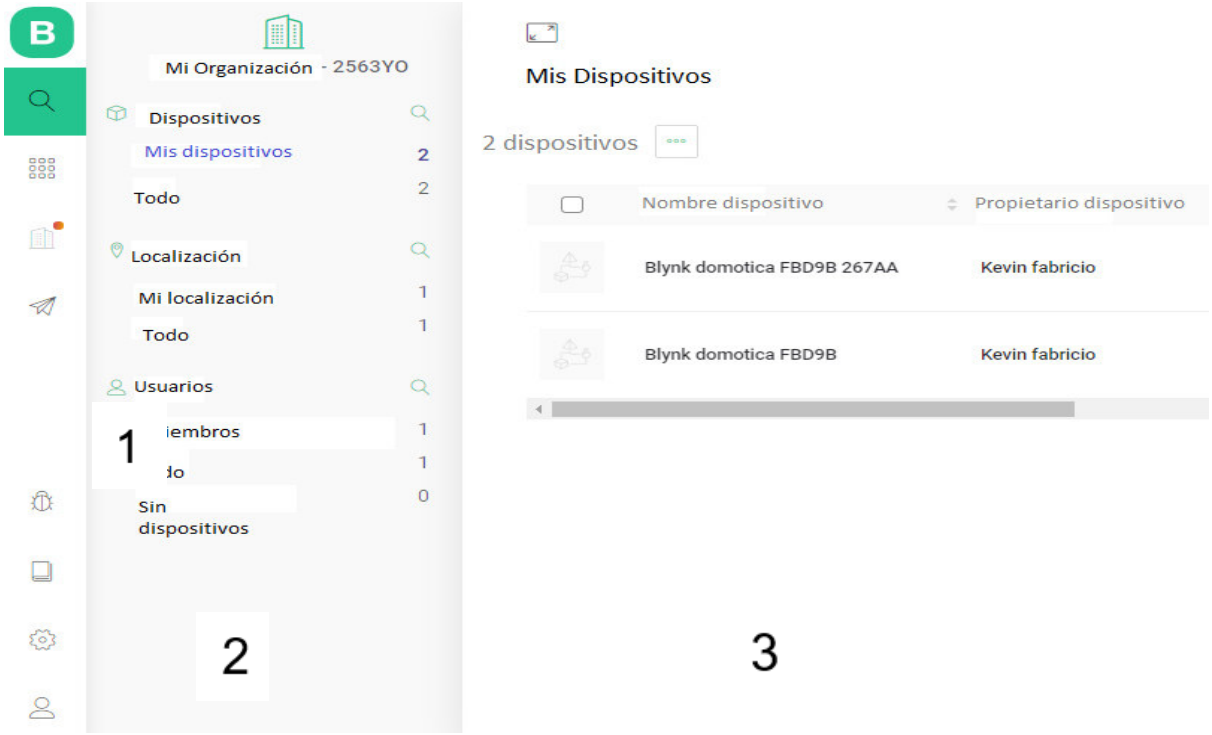


Figura 3.13 Interfaz *web Blynk*.

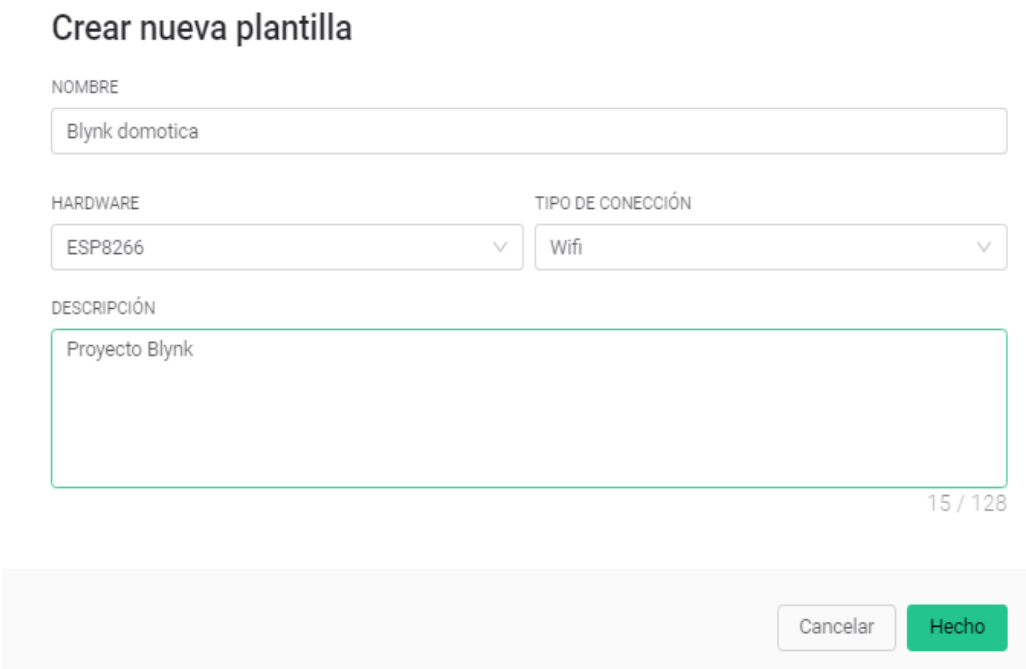


Figura 3.14 Crear una plantilla en *Blynk*.

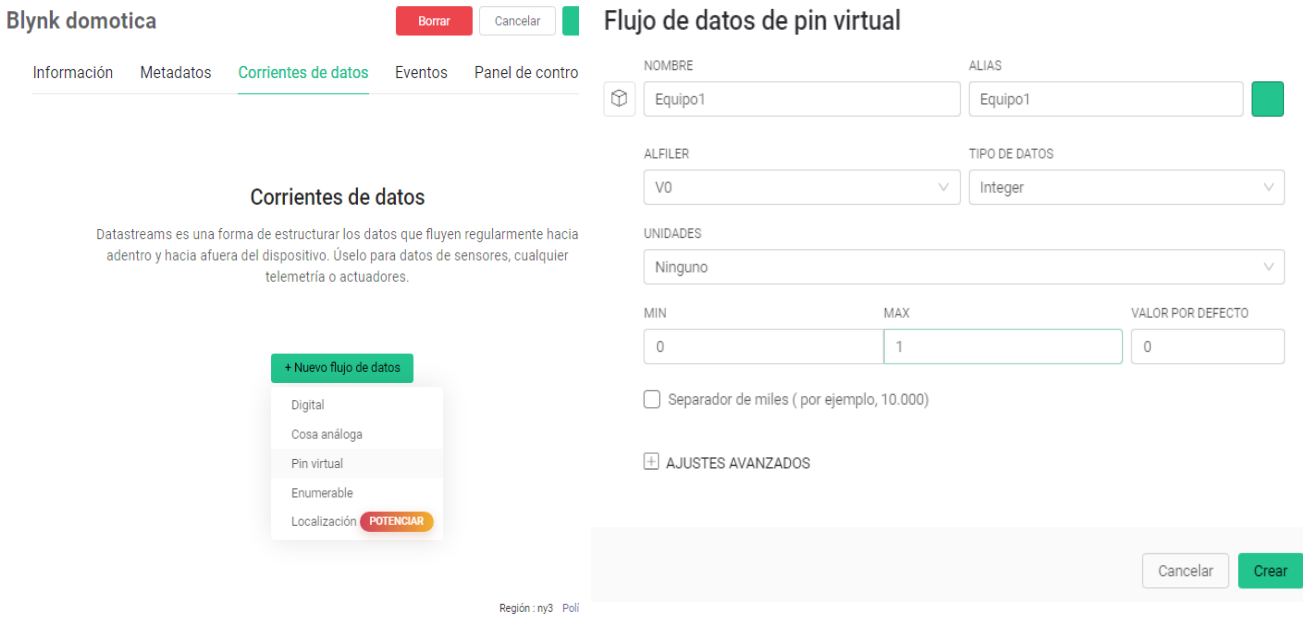


Figura 3.15 Configuración de pin virtual.

En el panel de control *web* se configuraron los botones que sirven para activar o desactivar cualquier dispositivo de la vivienda (ver Figura 3.16).

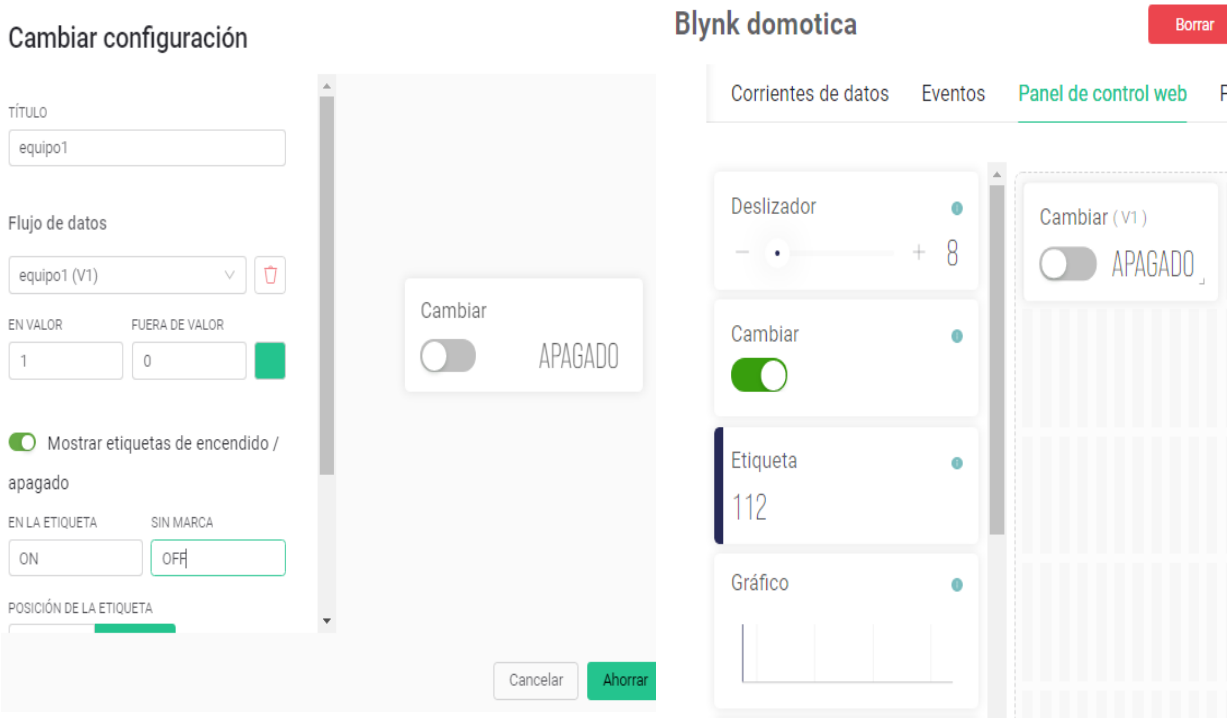


Figura 3.16 Configuración de botones.

En la Figura 3.17 se aprecia el panel de control *web* que permite el control de los 16 equipos de la vivienda.

Figura 3.17 Panel *web* completo.

Finalmente, la plataforma también ofrece una interfaz móvil a través de su aplicación que se accede mediante la *Play Store* (en dispositivos móviles con Android) o la *App Store* (en dispositivos móviles con *Apple O.S.*) de los teléfonos inteligentes, la cual permite controlar dispositivos, editar el espacio de trabajo y seleccionar el tipo de botón que se utiliza para el control de los dispositivos de la vivienda.

Para editar el espacio de trabajo se debe ingresar a la aplicación y seleccionar el proyecto que se desea controlar, para añadir los botones que controlarán los dispositivos se debe seleccionar la opción de configuración como se muestra en la Figura 3.18.

En la Figura 3.19 se puede visualizar el espacio de trabajo preparado para ser configurado, para añadir los ficheros que permitirán el control de los dispositivos del hogar se debe seleccionar la opción de añadir, en donde aparecerán los ficheros que pueden ser utilizados en el proyecto.

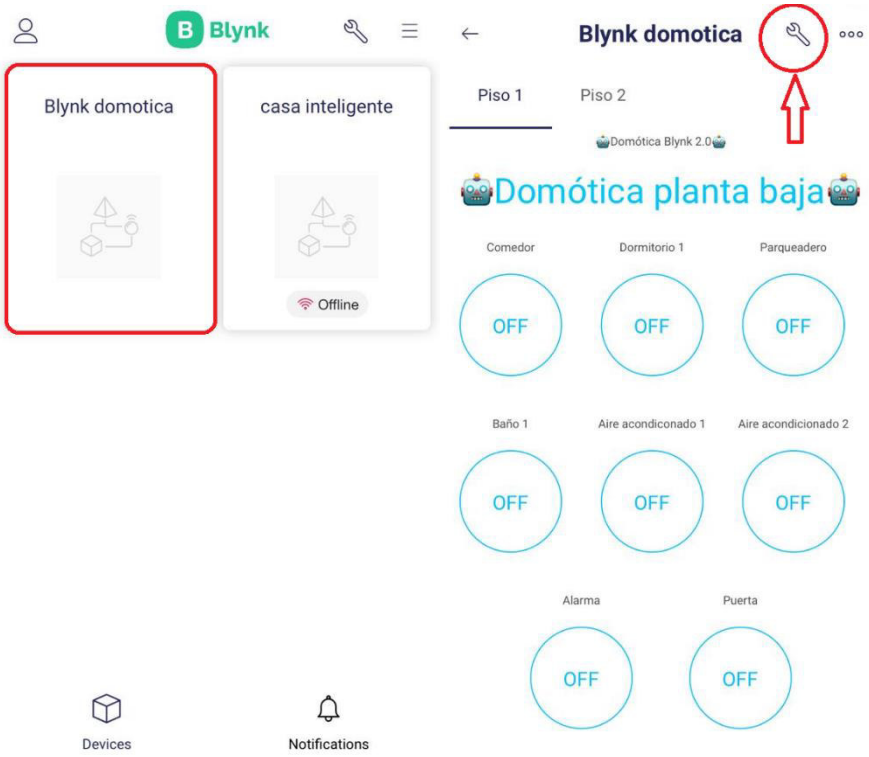


Figura 3.18 Configuración de la aplicación *Blynk.*

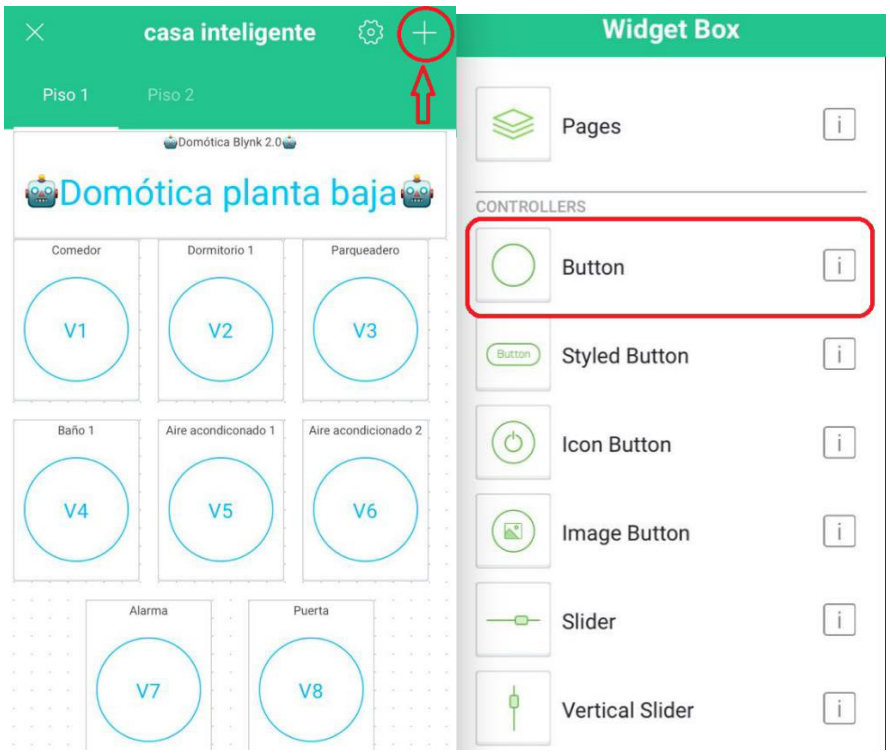


Figura 3.19 Espacio de trabajo de la aplicación *Blynk.*

En la Figura 3.20 se muestra la selección y configuración de los botones que son utilizados en la planta baja de la vivienda. Una vez seleccionado el tipo de botón que se

utilizó en el presente proyecto. Lo primero que se debe realizar es asignar un identificador al botón. Después, se debe seleccionar el pin virtual que está asociado con el dispositivo del hogar que se está configurando. Finalmente, se debe seleccionar el modo de accionamiento del botón.

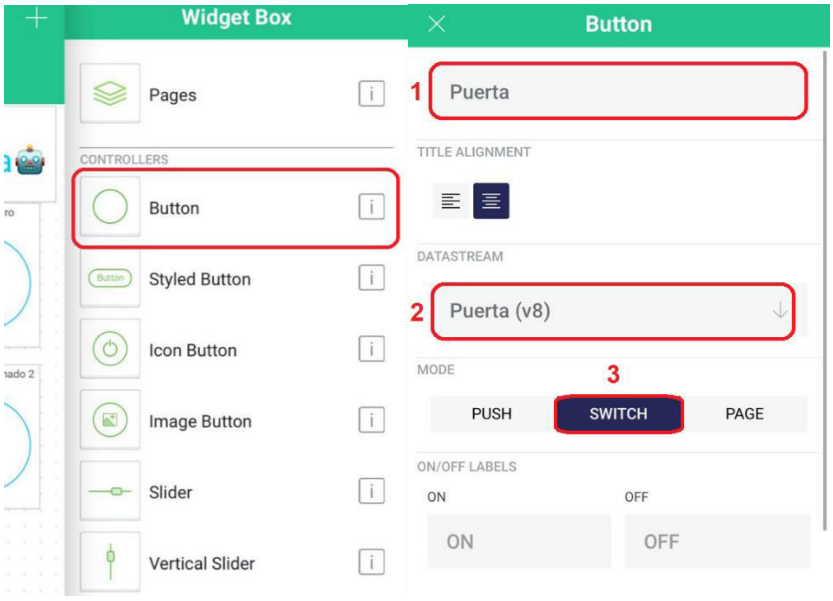


Figura 3.20 Configuración de los botones de la planta baja.

Para configurar el espacio de trabajo y añadir los botones que permitirán el control de los dispositivos de la planta alta de la vivienda, se debe seleccionar la opción de añadir en dónde se desplegarán los controles que pueden ser utilizados en el presente proyecto como se muestran en la

Figura **3.21**.

Figura 3.21 Espacio de trabajo de la aplicación *Blynk.*

En la Figura 3.22 se muestra la selección y configuración de los botones que fueron utilizados en la planta alta de la vivienda. Una vez seleccionado el tipo de botón que se utilizó. Lo primero que se debe hacer es asignarle un identificador al botón. También se debe seleccionar el pin virtual que está asociado con el dispositivo del hogar que se está configurando.

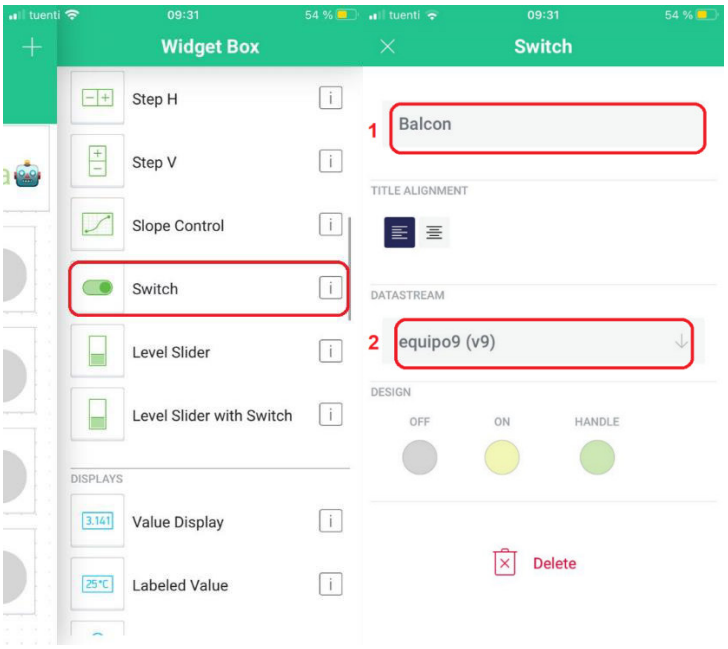


Figura 3.22 Configuración de los botones de la planta alta.

En la Figura 3.23 se muestra la interfaz móvil que permite el control de los dispositivos de la planta baja y planta alta de la vivienda.

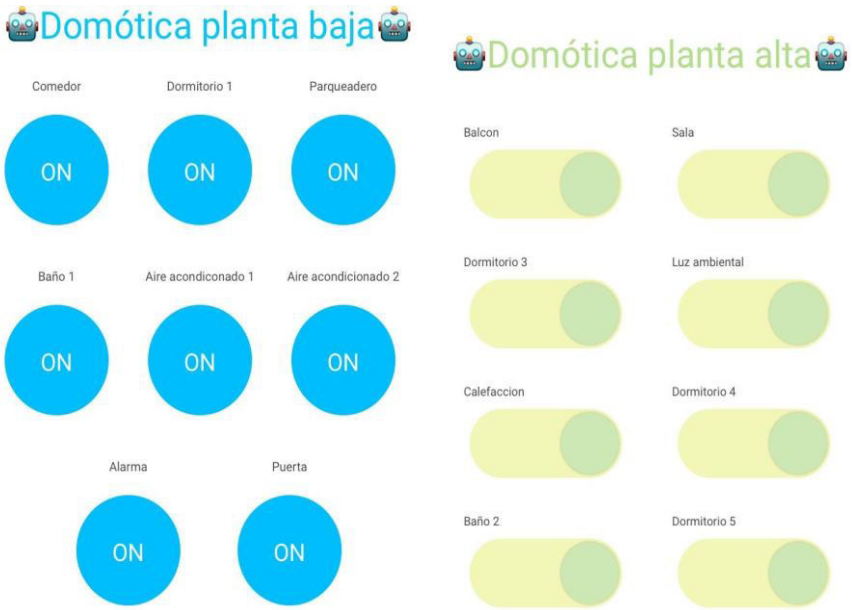


Figura 3.23 Interfaz móvil

Desarrollo del programa de control

Para el desarrollo de los programas que permitirán controlar mediante comandos de voz las salidas digitales, recibir datos del módulo *WeMos* D1 mini y a su vez subir datos al servidor *Blynk*, se utilizó el programa Arduino IDE y las plataformas *Voiceflow* y *Blynk*.

Comandos de voz en *Voiceflow*

Para lograr el control de dispositivos mediante comandos de voz se utilizó la plataforma *Voiceflow*, la cual permite crear habilidades o *Skills* de Alexa de forma visual, es decir sin codificar. Se maneja bloques en los cuales se debe introducir la información de activación de la *Skill*, la información de los dispositivos que se desea controlar entre otros aspectos.

Por otra parte, para lograr que los comandos de voz interactúen con los microcontroladores se utiliza API RESTful HTTP de *Blynk*. La cual permite el manejo de pines virtuales, lectura y escritura de valores desde y hacia los pines en la aplicación *Blynk* y en las tarjetas Arduino UNO y *WeMos* D1 mini.

Como se muestra en la Figura 3.24 *Voiceflow* envía una solicitud *GET* hacia *Blynk*, que especifica cuando se va a activar o desactivar un dispositivo mediante comandos de voz, debido a que esta solicitud permite leer y escribir el valor o estado actual de un pin determinado.

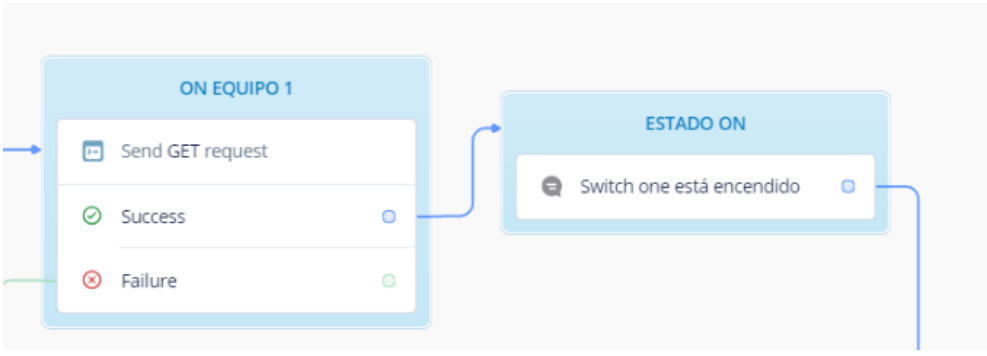


Figura 3.24 Solicitud *GET* de *Voiceflow*.

Para que la solicitud sea exitosa se deben agregar los siguientes parámetros: *URL* del servidor *Blynk*, control de acceso del dispositivo y el valor del pin con el cual se ejecutará esta solicitud. En la Figura 3.25 se muestra la estructura de la solicitud *GET*.

```
GET http://blynk-cloud.com/auth_token/update/pin?value=value
```

Figura 3.25 Parámetros para escribir el valor de un pin.

En la Figura 3.27 se muestra el flujo de bloques de *Voiceflow* para el control de 1 dispositivo mediante comandos de voz. Este procedimiento se debe realizar para todos los dispositivos que se deseé controlar.

Para controlar los dispositivos de la planta baja y planta alta de la vivienda se crearon dos *Skills,* las que serán activadas mediante comandos de voz creadas en *Voiceflow*.

Los comandos de voz para controlar los dispositivos de la planta baja y planta alta de la vivienda se muestran en la Tabla 5.1, Tabla 5.2 y Tabla 5.3 del Anexo 4.

Envió de datos al servidor *Blynk* en *WeMos* D1 mini

Para la placa *WeMos* D1 mini se desarrolló un programa que permitió enviar datos hacia la placa Arduino UNO para el control de cada elemento de la vivienda. Además, se estableció una conexión con el servido *Blynk*.

En la Figura 3.26 se muestra la parte inicial del programa en donde se agregan las librerías, se declaran las variables necesarias. Adicionalmente, contiene el token de autenticación para acceder al servidor, todos estos elementos se encuentran definidos a través del componente “#define”. Es un componente útil en el lenguaje C, ya que permite nombrar un valor constante antes que el programa se compile. Estas constantes definidas son de gran utilidad para la placa Arduino ya que no ocupa espacio de memoria de programa del microprocesador pues antes que el código sea compilado es remplazado por su valor [23].

```
#define BLYNK_TEMPLATE_ID "TMPLEnZDR3rF"              //ID de la plantilla
#define BLYNK_DEVICE_NAME "Blynk domotica FBD9B"      //Nombre del dispositivo
char auth[] = "kf1uBdWzGsJbBawRLgfe1gPNqiV_N0uI";     //Control de acceso

#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#include "BlynkEdgent.h"

#define USE_WEMOS_D1_MINI
```

Figura 3.26 Constantes y variables.

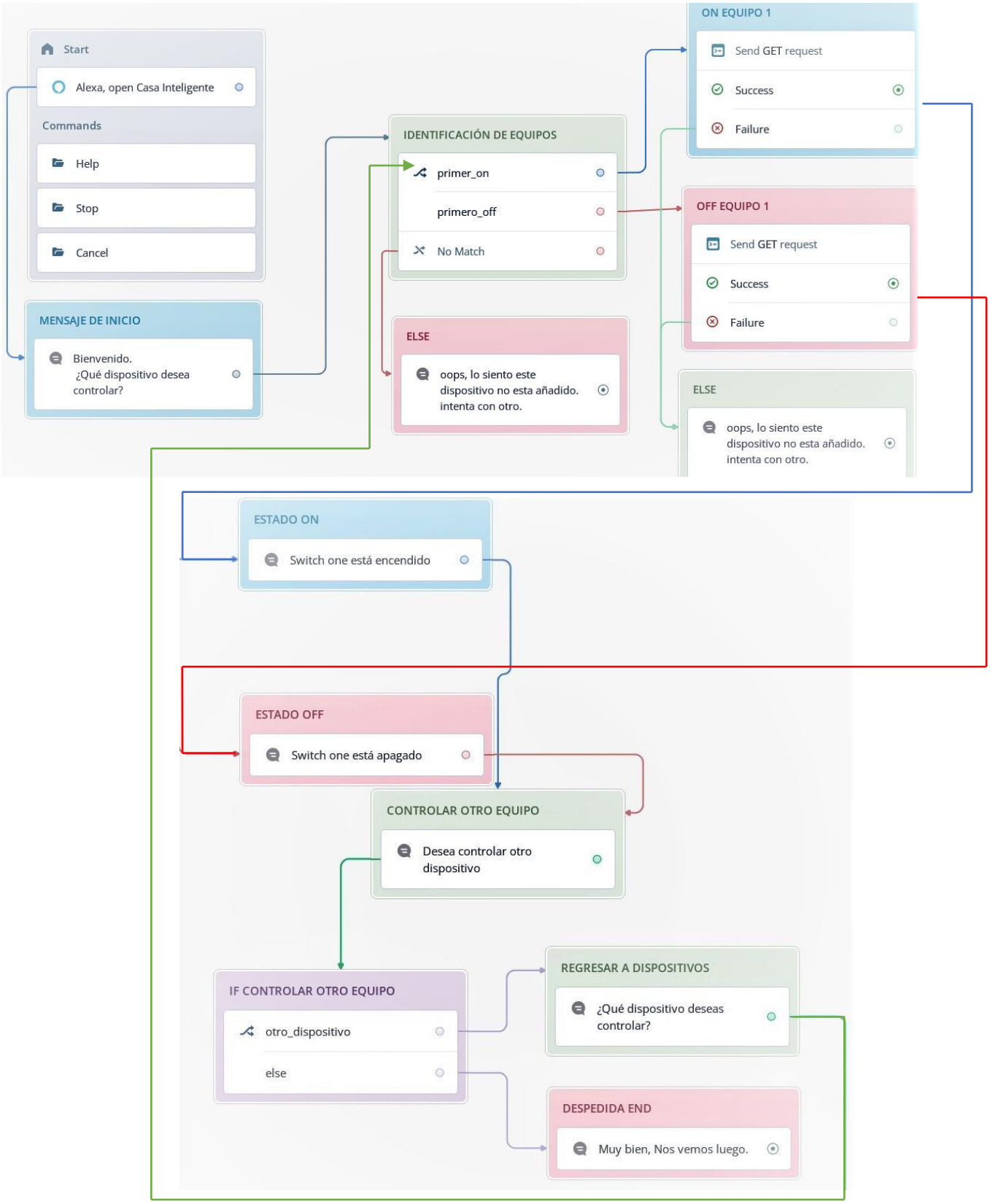


Figura 3.27 Flujo de bloques en *Voiceflow*.

La plataforma *Blynk* implementa la librería *Blynk.Edgent* la cual está disponible en *GitHub*, para conectar las tarjetas de desarrollo a una red *Wi-Fi* sin la necesidad de añadir las credenciales de la red en el código del programa, para esto la plataforma incluye un código de firmware y una interfaz móvil para guiar al usuario en la configuración de la red *Wi-Fi*.

Cuando el código se encuentra cargado en la tarjeta *WeMos* D1 mini, esta se comporta como un punto de acceso (AP), es decir el dispositivo va a transmitir su propia red *Wi-Fi* con el nombre de "*Blynk* domotica-267AA"(Figura 3.28), utilizando la aplicación móvil el teléfono se va a conectar al AP de la tarjeta de control, en donde se deben configurar las credenciales de la red *Wi-Fi* a la cual se desea conectar, como se muestra en la Figura 3.28.

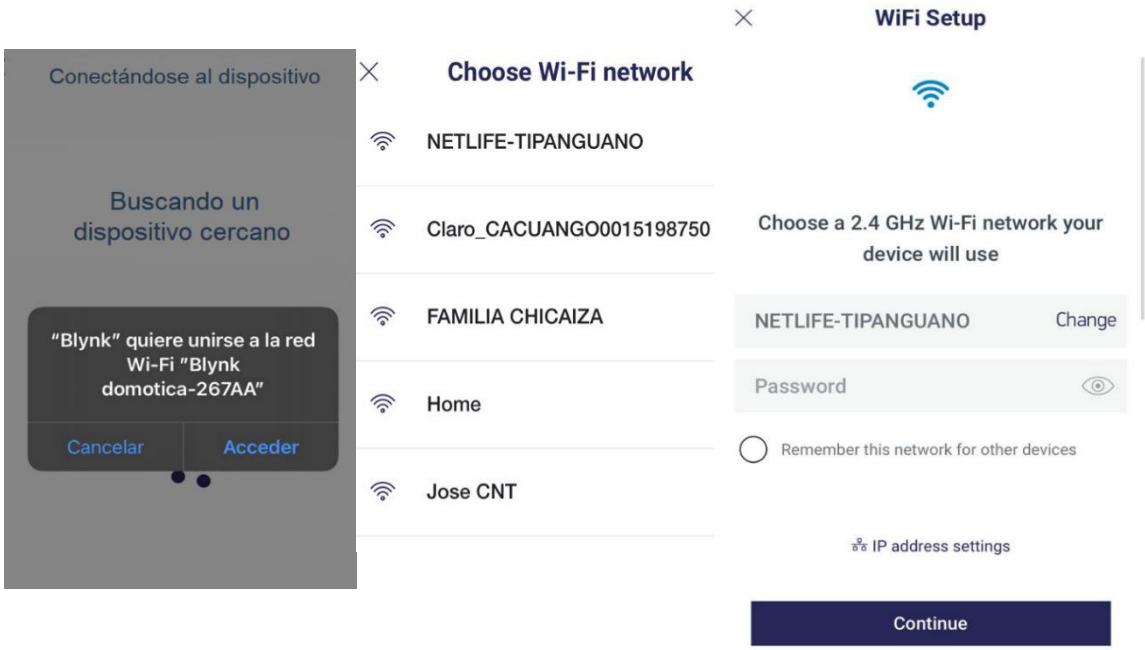


Figura 3.28 Configuración de las credenciales *Wi-Fi*.

Una vez se ingresaron las credenciales *Wi-Fi* estas son enviadas hacia la tarjeta *WeMos* D1 mini como se muestra en la Figura 3.29, la cual guarda estos elementos en la memoria *Flash* / EEPROM para conectarse a la red *Wi-Fi* una vez la configuración del dispositivo esté completa y la tarjeta de desarrollo se haya reiniciado automáticamente. Finalmente, el modo AP se apaga cuando la configuración está completa y el dispositivo está listo para ser utilizado como se puede visualizar en la Figura 3.29 [24].

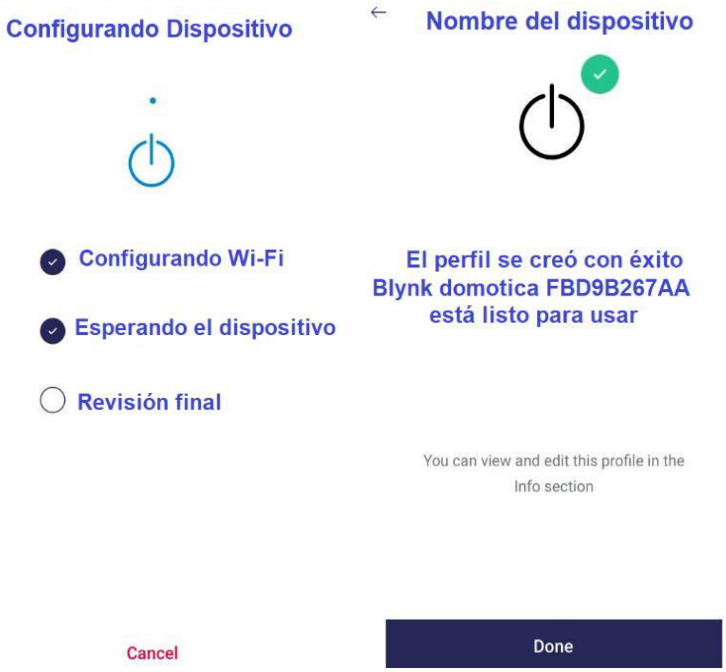


Figura 3.29 Configuración *Wi-Fi*.

En la Figura 3.30 se muestra la función de la librería *Blynk* con la cual se hace efectiva la conexión a la red *Wi-Fi*. Además, emite una advertencia en el caso que la conexión falle.

```
void enterConnectNet() {
  BlynkState::set(MODE_CONNECTING_NET);
  DEBUG_PRINT(String("Connecting to WiFi: ") + configStore.wifiSSID);

  WiFi.mode(WIFI_STA);
  if (WiFi.status() == WL_CONNECTED) {
    IPAddress localip = WiFi.localIP();
    if (configStore.getFlag(CONFIG_FLAG_STATIC_IP)) {
      BLYNK_LOG_IP("Using Static IP: ", localip);
    } else {
      BLYNK_LOG_IP("Using Dynamic IP: ", localip); }

    BlynkState::set(MODE_CONNECTING_CLOUD);
  } else {
    config_set_last_error(BLYNK_PROV_ERR_NETWORK);
    BlynkState::set(MODE_ERROR); }}
```

Figura 3.30 Establecer conexión con la red.

En la Figura 3.31 se muestra la función *BLYNK_WRITE (vPIN)* utilizada para el intercambio de datos entre la plataforma *Blynk* y el hardware implementado utilizando los pines virtuales. Esta función es llamada cada vez que el dispositivo recibe una

actualización del pin virtual proveniente del servidor o de la aplicación. La característica más relevante de esta función es que no puede ser utilizada dentro de ningún bucle ya que es una función independiente.

```
BLYNK_WRITE(V1)  // PIN VIRTUAL PARA ENCENDER EL  EQUIPO 1
{
  int Pinvalue  = param.asInt();

}
```

Figura 3.31 Función "*BLYNK*_WRITE".

En la Figura 3.32 se visualiza el bloque condicional que permite ejecutar una instrucción en el caso de cumplir con la condición establecida mediante programación. Primero se va a ejecutar la función *BLYNK_WRITE* la cual transmite los valores hacia el pin virtual (V1) solamente cuanto estos se hayan actualizado desde la aplicación móvil o desde el servidor. Después, se compara el valor obtenido de la instrucción *param.asInt* y si cumple con la condición establecida, se encargará de comunicarse con la placa Arduino UNO a través del puerto serial.

```
BLYNK_WRITE(V1)  // PIN VIRTUAL PARA ENCENDER LOS DISPOSITIVOS DE LA PLANTA BAJA
{
  int Pinvalue  = param.asInt(); // OBTIENE UN VALOR COMO ENTERO
  if ( Pinvalue == 1) {          // COMPARACIÓN DEL VALOR ENTERO

    Serial.println("A");         // CONDICIÓN PARA ENCENDER EL EQUIPO 1
                                 // A TRAVÉS DEL PUERTO SERIAL
  } else {

    Serial.println("B");         // CONDICIÓN PARA APAGAR EL EQUIPO 1
  }                              // A TRAVÉS DEL PUERTO SERIAL
}
```

Figura 3.32 Bloque condicional.

El diagrama de flujo correspondiente al programa de la placa *WeMos* D1 mini se puede visualizar en la Figura 3.33. Además, en la *URL* https://bit.ly/3EXyKtn se encuentra el código completo y comentado.

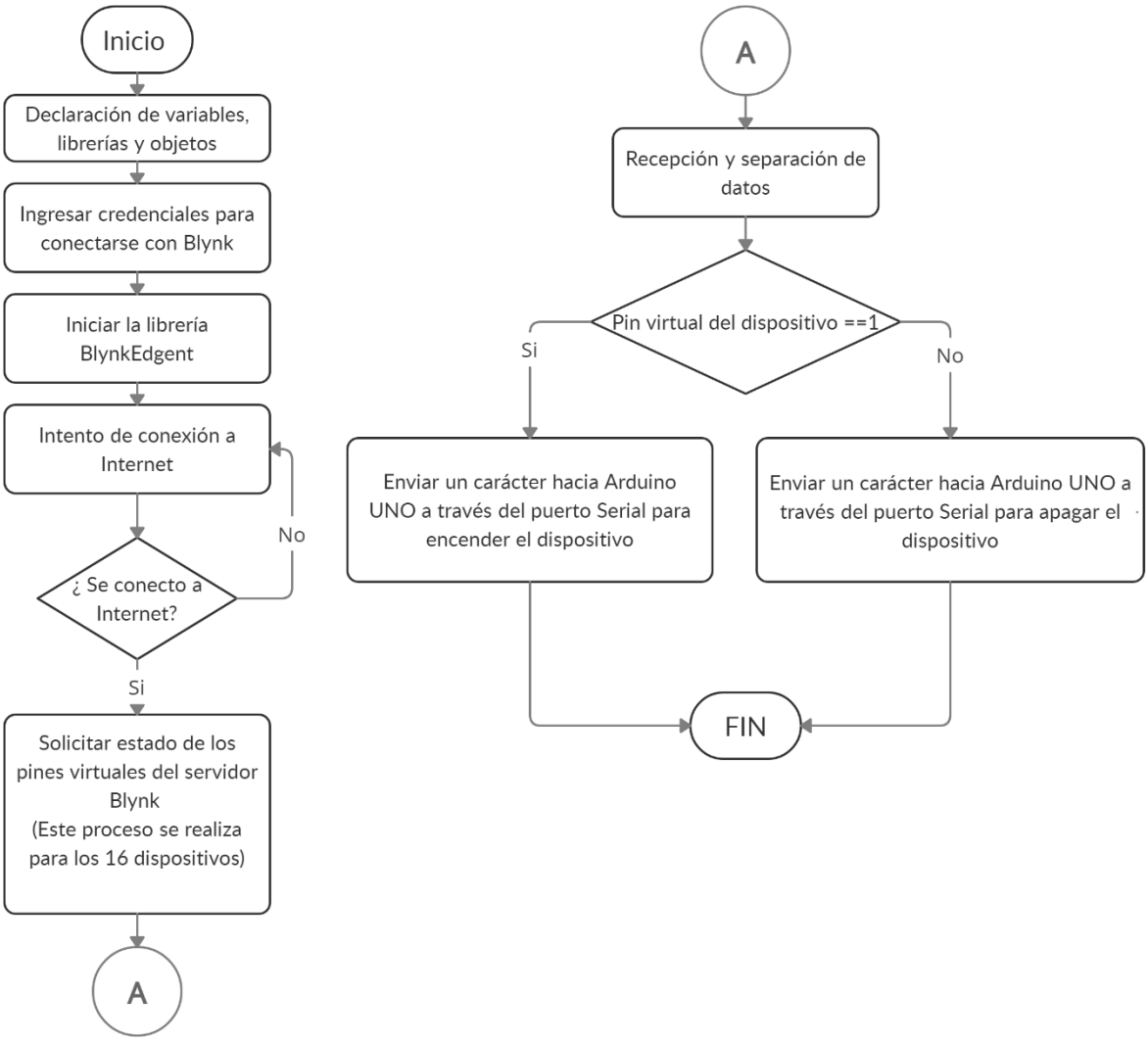


Figura 3.33 Diagrama de flujo del programa en *WeMos* D1 mini.

Lectura del código Arduino

La placa Arduino UNO recibe datos del módulo *WeMos* D1 mini y se ajusta en consecuencia a estos el acondicionamiento por medio de programación. La comunicación entre el Arduino UNO y el módulo *WeMos* D1 mini se realiza por medio del puerto serial (pines 0 y 1 de Arduino UNO).

Para la recepción de los datos que son enviados al puerto serial de la placa Arduino UNO se declara una variable para recibir los datos en forma de carácter y se debe iniciar la comunicación serial en la función *void setup ().* Como se muestra en la Figura 3.34.

```
char recibe;  //Variable para recibir el buffer de comunicacion

void setup() {
     Serial.begin(9600);   //Comunicacion establecida en 9600 baudios
```

Figura 3.34 Iniciar comunicación serial.

Para recibir los datos desde el módulo *WeMos* D1 mini hacia al Arduino UNO, se utiliza la función *Serial.available()* que devuelve el número de caracteres disponibles para leer, si no hay ninguno disponible devuelve un 0. Además, se utiliza la función *Serial.read()* para capturar el byte o carácter recibido desde el puerto serial como se puede visualizar en Figura 3.35.

Dependiendo del dato de entrada se escriben valores lógicos digitales en los pines de salida de la tarjeta Arduino UNO para controlar el encendido y apagado de los dispositivos de la vivienda. Por otra parte, se requiere que los pines sean declarados salidas.

```
void loop() {
     //Verifica que este abierto el puerto serial de comunicacion
     if(Serial.available()){
        recibe=Serial.read();  //Lee lo que llega del puerto Serial

     //Lee cada una de los datos para encender el equipo correspondiente
     if(recibe=='A'){digitalWrite(2,HIGH);}
     if(recibe=='B'){digitalWrite(2,LOW);}
     }}
```

Figura 3.35 Función "*Serial.available()".*

El diagrama de flujo correspondiente al programa de la placa Arduino UNO se muestra en el Anexo 3. Además, en la *URL* https://bit.ly/3F3BmWX se encuentra el código completo y comentado.

3.4 Implementación del sistema

La estructura de la maqueta se encuentra construida en madera MDF de 12 (mm). Para determinar las dimensiones de la maqueta se tomó en cuenta el tamaño de los dispositivos empleados (Focos de 120W, servomotor, calefactor, alarma, ventiladores), considerando que presenten el espacio necesario. Además, se determinó un área de

1.5 (m^2), con las siguientes dimensiones: 1.5 (m) de largo, 1 (m) de ancho. Como se aprecia en la Figura 3.36 y Figura 3.37 se muestra el acabado de la maqueta.

Figura 3.36 Planta Baja - Maquetado

Figura 3.37 Planta Alta - Maquetado

En el Anexo 2 láminas 1 y 2 se pueden observar los planos correspondientes para la implementación de la maqueta.

A continuación en la Figura 3.38 se visualiza el esquema del sistema que se implementó para controlar los 16 dispositivos del hogar mediante comandos de voz, así como el voltaje que manejaran cada uno de ellos.

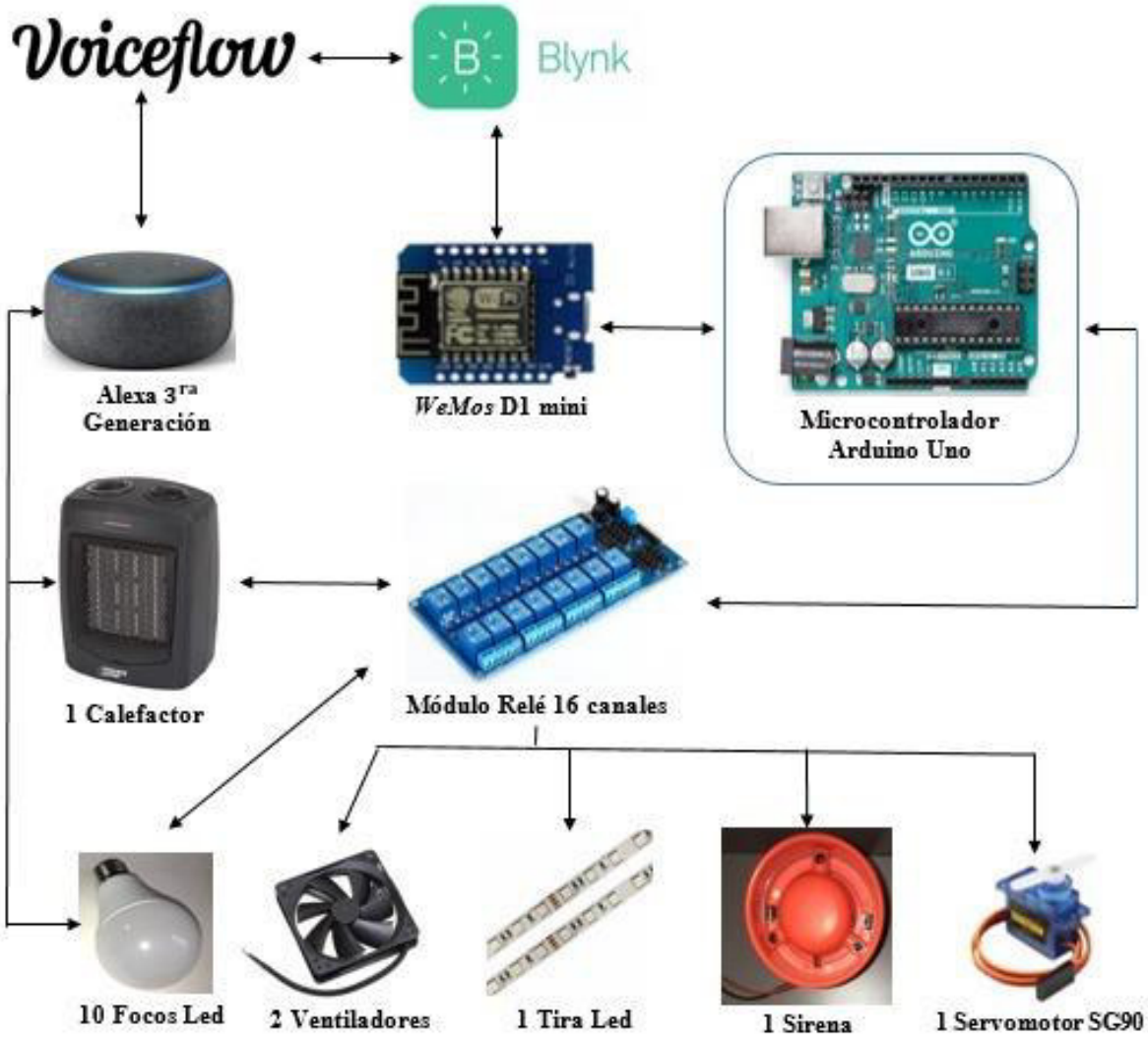


Figura 3.38 Esquema para el control de los dispositivos del hogar.

Ubicación de elementos

Se realizó un diagrama de los dispositivos que se controlarán dentro de los planos de la vivienda para que la implementación del sistema sea sencilla como se observa en la Figura 3.39 y Figura 3.40, además de su simbología en la Tabla 3.3.

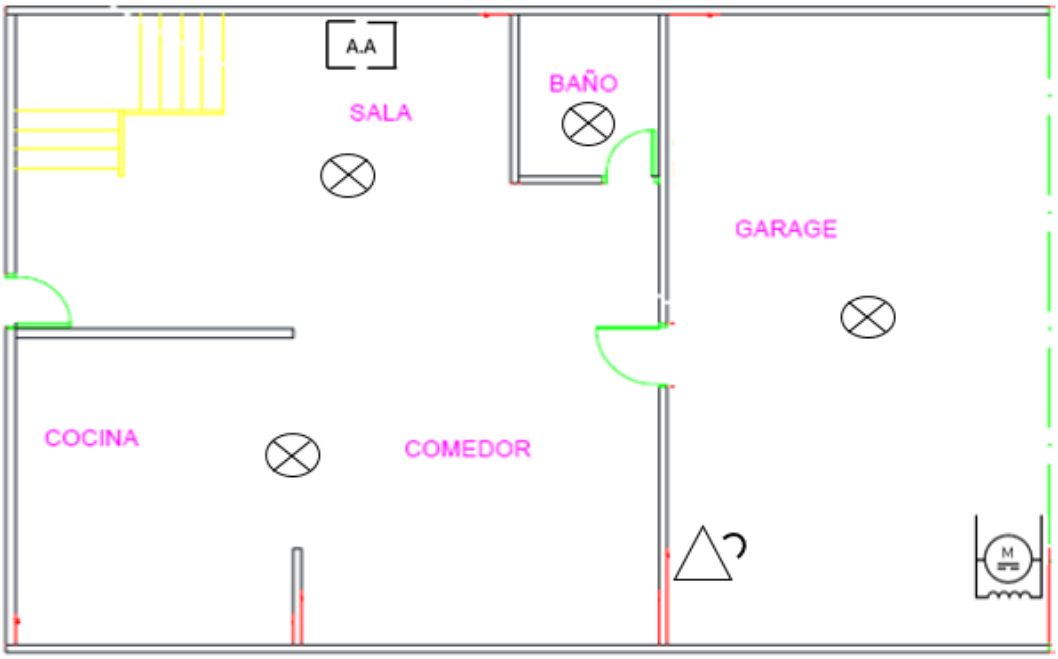


Figura 3.39 Ubicación de dispositivos – Planta Baja

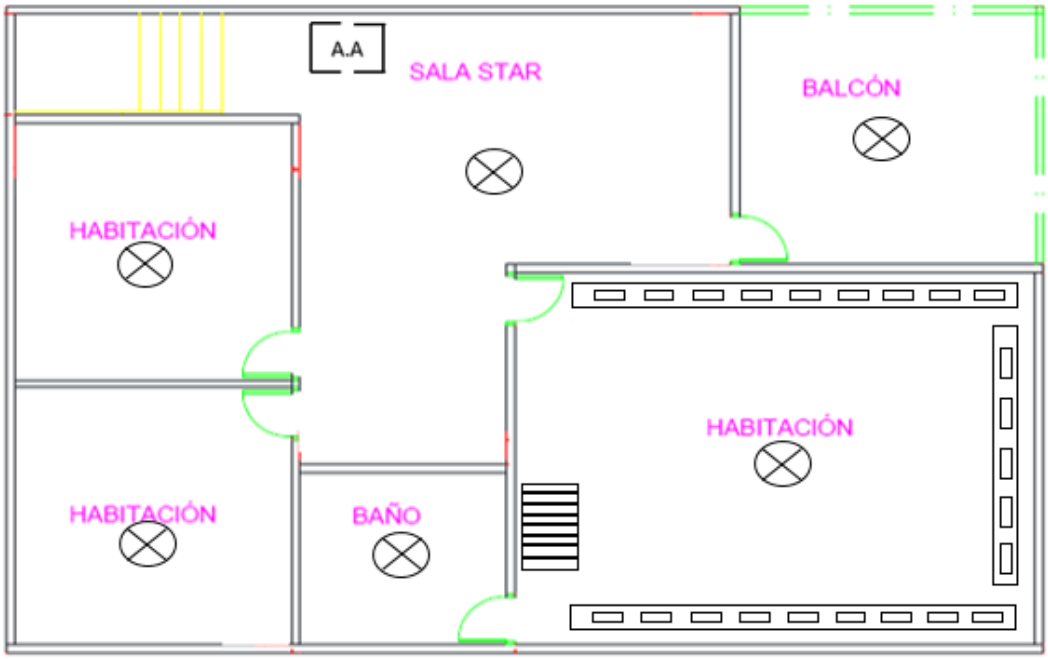


Figura 3.40 Ubicación de dispositivos – Planta Alta

Tabla 3.3 Simbología de los dispositivos

Símbolo	Descripción
	Bombilla
A.A	Aire Acondicionado
	Alarma
M	Servomotor
	Calefactor
	Tiras Led

Diseño de los circuitos electrónicos

Se utilizó el programa Proteus para la elaboración de los diagramas esquemáticos que permiten establecer las conexiones entre los dispositivos electrónicos. En la Figura 3.41 y Figura 3.43 se muestran los diagramas esquemáticos junto con los enlaces que se encuentran identificados con un número y detallados tanto para el circuito de control como para el módulo relé.

Circuito de control

- **1:** Este enlace muestra la salida de voltaje de 5 (V_{DC}) desde el Arduino UNO que permite la alimentación del módulo relé.
- **2:** Permite el envío de datos desde el *WeMos* D1 mini hasta el Arduino UNO, a través del puerto serial, se utiliza un DIP switch para controlar la conexión y desconexión entre los microcontroladores.
  - La conexión del pin RX del Arduino Uno debe ir conectado al pin TX del *WeMos* D1 mini a través del DIP switch.
  - La conexión del pin TX del Arduino UNO debe ir conectado al pin RX del *WeMos* D1 mini por medio del DIP switch.
- **3:** Permite las conexiones de las salidas digitales del Arduino UNO hacia el módulo relé por medio de borneras.

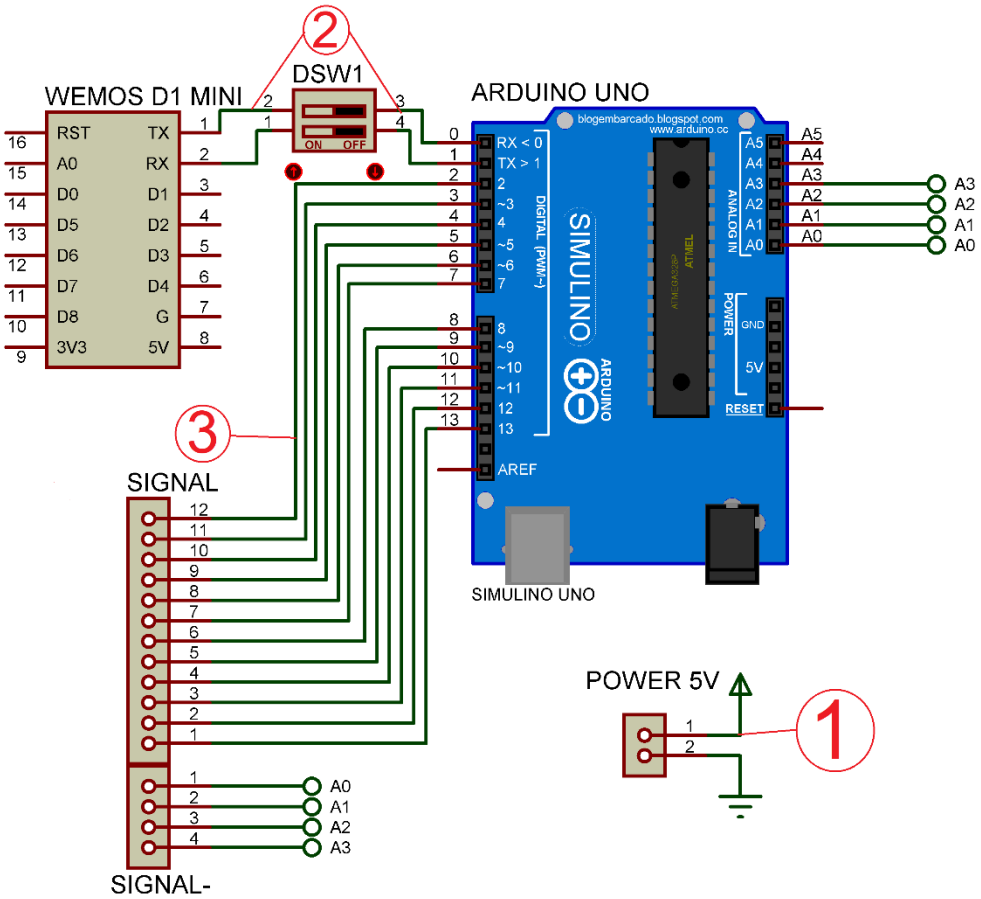


Figura 3.41 Diagrama esquemático (placa de control).

Una vez terminado el diseño del diagrama esquemático. Este se exportó hacia Ares, en donde se configuró la disposición final de los componentes del circuito, el ancho de las pistas de conexión y se estableció el tamaño de la PCB. El diseño del circuito en Ares se puede visualizar en la Figura 3.42 .

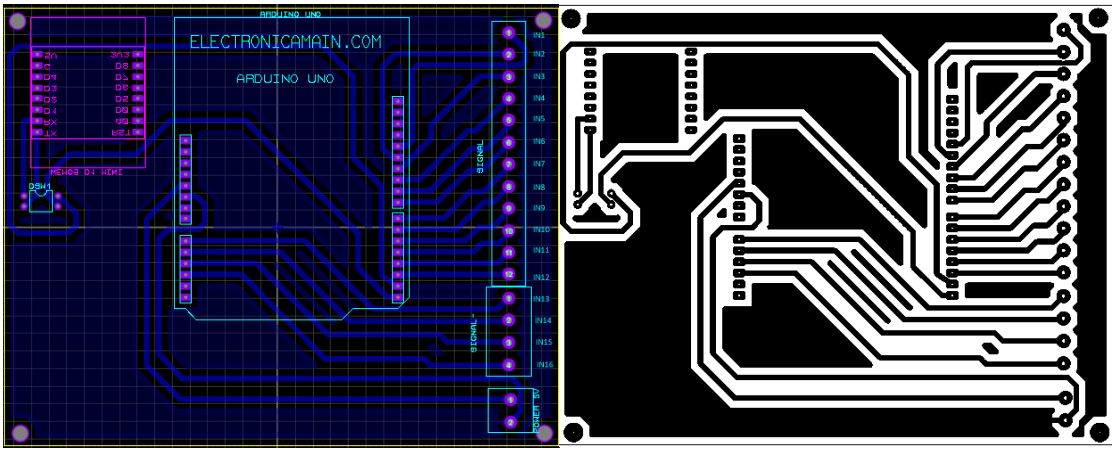


Figura 3.42 Diseño de la PCB en Ares para el circuito de control.

Módulo relé

La placa del módulo relé está construida para el control de componentes de alto voltaje o amperaje, las cuales no pueden ser manejadas directamente con la placa Arduino. Para esto se utilizó relés, que son dispositivos con los cuales se pueden controlar cargas de alto voltaje a través de una señal pequeña proveniente de la placa Arduino. En la Figura 3.43 se muestra el diagrama esquemático del módulo relé para 8 canales.

En el Anexo 2 láminas 5 y 6 se muestran los diagramas completos para el módulo relé de 16 canales.

- **1:** Este enlace conecta con el extremo de la alimentación del Arduino UNO por medio de borneras.
- **2:** Es el otro extremo de las conexiones de los pines del Arduino UNO y el módulo relé que se realizan a través de borneras.
- **3:** Este enlace permite el control de los canales del módulo relé, dependen de la señal proveniente del Arduino UNO.
- **4:** Se encarga de conectar al módulo relé con los dispositivos que serán controlados en la vivienda.

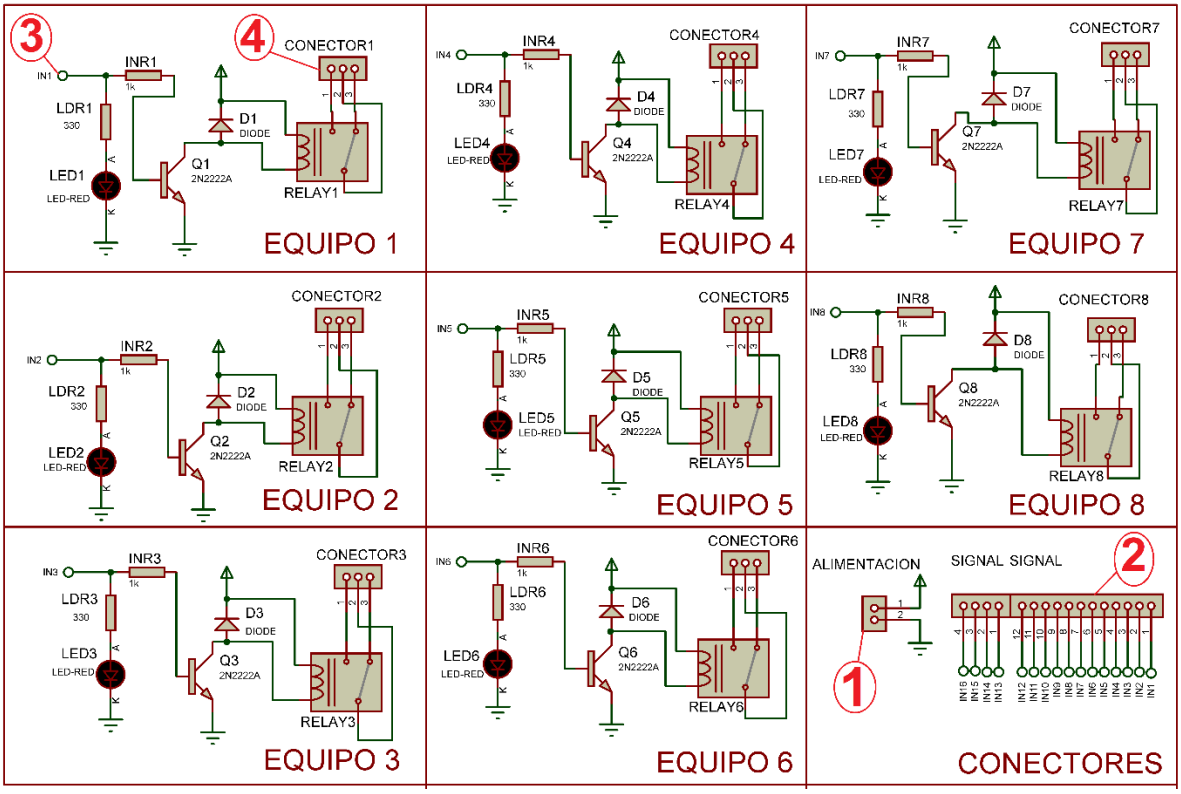


Figura 3.43 Diagrama esquemático (módulo relé).

Después de terminar el diseño del diagrama esquemático del módulo relé. Este se exportó hacia Ares, en donde se configuró la ubicación de los dispositivos electrónicos. Además, se identificaron los elementos que serán manipulados con mayor frecuencia para ser situados en los bordes de la placa como se puede visualizar en la Figura 3.44.

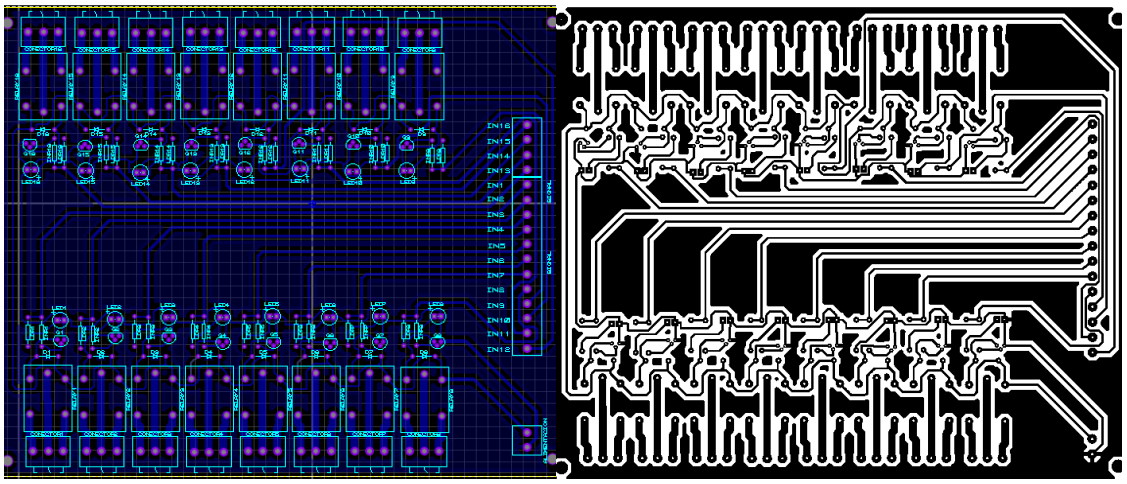


Figura 3.44 Diseño de la PCB en Ares para el módulo relé.

Fabricación de los circuitos impresos

Cuando los diseños se encuentran completos se deben realizar los PCBs a través del método de transferencia térmica del tóner o más conocido como el método de planchado. En la Figura 3.45 y Figura 3.46 se pueden observar la construcción de los circuitos impresos para el circuito de control y el módulo relé. Una vez los diseños están listos, se procedió a transferir los circuitos impresos hacia la baquelita utilizando una plancha a vapor de uso doméstico.

Para ello, se procedió a planchar las baquelitas junto con las impresiones de los circuitos, se aplica calor a través de presión al manipular la plancha, la cual permite transferir la tinta de las impresiones hacia el cobre del circuito de control y del módulo relé como se muestran en la Figura 3.45 y Figura 3.46.

Además, se debe tomar en cuenta el tiempo de exposición de la baquelita al calor de la plancha, ya que si se deja por mucho tiempo esta puede deformarse dejándola inservible.

Figura 3.45 Método de planchado de la placa de control.

Figura 3.46 Método de planchado del módulo relé.

Una vez terminado el planchado de las baquelitas, estas deben ser sumergidas en una mezcla de cloruro férrico y agua para realizar el proceso denominado quemado de las PCBs. De esta manera se logra desprender el cobre que no forma parte de las pistas de los circuitos como se muestran en la Figura 3.47 y Figura 3.48.

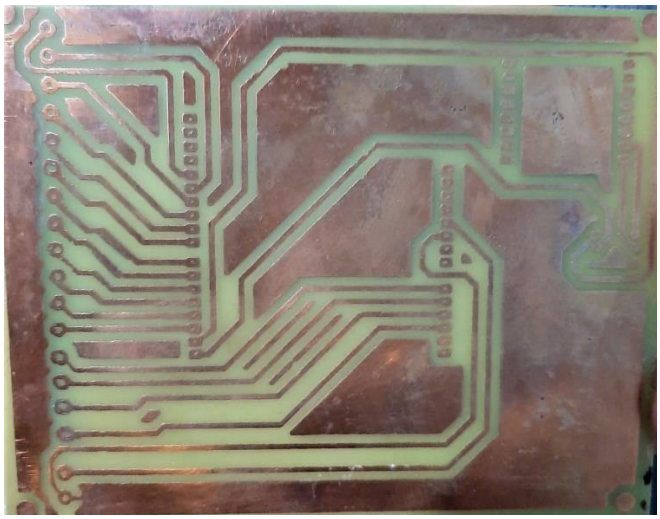


Figura 3.47 PCB para el circuito de control.

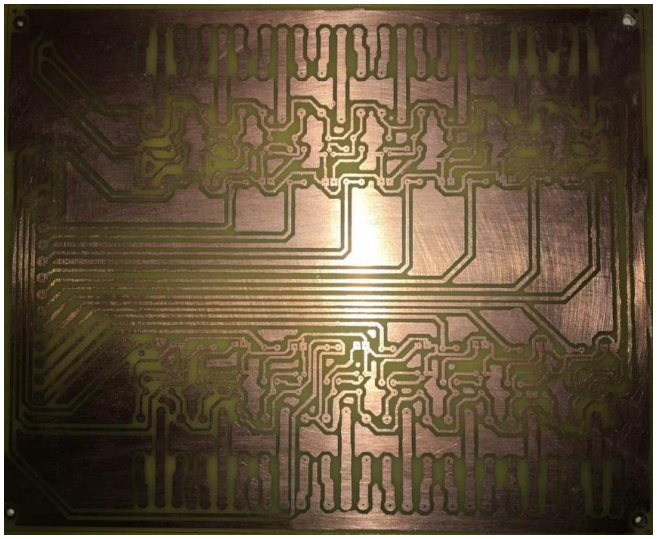


Figura 3.48 PCB para el módulo relé.

Para culminar con la construcción del circuito impreso se deben fijar los componentes eléctrica y mecánicamente a las placas PCBs por medio de soldadura como se pueden visualizar en la Figura 3.49 y Figura 3.50.

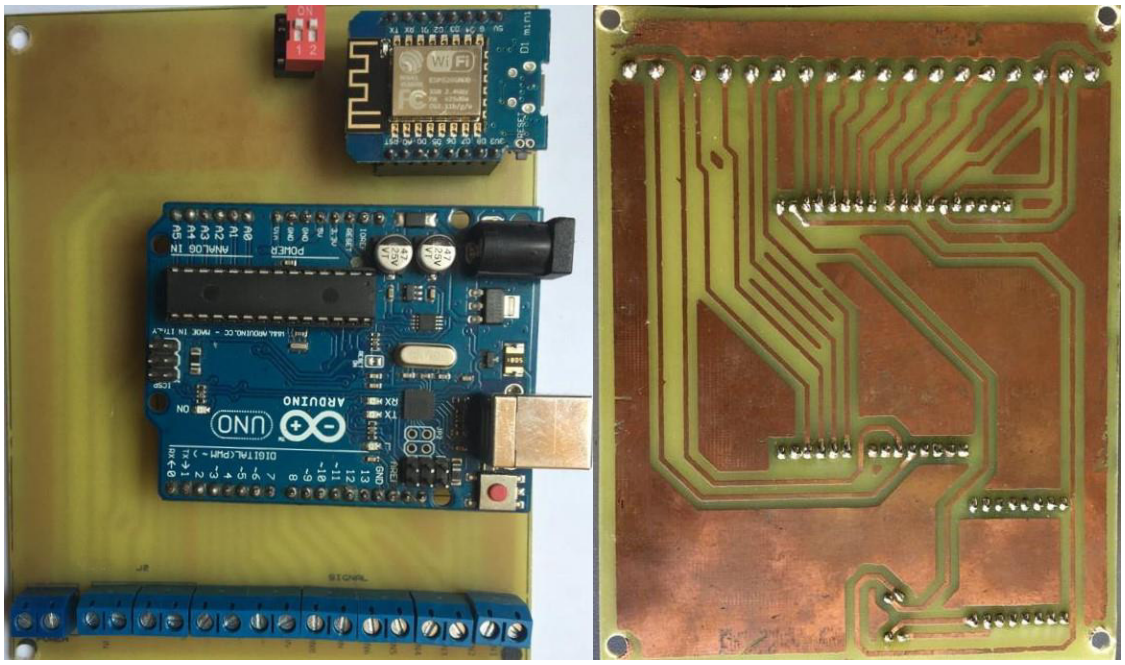


Figura 3.49 Vista frontal y posterior de la placa de control.

En el Anexo 2 láminas 3 y 4 se pueden visualizar el diagrama esquemático y la disposición de los elementos electrónicos del circuito de control.

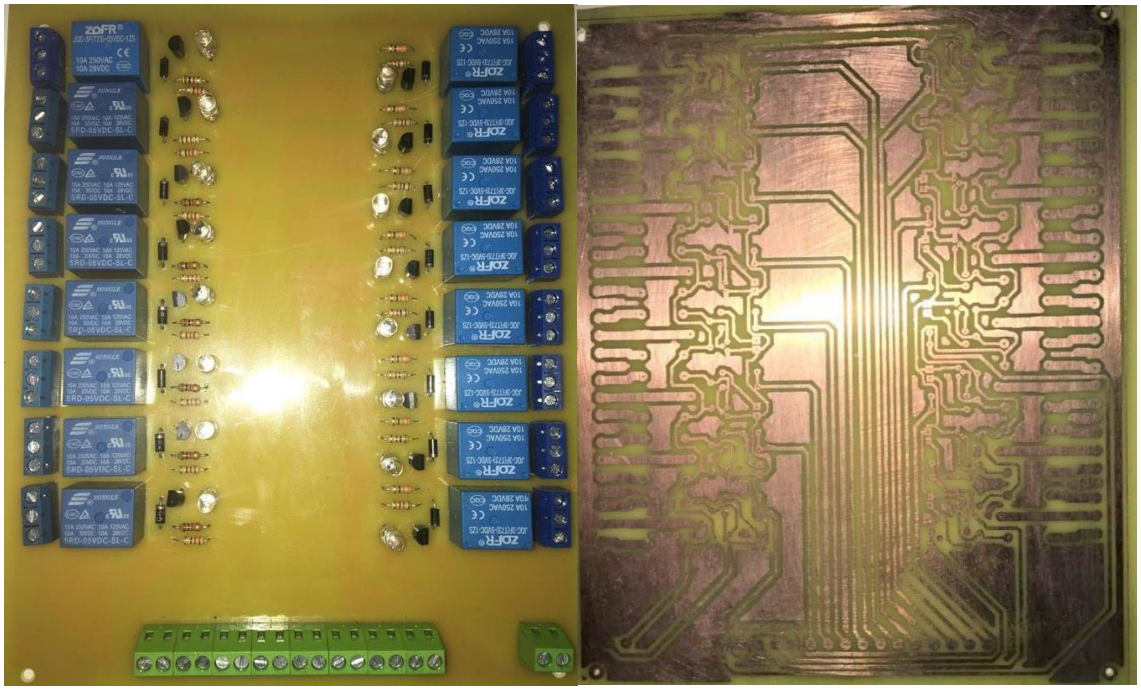


Figura 3.50 Vista frontal y posterior del módulo relé.

En el Anexo 2 láminas 5 y 6 se pueden visualizar el diagrama esquemático y la disposición de los elementos electrónicos del módulo relé.

3.5 Prueba y análisis de resultados

Una vez terminado el desarrollo de los programas y la construcción de la maqueta de la vivienda, se realizaron las respectivas pruebas de funcionamiento.

Prueba del circuito electrónico

La prueba se realizó con la finalidad de comprobar el correcto funcionamiento de la placa que estará en contacto con los dispositivos de automatización de la vivienda.

En la Figura 3.51 se muestra los 8 canales del módulo relé activos, estos sirven para el control domótico de la planta baja de la vivienda. En la Figura 3.52 se muestran los canales restantes del módulo relé, que se encargan del control domótico de la planta alta de la vivienda. Esto es posible gracias al correcto funcionamiento de los canales del módulo relé implementado.

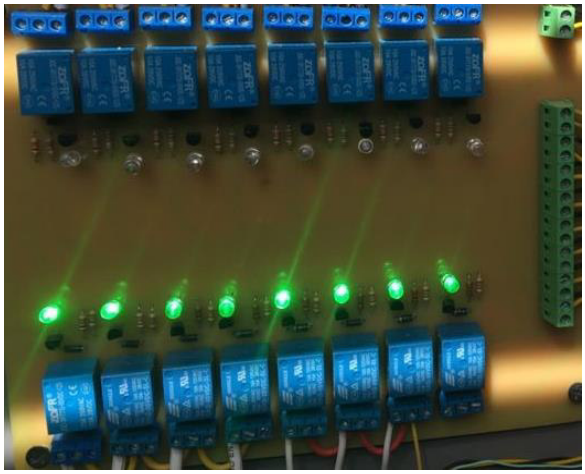


Figura 3.51 Módulo relé 16 canales planta baja.

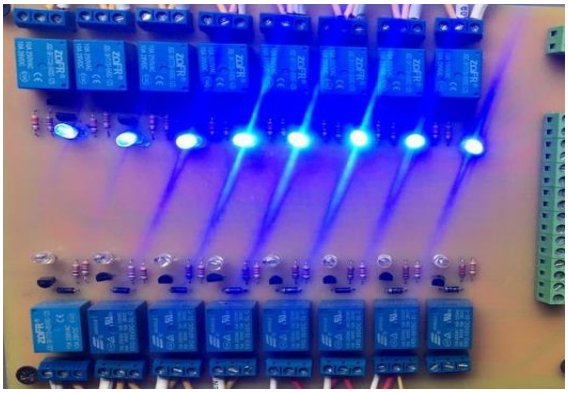


Figura 3.52 Módulo relé 16 canales planta alta.

En Figura 3.53 se muestra el módulo relé implementado el cual es capaz de controlar los dispositivos de la planta domótica. Para realizar la prueba de funcionamiento se energizó con 5 (V_{DC}) a cada una de las entradas del módulo relé, para confirmar su correcto funcionamiento.

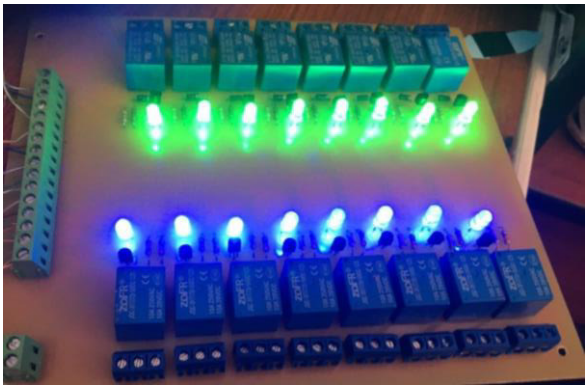


Figura 3.53 Módulo relé 16 canales.

Pruebas de comunicación

Se realizó la prueba de comunicación entre la tarjeta *WeMos* D1 mini y Arduino UNO para verificar si los datos están siendo enviados y recibidos correctamente. Además, dicha prueba se realizó para verificar que el desarrollo de los programas sean los correctos.

En la Figura 3.54 se puede visualizar la comunicación que existe entre la tarjeta Arduino UNO y *WeMos* D1 mini a través de sus puertos seriales. Cuando se ejecuta un comando de voz la tarjeta *WeMos* D1 mini (COM7) envía un carácter hacia el Arduino UNO (COM3), este recibe el carácter y ejecuta las instrucciones programadas para activar las salidas digitales y así controlar los dispositivos de la planta baja de la vivienda a través de la placa del módulo relé.

En la Figura 3.55 se muestra la comunicación entre la tarjeta *WeMos* D1 mini y Arduino UNO para controlar los dispositivos de la planta alta.

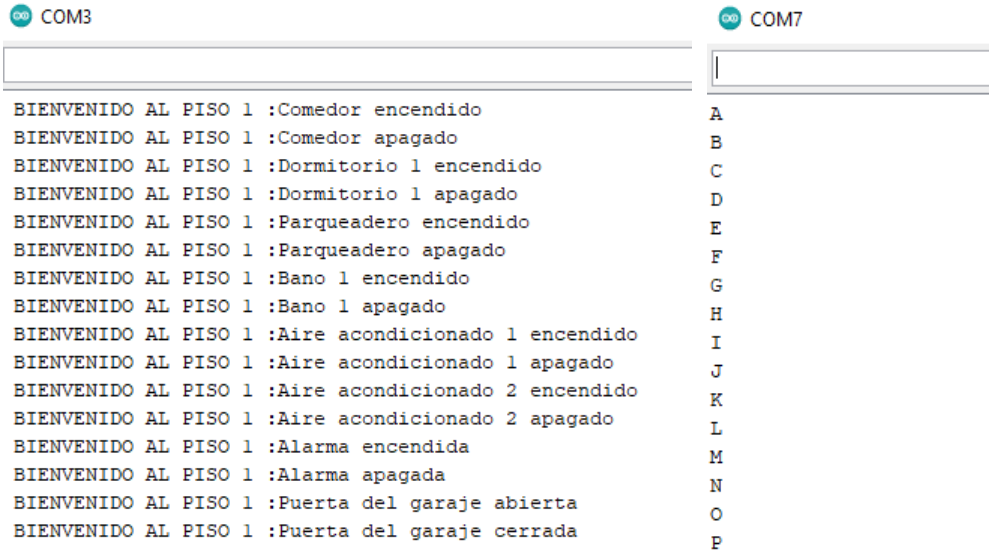


Figura 3.54 Comunicación *WeMos* y Arduino UNO planta baja.

```
COM3
BIENVENIDO AL PISO 2 :Balcon encendido
BIENVENIDO AL PISO 2 :Balcon apagado
BIENVENIDO AL PISO 2 :Dormitorio 3 encendido
BIENVENIDO AL PISO 2 :Dormitorio 3 apagado
BIENVENIDO AL PISO 2 :Sala encendida
BIENVENIDO AL PISO 2 :Sala apagada
BIENVENIDO AL PISO 2 :Luz ambiental encendida
BIENVENIDO AL PISO 2 :Luz ambiental apagada
BIENVENIDO AL PISO 2 :Calefacción encendida
BIENVENIDO AL PISO 2 :Calefacción apagada
BIENVENIDO AL PISO 2 :Dormitorio 4 encendido
BIENVENIDO AL PISO 2 :Dormitorio 4 apagado
BIENVENIDO AL PISO 2 :Bano 2 encendido
BIENVENIDO AL PISO 2 :Bano 2 apagado
BIENVENIDO AL PISO 2 :Dormitorio 5 encendido
BIENVENIDO AL PISO 2 :Dormitorio 5 apagado
```

```
COM7
Q
R
S
T
U
V
W
X
Y
Z
a
b
c
d
e
f
```

Figura 3.55 Comunicación *WeMos* y Arduino UNO planta alta.

Para finalizar. En la Figura 3.56 se muestra el funcionamiento de la parte de control con Arduino UNO y *WeMos* D1 mini junto con la parte de potencia con el módulo relé de 16 canales completamente funcional.

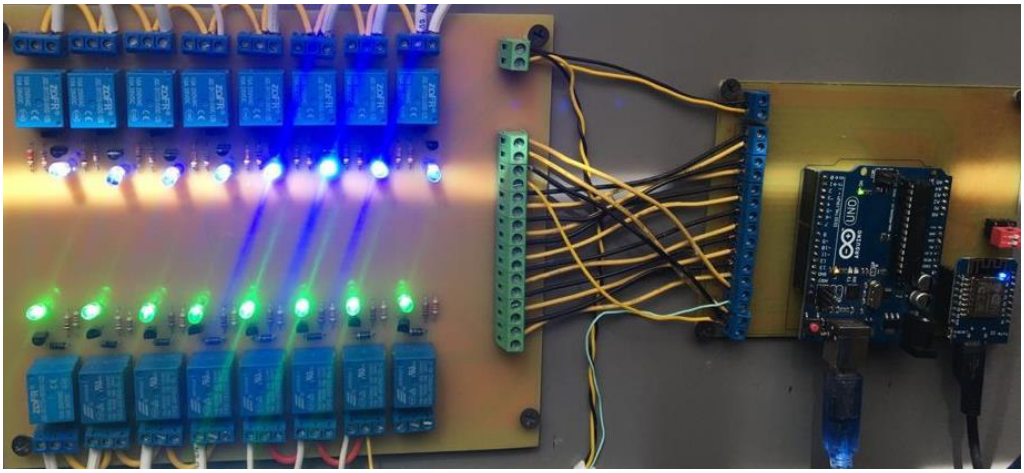


Figura 3.56 Sistema de control para la planta domótica.

Prueba de funcionamiento del sistema

Para la segunda parte de las pruebas de funcionamiento se verificó que el sistema de control cumpla con las condiciones establecidas para la planta domótica. Como primer paso se verificó el acceso al control de la planta baja y planta alta, en el cual se deben activar las *Skills* ya establecidas para acceder al control de los sistemas de iluminación, ventilación, seguridad y calefacción de la vivienda.

En la Figura 3.57 se muestra el accionamiento de la puerta de garaje de la planta baja, la cual se puede realizar por comandos de voz y mediante una interfaz *web* y móvil.

Los comandos de voz para acceder al control de la planta baja de la vivienda se encuentran en el Anexo 4 Tabla 5.1 y Tabla 5.2

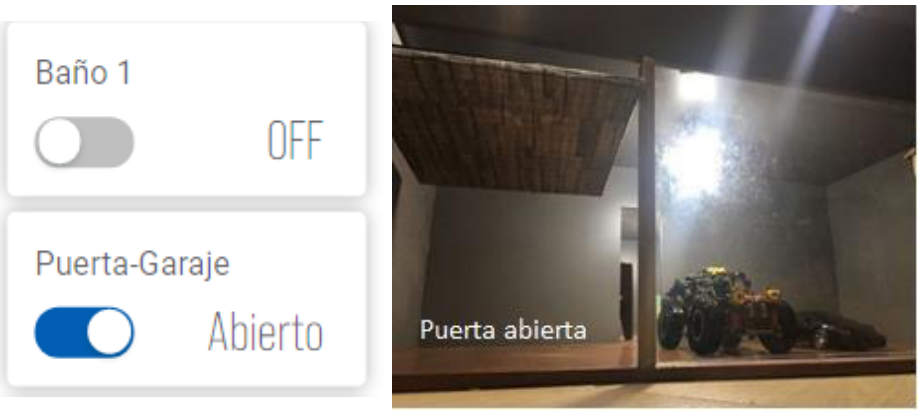


Figura 3.57 Puerta de garaje abierta.

En la Figura 3.58 se muestra el control del sistema de iluminación de la planta baja, que puede ser activado mediante comandos de voz y a través de una interfaz *web* y móvil.

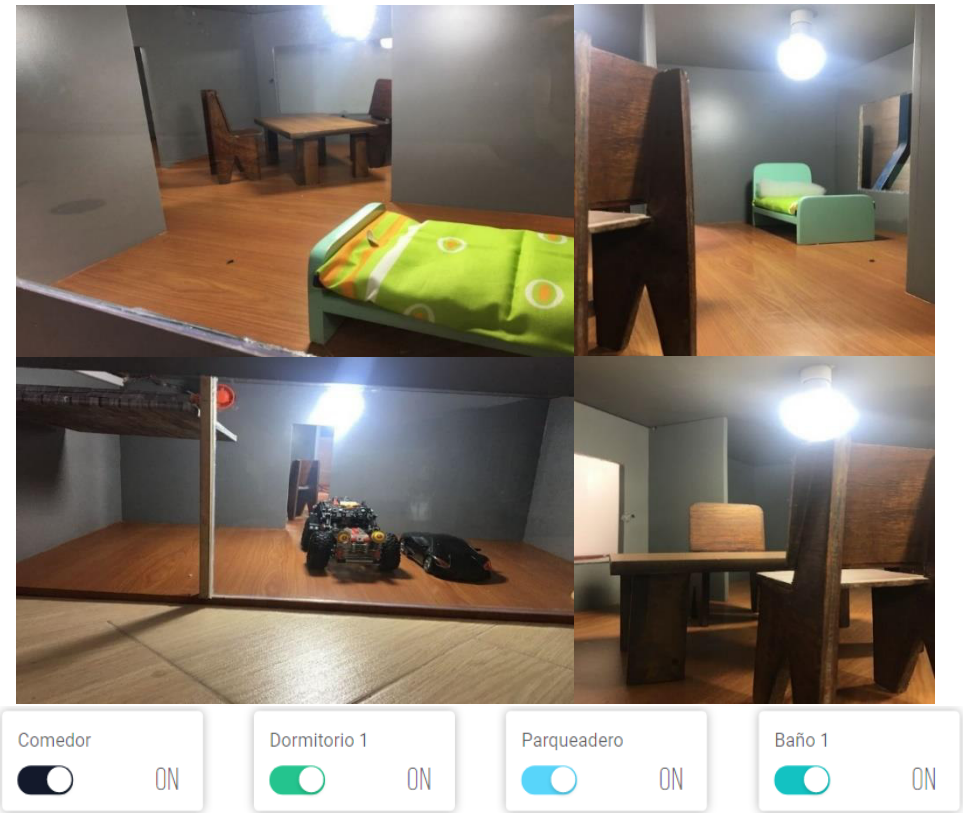


Figura 3.58 Control del sistema de iluminación de la planta baja.

En la Figura 3.59 se muestra el control por medio de la interfaz móvil del sistema de ventilación de la vivienda.

Figura 3.59 Control del sistema de ventilación.

En la Figura 3.60 se muestra el accionamiento del sistema de iluminación de la planta alta de la vivienda.

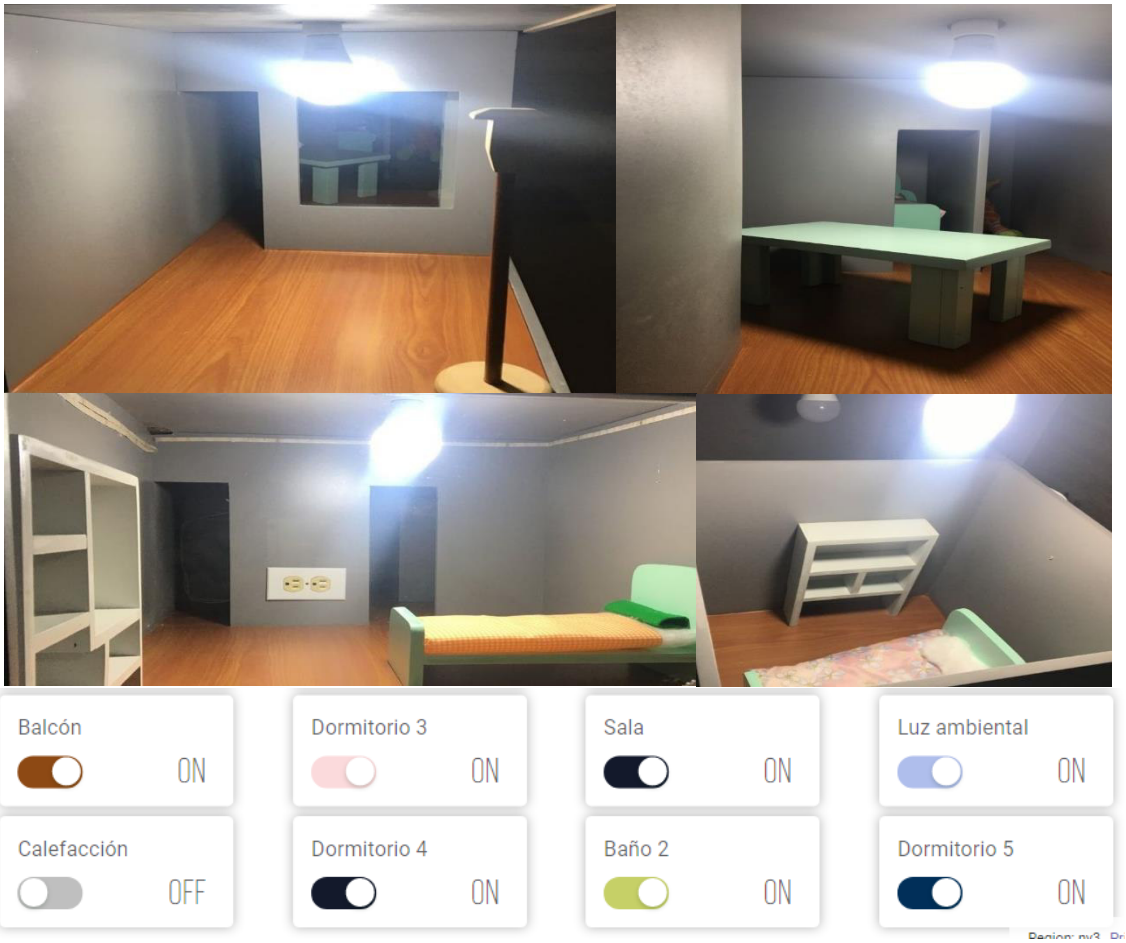


Figura 3.60 Sistema de iluminación planta alta de la vivienda.

El control del sistema de calefacción puede ser activado a través de comandos de voz, interfaz *web* y móvil. En la Figura 3.61 se muestra la interfaz móvil para activar el sistema de calefacción.

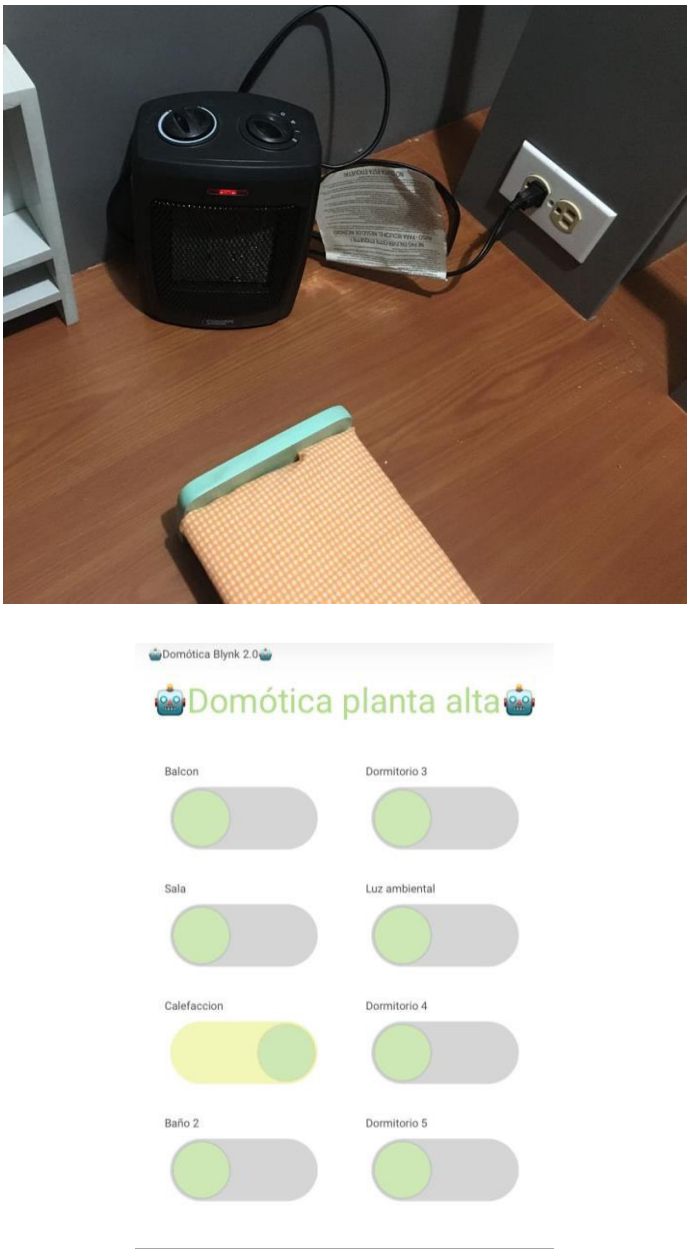


Figura 3.61 Sistema de calefacción.

Los comandos de voz para acceder al control de la planta alta de la vivienda se muestran en el Anexo 4 Tabla 5.1 y Tabla 5.3.

En la Tabla 3.4 y Tabla 3.5 se muestra los resultados de las pruebas realizadas a la planta baja y planta alta de la vivienda, que establecen su correcto funcionamiento.

Tabla 3.4 Resultados de las pruebas de los comandos de voz.

Función por comandos de voz	Funcionamiento correcto	Funcionamiento incorrecto
Activación de sirena	✓	
Activación de la puerta	✓	
Iluminación	✓	
Ventilación	✓	
Calefacción	✓	

Tabla 3.5 Resultados de las pruebas de la interfaz *web* y móvil

Función por interfaz *web* y móvil	Funcionamiento correcto	Funcionamiento incorrecto
Activación de sirena	✓	
Activación de la puerta	✓	
Iluminación	✓	
Ventilación	✓	
Calefacción	✓	

3.6 Manual de Uso y Mantenimiento

Para mostrar el manejo y mantenimiento del sistema domótico controlado por comandos de voz se realizaron dos videos explicativos. Se puede acceder a los videos mediante los códigos QR de la Figura 3.62 y Figura 3.63.

URL manual de uso: https://www.youtube.com/watch?v=GgeALtaOlqo

Figura 3.62 Código QR para el video del manual de uso.

URL manual de mantenimiento: https://www.youtube.com/watch?v=v0aGU87HddY

Figura 3.63 Código QR para el video del manual de mantenimiento.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El manejo del programa de diseño “AutoCAD”, para el modelado de una vivienda a escala resultó de gran ayuda, ya que permitió diseñar las habitaciones con el espacio adecuado para distribuir y utilizar los diferentes dispositivos de manera óptima y eficaz en el presente proyecto.
- Para determinar los requerimientos del prototipo se tomó en cuenta que se necesitaban dos sistemas, uno de control el cual es el encargado de administrar los datos tanto de recepción como de transmisión hacia la plataforma *Blynk* y otro sistema de potencia que se encarga de distribuir el suficiente voltaje a los dispositivos electrónicos para que funcionen de manera óptima.
- Para que el sistema domótico pueda reconocer los comandos de voz y ejecutar las acciones se utilizó la plataforma *Voiceflow*, a través de la cual se crearon dos *Skills* de Alexa mediante programación de bloques para acceder al control de los dispositivos distribuidos tanto en la planta baja como en la planta alta de la vivienda.
- Al realizar el estudio comparativo entre los diferentes modelos de Arduino, en términos de memoria flash para el almacenamiento del código, se determinó que el Arduino Leonardo posee una memoria igual al Arduino UNO de 32KB con la diferencia que 4KB son utilizados para la bootloader (gestor de arranque), el modelo Mega2560 tiene 256KB, lo cual es una cantidad exagerada de memoria para el presente proyecto. Por este motivo se seleccionó Arduino UNO ya que su memoria flash es suficiente para el óptimo desarrollo del proyecto.
- Para la creación de la interfaz humano-máquina se utilizó la interfaz de navegador *web Blynk*, ya que facilitó la realización de las plantillas que contienen todos los dispositivos que serán controlados por el usuario de una manera fácil y sencilla. Además, permite la comunicación entre dos aplicaciones a través de su *API.*
- Mediante el programa Ares (subprograma de Proteus), se diseñaron los diagramas esquemáticos de las dos placas PCBs y la distribución de los componentes que contendrán cada una de ellas. A través de dichos diagramas se generan pistas para interconectar un componente con otro evitando así la utilización de cables, esto es un punto favorable ya que al existir demasiadas

conexiones cableadas pueden generar interferencias y falsos contactos eléctricos en el sistema domótico produciendo fallas a la hora de su ejecución.

- Para verificar el correcto funcionamiento del sistema se comprobó que al ejecutar un comando de voz la tarjeta *WeMos* D1 mini envía un carácter al Arduino UNO estableciendo conexión, este a su vez lo recepta y ejecuta la instrucción previamente programada, lo cual permite el control de todos los dispositivos de la vivienda.
- Es importante recalcar que, si el usuario no ingresa el identificador correcto de las *Skills* de Alexa previamente definidas en la plataforma *Voiceflow*, el sistema no reconocerá el comando de voz y por ende los dispositivos no podrán encenderse o apagarse.

4.2 Recomendaciones

- Al momento de utilizar la placa Arduino UNO y *WeMos* D1 mini, se debe considerar los datos técnicos de los componentes que forman parte del sistema domótico. Esto con el fin de utilizar los niveles apropiados de tensión y corriente en la fuente de alimentación.
- Es importante considerar las conexiones eléctricas, los niveles de tensión, los terminales de las PCBs y los equipos que operan en la vivienda para garantizar un funcionamiento adecuado.
- Antes de controlar los dispositivos de la planta baja y planta alta de la vivienda, se recomienda una familiarización de los comandos de voz, que se muestran en la Tabla 5.1, Tabla 5.2 y Tabla 5.3 del Anexo 4.
- Al momento de configurar los botones para el control de los dispositivos de la vivienda en la interfaz *web*, se debe considerar la identificación que poseen estos, la distribución del espacio del panel para evitar acumulación visual que dificulte la manipulación de las herramientas dinámicas de control.
- Para la implementación del presente proyecto en una vivienda funcional, es recomendable utilizar la librería “SoftwareSerial” para realizar la comunicación UART utilizando pines distintos al TX y RX de Arduino, ya que estos se emplean para una comunicación mediante cable USB. Además, en ausencia de una conexión a Internet, se debe implementar conmutadores en paralelo a los relés para controlar manualmente los dispositivos.
- Para un correcto funcionamiento es recomendable realizar mantenimientos preventivos a las placas PCBs, ajustando las tuercas de las borneras, verificando

el funcionamiento de los canales del módulo relé y comprobar la continuidad de los conductores.

- Para futuras modificaciones del sistema domótico, en donde existan un mayor número de dispositivos para el hogar, se recomienda integrar al sistema un microcontrolador con mayor número de entradas y salidas, ya que el Arduino UNO es muy limitado en este sentido.

5 REFERENCIAS BIBLIOGRÁFICAS

[1] Asociación Española de Domótica e Inmótica, «CEDOM,» CEDOM, [En línea]. Available: http://www.cedom.es/sobre-domotica/que-es-domotica. [Último acceso: 09 Agosto 2021].

[2] Amazon, «Alexa,» [En línea]. Available: https://developer.amazon.com/es-ES/alexa. [Último acceso: 09 Agosto 2021].

[3] Amazon, «Alexa,» [En línea]. Available: https://developer.amazon.com/en-US/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html. [Último acceso: 09 Agosto 2021].

[4] Blynk, «Blynk,» GitBook, [En línea]. Available: https://docs.blynk.io/en/. [Último acceso: 09 Agosto 2021].

[5] Blynk, «Blynk,» GitBook, [En línea]. Available: https://docs.blynk.io/en/platform-overview/products. [Último acceso: 09 Agosto 2021].

[6] E. Tolocka, «Diseñando aplicaciones IoT con Blynk,» 07 mayo 2020. [En línea]. Available: https://www.profetolocka.com.ar/2020/05/07/disenando-aplicaciones-iot-con-blynk/. [Último acceso: 09 Agosto 2021].

[7] Arduino.cl, «Arduino.cl,» [En línea]. Available: https://arduino.cl/que-es-arduino/. [Último acceso: 14 Agosto 2021].

[8] Aprendiendo Arduino, «IDE Arduino,» 11 Diciembre 2016. [En línea]. Available: https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/. [Último acceso: 08 Agosto 2021].

[9] Arduino, «Arduino Uno Rev3,» 2021. [En línea]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Último acceso: 15 Agosto 2021].

[10] AV Electronics, «AV Electronics,» [En línea]. Available: https://avelectronics.cc/producto/wemos-d1-mini-esp8266/. [Último acceso: 07 Agosto 2021].

[11] Naylamp Mechatronics, «Naylamp Mechatronics,» 2021. [En línea]. Available: https://naylampmechatronics.com/espressif-esp/291-wemos-d1-mini-esp8266-wifi.html. [Último acceso: 07 Agosto 2021].

[12] DiyI0t, «DiyI0t,» [En línea]. Available: https://diyi0t.com/esp8266-wemos-d1-mini-tutorial/. [Último acceso: 07 Agosto 2021].

[13] Arduino, «Arduino Nano,» 2021. [En línea]. Available: https://store.arduino.cc/usa/arduino-nano. [Último acceso: 15 Agosto 2021].

[14] Arduino, «Arduino Leonardo con encabezados,» 2021. [En línea]. Available: https://store.arduino.cc/usa/leonardo. [Último acceso: 15 Agosto 2021].

[15] Arduino, «Arduino Mega 2560 Rev3,» 2021. [En línea]. Available: https://store.arduino.cc/usa/mega-2560-r3. [Último acceso: 15 Agosto 2021].

[16] Droneymas, «Amazon Echo Dot (3ª Generación),» 2021. [En línea]. Available: https://www.droneymas.com/amazon-echo-dot-3a-generacion/. [Último acceso: 17 octubre 2021].

[17] Naylamp Mechatronics, «Naylamp Mechatronics,» 2021. [En línea]. Available: http://naylampmechatronics.com/. [Último acceso: 25 Agosto 2021].

[18] UNIT ELECTRONICS, «Diodo rectificador 1N4001,» 2021. [En línea]. Available: https://uelectronics.com/producto/diodo-rectificador-1n4001-4001/. [Último acceso: 21 Agosto 2021].

[19] ALLDATASHEET.COM, «2N2222A Datasheet,» 21. [En línea]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/888491/SEMTECH_ELEC/2N2222A.html. [Último acceso: 21 Agosto 2021].

[20] ALLDATASHEET.COM, «SRD-05VDC-SL-C Datasheet,» 2021. [En línea]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/1132639/SONGLERELAY/SRD-05VDC-SL-C.html. [Último acceso: 21 Agosto 2021].

[21] Electrónicos CALDAS, «Leds,» 2021. [En línea]. Available: https://www.electronicoscaldas.com/es/diodos-led/147-led-transparente-5mm-alto-brillo.html. [Último acceso: 21 Agosto 2021].

[22] Electrónica DIY, «Relay 5v y 12v 5 Pines,» 2021. [En línea]. Available: https://www.electronicadiy.com/products/relay-5v-y-12v-5-pines. [Último acceso: 21 Agosto 2021].

[23] PROMETEC, «INTRODUCCION A LA PROGRAMACION DEL ARDUINO,» 2021. [En línea]. Available: https://www.prometec.net/intro-programacion/. [Último acceso: 18 Agosto 2021].

[24] Blynk, «Aprovisionamiento de WiFi,» 2021. [En línea]. Available: https://docs.blynk.io/en/getting-started/activating-devices/blynk-edgent-wifi-provisioning. [Último acceso: 21 Agosto 2021].

[25] electronicscomp, «electronicscomp,» [En línea]. Available: https://www.electronicscomp.com/wemos-d1-mini-iot-develoment-board-india. [Último acceso: 07 08 2021].

[26] Grupo Velasco, «Diodo led,» 2021. [En línea]. Available: http://www.velasco.com.ec/velasco/producto.php?id=1898. [Último acceso: 20 Agosto 2021].

ANEXOS

ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO

ESCUELA POLITECNICA NACIONAL
Campus Politécnico "J. Rubén Orellana R

Quito, 20 de octubre de 2021

CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *Leandro Antonio Pazmiño Ortiz,* docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado su correcto funcionamiento, este proyecto fue implementado por los estudiantes Kevin Clavijo y Kevin Tipanguano.

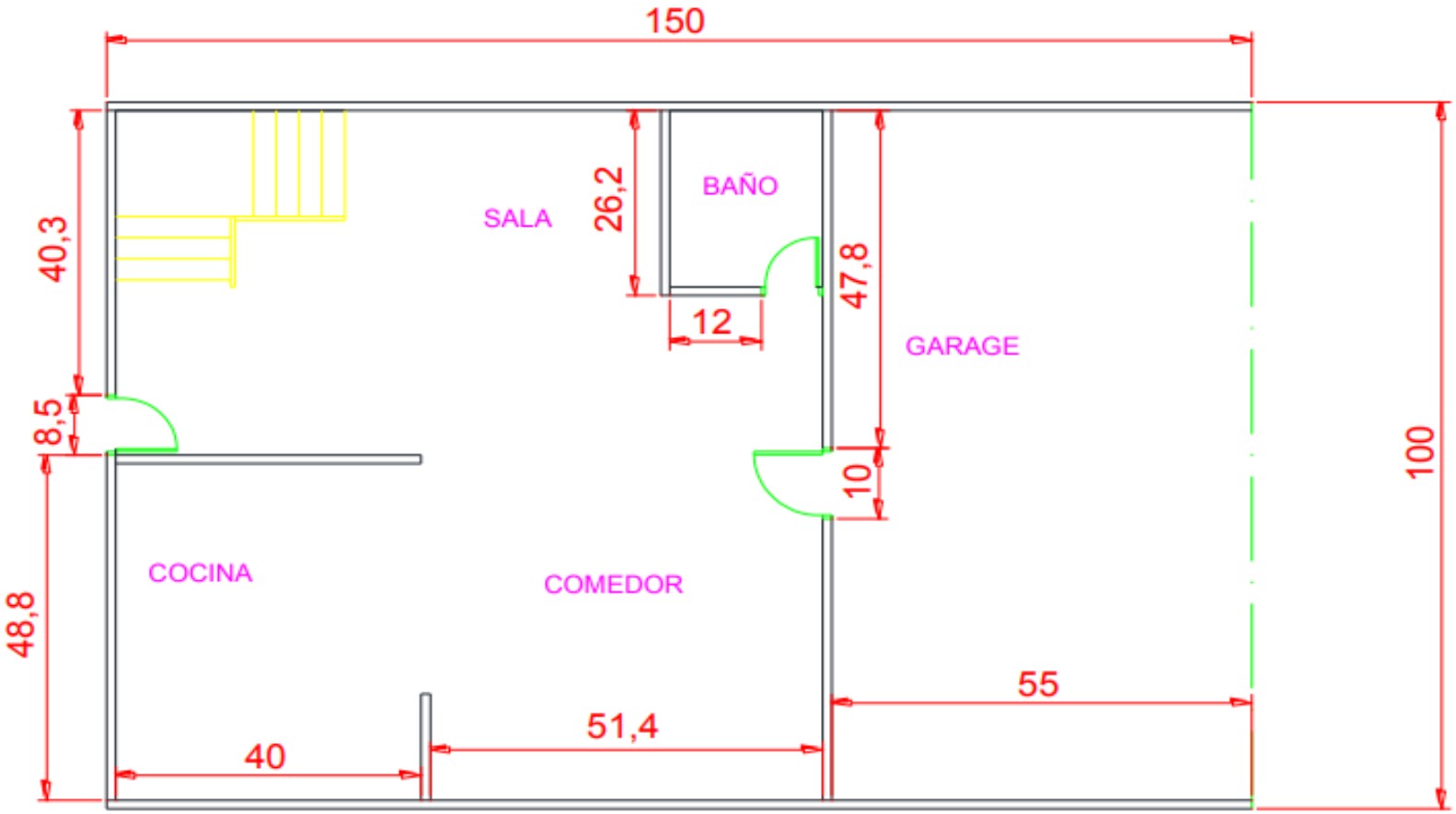
El proyecto cumple con todos los objetivos establecidos en el plan de titulación.

DIRECTOR

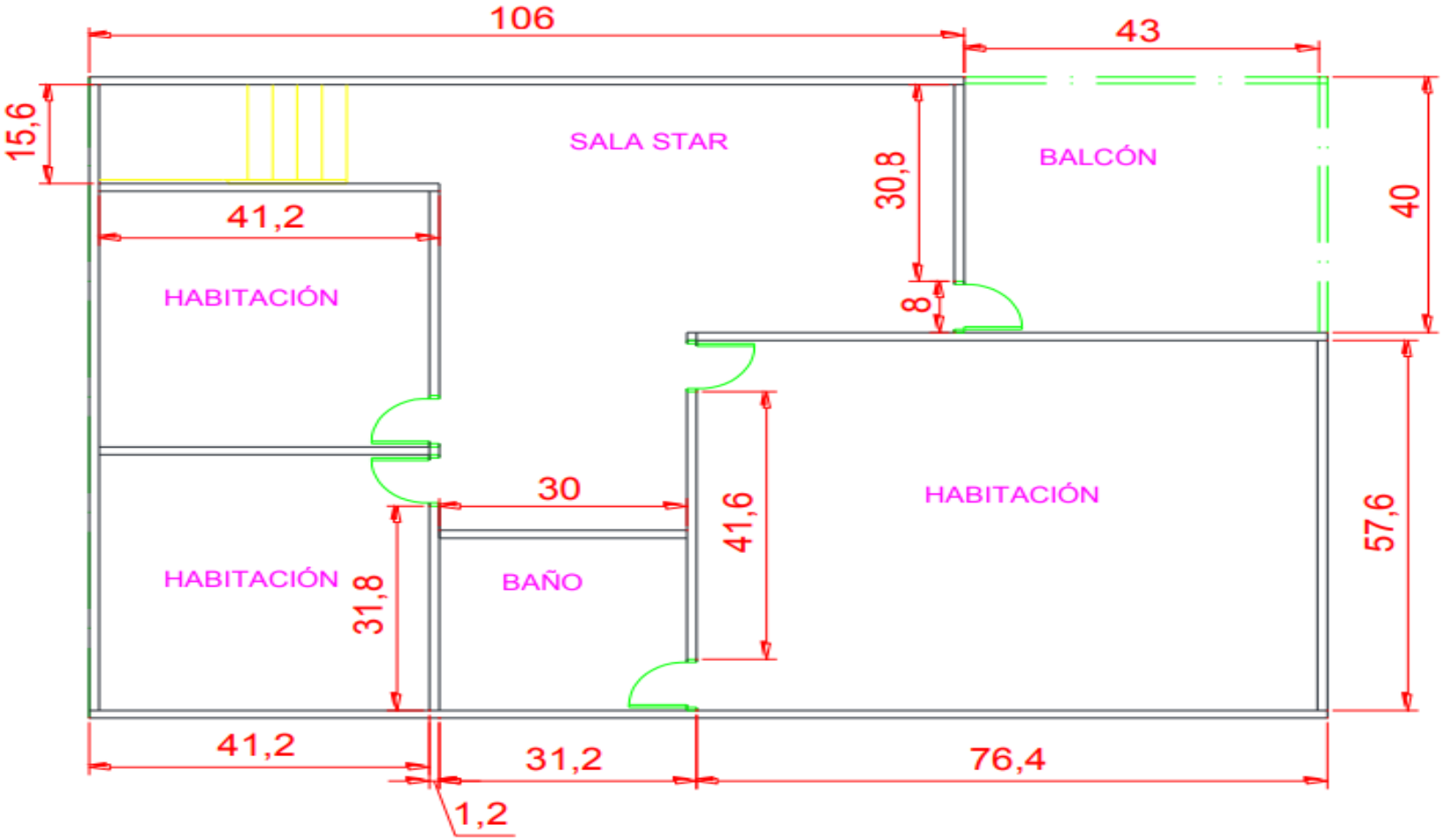
Ing. Leandro Pazmiño Ortiz, MSc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 8. EXT: 2743
email: leandro.pazmino@epn.edu.ec Quito-Ecuador

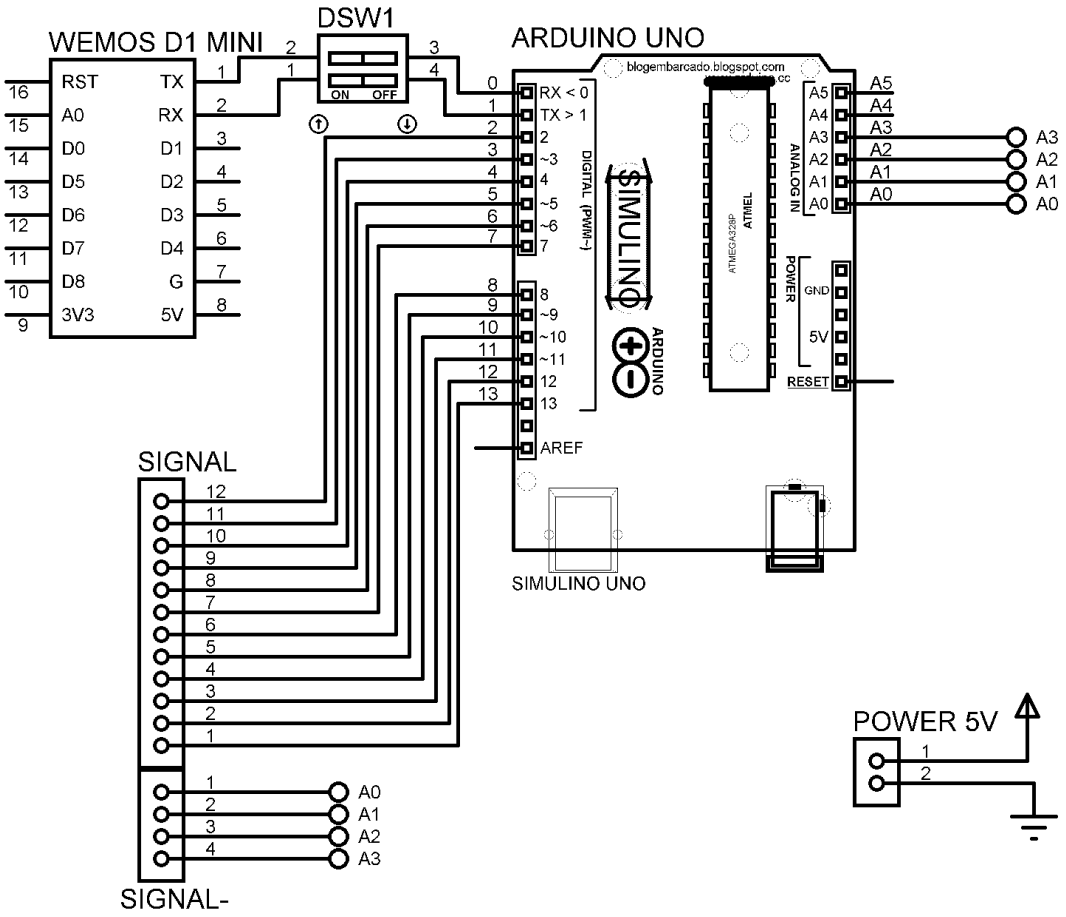
ANEXO 2: PLANOS Y ESQUEMAS



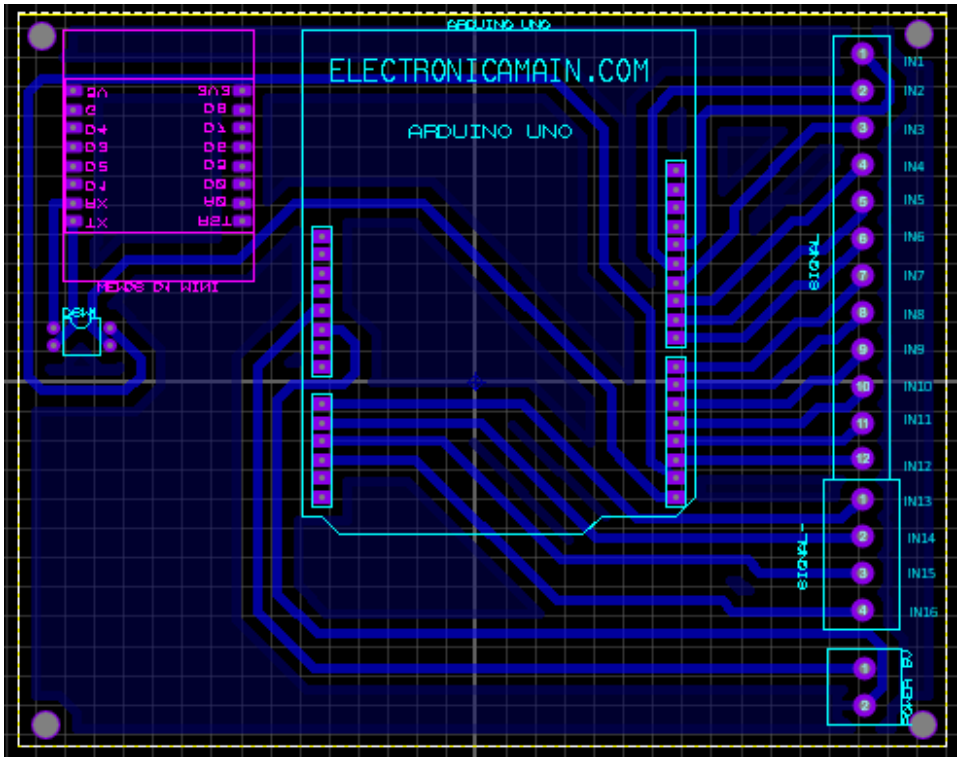
ESCUELA POLITÉCNICA NACIONAL - ESFOT		
Planos Maqueta Planta Baja	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Madera - MDF
	ESCALA: 1:10	**FORMATO:** A4
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 1



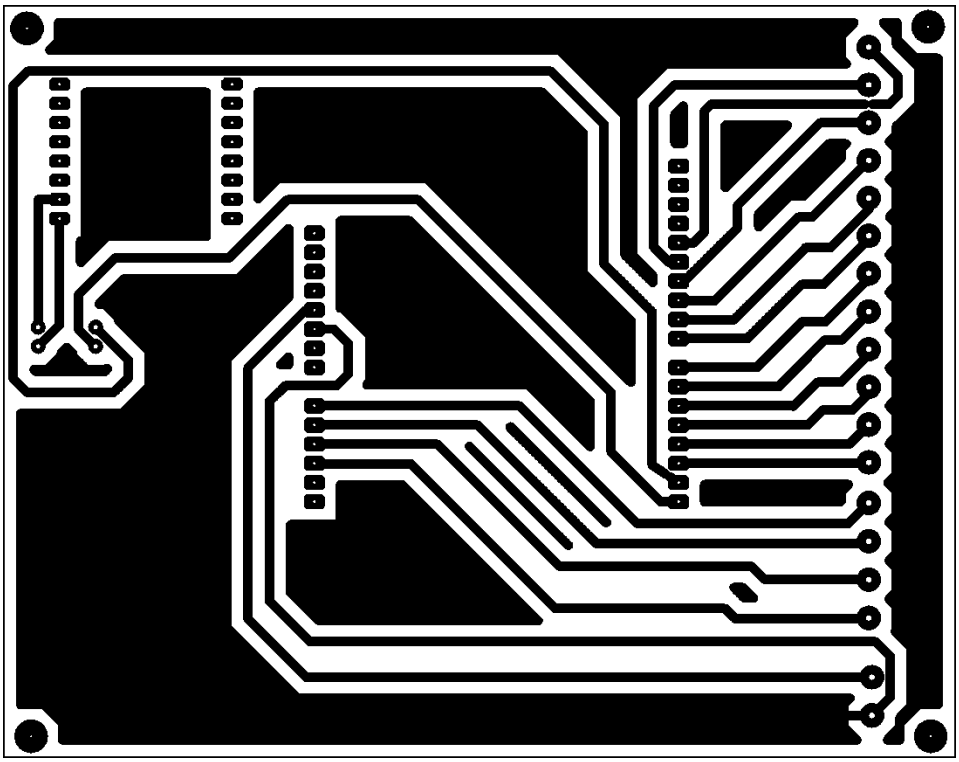
ESCUELA POLITÉCNICA NACIONAL - ESFOT		
Planos Maqueta Planta Alta	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Madera - MDF
	ESCALA: 1:10	**FORMATO:** A4
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 2



ESCUELA POLITÉCNICA NACIONAL - ESFOT		
Diagrama esquemático	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Eléctrico
	ESCALA: 1:1	**FORMATO:** A4
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 3



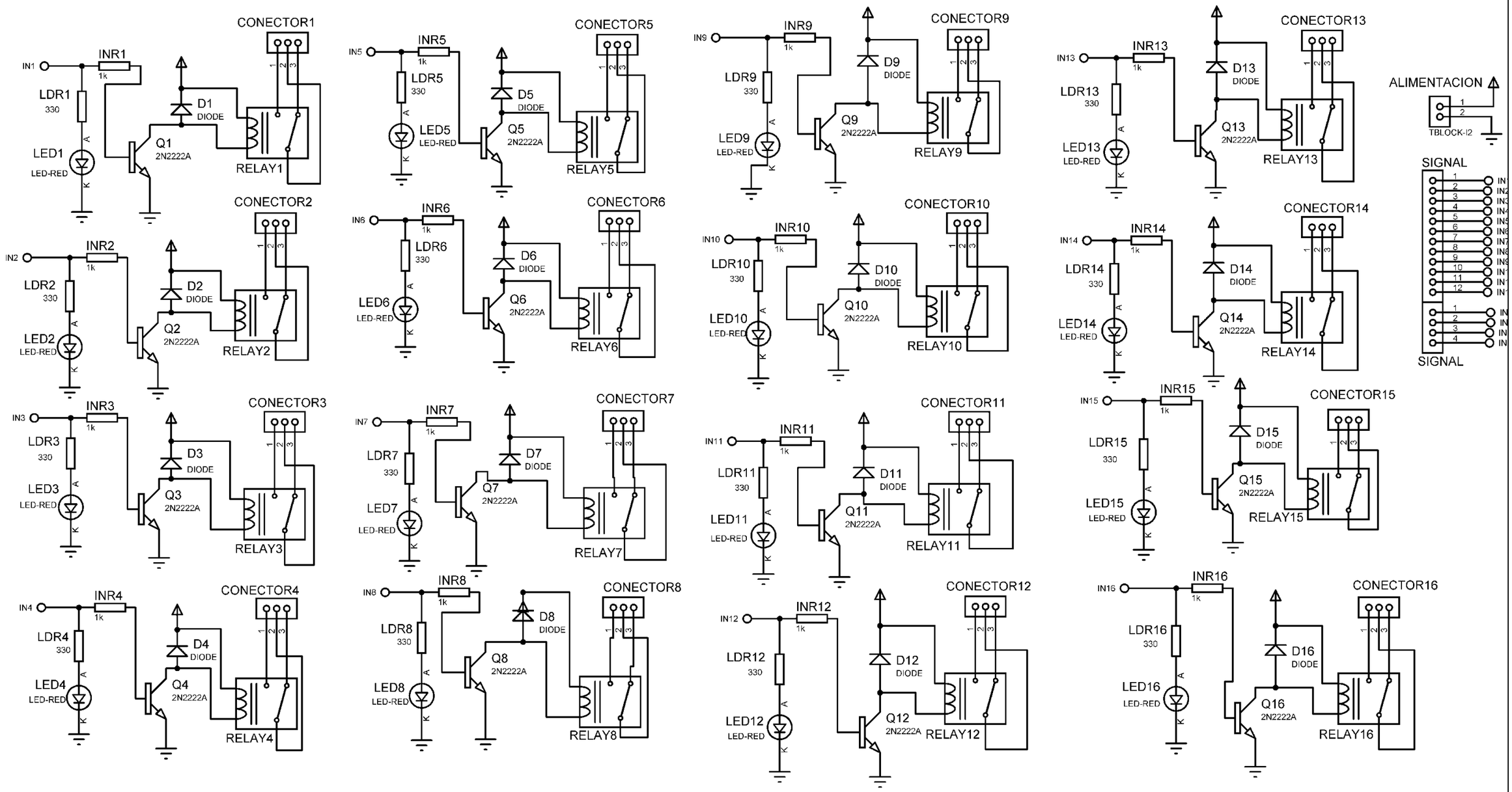
126 mm



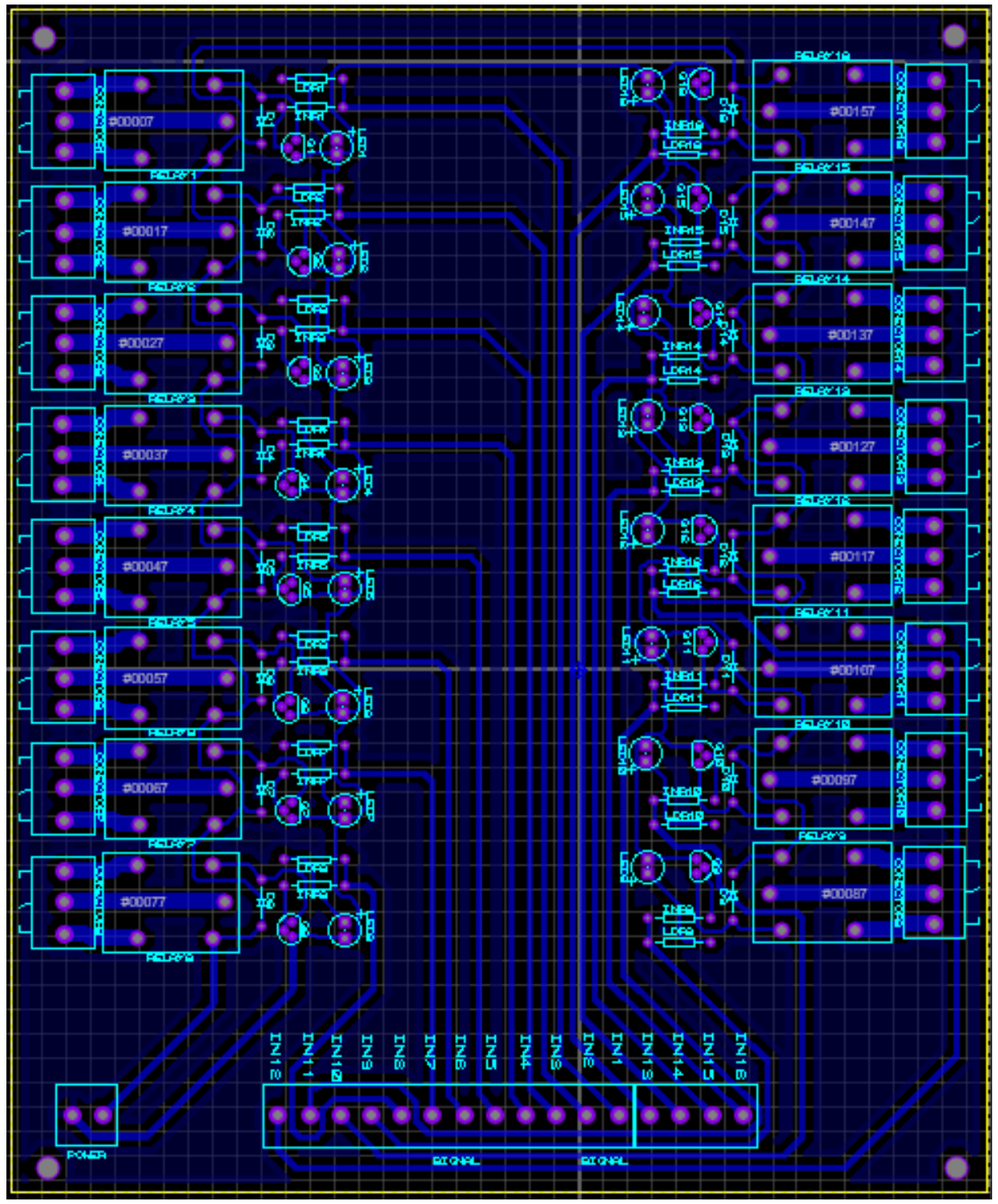
99.6 mm

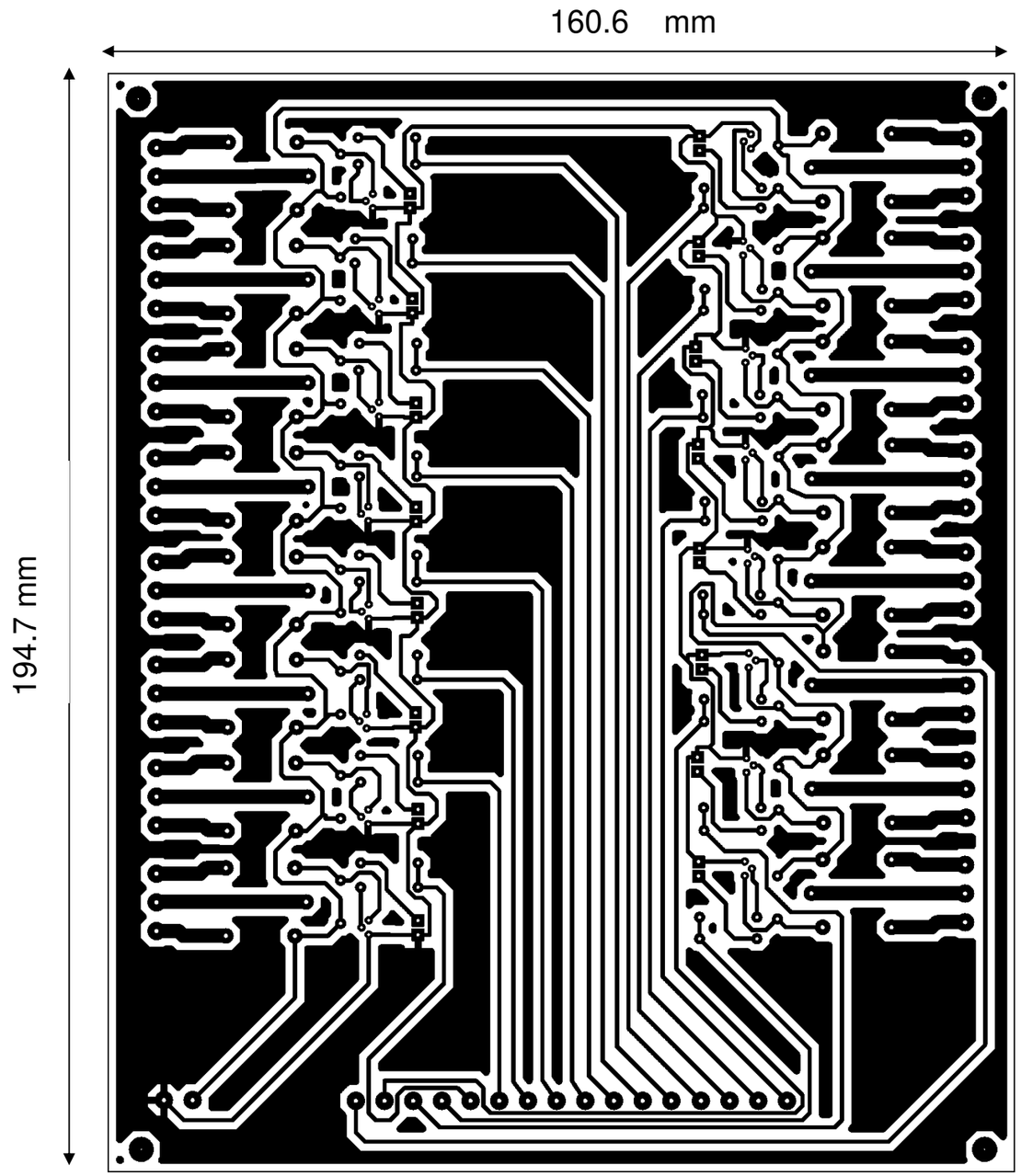
ESCUELA POLITÉCNICA NACIONAL - ESFOT

Esquema PCB	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Eléctrico
	ESCALA: 1:1	**FORMATO:** A4
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 4



ESCUELA POLITÉCNICA NACIONAL - ESFOT		
Diagrama Esquemático	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Eléctrico
	ESCALA: 1:1	**FORMATO:** A3
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 5





ESCUELA POLITÉCNICA NACIONAL - ESFOT		
Esquema PCB	**TRATAMIENTO TÉRMICO:** N/A	**MATERIAL:** Eléctrico
	ESCALA: 1:1	**FORMATO:** A3
DISEÑADO POR: Kevin David Clavijo Guañuna Kevin Fabricio Tipanguano Chicaiza	**FECHA:** 09/08/2021	**LÁMINA:** 6

ANEXO 3: DIAGRAMAS DE FLUJO

DIAGRAMA DE FLUJO *WEMOS* D1 MINI

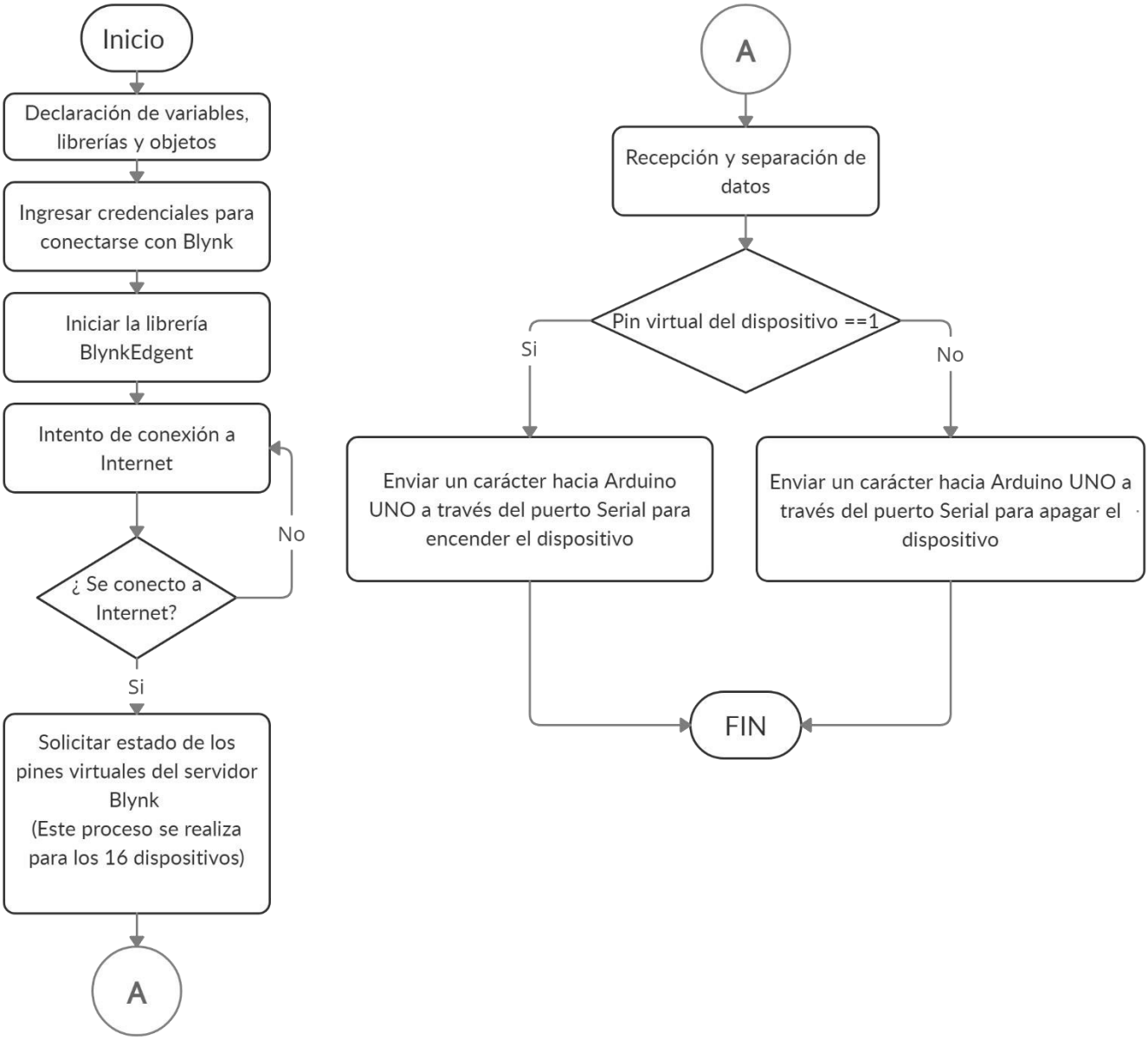
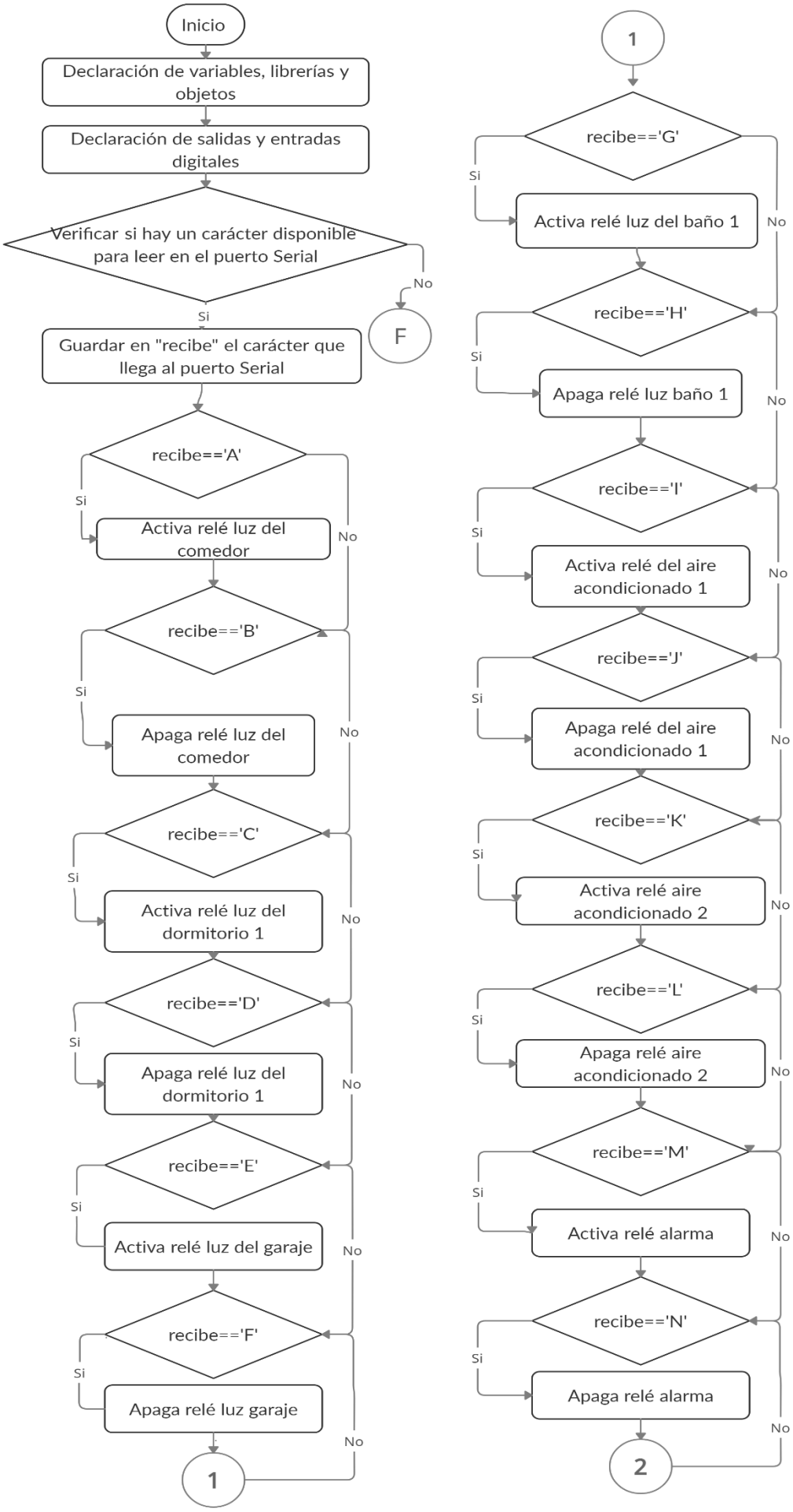
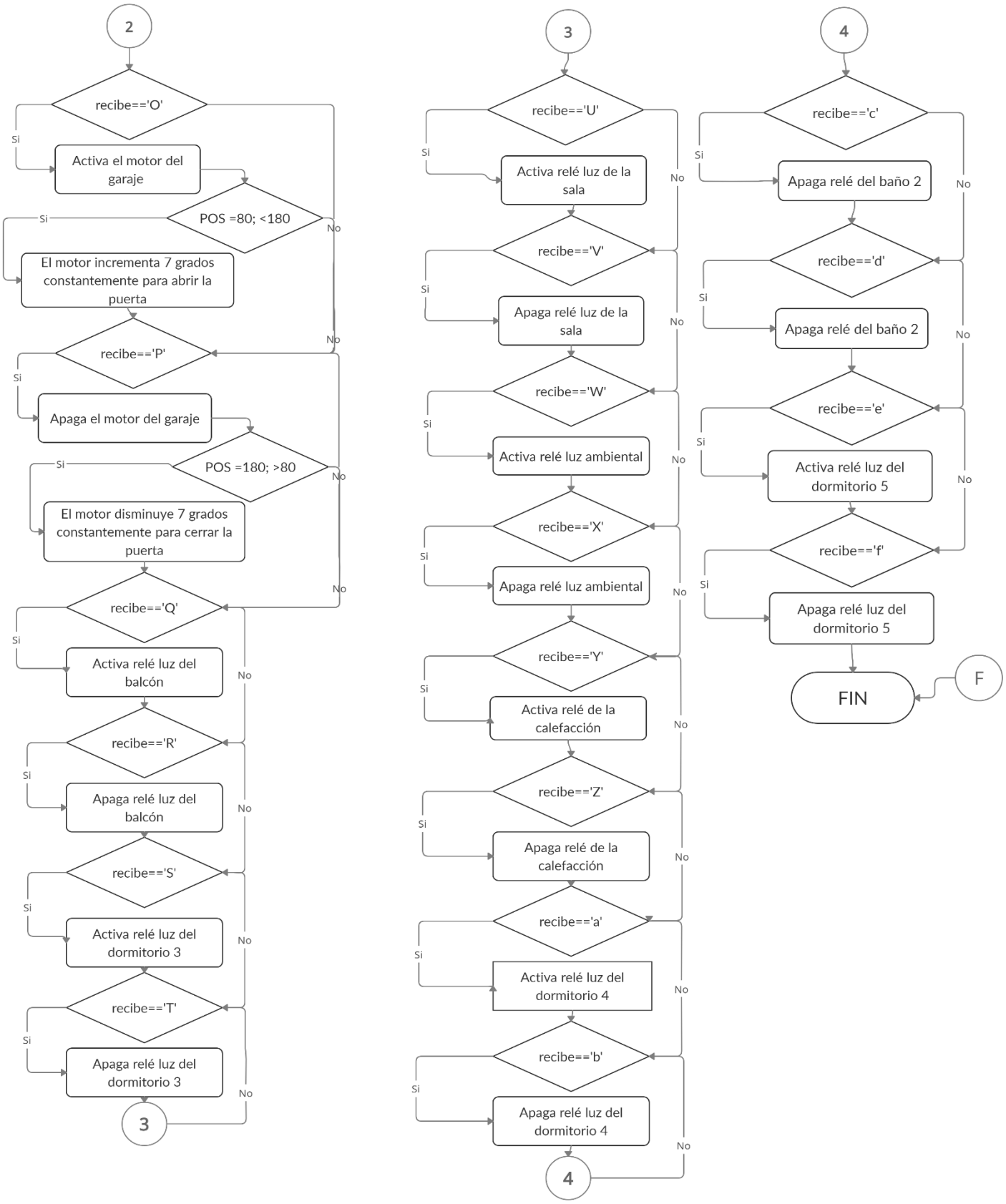


DIAGRAMA DE FLUJO ARDUINO UNO




2
recibe=='O'
Si
Activa el motor del garaje
POS =80; <180
Si
No
El motor incrementa 7 grados constantemente para abrir la puerta
recibe=='P'
No
Si
Apaga el motor del garaje
POS =180; >80
Si
No
El motor disminuye 7 grados constantemente para cerrar la puerta
recibe=='Q'
No
Si
Activa relé luz del balcón
recibe=='R'
No
Si
Apaga relé luz del balcón
recibe=='S'
No
Si
Activa relé luz del dormitorio 3
recibe=='T'
No
Si
Apaga relé luz del dormitorio 3
3
3
recibe=='U'
Si
No
Activa relé luz de la sala
recibe=='V'
No
Si
Apaga relé luz de la sala
recibe=='W'
No
Si
Activa relé luz ambiental
recibe=='X'
No
Si
Apaga relé luz ambiental
recibe=='Y'
No
Si
Activa relé de la calefacción
recibe=='Z'
No
Si
Apaga relé de la calefacción
recibe=='a'
No
Si
Activa relé luz del dormitorio 4
recibe=='b'
No
Si
Apaga relé luz del dormitorio 4
4
4
recibe=='c'
Si
No
Apaga relé del baño 2
recibe=='d'
No
Si
Apaga relé del baño 2
recibe=='e'
No
Si
Activa relé luz del dormitorio 5
recibe=='f'
No
Si
Apaga relé luz del dormitorio 5
FIN
F

ANEXO 4: COMANDOS DE VOZ

Tabla 5.1 Comandos de voz para activar los dispositivos de la planta baja y alta

Dispositivo	Comando de voz	Funcionamiento
Planta baja	Alexa, abre control planta baja.	Activar S*kill* que controla los dispositivos de la planta baja
Planta alta	Alexa, abre control planta alta.	Activar S*kill* que controla los dispositivos de la planta alta

En la Tabla 5.2 y Tabla 5.3 se muestran los comandos de voz para controlar los dispositivos de la planta baja y planta alta de la vivienda.

Tabla 5.2 Comandos de voz que controlan los dispositivos de la planta baja

Dispositivo	Comando de voz	Funcionamiento
Dormitorio 1	Enciende dormitorio uno. Encender dormitorio uno. Prender dormitorio uno	Enciende la luminaria del dormitorio 1
Dormitorio 1	Apaga dormitorio uno. Apagar dormitorio uno.	Apaga la luminaria del dormitorio 1
Comedor	Enciende comedor. Encender comedor dos. Prender comedor	Enciende la luminaria del comedor
Comedor	Apaga comedor. Apagar comedor.	Apaga la luminaria del comedor
Baño uno	Enciende baño uno. Encender baño uno. Prender baño uno.	Enciende la luminaria del baño 1
Baño uno	Apaga baño uno. Apagar baño uno	Apaga la luminaria del baño 1
Parqueadero	Enciende el parqueadero. Encender parqueadero. Prende el parqueadero.	Enciende la luminaria del parqueadero
Parqueadero	Apaga parqueadero. Apagar luz del parqueadero	Apaga la luminaria del parqueadero
Alarma	Enciende alarma. Encender alarma. Prender alarma.	Enciende la alarma
Alarma	Apaga alarma. Apagar alarma	Apaga la alarma
Puerta	Abrir la puerta. Abre la puerta. Abrir garaje.	Abrir puerta de garaje

Dispositivo	Comando de voz	Funcionamiento
Puerta	Cerrar puerta. Cierra la puerta. Cerrar garaje	Cerrar puerta de garaje
Aire acondicionado 1	Enciende aire uno. Encender aire acondicionado uno. Prender aire uno.	Encender ventilador
Aire acondicionado 1	Apaga aire uno. Apagar aire acondicionado uno	Apaga ventilador
Aire acondicionado 2	Enciende aire dos. Encender aire acondicionado dos. Prender aire dos.	Encender ventilador
Aire acondicionado 2	Apaga aire dos. Apagar aire acondicionado dos.	Apaga ventilador
Todos	Enciende todos. Encender todos. Prender todos.	Enciende todos los dispositivos a excepción de la alarma
Todos	Apagar todos. Apaga todos	Apaga todos los dispositivos a excepción de la alarma

Tabla 5.3 Comandos de voz que controlan los dispositivos del piso 2.

Dispositivo	Comando de voz	Funcionamiento
Dormitorio 3	Enciende dormitorio tres. Encender dormitorio tres. Prender dormitorio tres.	Enciende la luminaria del dormitorio 3.
Dormitorio 3	Apaga dormitorio tres. Apagar dormitorio tres.	Apaga la luminaria del dormitorio 3.
Dormitorio 4	Enciende dormitorio cuatro. Encender dormitorio cuatro. Prender dormitorio cuatro.	Enciende la luminaria del dormitorio 4.
Dormitorio 4	Apaga dormitorio cuatro. Apagar dormitorio cuatro.	Apaga la luminaria del dormitorio 4.
Dormitorio 5	Enciende dormitorio cinco. Encender dormitorio cinco. Prender dormitorio cinco.	Enciende la luminaria del dormitorio 5.

Dispositivo	Comando de voz	Funcionamiento
Dormitorio 5	Apaga dormitorio cinco. Apagar dormitorio cinco.	Apaga la luminaria del dormitorio 5.
Baño dos	Enciende baño dos. Encender baño dos. Prender baño dos.	Enciende la luminaria del baño 2
Baño dos	Apaga baño dos. Apagar baño dos	Apaga la luminaria del baño 2.
Balcón	Enciende balcón. Encender balcón. Prende balcón.	Enciende la luminaria del balcón.
Balcón	Apaga balcón. Apagar balcón	Apaga la luminaria del balcón.
Luz ambiental	Enciende luz ambiental. Encender luz ambiental. Prender luz ambiental.	Enciende la luminaria ambiental.
Luz ambiental	Apaga luz ambiental. Apagar luz ambiental.	Apaga la luminaria ambiental.
Sala	Enciende sala. Encender sala. Prender sala.	Enciende la luminaria de la sala.
sala	Apaga sala. Apagar sala.	Apaga la luminaria de la sala.
Calefacción	Enciende calefactor. Encender calefactor. Prender calefactor.	Enciende la calefacción.
Calefacción	Apaga calefactor. Apagar calefactor.	Apaga la calefacción.
Todos	Enciende todos. Encender todos. Prender todos.	Enciende todos los dispositivos.
Todos	Apagar todos. Apaga todos.	Apaga todos los dispositivos.