

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Mantenimiento Eléctrico

**PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA  
APLICACIÓN SMARTPHONE**

Trabajo de grado presentado ante la Ilustre Universidad Técnica del Norte previo a  
la obtención del título de Ingeniero en Mantenimiento Eléctrico

Autor:

Angel Jhonathan Tigse Cuzco

Director:

Ing. Francisco Roberto Naranjo Cobo MSc.

Ibarra – Ecuador

2023



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN**  
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

**1. IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

<b>DATOS DE CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	100403766-7		
<b>APELLIDOS Y NOMBRES:</b>	Tigse Cuzco Angel Jhonathan		
<b>DIRECCIÓN:</b>	Tabacundo, Barrio 6 de enero – parroquia la Esperanza		
<b>EMAIL:</b>	<a href="mailto:ajtigsec@utn.edu.ec">ajtigsec@utn.edu.ec</a>		
<b>TELÉFONO FIJO:</b>	(02) 112373	<b>TELÉFONO MÓVIL:</b>	0999796999

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE
<b>AUTOR (ES):</b>	Tigse Cuzco Angel Jhonathan
<b>FECHA: DD/MM/AAAA</b>	19/07/2023
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniero en Mantenimiento Eléctrico
<b>ASESOR /DIRECTOR:</b>	Ing. Francisco Roberto Naranjo Cobo MSc.

**Constancia**

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor a terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se asume responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 19 días del mes de julio de 2023

**EL AUTOR:**



Tigse Cuzco Angel Jhonathan

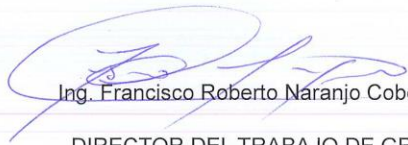
C.I.: 100403766-7

## ACEPTACIÓN DEL DIRECTOR

Ing. Francisco Roberto Naranjo Cobo MSc.

### CERTIFICA

Que después de haber examinado el presente trabajo implementación elaborado por el señor estudiante: Tigse Cuzco Angel Jhonathan, certifico que se ha cumplido con las normas establecidas en la elaboración del trabajo de implementación titulado **“PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE”**. Para la obtención del título de Ingeniero en Mantenimiento Eléctrico: aprobado la defensa, impresión y empastado.



Ing. Francisco Roberto Naranjo Cobo MSc.

DIRECTOR DEL TRABAJO DE GRADO

## **Dedicatoria**

El presente trabajo va dedicado a una persona muy especial que se encuentra en el cielo, sé que estará muy orgullosa al verme alcanzar este nuevo logro en mi vida, para ti con mucho cariño Sammy.

A mi madre Rosita Cuzco por darme todo su amor y apoyo incondicional en todo el transcurso de mi vida universitaria.

A mi padre Angel Tigse por su apoyo incondicional quien ha sido pilar fundamental para lograr culminar todos mis estudios.

A mis hermanos Darwin Tigse, Rosita Tigse, Jeremy Tigse y Zuleima Tigse por su gran apoyo, compañía y aliento durante todo este trayecto.

A Dianita por su apoyo y cariño incondicional que me ha brindado durante mis estudios y por siempre creer en mí.

## **Agradecimiento**

Quiero agradecer a Dios y a mis padres por haberme dado la vida, lo bonito de conocer a personas que con sus conocimientos me ayudaron y apoyaron en la culminación de mi etapa universitaria.

A mi familia:

En primer lugar, agradezco a mis padres por su sabiduría y paciencia que me brindaron, me enseñaron que con trabajo duro todo se puede alcanzar en esta vida. A mis hermanos por su apoyo y compañía que me compartieron en el transcurso de mi carrera.

A mi Director de Tesis:

Ing. Francisco Roberto Naranjo Cobo, por ser un excelente director y maravillosa persona, quien me ha brindado su apoyo y paciencia para la realización del presente trabajo de grado.

A mis maestros:

A todos los docentes de la Carrera de Ingeniería en Mantenimiento Eléctrico por haberme brindado sus conocimientos, que han contribuido a mi formación académica durante todos estos años.

A mis amigos:

Agradezco a mis amigos: Iván Bastidas, Santiago Espinoza y Fernando Ruiz quienes me ayudaron para la culminación de mi trabajo de grado con sus conocimientos compartidos.

## Tabla de Contenido

Dedicatoria.....	V
Agradecimiento .....	VI
Índice de figuras.....	XI
Índice de tablas .....	XIII
Resumen.....	XIV
Abstract .....	XV
INTRODUCCIÓN .....	XVI
Planteamiento del Problema .....	XVII
Problema de Investigación .....	XVII
Alcance.....	XVII
Objetivo general .....	XVIII
Objetivos específicos.....	XVIII
Capítulo 1.....	19
Descripción de los sistemas domóticos .....	19
1.1    Domótica y sus aplicaciones.....	19
1.1.1    Confort .....	19
1.1.2    Seguridad .....	19
1.1.3    Control energético .....	20
1.1.4    Accesibilidad .....	20
1.2    Tipos de arquitectura del sistema domótico .....	21
1.2.1    Arquitectura centralizada.....	21
1.2.2    Arquitectura descentralizada .....	21
1.2.3    Arquitectura mixta.....	22
1.3    Componentes y variables del sistema domótico.....	22
1.3.1    Sensores.....	23
1.3.2    Actuadores .....	26
1.3.3    Controladores.....	28
1.3.4    Interfaz .....	31
1.4    Tecnologías de comunicación .....	33
1.4.1    Wi-Fi.....	33
1.4.2    Bluetooth.....	33
1.4.3    Comunicación GSM .....	34
1.4.4    LoRa.....	34
1.5    Protocolos de comunicación .....	34
1.5.1    Protocolo ZigBee .....	34

1.5.2	Protocolo X10.....	34
1.5.3	Protocolo Z-Wave.....	35
1.6	Topologías de red .....	35
1.6.1	Bus .....	35
1.6.2	Anillo.....	36
1.6.3	Malla o Mesh .....	36
1.7	Plataformas para desarrollo de aplicaciones móviles .....	37
Capítulo 2.....		39
Desarrollo de aplicación móvil para Smartphone .....		39
2.1	Metodología para el desarrollo de sistema domótico.....	39
2.2	Reconocimiento de los espacios físicos de la vivienda .....	40
2.3	Definición de requisitos del sistema domótico .....	41
2.3.1	Soluciones domóticas aplicadas en cada espacio físico .....	41
2.4	Selección de materiales usados dentro del sistema domótico .....	42
2.4.1	Relé para control de dispositivos eléctricos.....	43
2.4.2	Sensor magnético para puertas y ventanas .....	43
2.4.3	Sensor de gases.....	44
2.4.4	Módulo dimmer .....	45
2.4.5	Detector de inundación .....	46
2.4.6	Sensor de temperatura .....	47
2.4.7	Detectores de incendios .....	47
2.4.8	Sensor de vibración .....	48
2.4.9	Cámara de video .....	49
2.4.10	Motores para persianas.....	50
2.4.11	Sistema de alarma .....	51
2.4.12	Módulo GSM .....	51
2.4.13	ESP8266.....	52
2.4.14	ESP32 .....	53
2.4.15	Tarjeta de control Raspberry Pi 3 Modelo B+.....	54
2.4.16	Sistema de respaldo de energía UPS .....	55
2.4.17	Tecnología de comunicación .....	56
2.5	Esquema del sistema domótico .....	56
2.5.1	Esquema de conexión del espacio físico 1.....	57
2.5.2	Esquema de conexión del espacio físico 2.....	57
2.5.3	Esquema de conexión del espacio físico 3.....	58
2.5.4	Esquema de conexión del espacio físico 4.....	58



2.5.5	Esquema de conexión del espacio físico 5.....	59
2.6	Software de programación.....	59
2.6.1	Raspberry Pi OS.....	59
2.6.2	Node-RED .....	59
2.6.3	Arduino IDE .....	60
2.6.4	Software Eagle.....	62
2.6.5	Plataforma para desarrollo de aplicaciones móviles APP Inventor .....	62
2.6.6	Plataforma Firebase.....	63
Capítulo 3.....		65
	Implementación y pruebas de funcionamiento del prototipo domótico en una vivienda ....	65
3.1	Implementación del prototipo.....	65
3.1.1	Diseño y fabricación de la placa electrónica de la cocina.....	65
3.1.2	Diseño y fabricación de la placa electrónica de los dormitorios.....	67
3.1.3	Diseño y fabricación de la placa electrónica de la sala.....	68
3.1.4	Diseño y fabricación de la placa electrónica del comedor.....	70
3.1.5	Diseño y fabricación de la placa electrónica del baño .....	71
3.2	Programación del sistema domótico.....	73
3.2.1	Programación central de control Raspberry Pi.....	73
3.2.2	Programación tarjeta de control ESP32 y ESP8266 .....	75
3.3	Diagrama de flujos de los espacios físicos .....	76
3.4	Diseño de la aplicación móvil en App Inventor .....	81
3.4.1	Diseño de la aplicación en la plataforma Firebase .....	82
3.4.2	Diseño de entorno principal de la aplicación móvil.....	83
3.4.3	Diseño de la aplicación móvil para los espacios físicos .....	83
3.5	Ubicación de sensores y actuadores dentro de la vivienda .....	87
3.6	Pruebas de funcionamiento .....	89
3.7	Análisis y resultados .....	103
3.8	Costo del prototipo .....	103
	Conclusiones .....	105
	Recomendaciones .....	106
1	Referencias .....	107
Anexos.....		113
	Anexo 1: Código de programación para tarjeta de control Raspberry Pi 3 Modelo B+ ....	113
	Anexo 2: Código de programación para tarjeta de control del espacio físico cocina .....	114
	Anexo 3: Código de programación para tarjeta de control del espacio físico sala .....	121
	Anexo 4: Código de programación para tarjeta de control del espacio baño .....	130

Anexo 5: Código de programación para tarjeta de control del espacio físico dormitorio 1 .....	134
Anexo 6: Código de programación para tarjeta de control del espacio físico dormitorio 2 .....	142
Anexo 7: Código de programación para tarjeta de control del espacio físico comedor ...	148
Anexo 8: Código de programación de la aplicación móvil desarrollada en App Inventor	153
Anexo 9: Datasheet Raspberry Pi 3 modelo B+ .....	155
Anexo 10: Datasheet NodeMCU ESP8266 .....	156
Anexo 11: Datasheet NodeMCU ESP32 .....	157
Anexo 12: Datasheet cámara ESP32 CAM.....	158
Anexo 13: Datasheet módulo relé JQC-3FF-S-Z.....	159
Anexo 14: Datasheet motor 28BYJ-48.....	160
Anexo 15: Datasheet sensor de gas MQ-6 .....	161
Anexo 16: Datasheet sensor de humo MQ-135.....	162
Anexo 17: Datasheet sensor de temperatura DS18B20.....	163
Anexo 18: Datasheet sensor de vibración SW-420.....	164
Anexo 19: Datasheet sensor magnético MC-38 .....	165
Anexo 20: Hoja técnica de unidad de respaldo (UPS) R-UPR758.....	166
Anexo 21: Implementación de sistema de control domótico mediante una aplicación Smartphone .....	167
Anexo 22: Implementación de sistema de control domótico mediante una aplicación Smartphone .....	168
Anexo 23: Aplicación móvil para control del prototipo domótico .....	169
Anexo 24: Manual de usuario del sistema domótico.....	170
Anexo 25: Manual técnico del sistema domótico.....	7

## Índice de figuras

Fig. 1.1 Arquitectura centralizada del sistema domótico .....	21
Fig. 1.2 Arquitectura descentralizada del sistema domótico .....	22
Fig. 1.3 Arquitectura mixta del sistema domótico .....	22
Fig. 1.4 Variables de control de un sistema domótico .....	23
Fig. 1.5 Detector magnético.....	24
Fig. 1.6 Sensor de presencia.....	25
Fig. 1.7 Detector de gas.....	25
Fig. 1.8 Detector de fuego .....	26
Fig. 1.9 Relé electromagnético.....	27
Fig. 1.10 Onda senoidal .....	27
Fig. 1.11 Actuador de persianas .....	28
Fig. 1.12 Placa Raspberry Pi.....	29
Fig. 1.13 Placa Arduino .....	30
Fig. 1.14 Módulo ESP32 .....	30
Fig. 1.15 Módulo ESP8266.....	31
Fig. 1.16 Interfaz HMI .....	32
Fig. 1.17 Interfaz con teléfonos inteligentes.....	32
Fig. 1.18 Topología bus. ....	35
Fig. 1.19 Topología anillo.....	36
Fig. 1.20 Topología malla. ....	37
Fig. 2.1 Metodología para el desarrollo del sistema domótico.....	39
Fig. 2.2 Áreas de aplicación del sistema domótico .....	40
Fig. 2.3 Módulo ESP8266 .....	53
Fig. 2.4 Módulo ESP32. ....	53
Fig. 2.5 Esquema general del sistema domótico .....	56
Fig. 2.6 Esquema de conexión de componentes domóticos.....	57
Fig. 2.7 Esquema de conexión de componentes domóticos.....	57
Fig. 2.8 Esquema de conexión de componentes domóticos.....	58
Fig. 2.9 Esquema de conexión de componentes domóticos.....	58
Fig. 2.10 Esquema de conexión de componentes domóticos. ....	59
Fig. 2.11 Software Node-RED.....	60
Fig. 2.12 Interfaz de IDE de Arduino .....	61
Fig. 2.13 Software de diseño EAGLE.....	62
Fig. 2.14 Diseño de aplicación móvil.....	63
Fig. 2.15 Plataforma Firebase .....	64
Fig. 3.1 Diseño placa electrónica cocina .....	66
Fig. 3.2 Diseño PCB cocina .....	66
Fig. 3.3 Placa electrónica cocina .....	67
Fig. 3.4 Diseño placa electrónica dormitorios .....	67
Fig. 3.5 Diseño PCB dormitorios.....	68
Fig. 3.6 Placa electrónica de los dormitorios.....	68
Fig. 3.7 Diseño placa electrónica de la sala .....	69
Fig. 3.8 Diseño PCB sala .....	69
Fig. 3.9 Diseño placa electrónica de la sala .....	70
Fig. 3.10 Diseño placa electrónica comedor .....	70

Fig. 3.11 Diseño PCB comedor.....	71
Fig. 3.12 Placa electrónica comedor .....	71
Fig. 3.13 Diseño placa electrónica baño.....	72
Fig. 3.14 Diseño PCB baño .....	72
Fig. 3.15 Placa electrónica baño.....	73
Fig. 3.16 Pantalla principal de Raspberry Pi .....	73
Fig. 3.17 Diagrama de flujos de funcionamiento Raspberry Pi.....	74
Fig. 3.18 Entorno de programación Arduino .....	75
Fig. 3.19 Diagrama de flujo de la cocina.....	76
Fig. 3.20 Diagrama de flujo de la sala.....	77
Fig. 3.21 Diagrama de flujo del baño .....	78
Fig. 3.22 Diagrama de flujo de los dormitorios.....	79
Fig. 3.23 Diagrama de flujo del comedor.....	80
Fig. 3.24 Diagrama de flujo del funcionamiento de la aplicación móvil .....	81
Fig. 3.25 Plataforma de desarrollo Firebase .....	82
Fig. 3.26 Diseño de la aplicación para los espacios físicos en Firebase .....	82
Fig. 3.27 Entorno principal App Inventor.....	83
Fig. 3.28 Diseño de aplicación móvil para sala.....	84
Fig. 3.29 Diseño de aplicación móvil para el comedor .....	84
Fig. 3.30 Diseño de aplicación móvil para la cocina .....	85
Fig. 3.31 Diseño de aplicación móvil para dormitorios .....	86
Fig. 3.32 Diseño de aplicación móvil para el baño .....	86
Fig. 3.33 Diseño de aplicación móvil para las cámaras de seguridad.....	87
Fig. 3.34 Ubicación de sensores y actuadores dentro del plano arquitectónico .....	89
Fig. 3.35 Instalación de elementos domóticos en el área de la sala. ....	90
Fig. 3.36 Prueba de funcionamiento del espacio físico sala .....	91
Fig. 3.37 Instalación de elementos domóticos en el área del comedor.....	92
Fig. 3.38 Prueba de funcionamiento del espacio físico comedor.....	93
Fig. 3.39 Instalación de elementos domóticos en el área de la cocina .....	93
Fig. 3.40 Prueba de funcionamiento del espacio físico cocina.....	94
Fig. 3.41 Instalación de elementos domóticos en el área del baño .....	95
Fig. 3.42 Prueba de funcionamiento del espacio físico baño.....	96
Fig. 3.43 Instalación de elementos domóticos en el área de dormitorios .....	97
Fig. 3.44 Prueba de funcionamiento del espacio físico dormitorios.....	98
Fig. 3.45 Muestreo de datos de sensores y actuadores.....	99
Fig. 3.46 Prueba de comunicación entre Raspberry Pi y Aplicación móvil reflejada en la plataforma Firebase.....	100
Fig. 3.47 Prueba de funcionamiento sistema de seguridad .....	100
Fig. 3.48 Prueba de funcionamiento cámaras ESP32 CAM .....	101

## Índice de tablas

TABLA 2.1 Función domótica en cada dependencia del domicilio .....	42
TABLA 2.2 Comparativa de módulos de relés .....	43
TABLA 2.3 Características del sensor magnético MC-38 .....	44
TABLA 2.4 Características del sensor de gas .....	44
TABLA 2.5 Características del módulo dimmer .....	45
TABLA 2.6 Características del sensor de agua .....	46
TABLA 2.7 Comparativa de sensores de temperatura .....	47
TABLA 2.8 Comparativa de detectores de incendios .....	48
TABLA 2.9 Características sensor de vibración .....	49
TABLA 2.10 Comparativas cámaras de video .....	49
TABLA 2.11 Comparativa motores para persianas .....	50
TABLA 2.12 Características de alarma del sistema.....	51
TABLA 2.13 Características del módulo GSM SIM800L .....	52
TABLA 2.14 Características tarjeta de control Raspberry Pi 3 modelo B+ .....	54
Tabla 2.15 Características de la unidad de respaldo de energía UPS R-UPR 758.....	55
Tabla 3.1 Pruebas de funcionamiento unitarias a sensores y actuadores en el área de la sala.....	90
Tabla 3.2 Pruebas de funcionamiento unitarias a sensores y actuadores en el área del comedor.....	92
Tabla 3.3 Pruebas de funcionamiento unitarias en el área de la cocina .....	94
Tabla 3.4 Pruebas de funcionamiento unitarias para el área del baño .....	96
Tabla 3.5 Pruebas de funcionamiento unitarias en el área de los dormitorios .....	97
Tabla 3.6 Cálculo del consumo de potencia de los componentes domóticos .....	102
Tabla 3.7 Presupuesto prototipo domótico .....	103

## Resumen

En la actualidad la domótica es un sistema capaz de automatizar un hogar o un edificio, tiene la capacidad brindar servicios tales como gestión energética, confort, accesibilidad, y seguridad para las personas y bienes materiales. La domótica permite controlar diferentes ambientes desde el interior o fuera del hogar mediante una aplicación móvil. El presente trabajo describe el diseño y construcción de un sistema domótico orientado a la seguridad de las personas y sus bienes materiales dentro de una vivienda, el sistema se enfocó en 5 áreas de aplicación domótica; una sala, una cocina, un comedor, un baño y 2 dormitorios, con la instalación de varios sensores y actuadores tales como; sensor de humo, sensor de gas, sensor de inundaciones, sensor de temperatura, sensor de vibración, sensor magnético para puertas y ventanas, módulo relé, módulo dimmer y motores para persianas, todos estos componentes se encuentran conectados a las tarjetas de control ESP32 y ESP8266 dependiendo del área de aplicación. Además, el sistema cuenta con 2 cámaras ESP32 CAM para el monitoreo de seguridad dentro de la vivienda y un sistema de alarma que se conforma de una sirena y él envió de notificaciones de seguridad tanto a la aplicación móvil como al teléfono celular por medio de la comunicación GSM (Global System for Mobile Communications). Todos los procesos del sistema están controlados por la tarjeta Raspberry Pi 3 Modelo B+ y la información se transmite por medio de la tecnología de comunicación Wi-Fi. El resultado fue un sistema domótico funcional enfocado en la seguridad residencial, que mantiene a los usuarios informados a través de notificaciones tanto a la aplicación móvil como al teléfono celular por medio de SMS la presencia de algún problema de seguridad dentro de la vivienda.

**Palabras clave:** domótica, sistema domótico, seguridad, tecnología, sensores, actuadores, aplicación móvil, comunicación GSM.

## **Abstract**

Currently, home automation is a system capable of automating a home or a building, it has the ability to provide services such as energy management, comfort, accessibility, and security for people and material goods. Home automation allows different environments to be controlled from inside or outside the home through a mobile application. This paper describes the design and construction of a home automation system aimed at the safety of people and their material goods within a home, the system focused on 5 home automation application areas; a living room, a kitchen, a dining room, a bathroom and 2 bedrooms, with the installation of several sensors and actuators such as; smoke sensor, gas sensor, flood sensor, temperature sensor, vibration sensor, magnetic sensor for doors and windows, relay module, dimmer module and blind motors, all these components are connected to the ESP32 and ESP8266 depending on the application area. In addition, the system has 2 ESP32 CAM cameras for security monitoring inside the home and an alarm system that consists of a siren and it sends security notifications to both the mobile application and the cell phone through the GSM (Global System for Mobile Communications) communications. All system processes are controlled by the Raspberry Pi 3 Model B+ card, and information is transmitted using Wi-Fi communication technology. The result was a functional home automation system focused on residential security, which keeps users informed through notifications both to the mobile application and to the cell phone via SMS of the presence of any security problem within the home.

**Keywords:** home automation, home automation system, security, technology, sensors, actuators, mobile application, GSM communication.

## INTRODUCCIÓN

En la actualidad los sistemas domóticos se centran en el consumo energético pues es un factor a tomar en cuenta a medio plazo que puede minimizarse mediante un uso eficiente de los dispositivos electrónicos como es la desconexión de estos tras el uso (Suescun, 2020). Es decir, la domótica es un sistema capaz de automatizar un hogar o edificio de cualquier tipo, que sea apto de brindar servicios de gestión energética, seguridad, bienestar, mantenimiento y comunicación, y a la vez que pueda integrarse mediante redes inalámbricas o tecnología Wi-Fi, desde el interior y fuera del lugar mediante el uso de una aplicación (Ovalles, 2018).

Además, la domótica nos permite controlar los sistemas ambientales interiores, los sistemas de iluminación y otros aparatos para aumentar la comodidad de los consumidores (Anubhav, 2020). Asimismo, se presenta un sistema de control que permite controlar persianas/ cortinas, con la principal intención de disminuir el costo por consumo energético (Gutiérrez, 2018).

Un punto clave dentro de la domótica está la seguridad de las personas, las medidas de apoyo de base técnica son cada vez más importantes en una sociedad que envejece. Un ejemplo de una medida de apoyo es un sistema de llamada de emergencia (sistema iHNR), que detecta automáticamente situaciones críticas (Albert, 2020).

La implementación de los nodos sensores para la vigilancia del hogar ha sido de gran importancia dentro de la seguridad, su funcionamiento es a través de un sensor digital efecto hall para monitorear la apertura y cierre de puertas o ventanas y un sensor analógico para el control de temperatura ambiental dentro del hogar (García & Vega, 2018).

Existen varias técnicas para realizar el comando de los dispositivos una de ellas es, el control a través de IP (Protocolo de internet), en la cual cada dispositivo es su propio sistema de control y recibe comandos del usuario por medio de un teléfono inteligente o un navegador web (Adeel, 2016). Sin embargo, otra alternativa es el uso de un módulo de conocimiento por voz que permite la identificación de comandos de diferentes usuarios a través de la implementación de un chip comercial (Errobidart, 2017).

Según Tamayo (2016), en su trabajo menciona que actualmente se ha impulsado a la domótica en el sector inmobiliario brindando un valor agregado al cliente. La construcción es uno de los principales promotores de la domótica, así también uno de los sectores de mayor crecimiento en el mercado ecuatoriano.

Hoy en día es posible encontrar aplicaciones (APPs) que no solo cumplan con las necesidades y requerimientos del usuario, con el pasar de los días estas presentan nuevas



mejoras y actualizaciones. Por otra parte, la combinación de los dispositivos móviles con el internet ha mejorado la vida cotidiana de las personas, le permite realizar tareas del día a día de una manera fácil y eficaz (Tubón Cando, 2020).

### **Planteamiento del Problema**

La percepción actual de la población es que ha aumentado los robos relacionados con la seguridad domiciliaria. Por lo tanto, es prioridad de cada persona cuidar de su familia y bienes, implementando tecnologías que monitoreen lo que sucede dentro de sus viviendas mientras las personas se encuentren ausente. De este modo, previniendo las acciones delictivas.

Asimismo, dentro de la domótica está la seguridad de las personas, las medidas de apoyo de base técnica son cada vez más importantes en una sociedad que envejece. También para las personas con capacidades reducidas, se trata de una ayuda e incluso de una necesidad para poder desenvolverse en el día a día de forma independiente.

### **Problema de Investigación**

¿Cómo desarrollar un prototipo de sistema de control domótico, que permita monitorear la seguridad dentro de una vivienda mediante una aplicación Smartphone?

### **Justificación**

Los sistemas domóticos pueden tener varias áreas de acción, la presente propuesta está enfocada hacia la seguridad de personas adultas, personas con capacidades especiales y en general a persona que deseen mejorar su calidad de vida, la utilización de una aplicación Smartphone permitirá controlar distintas funciones del sistema domótico, mientras la persona se encuentre dentro o fuera del hogar. La implementación de domótica surge como respuesta a una necesidad clara de establecer sistemas automáticos dentro de la rutina diaria, facilitando nuestras actividades. Entre estas destacan: sistemas inteligentes de seguridad, sistemas de encendido o apagado de luces, graduación de luces de una habitación, cierre de persianas, activación de equipos de sonido.

### **Alcance**

El presente trabajo de grado a realizarse está orientado a la seguridad, dentro de una vivienda, por medio de la elaboración de un prototipo de sistema de control domótico mediante una aplicación Smartphone. Las áreas de la vivienda dentro de las cuales se va a implementar este sistema domótico son: una cocina, 2 habitaciones, una sala, un comedor y un baño.

Dentro de las áreas anteriormente especificadas, se realizará la implementación de sensores ante la presencia de fugas de gas, detectores de incendio, sistema de video vigilancia y a la vez sensores en puertas y ventanas que indiquen si alguien irrumpe en la vivienda. Haciendo referencia al control de iluminación; encendido y apagado de luces, control de intensidad de luz y control de persianas. En cuanto a la parte de entretenimiento; control de dispositivos tales como: televisor, equipo de sonido. Ya que son dispositivos desmontables serán usados en el lugar que se necesite.

Además, del uso de controladores y actuadores que llevarán la información a la central del sistema domótico (microcontrolador), mediante una conexión inalámbrica o por medio de cables, esto será de acuerdo a la función de cada uno de los dispositivos, posteriormente el microcontrolador será el encargado de ejecutar todos estos procesos, el desarrollo de la aplicación para Smartphone será en una plataforma de Software libre y su sistema de comunicación con el microcontrolador será por GSM (Global System for Mobile Communications), esta aplicación realizará cada una de las funciones predeterminadas mediante los protocolos establecidos. Además, se desarrollará un manual técnico y de funcionamiento que permita la comprensión al usuario.

### **Objetivo general**

Desarrollar un prototipo de sistema domótico controlado mediante una aplicación móvil con software nativo para Android para automatizar una vivienda.

### **Objetivos específicos**

Describir los sistemas domóticos.

Desarrollar la aplicación móvil para Smartphone.

Realizar pruebas de funcionamiento del sistema domótico en una vivienda.

# Capítulo 1

## Descripción de los sistemas domóticos

En el siguiente apartado se describe información del concepto de domótica, su estructura y sus componentes. Además, los protocolos de comunicación entre los sistemas domóticos y los usuarios. También, se detalla las arquitecturas del sistema y sus aplicaciones, tales como: seguridad, confort, control energético y accesibilidad, haciendo énfasis en la seguridad hacia personas y bienes, que brinda esta tecnología.

### 1.1 Domótica y sus aplicaciones

La domótica está conformada por diferentes tecnologías aplicadas a la monitorización, control y automatización de diversos sistemas y dispositivos dentro de un espacio físico. Los principales objetivos de la domótica son mejorar la seguridad personal y patrimonial dentro del hogar, incrementar el confort y tener una mejor gestión del uso de la energía (García & Vega, 2018).

Según la teoría, las posibilidades de las aplicaciones domóticas sólo están limitadas por la creatividad humana. Sin embargo, se pueden utilizar las siguientes categorías para agrupar los campos de aplicación más utilizados: confort, seguridad, gestión de la energía y accesibilidad.

#### 1.1.1 Confort

La domótica juega un papel muy importante en este punto porque posee el control de todo el inmueble, dicho control es usado fundamentalmente para realizar de forma automática tareas rutinarias de acuerdo a la necesidad de cada persona (Boza, 2017). Esta tecnología brinda al usuario confort en su uso diario, facilitando el encendido y apagado de luces, así como su regulación, control de clima, control de persianas o cortinas, control multimedia y algunos otros sistemas (Ruque, 2016).

#### 1.1.2 Seguridad

Los elementos electrónicos tales como: detectores de fuga de gas, detectores de fuga de agua, detectores de incendio y el control de cierre/apertura de puertas y ventanas se pueden integrar dentro de una vivienda inteligente (Boza, 2017). Además, un sistema domótico inteligente permite al usuario monitorear su hogar con cámaras de seguridad a través de su propio móvil o dispositivo electrónico. Por lo tanto, en cualquier acto delictivo, el usuario puede monitorear lo que está sucediendo dentro de su domicilio y comunicar a tiempo a las autoridades pertinentes (Gazzawe & Lock, 2018).

Se aplica la domótica para crear una red de seguridad encargada de proteger a las personas y los bienes, de posibles intrusos y también para evitar accidentes domésticos, apoyándose en sus dos pilares que son la prevención y la detección.

Un sistema de seguridad se conforma de cuatro niveles:

- **Protección perimetral:** Incluye el control de acceso a todas las ventanas y puertas, para lo que se utilizan, entre otros, los sensores magnéticos y de rotura de cristales.
- **Protección de interior:** con la implementación de sensores de movimiento o de presencia se puede proteger el interior de una vivienda.
- **Teleasistencia/protección personal:** Para las personas mayores o los usuarios que necesitan algún tipo de asistencia, se emplean aparatos con capacidad para escuchar la voz, teleasistencia y pequeños llaveros antipánico.
- **Alarmas técnicas:** Para evitar posibles riesgos para uno mismo o para el hogar, se utilizan equipos de seguridad que incluyen alarmas de fugas de gas, incendios, humo e inundaciones.

### 1.1.3 Control energético

El consumo de energía dentro de nuestros hogares comprende un gasto muy importante. Pero este gasto, y por consiguiente su factura, se puede reducir con la implementación de un sistema domótico enfocado en el control de encendido y apagado de luces, la intensidad de luz se la puede controlar dependiendo de la tarea que se esté realizando. También, se puede llevar a cabo controles periódicamente del uso de energía (Gazzawe & Lock, 2018).

A continuación, se describen otras áreas de aplicación dentro del control energético que son de gran importancia para el usuario:

- Programación de la climatización, permitiendo ofrecer una temperatura adecuada dependiendo de las zonas de la vivienda.
- Eficiencia energética, desconectando equipos eléctricos no prioritarios en función del consumo en un determinado período.
- Detección de problemas en los electrodomésticos cuando se observa un aumento significativo del consumo.

### 1.1.4 Accesibilidad

Dentro de la domótica, la accesibilidad abre nuevas y numerosas posibilidades a las personas mayores o con algún grado de discapacidad, gracias a la disponibilidad de gestionar la vivienda de forma independiente, lo que contribuye enormemente a la mejora del desempeño de sus actividades cotidianas. Toda instalación domótica debe estar al alcance de cualquier

persona, facilitando el uso de los diferentes elementos que se encuentran dentro del hogar, acoplándose a sus necesidades de la manera fácil y sencilla (Game, 2020).

## 1.2 Tipos de arquitectura del sistema domótico

Según Herrera (2020), los tipos de arquitecturas más habituales dentro de la domótica son las siguientes: centralizada, descentralizada y mixta, determinando el modo en que diversos componentes o elementos se conectan o ubican dentro del domicilio.

### 1.2.1 Arquitectura centralizada

Esta arquitectura se caracteriza por poseer un solo controlador central, el cual es el único encargado de recibir toda la información emitida por los sensores, procesarla y realizar acciones a través de los actuadores (Hidalgo Constante, 2017). El controlador central debe ser potente y eficaz caso contrario, si se presenta una falla todo el sistema dejaría de funcionar, siendo esta una de las principales desventajas de esta arquitectura (Alvarado, 2018).

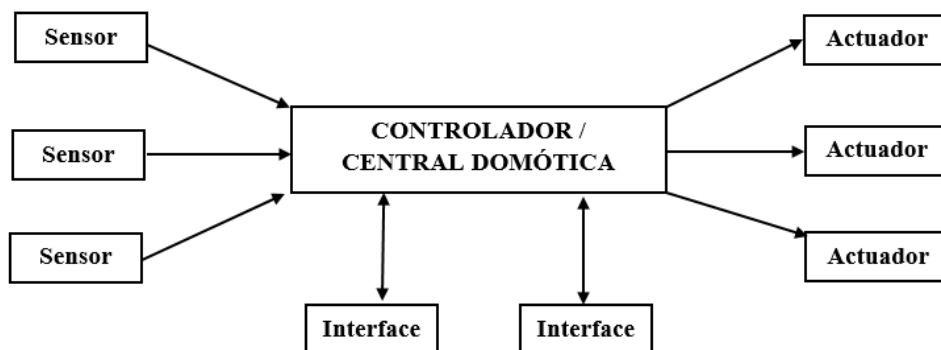


Fig. 1.1 Arquitectura centralizada del sistema domótico  
Fuente: (Alvarado, 2018), adaptado por el autor

En la Figura 1.1 se aprecia un sistema domótico con un controlador centralizado, recibe mediante los sensores y mediante la interfaz los actuadores realizan las funciones determinadas por el usuario.

### 1.2.2 Arquitectura descentralizada

Esta arquitectura no presenta una central de control, cada elemento del sistema cuenta con un controlador inteligente. Por lo tanto, son mucho más caros y presentan una gran flexibilidad cuando se desee realizar una ampliación de elementos (Sánchez, 2017). Además, si un elemento llegara a fallar, las tareas del resto del sistema no presentarían ninguna interrupción como sucede en la arquitectura centralizada (Gracia, 2018).

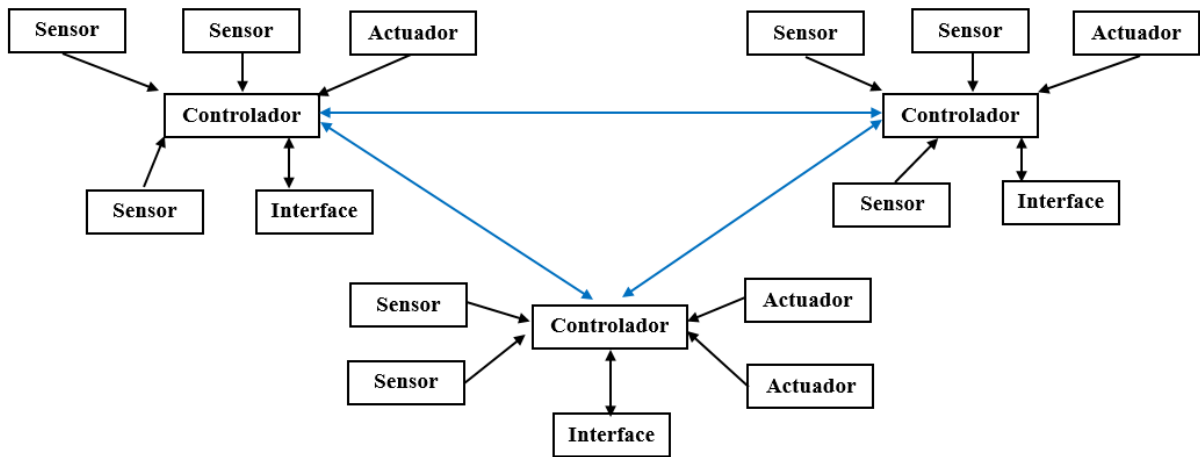


Fig. 1.2 Arquitectura descentralizada del sistema domótico  
Fuente: (Alvarado, 2018), adaptado por el autor

En la Figura 1.2 se observa varios controladores ubicados dentro de un sistema domótico, a diferencia del sistema centralizado que cuenta con un solo controlador central. El diseño de la arquitectura se realizará de acuerdo a los requisitos del usuario, este sistema tiene una mayor flexibilidad en caso de algún daño no colapsaría.

### 1.2.3 Arquitectura mixta

Esta arquitectura es una combinación de las dos anteriores en la que existen varios elementos de control encargados de adquirir y procesar información, posteriormente para transmitirla al resto de dispositivos que componen el sistema (Gracia, 2018). Por lo tanto, son sistemas muy flexibles, aunque en ocasiones complejos en cuanto a su estructura.

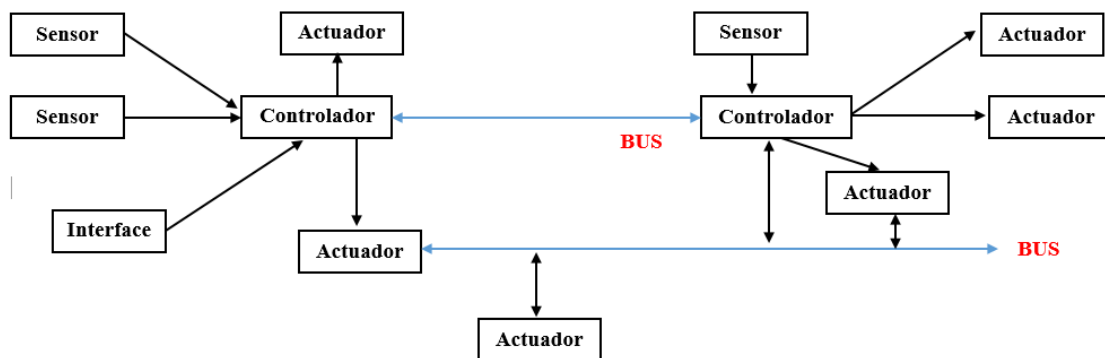


Fig. 1.3 Arquitectura mixta del sistema domótico  
Fuente: (Alvarado, 2018), adaptado por el autor

En la Figura 1.3 se observa una combinación tanto de la arquitectura centralizada con la descentralizada, que se encuentra conformada por diversos elementos de control, los cuales permiten adquirir y procesar información.

## 1.3 Componentes y variables del sistema domótico

Dentro de la domótica existen varios dispositivos que permiten al ser humano controlar todo su hogar con solo un toque, a través de sistemas automatizados con uso de redes de

internet. Los sistemas domóticos están conformados por los siguientes componentes: sensores, actuadores, controladores e interfaces, los mismos que se puntualizan más adelante (Vinueza, 2019).

De la misma forma, las áreas de la vivienda dentro de las cuales se va a implementar el siguiente prototipo de sistema domótico son: una cocina, 2 habitaciones, una sala, un comedor y un baño. Conjuntamente, se va a realizar el control de las siguientes variables que se basó en algunos datos históricos.

- Implementación de sensores de fuga de gas
- Implementación de detector de incendio
- Implementación de detector de inundación
- Monitoreo con cámara de seguridad
- Implementación de sensores de intrusión en puertas y ventanas
- Control de iluminación
- Monitoreo de temperatura
- Control de persianas
- Control de dispositivos multimedia



Fig. 1.4 Variables de control de un sistema domótico  
Fuente: (Ohms, 2016)

En la Figura 1.4 se aprecia las variables que se pueden controlar y monitorear dentro del prototipo de sistema domótico mediante una aplicación para Smartphone, utilizando la tecnología de comunicación inalámbrica.

### 1.3.1 Sensores

Permiten entender el entorno que se desea captar, algunos pueden ser analógicos que tiene como respuesta una señal analógica, es decir, asumen varios valores y los sensores digitales que tienen como respuesta una señal digital, asume valores de 0 y 1 (Rego & Semente, 2020). De la misma forma, los sensores sirven como transductores de señales que

convierten las magnitudes físicas de nuestro interés en señales eléctricas que podemos procesar (García, Chavez, & Jurado, 2017).

Respalda Sanclemente (2016), que existen varios tipos de sensores que son útiles dentro del campo de la domótica para el monitoreo y control de aplicaciones en diferentes entornos. Los siguientes sensores suelen ocuparse en la mayoría de los sistemas domóticos: sensor de presencia, sensor de nivel de luz, detector de gas, detectores magnéticos, detector de fuego, sensor de temperatura.

### a) Detectores magnéticos

Es un elemento que utiliza imanes o energía eléctrica para detectar campos magnéticos (la atracción o repulsión de elementos). Para crear este efecto se utilizan láminas magnetizadas encerradas en una cápsula. Su función principal es recibir una señal y transformarla en los resultados deseados por el usuario. (Gómez, 2020).

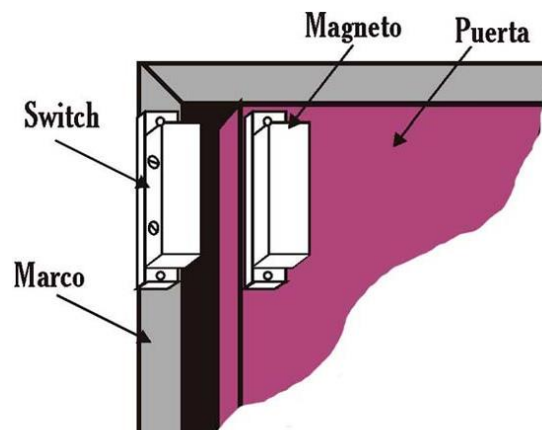


Fig. 1.5 Detector magnético  
Fuente: (Montes, 2018)

La Figura 1.5 muestra cómo se encuentran ubicados los detectores magnéticos, ya sean en puertas y ventanas, con el fin de registrar su estado de apertura o cierre. La mayoría de estos constan de dos partes bien diferenciadas. La primera, donde se encuentra un imán permanente y la otra, que incluye la electrónica correspondiente (Montes, 2018).

### b) Sensor de presencia

Son dispositivos electrónicos que se usan para detectar la presencia de personas u objetos dentro de un área monitoreada, generan una señal si un objeto se encuentra dentro de su zona de detección. Existen diversos tipos de sensores de presencia, ultrasónicos, infrarrojos, ópticos e inductivos (Rego & Semente, 2020).



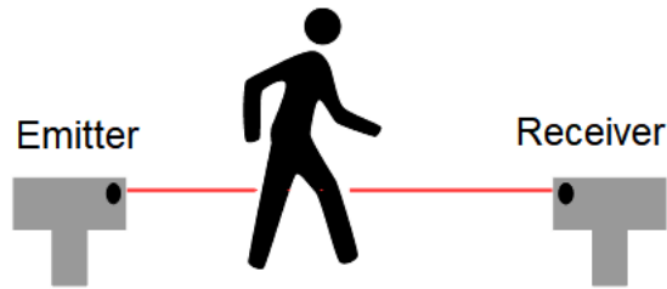


Fig. 1.6 Sensor de presencia  
Fuente: (Semente & Rego, 2018)

En la Figura 1.6 se observa un sensor de presencia que está conformado por dos partes; un emisor y un receptor, para su funcionamiento el emisor emite un haz de infrarrojos y cuando este rayo se interrumpe el sensor envía una señal de paro al controlador.

### c) Sensor de gas

Estos sensores son dispositivos electrónicos que se usan para determinar la concentración de algún gas específico, por lo general se los ubica en zonas donde puede existir la presencia de fugas de gases; dentro de la domótica estos sensores son utilizados para advertir la presencia de gas mediante una alarma sonora y visual, ya que son compatibles con las diferentes tarjetas de control (Feng & Luo, 2018).

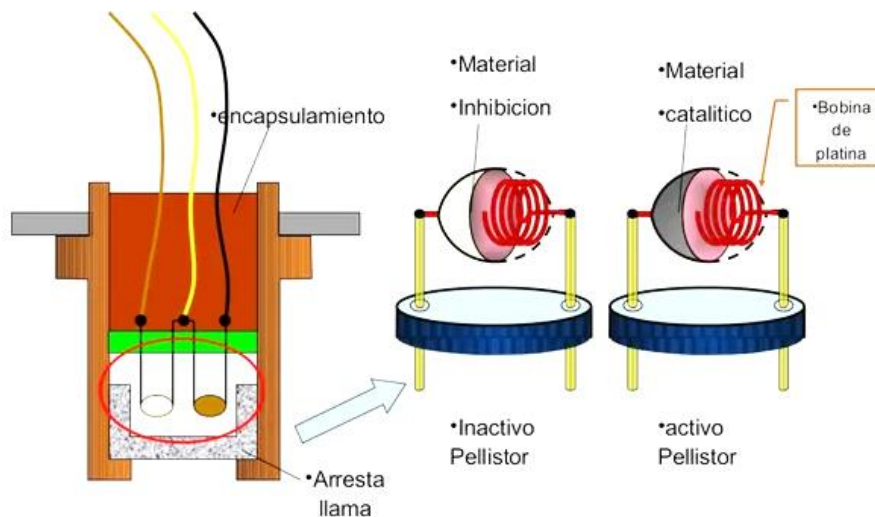


Fig. 1.7 Detector de gas  
Fuente: (Ohms, 2016)

En la Figura 1.7 se aprecia la estructura y funcionamiento de un detector de gas, una corriente pasa por las espiras lo que genera que el gas captado se queme, causando un incremento de la temperatura en su resistencia eléctrica interna, de esta forma el detector envía una señal eléctrica al controlador.

#### **d) Detectores de fuego**

Componente que dispone de un sensor encargado de controlar de forma permanente, prefijada o con intervalos de tiempo, varios fenómenos físicos y/o químicos con objeto de detectar un incendio en la zona o sector que le ha sido asignado y envía de forma automática la correspondiente señal a la central de control (Carrasco, 2016).



Fig. 1.8 Detector de fuego  
Fuente: (Carrasco, 2016)

En la Figura 1.8 se observa un detector de fuego convencional, que generalmente están diseñados para detectar una o más de las tres características elementales del fuego: el humo, el calor y la llama (radiación). De acuerdo al tipo de fuego que se pueda presentar dentro de un entorno, se opta por un detector específico.

#### **1.3.2 Actuadores**

Son dispositivos que se encargan de realizar diversas acciones recibidas desde el controlador del sistema, con el fin de generar un efecto. Al igual que los sensores, los actuadores pueden ser de diferentes tipos (Herrera, 2020). Por ejemplo, si dentro de la casa se detecta una fuga de gas, el actuador por medio del controlador llevará a cabo el corte del gas, evitando algún desastre.

Gracias a estos componentes, los sistemas domóticos son más óptimos, ya que los actuadores efectúan sus funciones de manera automática, brindando al usuario un hogar confortable, seguro e inteligente. Dentro de la domótica, los actuadores más utilizados son; relés, actuadores dimmer y actuadores de persianas.

#### **a) Relé**

Es un dispositivo electromagnético que funciona como un interruptor accionado eléctricamente que abre y cierra el paso de la corriente eléctrica, la señal de control polariza el solenoide generando un campo magnético que permite el cambio de las posiciones de sus contactos eléctricos abriéndolos o cerrándolos (Maldonado & Chacha, 2017).

De igual forma, se sitúan los relés de estado sólido, (Solid State Relays, SSR) los cuales ejecutan una función equivalente a los relés mecánicos, pero se diferencian de estos porque no tienen partes móviles y al ser dispositivos semiconductores adquieren una velocidad de conmutación superior a la de los relés electromecánicos (Went, 2017).



Fig. 1.9 Relé electromagnético  
Fuente: (Maldonado & Chacha, 2017)

En la Figura 1.9 se aprecia un relé electromagnético que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

### b) Actuadores dimmer

Un dimmer es un tipo de circuito electrónico que emplea principalmente componentes de potencia para regular la potencia variando el ángulo de conducción. Este tipo de circuitos trabajan principalmente con corriente alterna (CA), tipo de onda senoidal representada comúnmente en voltaje (Aguilar, 2017).

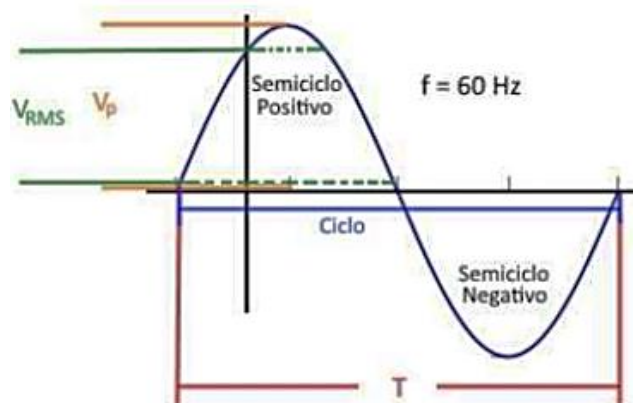


Fig. 1.10 Onda senoidal  
Fuente: (Aguilar, 2017)

En la Figura 1.10 se observa la onda senoidal que al variar su voltaje la intensidad cambia, mientras más grande sea la onda, la intensidad de la lámpara aumenta y viceversa; se puede señalar, que al usar estos dispositivos dimmer dentro de la domótica se logra un control de las luminarias.

### c) Actuadores de persianas

Es un actuador que consta de un motor de corriente continua o alterna, presentan un sistema de cambio de giro y finales de carrera que pueden ser tanto mecánicos como electrónicos. En el caso de los finales de carrera mecánicos, se los regula de manera manual en el dispositivo; en los electrónicos se realiza un recorrido de ensayo para saber que distancia debe recorrer (Maldonado & Chacha, 2017).



Fig. 1.11 Actuador de persianas  
Fuente: (Maldonado & Chacha, 2017)

En la Figura 1.11 se observa un actuador de persianas que, dentro de su estructura, su componente principal es un motor ciego de corriente alterna (AC), este será el encargado de iniciar y accionando el prototipo domótico controlando principalmente la intensidad de la luz natural a través de la apertura o cierre de las persianas.

### 1.3.3 Controladores

Son los encargados de controlar los dispositivos del sistema domótico, de acuerdo con los requisitos establecidos por el usuario (Gutierrez, Acero, & Rodrigo, 2018). Los controladores recogen información de todos los sensores de la vivienda, procesa esta información mediante un algoritmo introducido en su memoria y genera las órdenes necesarias para ejecutar los actuadores (Herrera, 2020).

Entre los controladores más utilizados en proyectos electrónicos de código abierto (open Hardware), dentro de la domótica, se encuentran los siguientes:

- Raspberry Pi
- Arduino
- ESP32

- ESP8266

### a) Raspberry Pi o RPi

Es una tarjeta de control del tamaño pequeño, presenta funcionalidades similares a un computador o computador de placa simple (SBC) de bajo costo, su principal lenguaje de programación es Python, fue desarrollado en Reino Unido por la fundación Raspberry Pi. Además, existe la plataforma RPi tiene una gran comunidad de usuarios activos, lo que redundo en una gran cantidad de documentación; foros, software libre, aplicaciones, así como ideas de proyectos muy atractivos para todas las personas interesadas (Asenjo, y otros, 2017)

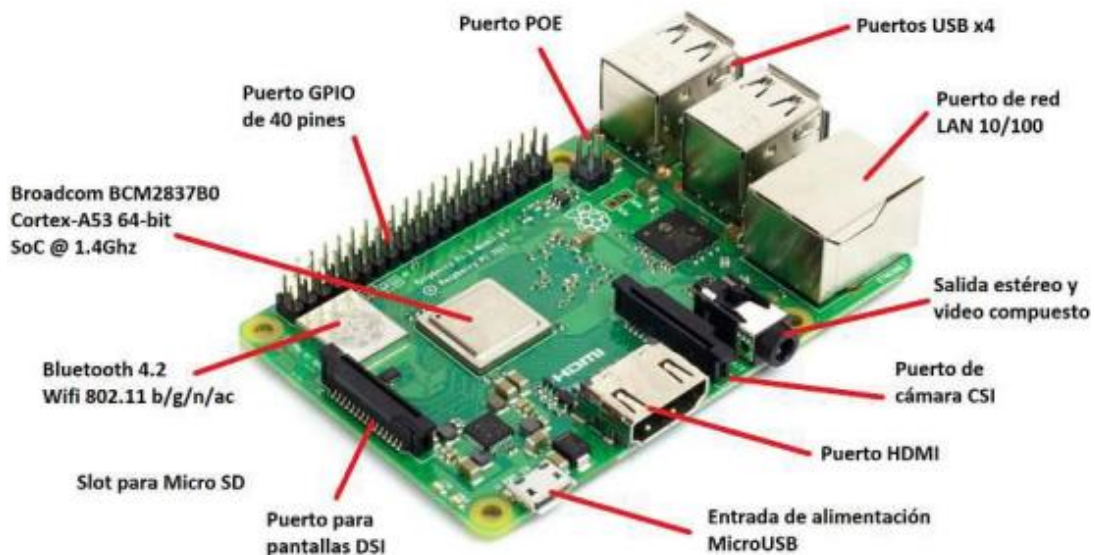


Fig. 1.12 Placa Raspberry Pi  
Fuente: (Franco, 2020)

En la Figura 1.12 se aprecia los componentes de una Raspberry Pi; se observa una salida de interfaz multimedia de alta definición, (High-Definition Multimedia Interface, HDMI) en esta opción la Raspberry funcionará como una computadora de escritorio, una fuente de alimentación de 5 Voltios a, 2.5 Amperios, una ranura para tarjeta micro SD y varios pines o conexiones que se pueden usar como entradas o salidas, (General Purpose Input/Output, GPIO) (Asenjo, y otros, 2017).

### b) Arduino

Arduino es una placa de control basada en un microcontrolador ATMEL. Son circuitos integrados en los que se pueden grabar instrucciones, las cuales se escriben con el lenguaje de programación en el entorno Arduino; esta es una plataforma de creación electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores (ARDUINO, 2018).

El microcontrolador de Arduino cuenta con una interfaz de entrada, con este puerto la placa puede conectarse a diferentes tipos de periféricos. Asimismo, cuenta con una interfaz de salida, que es la encargada de llevar la información que se ha procesado en el Arduino a otros periféricos (ARDUINO, 2018).



Fig. 1.13 Placa Arduino  
Fuente: (ARDUINO, 2018)

En la Figura 1.13 se aprecia el Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16 Mhz, conexión USB, Jack para alimentación DC, conector ICSP, y un botón de reseteo (ARDUINO, 2018).

### c) ESP32

La placa ESP32 es un microcontrolador que fue diseñado por Espressif Systems, una empresa China situada en Shanghái. Dicha empresa comenzó a sacar al mercado módulos Wi-Fi fabricados por Ai-Thinker, que permitían a otros microcontroladores conectarse a redes inalámbricas y realizar conexiones TCP/IP usando comandos AT (Herranz, 2019).



Fig. 1.14 Módulo ESP32  
Fuente: (Herranz, 2019)



### a) Interfaz hombre-máquina

Una interfaz hombre-máquina (HMI) que presenta información al usuario sobre el estado de un proceso, admite comandos y ejecuta los parámetros de control establecidos por el operador. Por lo general, en HMI, la información se muestra en un formato realista o en una interfaz gráfica de usuario (Patil, Bidari, & Sunag, 2016).



Fig. 1.16 Interfaz HMI  
Fuente: (Comunicados, 2021)

En la Figura 1.16 se observa el funcionamiento de la interfaz Hombre-Máquina (HMI), la cual permite al usuario realizar el control de los dispositivos que desee, por medio de una pantalla táctil, una computadora o un teclado.

### b) Interfaz con teléfonos inteligentes

Los teléfonos inteligentes se han vuelto más útiles y fáciles de operar; también, están atrayendo la atención como una herramienta de apoyo para las personas con discapacidad y las personas mayores, que tienen dificultades para operar interfaces anteriores, como teclados y ratones (Tanaka & Fujimoto, 2019).



Fig. 1.17 Interfaz con teléfonos inteligentes.  
Fuente: (Freepik, 2021)



En la Figura 1.17 se aprecia la interfaz de control que se puede realizar a los diferentes elementos instalados dentro del sistema domótico, a través de teléfonos inteligentes.

## **1.4 Tecnologías de comunicación**

Los sistemas domóticos vienen distinguidos por la forma que tienen para transferir toda la información captada por los diferentes dispositivos a la centralita, donde se localiza la programación del sistema, la cual ejecutará acciones dependiendo de la información recogida por dichos dispositivos. Hoy en día, la tecnología de comunicación que se utiliza con mayor frecuencia es la inalámbrica, de la cual se derivan diferentes tipos de tecnologías tales como; Wifi, Bluetooth, comunicación GSM y LoRa.

### **1.4.1 Wi-Fi**

Mediante el uso de radiofrecuencias o infrarrojos para la transferencia de datos, Wi-Fi es una tecnología de comunicación inalámbrica que permite conectar a Internet dispositivos electrónicos como ordenadores, tabletas, smartphones o teléfonos móviles, entre otros. El funcionamiento de una red Wi-Fi se basa en el estándar 802.11, que es una familia de normas inalámbricas creada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), que se lanzó en el año 2009 (Ceja, Renteria, Ruelas, & Ochoa, 2017).

Los router y los equipos Wi-Fi se conectan a uno o varios estándares de los que se mencionan a continuación:

- 802.11a: funciona con conexiones de hasta 54 Mbps, opera en la banda de 5 GHz.
- 802.11b: Funciona con conexiones de hasta 11 Mbps, opera en la banda de 2.4 GHz.
- 802.11g: funciona con conexiones de hasta 54 Mbps, opera en la banda de 2.4 GHz.
- 802.11n: Funciona con conexiones de hasta 600 Mbps, opera en la banda de 2.4 GHz y 5 GHz.
- 802.11ac: funciona con conexiones de hasta 1300 Mbps, opera en la banda de 5 GHz (Ceja, Renteria, Ruelas, & Ochoa, 2017).

### **1.4.2 Bluetooth**

La tecnología Bluetooth permite un alcance universal y conectividad inalámbrica de baja potencia entre dispositivos Bluetooth. Hoy en día, los módulos Bluetooth están integrados en la mayoría de los teléfonos inteligentes, auriculares inalámbricos y computadoras portátiles. No es de extrañar que la tecnología Bluetooth sea la piedra angular del hogar de sistemas de red (Sun, Mu, & Susilo, 2018).

### **1.4.3 Comunicación GSM**

El Sistema Mundial de Comunicaciones Móviles (GSM) es una red internacional de telefonía móvil para la transmisión digital de voz y datos de alta calidad. Para su correcto funcionamiento, debe disponer de un teléfono celular compatible, abonarse al servicio GSM, utilizar un cable GSM y emplear un software de comunicaciones compatible con GSM (Jaimes & Álvarez, 2017).

### **1.4.4 LoRa**

La tecnología inalámbrica LoRa, al igual que Wi-Fi, Bluetooth, y ZigBee, utilizan la modulación del tipo spread spectrum (amplio espectro). Permitiendo tolerar ruidos, caminos múltiples de señal y el efecto doppler; además, gracias a su ancho de banda logra mantener un bajo consumo de energía. LoRaWAN es un protocolo de comunicación inalámbrica desarrollado por LoRa Alliance que se usa para diferentes desafíos que enfrentan las comunicaciones de largo alcance (Bouguera, Diouris, Chaillout, Jaouadi, & Andrieux, 2018).

## **1.5 Protocolos de comunicación**

Los protocolos de comunicación se diferencian por la forma en que los mensajes empleados por los dispositivos y los elementos de control del sistema se comunican entre sí, de tal forma que todos intercambien información sin inconvenientes. A continuación, se detallarán algunos de los protocolos de comunicación que se usan con mayor frecuencia dentro de la domótica (Vázquez, 2019).

### **1.5.1 Protocolo ZigBee**

Este protocolo fue desarrollado por ZigBee Alliance siguiendo el estándar de red inalámbrica de baja velocidad. Es un protocolo de alto nivel y bajo costo que se utiliza para crear redes de área personal con radios digitales pequeñas de baja potencia que transmiten datos en áreas más grandes. También, se pueden realizar diferentes configuraciones de topologías, tales como: estrella, malla y árbol (Cano, Paredes, Alor, Salas, & Segura, 2019).

### **1.5.2 Protocolo X10**

El sistema de corrientes portadoras X10 es un protocolo domótico que hace uso de la red eléctrica de la vivienda para la transmisión de comandos, para las redes a 110-220 voltios y 60 Hertz, con una radiofrecuencia de 433 MHz. Este sistema de comunicación es uno de los más antiguos que existe en domótica. Por lo tanto, es poco fiable para utilizarla hoy en día en la automatización de viviendas (Cano et al., 2019).

### 1.5.3 Protocolo Z-Wave

Es un protocolo de comunicación inalámbrico que utiliza ondas de radio de baja energía para comunicarse de un dispositivo a otro: control de iluminación, sistemas de seguridad, control de puertas, control de estado de ventanas, etc. Al igual que otros protocolos, Z-Wave puede controlarse a través de Internet desde un mando inalámbrico, una pantalla táctil montada en la pared o a través de teléfonos inteligentes o computadoras (Yépez, 2018).

## 1.6 Topologías de red

La topología de red es una disposición de varios componentes tales como: nodos, enlaces y dispositivos; que a su vez permiten el intercambio de datos y cómo interactúan entre sí. Además, ayuda en el monitoreo de dispositivos y en el diagnóstico de algún problema dentro de la red. Existen dos formas de definir la geometría de red: la topología física y la topología lógica (Pathak, 2021).

**Topología de red física:** Las conexiones entre nodos se pueden realizar con cables, alambres, etc.

**Topología de red lógica:** Explica la comunicación interna entre ordenadores, el flujo de datos dentro de la red y la ruta que siguen. Viaja a través de la nube o de los recursos virtuales. De la red lógica se derivan varias tipologías de red, como bus, anillo, estrella y malla.

### 1.6.1 Bus

En esta topología se transfieren los datos por un solo canal de comunicación al que va conectado cada dispositivo de la red. Posee algunas ventajas; la fácil instalación, la poca cantidad de cableado que requiere y la gran disposición a la hora de aumentar o disminuir el número de nodos (Pathak, 2021).

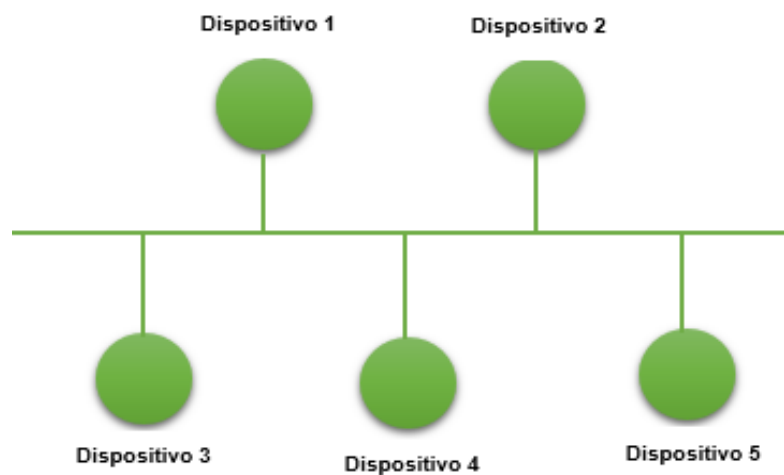


Fig. 1.18 Topología bus.  
Fuente: (Pathak, 2021), adaptado por el autor

La Figura 1.18 muestra la topología bus, en la que varios dispositivos se pueden conectar a la red central, por un solo canal de comunicación.

### 1.6.2 Anillo

En esta red los dispositivos son conectados entre sí formando una estructura de anillo. Los datos son enviados y en el caso de no ser el receptor correcto, lo pasa al siguiente, y así sucesivamente hasta que llegue a su destino. Es decir, la información viaja por todos los nodos hasta llegar a su receptor (Limonas, 2021).

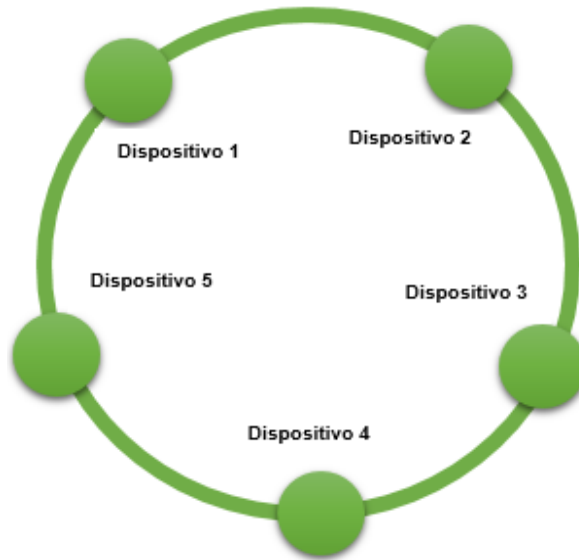


Fig. 1.19 Topología anillo.  
Fuente: (Limonas, 2021), adaptado por el autor

En la Figura 1.19 se observa la topología anillo, donde se pueden conectar varios dispositivos tales como: sensores, actuadores, controladores y varios elementos utilizados en las áreas de domótica.

### 1.6.3 Malla o Mesh

Cada nodo de la topología de malla está conectado a todos los demás, lo que permite enviar mensajes de un nodo a otro a través de varios canales o rutas. Si la red de malla está totalmente conectada, nunca se interrumpirá la comunicación. Cada servidor tiene sus propias conexiones con el resto de los demás servidores (Cruz & Collazos, 2016).

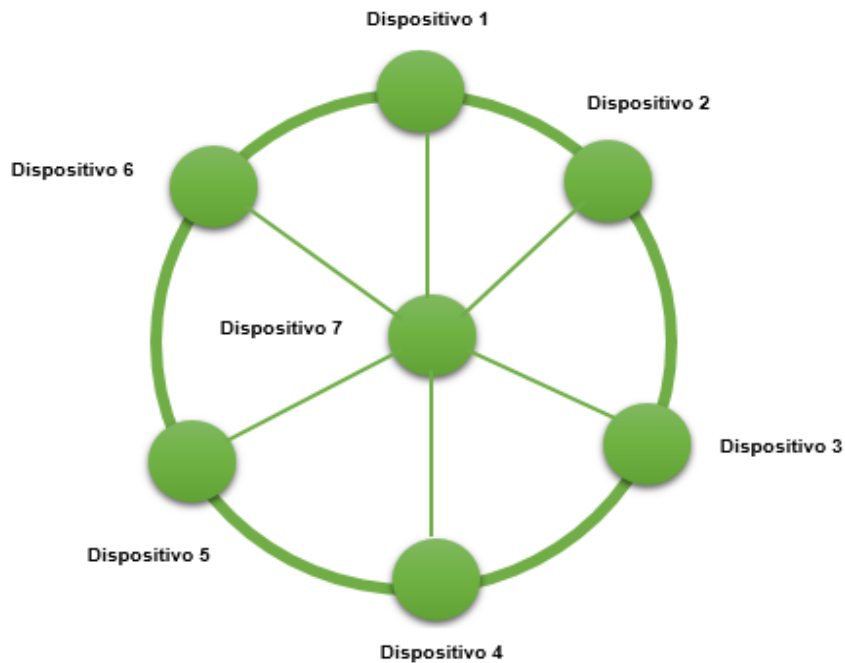


Fig. 1.20 Topología malla.  
Fuente: (Cruz & Collazos, 2016), adaptado por el autor

La Figura 1.20 muestra cómo van conectados los dispositivos dentro de una topología en malla, lo cual es muy beneficioso para las grandes redes del internet de las cosas, (The Internet of Things, IoT) con muchos sensores de baja potencia.

## 1.7 Plataformas para desarrollo de aplicaciones móviles

El desarrollo de aplicaciones móviles es el conjunto de procesos y procedimientos involucrados en la escritura de software para dispositivos informáticos pequeños e inalámbricos, como teléfonos inteligentes y otros dispositivos portátiles (Torres, 2021).

App Inventor es un entorno gratuito de programación visual e intuitivo que permite a todos, crear aplicaciones completamente funcionales para dispositivos móviles Android, la herramienta basada en bloques facilita la creación de aplicaciones complejas en mucho menos tiempo a diferencia de los entornos de programación tradicionales que utilizan lenguajes de programación como Java (Pizarro, 2017).

Antes de desarrollar una aplicación, se seleccionará la más adecuada de acuerdo a las características idóneas para su uso. A continuación, se muestran los tipos de aplicaciones móviles mayormente conocidas.

- **Aplicaciones nativas:** Estas aplicaciones se originan usando entornos de desarrollo integrados (IDE) e idiomas para un determinado sistema operativo móvil. Por lo general, deben de ser modificadas para que puedan adaptarse a distintos dispositivos o distintas plataformas (Torres, 2021).

- **Aplicaciones híbridas.** Son programas que combinan características de las aplicaciones en línea y nativas que pueden funcionar en distintos sistemas operativos, combinan aspectos de las aplicaciones nativas y de las aplicaciones web. Se desarrollan usando tecnologías como Lenguaje de Marcas de Hipertexto, (HyperText Markup Language, HTML), hojas de estilo en cascada (Cascading Style Sheets, CSS) y Java Script (Matthew, 2021).

## **Conclusión**

Luego de realizar el fundamento teórico sobre prototipos domóticos, sus aplicaciones, su estructura y sus componentes electrónicos, se puede tener una visión clara sobre los elementos que se emplearán en el siguiente capítulo, destacando las ventajas de la plataforma de desarrollo App Inventor al ser un programa gratuito y de código abierto que usa lenguaje de programación por bloques, muy popular en la actualidad.

## Capítulo 2

### Desarrollo de aplicación móvil para Smartphone

En este apartado se consideró los elementos específicos para la elaboración del prototipo de sistema de control domótico, de la misma manera se analizó el funcionamiento de los mismos y se comprobó si cumplen con las exigencias necesarias. Además, se detalló el proceso de diseño del sistema de control domótico, incluyendo una descripción de los elementos y variables a controlar en cada uno de los espacios físicos.

Se describió el software y hardware para monitorear y controlar el sistema domótico dentro de una vivienda, mediante el Internet. Por otra parte, se describió el software que se utilizó para el desarrollo de la aplicación móvil.

#### 2.1 Metodología para el desarrollo de sistema domótico

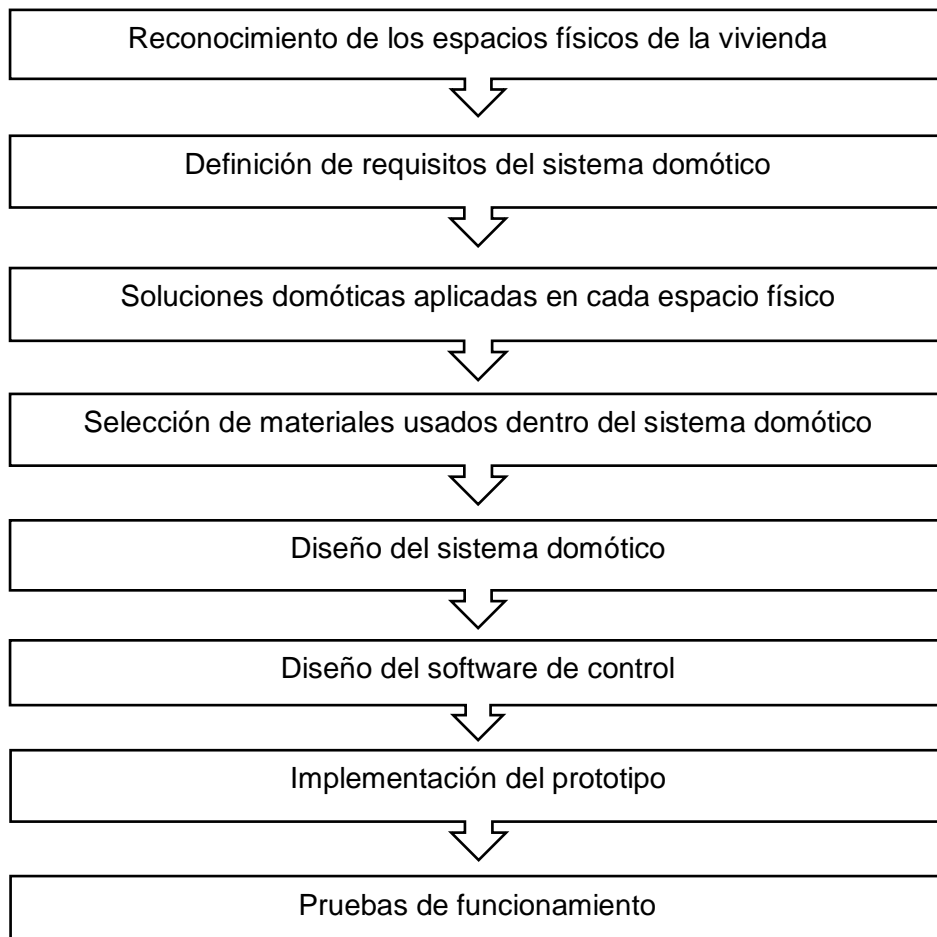


Fig. 21 Metodología para el desarrollo del sistema domótico  
Fuente: (Autor,2021)

El tipo de investigación que se utilizó en el presente proyecto es científico/experimental debido a que se basa en el estudio de otros prototipos domóticos ya existentes, con el fin de obtener información importante para realizar el trabajo propuesto. Además, se hizo uso de contexto de referencias bibliográficas tales como: artículos científicos, tesis de grado, libros digitales, etc. Que permitió la elección de información que se ajustan al diseño de un sistema domótico para finalmente llegar a un resultado por medio de las pruebas de funcionamiento del prototipo domótico.

## 2.2 Reconocimiento de los espacios físicos de la vivienda

A continuación, la Figura 2.2 muestra las áreas de aplicación dentro de una vivienda de una sola planta que cuenta con un área de 55 m<sup>2</sup>, donde se realizó las pruebas de funcionamiento de los diferentes elementos que conforman el prototipo de sistema de control domótico.

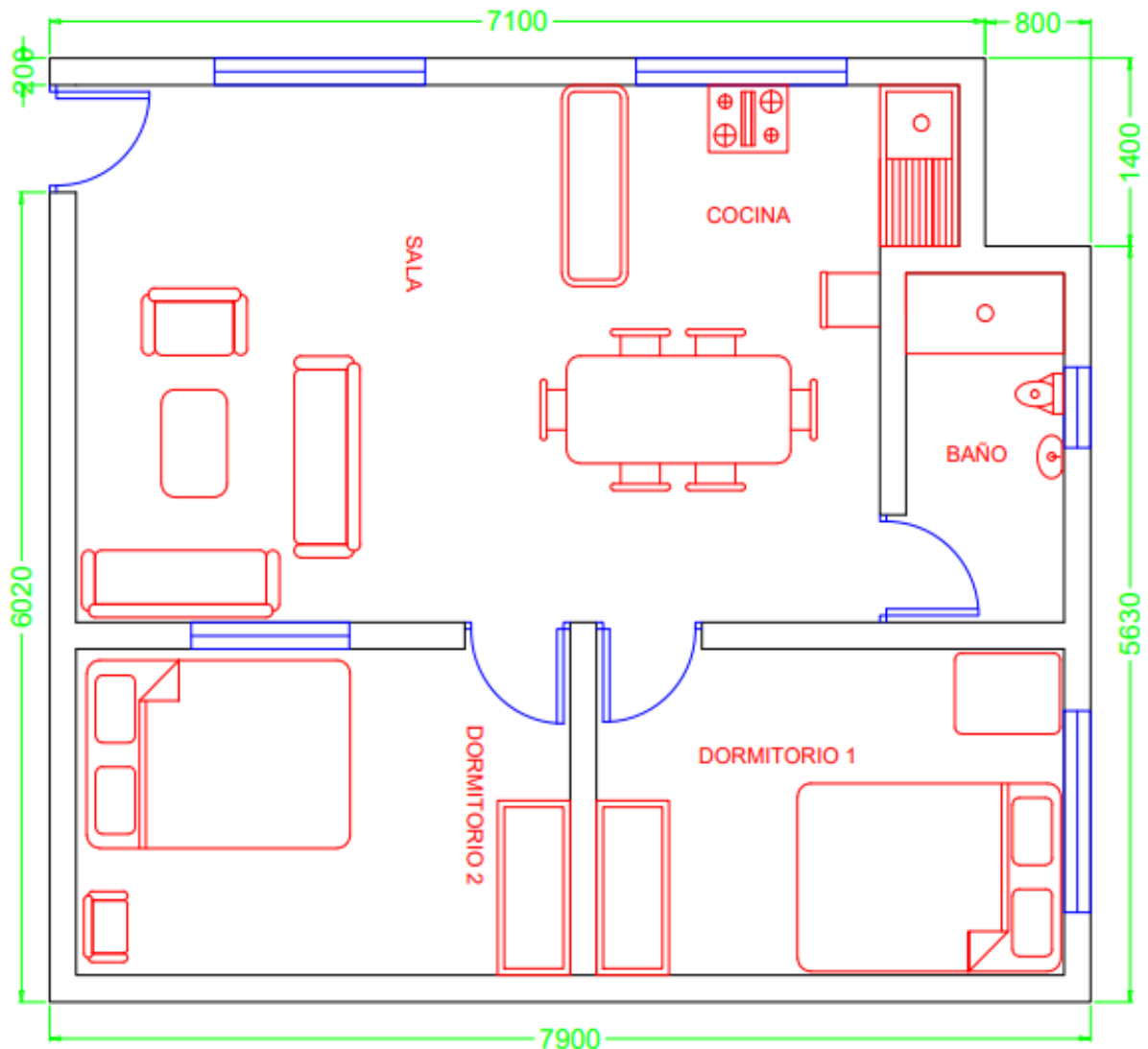


Fig. 22 Áreas de aplicación del sistema domótico  
Fuente: (Autor, 2021)



## 2.3 Definición de requisitos del sistema domótico

El prototipo busca satisfacer ciertas necesidades que el usuario requiera dentro de la vivienda teniendo en cuenta las aplicaciones domóticas descritas en el capítulo anterior, por lo tanto, se procedió a detallar las características y funciones de dicho sistema domótico el cual debe cumplir con los siguientes requisitos enfocados en el control energético, confort y la seguridad y protección de los bienes del usuario:

- **Control de luminarias.** \_ mediante un dispositivo móvil, deberá brindar un mecanismo de encendido, apagado y control de intensidad de luminarias de acuerdo a las necesidades del usuario.
- **Control de equipos de entretenimiento.** \_ mediante un dispositivo móvil, deberá brindar un mecanismo de encendido y apagado de equipos de entretenimiento.
- **Monitoreo de acceso a la vivienda.** \_ deberá informar a través del dispositivo móvil si las puertas y ventanas de la vivienda se encuentra abierta o cerrada.
- **Monitoreo de temperatura.** \_ el sistema deberá brindar una temperatura adecuada al usuario por medio de dispositivos de calefacción, controlados a través de un Smartphone.
- **Monitoreo con cámara de video.** \_ implementar un mecanismo que permita monitorear intrusiones de personas ajenas a la vivienda.
- **Monitoreo de fugas de gas o incendios.** \_ deberá informar si existe fugas de gas o presencia de fuego dentro de la vivienda.
- **Monitoreo de inundaciones.** \_ al sistema le corresponderá informar al usuario la presencia de fugas de agua dentro del hogar.
- **Accionamiento de persianas.** \_ accionamiento de un mecanismo automático para el control de apertura y cierre de persianas mediante una interfaz móvil.
- **Sistema de seguridad.** \_ implementar un grupo de sensores en diferentes puntos estratégicos dentro de la casa que, al censar algún problema en el sistema activen una alarma sonora y luminosa y a su vez envíen mensajes de texto al celular del usuario mediante la comunicación GSM.
- **Monitoreo de estado de ventanas.** \_ ante la ruptura de una de las ventanas en la vivienda y por medio de sensores se activará la alarma del sistema de seguridad.

### 2.3.1 Soluciones domóticas aplicadas en cada espacio físico

En la Tabla 2.1 se detalló las funciones domóticas comprendidas en cada uno de los espacios físicos dentro de la vivienda, para el diseño de este prototipo se consideró las siguientes áreas: una cocina, dos dormitorios, una sala, un comedor y un baño. Además del sistema de seguridad que estará presente en cada una de estas áreas.

Tabla 2.1 Función domótica en cada dependencia del domicilio

<b>Espacio físico 1</b>	<b>Solución domótica</b>
Cocina	<ul style="list-style-type: none"> <li>• Control de luminarias.</li> <li>• Monitoreo de estado de ventanas.</li> <li>• Monitoreo de fuga de gas.</li> <li>• Monitoreo de incendios</li> <li>• Monitoreo de inundaciones.</li> </ul>
<b>Espacio físico 2</b>	<b>Solución domótica</b>
Dormitorios	<ul style="list-style-type: none"> <li>• Control de luminarias.</li> <li>• Monitoreo de estado de ventanas.</li> <li>• Control de persianas.</li> <li>• Monitoreo de acceso a la vivienda.</li> </ul>
<b>Espacio físico 3</b>	<b>Solución domótica</b>
Sala	<ul style="list-style-type: none"> <li>• Control de luminarias.</li> <li>• Monitoreo de estado de ventanas.</li> <li>• Monitoreo de acceso a la vivienda.</li> <li>• Control de persianas.</li> <li>• Control de equipos de entretenimiento.</li> <li>• Control de temperatura.</li> <li>• Monitoreo con cámara de video.</li> </ul>
<b>Espacio físico 4</b>	<b>Solución domótica</b>
Comedor	<ul style="list-style-type: none"> <li>• Control de luminarias.</li> <li>• Monitoreo de estado de ventanas.</li> <li>• Monitoreo de acceso a la vivienda.</li> </ul>
<b>Espacio físico 5</b>	<b>Solución domótica</b>
Baño	<ul style="list-style-type: none"> <li>• Control de luminarias.</li> <li>• Monitoreo de estado de ventanas.</li> <li>• Monitoreo de inundaciones.</li> </ul>

Fuente: (Autor,2021)


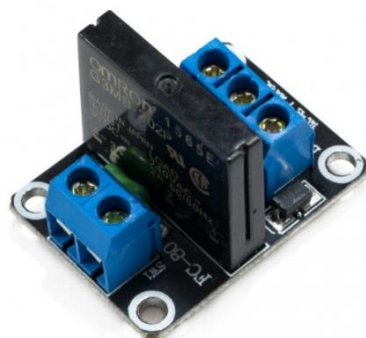
## 2.4 Selección de materiales usados dentro del sistema domótico

A continuación, se describen los principales componentes electrónicos y sus características principales, que se utilizó para el presente prototipo domótico y los servicios a brindar en cada espacio físico de la vivienda.

### 2.4.1 Relé para control de dispositivos eléctricos

Hardware para el control de luminarias, aparatos de calefacción y equipos de entretenimiento, la mayor parte de los domicilios cuentan con un voltaje de 120V, por lo cual es necesario la utilización de relés que permitan el accionamiento de estos dispositivos. A continuación, se detalló las características de dos tipos de relés que son utilizados con mayor frecuencia dentro de la domótica

Tabla 2.2 Comparativa de módulos de relés

Módulo de relé JQC-3FF-S-Z		Módulo de relé G3MB-202P-DC5	
			
Características	Descripción	Características	Descripción
<b>Voltaje nominal</b>	5 VDC	<b>Voltaje nominal</b>	5 VDC
<b>Corriente máxima</b>	10 A	<b>Corriente máxima</b>	2 A
<b>Tiempo de acción</b>	5 ms - 10 ms	<b>Tiempo de acción</b>	1 ms
<b>Voltaje de carga</b>	10A/250VAC	<b>Voltaje de carga</b>	2A/240VAC
<b>Consumo de corriente</b>	90 mA	<b>Consumo de corriente</b>	12.5 mA
<b>Precio</b>	2.50 dólares	<b>Precio</b>	5 dólares

Fuente: (Llamas, 2016)

### Conclusión

Para el presente prototipo se optó por el módulo de relé JQC-3FF-S-Z por su mayor rango de corriente de carga y su bajo precio, así como otras características definidas en la tabla 2.2. Además, el relé es compatible con sistemas de 5v, como Arduino y también con sistemas de 3.3 V como Raspberry Pi.

### 2.4.2 Sensor magnético para puertas y ventanas

Para la detección de apertura y cierre de puertas y ventanas se utilizó un sensor magnético MC-38, el cual consta de dos partes, una móvil donde se encuentra un imán y la otra parte fija que está conformada de un interruptor. Las características de este sensor se muestran en la Tabla 2.3.

Tabla 2.3 Características del sensor magnético MC-38

**Sensor magnético MC-38**



Características	Descripción
Voltaje nominal	200 VDC
Corriente máxima	0.5 A
Salida del sensor	Normalmente cerrado
Distancia de funcionamiento	15 – 25 mm
Longitud del cable	30,5 cm
Precio	2 dólares

Fuente: (Components, 2021)

**2.4.3 Sensor de gases**

Este sensor al estar en contacto con el gas licuado de petróleo (GLP), su resistencia interna varía, su módulo contiene un circuito electrónico que permite realizar la conexión con alguna tarjeta de desarrollo, y posee una salida analógica y otra digital, dentro de la familia de los sensores MQ, existen varios tipos tales como: MQ2, MQ3, MQ4, MQ5, MQ6, MQ135, MQ131.

Tabla 2.4 Características del sensor de gas

**Sensor MQ2**

**Sensor MQ6**



Características	Descripción	Características	Descripción
-----------------	-------------	-----------------	-------------

<b>Voltaje de operación</b>	5 VDC	<b>Voltaje de operación</b>	5 VDC
<b>Detección de gases</b>	Gas combustible y humo	<b>Detección de gases</b>	Butano, propano y GLP
<b>Temperatura de trabajo</b>	-20 a 55 °C	<b>Temperatura de trabajo</b>	-10 a 50 °C
<b>Rango de detección</b>	300 a 10000 ppm	<b>Rango de detección</b>	300 a 10000 ppm
<b>Precio</b>	4.50 dólares	<b>Precio</b>	5 dólares
<b>Consumo de corriente</b>	150 mA	<b>Consumo de corriente</b>	160 mA

Fuente: (Ideastronic, 2021)

## Conclusión

Con el propósito de alcanzar un correcto funcionamiento del prototipo, se eligió el sensor de gas MQ6 descrito en la Tabla 2.4, que al contar con una mayor sensibilidad a la detección de gas licuado de petróleo (GLP) fue el más adecuado para el presente proyecto; cabe destacar, que este módulo es compatible con tarjetas de control como Arduino, ESP32 y Raspberry Pi.

### 2.4.4 Módulo dimmer

El módulo dimmer permite controlar cargas de voltaje alterno desde un microcontrolador como Arduino, Raspberry y ESP32, permitiendo obtener voltajes intermedios entre 0 y 220V AC y de esa forma variar la intensidad de brillo en bombillas incandescentes, la velocidad en motores de corriente alterna como ventiladores (Varper, 2021).

En la Tabla 2.5 se detallaron las características más relevantes del módulo dimmer AC BTA16-600BT, se utilizó el elemento anteriormente mencionado por su alto rango de voltaje de carga para el control de la intensidad de luces instaladas dentro de los espacios físicos del sistema domótico.

Tabla 2.5 Características del módulo dimmer

#### módulo dimmer AC BTA16-600BT



#### Características

#### Descripción

<b>Voltaje de control</b>	3.3 - 5 VDC
<b>Voltaje de carga</b>	110/220 VAC
<b>Corriente máxima de carga</b>	5 A
<b>Potencia de carga</b>	440 Watts
<b>Peso</b>	24 g
<b>Temperatura de operación</b>	-20 °C a 80 °C

Fuente: (Varper, 2021)

#### 2.4.5 Detector de inundación

Los detectores de inundaciones son dispositivos electrónicos que permiten detectar fugas de agua con la mayor brevedad posible, notificando a los propietarios por medio de alarmas instaladas dentro del hogar, con el propósito de evitar daños en aparatos eléctricos. Los lugares más habituales donde se instalan estos dispositivos suelen ser baños, cocinas y lugares susceptibles de inundarse (Lahera, 2022).

El sensor detallado en la Tabla 2.6 se lo utilizó para detectar la posible presencia de fugas de agua, el mismo se lo instaló en el baño y la cocina dentro de la vivienda, que son puntos específicos donde se pueden presentar inundaciones.

Tabla 2.6 Características del sensor de agua

#### Sensor capacitivo de nivel de líquido



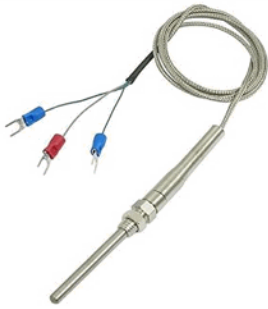

<b>Características</b>	<b>Descripción</b>
<b>Voltaje de operación</b>	5 VDC
<b>Señal de salida</b>	Digital
<b>Temperatura de operación</b>	0 °C a 40 °C
<b>Tiempo de respuesta</b>	1 segundo
<b>Precio</b>	3 dólares
<b>Consumo de corriente</b>	5 mA

Fuente: (Prastyo, 2020)

### 2.4.6 Sensor de temperatura

Un sensor de temperatura es un dispositivo que mide la temperatura a través de señales eléctricas. Este dispositivo electrónico convierte la información obtenida en datos electrónicos que se pueden monitorear y registrar. El control de la temperatura es usado muy frecuentemente en el campo de la domótica para brindar al usuario confort dentro de su casa (Hernández, 2018).

Tabla 2.7 Comparativa de sensores de temperatura

Sensor de temperatura PT-100		Sensor de temperatura DS18B20	
			
Características	Descripción	Características	Descripción
<b>Voltaje nominal</b>	5 VDC	<b>Voltaje nominal</b>	5 VDC
<b>Rango de temperatura</b>	-100 °C a 200°C	<b>Rango de temperatura</b>	-55 °C a 125°C
<b>Tipo de sensor</b>	Termoresistencia	<b>Tipo de sensor</b>	Termopar
<b>Precio</b>	8 dólares	<b>Precio</b>	5 dólares
<b>Consumo de corriente</b>	5 mA	<b>Consumo de corriente</b>	5 mA

Fuente: (Hernández, 2018)

### Conclusión

Para el presente prototipo se seleccionó el sensor de temperatura DS18B20 por sus características técnicas detalladas en la Tabla 2.7 y al contar con un adaptador para la placa Arduino es una gran ventaja para su funcionamiento.

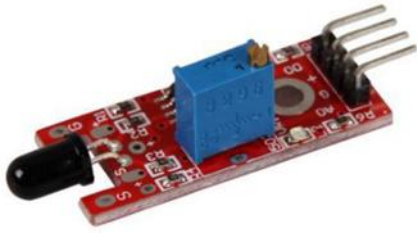
### 2.4.7 Detectores de incendios

Los detectores de incendio son dispositivos capaces de detectar el fuego lo más pronto posible a través de los fenómenos que este presenta tales como: gases, humos, temperaturas y radiaciones UV o infrarroja; con el fin de proteger a las personas y los bienes materiales (Oreja, 2019).

Tabla 2.8 Comparativa de detectores de incendios

**Sensor de flama KY-026**

**Sensor MQ-135**



Características	Descripción	Características	Descripción
<b>Voltaje de operación</b>	3.3 a 5.5 VDC	<b>Voltaje de operación</b>	5 VDC
<b>Tipo de detección</b>	Luz infrarroja	<b>Tipo de detección</b>	Humo, alcohol y CO2
<b>Rango de detección</b>	760 a 1100 ppm	<b>Rango de detección</b>	10 a 1000 ppm
<b>Precio</b>	2.50 dólares	<b>Precio</b>	4.50 dólares
<b>Consumo de corriente</b>	10 mA	<b>Consumo de corriente</b>	150 mA

Fuente: (Anubhav, 2020)

**Conclusión**

Se seleccionó al sensor MQ-135 por sus características fundamentales para el presente prototipo detalladas en la Tabla 2.8 y al detectar humo a comparación de los sensores de luz infrarroja cuenta con un mayor rango de detección.

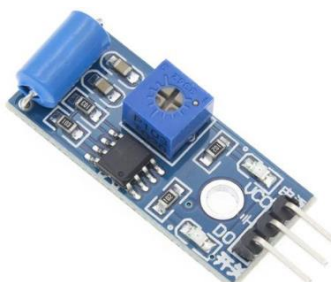
**2.4.8 Sensor de vibración**

Un sensor de vibración es un dispositivo que reacciona a movimientos bruscos, golpes o vibraciones. Su mecanismo está compuesto de un cilindro en el cual se encuentran dos contactos; uno de estos está unido a una varilla metálica ubicada en el centro del cilindro y el otro contacto se encuentra al rededor del cilindro en forma de muelle. En el caso de detectar una vibración genera una señal digital la cual puede ser procesada por un microcontrolador (Llamas, 2016).



Tabla 2.9 Características sensor de vibración

**Sensor de vibración SW-420**



Características	Descripción
Voltaje de operación	3.3 a 5 VDC
Señal de salida	Digital
Consumo de corriente	15 mA
Precio	3 dólares

Fuente: (Aguilar, 2017)

Para el control de estado de las ventanas en el presente prototipo se eligió el sensor de vibración SW-420 descrito en la Tabla 2.9, el cual se activa ante la presencia de golpes fuertes en las ventanas de la vivienda en caso de algún robo. Además, cuenta con un potenciómetro para variar su sensibilidad y su precio es muy accesible.

**2.4.9 Cámara de video**

Cámaras de red o IP es un dispositivo electrónico capaz de captar y transmitir una señal de audio y video digital mediante una red IP estándar hacia otros dispositivos de red, como una PC, un teléfono inteligente. Mediante el uso de una dirección IP, los usuarios puede visualizar, almacenar y gestionar video de forma local o remota (Boza, 2017).

Tabla 2.10 Comparativas cámaras de video

**ESP32-CAM**

**Cámara Wi-Fi**



Características	Descripción	Características	Descripción
-----------------	-------------	-----------------	-------------

<b>Voltaje de operación</b>	5 VDC	<b>Voltaje de operación</b>	3.3 VDC
<b>Tipo de conexión</b>	Wi-Fi y Bluetooth	<b>Tipo de conexión</b>	No tiene
<b>Resolución</b>	2 megapíxeles	<b>Resolución</b>	5 megapíxeles
<b>Puerto MicroSD</b>	Hasta 4 G	<b>Puerto MicroSD</b>	No tiene
<b>Precio</b>	15 dólares	<b>Precio</b>	25 dólares
<b>Consumo de corriente</b>	150 mA	<b>Consumo de corriente</b>	250 mA

Fuente: (Joy-it, 2020)

## Conclusión

Se optó por la cámara ESP32-CAM, siendo la más adecuada para el presente prototipo por sus características detalladas en la Tabla 2.10 además de ser compatible con la tarjeta de control que se utilizó, su precio es accesible lo cual permite el uso en varios proyectos enfocados a la seguridad personal.

### 2.4.10 Motores para persianas

El convertidor electromecánico, o motor paso a paso, permite traducir la información eléctrica en energía mecánica e información posicional. Está formado por un rotor con varios polos y un estator con bobinados denominados fases. Su funcionamiento es síncrono y la alimentación cíclica de sus fases debe originar en cambio de configuración un giro elemental del rotor, constante, llamado paso (Fonseca & Soria, 2020).

Tabla 11 Comparativa motores para persianas

<b>Motor paso a paso 28BYJ-48</b>		<b>Motor paso a paso nema 17</b>	
			
<b>Características</b>	<b>Descripción</b>	<b>Características</b>	<b>Descripción</b>
<b>Voltaje de operación</b>	5 o 12 VDC	<b>Voltaje de operación</b>	12 VDC
<b>Torque</b>	3.5 kg/cm	<b>Torque</b>	3.2 kg/cm
<b>Peso</b>	41 gramos	<b>Peso</b>	35 gramos
<b>Precio</b>	7 dólares	<b>Precio</b>	16 dólares
<b>Consumo de corriente</b>	55 mA	<b>Consumo de corriente</b>	350 mA

Fuente: (Game, 2020)

## Conclusión

Para el presente proyecto se seleccionó el motor paso a paso 28BYJ-48 descrito en la Tabla 2.11, el cual es encargado de levantar o bajar las persianas de acuerdo a las necesidades del usuario, por su precio y su placa para conexión con las tarjetas ESP32 Y ESP8266 fue apto para la implementación en el sistema domótico.

### 2.4.11 Sistema de alarma

Un sistema de alarma consiste en la instalación de una serie de equipos electrónicos dentro del hogar, considerados estratégicos desde el punto de vista de la seguridad y que están conectados hacia la central de monitoreo. Estos dispositivos pueden ser sensores de movimiento, contactos magnéticos, detectores de humo, etc (Jurado, 2018).

En la tabla 2.12 se describe las características del dispositivo electrónico que se utilizó para el sistema de alarma del prototipo domótico, que en conjunto con los sensores instalados notificaran a través de la aplicación móvil si existe algún problema de seguridad dentro del hogar.

Tabla 12 Características de alarma del sistema

#### Sirena tipo cuerno



Características	Descripción
Voltaje de operación	5 a 12 VDC
Señal de salida	Digital
Consumo de corriente	90 mA
Precio	8 dólares

Fuente: (Game, 2020)

### 2.4.12 Módulo GSM

El sistema global de comunicaciones móviles (GSM) nace en el siglo XXI, el estándar denominado segunda generación (2G) usa las bandas de frecuencia de 900 MHz y 1 800 MHz que permite transmisiones de voz y de datos digitales de volumen bajo tales como: servicio de mensajes cortos (SMS) o servicio de mensajes multimedia (MMS), con un rendimiento máximo de 9,6 kbps (López C. , 2021).

En la Tabla 2.13 se describe las características del módulo GSM que se utilizó para el envío de información del sistema domótico emitido por el sistema de seguridad a través del servicio de mensajes cortos (SMS).

Tabla 13 Características del módulo GSM SIM800L

### Módulo GSM SIM800L



Características	Descripción
Voltaje de operación	3.3 a 5 VDC
Interfaz	Serial UART
Velocidad de transmisión	1200bps hasta 115200bps
Precio	12 dólares
Frecuencia	Quad- Banda 850/900/1800/1900MHz
Consumo de corriente	1 A

Fuente: (Megatrónica, 2022)

#### 2.4.13 ESP8266

Es un módulo de desarrollo de Firmware, basada en el módulo Wi-Fi ESP8266, integra GPIO, I2C, PWM, 1-Wire y ADC en una sola tarjeta. Además, Posee una API avanzada para el control de entradas y salidas, lo que puede reducir drásticamente el trabajo para configurar y manipular hardware (Jurado, 2018). También, el código de programación se lo puede realizar con la IDE de Arduino, posee las siguientes características:

- Voltaje de funcionamiento 3.3 V.
- Procesador Tensilica L106 32 bit.
- 17 puertos GPIOs.
- Velocidad del procesador 80 – 160 MHz.
- Consumo de corriente 10 – 170 mA.
- Wi-Fi: 802.11 b/g/n (HT20)



Fig. 23 Módulo ESP8266  
Fuente: (Albert, 2020)

En la Figura 2.3 se aprecia la tarjeta ESP8266 que se utilizó para el control de sensores ubicados en la mayor parte de espacios físicos dentro de la vivienda debido que en estos espacios no existen muchas variables a controlar y su precio es mucho más económico a comparación con la tarjeta ESP32.

#### 2.4.14 ESP32

Creado por Espressif Systems, ESP32 es una tarjeta de control de bajo consumo y bajo costo conformando por un chip System On Chip (SoC) con Wi-Fi y Bluetooth. En su interior, consta de un microprocesador Tensilica Xtensa LX6 de doble núcleo o de un solo núcleo (Bruno, 2019).

Diseñado para dispositivos móviles; tanto en las aplicaciones de electrónica, y del Internet de las cosas (IoT), ESP32 logra un consumo de energía ultra bajo a través de funciones de ahorro de energía. Además, cuenta con algunas características principales:

- Procesador principal: Tensilica Xtensa LX6 de 32 bits.
- Voltaje de funcionamiento 3.3 V.
- Wi-Fi: 802.11 b / g / n / e / i.
- 3 interfaces UART.
- 25 pines GPIOs.
- Consumo de corriente 80 a 180 mA (Bruno, 2019).

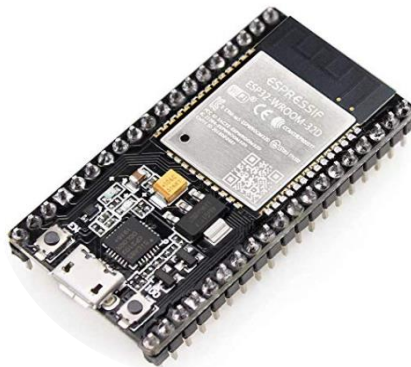


Fig. 24 Módulo ESP32.  
Fuente: (Herranz, 2019)

En la Figura 2.4 se aprecia la tarjeta ESP32, la misma que se utilizó para el control de los sensores ubicados dentro del espacio físico 3 correspondiente a la sala, debido a sus características de mayor cantidad de pines de entrada y salida es correcto para el funcionamiento en esta área.

#### 2.4.15 Tarjeta de control Raspberry Pi 3 Modelo B+

Es una tarjeta de control similar a un ordenador que consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Fue lanzado en 2006 por la Fundación Raspberry Pi con el objeto de estimular la enseñanza de la informática es fácil de instalar en la mayoría del software Linux, por lo tanto, permite la codificación en varios lenguajes Python y C++ (Raspberrypi, 2019).

Tabla 14 Características tarjeta de control Raspberry Pi 3 modelo B+

#### Raspberry Pi 3 Modelo B+



Características	Descripción
<b>Procesador</b>	Broadcom BCM2837B0, Cortex-A53 SoC de 64 bits a 1,4 GHz.
<b>Memoria</b>	1GB LPDDR2 SDRAM.
<b>Conectividad</b>	2,4 GHz y 5 GHz IEEE 802.11.b / g / n / ac inalámbrico, LAN, Bluetooth 4.2 y Gigabit Ethernet sobre USB 2.0.
<b>Acceso</b>	4 puertos USB 2.0, encabezado GPIO de 40 pines.
<b>Video y sonido</b>	1 x HDMI de tamaño completo, puerto de visualización MIPI DSI, puerto de cámara MIPI CSI.
<b>Multimedia</b>	Salida estéreo de 4 polos y puerto de video.
<b>Consumo de corriente</b>	350 mA a 2 A
<b>Puerto Micro SD</b>	formato Micro SD para cargar el sistema operativo y almacenamiento de datos.

Fuente: (Raspberrypi, 2019)

La tabla 2.14 se puede observar la placa Raspberry pi, que se usó como controlador principal del sistema de control domótico, ya que cuenta con varias especificaciones técnicas que son muy útiles como: resoluciones para video, puertos de memoria micro SD, puerto para cable de internet, puertos USB, puerto HDMI, etc.

#### 2.4.16 Sistema de respaldo de energía UPS

Un sistema de Alimentación Ininterrumpida (SAI), también conocido en el mercado como Uninterruptible Power Supply (UPS), es un dispositivo electrónico capaz de asegurar y mantener la alimentación eléctrica, de forma ininterrumpida y durante un tiempo delimitado de acuerdo a su capacidad de almacenamiento (Quispe, 2019).

Tabla 15 Características de la unidad de respaldo de energía UPS R-UPR 758

#### UPS R-UPR 758



Características	Descripción
<b>Voltaje de operación</b>	120 VAC
<b>Corriente máxima</b>	5.6 Amperios
<b>Capacidad</b>	750/375 Watts
<b>Tiempo de transferencia</b>	2 milisegundos
<b>Tipo de batería interna</b>	12 VDC/7AH
<b>Tiempo de recarga</b>	4hr a 90% después de completar la carga
<b>Tiempo de respaldo</b>	30 minutos

Fuente: (Anubhav, 2020)

La tabla 2.15 describe el tipo de UPS que se instaló como respaldo de energía para el sistema domótico por sus características de capacidad de almacenamiento y tiempo de respaldo de energía en el caso que la alimentación principal del sistema sea interrumpida.

### 2.4.17 Tecnología de comunicación

Mediante la tecnología de comunicación Wi-Fi se logró transferir la información almacenada en las tarjetas de control ESP8266 y ESP32 hacia la central de control Raspberry pi, donde se localiza la programación del sistema la cual ejecuta acciones dependiendo de la información recogida por los sensores y actuadores instalados en el sistema domótico.

### 2.5 Esquema del sistema domótico

En la Figura 2.5 se aprecia el esquema general del sistema domótico, su conexión con los 5 espacios físicos, su interacción con la aplicación móvil y con la central de control Raspberry Pi, el sistema cuenta con una conexión serial para la comunicación GSM que en conjunto con la alarma que se instaló en la parte central del hogar notificarán al usuario la existencia de algún problema captado por los sensores ubicados dentro de la casa.

Además, se recuerda que los elementos electrónicos del sistema domótico están conectados con cables a las tarjetas de control ESP32 y ESP8266 y estas, mediante conexión inalámbrica Wi-Fi a la central de control Raspberry Pi.

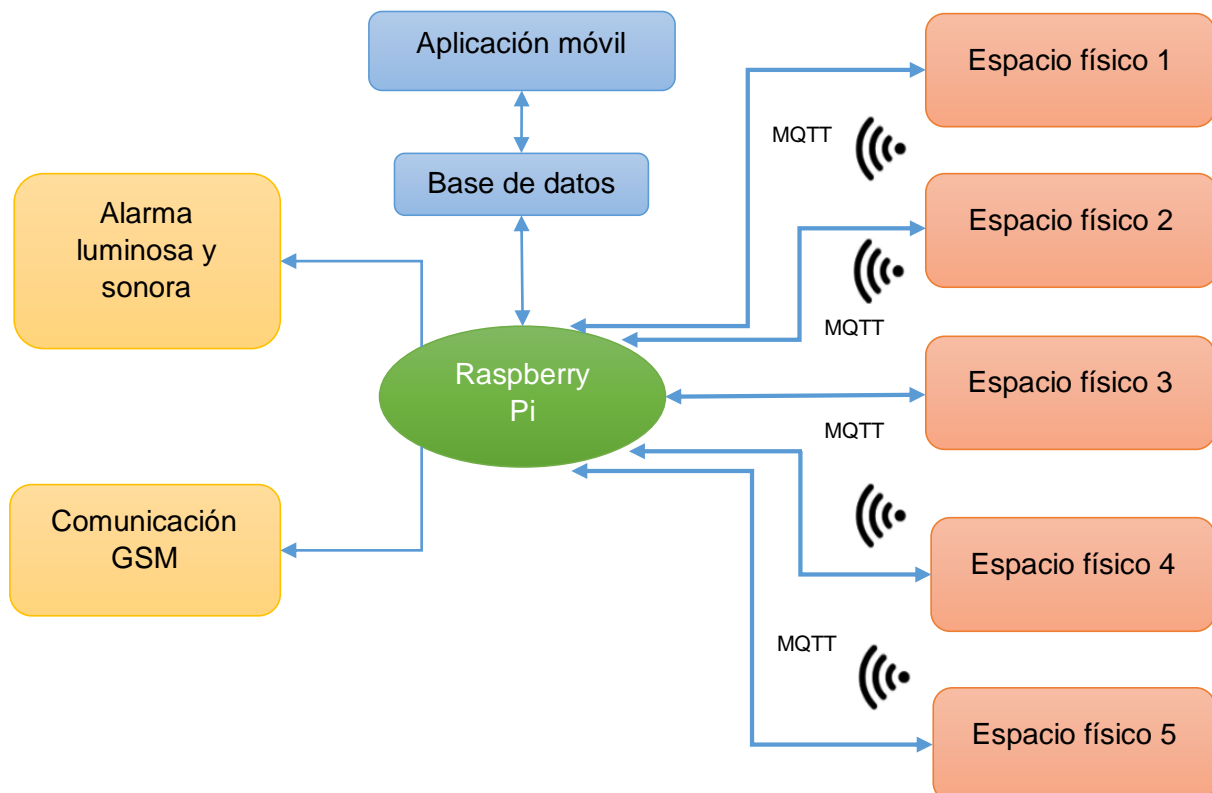


Fig. 25 Esquema general del sistema domótico  
Fuente: Autor



El esquema del sistema domótico se lo realizó de acuerdo a otros prototipos ya existentes, que de acuerdo al análisis realizado en el capítulo 1 se optó por una arquitectura mixta, dentro de esta arquitectura existen varios elementos de control encargados de adquirir y procesar información con el sustento de los sensores y controladores del sistema.

### 2.5.1 Esquema de conexión del espacio físico 1

En la Figura 2.6 se aprecian las soluciones domóticas que se aplicaron en el espacio físico 1 (cocina) correspondientes a las siguientes: control de luminarias, monitoreo de estado de ventanas, monitoreo de fuga de gas, presencia de fuego y monitoreo de fuga de agua.

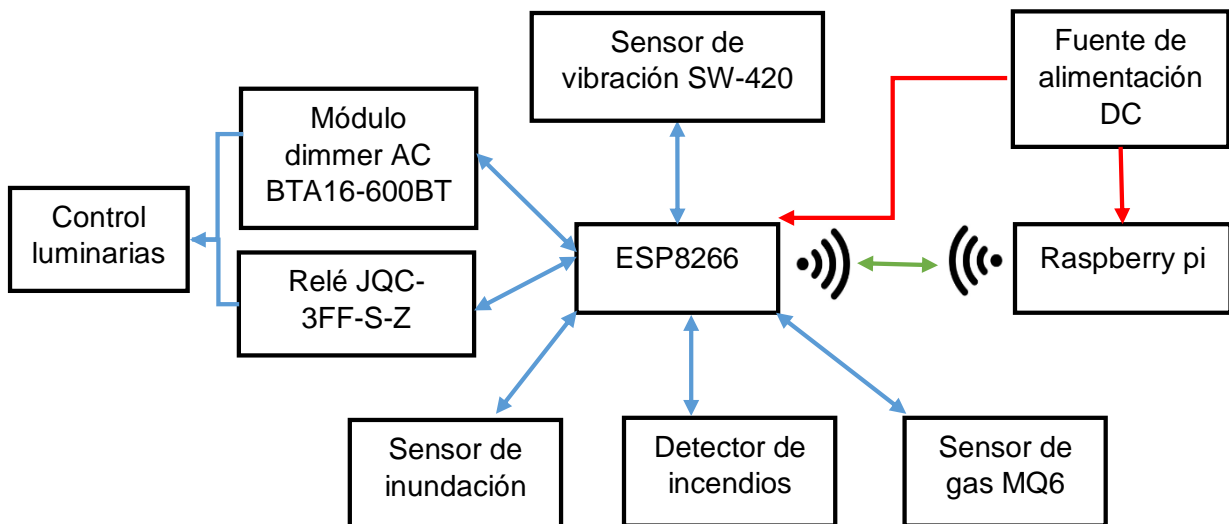


Fig. 26 Esquema de conexión de componentes domóticos.  
Fuente: (Autor,2021)

### 2.5.2 Esquema de conexión del espacio físico 2

La figura 2.7 muestra el espacio físico 2 correspondiente a los dormitorios, donde se realizó las conexiones de las siguientes aplicaciones domóticas: control de luminarias, monitoreo de estado de ventanas, control de persianas.

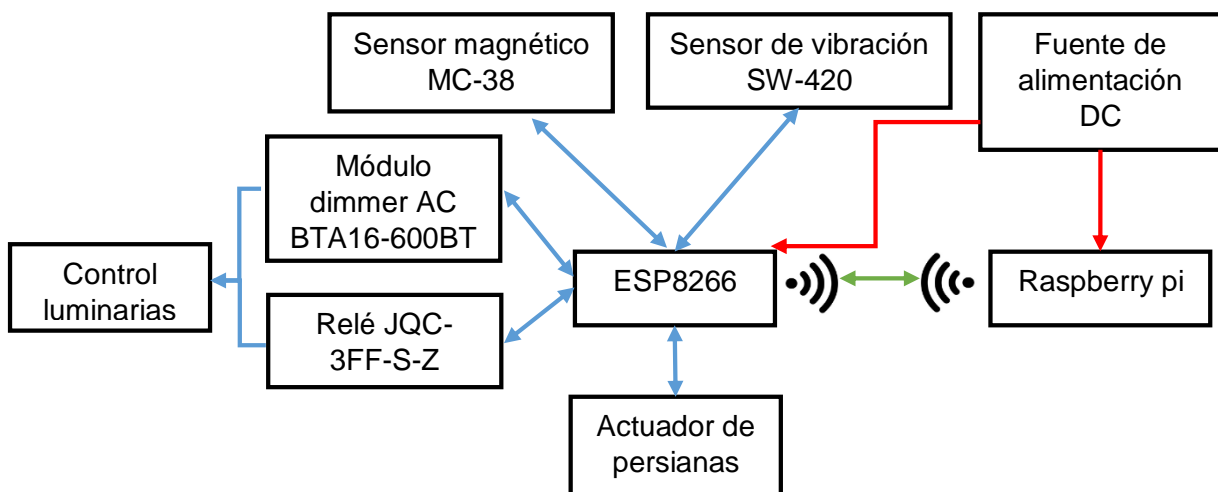


Fig. 27 Esquema de conexión de componentes domóticos  
Fuente: (Autor,2021)

### 2.5.3 Esquema de conexión del espacio físico 3

La Figura 2.8 presenta el diagrama de conexión del espacio físico 3 que corresponde a la sala, donde se instaló los siguientes elementos domóticos: control de luminarias, monitoreo de estado de ventanas, monitoreo de acceso a la vivienda, control de persianas, control de equipos de entretenimiento, control de temperatura, sistema de seguridad y monitoreo con cámara de video.

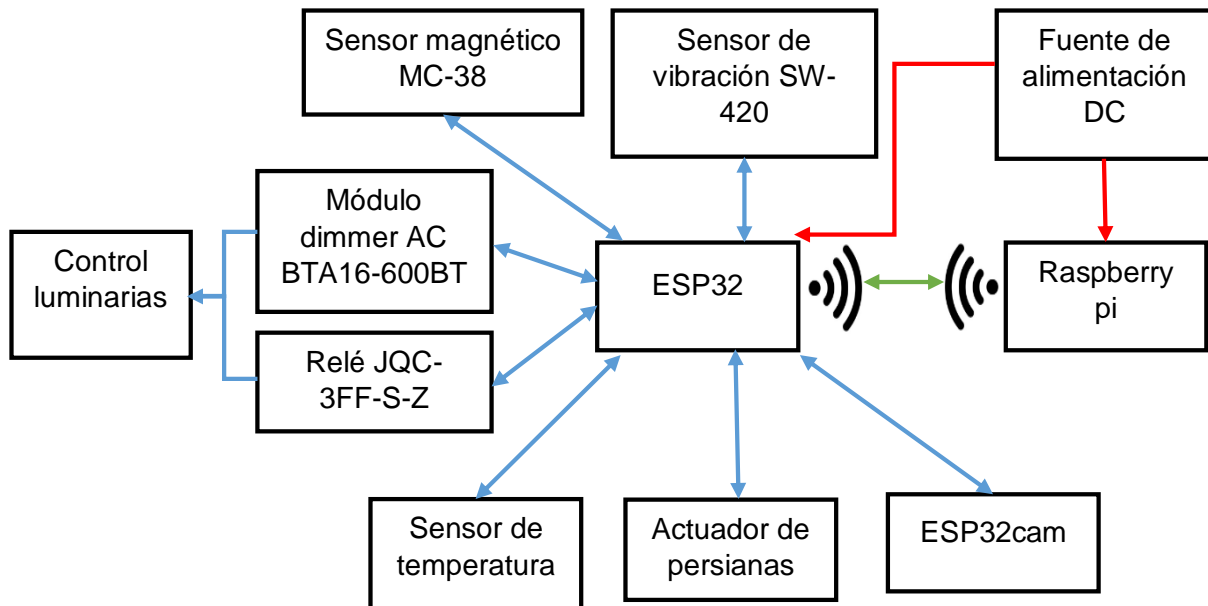


Fig. 28 Esquema de conexión de componentes domóticos  
Fuente: (Autor,2021)

### 2.5.4 Esquema de conexión del espacio físico 4

En la Figura 2.9 se aprecia el esquema de los diferentes elementos ubicados dentro del espacio físico 4 correspondiente al comedor, donde se instaló los siguientes componentes: control de luminarias y monitoreo de estado de ventanas.

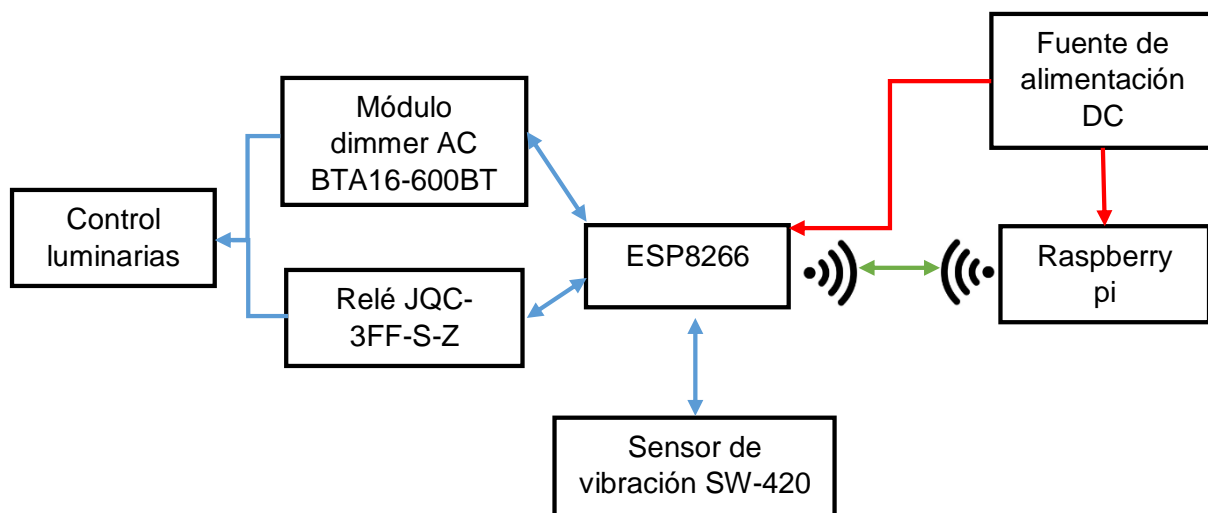


Fig. 29 Esquema de conexión de componentes domóticos  
Fuente: (Autor,2021)

### 2.5.5 Esquema de conexión del espacio físico 5

La Figura 2.10 muestra el esquema de conexión que se realizó en el espacio físico 5 correspondiente al baño, donde se aplicó las siguientes soluciones domóticas; control de luminarias apagado y encendido, monitoreo de estado de ventanas y monitoreo de inundaciones.

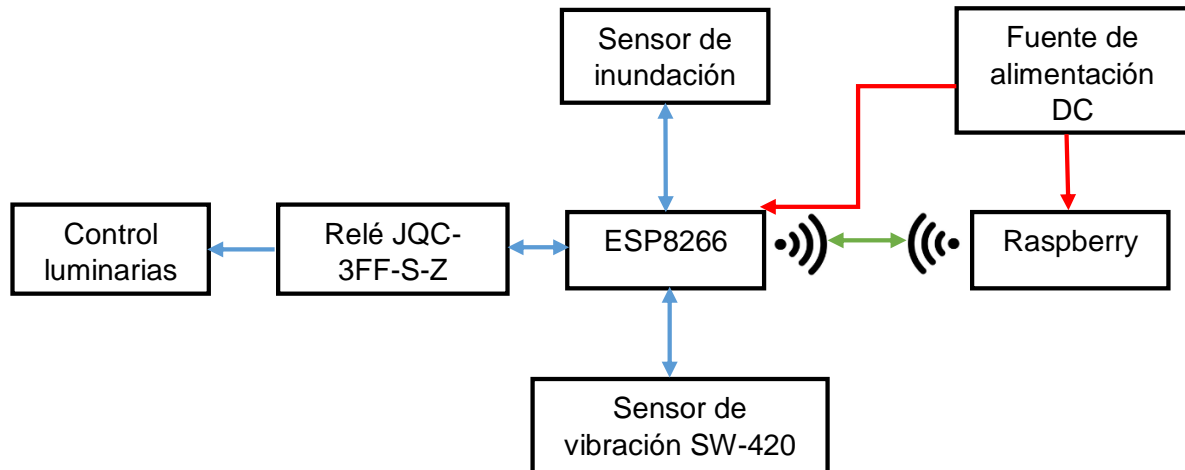


Fig. 30 Esquema de conexión de componentes domóticos.  
Fuente: (Autor,2021)

## 2.6 Software de programación

Para el siguiente trabajo se utilizó diferentes softwares libres de programación que hacen posible la ejecución de tareas específicas para el control del sistema domóticos. A continuación, se especifica el uso del software del sistema operativo, software de programación y software para la aplicación móvil.

### 2.6.1 Raspberry Pi OS

El sistema operativo de Raspberry Pi se denomina Raspberry Pi OS, anteriormente llamado Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre. Usa un entorno de escritorio ligero y rápido (LXDE) y Chromiun como navegador web (Anubhav, 2020).

Para el presente trabajo se utilizó una Raspberry Pi como controlador principal del sistema domótico que gracias a sus características se optó por esta tarjeta específica para el prototipo domótico. Para un desempeño adecuado de la tarjeta de control se instaló el sistema operativo Raspberry Pi OS.

### 2.6.2 Node-RED

El software de código abierto Node-RED es una herramienta de programación visual mediante la cual los desarrolladores pueden programar sin escribir líneas de código, en vista

de que su funcionamiento es por medio de nodos, estos se organizan en flujos o flows que agrupan nodos que se conectan entre ellos. En la Figura 2.11 se visualiza como el Sistema Operativo Raspbian trae integrado Node-RED (Moral, 2021).

Node-RED está creado a partir de NodeJS y la librería de JavaScript D3.js. NodeJS es un software muy potente que hace viable la programación del lado del servidor en JavaScript, lo que es una gran ayuda al ser fiable y sobre todo escalable. Es un entorno que se ha diseñado para desarrollar aplicaciones escalables. Una de sus principales ventajas es que está para poder soportar múltiples conexiones simultáneas (Moral, 2021).

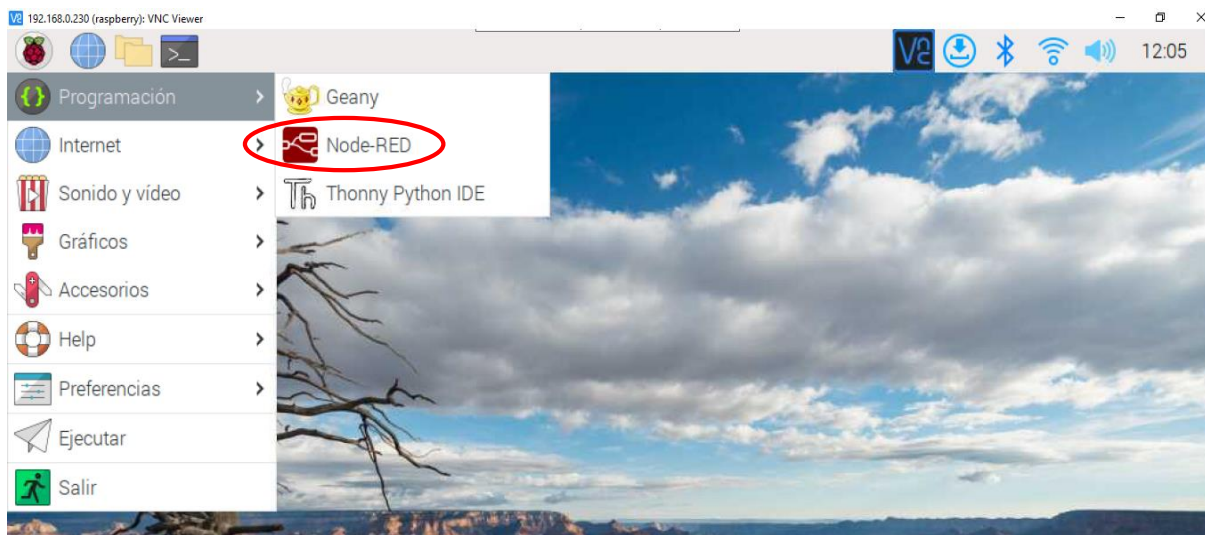


Fig. 31 Software Node-RED  
Fuente: (Autor,2021)

Para integrar el hardware del sistema domótico con la placa de control Raspberry Pi en el presente trabajo se utilizó el software de programación visual Node-RED, el cual es de gran uso en el campo de las aplicaciones del internet de las cosas. Además, este software viene incorporado en el sistema operativo Raspbian de Raspberry.

### 2.6.3 Arduino IDE

El entorno de desarrollo integrado, (Integrated Development Enviroment, IDE) es un conjunto de herramientas que permiten a los programadores desarrollar, editar, compilar y grabar todo el código necesario en una placa de desarrollo, satisfaciendo las necesidades de los usuarios (Fonseca & Soria, 2020).

Este entorno es la principal herramienta que se utilizó en el presente prototipo para desarrollar la programación de los diferentes sensores y actuadores que se instalaron dentro de la vivienda en los diferentes espacios físicos se lo puede apreciar en la Figura 2.12. En efecto, la mayoría de los dispositivos electrónicos que se encuentran disponibles en el mercado son compatibles con esta IDE de Arduino.

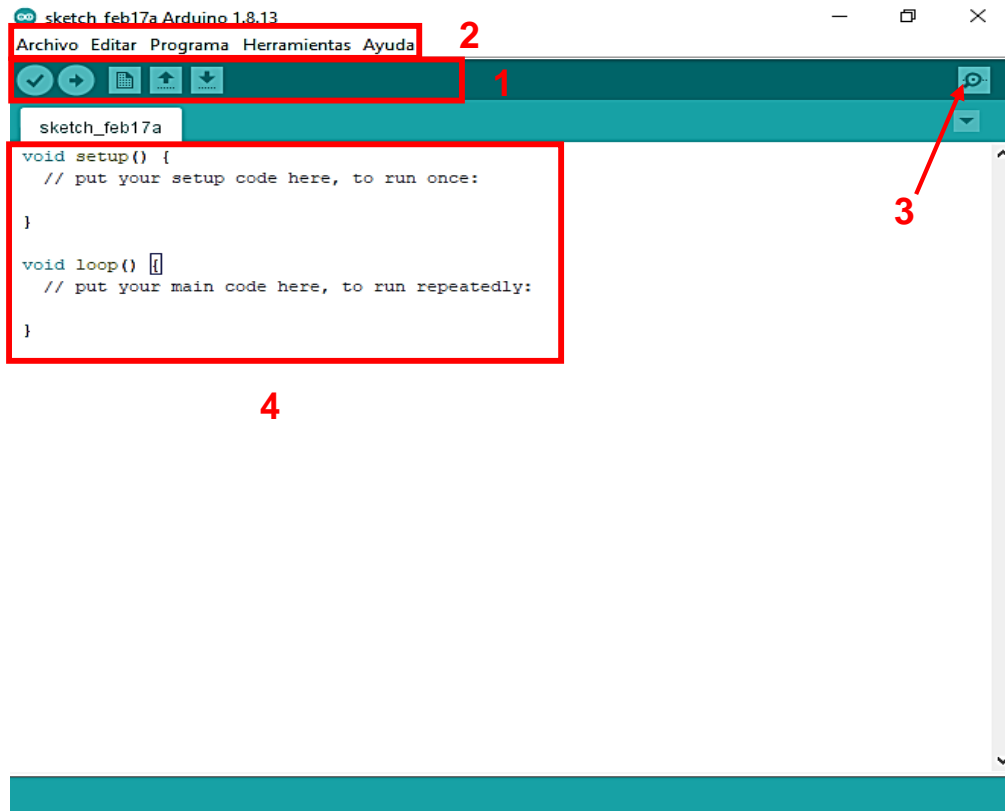


Fig. 32 Interfaz de IDE de Arduino  
Fuente: (Autor,2023)

1. En primer lugar, en la esquina superior izquierda, tenemos los botones para comprobar que el código funciona correctamente y el más importante, la flecha que actúa como compilador y envía las sentencias al microcontrolador.
2. Por otro lado, en la parte superior está el menú, en el cuál, se pueden instalar librerías externas, seleccionar el puerto en el que se encuentra la placa y seleccionar el modelo de la placa que se va a utilizar.
3. En la esquina superior derecha, se encuentra el botón para desplegar la respuesta de la consola, también llamada monitor serial.
4. Para finalizar, en el centro se encuentra el espacio para escribir el código que haga funcionar el programa. Por defecto se generan las funciones void setup, para establecer las variables principales y todo lo que se quiera que se ejecute nada más enviar el programa, y void loop, la cual ejecutará el código que hay en su interior

## 2.6.4 Software Eagle

Eagle es un software de automatización del diseño electrónico (EDA) que permite a los usuarios diseñar placas de circuito impreso (PCB), conectar sin dificultades los diagramas esquemáticos, la colocación de componentes, el enrutamiento de las PCB y el contenido de una amplia biblioteca. Editor de esquemas fácil de usar en EAGLE Transforme sus ideas en realidad con la captura de esquemas.

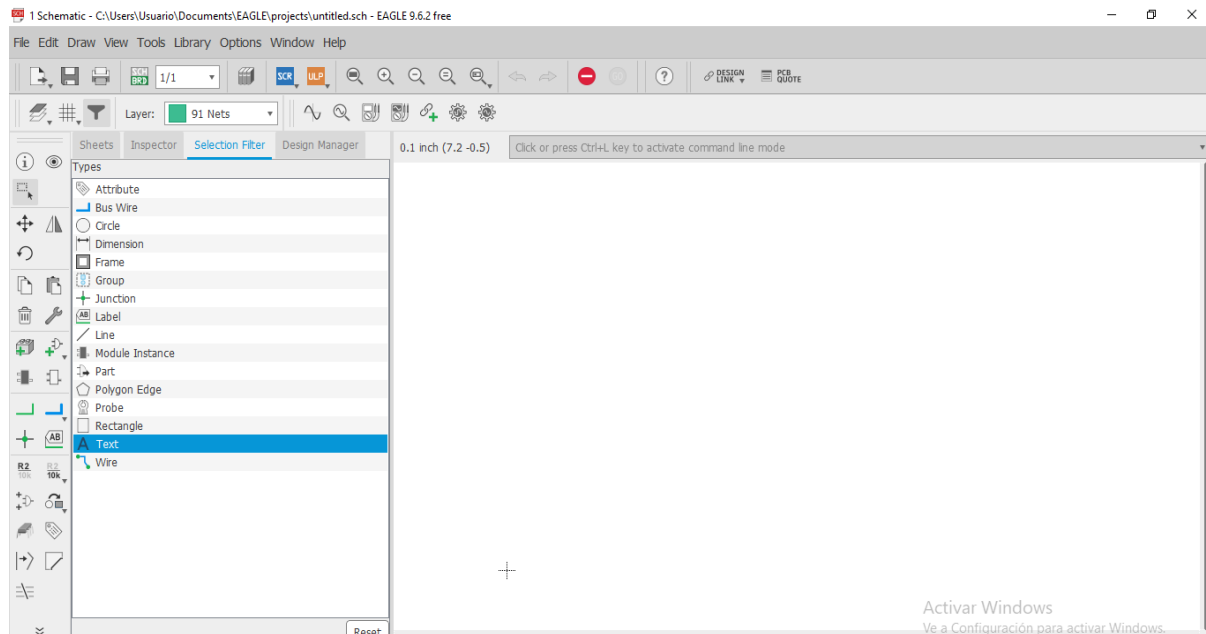


Fig. 33 Software de diseño EAGLE  
Fuente: (Autor,2021)

El software que se utilizó para el diseño de las placas electrónicas de cada uno de los espacios físicos fue EAGLE, en la Figura 2.13 se aprecia el espacio principal de trabajo donde se pueden desarrollar diversos diseños de placas PCB de acuerdo a las necesidades del usuario. Además, el software cuenta con una gran variedad de librerías.

## 2.6.5 Plataforma para desarrollo de aplicaciones móviles APP Inventor

App Inventor es una plataforma de desarrollo de aplicaciones móviles que surge de una idea conjunta del Instituto Tecnológico de Massachusetts y de un equipo de Google Education. Se trata de una herramienta web de desarrollo para iniciarse en el mundo de la programación (Torres, 2021).

La aplicación móvil para el presente trabajo se realizó en la plataforma de desarrollo de software libre APP Inventor, ya que permite el desarrollo de una gran variedad de funcionalidades. Además, gracias a su lenguaje de programación basado en bloques que a comparación de otras plataformas es mucho más fácil para programadores que estén empezando en la creación de aplicaciones.

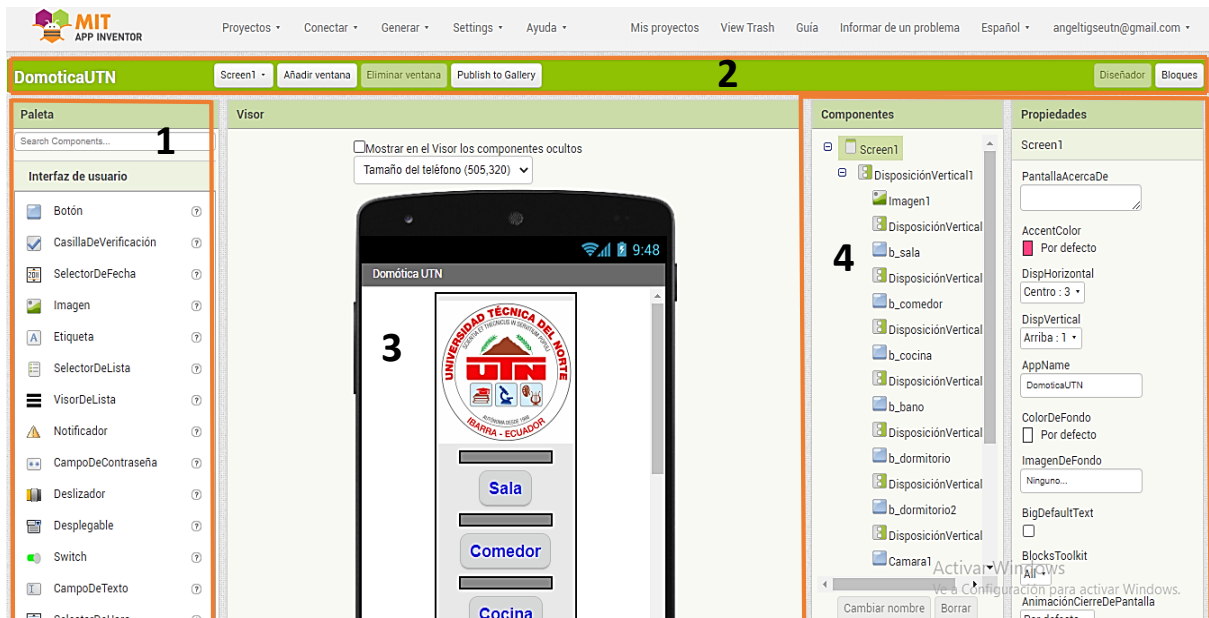


Fig. 34 Diseño de aplicación móvil.  
Fuente: (Autor,2021)

En la Figura 2.14 se puede apreciar la pantalla principal del entorno de desarrollo de App Inventor. Este software se utilizó para el desarrollo de la aplicación móvil con los componentes necesarios para el control del sistema domótico. A continuación, se describe las 4 partes más importantes que conforman el presente software.

- 1) En el lado izquierdo de la aplicación están ubicados los componentes que se pueden utilizar dentro del entorno de desarrollo, tales como; botones, sensores, la conectividad, el almacenamiento, etc.
- 2) La parte superior es la más importante aquí se encuentran dos iconos; el primero llamado bloques, donde se puede desarrollar el código de programación por bloques. El segundo llamado diseñador donde se realiza el diseño de la aplicación de acuerdo a las necesidades del usuario.
- 3) En la parte central, se encuentra el sitio de trabajo, donde se arrastran los elementos que se van a utilizar.
- 4) En la parte derecha se encuentran los componentes que se van utilizando en el espacio de trabajo y las propiedades de cada uno de estos.

## 2.6.6 Plataforma Firebase

Es una plataforma digital que facilita el desarrollo de aplicaciones web y móviles, se la puede encontrar en la nube y está útil para los sistemas Android, ios y web entre otras tecnologías. Fue creada en 2011 pasó a ser parte de Google en 2014, comenzando como una base de datos en tiempo real para el envío y recepción de datos (López S. , 2020).

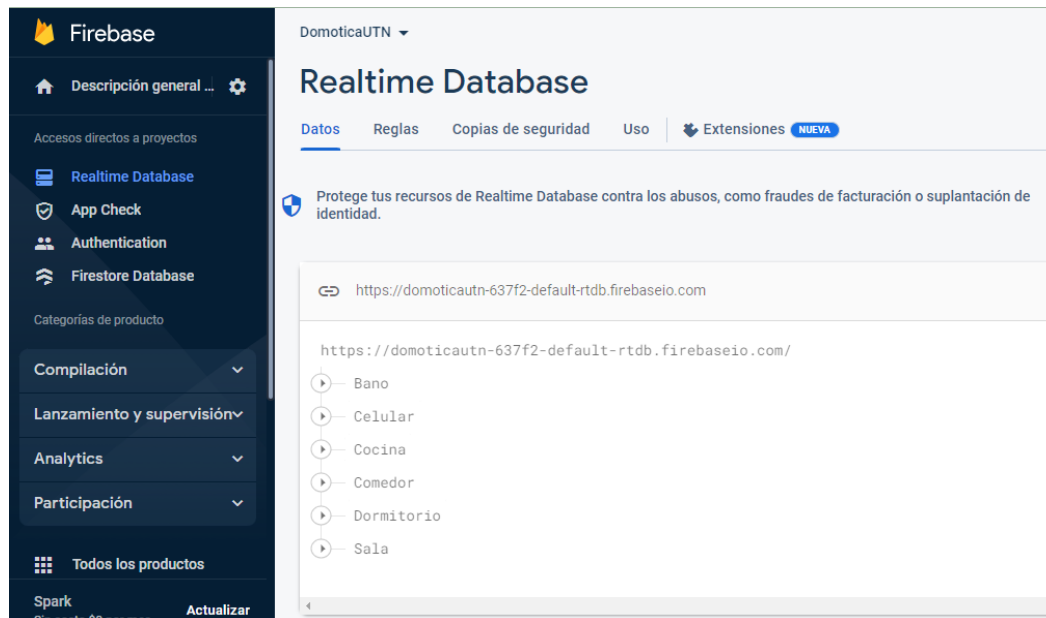


Fig. 35 Plataforma Firebase  
Fuente: (Autor,2021)

La Figura 2.15 muestra una de las herramientas más destacadas de Firebase llamada Realtime Database, esta permitió el almacenamiento de datos e información de la aplicación domótica en tiempo real, por ser gratuita y su rápida sincronización de datos se hizo uso de esta plataforma digital para el desarrollo del presente prototipo domótico.

## Conclusión

Una vez concluido este capítulo se analizó la comparativa de algunos de los elementos domóticos y se optó por el más adecuado para el presente trabajo, teniendo en cuenta sus características y su disponibilidad en el mercado. Además, se investigó algunas funciones que dispone la plataforma de desarrollo de aplicaciones App Inventor para aplicaciones domóticas.



## Capítulo 3

### Implementación y pruebas de funcionamiento del prototipo domótico en una vivienda

En el presente capítulo se realizó la implementación del sistema domótico en una vivienda, detallando la configuración necesaria para establecer la comunicación entre la aplicación móvil para Android con la central de control Raspberry Pi por medio del software Node-RED. Además, se describió el proceso de diseño y elaboración de las placas electrónicas de cada una de las áreas donde se montaron los componentes domóticos. Por último, se llevó a cabo las pruebas de funcionamiento dentro de una vivienda y la evaluación de resultados.

#### 3.1 Implementación del prototipo

La implementación y las pruebas de funcionamiento del prototipo domótico se las realizó en una vivienda de una sola planta con un área de construcción de 55 m<sup>2</sup>, esta se encuentra ubicada en la provincia de Pichincha, Cantón Pedro Moncayo, ciudad de Tabacundo, con las siguientes coordenadas geográficas 0°02'28.1"N 78°14'56.4"W. En esta se instaló todos los componentes domóticos anteriormente mencionados distribuidos en varios lugares específicos dentro de la casa. A continuación, se detallan los diseños de cada placa electrónica y su fabricación para los 5 espacios físicos; cocina, dormitorios, sala, comedor y baño.

Además, el prototipo se implementó dentro de una maqueta, para el conocimiento de los estudiantes de la carrera de Electricidad, de la Universidad Técnica del Norte de la ciudad de Ibarra, Provincia de Imbabura.

##### 3.1.1 Diseño y fabricación de la placa electrónica de la cocina

En la Figura 3.1 se aprecia el diseño de la placa electrónica de la cocina, que se realizó en el software Eagle de acuerdo a los requerimientos domóticos especificados para esta área, dentro del diseño están los siguientes sensores y actuadores: sensor de inundación, un sensor de gas, un sensor de humo, un sensor de vibración; un relé de un canal y un dimmer para el control de luces; todos estos elementos fueron conectados a la tarjeta de control ESP8266.



Fig. 36 Diseño placa electrónica cocina  
Fuente: (Autor,2023)

La Figura 3.2 muestra las conexiones de la placa de circuito impreso (PCB) de la cocina. Posteriormente, se envió el diseño de la placa para su elaboración, el proceso de planchado sobre una baquelita de cobre, el perforado de la placa de acuerdo a los componentes electrónicos que se instaló aquí.

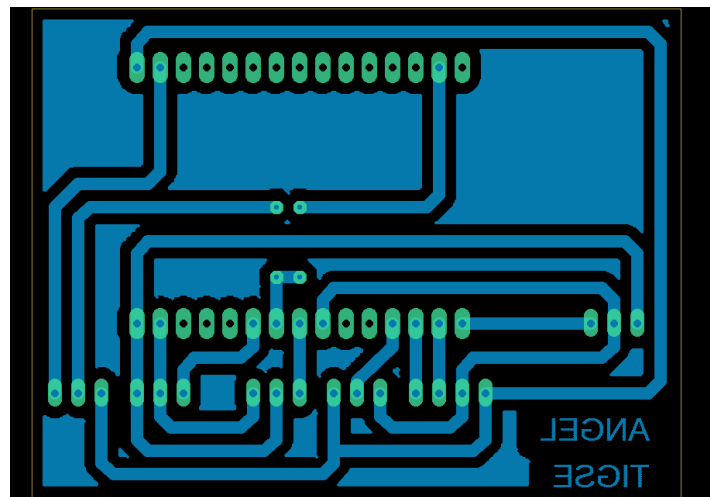


Fig. 37 Diseño PCB cocina  
Fuente: (Autor,2023)

Por último, se realizó el fijado y montaje de los elementos electrónicos sobre la placa de cobre correspondiente al espacio físico de la cocina, como se puede apreciar en la siguiente Figura 3.3.

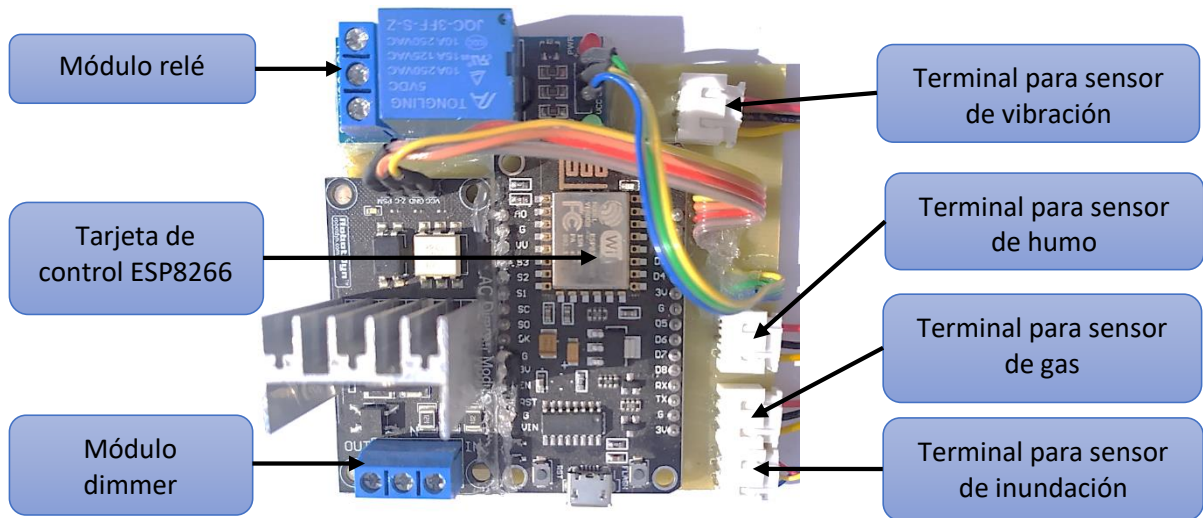


Fig. 38 Placa electrónica cocina  
Fuente: (Autor,2023)

### 3.1.2 Diseño y fabricación de la placa electrónica de los dormitorios

En la Figura 3.4 se aprecia el diseño de la placa electrónica de los dormitorios, que se realizó de acuerdo a los requerimientos domóticos especificados para esta área, dentro del diseño están los siguientes sensores y actuadores: un sensor magnético para ventanas, un motor para persianas, un sensor de vibración para ventanas, un relé para cargas y un dimmer para control de luces; todos estos elementos fueron conectados a la tarjeta de control ESP32.

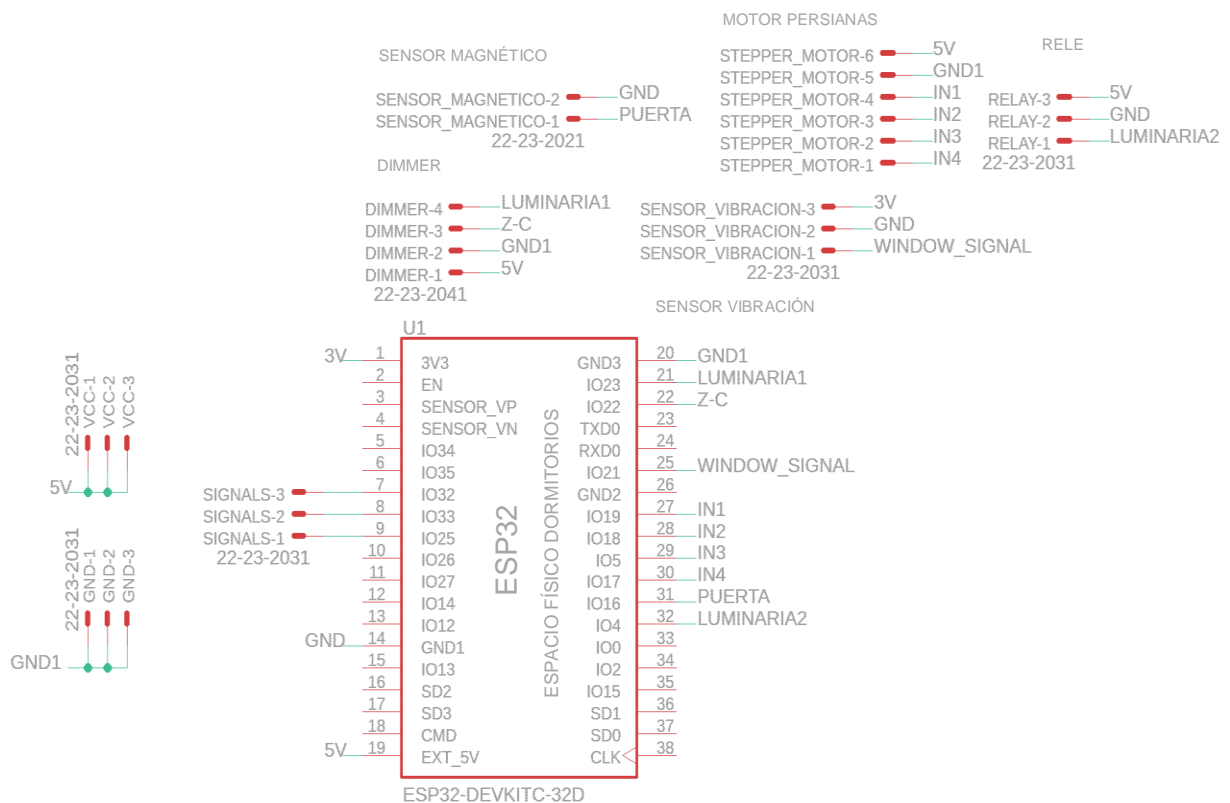


Fig. 39 Diseño placa electrónica dormitorios  
Fuente: (Autor,2023)

La Figura 3.5 muestra las conexiones de la placa de circuito impreso (PCB) de los dormitorios. Posteriormente, se envió el diseño de la placa para su elaboración, el proceso de grabado sobre una baquelita de cobre y el perforado de la placa de acuerdo a los componentes electrónicos que se instaló aquí.

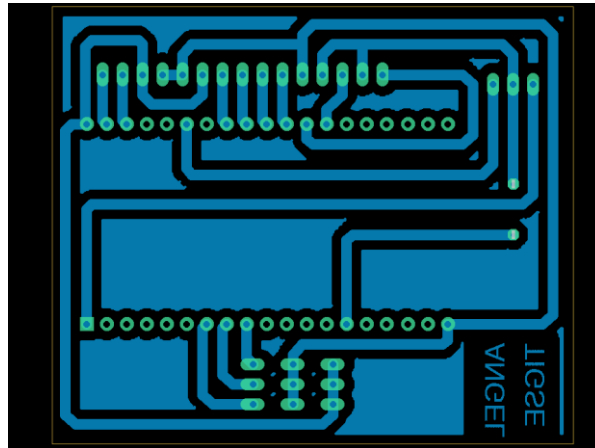


Fig. 40 Diseño PCB dormitorios  
Fuente: (Autor,2023)

Finalmente, se realizó montaje y soldadura de los elementos electrónicos sobre la placa de cobre correspondiente a los dormitorios como se puede apreciar en la siguiente Figura 3.6.

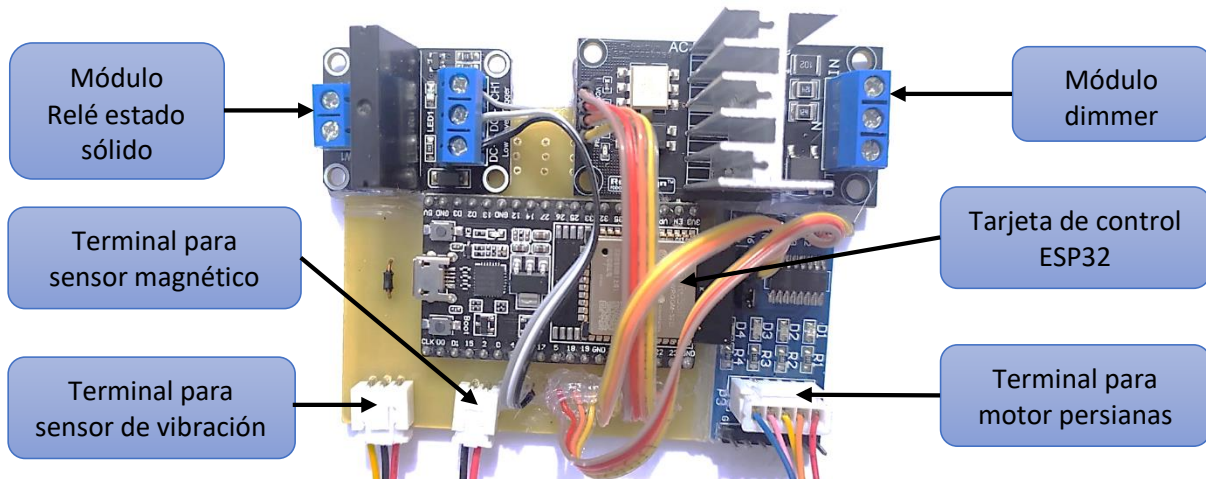


Fig. 41 Placa electrónica de los dormitorios  
Fuente: (Autor,2023)

### 3.1.3 Diseño y fabricación de la placa electrónica de la sala

En la Figura 3.7 se aprecia el diseño de la placa electrónica de la sala, que se realizó de acuerdo a los requerimientos domóticos especificados para esta área, dentro del esquema están los siguientes sensores y actuadores: dos sensores magnéticos para ventanas, un motor para persianas, dos sensores de vibración para ventanas, un sensor de temperatura, un relé de 4 canales y un dimmer para control de luces; todos estos elementos fueron conectados a la tarjeta de control ESP32.

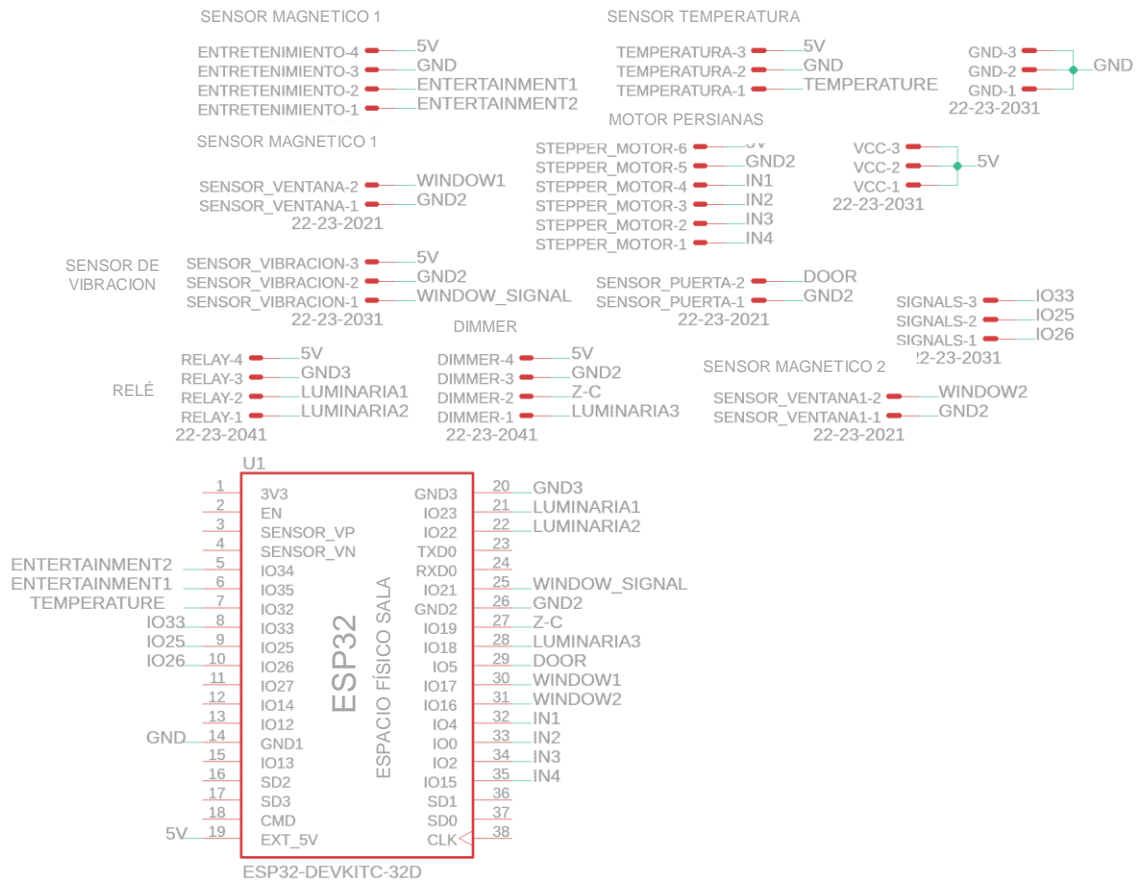


Fig. 42 Diseño placa electrónica de la sala  
Fuente: (Autor,2023)

La Figura 3.8 muestra las conexiones de la placa de circuito impreso (PCB) de la sala. Posteriormente, se envió el diseño de la placa para su elaboración, el proceso de grabado sobre una baquelita de cobre y el perforado de la placa de acuerdo a los componentes electrónicos que se instaló aquí.

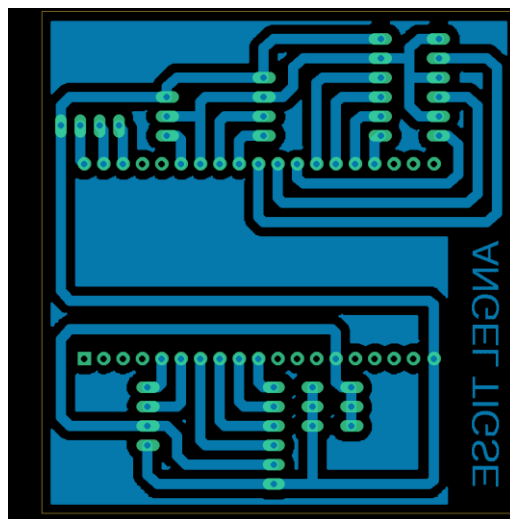


Fig. 43 Diseño PCB sala  
Fuente: (Autor,2023)

Finalmente se realizó el montaje y soldadura de los elementos electrónicos sobre la placa de cobre correspondiente a la sala como se puede observar en la siguiente Figura 3.9.

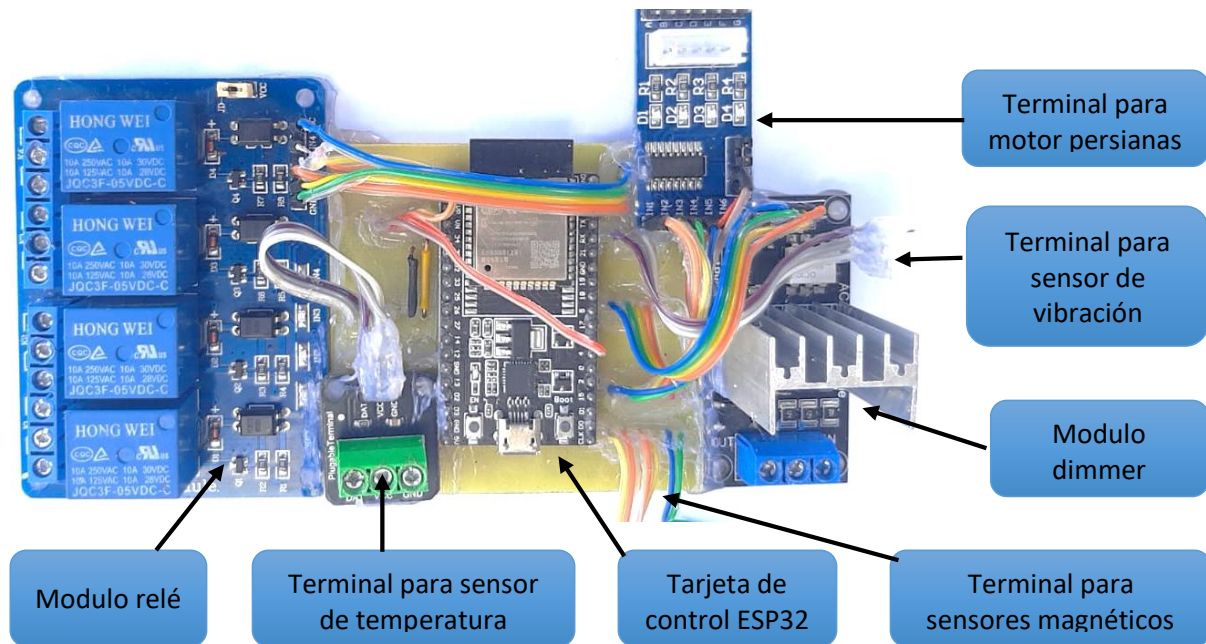


Fig. 44 Diseño placa electrónica de la sala  
Fuente: (Autor,2023)

### 3.1.4 Diseño y fabricación de la placa electrónica del comedor

Para el diseño de la placa electrónica del comedor se tomó en cuenta los siguientes sensores y actuadores: un sensor de vibración, un relé de 2 canales para cargas y un dimmer para control de luces; todos estos elementos fueron conectados a la tarjeta de control ESP8266 como se puede apreciar en la Figura 3.10.



Fig. 45 Diseño placa electrónica comedor  
Fuente: (Autor,2023)

La Figura 3.11 muestra las conexiones de la placa de circuito impreso (PCB) del comedor. Posteriormente, se envió el diseño de la placa para su elaboración, el proceso de grabado sobre una baquelita de cobre y el perforado de la placa de acuerdo a los componentes electrónicos que se instaló aquí.

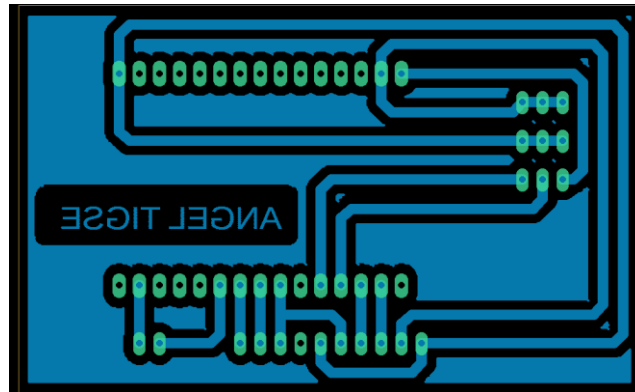


Fig. 46 Diseño PCB comedor  
Fuente: (Autor,2023)

Finalmente se realizó el montaje y soldadura de los elementos electrónicos sobre la placa de cobre correspondiente al comedor como se puede observar en la siguiente Figura 3.12.

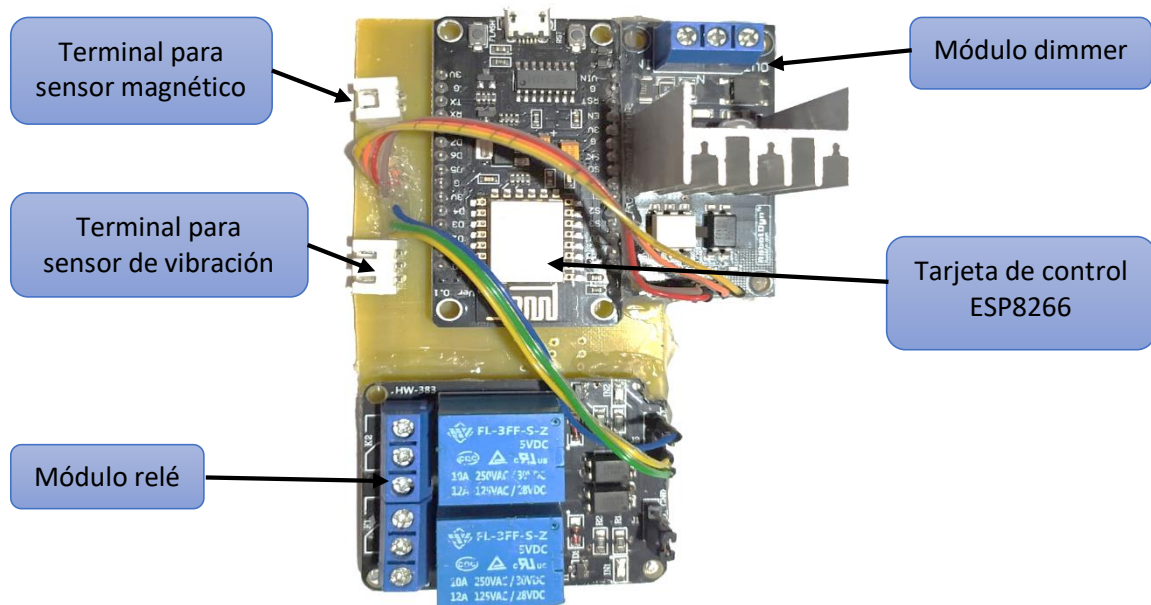


Fig. 47 Placa electrónica comedor  
Fuente: (Autor,2023)

### 3.1.5 Diseño y fabricación de la placa electrónica del baño

El diseño de la placa electrónica correspondiente al baño está conformado por los siguientes sensores y actuadores: un sensor de inundación, un sensor de vibración para ventanas y un relé para el control de luces; todos estos elementos fueron conectados a la tarjeta de control ESP8266 como se puede apreciar en la Figura 3.13.

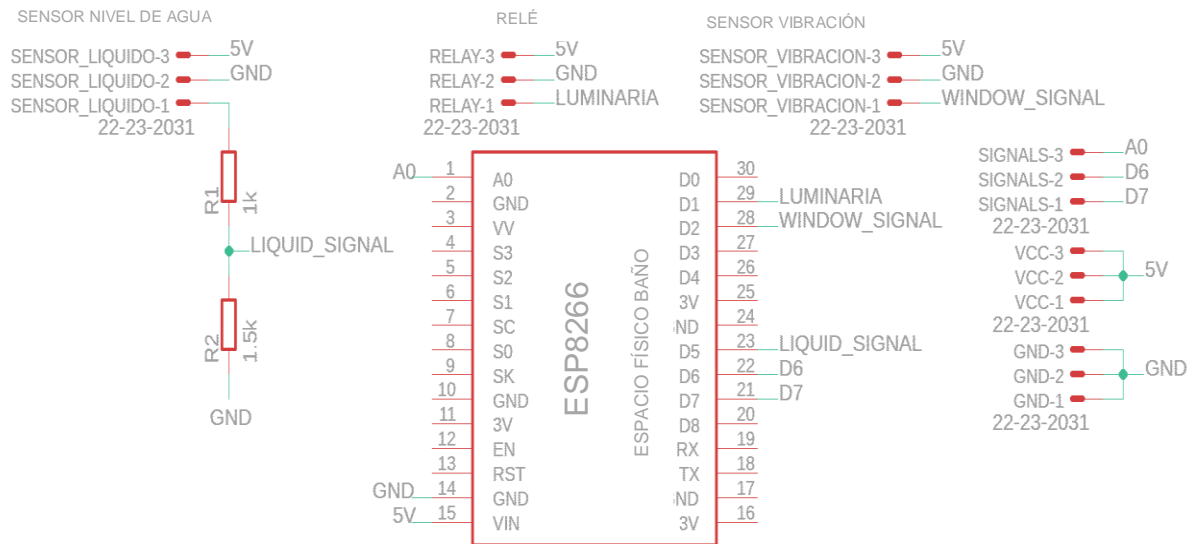


Fig. 48 Diseño placa electrónica baño  
Fuente: (Autor,2023)

En la Figura 3.14 se puede apreciar las conexiones de la placa de circuito impreso (PCB) del área del baño. Posteriormente, se envió el diseño de la placa para su elaboración, el proceso de planchado sobre una baquelita de cobre y el perforado de la placa de acuerdo a los componentes electrónicos que se instaló aquí.

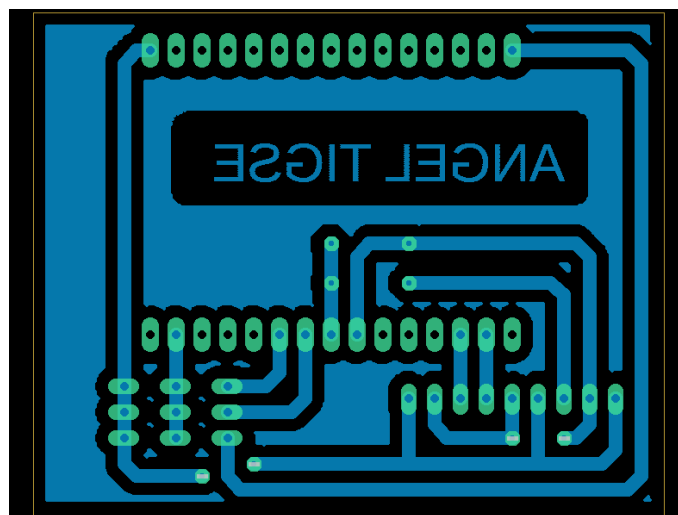


Fig. 49 Diseño PCB baño  
Fuente: (Autor,2023)

Finalmente se realizó el proceso de montaje y soldadura de los elementos electrónicos sobre la placa de cobre correspondiente al baño como se puede apreciar en la siguiente Figura 3.15.





Fig. 50 Placa electrónica baño  
Fuente: (Autor,2023)

### 3.2 Programación del sistema domótico

Para la programación de todos los componentes que comprenden el sistema domótico se utilizó los siguientes softwares; para la placa principal de control se instaló el sistema operativo Raspbian dentro de la Raspberry Pi 3 modelo B+, dentro de este sistema por defecto viene integrado el programa Node-Red el cual permitió realizar la comunicación entre la Raspberry y las placas de control ESP8266 y ESP32.

#### 3.2.1 Programación central de control Raspberry Pi

A la hora de llevar a cabo el presente proyecto surgió la necesidad de escoger un sistema operativo que proporcione soporte para todas las herramientas necesarias que se van a realizar. Raspbian es una distribución del sistema operativo GNU/Linux basado en Debian ideal para el hardware que se ha seleccionado una Raspberry Pi 3 modelo B+, la Figura 3.16 muestra el logo del sistema operativo Raspbian instalado en la tarjeta principal de control.



Fig. 51 Pantalla principal de Raspberry Pi  
Fuente: (Autor,2023)

El diagrama de flujo descrito en la Figura 3.17 explica el modo de funcionamiento del prototipo domótico, partiendo con la inicialización de la central de control Raspberry Pi 3, una vez establecido la conexión a Internet la central recibe los datos de las placas ESP32 y ESP8266, luego si existe la presencia de alguna alerta de seguridad envía un mensaje a la aplicación móvil o mediante la comunicación GSM alerta al usuario vía mensajes de texto al número que se encuentre establecido en la programación que se detalla en el anexo 1.

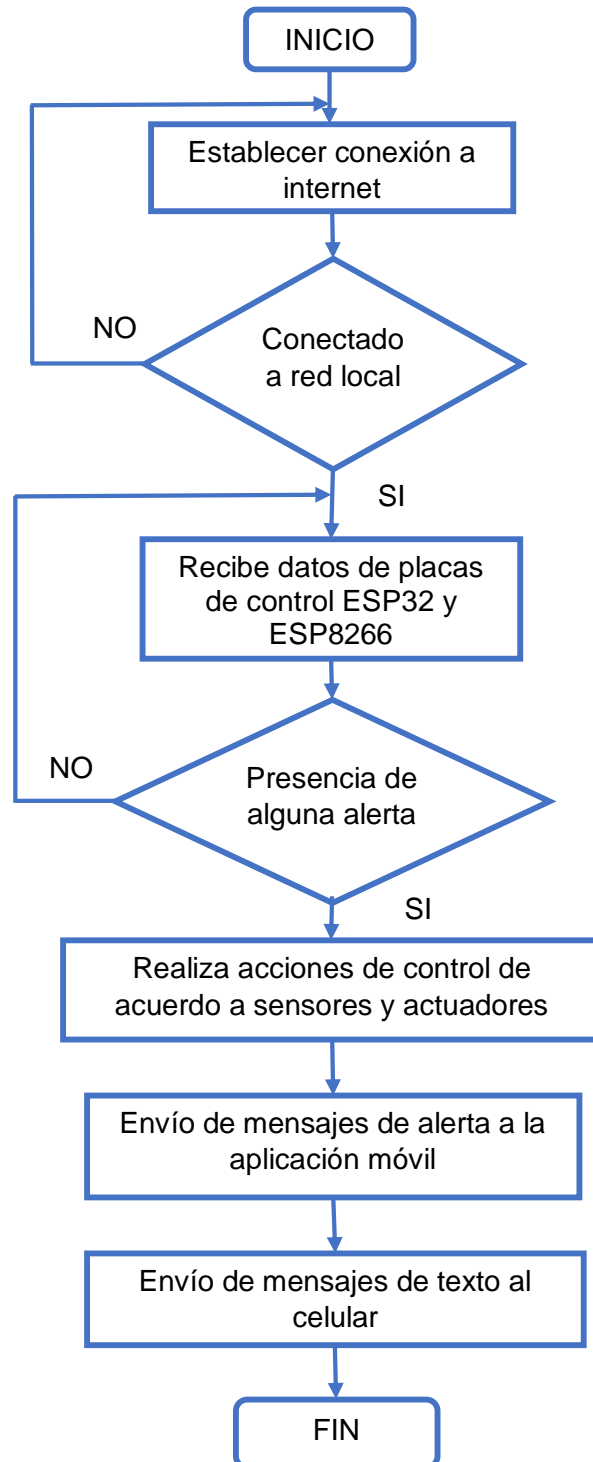


Fig. 52 Diagrama de flujos de funcionamiento Raspberry Pi  
Fuente: (Autor,2023)

### 3.2.2 Programación tarjeta de control ESP32 y ESP8266

Para la programación de las tarjetas de control ESP32 Y ESP8266 se utilizó el entorno de desarrollo (IDE) Arduino, dentro de este software se encuentran varias funcionalidades que permiten desarrollar, compilar y cargar el código de programación a las tarjetas de control. A continuación, la Figura 3.18 muestra la ubicación del gestor de tarjetas, donde se puede seleccionar el tipo de tarjeta con el cual se vaya a realizar el proyecto.

Por otra parte, una vez ya finalizado el código de programación es importante seleccionar el puerto en el cual se encuentra conectada la placa de control a nuestro computador, estos dos puntos son esenciales para no tener errores en nuestra programación y a la hora de subir el código a la placa que se está utilizando.

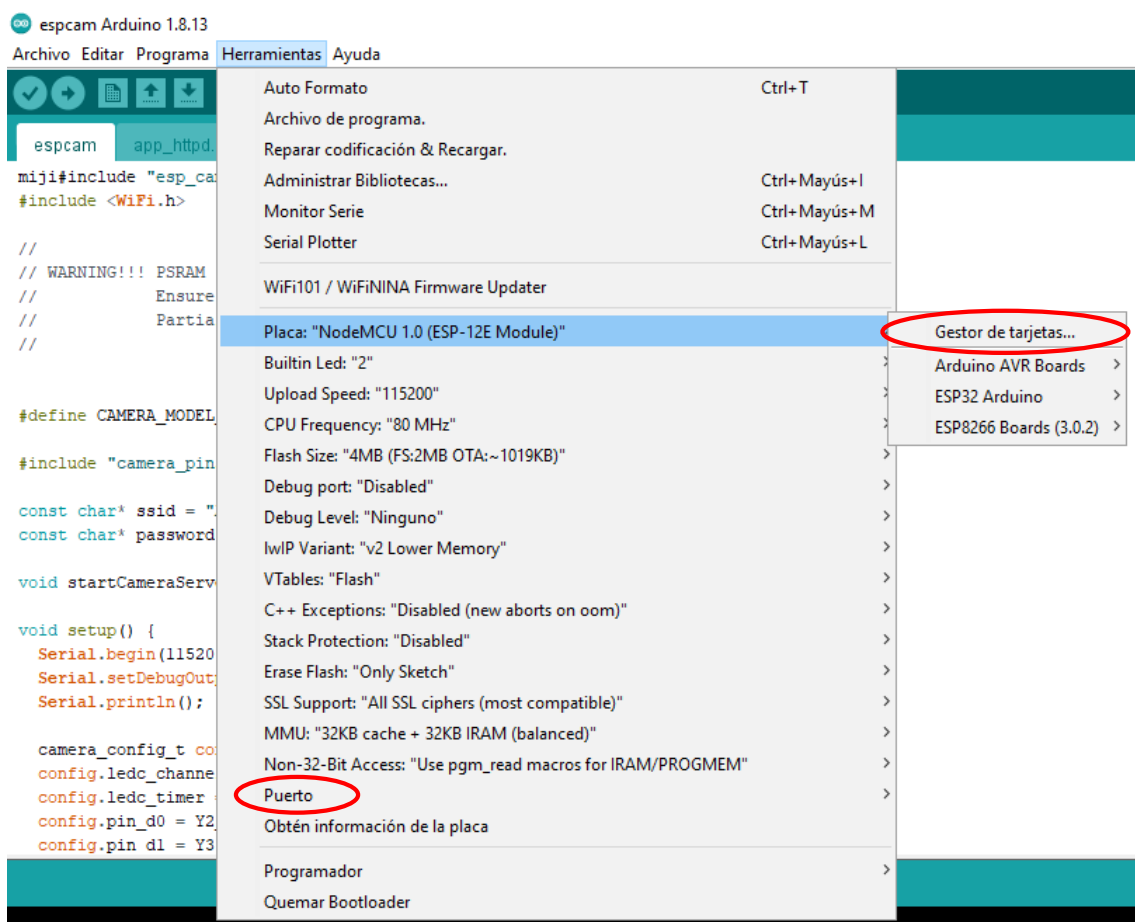


Fig. 53 Entorno de programación Arduino  
Fuente: (Autor,2023)

Para el presente proyecto se utilizó tres modelos de placas de control; Node32s para el control del espacio físico de la sala por su gran cantidad de variables a controlar, Generic ESP8266 Module esta placa se utilizó para el control del resto de espacios físicos; cocina, dormitorio 1, dormitorio 2, comedor y baño. Por último, se usó el modelo AI Thinker ESP32-CAM para el monitoreo de seguridad con cámaras ESP32 Cam.

### 3.3 Diagrama de flujos de los espacios físicos

A continuación, se describen los diagramas de flujos de cada uno de los espacios físicos con sus respectivos componentes domóticos designados para cada área dentro de la vivienda, con el fin de representar de forma clara, esquemática y secuencial el proceso de funcionamiento del sistema domótico.

- **Diagrama de flujo del área de la cocina**

La siguiente Figura 3.19 describe el funcionamiento del sistema domótico instalado en el área de la cocina, como primer punto la tarjeta de control ESP8266 establece una conexión con la red local de Internet, una vez conectada, la tarjeta se encarga de leer el estado de los sensores y la activación y desactivación de los actuadores y, por último, todos los datos son enviados a la central de control Raspberry Pi 3, el código de programación de esta sección se la puede apreciar en el anexo 2.

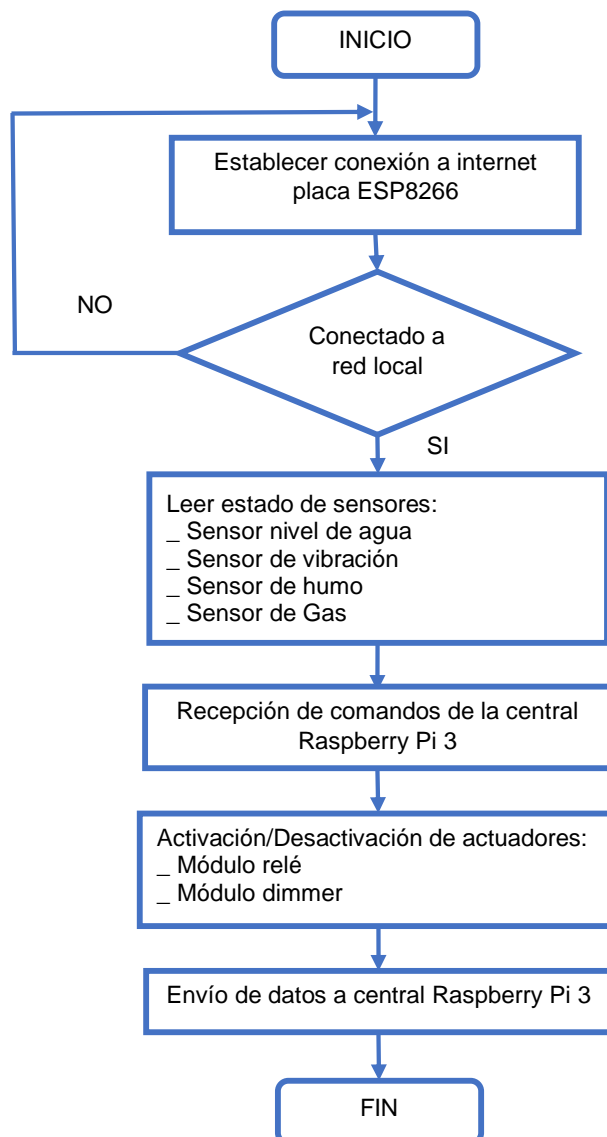


Fig. 54 Diagrama de flujo de la cocina  
Fuente: (Autor,2023)

- **Diagrama de flujo del área de la sala**

En la Figura 3.20 se puede apreciar el funcionamiento del sistema domótico instalado en el área de la sala, para que empiece a operar el prototipo la tarjeta de control ESP32 establece una conexión con la red local de Internet, una vez conectada, la tarjeta se encarga de leer el estado de los sensores y la activación y desactivación de los actuadores designados para esta área y, por último, todos los datos son enviados a la central de control Raspberry Pi 3, el código de programación de esta sección se puede apreciar en el anexo 3.

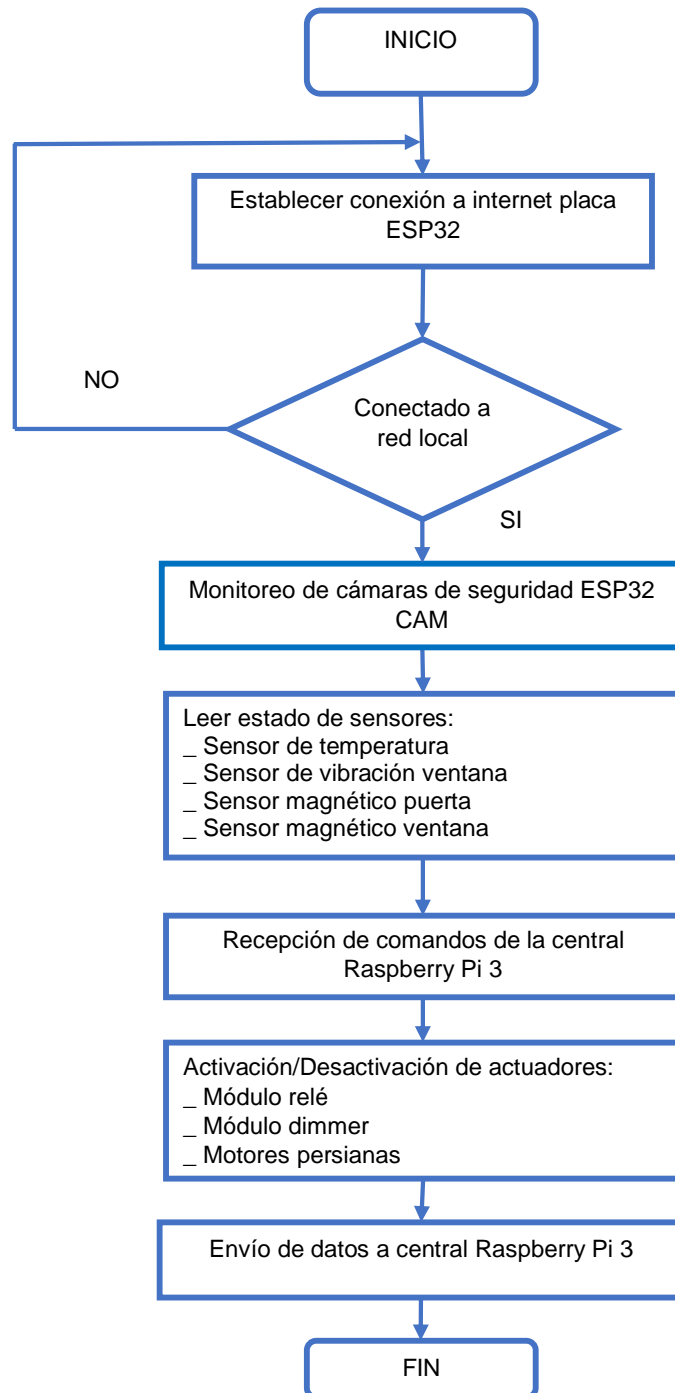


Fig. 55 Diagrama de flujo de la sala  
Fuente: (Autor,2023)

- **Diagrama de flujo del área del baño**

La presente Figura 3.21 describe el funcionamiento del prototipo domótico instalado en el área del baño, primero la tarjeta de control ESP8266 establece una conexión con la red local de Internet, una vez conectada, la tarjeta se encarga de leer el estado de los sensores y la activación y desactivación de los actuadores designados para esta área. Posteriormente, todos los datos recopilados son enviados a la central de control Raspberry Pi 3, el código de programación de esta sección se encuentra en el anexo 4.

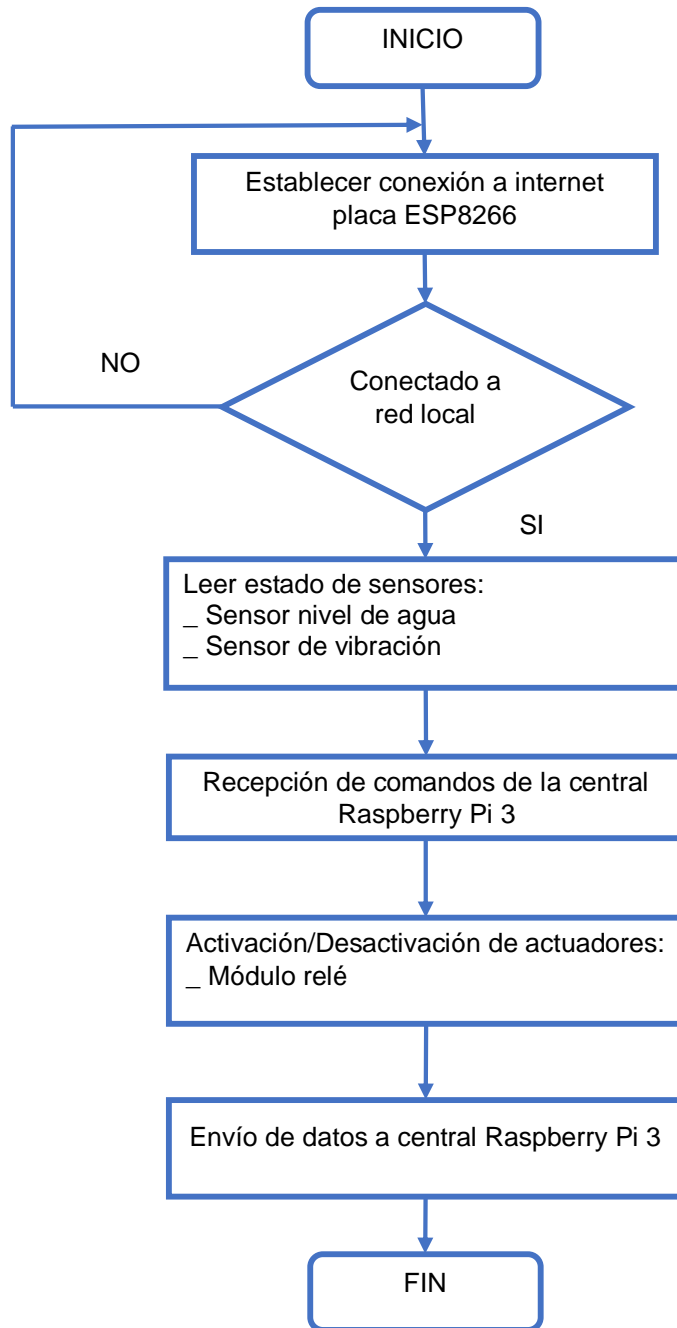


Fig. 56 Diagrama de flujo del baño  
Fuente: (Autor,2023)

- **Diagrama de flujo del área de los dormitorios**

En la Figura 3.22 se puede apreciar el funcionamiento del sistema domótico instalado en el área de los dormitorios, para que empiece el funcionamiento del prototipo la tarjeta de control ESP8266 establece una conexión con la red local de Internet, una vez conectada, la tarjeta se encarga de leer el estado de los sensores y actuadores designados para esta área y, por último, todos los datos son enviados a la central de control Raspberry Pi 3, el código de programación de esta sección se puede apreciar en el anexo 5 y 6.

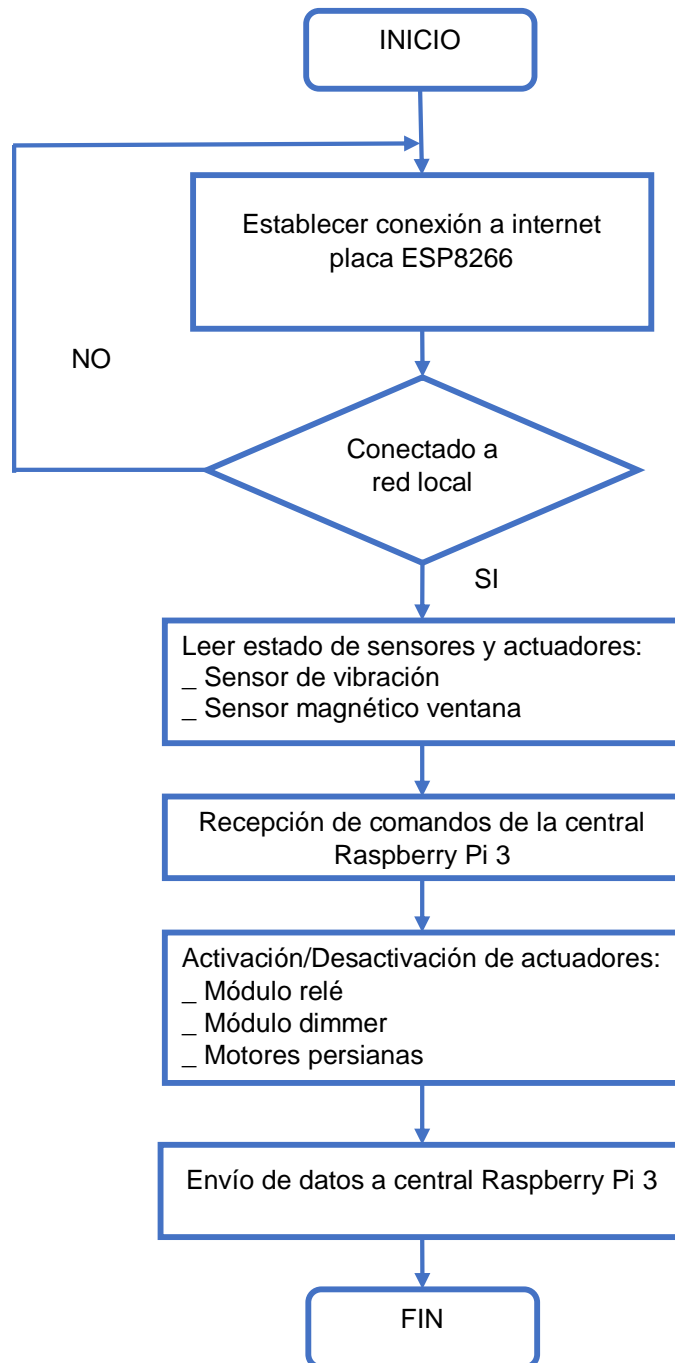


Fig. 57 Diagrama de flujo de los dormitorios  
Fuente: (Autor,2023)

- **Diagrama de flujo del área del comedor**

En la Figura 3.23 se puede apreciar el funcionamiento del sistema domótico instalado en el área del comedor, para que empiece a operar el prototipo la tarjeta de control ESP8266 establece una conexión con la red local de Internet, una vez conectada, la tarjeta se encarga de leer el estado de los sensores y actuadores designados para esta área y, por último, todos los datos son enviados a la central de control Raspberry Pi 3, el código de programación de esta sección se encuentra en el anexo 7.

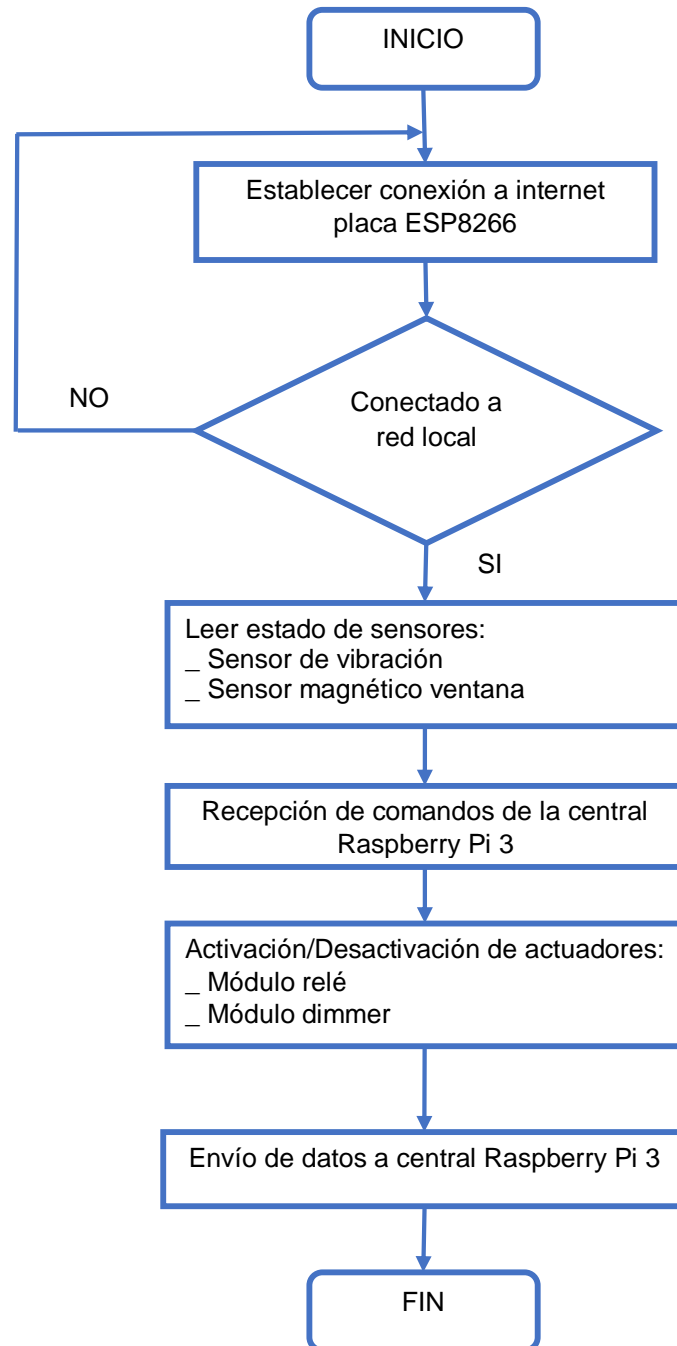


Fig. 58 Diagrama de flujo del comedor  
Fuente: (Autor,2023)



### 3.4 Diseño de la aplicación móvil en App Inventor

Para que la persona pueda interactuar con el sistema domótico, se ha desarrollado un software de control que sirve de interfaz de usuario, permitiendo a éste, seleccionar una unidad remota y a continuación consultar el estado de los sensores, así como activar o desactivar los actuadores. El proceso finaliza solamente cuando no exista problemas de seguridad. Por lo contrario si existe algún problema de seguridad vuelve al bloque de recepción de datos, para nuevamente procesar la información, el código de programación de esta sección se puede observar en el anexo 8.

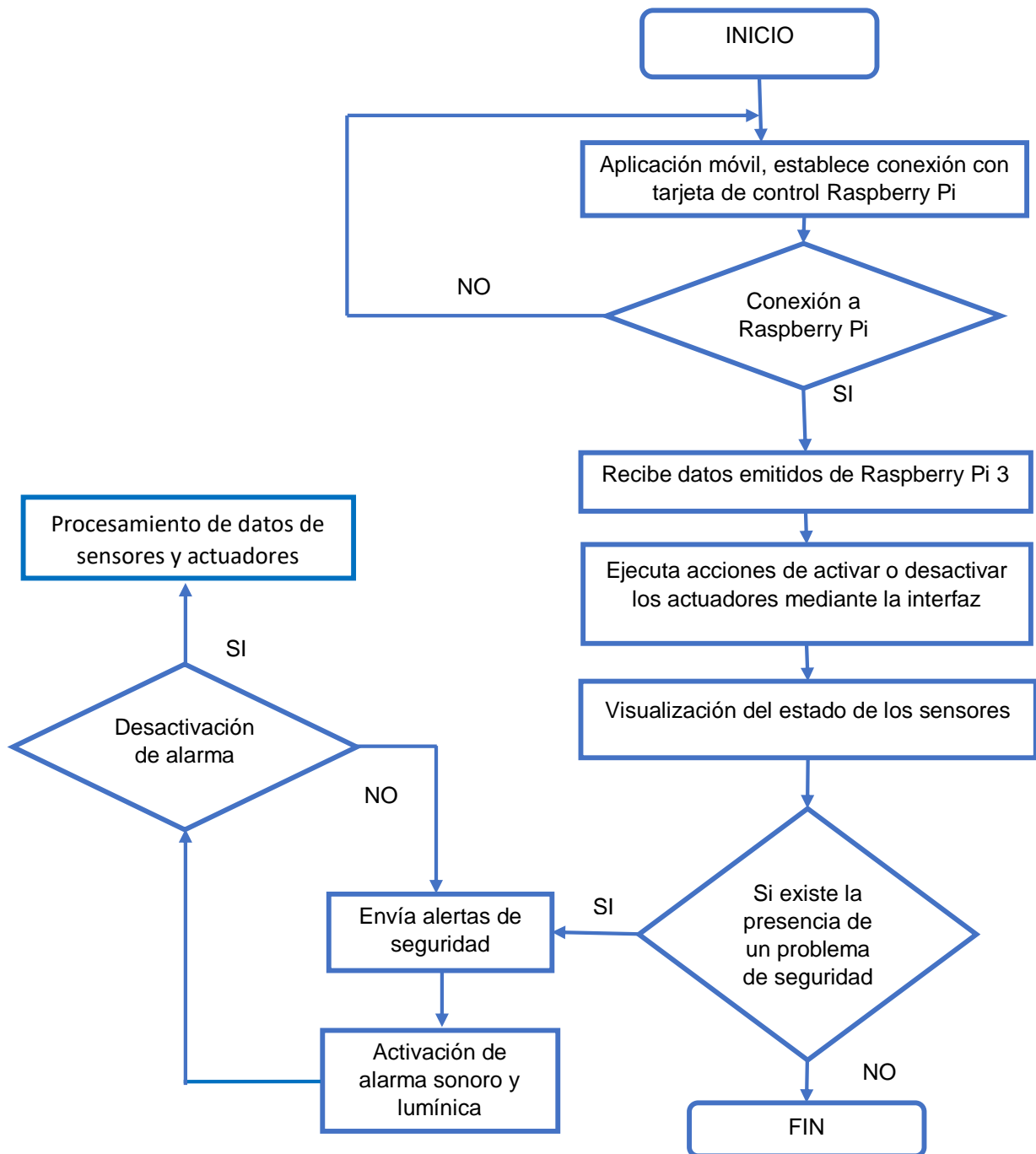


Fig. 59 Diagrama de flujo del funcionamiento de la aplicación móvil  
Fuente: (Autor,2023)

### 3.4.1 Diseño de la aplicación en la plataforma Firebase

Para el almacenamiento de datos en la nube del prototipo domótico se desarrolló una aplicación en la plataforma Firebase que permite almacenar y sincronizar los datos en tiempo real. A continuación, la Figura 3.25 muestra las 5 áreas domóticas creadas para la interacción con la aplicación móvil.

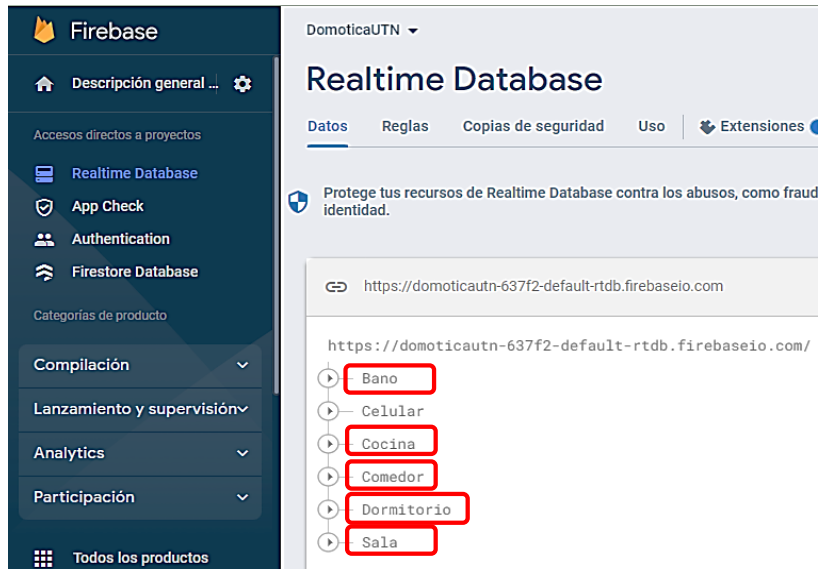


Fig. 60 Plataforma de desarrollo Firebase  
Fuente: (Autor,2023)

- Además, dentro de estas áreas se encuentran los sensores y actuadores con sus respectivos nombres que se diseñó para cada uno de estos espacios físicos de control domótico, en la Figura 3.26 se puede apreciar el estado de los sensores y actuadores de una de las áreas de aplicación domótica.

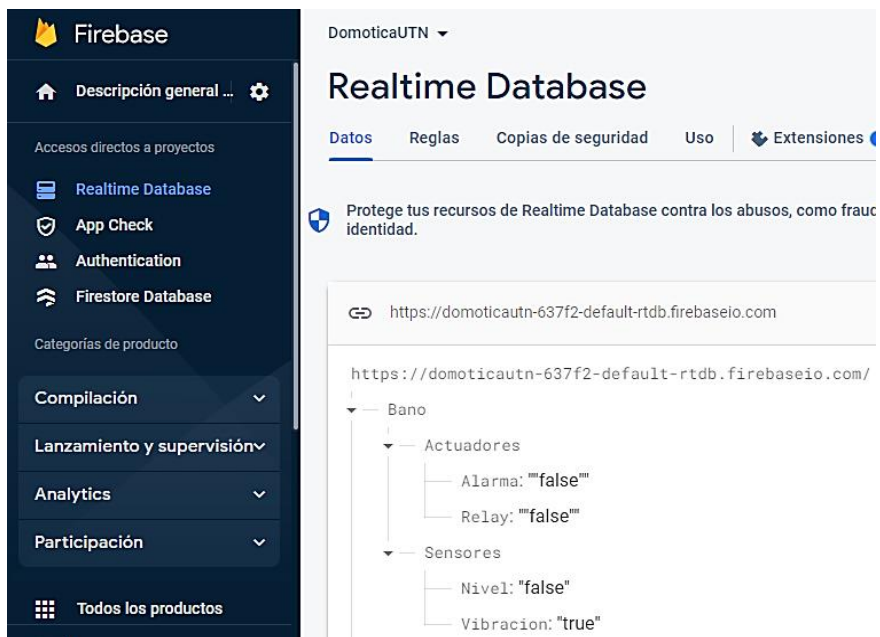


Fig. 61 Diseño de la aplicación para los espacios físicos en Firebase  
Fuente: (Autor,2023)

### 3.4.2 Diseño de entorno principal de la aplicación móvil

El diseño principal para la aplicación móvil se desarrolló de acuerdo a los espacios físicos especificados en el anterior capítulo, donde se describió las siguientes áreas: una sala, un comedor, una cocina, 2 dormitorios y un baño.

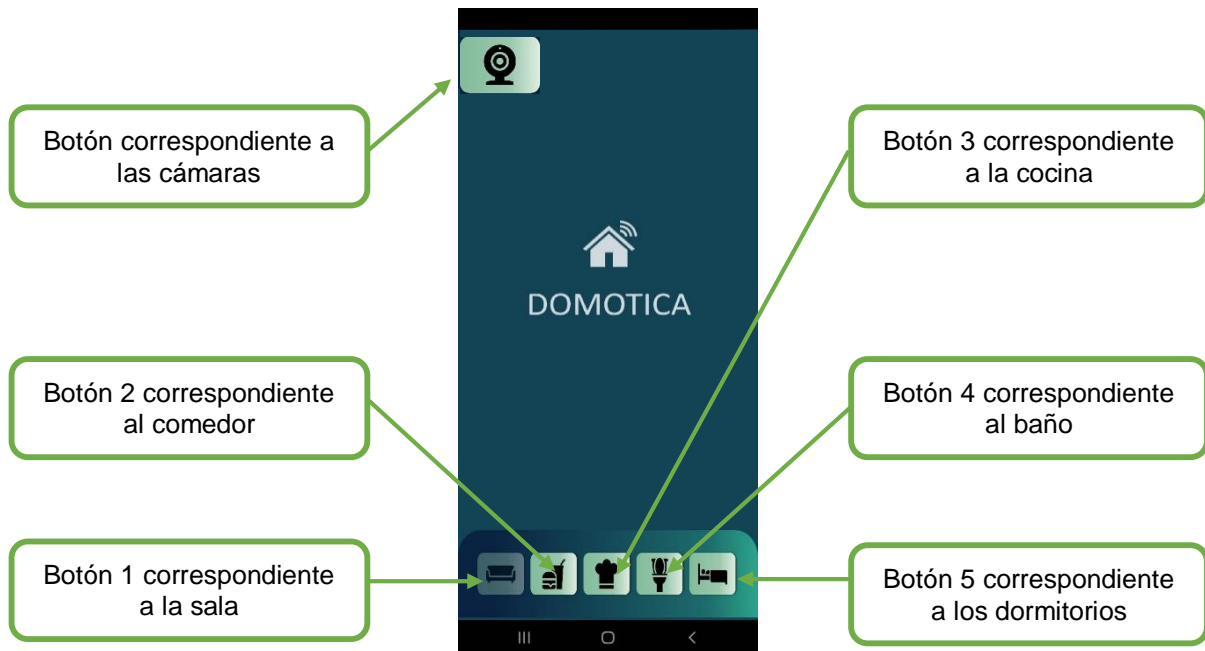


Fig. 62 Entorno principal App Inventor  
Fuente: (Autor,2023)

La Figura 3.27 describe el entorno principal de la aplicación móvil, donde se encuentran 5 botones correspondientes a los espacios físicos del sistema domótico, el primer botón corresponde al espacio denominado sala, el segundo corresponde al comedor, el tercero corresponde a la cocina, el cuarto botón corresponde al baño y el último botón corresponde a los dormitorios, existe un botón en la parte superior izquierda que pertenece a las cámaras de seguridad.

### 3.4.3 Diseño de la aplicación móvil para los espacios físicos

A continuación, se describe el desarrollo de la aplicación móvil para cada uno de los espacios físicos que comprenden el sistema domótico.

#### ➤ Diseño de la aplicación móvil para la sala

El área de la sala fue donde más soluciones domóticas se aplicaron, en la siguiente Figura 3.28 se puede apreciar la interfaz gráfica en la cual se muestran las diferentes aplicaciones domóticas que se realizó, dentro de estas tenemos las siguientes: el encendido y apagado de cargas, control de luces, apertura y cierre de persianas, así como el estado de sensores en puertas y ventanas, sensor de temperatura y sensor de vibración de ventanas.

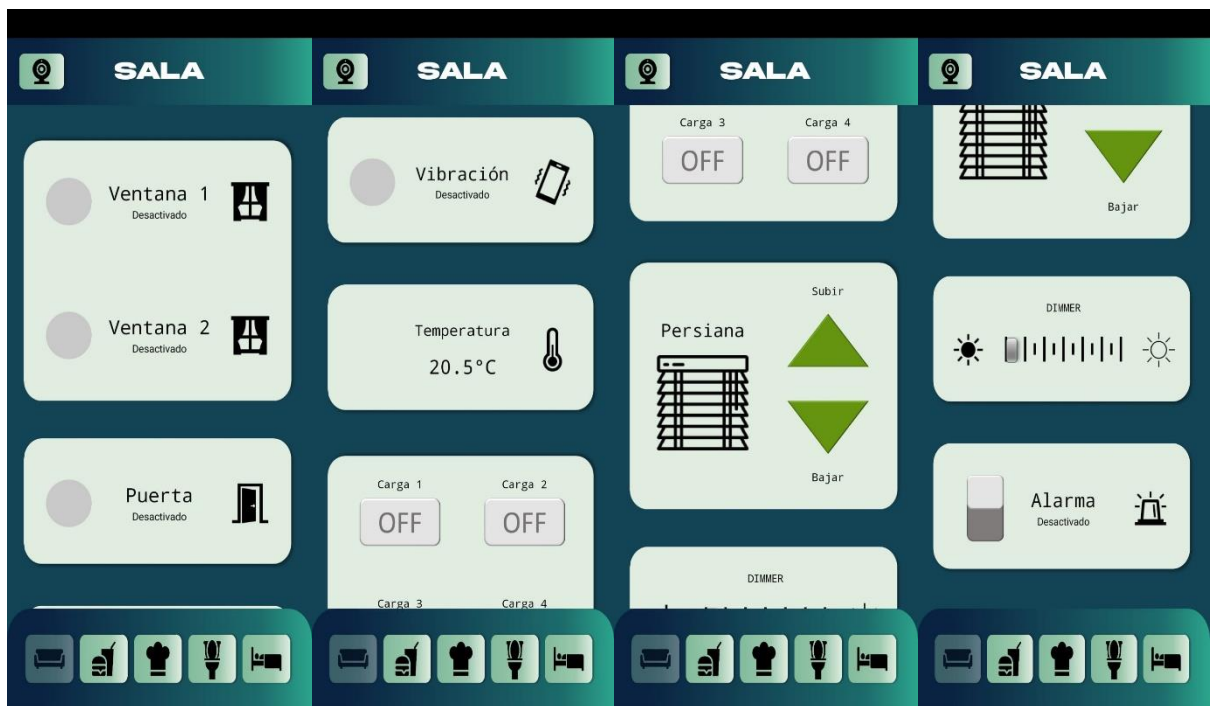


Fig. 63 Diseño de aplicación móvil para sala  
Fuente: (Autor,2023)

### ➤ Diseño de la aplicación móvil para el comedor

Para el área del comedor se realizó el diseño de la aplicación móvil como se muestra en la Figura 3.29, dentro de esta sección se encuentran las siguientes aplicaciones domóticas; el encendido y apagado de cargas, control de luces, así como el estado del sensor magnético en la venta y sensor de vibración de ventanas.



Fig. 64 Diseño de aplicación móvil para el comedor  
Fuente: (Autor,2023)

### ➤ **Diseño de la aplicación móvil para la cocina**

La Figura 3.30 muestra el diseño de la aplicación móvil para el área de la cocina, dentro de esta sección se encuentran las siguientes aplicaciones domóticas; el encendido y apagado de cargas, control de luces, así como el sensor de vibración de ventanas, sensor de inundación, sensor de humo y sensor de gas.



Fig. 65 Diseño de aplicación móvil para la cocina  
Fuente: (Autor,2023)

### ➤ **Diseño de la aplicación móvil para los dormitorios**

En la presente Figura 3.31 se observa el diseño de la aplicación móvil que se realizó para el área de los dos dormitorios; para el primer dormitorio se aplicó las siguientes aplicaciones domóticas: el encendido y apagado de cargas, control de luces, apertura y cierre de persianas, así como el sensor de vibración de ventanas y sensor magnético para ventanas. Para el segundo dormitorio se instaló las siguientes soluciones domóticas: el encendido y apagado de cargas, control de luces y apertura y cierre de persianas.



Fig. 66 Diseño de aplicación móvil para dormitorios  
Fuente: (Autor,2023)

### ➤ Diseño de la aplicación móvil para el baño

En la Figura 3.32 se puede apreciar el diseño de la aplicación móvil para el área del baño se aplicó las siguientes soluciones domóticas; el encendido y apagado de cargas, así como el estado del sensor de vibración de ventanas y sensor de inundación.



Fig. 67 Diseño de aplicación móvil para el baño  
Fuente: (Autor,2023)

### ➤ **Diseño de la aplicación móvil para las cámaras de seguridad**

En el diseño de la aplicación se encuentra un icono en la parte superior izquierda, el mismo se puede visualizar en todos los espacios físicos del domicilio, lo cual facilita el ingreso en cualquier momento a las cámaras 1 y 2 como se puede apreciar en la Figura 3.33.



Fig. 68 Diseño de aplicación móvil para las cámaras de seguridad  
Fuente: (Autor,2023)







Por último, es importante mencionar que cada espacio físico de la aplicación móvil cuenta con un icono llamado alarma, la misma que trabajará en conjunto con varios sensores que se encuentran ubicados en sitios estratégicos dentro de la vivienda, en el caso de que exista una intrusión por terceros al hogar o la presencia de fugas de gas o agua.

Además, este sistema de alarma está conformado de dos partes; la primera que emite alertas de seguridad a la aplicación y de igual manera envía mensajes de alerta al celular que el usuario establezca, para cumplir con su objetivo la alarma se encuentra ubicada de manera física dentro de la vivienda y su control se la puede realizar mediante la aplicación móvil.




### **3.5 Ubicación de sensores y actuadores dentro de la vivienda**

En la Figura 3.34 se describe la ubicación de los sensores y actuadores que se implementó dentro de cada uno de los espacios físicos que comprenden el prototipo domótico, estos están representados en el plano del domicilio mediante un círculo de diferentes colores de acuerdo al tipo de sensor, de igual manera los actuadores están representados por un cuadrado de diferentes colores de acuerdo al tipo de actuador, algunos otros elementos que conforman el sistema domótico se encuentran representados por un triángulo de diferentes colores.






En la parte de sensores tenemos los siguientes:

<b>Tipo de sensor</b>	<b>Símbolo</b>
Sensor de nivel de agua, representado por un círculo azul	
Sensor de gas, representado por un círculo amarillo	
Sensor de humo, representado por un círculo plomo	
Sensor de temperatura, representado por un círculo verde	
Sensor de vibración, representado por un círculo rojo	
Sensor de magnético, representado por un círculo café	

En la parte de actuadores tenemos los siguientes:

<b>Tipo de actuador</b>	<b>Símbolo</b>
Módulo relé, representado por un cuadrado morado	
Motores para persianas, representado por un cuadrado naranja	
Módulo dimmer, representado por un cuadrado verde	

Otros elementos del sistema domótico:

<b>Tipo de actuador</b>	<b>Símbolo</b>
Cámara de seguridad ESP32 CAM, representado con un triángulo azul	
Controladora Raspberry Pi 3 modelo B+, representado por un triángulo naranja	
Tarjeta de control ESP32, representado por un triángulo verde	
Tarjeta de control ESP8266, representado por un triángulo amarillo	
Sistema de alarma, representado por un triángulo rojo	



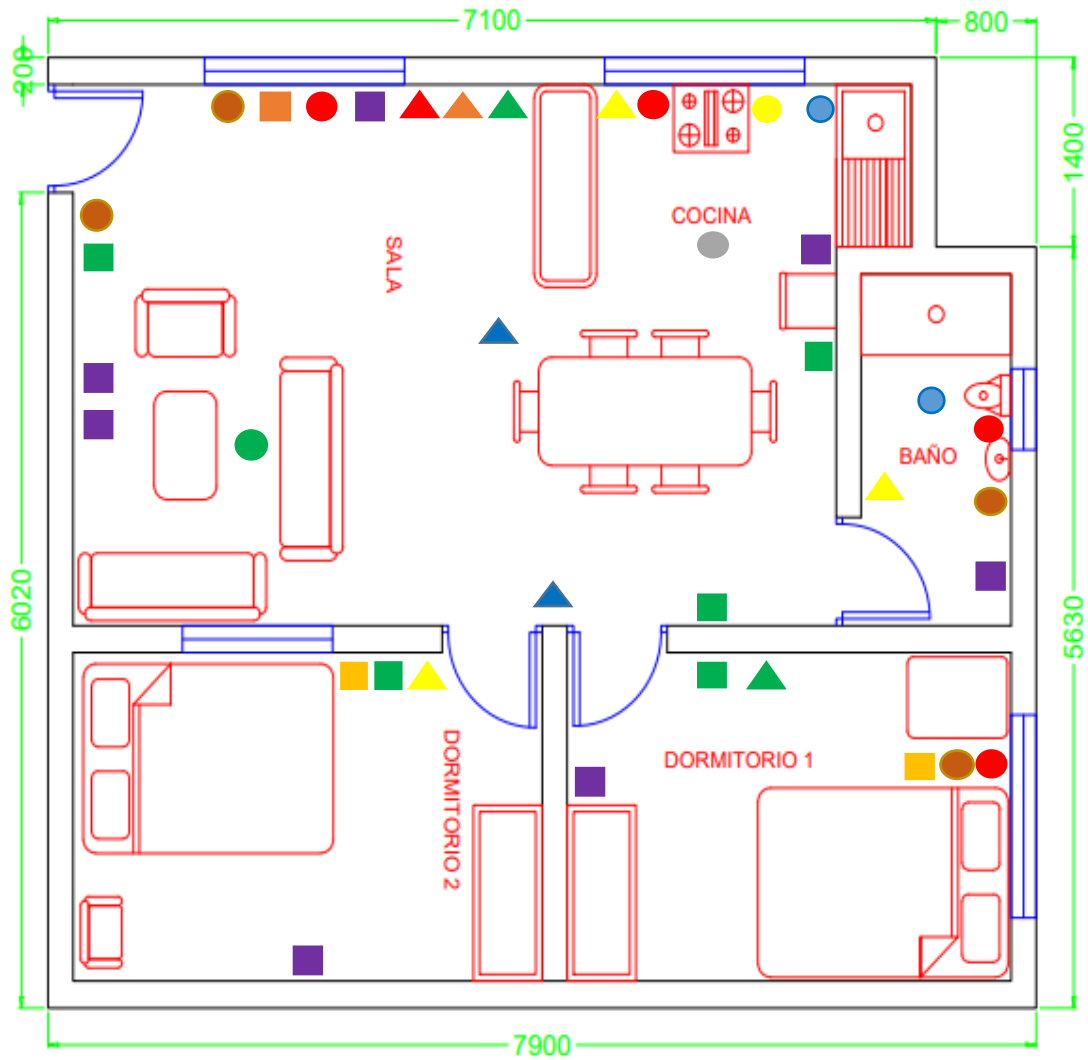


Fig. 69 Ubicación de sensores y actuadores dentro del plano arquitectónico  
Fuente: (Autor,2023)

### 3.6 Pruebas de funcionamiento

Las pruebas de funcionamiento constan de tres fases; la primera, se realizó el montaje de las pacas de control con sus respectivos sensores y actuadores designados para cada área. Segundo, se llevó a cabo las conexiones eléctricas de los componentes domóticos y, por último, se realizó las pruebas de funcionamiento unitarias y conjuntas para cada espacio físico de la vivienda, con el uso de la aplicación móvil desarrollada en App Inventor, que se encuentra instalada en un teléfono inteligente con sistema Android.

Cabe mencionar que las pruebas ejecutadas a los diferentes sensores y actuadores, se las realizó durante una semana en diferentes horarios para comprobar el funcionamiento del prototipo domótico dentro de una vivienda, por lo tanto, la respuesta de estos depende mucho de la velocidad de la red del internet a la que se encuentre conectado el sistema domótico.

- **Pruebas de funcionamiento dentro del espacio físico sala**

A continuación, en la Figura 3.35 se puede observar la instalación de la caja blanca, dentro de esta se encuentra la placa de control ESP32 con los diferentes sensores y actuadores, los mismos que se encuentran distribuidos en diferentes partes de la sala.



Fig. 70 Instalación de elementos domóticos en el área de la sala.  
Fuente: (Autor,2023)

La siguiente Tabla 3.1 describe las diferentes pruebas de funcionamiento que se realizó a cada uno de los elementos domóticos instalados en el área de la sala, con el uso de la aplicación móvil para Android que se instaló en el teléfono inteligente.

Tabla 16 Pruebas de funcionamiento unitarias a sensores y actuadores en el área de la sala

<b>Elemento</b>	<b>Tiempo de respuesta del sensor</b>	<b>Tipo de prueba realizada</b>	<b>Observaciones</b>
Sensor magnético en puerta	De 1 a 5 segundos	Apertura y cierre de puerta	De las 10 repeticiones que se hizo, el 90 % responden a este tiempo
Sensor magnético en ventana	De 2 a 5 segundos	Apertura y cierre de ventana	De las 10 repeticiones que se hizo, el 90 % responden a este tiempo
Sensor de temperatura	De 5 a 10 segundos	Variación de temperatura aplicando calor y frío al sensor	
Sensor de vibración	De 2 a 5 segundos	Golpes leves y fuertes en la ventana	Se realizó 10 repeticiones, de las cuales el 90 % responde a este tiempo
Actuador de persianas	De 1 a 2 minutos	Apertura y cierre de una persiana	Este tiempo se demora en abrir o cerrar una cortina de 1.50 metros de alto

Actuador dimmer	De 3 a 5 segundos	Atenuación de luces en corriente AC no mayores a 8 Amperios
Actuador relé	De 1 a 5 segundos	Encendido y apagado de cargas en corriente AC no mayores a 10 Amperios

Fuente: (Autor, 2023)



Fig. 71 Prueba de funcionamiento del espacio físico sala  
Fuente: (Autor,2023)

La Figura 3.36 describe la prueba de funcionamiento conjuntas que se realizó en el área de la sala, donde se puede apreciar el estado de los siguientes sensores; sensor de vibración de ventanas, sensor magnético en puerta y ventana y sensor de temperatura, cabe mencionar que todos estos sensores a diferencia del sensor de temperatura se encuentran conectados al sistema de seguridad que mediante una alarma sonora y visual indican al usuario la presencia de un problema en el prototipo domótico.

Por otra parte, se encuentra el funcionamiento de los siguientes actuadores; las cargas 1, 2, 3 y 4 para el encendido y apagado de cualquier tipo de electrodoméstico, dimmer para atenuación de luz y por último la apertura y cierre de persianas.

- **Pruebas de funcionamiento dentro del espacio físico comedor**

En la Figura 3.37 se aprecia la caja blanca donde se encuentra la placa de control ESP8266 con los diferentes sensores y actuadores, solo en esta área no se implementó físicamente dentro de la vivienda por motivos de diseño, queda como una proyección para casas que cuenten con un área de comedor.



Fig. 72 Instalación de elementos domóticos en el área del comedor  
Fuente: (Autor,2023)

La Tabla 3.2 describe las diferentes pruebas de funcionamiento que se realizó a cada uno de los elementos domóticos que conforman el área del comedor, con el uso de la aplicación móvil para Android que se instaló en el teléfono inteligente.

Tabla 17 Pruebas de funcionamiento unitarias a sensores y actuadores en el área del comedor

<b>Elemento</b>	<b>Tiempo de respuesta del sensor</b>	<b>Tipo de prueba realizada</b>	<b>Observaciones</b>
Sensor magnético en ventana	De 2 a 5 segundos	Apertura y cierre del sensor	De las 10 repeticiones que se hizo, el 90 % responden a este tiempo
Sensor de vibración	De 2 a 5 segundos	Golpes leves y fuertes en el sensor	Se realizó 10 repeticiones, de las cuales el 90 % responde a este tiempo
Actuador dimmer	De 3 a 5 segundos	Atenuación de luces en corriente AC no mayores a 8 Amperios	
Actuador relé	De 1 a 5 segundos	Encendido y apagado de cargas en corriente AC no mayores a 10 Amperios	

Fuente: (Autor, 2023)



Fig. 73 Prueba de funcionamiento del espacio físico comedor  
Fuente: (Autor,2023)

En la Figura 3.38 se puede apreciar las pruebas de funcionamiento conjuntas que se realizó en el área del comedor, aquí se puede observar el estado de los siguientes sensores; sensor de vibración en la ventana y sensor magnético en la ventana, cabe mencionar que estos dos sensores se encuentran conectados al sistema de seguridad que mediante una alarma sonora y visual indican al usuario la presencia de un problema en el prototipo domótico.

Por otra parte, se encuentra el funcionamiento de los siguientes actuadores; relé para el encendido y apagado de cargas y dimmer para atenuación de luz.

- **Pruebas de funcionamiento dentro del espacio físico cocina**

Para el área de la cocina se realizó la instalación de la caja blanca como se puede apreciar en la Figura 3.39, donde se encuentra ubicada la placa de control ESP8266 con los diferentes sensores y actuadores, los mismos que se encuentran distribuidos en lugares específicos designados para esta área.



Fig. 74 Instalación de elementos domóticos en el área de la cocina  
Fuente: (Autor,2023)

La Tabla 3.3 describe las diferentes pruebas de funcionamiento que se realizó a cada uno de los elementos domóticos que conforman el área de la cocina, con el uso de la aplicación móvil para Android que se instaló en el teléfono inteligente.

Tabla 18 Pruebas de funcionamiento unitarias en el área de la cocina

Elemento	Tiempo de respuesta del sensor	Tipo de prueba realizada	Observaciones
Detector de gas	De 5 a 10 segundos	Fuga de gas de la hornilla de la cocina	
Detector de incendios	De 5 a 10 segundos	Simulación con humo aplicado directamente al detector	
Sensor magnético en ventana	De 2 a 5 segundos	Apertura y cierre de ventana	De las 10 repeticiones que se hizo, el 90 % responden a este tiempo
Sensor de vibración	De 2 a 5 segundos	Golpes leves y fuertes en la ventana	Se realizó 10 repeticiones, de las cuales el 90 % responde a este tiempo
Sensor de inundación	De 3 a 6 segundos	Simulación de inundación con un envase lleno de agua	
Actuador dimmer	De 3 a 5 segundos	Atenuación de luces en corriente AC no mayores a 8 Amperios	
Actuador relé	De 1 a 5 segundos	Encendido y apagado de cargas en corriente AC no mayores a 10 Amperios	

Fuente: (Autor, 2023)



Fig. 75 Prueba de funcionamiento del espacio físico cocina  
Fuente: (Autor,2023)

En la Figura 3.40 se observa las pruebas de funcionamiento conjuntas que se realizó en el área de la cocina, aquí se puede apreciar el estado de los siguientes sensores; sensor de vibración en la ventana, sensor magnético en la ventana, sensor de nivel de agua, sensor de gas y sensor de humo, todos estos sensores se encuentran conectados al sistema de seguridad que mediante una alarma sonora y visual indican al usuario la presencia de un problema en el prototipo domótico.

Por otra parte, se encuentra el funcionamiento de los siguientes actuadores; relé para el encendido y apagado de cargas y dimmer para atenuación de luz.

- **Pruebas de funcionamiento dentro del espacio físico baño**

Para el área del baño se realizó la instalación de la caja blanca como se puede apreciar en la Figura 3.41, donde se encuentra la placa de control ESP8266 conectada a los diferentes sensores y actuadores, los mismos que se encuentran distribuidos en diferentes partes específicas para esta área.



Fig. 76 Instalación de elementos domóticos en el área del baño  
Fuente: (Autor,2023)

La Tabla 3.4 describe las diferentes pruebas de funcionamiento unitarias que se realizó a cada uno de los elementos domóticos que conforman el área de la cocina, con el uso de la aplicación móvil para Android que se instaló en el teléfono inteligente.

Tabla 19 Pruebas de funcionamiento unitarias para el área del baño

Elemento	Tiempo de respuesta del sensor	Tipo de prueba realizada	Observaciones
Sensor de vibración	De 2 a 5 segundos	Golpes leves y fuertes en la ventana	Se realizó 10 repeticiones, de las cuales el 90 % responde a este tiempo
Sensor de inundación	De 3 a 6 segundos	Simulación de inundación con un envase lleno de agua	
Actuador relé	De 1 a 5 segundos	Encendido y apagado de cargas en corriente AC no mayores a 10 Amperios	

Fuente: (Autor, 2023)



Fig. 77 Prueba de funcionamiento del espacio físico baño  
Fuente: (Autor,2023)

La Figura 3.42 describe las pruebas de funcionamiento que se realizó en el área del baño, aquí se puede apreciar el estado de los siguientes sensores; sensor de vibración en la ventana, y sensor de nivel de agua, estos dos sensores se encuentran conectados al sistema de seguridad que mediante una alarma sonora y visual indican al usuario la presencia de un problema en el prototipo domótico.

Por otra parte, se encuentra el funcionamiento del siguiente actuador; relé para el encendido y apagado de la luz.



- **Pruebas de funcionamiento dentro del espacio físico dormitorios**

Para el área de los dormitorios se realizó la instalación de la caja blanca como se puede apreciar en la Figura 3.43, donde se encuentra implementada la placa de control ESP8266 con los diferentes sensores y actuadores, los mismos que se encuentran distribuidos en distintos lugares específicos del área de los dormitorios.



Fig. 78 Instalación de elementos domóticos en el área de dormitorios  
Fuente: (Autor,2023)

La Tabla 3.5 describe las diferentes pruebas de funcionamiento que se realizó a cada uno de los elementos domóticos que conforman el área de los dormitorios, con el uso de la aplicación móvil para Android que se instaló en el teléfono inteligente.

Tabla 20 Pruebas de funcionamiento unitarias en el área de los dormitorios

Elemento	Tiempo de respuesta del sensor	Tipo de prueba realizada	observaciones
Sensor magnético en ventana	De 2 a 5 segundos	Apertura y cierre de ventana	De las 10 repeticiones que se hizo, el 90 % responden a este tiempo
Sensor de vibración	De 2 a 5 segundos	Golpes leves y fuertes en la ventana	Se realizó 10 repeticiones, de las cuales el 90 % responde a este tiempo
Actuador de persianas	De 1 a 2 minutos	Apertura y cierre de una persiana	Este tiempo se demora en abrir o cerrar una cortina de 1.50 metros de alto
Actuador dimmer	De 3 a 5 segundos	Atenuación de luces en corriente AC no mayores a 8 Amperios	
Actuador relé	De 1 a 5 segundos	Encendido y apagado de cargas en corriente AC no mayores a 10 Amperios	

Fuente: (Autor, 2023)



Fig. 79 Prueba de funcionamiento del espacio físico dormitorios  
Fuente: (Autor,2023)

La Figura 3.44 muestra las pruebas de funcionamiento que se realizó en el área de los dormitorios, donde se puede apreciar el estado de los siguientes sensores; sensor de vibración en la ventana y sensor magnético en la ventana, estos dos sensores se encuentran conectados al sistema de seguridad que mediante una alarma sonora y visual indican al usuario la presencia de un problema en el prototipo domótico, cabe mencionar que se ocupó estos sensores solo para el dormitorio 1 porque este tiene una venta que da al exterior de la casa.

Por otra parte, se encuentra el funcionamiento de los siguientes actuadores que se aplicaron a los dos dormitorios; relé para el encendido y apagado de cargas y dimmer para atenuación de luz.

- **Pruebas de funcionamiento de comunicación entre la central de control Raspberry Pi 3 modelo B+ y las placas ESP32 Y ESP8266**

Para apreciar el envío de datos desde las placas ESP32 y ESP8266 hacia la central de control Raspberry Pi, se hizo uso del software Arduino en el cual por medio del monitor serial se puede apreciar en tiempo real el estado de sensores y actuadores cada 300 milisegundos como se muestra en la Figura 3.45, lo que resulta una oportuna y rápida comunicación entre las placas de control del sistema domótico.

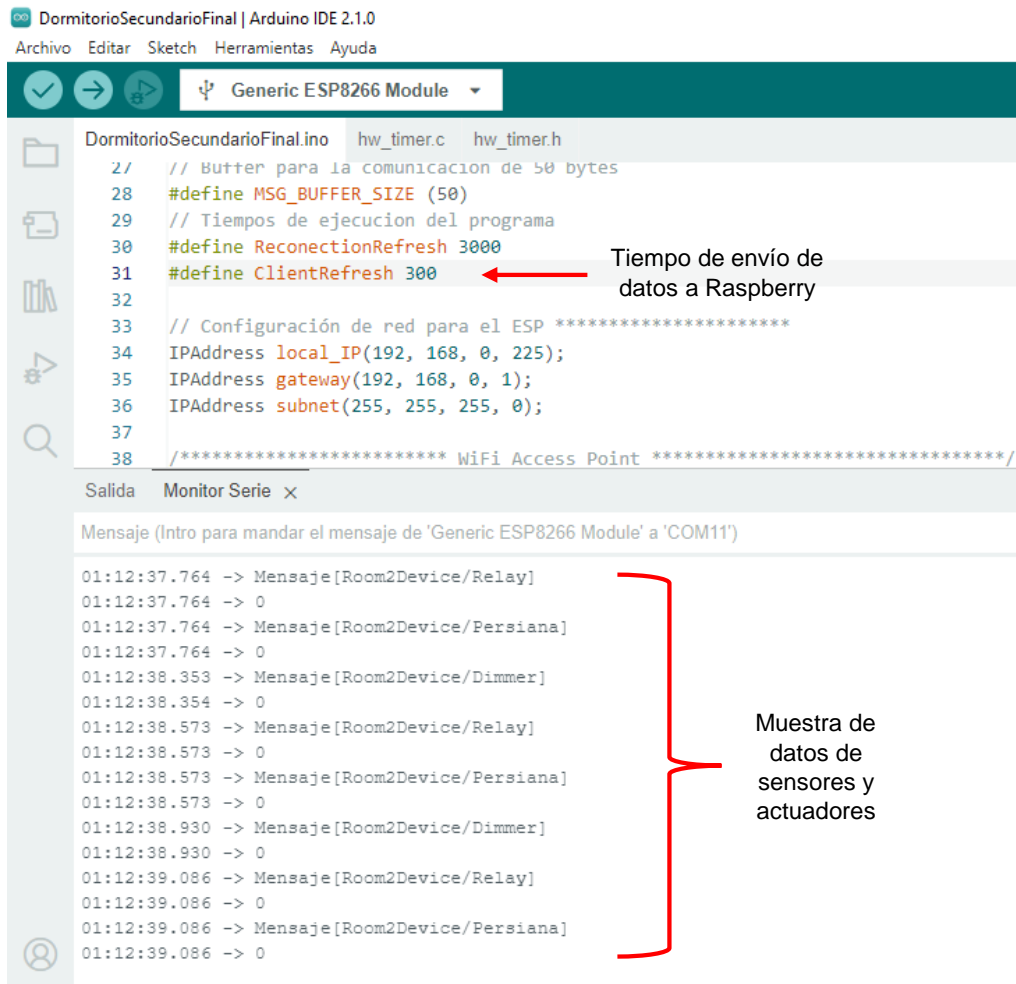


Fig. 80 Muestreo de datos de sensores y actuadores  
 Fuente: (Autor,2023)

- **Prueba de funcionamiento de comunicación entre la central de control Raspberry Pi 3 modelo B+ y la aplicación móvil**

Una vez que la placa Raspberry Pi recibe los datos de las otras placas ESP32 y ESP8266 los procesa y envía a la nube, lo cuales se pueden ver reflejados tanto en la plataforma Firebase y en la aplicación móvil, en la Figura 3.46 se puede apreciar el estado de algunos sensores y actuadores dentro del área de los dormitorios.

El tiempo de respuesta de comunicación entre la aplicación móvil y la tarjeta de control Raspberry Pi oscila entre 500 milisegundos a 1 segundo, dependiendo mucho de la velocidad de internet que se disponga.

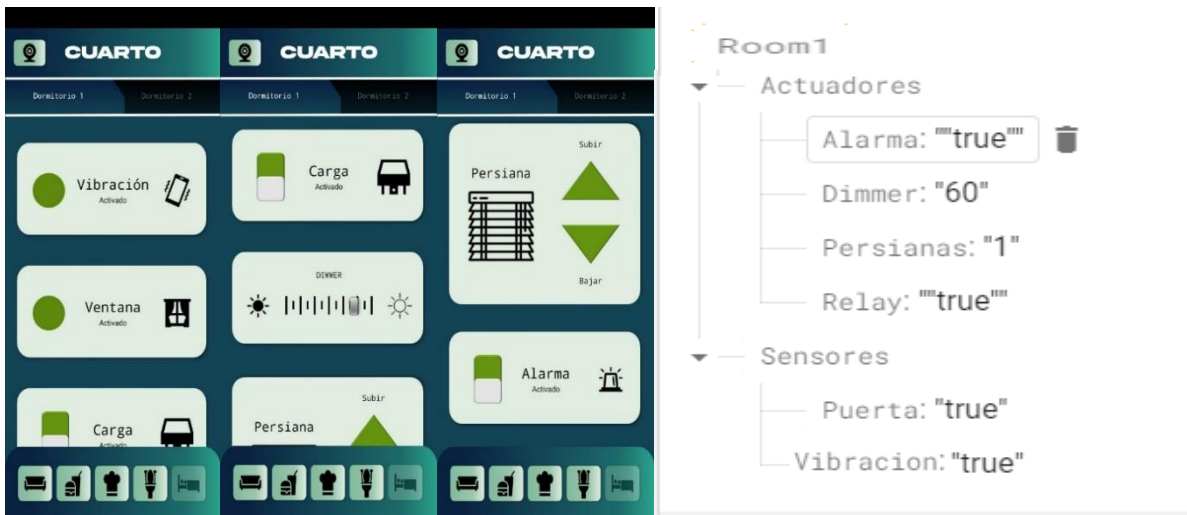


Fig. 81 Prueba de comunicación entre Raspberry Pi y Aplicación móvil reflejada en la plataforma Firebase  
Fuente: (Autor,2023)

- **Prueba de funcionamiento para el sistema de seguridad**

El sistema de seguridad esta conformado por varios sensores ubicados en sitios específicos dentro de la vivienda, una sirena y el envío de notificaciones de alerta a la aplicación y al celular por medio de la comunicación GSM, para llevar a cabo las pruebas de funcionamiento se activó algunos sensores que se encuentran dentro de la vivienda, al momento que el sistema detecta una señal de alerta se activa la sinera y se envía una notificación tanto a la aplicación móvil como al numero de celular del usuario como se muestra en la siguiente Figura 3.47.

El tiempo de recepción de los mensajes al teléfono celular del usuario oscila entre 1 a 2 minutos dependiendo de la cobertura de señal que disponga en ese instante.



Fig. 82 Prueba de funcionamiento sistema de seguridad  
Fuente: (Autor,2023)

- **Pruebas de funcionamiento para cámaras ESP32 CAM**

Las cámaras de seguridad se instalaron de acuerdo a los requerimientos del usuario, para este caso las dos cámaras se enfocan en la entrada principal de la vivienda, con el fin de monitorear a personas que ingresan a la vivienda. La Figura 3.48 muestra la interfaz dentro de la aplicación móvil de las 2 cámaras ESP32 CAM instaladas dentro de la vivienda.



Fig. 83 Prueba de funcionamiento cámaras ESP32 CAM  
Fuente: (Autor,2023)

- **Pruebas de funcionamiento para sistema de respaldo de energía**

. Para calcular el tiempo de respaldo de la UPS que se instaló para el prototipo domótico se realizó un cálculo del consumo de cada uno de los componentes que conforman el prototipo domótico expresado en Watts de potencia, aplicando la siguiente fórmula:

$$P = V * I \tag{1}$$

**Donde:**

P = Potencia [Watts]

V = Voltaje [Voltios]

I = Corriente [Amperios]

- **Cálculo del consumo de potencia de cada componente doméstico**

La siguiente tabla 3.6 describe el cálculo de potencia que se realizó a cada uno de los componentes que conforman el sistema domótico, los valores de voltaje y corriente de consumo se tomó de las hojas de especificaciones técnicas de los elementos que se encuentran en Anexos.

Tabla 21 Cálculo del consumo de potencia de los componentes domóticos

<b>Elemento</b>	<b>Cantidad de elementos instalados</b>	<b>Voltaje de operación [VDC]</b>	<b>Corriente de consumo [A]</b>	<b>Potencia individual consumida [W]</b>	<b>Potencia total consumida [W]</b>
Módulo relé	6	5	0.09	0.45	2.7
Detector de gas	1	5	0.15	0.75	0.75
Módulo dimmer	5	5	0.09	0.45	2.25
Detector de inundación	2	5	0.005	0.025	0.05
Sensor de temperatura	1	5	0.005	0.025	0.025
Detector de incendios	1	5	0.15	0.75	0.75
Sensor de vibración	4	5	0.015	0.075	0.3
Cámara Esp32 CAM	2	5	0.15	0.75	1.5
Motores persianas	3	5	0.055	0.275	0.83
Sirena	1	12	0.09	1.08	1.08
Módulo GSM	1	5	1	5	5
Tarjeta ESP32	2	3.3	0.17	0.561	1.12
Tarjeta ESP8266	4	3.3	0.18	0.594	2.38
Tarjeta de control Raspberry Pi 3 Modelo B+	1	5	3	10	10
<b>TOTAL</b>					<b>28.73</b>

Fuente: (Autor, 2023)

Con el resultado total obtenido del consumo de potencia del sistema domótico se realizó las pruebas de funcionamiento de la UPS para saber que tiempo de respaldo de energía brinda esta fuente en el caso que se interrumpiera la alimentación principal del sistema.

Como resultado se obtuvo un tiempo de 1 hora de respaldo de energía para todo el sistema domótico, lo cual es oportuno porque el sistema de seguridad conformado por los sensores, alarma y comunicación GSM sigue funcionando normalmente.

### 3.7 Análisis y resultados

Con las pruebas de funcionamiento unitarias y en conjunto que se realizó a cada uno de los elementos que conforman el prototipo domótico, se logró obtener un sistema domótico funcional que cumple con las expectativas señaladas anteriormente. El sistema se encuentra en condiciones óptimas para su uso dentro de viviendas reales y puede brindar al usuario una experiencia satisfactoria de automatización del hogar, gracias a su aplicación móvil que fue desarrollada exclusivamente para el presente prototipo.

Para un mejor funcionamiento del prototipo se cambió los tiempos de envío de datos a la nube con el fin de disminuir el tráfico en la red, dando como resultado una mejor velocidad de respuesta de todo el sistema domótico.

Con la implementación del sistema de respaldo de energía se obtuvo un respaldo de 1 hora en el caso de que falle la alimentación primaria de la vivienda o por motivos de corte de energía intencionados por parte de personas externas, el sistema quedará funcional durante este tiempo, permitiendo al usuario una actuación rápida ante cualquier altercado a la integridad de las personas o los bienes del hogar.

Con la implementación del sistema de comunicación GSM en conjunto con la alarma instalada dentro de la casa se logró que el sistema domótico cumpla con las condiciones de seguridad establecidas en los capítulos anteriores porque informa al usuario en tiempo real si existe algún problema de seguridad dentro de la vivienda, enviando mensajes SMS al número de celular que se encuentre dentro de la programación, permitiendo la ejecución oportuna de acciones por parte del usuario.

### 3.8 Costo del prototipo

Para la implementación del prototipo domótico se hizo uso de los componentes electrónico que se describió con anterioridad en el capítulo dos; donde se detallaba el precio aproximado de acuerdo con tiendas electrónicas, mercado libre, distribuidores eléctricos/electrónicos, etc. La siguiente Tabla 3.7 muestra los valores reales comerciales que presentan cada uno de los componentes electrónicos que se utilizaron para el prototipo de control domótico.

Tabla 22 Presupuesto prototipo domótico

Descripción	Cantidad	Precio unitario	Precio total
Módulo de relé JQC-3FF-S-Z	6	2.50	15.00
Sensor magnético MC-38	5	2.00	10.00
Sensor MQ2	1	4.50	4.50
módulo dimmer AC BTA16-600BT	5	20.00	100.00
Sensor capacitivo de nivel de líquido	2	3.00	6.00

Sensor de temperatura DS18B20	1	5.00	5.00
Sensor MQ-135	1	5.00	5.00
Sensor de vibración SW-420	4	3.00	12.00
ESP32-CAM	2	15.00	30.00
Motor paso a paso 28BYJ-48	3	7.00	21.00
Sirena tipo cuerno	1	8.00	8.00
Módulo GSM SIM800L	1	12.00	12.00
Tarjeta de control ESP8266	4	10.00	40.00
Tarjeta de control ESP32	2	12.00	24.00
Tarjeta de control Raspberry Pi 3 modelo B+	1	200.00	200.00
UPS R-UPR 758	1	50.00	50.00
Otros	1	100.00	100.00
			642.50

Fuente: (Autor, 2023)

El total de todos los componentes electrónicos es de 642.50 dólares americanos, pero a este valor se le debe añadir algunos elementos extras; como lo son cables de conexión, soportes, carcazas diseñadas en 3D, pernos, tornillos y mano de obra.



## **Conclusiones**

- Los tipos de sistemas domóticos que se pueden implantar en una vivienda, así como sus componentes electrónicos, aplicaciones y protocolos de comunicación, se llegaron a conocer gracias a la investigación realizada, esto sirvió de base para la creación de este prototipo de sistema de control domótico.
- Se logró desarrollar un prototipo domótico, compuesto por numerosos sensores y actuadores, que permite llevar a cabo actividades dirigidas a la seguridad dentro de una vivienda. Además, para este sistema se creó una aplicación móvil que permitió ejecutar y controlar las soluciones domóticas instaladas en cada una de las regiones de control.
- Una vez implementado el prototipo dentro de la vivienda, se procedió a ejecutar las pruebas de funcionamiento unitarias y en conjunto con el uso de la aplicación para smartphone, dando como resultado un prototipo domótico capaz de monitorear en tiempo real el estado de los dispositivos y sistemas conectados, lo que brinda tranquilidad y permite tomar acciones inmediatas si es necesario.

## **Recomendaciones**

- Hay sistemas mucho más antiguos que su precio de implementación es mas costoso que los sistemas que existen ahora y cuya instalación es mucho más difícil, por lo que es recomendable estudiar con más detalle los sistemas domóticos, sus componentes y sus aplicaciones. Hay que seleccionar un sistema que tenga en cuenta las necesidades del usuario.
- Para el diseño del prototipo domótico es fundamental tener en cuenta las características técnicas de los dispositivos que se vayan a utilizar tales como; tarjetas de control, sensores y actuadores, con el fin de obtener una mejor compatibilidad con el Software de programación que se vaya a utilizar.
- Basándose en las pruebas de funcionamiento realizadas al prototipo, se recomienda para futuros trabajos que se realice una evaluación previa de las instalaciones eléctricas del domicilio con el fin de determinar las condiciones y dimensionar adecuadamente los sensores y actuadores.
- Al ser un prototipo el sistema de control domótico está abierto a la posibilidad de mejora, es posible aumentar los niveles de seguridad con la implementación de un sistema de mando por voz, todo lo que se ejecuta mediante la aplicación móvil se lo podría controlar por medio de reconocimiento de voz que sería de gran ayuda para personas con capacidades especiales.

## 1 Referencias

- Adeel, A. (2016). IP network based home automation system using AVR microcontroller. *International Conference on Computing, Electronic and Electrical Engineering* (págs. 14-21). Cuba: Instituto de Ingenieros Eléctricos y Electrónicos Inc.
- Aguilar, C. (2017). Sistema IR para la activación térmica de fibra de carbono preimpregnada. (*Tesis de tecnología*). Centro de Ingeniería y desarrollo industrial, Santiago de Querétaro.
- Albert, L. (2020). *Scopus*. Obtenido de Scopus: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85088144904&origin=resultslist&sort=plf-f&src=s&st1=home+domotic&st2=&sid=d6bf4ca899cbf7e71e7620dfac4597d&sot=b&sdt=b&sl=27&s=TITLE-ABS-KEY%28home+domotic%29&relpos=17&citeCnt=0&searchTerm=#>
- Alvarado, K. (2018). *Diseño de un Sistema Domótico Basado en Tecnología Arduino para personas con discapacidad física*. Guayaquil. Obtenido de <http://repositorio.ug.edu.ec/bitstream/redug/36288/4/DISE%C3%91O%20DE%20UN%20SISTEMA%20DOM%C3%93TICO%20BASADO%20EN%20TECNOLOG%C3%8DA%20ARDUINO%20PARA%20PERSONAS%20CON%20DISCAPACIDAD%20F%C3%8DSICA-convertido.pdf>
- Anubhav, M. (2020). Artificial Intelligence for Smart Interiors - Colours, Lighting and Domotics. *International Conference on Reliability, Infocom Technologies and Optimization* (págs. 1335-1338). Dubai: Institute of Electrical and Electronics Engineers Inc.
- ARDUINO. (5 de Febrero de 2018). *Arduino*. Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- Asenjo, R., González, S., Corbera, F., Navarro, Á., Rodríguez, A., Villalba, J., & Hendrix, H. (Julio de 2017). *La plataforma Raspberry Pi como base para la coordinación vertical*. Obtenido de [https://www.researchgate.net/publication/319018535\\_La\\_plataforma\\_Raspberry\\_Pi\\_como\\_base\\_para\\_la\\_coordinacion\\_vertical](https://www.researchgate.net/publication/319018535_La_plataforma_Raspberry_Pi_como_base_para_la_coordinacion_vertical)
- Bouguera, T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., & Andrieux, G. (2018). Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. *Sensors*, 1-23.
- Boza, M. (2017). Sistema del Control Domótico y Confort de Edificaciones Modernas. (*Tesis de pregrado*). Universidad Cesar Vallejo, Perú.
- Bruno, A. (2019). *ESP32 NODE MCU*. Obtenido de [https://www.microelectronicash.com/downloads/ESP32\\_MANUAL.pdf](https://www.microelectronicash.com/downloads/ESP32_MANUAL.pdf)
- Cano, I., Paredes, M., Alor, G., Salas, M., & Segura, M. (2019). Smart Platform for Monitoring and Controlling Smart Home Devices and Sensors. *Technologies and Innovation* (págs. 145-171). Guayaquil: Board Members.
- Carrasco, M. (2016). *Sistemas de detección y alarmas*. Barcelona: Departament de Formació, Comunicació i Màrqueting.
- Ceja, J., Renteria, R., Ruelas, R., & Ochoa, G. (8 de Agosto de 2017). *Módulo ESP8266 y sus aplicaciones en el internet de las cosas*. Obtenido de [https://www.ecorfan.org/republicofperu/research\\_journals/Revista\\_de\\_Ingenieria\\_Electrica/vol1num2/ECORFAN\\_Revista\\_de\\_Ingenier%C3%ADa\\_EI%C3%A9ctrica\\_VI\\_N2\\_3.pdf](https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ingenieria_Electrica/vol1num2/ECORFAN_Revista_de_Ingenier%C3%ADa_EI%C3%A9ctrica_VI_N2_3.pdf)

- Components. (25 de Mayo de 2021). *MC-38 - Magnetic Switch for Home Alarm System*. Obtenido de <https://components101.com/sensors/mc38-magnetic-switch-sensor-pinout-features-datasheet-working-alternative-application>
- Comunicados. (22 de Enero de 2021). *Digital Sevilla*. Obtenido de <https://digitalsevilla.com/2021/01/22/10-aplicaciones-top-para-la-domotica-de-tu-hogar/>
- Condor, E., & De la Cruz, M. (2020). *Algoritmos resueltos con Phyton*. EIDEC.
- Cruz, V., & Collazos, J. (2016). Modelo de Infraestructura para Redes Mesh. (*Tesis de pregrado*). Universidad Libre, Bogotá. Obtenido de <https://repository.unilibre.edu.co/bitstream/handle/10901/10815/Tesis-Final.pdf?sequence=1>
- Cruz, W. (2018). Diseño del sistema de seguridad y de control de iluminación para el conjunto cerrado el portal del bosque en la ciudad de Tunja. (*Título de Pregrado*). Universidad Pedagógica y Tecnológica de Colombia, Colombia. Obtenido de <https://repositorio.uptc.edu.co/bitstream/001/2535/1/TGT-1119.pdf>
- Errobidart, J. (2017). Desconectado domoticasistema que utiliza comandos de voz. *Simposio y Congreso Argentino de Sistemas Embebidos* (págs. 25-30). Buenos Aires: Instituto de Ingenieros Eléctricos y Electrónicos Inc.
- Feng, C., & Luo, L. (8 de Agosto de 2018). *A Portable Readout System for Micro-pattern Gas Detectors and Scintillation Detectors*. Obtenido de [https://link.springer.com/chapter/10.1007/978-981-13-1313-4\\_41](https://link.springer.com/chapter/10.1007/978-981-13-1313-4_41)
- Fonseca, J., & Soria, D. (2020). Diseño e implementación de control domótico (con sistemas embebidos) para conectarse con aplicaciones adaptivas basados en IoT. (*Tesis de pregrado*). Universidad Politécnica Salesiana, Quito. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/18986/1/UPS%20-%20TTS056.pdf>
- Franco, G. (2020). Raspberry Pi, connectivity and programming through GPIO ports. *Ingeniería Innovativa*, 1-13. Obtenido de <https://www.unipiloto.edu.co/descargas/RaspberryPlypcDuino.pdf>
- Freepik. (2021). *Interfaz de asistente de domótica inteligente en una pantalla virtual*. Obtenido de [https://www.freepik.es/vector-premium/interfaz-asistente-domotica-inteligente-pantalla-virtual-concepto-sistema-control-automatizacion-closeup-smartphone-mano-estructura-metalica\\_7395937.htm](https://www.freepik.es/vector-premium/interfaz-asistente-domotica-inteligente-pantalla-virtual-concepto-sistema-control-automatizacion-closeup-smartphone-mano-estructura-metalica_7395937.htm)
- Game. (24 de Febrero de 2020). *Beneficios de la domótica para la accesibilidad*. Obtenido de <https://gametelecomunicaciones.com/beneficios-de-la-domotica-para-la-accesibilidad/>
- Gaotao, S., & Keqiu, L. (2017). *Signal Interference in WiFi and ZigBee Networks*. Springer International Publishing.
- García, E., Chavez, J., & Jurado, A. (29 de Marzo de 2017). *Modelado de una red desensores y actuadores inalámbrica para aplicaciones en agricultura de precisión*. Obtenido de <https://ieeexplore.ieee.org/document/7926210>
- García, V., & Vega, N. (2018). Low Power sensor node applied to domotic using IoT. *International Congress of Telematics and Computing*. Mexico: Springer Verlag.

- García, V., & Vega, N. (2 de Noviembre de 2018). *Low Power Sensor Node Applied to Domotic Using IoT*. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-030-03763-5\\_6](https://link.springer.com/chapter/10.1007/978-3-030-03763-5_6)
- Gazzawe, F., & Lock, R. (2018). *Devices, Applications and Their Potential Benefits*. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-030-01174-1\\_82](https://link.springer.com/chapter/10.1007/978-3-030-01174-1_82)
- Gómez, P. (23 de Octubre de 2020). *Cotrol de puertas con sensores WiFi inteligentes*. Obtenido de <https://topesdegama.com/listas/accesorios/sensores-wifi-inteligentes-puertas>
- Gracia, J. (2018). *Montaje de los cuadros de control y dispositivos eléctricos y electrónicos de los sistemas domóticos e inmóticos*. Recuperado de <https://elibro.net/es/ereader/utnorte/59255?page=12>.
- Gutiérrez, S. (2018). Domoticasistema de control de persianas con dispositivo móvil de comunicación zigbee. *Conferencia Internacional IEEE sobre Investigación en Inteligencia y Computación en Ingeniería*. El salvador : Instituto de Ingenieros Eléctricos y Electrónicos Inc.
- Gutierrez, S., Acero, C., & Rodrigo, P. (2018). Domotic control system for blinds with zigbee communication mobile device. *International Conference on Research in Intelligent and Computing in Engineering*. San Salvador, El Salvador. doi:10.1109/RICE.2018.8627899
- Hernández, L. d. (5 de Febrero de 2018). *Programar fácil*. Obtenido de <https://programarfacil.com/podcast/82-escoger-mejor-sensor-temperatura-arduino/>
- Herranz, Á. (2019). Desarrollo de aplicaciones para IoT con el módulo ESP32. *(Tesis de pregrado)*. Universidad de Alcalá, Madrid. Obtenido de [https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG\\_Benito\\_Herranz\\_2019.pdf?sequence=1&isAllowed=y](https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y)
- Herrera, D. (2020). Diseño e implementación de un prototipo de seguridad para control domótico basado en IOT bajo ambientes de dispositivos móviles con Android. *(Tesis de pregrado)*. Escuela Politécnica Nacional, Quito. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/20857/1/CD%2010380.pdf>
- Hidalgo Constante, B. (2017). *Elaboración de un instructivo de instalación de un sistema domótico para clientes PYMES*. Guayaquil. Obtenido de <http://repositorio.ucsg.edu.ec/bitstream/3317/9239/1/T-UCSG-PRE-TEC-ITEL-253.pdf>
- Ideastronic. (15 de Marzo de 2021). *MQ-6 sensor de gas*. Obtenido de <https://www.ideastronic.com/producto/mq-6-sensor-de-gas/>
- IoTsmart. (2016). *IoTsmart Home Automation*. Obtenido de <http://www.iotsmarth.com/producto/sensor-de-luz/>
- Joy-it. (2020). *Esp32 camera module*. Obtenido de [https://elektronik-lavpris.dk/files/sup224/148358\\_1616149541.pdf](https://elektronik-lavpris.dk/files/sup224/148358_1616149541.pdf)
- Jurado, G. (2018). SISTEMA DE CCTV CON CÁMARAS IP Y CENTRAL DE ALARMA CON RADIO ENLACES DE MICROONDAS PARA CENTRO DE SALUD INTEGRAL BELLA VISTA. *(Tesis de grado, Universidad Mayor de San Andrés)*. Obtenido de <http://repositorio.umsa.bo/xmlui/handle/123456789/18537>
- Lahera, G. (22 de Mayo de 2022). *Selectra*. Obtenido de <https://selectra.es/alarmas/accesorios-alarmas/detector-inundacion>

- Limones, E. (7 de Abril de 2021). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/topologia-de-redes-informaticas/>
- Llamas, L. (22 de Diciembre de 2016). *Conmutar cargas con Arduino y relé de estado sólido*. Obtenido de <https://www.luisllamas.es/arduino-rele-estado-solido-ssr/>
- López, C. (2021). *¿Qué es la red GSM y cómo funciona?* Obtenido de <https://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>
- López, S. (2020). *DIGITAL55*. Obtenido de Firebase: qué es, para qué sirve, funcionalidades y ventajas : <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- Maldonado, H., & Chacha, E. (2017). Implementación de dos módulos técnicos para prácticas de domótica e inmótica mediante protocolos de comunicación x10. (*Tesis de pregrado*). Escuela Superior Politécnica de Chimborazo, Chimborazo.
- Matthew, D. (Agosto de 2021). *Computer Weekly*. Obtenido de <https://www.computerweekly.com/es/definicion/Desarrollo-de-aplicaciones-moviles>
- Megatrónica. (2022). *Módulo Sim800 V2.0 Sim800l Gsm Gprs Arduino*. Obtenido de <https://megatronica.cc/producto/modulo-sim800-sim800l-gsm-gprs-arduino-v2/>
- Montes, L. (2018). *Seguridad Compartida* . Obtenido de <https://seguridadcompartida.mx/sensores-de-alarmas/>
- Moral, M. (2021). NODE-RED COMO HERRAMIENTA VISUAL DE DISPOSITIVOS IoT. (*Tesis de pregrado, Universidad de Jaén*). Obtenido de [https://tauja.ujaen.es/bitstream/10953.1/15335/1/Memoria\\_TFG\\_Miguel\\_Angel\\_Moral\\_Perez.pdf](https://tauja.ujaen.es/bitstream/10953.1/15335/1/Memoria_TFG_Miguel_Angel_Moral_Perez.pdf)
- Nahim de Anda, M. (24 de Octubre de 2018). *Sensor de luz* . Obtenido de <https://www.factor.mx/portal/base-de-conocimiento/sensor-de-luz/>
- Ohms, 3. (11 de Julio de 2016). *Como funcionan los sensores de gas*. Obtenido de <https://blog.330ohms.com/2016/07/11/como-funcionan-los-sensores-de-gas/>
- Olsson, M. (2016). *PHP 7 Quick Scripting Reference*. Hammarland: Apress, Berkeley, CA.
- Oreja, C. D. (10 de Mayo de 2019). *Innovación seguridad electrónica*. Obtenido de [https://revistainnovacion.com/nota/10467/nociones\\_basicas\\_de\\_un\\_sistema\\_de\\_deteccion\\_de\\_incendios/](https://revistainnovacion.com/nota/10467/nociones_basicas_de_un_sistema_de_deteccion_de_incendios/)
- Ovalles, F. (2018). Uso de un sistema integrado con tecnología wifi para domoticontrol de entornos convencionales. *Encuentro Internacional de Ciencias Aplicadas e Ingeniería*. Colombia: Institute of Physics Publishing.
- Pathak, A. (Julio de 2021). *Geekflare*. Obtenido de <https://geekflare.com/es/understanding-network-topology/>
- Patil, S., Bidari, I., & Sunag, B. (2016). Application of HMI technology in automotive sector. *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECOT)*, (págs. 322-324). Mysuru, India. doi:10.1109/ICEECOT.2016.7955238

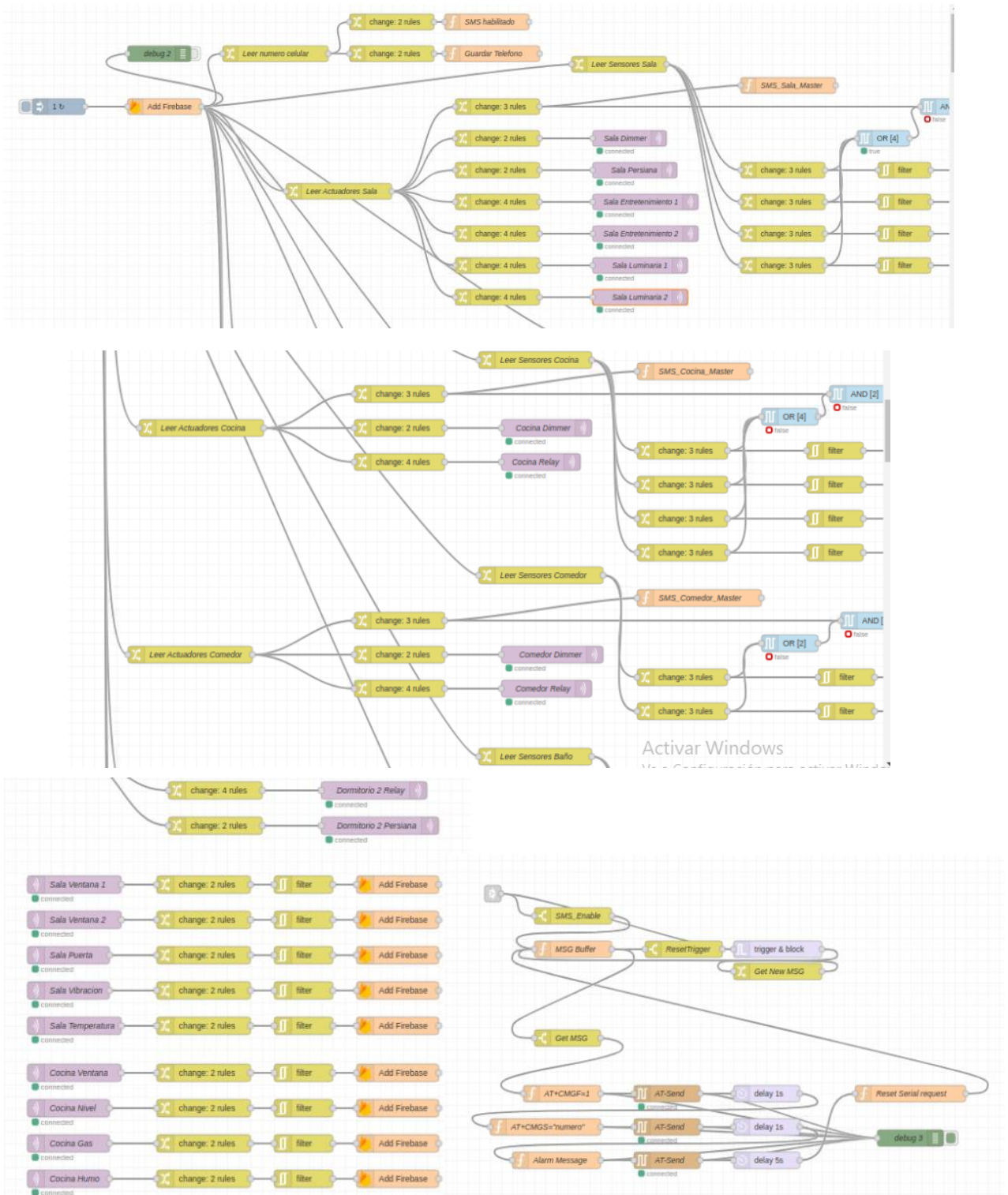
- Pizarro, E. (2017). *Desarrollo de aplicaciones móviles con App Inventor*. Obtenido de [https://ecad2017.udd.cl/files/2017/12/Desarrollo-de-Aplicaciones-M%C3%B3viles-con-App-Inventor\\_ULS.pdf](https://ecad2017.udd.cl/files/2017/12/Desarrollo-de-Aplicaciones-M%C3%B3viles-con-App-Inventor_ULS.pdf)
- Prastyo, E. (2020). *edukasielektronika*. Obtenido de <https://www.edukasielektronika.com/2020/10/water-level-sensor-k-0135.html>
- Quispe, O. (2019). SISTEMA DE ALIMENTACION ININTERRUMPIDA (UPS) TIPO OFF-LINE CON EL ATMEGA 328p COMO SISTEMA DE TRANSFERENCIA PARA DISPOSITIVOS SENSIBLES DE BAJA POTENCIA. (*Tesis de Pregrado*). Obtenido de <https://repositorio.umsa.bo/bitstream/handle/123456789/28017/EG-2352.pdf?sequence=1&isAllowed=y>
- Raspberrypi. (Junio de 2019). *Static*. Obtenido de <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>
- Rego, R., & Semente, R. (16 de Diciembre de 2020). *Presence sensors: a comparasion between the ultrasonic and the infrared in the detection of people and vehicles*. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-030-57548-9\\_54](https://link.springer.com/chapter/10.1007/978-3-030-57548-9_54)
- Robotesfera. (2020). *Aplicaciones de la domótica*. Obtenido de <https://robotesfera.com/que-es-domotica-aplicaciones#Iluminacion>
- Ruque, J. (2016). Estudio del estado del arte de la domótica en la ciudad de Loja. (*Tesis de pregrado*). Universidad de Guayaquil, Guayaquil.
- Sánchez, M. (2017). *Montaje de los cuadros de control y dispositivos eléctricos y electrónicos de los sistemas domóticos e inmóticos*. Recuperado de <https://elibro.net/es/ereader/utnorte/106575?page=29>.
- Semente, R., & Rego, R. (2018). *Presence sensor for people detection and reduction of pedestrian waiting time in traffic light*. Obtenido de <https://www.lcv.fee.unicamp.br/images/BTSym18/Papers/030.pdf>
- Suescun, P. (2020). Use of ANN for embedded domotic system based on IoT. *International Journal of Mathematical, Engineering and Management Sciences*, V(5), 971-984.
- Sun, D.-Z., Mu, Y., & Susilo, W. (2018). rrectioMan in the middle attacks on Secure Simple Pairing in Bluetooth standard V5.0 and its countermeasure. *Personal and Ubiquitous Computing*, 22-69.
- Tanaka, Y., & Fujimoto, T. (2019). Design of the External Smartphone Interface to Prevent Users' Mistapping on Advertisements and Eliminate Difficulty of Reading. *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*, (págs. 995-1000). Toyama, Japan. doi:10.1109/IIAI-AAI.2019.00199
- Torres, J. (2021). Aplicación móvil multiplataforma para la gestión de información georeferencial y servicio técnico comunitario de plomería, aplicando geolocalización offline, en la junta administradora de agua potable de los barrios occidentales de Aloasí. (*Tesis de pregrado, Universidad Técnica de Ambato*). Obtenido de <https://repositorio.uta.edu.ec/bitstream/123456789/32046/1/t1754si.pdf>
- Tubón Cando, G. (2020). Aplicación móvil con Georreferenciación. (*Título de Ingeniería*). Pontifica Universidad Católica del Ecuador, Ambato.

- Varper. (8 de Abril de 2021). *Dimmer 1 canal Triac AC 3.3V/5V AC 50/60hz 220V/110V Arduino*.  
Obtenido de <https://varpertechnologies.com/producto/dimmer-1-canal-triac-ac-3-3v-5v-ac-50-60hz-220v-110v-arduino/?v=5bc574a47246>
- Vázquez, J. (2019). Diseño e implementación de un sistema de control y monitoreo inalámbrico mediante radiofrecuencia, Bluetooth y SMS en aplicaciones domóticas. (*Tesis de maestría*).  
Universidad Católica de Santiago de Guayaquil, Guayaquil.
- Vinueza, D. (2019). Implementación de un prototipo de sistema domótico basado en la plataforma arduino gestionado a través del internet. (*Tesis de pregrado*). Escuela Politécnica Nacional, Quito. Obtenido de <https://bibdigital.epn.edu.ec/bitstream/15000/19952/1/CD-9412.pdf>
- Went, Z. (31 de Mayo de 2017). *Relés de estado sólido y relés electromecánicos*. Obtenido de El sistema global de comunicaciones móviles (GSM) es un sistema de telefonía celular para la transmisión digital de voz y datos con una gran calidad que se está extendiendo por todo el mundo. Para su correcto funcionamiento, debe disponer de un teléfono c
- Yépez, M. (2018). La domótica en el diseño de módulos comerciales. (*Tesis de pregrado*).  
Universidad Técnica de Ambato, Ambato. Obtenido de <http://repositorio.uta.edu.ec/handle/123456789/28958>



# Anexos

## Anexo 1: Código de programación para tarjeta de control Raspberry Pi 3 Modelo B+



## Anexo 2: Código de programación para tarjeta de control del espacio físico cocina

```
/*
*****
Cliente 1 - Cocina
*****
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "hw_timer.h"

/*
***** Pin Definition *****
//Relays for switching appliances
#define dimmer 4 //D2
#define relay 0 //D3
#define ventanaSensor 16 //D0
#define gasSensor 13 //D7
#define aguaSensor 12 //D6
#define humoSensor 14 //D5
#define zeroCrossing 5 //D1
// Tag actuadores *****
#define Dimmer "CocinaDevice/Dimmer"
#define Relay "CocinaDevice/Relay"
// Tag sensores
#define Ventana "CocinaDevice/Ventana" // Topic del sensor de ventana
#define Nivel "CocinaDevice/Nivel" // Topic del sensor de nivel
#define Humo "CocinaDevice/Humo" // Topic del sensor de Humo
#define Gas "CocinaDevice/Gas" // Topic del sensor de gas
// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000
#define ClientRefresh 300
#define VentanaHoldRefresh 3000

// Configuración de red para el ESP *****
IPAddress local_IP(192, 168, 0, 223);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);

/*
***** WiFi Access Point
*****
const char* ssid = "TP-Link_27E0";
const char* password = "66608189";
const char* mqtt_server = "192.168.0.230"; //IP address of RPi
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;

```

```

volatile unsigned long ClientTime = 0;
volatile unsigned long VentanaHoldTime = 0;

//char// Variables de uso general ***** msg[MSG_BUFFER_SIZE];
String val_dimm = "";
volatile boolean liquid_detect = false;
volatile boolean smoke_detect = false;
volatile boolean gas_detect = false;
volatile boolean ventana_detect = false;
volatile boolean ventanaBuffer = false;
// Variables del Dimmer
byte fade = 0;
byte state = 1;
byte tarBrightness = 255;
byte curBrightness = 0;
byte zcState = 0; // 0 = ready; 1 = processing;

// Objeto para la comunicacion MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Lee los sensores
void ReadSensor() {
  //Función para el Sensor de Líquido.
  if (!digitalRead(aguaSensor)) {
    Serial.println("Líquido Detectado!");
    liquid_detect = true;
  } else liquid_detect = false;

  //Función para el Sensor de Humo.
  if (!digitalRead(humoSensor)) {
    Serial.println("Humo Detectado!");
    smoke_detect = true;
  } else smoke_detect = false;

  //Función para el Sensor de Gas.
  if (!digitalRead(gasSensor)) {
    Serial.println("Gas Detectado!");
    gas_detect = true;
  } else gas_detect = false;

  //Funcion para el sensor de vibracion
  if (digitalRead(ventanaSensor) != ventanaBuffer) {
    ventanaBuffer = digitalRead(ventanaSensor);
    Serial.println("Ventana Detectada!");
    ventana_detect = true;
    VentanaHoldTime = CurrentTime;
  }
}

```

```

}

// Funcion de reconexión
void reconnect() {
  // Si esta conectado a la red, no se realiza nada
  if (client.connected()) {
    return;
  }
  Serial.print("Intentando conexión MQTT...");
  String clientId = "ESP8266Cocina";
  clientId += String(random(0xffff), HEX);
  //Intento de Conexión
  if (client.connect(clientId.c_str(), username, pass)) {
    Serial.println("Conectado");
    //Una vez conectado, publica un anuncio...
    // ... y vuelve a suscribir.
    client.subscribe(Dimmer);
    client.subscribe(Relay);
    Serial.println(client.state());
  } else {
    Serial.print("Fallo, rc=");
    Serial.println(client.state());
    Serial.println("Intente nuevamente en 5 segundos");
  }
}

// Handler de envio de datos a la raspberry
void ClientHandler() {
  // Serial.println("Publicar mensaje ");
  // Serial.print("Nivel de agua: ");
  // if (liquid_detect) Serial.println("Alto");
  // else Serial.println("Bajo");
  // dtostrf(liquid_detect, 1, 2, str_liquid);
  if (liquid_detect) client.publish(Nivel, "true");
  else client.publish(Nivel, "false");

  // Serial.print("Nivel de humo: ");
  // if (smoke_detect) Serial.println("Alto");
  // else Serial.println("Bajo");
  // dtostrf(smoke_detect, 1, 2, str_smoke);
  if (smoke_detect) client.publish(Humo, "true");
  else client.publish(Humo, "false");

  // Serial.print("Nivel de gas: ");
  // if (gas_detect) Serial.println("Alto");
  // else Serial.println("Bajo");
  // dtostrf(gas_detect, 1, 2, str_gas);
  if (gas_detect) client.publish(Gas, "true");
  else client.publish(Gas, "false");
}

```

```

// Serial.print("Nivel de vibracion: ");
// if (ventana_detect) Serial.println("Alta");
// else Serial.println("Baja");
// dtostrf(ventana_detect, 1, 2, str_ventana);
if (ventana_detect) client.publish(Ventana, "true");
else client.publish(Ventana, "false");
}
// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
// Lee el mensaje que ha recibido
Serial.print("Mensaje["");
Serial.print(topic);
Serial.println("]");
for (int i = 0; i < length; i++) {
Serial.print((char)payload[i]);
}
Serial.println();

// Si el mensaje corresponde al control del rele, entonces se realiza la
accion correspondiente
if (strstr(topic, Relay)) {
if ((char)payload[0] == '1') digitalWrite(relay, HIGH);
else digitalWrite(relay, LOW);
}

// Si el mensaje corresponde al control del dimmer, entonces se realiza el
control del dimmer
else if (strstr(topic, Dimmer)) {
val_dimm = "";
// Obtiene el dato del dimmer en str
for (int i = 0; i < length; i++) {
val_dimm += payload[i] - 48;
}
// Convierte de str a int para poder trabajar con el valor
int val = val_dimm.toInt();
tarBrightness = val;
// Serial.print("El valor del dimmer es: ");
// Serial.println(tarBrightness);
}

else {
Serial.println("Tema no suscrito");
}
}

ICACHE_RAM_ATTR void dimTimerISR() {

```

```

    // Fade == 1: Significa que se llegara al valor por medio de un
    desvanecimiento
    if (fade == 1) {
        if (curBrightness > tarBrightness && curBrightness > 0) {
            --curBrightness;
        } else if (curBrightness < tarBrightness && curBrightness < 255) {
            ++curBrightness;
        }
    }
    // Fade == 0: Significa que se llegara al valor de forma directa
    } else {
        if (tarBrightness != 0) {
            curBrightness = tarBrightness;
        } else {
            curBrightness = 0;
        }
    }
}

// Enciende o apaga el pin del dimmer para producir el control de
luminosidad
if (curBrightness == 0) {
    digitalWrite(dimmer, 0);
} else {
    digitalWrite(dimmer, 1);
}

// Reestablece la bandera de zerocrossing
zcState = 0;
}

ICACHE_RAM_ATTR void zcDetectISR() {
    // Si se detecta un zero crossing entonecs se inicia el proceso de control
    if (zcState == 0) {
        zcState = 1;
        if (curBrightness < 255 && curBrightness > 0) {
            digitalWrite(dimmer, 0);

            int dimDelay = 30 * (255 - curBrightness) + 400; //400
            hw_timer_arm(dimDelay);
        }
    }
}

void setup() {
    // Configuración inicial de entradas y salidas
    pinMode(relay, OUTPUT);
    pinMode(ventanaSensor, INPUT);
    pinMode(gasSensor, INPUT);
    pinMode(aguaSensor, INPUT);
}

```

```

pinMode(humoSensor, INPUT);
pinMode(zeroCrossing, INPUT);
pinMode(dimmer, OUTPUT);
// Estado inicial de las salidas
digitalWrite(relay, LOW);
// Configuración de la comunicación serial
Serial.begin(115200);
// Configuración de red *****+
WiFi.config(local_IP, gateway, subnet);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
// Conexión con la red
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// IP conectada
Serial.println("");
Serial.println("WiFi connected");
Serial.print("Use: ");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
// Configuración del servidor mqtt
client.setServer(mqtt_server, port);
client.setCallback(callback);
reconnect(); // Se suscribe al topic establecido para la recepción de los
datos

attachInterrupt(zeroCrossing, zcDetectISR, RISING); // Attach an Interrupt
to Pin 2 (interrupt 0) for Zero Cross Detection
hw_timer_init(NMI_SOURCE, 0);
hw_timer_set_func(dimTimerISR);
}

void loop() {
    CurrentTime = millis();
    // Lee los sensores
    ReadSensor();

    // Controlador de reconexión *****
    if (client.connected()) {
        ReconnectTime = CurrentTime;
    } else {
        if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
            reconnect();
            ReconnectTime = CurrentTime;
        }
    }
}

```

```
// Controlador de envio de datos *****
if ((CurrentTime - ClientTime) > ClientRefresh) {
    ClientHandler();
    ClientTime = CurrentTime;
}

// Controlador de tiempo de activacion del sensor de vibracion
*****
if (ventana_detect) {
    if ((CurrentTime - VentanaHoldTime) > VentanaHoldRefresh) {
        ventana_detect = false;
    }
}
client.loop();
}
```



### Anexo 3: Código de programación para tarjeta de control del espacio físico sala

```
/*
*****
Cliente 5 - Sala
*****
#include <WiFi.h>
#include <PubSubClient.h>
#include <Stepper.h>
#include <OneWire.h>
#include <DallasTemperature.h>

/****** Pin Definition *****/

#define dimmer 18
#define luminaria1 23
#define luminaria2 22
#define vibrationSensor 21
#define puertaSensor 33
#define ventanaSensor1 17
#define ventanaSensor2 14
#define entretenimiento1 26
#define entretenimiento2 25
#define oneWireBus 32
#define IN1 4
#define IN2 2
#define IN3 0
#define IN4 15
#define zeroCrossing 19
#define STEPS 2048 // Pasos para dar la vuelta completa en modo 4
pasos
#define StepSpeed 18 // RPM
#define MAXMotorTurn 20 // Vueltas maximas
#define MAXMotorStep MAXMotorTurn* STEPS
// Tag actuadores *****
#define Dimmer "SalaDevice/Dimmer"
#define Entretenimiento1 "SalaDevice/Entretenimiento1"
#define Entretenimiento2 "SalaDevice/Entretenimiento2"
#define Luminaria1 "SalaDevice/Luminaria1"
#define Luminaria2 "SalaDevice/Luminaria2"
#define Persiana "SalaDevice/Persiana"
// Tag sensores
#define Ventana1 "SalaDevice/Ventana1" // Topic del sensor de ventana
#define Ventana2 "SalaDevice/Ventana2" // Topic del sensor de ventana
#define Vibracion "SalaDevice/Vibracion" // Topic del sensor de ventana
#define Puerta "SalaDevice/Puerta" // Topic del sensor de nivel
#define Temperatura "SalaDevice/Temperatura" // Topic del sensor de nivel
// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000 // eN MILISEGUNDOS
```

```

#define ClientRefresh 300          // Envio de datos a raspberry cada 300mS
#define TemperatureRefresh 2000
#define VibrationHoldRefresh 5000 // 3 segundos mantiene la señal
// Configuración de red para el ESP *****
IPAddress local_IP(192, 168, 0, 226); // nota cambiar solo el tercer dígito de
esta ip para router nuevos
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);

/***** WiFi Access Point
*****/
const char* ssid = "TP-Link_27E0";
const char* password = "66608189";
const char* mqtt_server = "192.168.0.230"; //IP address of RPi, para
conectarse a mi red cambio la IP el tercer dígito solo ese lo demas queda
igual
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;
volatile unsigned long TemperatureTime = 0;
volatile unsigned long ClientTime = 0;
volatile unsigned long VibrationHoldTime = 0;

// Variables de uso general *****
//char msg[MSG_BUFFER_SIZE];
String val_dimm = "";
int CurrentStep = 0;
int MoveMotor = 0;
float temperatureC = 0;
volatile boolean door_detect = false;
volatile boolean vibration_detect = false;
volatile boolean vibration_buffer = false;
volatile boolean ventana_detect1 = false;
volatile boolean ventana_detect2 = false;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

// Variables del Dimmer
byte fade = 0; // desplazamiento suave hacia el valor del dimmer
byte state = 1;
byte tarBrightness = 0;
byte curBrightness = 0;
byte zcState = 0; // 0 = ready; 1 = processing;

// Objeto para la comunicacion MQTT

```

```

WiFiClient espClient;
PubSubClient client(espClient);

// Objeto del temporizador
hw_timer_t* timer = NULL;

// Objeto para el control del motor de paso
Stepper stepper(STEPS, IN1, IN2, IN3, IN4);

// Objeto OneWire para comunicarse con el sensor DS18B20
OneWire oneWire(oneWireBus);
DallasTemperature temperature_sensor(&oneWire);

DeviceAddress TempSensor = { 0x28, 0x18, 0x92, 0x79, 0xA2, 0x00, 0x03, 0xB0 };

// Controlador de interrupcion para el control del dimmer
ICACHE_RAM_ATTR void onTimer(){
    portENTER_CRITICAL(&timerMux);
    // Fade == 1: Significa que se llegara al valor por medio de un
desvanecimiento
    if (fade == 1) {
        if (curBrightness > tarBrightness && curBrightness > 0) {
            --curBrightness;
        } else if (curBrightness < tarBrightness && curBrightness < 255) {
            ++curBrightness;
        }
        // Fade == 0: Significa que se llegara al valor de forma directa
    } else {
        if (tarBrightness != 0) {
            curBrightness = tarBrightness;
        } else {
            curBrightness = 0;
        }
    }
    // Enciende o apaga el pin del dimmer para producir el control de
luminosidad
    if (curBrightness == 0) {
        digitalWrite(dimmer, 0);
    } else {
        digitalWrite(dimmer, 1);
    }
    // Reestablece la bandera de zerocrossing
    zcState = 0;
    portEXIT_CRITICAL(&timerMux);
}

// Detector de cruce zero
ICACHE_RAM_ATTR void zcDetectISR() {

```

```

//portENTER_CRITICAL(&timerMux);
// Si se detecta un zero crossing entonecs se inicia el proceso de control
if (zcState == 0) {
    zcState = 1;
    if (curBrightness <= 255 && curBrightness >= 0) {
        digitalWrite(dimmer, 0);

        uint32_t dimDelay = 30 * (255 - curBrightness) + 400; //400
        timerWrite(timer, 0);
        timerAlarmWrite(timer, dimDelay, false);
        timerAlarmEnable(timer);
    }
}
}

// Funcion de lectura de los sensores
void ReadSensor() {
    //Función para el Sensor de la ventana 1
    if (digitalRead(ventanaSensor1)) {
        Serial.println("Ventana 1 detectada!");
        ventana_detect1 = true;
    } else ventana_detect1 = false;
    ///Función para el Sensor de la ventana 2
    if (digitalRead(ventanaSensor2)) {
        Serial.println("Ventana 2 detectada!");
        ventana_detect2 = true;
    } else ventana_detect2 = false;
    //Función para el Sensor de Puerta.
    if (digitalRead(puertaSensor)) {
        Serial.println("Puerta abierta!");
        door_detect = true;
    } else door_detect = false;
    //Función para el Sensor de ventana.
    if (digitalRead(vibrationSensor) != vibration_buffer) {
        vibration_buffer = digitalRead(vibrationSensor);
        Serial.println("Vibracion Detectada!");
        vibration_detect = true;
        VibrationHoldTime = CurrentTime;
    }
}

// Handler de envio de datos a la raspberry
void ClientHandler() {
    // Serial.println("Publicar mensaje ");
    // Serial.print("Estado de la puerta: ");
    // dtostrf(door_detect, 1, 2, str_door);
    char Temp2Send[10];
    dtostrf(temperatureC, 1, 2, Temp2Send);
}

```

```

client.publish(Temperatura, Temp2Send);

if (vibration_detect) client.publish(Vibracion, "true");
else client.publish(Vibracion, "false");

if (door_detect) client.publish(Puerta, "true");
else client.publish(Puerta, "false");

// Serial.print("Nivel de vibracion: ");
if (ventana_detect1) client.publish(Ventana1, "true");
else client.publish(Ventana1, "false");

if (ventana_detect2) client.publish(Ventana2, "true");
else client.publish(Ventana2, "false");
}

// Funcion de reconexión
void reconnect() {
  // Si esta conectado a la red, no se realiza nada
  if (client.connected()) {
    return;
  }
  Serial.print("Intentando conexión MQTT...");
  String clientId = "ESP32Sala";
  clientId += String(random(0xffff), HEX);
  //Intento de Conexión
  if (client.connect(clientId.c_str(), username, pass)) {
    Serial.println("Conectado");
    //Una vez conectado, publica un anuncio...
    // ... y vuelve a suscribir.
    client.subscribe(Entretenimiento1);
    client.subscribe(Entretenimiento2);
    client.subscribe(Luminaria1);
    client.subscribe(Luminaria2);
    client.subscribe(Dimmer);
    client.subscribe(Persiana);
    Serial.println(client.state());
  } else {
    Serial.print("Fallo, rc=");
    Serial.println(client.state());
    Serial.println("Intente nuevamente en 5 segundos");
  }
}

// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Mensaje[");
  Serial.print(topic);
  Serial.println("]");
}

```

```

// Si el mensaje corresponde al control del entretenimiento, entonces se
realiza la accion correspondiente
if (strstr(topic, Entretenimiento1)) {
    if ((char)payload[0] == '1') digitalWrite(entretenimiento1, LOW);
    else digitalWrite(entretenimiento1, HIGH);
}
// Si el mensaje corresponde al control del entretenimiento, entonces se
realiza la accion correspondiente
else if (strstr(topic, Entretenimiento2)) {
    if ((char)payload[0] == '1') digitalWrite(entretenimiento2, LOW);
    else digitalWrite(entretenimiento2, HIGH);
}
// Si el mensaje corresponde al control de la luminaria, entonces se realiza
la accion correspondiente
else if (strstr(topic, Luminaria1)) {
    if ((char)payload[0] == '1') digitalWrite(luminaria1, LOW);
    else digitalWrite(luminaria1, HIGH);
}
// Si el mensaje corresponde al control del entretenimiento, entonces se
realiza la accion correspondiente
else if (strstr(topic, Luminaria2)) {
    if ((char)payload[0] == '1') digitalWrite(luminaria2, LOW);
    else digitalWrite(luminaria2, HIGH);
}
// Si el mensaje corresponde al control del dimmer, entonces se realiza la
accion correspondiente
else if (strstr(topic, Dimmer)) {
    val_dimm = "";
    for (int i = 0; i < length; i++) {
        val_dimm += payload[i] - 48;
    }
    //int val = map(val_dimm.toInt(), 0, 255, 0, 100);
    int val = val_dimm.toInt();
    Serial.print("El valor del dimmer es: ");
    Serial.println(val);
    portENTER_CRITICAL(&timerMux);
    tarBrightness = val;
    portEXIT_CRITICAL(&timerMux);
    //DimmerControl.setPower(val);
}
// Si el mensaje corresponde al control de persianas, entonces se realiza la
accion correspondiente
else if (strstr(topic, Persiana)) {
    if ((char)payload[0] == '1') {
        // Si la persiana esta totalmente subida, no se le permite subir más
        if (CurrentStep > 0) {
            MoveMotor = 1;

```

```

    } else {
        MoveMotor = 0;
    }
} else if ((char)payload[0] == '2') {
    // Si la persiana esta totalmente bajada, no se le permite bajar más
    if (CurrentStep < MAXMotorStep) {
        MoveMotor = 2;
    } else {
        MoveMotor = 0;
    }
} else {
    MoveMotor = 0;
    return;
}
Serial.print("El Motor esta en la vuelta: ");
Serial.println(round((CurrentStep * 100 / STEPS)) / 100);
} else {
    Serial.println("Tema no suscrito");
}
}
// Controla el movimiento del motor *****
void MotorHandler() {
    if (MoveMotor == 0) {
        // Apaga las bobinas del motor para evitar el sobrecalentamiento. Si es
        // necesario, se puede comentar este bloque if si se necesita mantener el torque
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
    }
    if (MoveMotor == 1) {
        stepper.step(1);
        CurrentStep--;
    }
    if (MoveMotor == 2) {
        stepper.step(-1);
        CurrentStep++;
    }
}
void setup() {
    // Configuración inicial de entradas y salidas
    pinMode(dimmer, OUTPUT);
    pinMode(entretenimiento1, OUTPUT);
    pinMode(entretenimiento2, OUTPUT);
    pinMode(luminaria1, OUTPUT);
    pinMode(luminaria2, OUTPUT);
    pinMode(ventanaSensor1, INPUT_PULLUP);
    pinMode(ventanaSensor2, INPUT_PULLUP);
}

```

```

pinMode(puertaSensor, INPUT_PULLUP);
pinMode(vibrationSensor, INPUT);
pinMode(zeroCrossing, INPUT);
// Estado inicial de las salidas
digitalWrite(entretenimiento1, HIGH);
digitalWrite(entretenimiento2, HIGH);
digitalWrite(luminaria1, HIGH);
digitalWrite(luminaria2, HIGH);
// Configuración de la comunicación serial
Serial.begin(115200);
// Configuración de red *****+
WiFi.config(local_IP, gateway, subnet);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
// Conexión con la red
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// IP conectada
Serial.println("");
Serial.println("WiFi connected");
Serial.print("Use: ");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
// Configuración del servidor mqtt
client.setServer(mqtt_server, port);
client.setCallback(callback);
reconnect(); // Se suscribe al topic establecido para la recepción de los
datos
// Configuración del motor de paso
stepper.setSpeed(StepSpeed);
// Set 80 divider for prescaler (see ESP32 Technical Reference Manual for
more
// info).
timer = timerBegin(0, 80, true);
// Attach onTimer function to our timer.
timerAttachInterrupt(timer, &onTimer, true);
// Set alarm to call onTimer function every 400uS (value in microseconds).
// Repeat the alarm (third parameter)
timerAlarmWrite(timer, 400, false);
// Start an alarm
timerAlarmEnable(timer);
// Interrupción para la detección del zeroCrossing
attachInterrupt(zeroCrossing, zcDetectISR, RISING); // Attach an Interrupt
to Pin 2 (interrupt 0) for Zero Cross Detection
temperature_sensor.setWaitForConversion(false);
temperature_sensor.requestTemperaturesByAddress(TempSensor);

```



```

    delay(100);
}

void loop() {
    // Obtiene tiempo de sistema
    CurrentTime = millis();
    // Controlador de movimiento del motor de paso
    MotorHandler();
    // Lee los sensores
    ReadSensor();
    // Controlador de reconexion *****
    if (client.connected()) {
        ReconnectTime = CurrentTime;
    } else {
        if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
            reconnect();
            ReconnectTime = CurrentTime;
        }
    }
    // Controlador de envio de datos *****
    if ((CurrentTime - ClientTime) > ClientRefresh) {
        ClientHandler();
        ClientTime = CurrentTime;
    }
    // Controlador de tiempo de activacion del sensor de vibracion
    *****
    if (vibration_detect) {
        if ((CurrentTime - VibrationHoldTime) > VibrationHoldRefresh) {
            vibration_detect = false;
        }
    }
    // Controlador de lectura del sensor de temperatura *****
    if ((CurrentTime - TemperatureTime) > 2000) {
        //Función para el Sensor de Temperatura.
        temperatureC = temperature_sensor.getTempC(TempSensor);
        temperature_sensor.requestTemperaturesByAddress(TempSensor);
        Serial.print("Temperatura: ");
        Serial.print(temperatureC);
        Serial.println(" °C");
        TemperatureTime = CurrentTime;
    }
    client.loop();
}

```

## Anexo 4: Código de programación para tarjeta de control del espacio baño

```
/*
*****
Cliente 4 - Baño
*****
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

/*
***** Pin Definition *****
//Relays for switching appliances
#define relay 5 //D1
#define ventanaSensor 4 //D2
#define aguaSensor 14 //D5
// Tag actuadores *****
#define Rele "BañoDevice/Rele" // Topic de suscripcion de la relay
// Tag sensores
#define Vibracion "BañoDevice/Vibracion" // Topic del sensor de vibracion
#define Nivel "BañoDevice/Nivel" // Topic del sensor de nivel
// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000
#define ClientRefresh 300
#define VibrationHoldRefresh 3000

// Configuración de red para el ESP *****
IPAddress local_IP(192, 168, 0, 222);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);

/*
***** WiFi Access Point
*****
const char* ssid = "TP-Link_27E0";
const char* password = "66608189";
const char* mqtt_server = "192.168.0.230"; //IP address of RPi
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;
volatile unsigned long ClientTime = 0;
volatile unsigned long VibrationHoldTime = 0;

// Variables de uso general *****
volatile boolean liquid_detect = false;
volatile boolean vibration_detect = false;
```

```

// Objeto para la comunicacion MQTT
WiFiClient espClient;
PubSubClient client(espClient);

//Función para el Sensor de Líquido.
ICACHE_RAM_ATTR void detectLiquid() {
  if (!digitalRead(aguaSensor)) {
    Serial.println("Líquido Detectado!");
    liquid_detect = true;
  } else liquid_detect = false;
}

//Función para el Sensor de Vibración.
ICACHE_RAM_ATTR void detectVibration() {
  if (!digitalRead(ventanaSensor)) {
    Serial.println("Vibracion Detectada!");
    vibration_detect = true;
    VibrationHoldTime = CurrentTime;
  }
}

// Funcion de reconexión
void reconnect() {
  // Si esta conectado a la red, no se realiza nada
  if (client.connected()) {
    return;
  }
  Serial.print("Intentando conexión MQTT...");
  String clientId = "ESP8266Baño";
  clientId += String(random(0xffff), HEX);
  //Intento de Conexión
  if (client.connect(clientId.c_str(), username, pass)) {
    Serial.println("Conectado");
    //Una vez conectado, publica un anuncio...
    // ... y vuelve a suscribir.
    client.subscribe(Rele);
    Serial.println(client.state());
  } else {
    Serial.print("Fallo, rc=");
    Serial.println(client.state());
    Serial.println("Intente nuevamente en 5 segundos");
  }
}

// Handler de envio de datos a la raspberry
void ClientHandler() {
  //Serial.println("Publicar mensaje ");

  //Serial.print("Nivel de agua: ");
  // if (liquid_detect) Serial.println("Alto");
}

```

```

// else Serial.println("Bajo");

if (liquid_detect) client.publish(Nivel, "true");
else client.publish(Nivel, "false");

//Serial.print("Nivel de vibracion: ");
//if (vibration_detect) Serial.println("Alta");
//else Serial.println("Baja");
if (vibration_detect) client.publish(Vibracion, "true");
else client.publish(Vibracion, "false");

}
// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Topic[");
  Serial.print(topic);
  Serial.println("]");
  Serial.print("payload[");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println("]");
  if (strstr(topic, Rele)) {
    if ((char)payload[0] == '1') digitalWrite(relay, LOW);
    else digitalWrite(relay, HIGH);
  } else {
    Serial.println("Tema no suscrito");
  }
}
void setup() {
  // Configuración inicial de entradas y salidas
  pinMode(relay, OUTPUT);
  pinMode(ventanaSensor, INPUT);
  pinMode(aguaSensor, INPUT);
  // Estado inicial de las salidas
  digitalWrite(relay, HIGH);
  // Configuración de la comunicación serial
  Serial.begin(115200);
  // Configuración de las interrupciones
  attachInterrupt(digitalPinToInterrupt(aguaSensor), detectLiquid, CHANGE);
  attachInterrupt(digitalPinToInterrupt(ventanaSensor), detectVibration,
CHANGE);

  // Configuración de red *****+
  WiFi.config(local_IP, gateway, subnet);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  // Conexión con la red

```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// IP conectada
Serial.println("");
Serial.println("WiFi connected");
Serial.print("Use: ");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
// Configuración del servidor mqtt
client.setServer(mqtt_server, port);
client.setCallback(callback);
reconnect(); // Se suscribe al topic establecido para la recepcion de los
datos
}

void loop() {
    CurrentTime = millis();
    // Controlador de reconexion *****
    if (client.connected()) {
        ReconnectTime = CurrentTime;
    } else {
        if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
            reconnect();
            ReconnectTime = CurrentTime;
        }
    }
    // Controlador de envio de datos *****
    if ((CurrentTime - ClientTime) > ClientRefresh) {
        ClientHandler();
        ClientTime = CurrentTime;
    }
    // Controlador de tiempo de activacion del sensor de vibracion
    *****
    if (vibration_detect) {
        if ((CurrentTime - VibrationHoldTime) > VibrationHoldRefresh) {
            vibration_detect = false;
        }
    }
    client.loop();
}

```

## Anexo 5: Código de programación para tarjeta de control del espacio físico dormitorio

1

```
#include <RBD_Timer.h>

/*****
  Cliente 2 - Dormitorio Master
  *****/
#include <WiFi.h>
#include <PubSubClient.h>
#include <Stepper.h>
#include <RBDdimmer.h>

/***** Pin Definition *****/

#define dimmer 23
#define relay 4
#define ventanaSensor 21
#define puertaSensor 16
#define IN1 19
#define IN2 5
#define IN3 18
#define IN4 17
#define zeroCrossing 22
#define STEPS 2048
#define StepSpeed 18
#define MAXMotorTurn 20
#define MAXMotorStep MAXMotorTurn* STEPS
// Tag actuadores *****/
#define Dimmer "Room1Device/Dimmer"
#define Relay "Room1Device/Relay"
#define Persiana "Room1Device/Persiana"
// Tag sensores
#define Ventana "Room1Device/Ventana" // Topic del sensor de ventana
#define Puerta "Room1Device/Puerta" // Topic del sensor de nivel
// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000
#define ClientRefresh 300
#define VentanaHoldRefresh 3000
// Configuración de red para el ESP *****/
IPAddress local_IP(192, 168, 2, 224);
IPAddress gateway(192, 168, 2, 1);
IPAddress subnet(255, 255, 255, 0);

/***** WiFi Access Point
  *****/
const char* ssid = "MANUEL";
const char* password = "1207263060";
```

```

const char* mqtt_server = "192.168.2.230"; //IP address of RPi
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;
volatile unsigned long ClientTime = 0;
volatile unsigned long VentanaHoldTime = 0;

// Variables de uso general *****
String val_dimm = "";
int CurrentStep = 0;
int MoveMotor = 0;
volatile boolean door_detect = false;
volatile boolean ventana_detect = false;
volatile boolean ventanaBuffer = false;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

// Variables del Dimmer
volatile byte fade = 0;
byte state = 1;
volatile byte tarBrightness = 0;
volatile byte curBrightness = 0;
volatile byte zcState = 0; // 0 = ready; 1 = processing;
unsigned long CurrentMicros = 0;
unsigned long MeasureMicros = 0;

// Objeto para la comunicacion MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Objeto del temporizador
hw_timer_t* timer = NULL;

// Objeto para el control del motor de paso
Stepper stepper(STEPS, IN1, IN2, IN3, IN4);

void ARDUINO_ISR_ATTR onTimer() {
    portENTER_CRITICAL(&timerMux);
    // Fade == 1: Significa que se llegara al valor por medio de un
    desvanecimiento
    if (fade == 1) {
        if (curBrightness > tarBrightness && curBrightness > 0) {
            --curBrightness;
        } else if (curBrightness < tarBrightness && curBrightness < 255) {
            ++curBrightness;
        }
    }
}

```

```

    }
    // Fade == 0: Significa que se llegara al valor de forma directa
} else {
    if (tarBrightness != 0) {
        curBrightness = tarBrightness;
    } else {
        curBrightness = 0;
    }
}
// Enciende o apaga el pin del dimmer para producir el control de
luminosidad
if (curBrightness == 0) {
    digitalWrite(dimmer, 0);
} else {
    digitalWrite(dimmer, 1);
}
// Reestablece la bandera de zerocrossing
zcState = 0;
portEXIT_CRITICAL(&timerMux);
}

ICACHE_RAM_ATTR void zcDetectISR() {
    //portENTER_CRITICAL(&timerMux);
    // Si se detecta un zero crossing entonces se inicia el proceso de control
    if (zcState == 0) {
        zcState = 1;
        if (curBrightness <= 255 && curBrightness >= 0) {
            digitalWrite(dimmer, 0);

            uint32_t dimDelay = 30 * (255 - curBrightness) + 400; //400
            timerWrite(timer, 0);
            timerAlarmWrite(timer, dimDelay, false);
            timerAlarmEnable(timer);
            //Serial.println("xD");
        }
    }
    //portEXIT_CRITICAL(&timerMux);
}

// Funcion de lectura de los sensores
void ReadSensor() {
    //Función para el Sensor de ventana.
    if (digitalRead(ventanaSensor) != ventanaBuffer) {
        ventanaBuffer = digitalRead(ventanaSensor);
        Serial.println("Ventana Detectada!");
        ventana_detect = true;
        VentanaHoldTime = CurrentTime;
    }
}

```



```

//Función para el Sensor de Puerta.
if (digitalRead(puertaSensor)) {
    Serial.println("Puerta abierta!");
    door_detect = true;
} else door_detect = false;
}

// Handler de envio de datos a la raspberry
void ClientHandler() {
    // Serial.println("Publicar mensaje ");
    // Serial.print("Estado de la puerta: ");
    // dtostrf(door_detect, 1, 2, str_door);
    if (door_detect) client.publish(Puerta, "true");
    else client.publish(Puerta, "false");

    // Serial.print("Nivel de vibracion: ");
    // dtostrf(vibration_detect, 1, 2, str_vibration);
    if (ventana_detect) client.publish(Ventana, "true");
    else client.publish(Ventana, "false");
}

// Funcion de reconexión
void reconnect() {
    // Si esta conectado a la red, no se realiza nada
    if (client.connected()) {
        return;
    }
    Serial.print("Intentando conexión MQTT...");
    String clientId = "ESP32DormitorioMaster";
    clientId += String(random(0xffff), HEX);
    //Intento de Conexión
    if (client.connect(clientId.c_str(), username, pass)) {
        Serial.println("Conectado");
        //Una vez conectado, publica un anuncio...
        // ... y vuelve a suscribir.
        client.subscribe(Dimmer);
        client.subscribe(Relay);
        client.subscribe(Persiana);
        Serial.println(client.state());
    } else {
        Serial.print("Fallo, rc=");
        Serial.println(client.state());
        Serial.println("Intente nuevamente en 5 segundos");
    }
}

// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Mensaje[");
    Serial.print(topic);

```

```

Serial.println("");

// Si el mensaje corresponde al control del rele, entonces se realiza la
accion correspondiente
if (strstr(topic, Relay)) {
    if ((char)payload[0] == '1') digitalWrite(relay, LOW);
    else digitalWrite(relay, HIGH);
}
// Si el mensaje corresponde al control del dimmer, entonces se realiza la
accion correspondiente
else if (strstr(topic, Dimmer)) {
    val_dimm = "";
    for (int i = 0; i < length; i++) {
        val_dimm += payload[i] - 48;
    }
    //int val = map(val_dimm.toInt(), 0, 255, 0, 100);
    int val = val_dimm.toInt();
    Serial.print("El valor del dimmer es: ");
    Serial.println(val);
    portENTER_CRITICAL(&timerMux);
    tarBrightness = val;
    portEXIT_CRITICAL(&timerMux);
    //DimmerControl.setPower(val);
}

// Si el mensaje corresponde al control de persianas, entonces se realiza la
accion correspondiente
else if (strstr(topic, Persiana)) {
    if ((char)payload[0] == '1') {
        // Si la persiana esta totalmente subida, no se le permite subir más
        if (CurrentStep > 0) {
            MoveMotor = 1;
        } else {
            MoveMotor = 0;
        }
    } else if ((char)payload[0] == '2') {
        // Si la persiana esta totalmente bajada, no se le permite bajas más
        if (CurrentStep < MAXMotorStep) {
            MoveMotor = 2;
        } else {
            MoveMotor = 0;
        }
    } else {
        MoveMotor = 0;
        return;
    }
    Serial.print("El Motor esta en la vuelta: ");
    Serial.println(round((CurrentStep * 100 / STEPS)) / 100);
}

```

```

}

else {
    Serial.println("Tema no suscrito");
}
}
void setup() {
    // Configuración inicial de entradas y salidas
    pinMode(dimmer, OUTPUT);
    pinMode(relay, OUTPUT);
    pinMode(ventanaSensor, INPUT);
    pinMode(puertaSensor, INPUT_PULLUP);
    pinMode(zeroCrossing, INPUT);
    // Estado inicial de las salidas
    digitalWrite(relay, HIGH);
    // Configuración de la comunicación serial
    Serial.begin(115200);
    // Configuración de red *****+
    WiFi.config(local_IP, gateway, subnet);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    // Conexión con la red
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    // IP conectada
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.print("Use: ");
    Serial.print(WiFi.localIP());
    Serial.println("' to connect");
    // Configuración del servidor mqtt
    client.setServer(mqtt_server, port);
    client.setCallback(callback);
    reconnect(); // Se suscribe al topic establecido para la recepción de los
datos
    // Configuración del motor de paso
    stepper.setSpeed(StepSpeed);
    // Set 80 divider for prescaler (see ESP32 Technical Reference Manual for
more
    // info).
    timer = timerBegin(0, 80, true);
    // Attach onTimer function to our timer.
    timerAttachInterrupt(timer, &onTimer, true);
    // Set alarm to call onTimer function every 400uS (value in microseconds).
    // Repeat the alarm (third parameter)
    timerAlarmWrite(timer, 400, false);
}

```

```

// Start an alarm
timerAlarmEnable(timer);
// Interrupcion para la deteccion del zeroCrossing
attachInterrupt(zeroCrossing, zcDetectISR, RISING); // Attach an Interupt
to Pin 2 (interupt 0) for Zero Cross Detection
}
// Controla el movimiento del motor *****
void MotorHandler() {
    if (MoveMotor == 0) {
        // Apaga las bobinas del motor para evitar el sobrecalentamiento. Si es
necesario, se puede comentar este bloque if si se necesita mantener el torque
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
    }
    if (MoveMotor == 1) {
        stepper.step(1);
        CurrentStep--;
    }
    if (MoveMotor == 2) {
        stepper.step(-1);
        CurrentStep++;
    }
}
void loop() {
    CurrentTime = millis();
    // Controlador de movimiento del motor de paso
    MotorHandler();
    // Lee los sensores
    ReadSensor();
    // Controlador de reconexion *****
    if (client.connected()) {
        ReconnectTime = CurrentTime;
    } else {
        if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
            reconnect();
            ReconnectTime = CurrentTime;
        }
    }
    // Controlador de envio de datos *****
    if ((CurrentTime - ClientTime) > ClientRefresh) {
        ClientHandler();
        ClientTime = CurrentTime;
    }

    // Controlador de tiempo de activacion del sensor de vibracion
*****

```

```
if (ventana_detect) {  
    if ((CurrentTime - VentanaHoldTime) > VentanaHoldRefresh) {  
        ventana_detect = false;  
    }  
}  
client.loop();  
}
```

## Anexo 6: Código de programación para tarjeta de control del espacio físico dormitorio 2

```
/*
  Cliente 6 - Dormitorio 2
  */
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "hw_timer.h"
#include <Stepper.h>

/* ***** Pin Definition ***** */

#define dimmer 5 //D1
#define relay 16 //D0
#define IN1 14 //D5
#define IN2 13 //D7
#define IN3 12 //D6
#define IN4 15 //D8
#define zeroCrossing 4 //D2
#define STEPS 2048
#define StepSpeed 15
#define MAXMotorTurn 20
#define MAXMotorStep MAXMotorTurn* STEPS

// Tag actuadores *****
#define Dimmer "Room2Device/Dimmer"
#define Relay "Room2Device/Relay"
#define Persiana "Room2Device/Persiana"
// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000
#define ClientRefresh 300

// Configuración de red para el ESP *****
IPAddress local_IP(192, 168, 2, 225);
IPAddress gateway(192, 168, 2, 1);
IPAddress subnet(255, 255, 255, 0);

/* ***** WiFi Access Point ***** */
const char* ssid = "JESSENIA";
const char* password = "rosita2000@";
const char* mqtt_server = "192.168.2.230"; //IP address of RPi
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
```

```

volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;
volatile unsigned long ClientTime = 0;
unsigned long CurrentMicros = 0;
unsigned long MeasureMicros = 0;
// Variables de uso general *****
String val_dimm = "";
int CurrentStep = 0;
int MoveMotor = 0;

// Variables del Dimmer
volatile byte fade = 0;
byte state = 1;
volatile byte tarBrightness = 255;
volatile byte curBrightness = 0;
volatile byte zcState = 0; // 0 = ready; 1 = processing;

// Objeto para la comunicacion MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Objeto para el control del motor de paso
Stepper stepper(STEPS, IN1, IN2, IN3, IN4);

// Interrupcion de control del brillo del Dimmer
ICACHE_RAM_ATTR void dimTimerISR() {
  //CurrentMicros = micros();
  //Serial.println(CurrentMicros - MeasureMicros);
  //MeasureMicros = CurrentMicros;

  // Fade == 1: Significa que se llegara al valor por medio de un
  desvanecimiento
  if (fade == 1) {
    if (curBrightness > tarBrightness && curBrightness > 0) {
      --curBrightness;
    } else if (curBrightness < tarBrightness && curBrightness < 255) {
      ++curBrightness;
    }
    // Fade == 0: Significa que se llegara al valor de forma directa
  } else {
    if (tarBrightness != 0) {
      curBrightness = tarBrightness;
    } else {
      curBrightness = 0;
    }
  }
}

```

```

// Enciende o apaga el pin del dimmer para producir el control de
luminosidad
if (curBrightness == 0) {
    digitalWrite(dimmer, 0);
} else {
    digitalWrite(dimmer, 1);
}

// Reestablece la bandera de zerocrossing
zcState = 0;
}

// Interrupcion de deteccion de zero crossing
ICACHE_RAM_ATTR void zcDetectISR() {
    // Si se detecta un zero crossing entonecs se inicia el proceso de control
    if (zcState == 0) {
        zcState = 1;
        if (curBrightness < 255 && curBrightness > 0) {
            digitalWrite(dimmer, 0);

            int dimDelay = 30 * (255 - curBrightness) + 400; //400
            Serial.println(dimDelay);
            hw_timer_arm(dimDelay);
        }
    }
}

// Handler de envio de datos a la raspberry
void ClientHandler() {
    // No hay sensores en este dispositivo
}

// Funcion de reconexión
void reconnect() {
    // Si esta conectado a la red, no se realiza nada
    if (client.connected()) {
        return;
    }
    Serial.print("Intentando conexión MQTT...");
    String clientId = "ESP8266DormitorioSecundario";
    clientId += String(random(0xffff), HEX);
    //Intento de Conexión
    if (client.connect(clientId.c_str(), username, pass)) {
        Serial.println("Conectado");
        //Una vez conectado, publica un anuncio...
        // ... y vuelve a suscribir.
        client.subscribe(Dimmer);
        client.subscribe(Relay);
        client.subscribe(Persiana);
        Serial.println(client.state());
    }
}

```



```

    } else {
        Serial.print("Fallo, rc=");
        Serial.println(client.state());
        Serial.println("Intente nuevamente en 5 segundos");
    }
}

// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
    // Lee el mensaje que ha recibido
    Serial.print("Mensaje[");
    Serial.print(topic);
    Serial.println("]");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();

    // Si el mensaje corresponde al control del rele, entonces se realiza la
    accion correspondiente
    if (strstr(topic, Relay)) {
        if ((char)payload[0] == '1') digitalWrite(relay, LOW);
        else digitalWrite(relay, HIGH);
    }

    // Si el mensaje corresponde al control del dimmer, entonces se realiza el
    control del dimmer
    else if (strstr(topic, Dimmer)) {
        val_dimm = "";
        // Obtiene el dato del dimmer en str
        for (int i = 0; i < length; i++) {
            val_dimm += payload[i] - 48;
        }
        // Convierte de str a int para poder trabajar con el valor
        int val = val_dimm.toInt();
        tarBrightness = val;
        // Serial.print("El valor del dimmer es: ");
        // Serial.println(tarBrightness);
    }

    // Si el mensaje corresponde al control de persianas, entonces se realiza la
    accion correspondiente
    else if (strstr(topic, Persiana)) {
        if ((char)payload[0] == '1') {
            // Si la persiana esta totalmente subida, no se le permite subir más
            if (CurrentStep > 0) {
                MoveMotor = 1;
            } else {

```

```

        MoveMotor = 0;
    }
} else if ((char)payload[0] == '2') {
    // Si la persiana esta totalmente bajada, no se le permite bajar más
    if (CurrentStep < MAXMotorStep) {
        MoveMotor = 2;
    } else {
        MoveMotor = 0;
    }
} else {
    MoveMotor = 0;
    return;
}
Serial.print("El Motor esta en la vuelta: ");
Serial.println(round((CurrentStep * 100 / STEPS)) / 100);
}
else {
    Serial.println("Tema no suscrito");
}
}
// Controla el movimiento del motor *****
void MotorHandler() {
    if (MoveMotor == 0) {
        // Apaga las bobinas del motor para evitar el sobrecalentamiento. Si es
        // necesario, se puede comentar este bloque if si se necesita mantener el torque
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
    }
    if (MoveMotor == 1) {
        stepper.step(1);
        CurrentStep--;
    }
    if (MoveMotor == 2) {
        stepper.step(-1);
        CurrentStep++;
    }
}
}
void setup() {
    // Configuración inicial de entradas y salidas
    pinMode(dimmer, OUTPUT);
    pinMode(relay, OUTPUT);
    pinMode(zeroCrossing, INPUT);
    // Estado inicial de las salidas
    digitalWrite(relay, HIGH);
    // Configuración de la comunicación serial
    Serial.begin(115200);
}

```

```

// Configuración de red *****+
WiFi.config(local_IP, gateway, subnet);
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
// Conexión con la red
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
// IP conectada
Serial.println("");
Serial.println("WiFi connected");
Serial.print("Use: ");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
// Configuración del servidor mqtt
client.setServer(mqtt_server, port);
client.setCallback(callback);
reconnect(); // Se suscribe al topic establecido para la recepción de los
datos
// Configuración del motor de paso
stepper.setSpeed(StepSpeed);
// Interrupción Zero crossing
attachInterrupt(zeroCrossing, zcDetectISR, RISING); // Attach an Interrupt
for Zero Cross Detection
hw_timer_init(NMI_SOURCE, 0);
hw_timer_set_func(dimTimerISR);
}
void loop() {
    CurrentTime = millis();
    // Controlador de movimiento del motor de paso
    MotorHandler();
    // Controlador de reconexión *****
    if (client.connected()) {
        ReconnectTime = CurrentTime;
    } else {
        if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
            reconnect();
            ReconnectTime = CurrentTime;
        }
    }
    // Controlador de envío de datos *****
    if ((CurrentTime - ClientTime) > ClientRefresh) {
        ClientHandler();
        ClientTime = CurrentTime;
    }
    client.loop();
}

```

## Anexo 7: Código de programación para tarjeta de control del espacio físico comedor

```
/*
*****
Cliente 3 - Cocina
*****
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "hw_timer.h"

/*
***** Pin Definition *****
*/

#define dimmer 12 //D6
#define relay 4 //D2
#define ventanaSensor 5 //D1
#define puertaSensor 13 //D7
#define zeroCrossing 14 //D5

// Tag actuadores *****
#define Dimmer "ComedorDevice/Dimmer"
#define Relay "ComedorDevice/Relay"
// Tag sensores
#define Ventana "ComedorDevice/Ventana" // Topic del sensor de ventana
#define Puerta "ComedorDevice/Puerta" // Topic del sensor de puerta

// Buffer para la comunicacion de 50 bytes
#define MSG_BUFFER_SIZE (50)
// Tiempos de ejecucion del programa
#define ReconnectionRefresh 3000
#define ClientRefresh 300
#define VentanaHoldRefresh 3000

// Configuración de red para el ESP *****
IPAddress local_IP(192, 168, 2, 227);
IPAddress gateway(192, 168, 2, 1);
IPAddress subnet(255, 255, 255, 0);

/*
***** WiFi Access Point
*****
const char* ssid = "Angel";
const char* password = "1710721489";
const char* mqtt_server = "192.168.2.230"; //IP address of RPi
const int port = 1883;
const char* username = "";
const char* pass = "";

// Variables de control de ejecución *****
volatile unsigned long CurrentTime = 0;
volatile unsigned long ReconnectTime = 0;
volatile unsigned long ClientTime = 0;
volatile unsigned long VentanaHoldTime = 0;
```

```

// Variables de uso general *****
String val_dimm = "";
volatile boolean door_detect = false;
volatile boolean ventana_detect = false;
volatile boolean ventanaBuffer = false;

// Variables del Dimmer
byte fade = 0;
byte state = 1;
byte tarBrightness = 255;
byte curBrightness = 0;
byte zcState = 0; // 0 = ready; 1 = processing;

// Objeto para la comunicacion MQTT
WiFiClient espClient;
PubSubClient client(espClient);

// Lee los sensores
void ReadSensor() {
    //Función para el Sensor de puerta
    if (!digitalRead(puertaSensor)) {
        Serial.println("Puerta detectada!");
        door_detect = true;
    } else door_detect = false;

    //Funcion para el sensor de ventana
    if (digitalRead(ventanaSensor) != ventanaBuffer) {
        ventanaBuffer = digitalRead(ventanaSensor);
        Serial.println("Ventana Detectada!");
        ventana_detect = true;
        VentanaHoldTime = CurrentTime;
    }
}

// Funcion de reconexión
void reconnect() {
    // Si esta conectado a la red, no se realiza nada
    if (client.connected()) {
        return;
    }
    Serial.print("Intentando conexión MQTT...");
    String clientId = "ESP8266Comedor";
    clientId += String(random(0xffff), HEX);
    //Intento de Conexión
    if (client.connect(clientId.c_str(), username, pass)) {
        Serial.println("Conectado");
        //Una vez conectado, publica un anuncio...

```

```

    // ... y vuelve a suscribir.
    client.subscribe(Dimmer);
    client.subscribe(Relay);
    Serial.println(client.state());
} else {
    Serial.print("Fallo, rc=");
    Serial.println(client.state());
    Serial.println("Intente nuevamente en 5 segundos");
}
}
}
// Handler de envio de datos a la raspberry
void ClientHandler() {
    if (door_detect) client.publish(Puerta, "true");
    else client.publish(Puerta, "false");

    if (ventana_detect) client.publish(Ventana, "true");
    else client.publish(Ventana, "false");
}
// Handler de recepción de datos de la raspberry
void callback(char* topic, byte* payload, unsigned int length) {
    // Lee el mensaje que ha recibido
    Serial.print("Mensaje[");
    Serial.print(topic);
    Serial.println("]");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();

    // Si el mensaje corresponde al control del rele, entonces se realiza la
    accion correspondiente
    if (strstr(topic, Relay)) {
        if ((char)payload[0] == '1') digitalWrite(relay, LOW);
        else digitalWrite(relay, HIGH);
    }

    // Si el mensaje corresponde al control del dimmer, entonces se realiza el
    control del dimmer
    else if (strstr(topic, Dimmer)) {
        val_dimm = "";
        // Obtiene el dato del dimmer en str
        for (int i = 0; i < length; i++) {
            val_dimm += payload[i] - 48;
        }
        // Convierte de str a int para poder trabajar con el valor
        int val = val_dimm.toInt();
        tarBrightness = val;
        // Serial.print("El valor del dimmer es: ");

```

```

    // Serial.println(tarBrightness);
}

else {
    Serial.println("Tema no suscrito");
}
}

ICACHE_RAM_ATTR void dimTimerISR() {
    // Fade == 1: Significa que se llegara al valor por medio de un
    desvanecimiento
    if (fade == 1) {
        if (curBrightness > tarBrightness && curBrightness > 0) {
            --curBrightness;
        } else if (curBrightness < tarBrightness && curBrightness < 255) {
            ++curBrightness;
        }
    }
    // Fade == 0: Significa que se llegara al valor de forma directa
} else {
    if (tarBrightness != 0) {
        curBrightness = tarBrightness;
    } else {
        curBrightness = 0;
    }
}

// Enciende o apaga el pin del dimmer para producir el control de
luminosidad
if (curBrightness == 0) {
    digitalWrite(dimmer, 0);
} else {
    digitalWrite(dimmer, 1);
}

// Reestablece la bandera de zerocrossing
zcState = 0;
}

ICACHE_RAM_ATTR void zcDetectISR() {
    // Si se detecta un zero crossing entonecs se inicia el proceso de control
    if (zcState == 0) {
        zcState = 1;
        if (curBrightness < 255 && curBrightness > 0) {
            digitalWrite(dimmer, 0);

            int dimDelay = 30 * (255 - curBrightness) + 400; //400
            hw_timer_arm(dimDelay);
        }
    }
}

```

```

    }
}
void setup() {
  // Configuración inicial de entradas y salidas
  pinMode(relay, OUTPUT);
  pinMode(ventanaSensor, INPUT);
  pinMode(puertaSensor, INPUT_PULLUP);
  pinMode(dimmer, OUTPUT);
  // Estado inicial de las salidas
  digitalWrite(relay, HIGH);
  // Configuración de la comunicación serial
  Serial.begin(115200);
  // Configuración de red *****+
  WiFi.config(local_IP, gateway, subnet);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  // Conexión con la red
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // IP conectada
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("Use: ");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
  // Configuración del servidor mqtt
  client.setServer(mqtt_server, port);
  client.setCallback(callback);
  reconnect(); // Se suscribe al topic establecido para la recepción de los
datos

  attachInterrupt(zeroCrossing, zcDetectISR, RISING); // Attach an Interrupt
to Pin 2 (interrupt 0) for Zero Cross Detection
  hw_timer_init(NMI_SOURCE, 0);
  hw_timer_set_func(dimTimerISR);
}

void loop() {
  CurrentTime = millis();
  // Lee los sensores
  ReadSensor();

  // Controlador de reconexión *****
  if (client.connected()) {
    ReconnectTime = CurrentTime;
  } else {

```



```

    if ((CurrentTime - ReconnectTime) > ReconnectionRefresh) {
        reconnect();
        ReconnectTime = CurrentTime;
    }
}
// Controlador de envio de datos *****
if ((CurrentTime - ClientTime) > ClientRefresh) {
    ClientHandler();
    ClientTime = CurrentTime;
}

// Controlador de tiempo de activacion del sensor de vibracion
*****
if (ventana_detect) {
    if ((CurrentTime - VentanaHoldTime) > VentanaHoldRefresh) {
        ventana_detect = false;
    }
}
}

```

## Anexo 8: Código de programación de la aplicación móvil desarrollada en App Inventor

Visor

```
inicializar global SelectorDormitorio como falso
inicializar global SelectorBuffer como 0
inicializar global Selector como 0
inicializar global LastScreen como 0
inicializar global DataReceived como falso
inicializar global DataReceived2 como falso
inicializar global StartScreen como falso
inicializar global StartScreen2 como falso

cuando Sala . Inicializar
ejecutar
  poner ButtonDormitorio . Habilitado como cierto
  poner ButtonCocina . Habilitado como cierto
  poner ButtonBaño . Habilitado como cierto
  poner ButtonSala . Habilitado como cierto
  poner GetDataBaseData . TemporizadorHabilitado como falso
  Llamar CargaControlesSala
  poner GetDataBaseData . TemporizadorHabilitado como cierto
```

```
inicializar global IP1 como 192.168.0.220
inicializar global IP2 como 192.168.0.221
```

```
cuando Sala . Error
componente nombreDeFunción númeroDeError mensaje
ejecutar llamar Notificador . MostrarDiálogoTexto
  mensaje unir Error en el componente
  tomar componente
  función
  llamar nombreDeFunción
  error
  tomar númeroDeError
  Mensaj
  tomar mensaje
  título <Error>
  cancelable cierto
```

```
como CambiarScreen
ejecutar
  si
    tomar global SelectorBuffer = 0
  entonces
    Llamar ApagarCamaras
    poner Título . Texto como Sala
    poner ButtonDormitorio . Habilitado como cierto
    poner ButtonComedor . Habilitado como cierto
    poner ButtonCocina . Habilitado como cierto
    poner ButtonBaño . Habilitado como cierto
    poner ButtonSala . Habilitado como cierto
    poner ButtonCamara . Habilitado como cierto
    poner ButtonSala . Imagen como BotonSalaBloqueado.png
    poner ButtonComedor . Imagen como BotonComedor.png
    poner ButtonBaño . Imagen como BotonBaño.png
    poner ButtonCocina . Imagen como BotonCocina.png
    poner ButtonDormitorio . Imagen como BotonDormitorio.png
    poner ButtonCamara . Imagen como BotonCamara.png
    poner ScrollSala . Visible como cierto
    poner ScrollComedor . Visible como falso
    poner ScrollCocina . Visible como falso
    poner ScrollBaño . Visible como falso
    poner ScrollDormitorio1 . Visible como falso
    poner ScrollDormitorio2 . Visible como falso
    poner ScrollCamara . Visible como falso
  si
    tomar global SelectorBuffer = 2
  entonces
    Llamar ApagarCamaras
    poner Título . Texto como Sanitario
    poner ButtonDormitorio . Habilitado como cierto
    poner ButtonComedor . Habilitado como cierto
    poner ButtonCocina . Habilitado como cierto
    poner ButtonBaño . Habilitado como falso
    poner ButtonSala . Habilitado como cierto
    poner ButtonCamara . Habilitado como cierto
    poner ButtonSala . Imagen como BotonSala.png
    poner ButtonBaño . Imagen como BotonBañoBloqueado.png
    poner ButtonComedor . Imagen como BotonComedor.png
    poner ButtonCocina . Imagen como BotonCocina.png
    poner ButtonDormitorio . Imagen como BotonDormitorio.png
    poner ButtonCamara . Imagen como BotonCamara.png
    poner ScrollSala . Visible como falso
    poner ScrollComedor . Visible como falso
    poner ScrollCocina . Visible como falso
    poner ScrollBaño . Visible como cierto
    poner SelectorDormitorio . Visible como falso
    poner ScrollDormitorio1 . Visible como falso
    poner ScrollDormitorio2 . Visible como falso
    poner ScrollCamara . Visible como falso
```

```
si
  tomar global SelectorBuffer = 1
entonces
  Llamar ApagarCamaras
  poner Título . Texto como Cocina
  poner ButtonDormitorio . Habilitado como cierto
  poner ButtonComedor . Habilitado como cierto
  poner ButtonCocina . Habilitado como falso
  poner ButtonBaño . Habilitado como cierto
  poner ButtonSala . Habilitado como cierto
  poner ButtonCamara . Habilitado como cierto
  poner ButtonSala . Imagen como BotonSala.png
  poner ButtonBaño . Imagen como BotonBaño.png
  poner ButtonComedor . Imagen como BotonComedor.png
  poner ButtonCocina . Imagen como BotonCocinaBloqueado.png
  poner ButtonDormitorio . Imagen como BotonDormitorio.png
  poner ButtonCamara . Imagen como BotonCamara.png
  poner ScrollSala . Visible como falso
  poner ScrollComedor . Visible como falso
  poner ScrollCocina . Visible como cierto
  poner ScrollBaño . Visible como falso
  poner SelectorDormitorio . Visible como falso
  poner ScrollDormitorio1 . Visible como falso
  poner ScrollDormitorio2 . Visible como falso
  poner ScrollCamara . Visible como falso

si
  tomar global SelectorBuffer = 3
entonces
  Llamar ApagarCamaras
  poner Título . Texto como Cuarto
  poner ButtonDormitorio . Habilitado como falso
  poner ButtonComedor . Habilitado como cierto
  poner ButtonCocina . Habilitado como cierto
  poner ButtonBaño . Habilitado como cierto
  poner ButtonSala . Habilitado como cierto
  poner ButtonCamara . Habilitado como cierto
  poner ButtonSala . Imagen como BotonSala.png
  poner ButtonBaño . Imagen como BotonBaño.png
  poner ButtonComedor . Imagen como BotonComedor.png
  poner ButtonCocina . Imagen como BotonCocina.png
  poner ButtonDormitorio . Imagen como BotonDormitorioBloqueado.png
  poner ButtonCamara . Imagen como BotonCamara.png
  poner ScrollSala . Visible como falso
  poner ScrollComedor . Visible como falso
  poner ScrollCocina . Visible como falso
  poner ScrollBaño . Visible como falso
  poner SelectorDormitorio . Visible como cierto
  poner ScrollCamara . Visible como falso
```

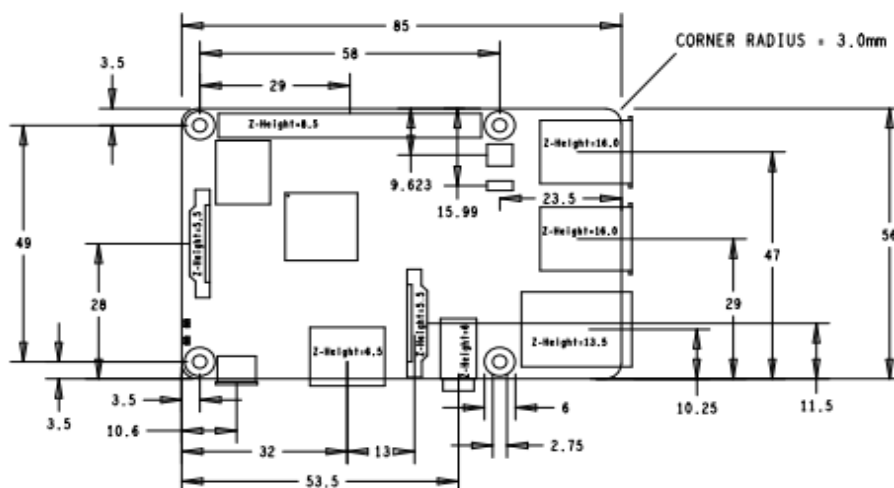
## Anexo 9: Datasheet Raspberry Pi 3 modelo B+

### Specifications

<b>Processor:</b>	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
<b>Memory:</b>	1GB LPDDR2 SDRAM
<b>Connectivity:</b>	<ul style="list-style-type: none"><li>■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE</li><li>■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)</li><li>■ 4 × USB 2.0 ports</li></ul>
<b>Access:</b>	Extended 40-pin GPIO header
<b>Video &amp; sound:</b>	<ul style="list-style-type: none"><li>■ 1 × full size HDMI</li><li>■ MIPI DSI display port</li><li>■ MIPI CSI camera port</li><li>■ 4 pole stereo output and composite video port</li></ul>
<b>Multimedia:</b>	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
<b>SD card support:</b>	Micro SD format for loading operating system and data storage
<b>Input power:</b>	<ul style="list-style-type: none"><li>■ 5V/2.5A DC via micro USB connector</li><li>■ 5V DC via GPIO header</li><li>■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)</li></ul>
<b>Environment:</b>	Operating temperature, 0–50 °C
<b>Compliance:</b>	For a full list of local and regional product approvals, please visit <a href="http://www.raspberrypi.org/products/raspberrypi-3-model-b+">www.raspberrypi.org/products/raspberrypi-3-model-b+</a>
<b>Production lifetime:</b>	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

Active  
Ve a Co

### Physical specifications



## Anexo 10: Datasheet NodeMCU ESP8266

### NodeMCU ESP8266



Front View



Front View

### Specifications of ESP-12E WiFi Module

<b>Wireless Standard</b>	IEEE 802.11 b/g/n
<b>Frequency Range</b>	2.412 - 2.484 GHz
<b>Power Transmission</b>	802.11b : +16 ± 2 dBm (at 11 Mbps) 802.11g : +14 ± 2 dBm (at 54 Mbps) 802.11n : +13 ± 2 dBm (at HT20, MCS7)
<b>Receiving Sensitivity</b>	802.11b : -93 dBm (at 11 Mbps, CCK) 802.11g : -85 dBm (at 54 Mbps, OFDM) 802.11n : -82 dBm (at HT20, MCS7)
<b>Wireless Form</b>	On-board PCB Antenna
<b>IO Capability</b>	UART, I2C, PWM, GPIO, 1 ADC
<b>Electrical Characteristic</b>	3.3 V Operated 15 mA output current per GPIO pin 12 - 200 mA working current Less than 200 uA standby current
<b>Operating Temperature</b>	-40 to +125 °C
<b>Serial Transmission</b>	110 - 921600 bps, TCP Client 5
<b>Wireless Network Type</b>	STA / AP / STA + AP
<b>Security Type</b>	WEP / WPA-PSK / WPA2-PSK
<b>Encryption Type</b>	WEP64 / WEP128 / TKIP / AES
<b>Firmware Upgrade</b>	Local Serial Port, OTA Remote Upgrade
<b>Network Protocol</b>	IPv4, TCP / UDP / FTP / HTTP
<b>User Configuration</b>	AT + Order Set, Web Android / iOS, Smart Link APP

# NodeMCU ESP32

Microcontroller Development Board



## Technical Specifications

Model	NodeMCU ESP32
Artide No.	SBC-NodeMCU-ESP32
Type	ESP32
Processor	Tensilica LX6 Dual-Core
Clock Frequency	240 MHz
SRAM	512 kB
Memory	4 MB
Wireless Standard	802.11 b/g/n
Frequency	2.4 GHz
Bluetooth	Classic / LE
Data Interfaces	UART / I2C / SPI / DAC / ADC
Operating Voltage	3,3V (operable via 5V-microUSB)
Operating Temperature	-40°C - 125°C
Dimensions (W x D x H)	48 x 26 x 11.5 mm
Scope Of Delivery	NodeMCU ESP32
EAN	4250236816104

## Anexo 12: Datasheet cámara ESP32 CAM

### FEATURES

- Up to 160MHz clock speed, Summary computing power up to 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, Built-in Flash lamp.
- Support image WiFi upload
- Support TF card
- Supports multiple sleep modes.
- Embedded Lwip and FreeRTOS
- Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

### SPECIFICATION

- SPI Flash: default 32Mbit
- RAM: built-in 520 KB+external 4MPSRAM
- Dimension: 27\*40.5\*4.5 (±0.2) mm/1.06\*1.59\*0.18"
- Bluetooth: Bluetooth 4.2 BR/EDR and BLE standards
- Wi-Fi: 802.11b/g/n/e/i
- Support Interface: UART, SPI, I2C, PWM
- Image Output Format: JPEG( OV2640 support only ), BMP, GRAYSCALE
- Spectrum Range: 2412 ~2484MHz
- Antenna: onboard PCB antenna, gain 2dBi
- Transmit Power: 802.11b: 17±2 dBm (@11Mbps);  
802.11g: 14±2 dBm (@54Mbps);  
802.11n: 13±2 dBm (@MCS7)
- Receiving Sensitivity: CCK, 1 Mbps : -90dBm;  
CCK, 11 Mbps: -85dBm;  
6 Mbps (1/2 BPSK): -88dBm;  
54 Mbps (3/4 64-QAM): -70dBm;  
MCS7 (65 Mbps, 72.2 Mbps): -67dBm
- Power consumption: Turn off the flash: 180mA@5V  
Turn on the flash and adjust the brightness to the maximum:  
310mA@5V  
Deep-sleep: the lowest power consumption can reach 6mA@5V  
Modern-sleep: up to 20mA@5V  
Light-sleep: up to 6.7mA@5V
- Security: WPA/WPA2/WPA2-Enterprise/WPS
- Power supply range: 5V
- Operating temperature: -20 °C ~ 85 °C
- Storage environment: -40 °C ~ 90 °C, < 90%RH
- Weight: 10g

### SHIPPING LIST

- ESP32-CAM Development Board x1

Activar  
Ve a Con

Activar  
Ve a Con

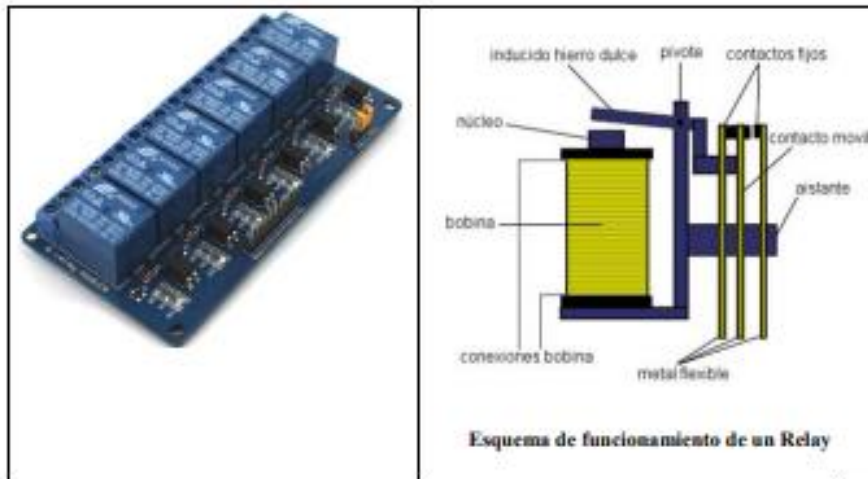
## Anexo 13: Datasheet módulo relé JQC-3FF-S-Z

### MÓDULO DE RELÉS

(Versión 15-6-19)

El módulo que vamos a emplear es uno muy típico y a un precio aceptable:

Como podemos ver, en este módulo (que es el que he empleado para el tutorial) trae dos entradas para poder activar dos relés, pero podemos encontrar módulos con más relés e incluso de un solo relé, lo que necesitemos para nuestros proyectos.



#### Descripción del Producto

Plataforma: Arduino 8051 AVR PIC DSP ARM MSP430 TTL. (Son distintas plataformas)  
Control Dispositivos varios/cargas  
Voltaje de operación 250VAC/30VDC  
Voltaje de la bobina (relé) 5V  
Corriente de operación 10A. Algunos fabricantes aclaran:

**Corriente máx: 10A (NO), 5A (NC)**  
**Tiempo de acción: 10 ms / 5 ms**

**Corriente de activación por relé 15mA-20mA** **[ATENCIÓN: VEA ANEXO1]**  
Canales 1-2 - 4 - 6 -8 (independientes protegidos con Optoacopladores)  
LED indicador Para cada canal (cuando bobina está activa)

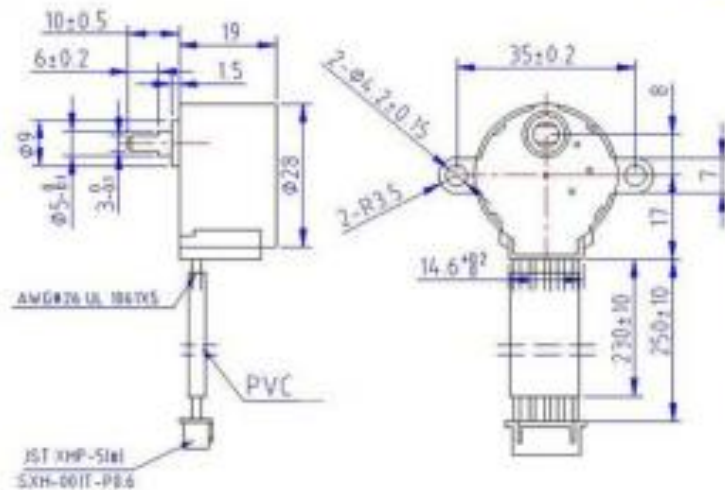
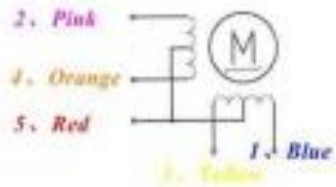
## Anexo 14: Datasheet motor 28BYJ-48

### 28BYJ-48 – 5V Stepper Motor

The 28BYJ-48 is a small stepper motor suitable for a large range of applications.



Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 800Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	≤40K(120Hz)
Noise	≤35dB(120Hz, No load, 10cm)
Model	28BYJ-48 – 5V





## Anexo 15: Datasheet sensor de gas MQ-6

### MQ-6 Semiconductor Sensor for LPG

Sensitive material of MQ-6 gas sensor is  $\text{SnO}_2$  which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-6 gas sensor has high sensitivity to Propane, Butane and LPG, also response to Natural gas. The sensor could be used to detect different combustible gas, especially Methane, it is with low cost and suitable for different application.

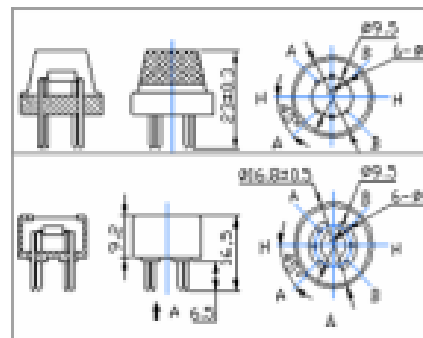
#### Character

- \* Good sensitivity to Combustible gas in wide range
- \* High sensitivity to Propane, Butane and LPG
- \* Long life and low cost
- \* Simple drive circuit

#### Application

- \* Domestic gas leakage detector
- \* Industrial Combustible gas detector
- \* Portable gas detector

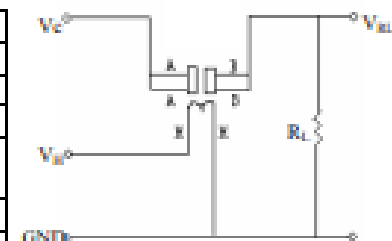
#### Configuration



#### Technical Data

Model No.		MQ-6	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Isobutane, Butane, LPG	
Concentration		300-10000ppm ( Butane, Propane, LPG )	
Circuit	Loop Voltage	$V_L$	≤24V DC
	Heater Voltage	$V_H$	5.0V±0.2V ACorDC
	Load Resistance	$R_L$	Adjustable
Character	Heater Resistance	$R_H$	310±30 (Room Tem.)
	Heater consumption	$P_H$	≤900mW
	Sensing Resistance	$R_s$	2KΩ-20KΩ(in 2000ppm C.I.)
	Sensitivity	$S$	$R_s(\text{in air})/R_s(1000\text{ppm C.I.})\geq 5$
	Slope	$\alpha$	±0.6( $R_{1000}/R_{10000}$ LPG )
Condition	Tem. Humidity	20°C±2°C, 65%±5%RH	
	Standard test circuit	$V_C$ : 5.0V±0.1V, $V_H$ : 5.0V±0.1V	
	Preheat time	Over 48 hours	

#### Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage(  $V_H$  ) and test voltage(  $V_C$  ).  $V_H$  used to supply certified working temperature to the sensor, while  $V_C$  used to detect voltage(  $V_{RL}$  ) on load resistance (  $R_L$  ) whom is in series with sensor. The sensor has light polarity,  $V_C$  need DC power.  $V_C$  and  $V_H$  could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable  $R_L$  value is needed: Power of Sensitivity body( $P_s$ ):

# Anexo 16: Datasheet sensor de humo MQ-135

## TECHNICAL DATA MQ-135 GAS SENSOR

### FEATURES

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Simple drive circuit

### APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH<sub>3</sub>, NO<sub>x</sub>, alcohol, Benzene, smoke, CO<sub>2</sub>, etc.

### SPECIFICATIONS

#### A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V <sub>c</sub>	Circuit voltage	5V±0.1	AC OR DC
V <sub>H</sub>	Heating voltage	5V±0.1	AC OR DC
R <sub>L</sub>	Load resistance	can adjust	
R <sub>H</sub>	Heater resistance	33Ω±5%	Room Tem
P <sub>H</sub>	Heating consumption	less than 800mw	

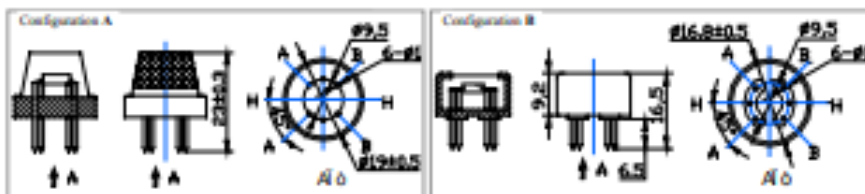
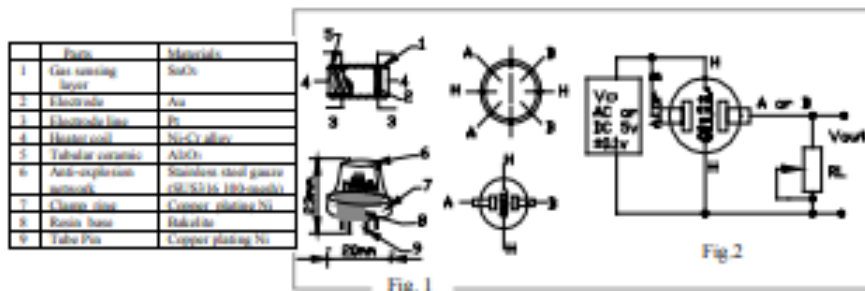
#### B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T <sub>ao</sub>	Using Tem	-10 ~45	
T <sub>as</sub>	Storage Tem	-20 ~70	
R <sub>h</sub>	Relative humidity	less than 95%RH	
O <sub>2</sub>	Oxygen concentration	21%(standard condition)(Oxygen concentration can affect sensitivity)	minimum value is over 2%

#### C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R <sub>s</sub>	Sensing Resistance	30KΩ-200KΩ (100ppm NH <sub>3</sub> )	Detecting concentration scope 10ppm-300ppm NH <sub>3</sub> 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200:50) NH <sub>3</sub>	Concentration Slope rate	±0.65	
Standard Detecting Condition	Temp: 20 ±2 V <sub>c</sub> : 5V±0.1 Humidity: 65%±5% V <sub>H</sub> : 5V±0.1		
Preheat time	Over 24 hour		

#### D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL<sub>2</sub>O<sub>3</sub> ceramic tube, Tin Dioxide (SnO<sub>2</sub>) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive

# Anexo 17: Datasheet sensor de temperatura DS18B20

DS18B20

Programmable Resolution  
1-Wire Digital Thermometer

## Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground .....-0.5V to +6.0V  
Operating Temperature Range.....-55°C to +125°C

Storage Temperature Range.....-55°C to +125°C  
Solder Temperature.....Refer to the IPC/JEDEC  
J-STD-020 Specification.

*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.*

## DC Electrical Characteristics

(-55°C to +125°C; V<sub>DD</sub> = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>DD</sub>	Local power (Note 1)	+3.0		+5.5	V
Pullup Supply Voltage	V <sub>PU</sub>	Parasite power	+3.0		+5.5	V
		Local power	+3.0		V <sub>DD</sub>	
Thermometer Error	t <sub>ERR</sub>	-10°C to +85°C			±0.5	°C
		-30°C to +100°C			±1	
		-55°C to +125°C			±2	
Input Logic-Low	V <sub>IL</sub>	(Notes 1, 4, 5)	-0.3		+0.8	V
Input Logic-High	V <sub>IH</sub>	Local power	+2.2		The lower of 5.5 or V <sub>DD</sub> + 0.3	V
		Parasite power	+3.0			
Sink Current	I <sub>L</sub>	V <sub>IO</sub> = 0.4V	4.0			mA
Standby Current	I <sub>DDS</sub>	(Notes 7, 8)		750	1000	nA
Active Current	I <sub>DD</sub>	V <sub>DD</sub> = 5V (Note 9)		1	1.5	mA
DQ Input Current	I <sub>DQ</sub>	(Note 10)		5		µA
Drift		(Note 11)			±0.2	°C

**Note 1:** All voltages are referenced to ground.

**Note 2:** The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V<sub>PU</sub>. In order to meet the V<sub>IH</sub> spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: V<sub>PU\_ACTUAL</sub> = V<sub>PU\_IDEAL</sub> + V<sub>TRANSISTOR</sub>.

**Note 3:** See typical performance curve in Figure 1. Thermometer Error limits are 3-sigma values.

**Note 4:** Logic-low voltages are specified at a sink current of 4mA.

**Note 5:** To guarantee a presence pulse under low voltage parasite power conditions, V<sub>ILMAX</sub> may have to be reduced to as low as 0.5V.

**Note 6:** Logic-high voltages are specified at a source current of 1mA.

**Note 7:** Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.

**Note 8:** To minimize I<sub>DDs</sub>, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V<sub>DD</sub> - 0.3V ≤ DQ ≤ V<sub>DD</sub>.

**Note 9:** Active current refers to supply current during active temperature conversions or EEPROM writes.

**Note 10:** DQ line is high ("high-Z" state).

Activar V  
Ve a Config

## Anexo 18: Datasheet sensor de vibración SW-420

### Vibration Sensor



This module features an adjustable potentiometer, a vibration sensor, and a LM393 comparator chip to give an adjustable digital output based on the amount of vibration. The potentiometer can be adjusted to both increase and decrease the sensitivity to the desired amount. The module outputs a logic level high (VCC) when it is triggered and a low (GND) when it isn't. Additionally there is an onboard LED that turns on when the module is triggered.

#### Features

- The default state of the switch is close
- Digital output Supply voltage:3.3V-5V
- On-board indicator LED to show the results
- On-board LM393 chip
- SW-420 based sensor, normally closed type vibration sensor
- Dimension of the board: 3.2cm x 1.4cm
- 

Many Applications can be created by measuring Vibration level, but sensing vibration accurately is a difficult job. This article describes about vibration sensor SW-420 and Arduino interface then it may help you to design effort less vibration measurement.

The vibration sensor SW-420 Comes with breakout board that includes comparator LM 393 and Adjustable on board potentiometer for sensitivity threshold selection, and signal indication LED.

## Anexo 19: Datasheet sensor magnético MC-38

### Door & Window Magnetic Sensor Switch for Arduino / IOT / Alarm System



MC-38 Wired Door Window Sensor | Magnetic Switch | Home Alarm System, Recess able style (which means they can be "set into" for example: a door or window). The MC38 can be wired to your door, or window, any where you want a magnetic sensor to alarm when opened.

Metal shield anti-fire ABS, the alarm sounds when the magnets separated. No external power supply is required-- simply connect to wired or wireless alarm control panel GND and N.C ports directly!

#### **SPECIFICATIONS:**

- Connecting Mode: N.C.
- Rated current: 100mA
- Rated voltage: 200VDC
- Operating distance: more than 15mm, less than 25mm
- Rated power: 3W
- Dimension: 28x15x0.9cm
- Cable Length: 30.5cm ± 12mm
- Switch output: normally closed (switch and magnet are together when the switch is closed)

#### **FEATURES:**

- Easy installation Reliable performance
- Good characteristic of abrasion-proof
- Best Choice for you to protect family

## Anexo 20: Hoja técnica de unidad de respaldo (UPS) R-UPR758

### Detección de Problemas

Problema	Posible Causa	Pasos a Tomar
El UPS no se enciende	El voltaje de la batería es menor a 10VDC	Recargue la batería del UPS
	Fallo de tarjeta	Llame para obtener servicio
El UPS siempre está en modo batería	El cordón eléctrico está flojo	Inserte bien el cable
	Breaker abierto	Restablezca el Breaker
	La tensión eléctrica es muy alta, muy baja o ha fallado	La condición es normal
	Fallo de placa	Llame para obtener servicio
El tiempo de autonomía es demasiado corto	La batería no está totalmente cargada	Recargue la batería durante 6 horas por lo menos
	Fallo de tarjeta	Llame para obtener servicio
La alarma suena constantemente	Sobre carga en operación normal	Quite algunas cargas y vuelva a encender el UPS
	Sobre carga en el modo de batería	Quite algunas cargas
	Desgaste de la batería	Cargue la batería

### Especificaciones Técnicas

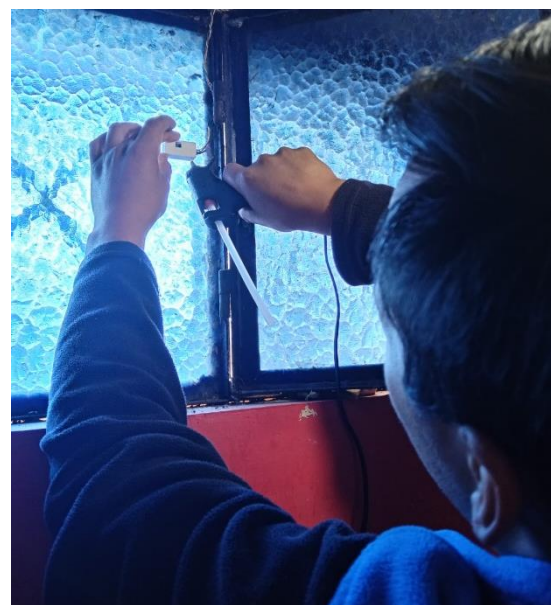
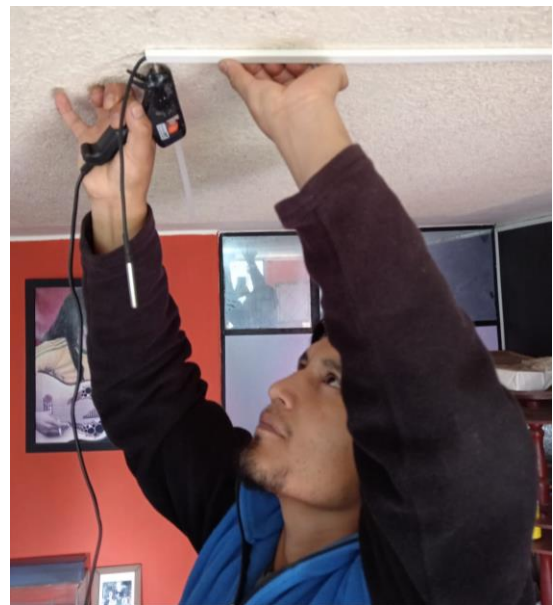
Modelo		R-UPR508	R-UPR758	R-UPR1008
Capacidad		500/250W	750/375W	1000/500W
Entrada	Voltaje	120Vca		
	Rango del Voltaje	80-145Vca		
	Corriente Máxima	5.6A/672W	8A/960W	10A/1200W
	Frecuencia	50Hz o 60Hz (Ajuste Automático)		
Salida	Modo de Regulación de Voltaje ICA	120Vca		
	Frecuencia	50Hz o 60Hz +/- 1Hz		
	Forma de Onda	Onda Senoidal Simulada		
	Tiempo de Transferencia	2ms (Típico)		
	Tipo de Salidas Nema 5-15R	4 Con Regulación + 4 Con Supresor de Picos		
Batería	Breaker	5.6A Max.	8A Max.	10A Max.
	Voltaje	12VDC		
	Tipo	12V/4.5AH 1pza.	12V/7AH 1pza.	12V/9AH 1pza.
	Tiempo de Recarga	4hr a 90% después de completar la recarga		
	Protección	Protección contra Sobrecarga y Protección Contra Sobredescarga		
	Tiempo de Respaldo	15 Minutos	30 Minutos	40 Minutos
	Indicador de Modo de Batería	Dos Tonos Cortos cada 8 segundos		
Alarma	Batería Baje	Cuatro Tonos cada Segundo		
	Sobre carga	Zumbido Continuo		
	Fallo	Pitido Continuo		
Funciones	Reinicio Automático	si		
	Silenciar Alarma	si		
	Indicador de Modo de Carga	si		
	Protección contra Corto Circuito	si		
Ambiente	Supresor de Picos	300 Joules		
	Temperatura	0°C - 40°C		
	Humedad	0 - 95% (sin condensación)		
Dimensiones		300*95*140		

\*Calculado con una PC básico y monitor de 15LCD. Tiempos aproximados.

\*\*Las especificaciones están sujetas a cambios sin previo aviso.

\*\*\*Unidad diseñada para ser usada con computadoras. No se recomienda para uso con cargas lineales.

## Anexo 21: Implementación de sistema de control domótico mediante una aplicación Smartphone



## Anexo 22: Implementación de sistema de control domótico mediante una aplicación Smartphone





## Anexo 23: Aplicación móvil para control del prototipo domótico



**Anexo 24: Manual de usuario del sistema domótico**  
**UNIVERSIDAD TÉCNICA DEL NORTE**



Facultad de Ingeniería en Ciencias Aplicadas Carrera de Ingeniería en  
Mantenimiento Eléctrico

**MANUAL DE USUARIO DEL PROTOTIPO DE SISTEMA DE CONTROL  
DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE**

Trabajo de grado previo la obtención del título de Ingeniero en Mantenimiento  
Eléctrico

Autor:

Tigse Cuzco Angel Jhonathan

Director:

Ing. Francisco Roberto Naranjo Cobo MSc.

Ibarra – Ecuador

2023

## Contenido

Introducción.....	1
1. Encendido del sistema domótico .....	1
2. Descarga de aplicación móvil de la plataforma App Inventor.....	2
3. Características de los elementos de la aplicación domótica.....	4
4. Apagado del sistema domótico .....	6

## ÍNDICE DE FIGURAS

Figura 1 Plataforma App Inventor.....	2
Figura 2 Ingreso de credenciales .....	2
Figura 3 Plataforma App Inventor.....	3
Figura 4 Plataforma App Inventor.....	3
Figura 5 Interfaz de aplicación domótica.....	4
Figura 6 Interfaz de alarma.....	6
Figura 7 Interfaz de alarma.....	6

## ÍNDICE DE TABLAS

Tabla 1 Características de los elementos del sistema domótico .....	4
---	---

# MANUAL DE USUARIO PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE

## 2 Introducción

La información que se detalla a continuación es una guía del funcionamiento del prototipo domótico, como descargar la aplicación móvil para el sistema de control domótico y a su vez encontramos en el apartado final las características de funcionamiento del sistema, cabe recalcar que los pasos a seguir deben ser realizados al pie de la letra para una correcta descarga y evitar inconvenientes. Esta aplicación se encuentra disponible únicamente para dispositivos Android.

**Nota:** Es importante mencionar antes de seguir los siguientes pasos, revisar el manual técnico del prototipo domótico en el literal “**Configuración de tarjeta de control Raspberry Pi**”, donde se detalla el procedimiento para la conexión a internet de las placas de control; Raspberry Pi, ESP32 y Esp8266

### 1. Encendido del sistema domótico

Como primer punto para poner en funcionamiento el prototipo domótico, conecte la central de control Raspberry Pi a una fuente de alimentación de 110V, 60 Hz con su respectivo cargador de 5VDC a 3 Amperios, luego el sistema operativo tarda alrededor de 2 minutos en conectarse a la red de internet y ya está lista para su funcionamiento.

Como segundo paso, conecte las 6 placas electrónicas del sistema domótico a una fuente de alimentación de 110V, 60 Hz con sus respectivos cargadores de 5VDC a 2 Amperios independientemente, ya encendidas las placas se conectan automáticamente a la central de control Raspberry Pi, este proceso tarda alrededor de 3 minutos.

Por último, luego de seguir estos dos pasos, el sistema domótico ya estaría listo para ser utilizado.

**Importante:** en el caso que alguna placa no encienda asegúrese que los cargadores sean los adecuados para cada placa y revise que los cables se encuentren bien conectados, otro inconveniente que se pueda presentar es que las placas no se conecten a internet, una solución rápida sería presionar el botón de “**RESET**” que posee cada una de las placas.

**Nota:** el funcionamiento del sistema se lo realiza por medio de la aplicación móvil la cual se debe descargar siguiendo los pasos que se detallan a continuación:

## 2. Descarga de aplicación móvil de la plataforma App Inventor

Para descargar la aplicación de control domótico se detallan los siguientes pasos:

- Como primera instancia debe ingresar a un navegador web, de preferencia Google Chrome.
- Ingrese a la barra de búsqueda y diríjase a la plataforma de App Inventor, aquí seleccione la siguiente opción como se muestra en la Figura 1.

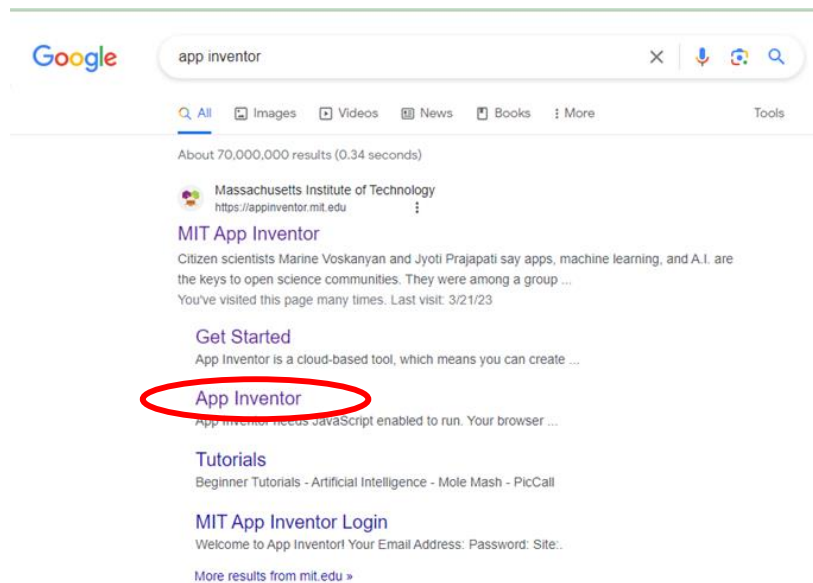


Figura 84 Plataforma App Inventor  
Fuente: (Autor, 2023)

- Tercero, una vez selecciona la opción “App Inventor” se abre una ventana donde el usuario debe iniciar sesión con las siguientes credenciales de correo electrónico y su contraseña que se detalla a continuación en la Figura 2.

A screenshot of the Google sign-in page for App Inventor. The page title is 'Iniciar sesión con Google'. The main heading is 'Inicia sesión'. Below the heading, there is a link: 'Ir a Autenticación de inventor de aplicaciones'. There is a text input field for 'Correo electrónico o teléfono'. Below the input field, there is a link: '¿Has olvidado tu correo electrónico?'. Below that, there is a paragraph of text: 'Para continuar, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu foto de perfil con App Inventor Authentication.' At the bottom left, there is a link: 'Crear cuenta'. At the bottom right, there is a blue button: 'Siguiente'. Below the sign-in form, there is a blue-bordered box containing the login credentials: 'Usuario: [angeltigseutn@gmail.com](mailto:angeltigseutn@gmail.com)' and 'Contraseña: Domoticautn2023'.

Figura 85 Ingreso de credenciales  
Fuente: (Autor, 2023)

- Cuarto, una vez ingresado la identificación de usuario se despliega la siguiente ventana donde se puede visualizar el archivo (.apk) de la aplicación domótica llamado “AplicacionDomotica” como se observa en la Figura 3.

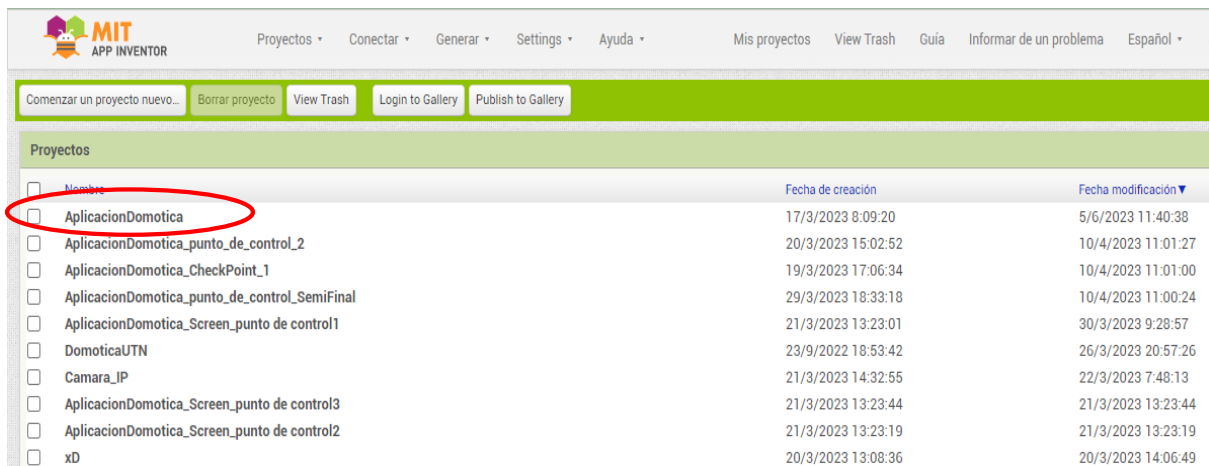


Figura 86 Plataforma App Inventor  
Fuente: (Autor, 2023)

- Quinto, se despliega una pantalla con el diseño de la aplicación domótica, para descargar el archivo, seleccione la opción que dice “Generar” Android App (.apk) y puede seleccionar la opción para descargar directamente en el dispositivo que se esté utilizando o escanearlo desde el celular mediante la opción de código QR.

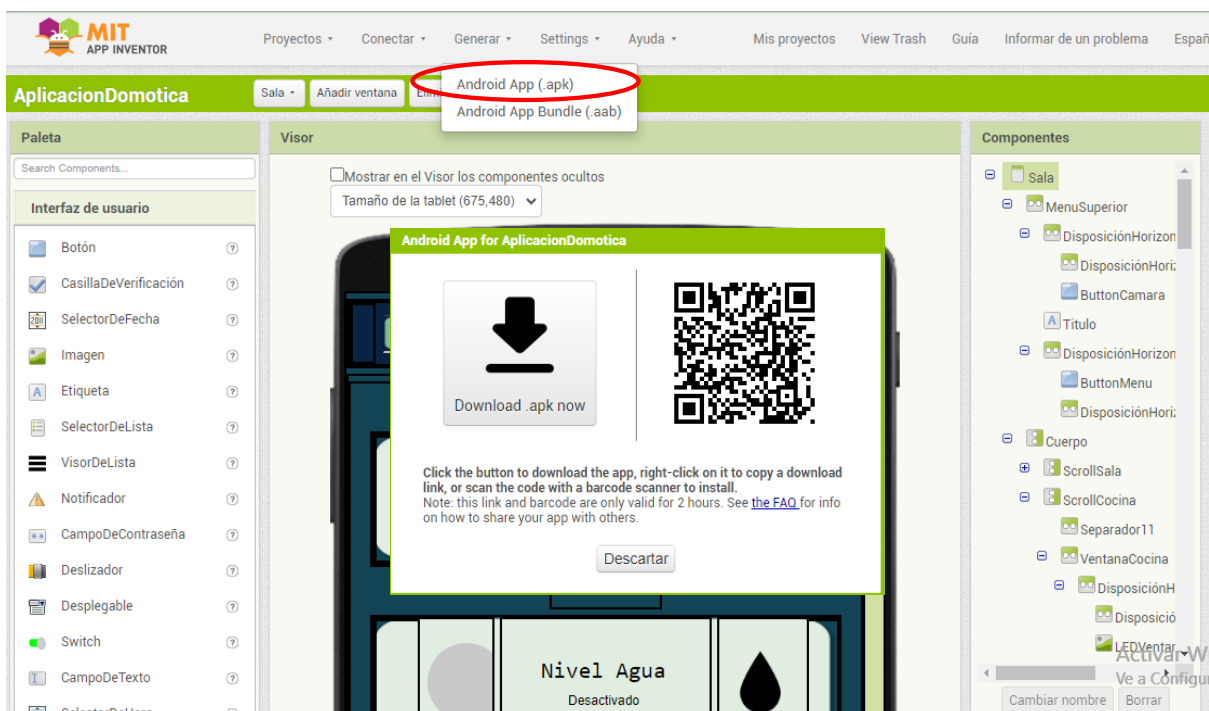


Figura 87 Plataforma App Inventor  
Fuente: (Autor, 2023)

- Por último, instale como cualquier otra aplicación y debe aparecer el siguiente logo en su celular como se muestra en la Figura 5, al abrir la aplicación domótica aparece la siguiente interfaz de control donde se encuentran varias aplicaciones domóticas que se pueden controlar y monitorear.

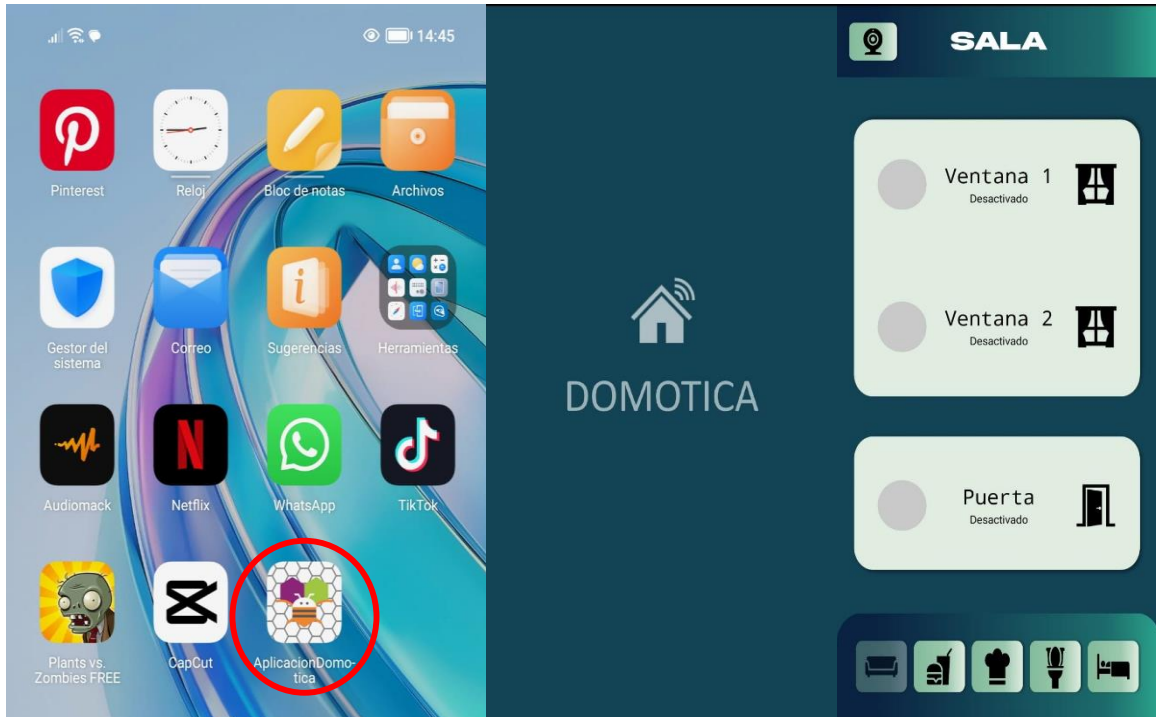

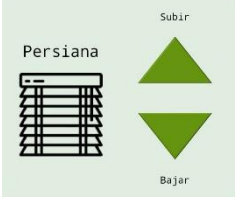




Figura 88 Interfaz de aplicación domótica  
Fuente: (Autor, 2023)

### 3. Características de los elementos de la aplicación domótica

En la Tabla 1 se detalla las funciones que cumplen cada una de las soluciones domóticas que contiene la aplicación móvil para Android.

Tabla 23 Características de los elementos del sistema domótico

Logo	Funcionamiento
	<p>Aquí se aprecia 5 botones que corresponden a la sala, comedor, cocina, baño y dormitorios respectivamente, dentro de estos espacios se encuentran varias funciones domóticas.</p>
	<p>Esta opción permite subir y bajar las cortinas o persianas de acuerdo las disposiciones del usuario.</p>

	<p>Los 5 espacios físicos a excepción del dormitorio 2 contienen el ícono de alarma la cual permite activar o desactivar la misma de acuerdo a las necesidades del usuario.</p>
	<p>La opción de dimmer nos permite regular la intensidad de luz, así como el encendido y apagado de las mismas.</p>
	<p>La opción de carga permite encender y apagar electrodomésticos que no superen los 10 Amperios en corriente alterna (AC).</p>
	<p>Este ícono representa la interfaz de las 2 cámaras de seguridad ESP32 CAM.</p>
	<p>Este gráfico representa el sensor magnético en la ventana, se activa con la apertura de la ventana.</p>
	<p>Este gráfico representa el sensor magnético en la puerta, se activa con la apertura de la puerta.</p>
	<p>Este gráfico representa el sensor de vibración, se activa al sentir un golpe en la venta sea leve o fuerte.</p>
	<p>Este ícono representa el nivel de agua, el sensor se activa al detectar fugas de agua.</p>
	<p>Este ícono representa las fugas de gas, el sensor se activa ante la presencia de gas doméstico.</p>
	<p>Este ícono representa la presencia de humo, el sensor se activa al detectar dióxido de carbono CO2.</p>

Fuente: (Autor, 2023)

**Importante:** el sistema de seguridad cuenta con una alarma para cada espacio físico, la cual se activa con la señal de los diferentes sensores que conforman el sistema domótico, para la desactivación de esta se lo puede realizar presionando en el ícono de **“Alarma”** como se muestra en la siguiente Figura 6.





Figura 89 Interfaz de alarma  
Fuente: (Autor, 2023)

#### 4. Apagado del sistema domótico

Las placas de control de la sala, comedor, cocina, baño y dormitorios se las puede desconectar de su fuente de alimentación sin ningún problema. Sin embargo, la placa de control principal Raspberry Pi se debe apagar desde su pantalla principal en el escritorio seleccionando la siguiente opción de “**Shutdown**” en el ícono de aplicaciones del menú como se aprecia en la Figura 7.

**Nota:** Para visualizar la pantalla de la Raspberry Pi se puede proyectar con un cable HDMI hacia un monitor o también se puede utilizar el programa” **VNC Viewer**” que permite controlar un equipo remoto.

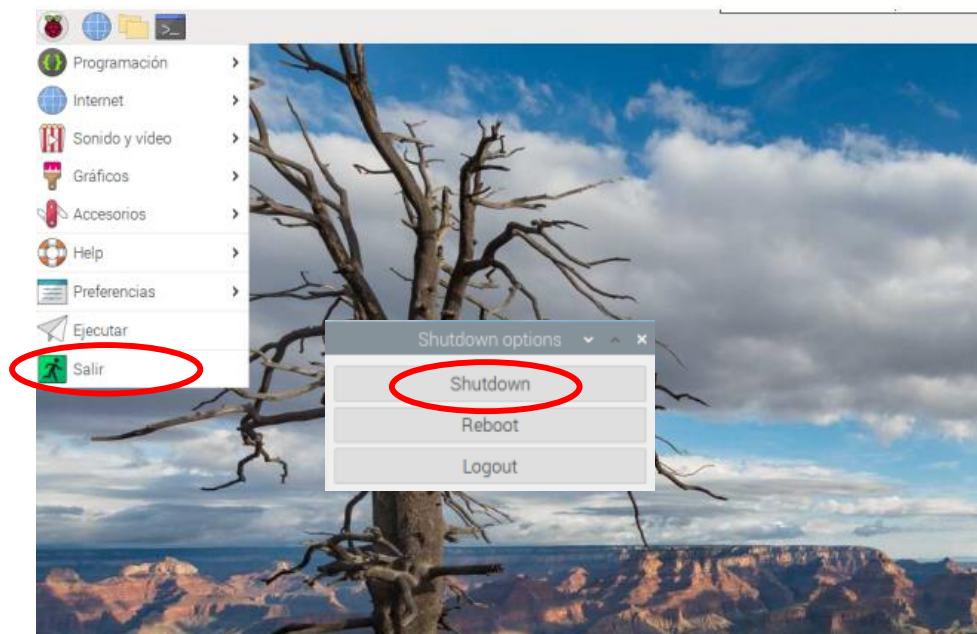


Figura 90 Pantalla principal de la Raspberry Pi  
Fuente: (Autor, 2023)

**Importante:** No desconecte la placa Raspberry Pi directamente de su fuente de alimentación principal, esto podría dañar los archivos que se encuentran en la tarjeta microSD provocando un funcionamiento incorrecto.

Anexo 25: Manual técnico del sistema domótico

## UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas Carrera de Ingeniería en  
Mantenimiento Eléctrico

### MANUAL TÉCNICO DEL PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE

Trabajo de grado previo la obtención del título de Ingeniero en Mantenimiento  
Eléctrico

Autor:

Tigse Cuzco Angel Jhonathan

Director:

Ing. Francisco Roberto Naranjo Cobo MSc.

**Contenido**

Introducción.....9

1. Configuración de software del sistema domótico .....9

    1.1. Configuración de tarjeta central Raspberry Pi 3.....9

    1.2. Configuración de tarjetas de control ESP32 Y ESP8266 para los espacios físicos que comprenden el prototipo domótico ..... 11

2. Elementos electrónicos utilizados en la construcción del prototipo domótico ..... 14

3. Conexión de las placas electrónicas de las áreas del sistema domótico .....20

    3.1. Esquema de conexión espacio físico cocina.....20

    3.2. Esquema de conexión espacio físico dormitorios .....21

    3.3. Esquema de conexión espacio físico sala.....22

    3.4. Esquema de conexión espacio físico comedor .....23

    3.5. Esquema de conexión espacio físico baño .....24

    3.6. Esquema de conexión Raspberry pi 3 modelo B+ .....25

4. Mantenimiento de los elementos del sistema domótico .....25

5. Mantenimiento preventivo .....26

6. Mantenimiento del sistema eléctrico .....26

# MANUAL TÉCNICO PARA EL PROTOTIPO DE SISTEMA DE CONTROL DOMÓTICO MEDIANTE UNA APLICACIÓN SMARTPHONE

## 1. Introducción

La información que se detalla a continuación, es una guía de cómo se debe configurar el prototipo domótico para su funcionamiento, los elementos electrónicos con sus principales características, los esquemas de conexión de cada una de las áreas de aplicación y el tipo de mantenimiento que se puede realizar al sistema domótico en el caso que llegara a fallar alguno de estos elementos.

### 1. Configuración de software del sistema domótico

Como primer paso se tiene el encendido del prototipo, el cual se debe conectar a una fuente de 110 V de corriente alterna con su respectivo cargador de 5 V a 3 A, tanto la placa principal de control Raspberry Pi 3 y sus otras placas ESP32 Y ESP8266 con su cargador de 5 V a 1 o 2 A.

#### 1.1. Configuración de tarjeta central Raspberry Pi 3

Como primer punto se debe conectar a la red de internet la tarjeta de control Raspberry Pi siguiendo los siguientes pasos:

- Primero: diríjase a la pantalla principal de la Raspberry Pi y de clic sobre el ícono de internet como se muestra en la Figura 1, aquí ingrese la clave del internet al que se va a conectar.

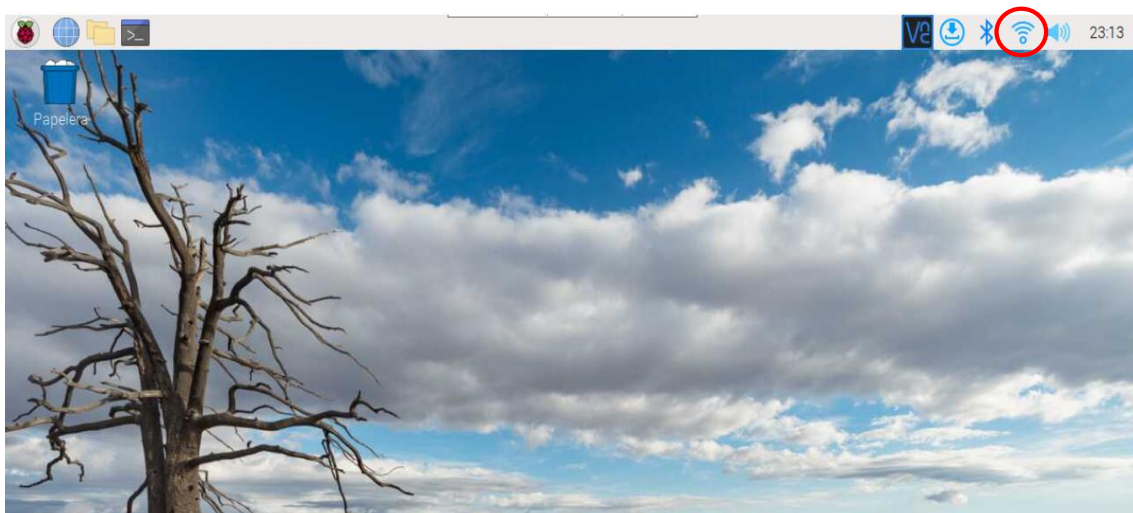


Figura 91 Pantalla principal de Raspberry Pi

Segundo: abra la ventana de comandos que se muestra en la Figura 2, aquí ingrese los siguientes comandos para cambiar la IP dentro de la Raspberry Pi:

```
raspberrypi@raspberrypi:~ $ cd /
raspberrypi@raspberrypi:/ $ ls
raspberrypi@raspberrypi:/ $ cd etc
raspberrypi@raspberrypi:/etc $ ls
raspberrypi@raspberrypi:/etc $ sudo nano dhcpd.conf
```

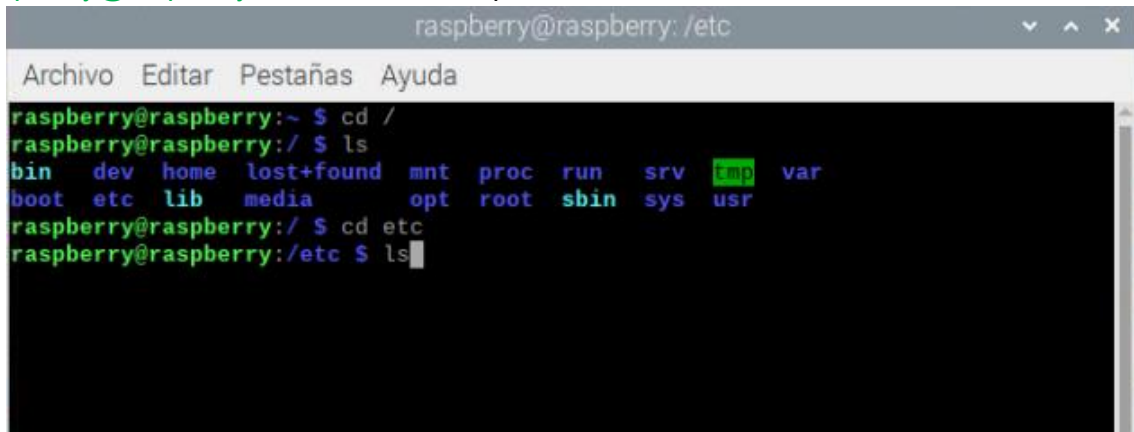


Figura 92 Ventana de comandos

- Tercero: una vez ingresado los comandos anteriores se despliega la siguiente pantalla que se puede apreciar en la Figura 3, aquí tiene que cambiar a la nueva IP donde se encuentra señalado con las flechas rojas.

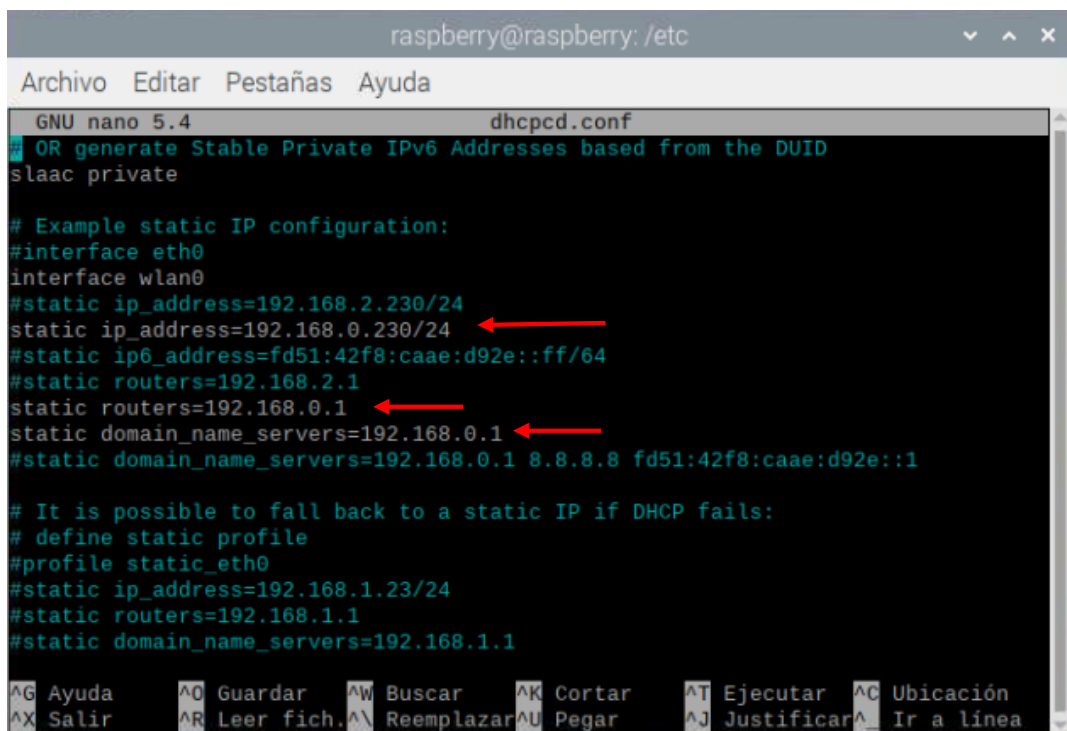


Figura 93 Interfaz en Raspberry Pi para cambiar IP de internet

- Cuarto: una vez cambiada a la IP nueva cierre la pantalla de comandos y reinicie la Raspberry Pi para evitar inconvenientes de conexión, luego diríjase a la parte superior derecha de la pantalla principal y de clic en el ícono que dice Vnc viewer como se muestra en la Figura 4 aquí se despliega la siguiente pantalla con la nueva IP a la que posteriormente se debe conectar las placas ESP32 y ESP8266.

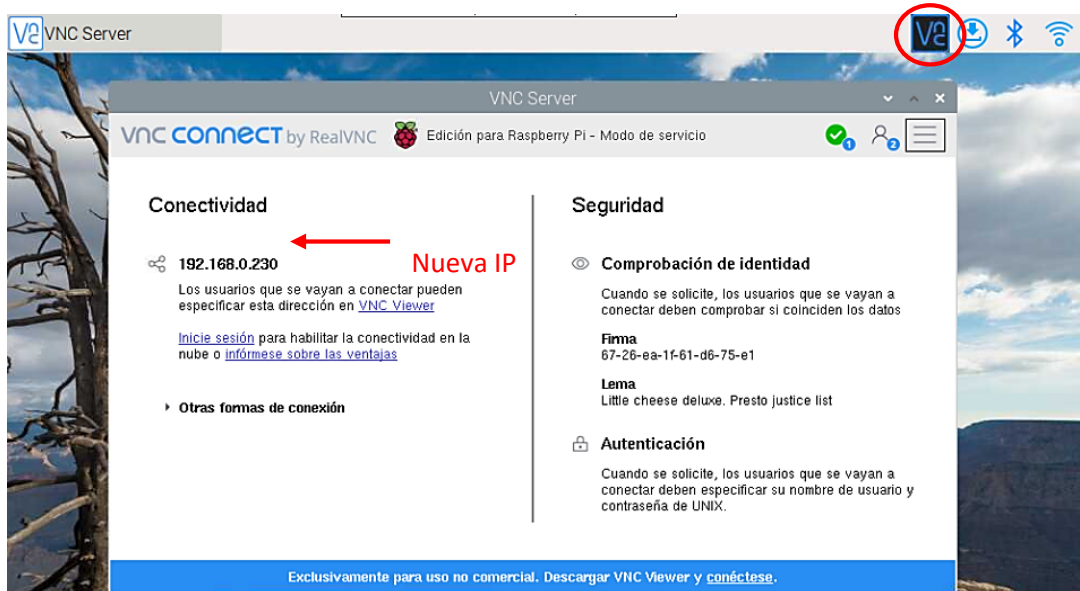


Figura 94 Interfaz de VNC Viewer

## 1.2. Configuración de tarjetas de control ESP32 y ESP8266 para los espacios físicos que comprenden el prototipo domótico

Una vez ya configurada la Raspberry Pi 3, ingresamos la nueva IP en las tarjetas de control que se encuentran conectadas a los sensores y actuadores realizando los siguientes pasos:

- Primero: abrimos el entorno de desarrollo (IDE) de Arduino, aquí debemos ingresar la nueva IP y las credenciales de la red de internet a la que se vaya a conectar. Es importante mencionar que en las tres partes donde debe ingresar la nueva IP solo hay que cambiar el tercer dígito, ya que para cada espacio físico del sistema domótico el cuarto dígito no presenta cambios.

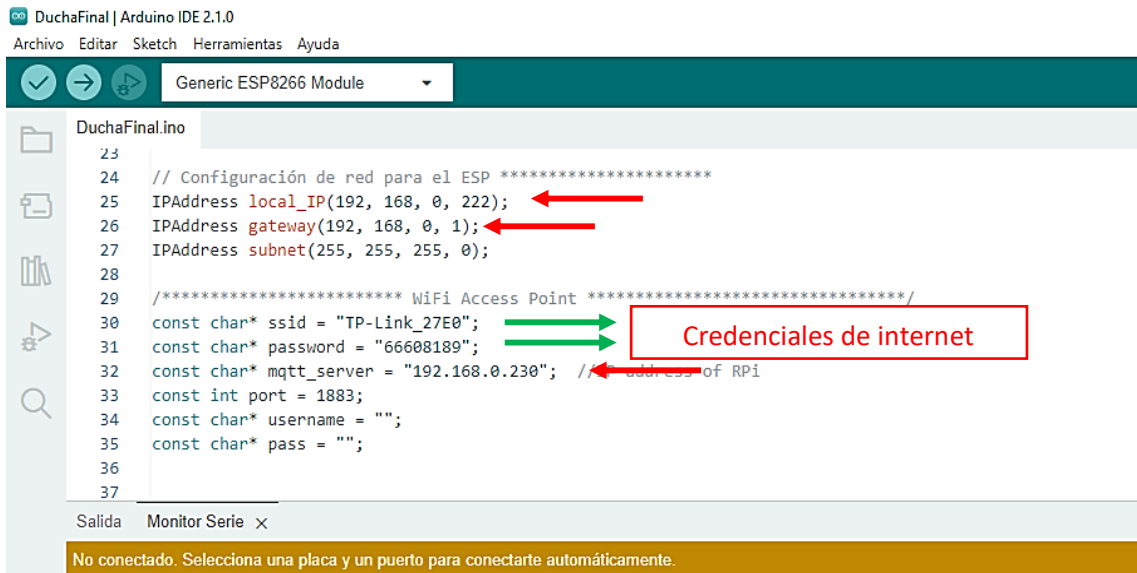


Figura 95 Interfaz de Arduino

- Una vez ya ingresados los datos para la conexión a internet, seleccione el tipo de tarjeta de control que se vaya a utilizar que se encuentra en la parte superior en herramientas como se aprecia en la Figura 5, aquí debemos seleccionar las siguientes placas; Generic ESP8266 Module, Node32s y AI Thinker ESP32-CAM, estas serán seleccionadas de acuerdo a la placa que se vaya a programar.

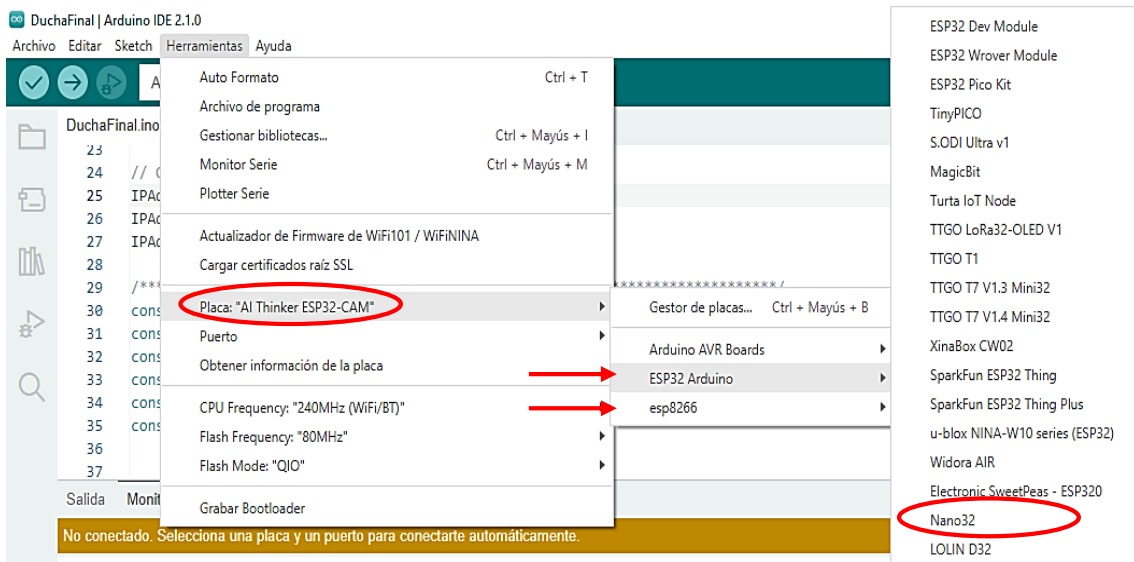


Figura 96 Interfaz de Arduino

- Luego, cargamos el código desarrollado en el software Arduino para los 5 espacios físicos que conforman el sistema domótico, estos son; una cocina, un comedor, una sala, un baño y dos dormitorios, para cargar el código se envía a verificar que no presente errores y hacemos clic en la opción de cargar como se puede apreciar en la Figura 6.

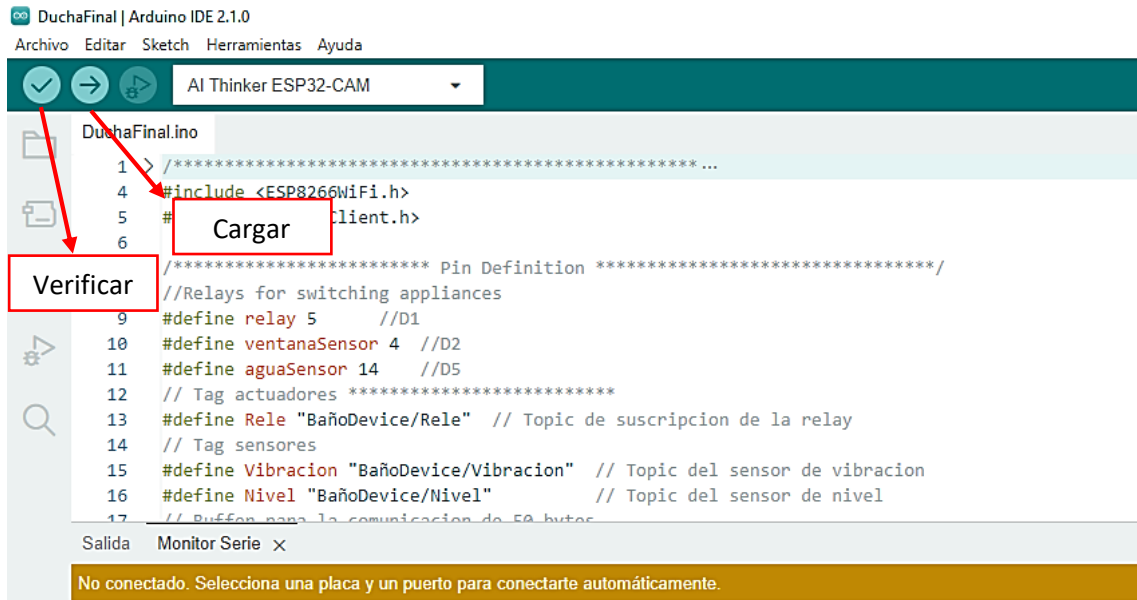


Figura 98 Interfaz de Arduino

- Por último, ya cargado el código a las diferentes placas verificamos que se hayan conectado con la central de control Raspberry Pi. Una vez conectado las tarjetas se mostrará la siguiente información indicada en la Figura 7, en el “monitor serial” se visualiza un mensaje en el que indica el espacio físico y los elementos que están conectados a la placa de control.

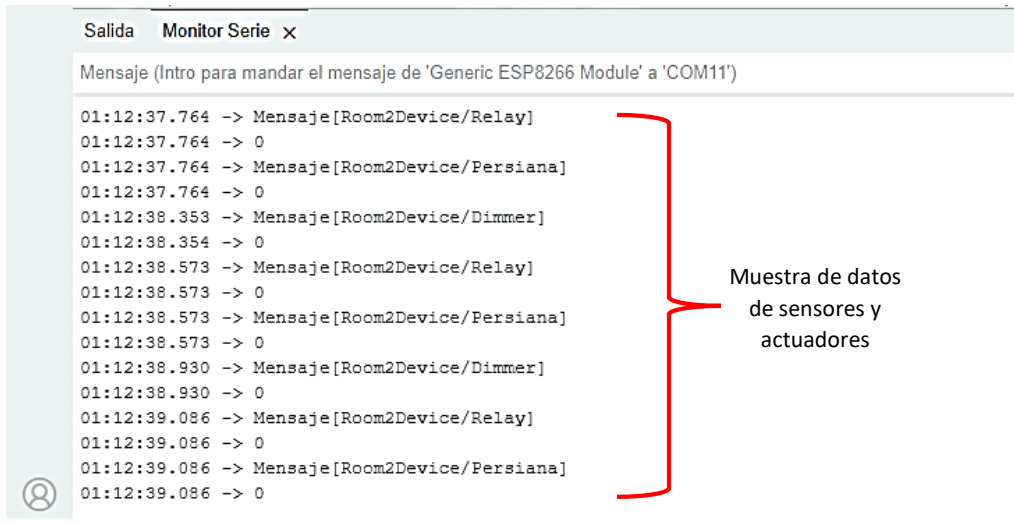


Figura 99 Conexión de placas de control con la central Raspberry Pi 3.

**Nota:** El paso anterior debe repetirse al conectar la tarjeta de control de cada área física que comprende el sistema domótico, de esta manera queda completamente activado el sistema y listo para su funcionamiento.

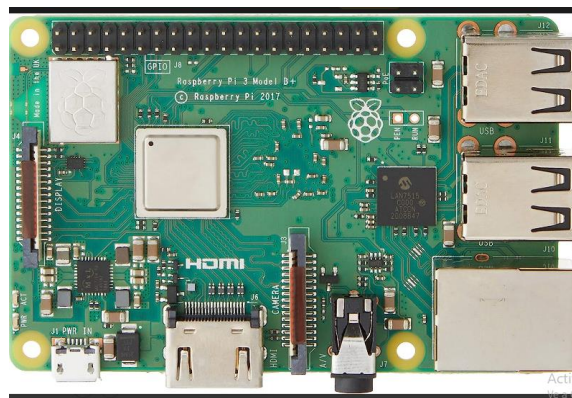


## 2. Elementos electrónicos utilizados en la construcción del prototipo domótico

Para la construcción del prototipo domótico se utilizó varios elementos electrónicos tales como; tarjetas de control, sensores y actuadores con el fin de cumplir las necesidades de control y seguridad dentro de la vivienda, a continuación, se describe cada uno de los componentes con sus características técnicas y su funcionamiento dentro del sistema domótico.

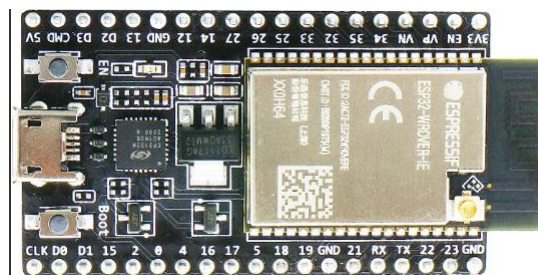
### a. Tarjeta principal de control Raspberry Pi

Para el control principal del sistema domótico se utilizó una tarjeta de desarrollo Raspberry Pi 3 Modelo B+, sus principales características son; un procesador de 64-bit SoC a 1.4GHz con una memoria RAM de 1GB, para la conectividad inalámbrica cuenta con 2.4GHz y 5GHz IEEE 802.11.b/g/n/a/c Wireless LAN, Bluetooth 4.2 y se debe alimentar con una fuente de 5 VDC a 3 A, para conocer más características del elemento dirijase al Anexo 9.



### b. Tarjeta de control ESP32

La tarjeta ESP32 se utilizó para el control del espacio físico sala, a esta placa se conectaron varios sensores y actuadores que comprenden el sistema domótico dentro de esta área, las principales características de la tarjeta son; un procesador Tensilica LX6 Dual-Core con una memoria SRAM de 512 kB, una conexión inalámbrica 802.11 b/g/n, Bluetooth Classic / LE y su voltaje de operación es de 5 V a 1 A, para mayor información de las características del elemento dirijase al Anexo 11.



### c. Tarjeta de control ESP8266

La tarjeta ESP8266 se utilizó para el control de los espacios físicos cocina, comedor, baño y dormitorios dentro de esta se conectaron varios sensores y actuadores dependiendo del área de aplicación del sistema domótico, las principales características de la tarjeta son; un procesador de 32-bit Tensilica Xtensa LX106 con una memoria RAM de 96 kB, utiliza el protocolo inalámbrico 802.11b/g/n y su voltaje de operación es de 5 V a 1 A, para mayor información de las características del elemento diríjase al Anexo 10.



### d. Sensor Magnético

El sensor magnético MC-38 se instaló en las puertas y ventanas dentro de todas las áreas que comprenden el sistema domótico con el fin de alertar al usuario por medio de una sirena ante la presencia de intrusos que quieran entrar al domicilio, este sensor cuenta con las siguientes características; voltaje máximo de funcionamiento 100 VDC con una corriente de consumo de 100 mA, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



### e. Sensor de humo

El sensor MQ-135 se instaló dentro del área de la cocina para alertar al usuario la presencia de humo provocado por un posible incendio dentro de la vivienda, el voltaje de operación de este sensor es de 5 VDC con un consumo de corriente de 160 mA, para mayor información de sus características diríjase al Anexo 16, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



#### **f. Sensor de gas**

El sensor MQ-6 se instaló dentro del área de la cocina para alertar al usuario la presencia de una posible fuga de gas dentro de la vivienda, el voltaje de operación de este sensor es de 5 VDC con un consumo de corriente de 180 mA, para mayor información de las características técnicas diríjase al Anexo 15, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



#### **g. Sensor de inundación**

El sensor capacitivo de nivel de líquidos se instaló en las áreas de la cocina y el baño para alertar al usuario la presencia de fugas de agua dentro de la vivienda, las características principales de este sensor son: voltaje de operación 5 VDC, corriente de consumo 150 mA y una temperatura de operación de 0 a 40 °C, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



#### **h. Sensor de vibración**

El sensor SW-420 se instaló en las áreas de la sala, cocina, baño y dormitorio para alertar al usuario por medio de una sirena la ruptura de un vidrio ante la presencia de intrusos a la vivienda, las características de este sensor son las siguientes: voltaje de operación de 3.3 a 5 VDC y consumo de corriente de 15 mA, para mayor información de las características técnicas del sensor diríjase al Anexo 18.



#### **i. Sensor de temperatura**

El sensor de temperatura DS18B20 se instaló exclusivamente en el área de la sala, para informar al usuario la temperatura ambiente en tiempo real dentro de la casa. El sensor cuenta con las siguientes características: voltaje máximo de funcionamiento 6 VDC y temperatura de operación de  $-55\text{ }^{\circ}\text{C}$  a  $125\text{ }^{\circ}\text{C}$ , para mayor información del sensor diríjase al Anexo 17.



#### **j. Actuador relé**

El relé JQC-3FF-S-Z se instaló dentro de todas las áreas que conforman el sistema domótico, para el encendido y apagado de cargas en corriente alterna las principales características de este actuador son las siguientes: voltaje de funcionamiento de 5 VDC con un consumo de corriente de 90 mA y una corriente máxima carga de 10 A, para más detalles de las características diríjase al Anexo 13.



#### k. Actuador dimmer

El Módulo Dimmer AC con cruce por cero TTL se instaló dentro de todas las áreas que comprenden el sistema domótico, para la atenuación de luces con el fin de brindar un ambiente agradable al usuario sus principales características son: voltaje de funcionamiento 5 VDC, voltaje de carga 110/220 V, corriente máxima de carga de 8 A y una potencia de carga de 1000 W, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



#### l. Actuador de persianas

Los motores 28BYJ-48 se utilizaron dentro de las áreas de la sala y los dormitorios, para la apertura y cierre de persianas con el fin de brindar al usuario una mejor comodidad al momento de realizar esta tarea, las características principales de estos motores son: voltaje de operación de 5 a 12 VDC con una corriente de consumo de 80 mA, para mayor información del elemento diríjase al Anexo 14.



#### m. Sirena alarma

La sirena tipo cuerno se instaló dentro del área de la sala, para alertar al usuario la presencia de algún inconveniente en los sensores que se encuentran instalados dentro de la vivienda, sus principales características son: voltaje de funcionamiento 12 VDC con una corriente de consumo de 90 mA, se recomienda utilizar la interfaz de Arduino (IDE) para su programación.



#### n. Fuente de respaldo de energía

La fuente de respaldo UPS CDP R-UPR758 750VA-375W, se utilizó dentro del prototipo domótico para brindar un respaldo de energía en el caso que la fuente principal de alimentación 110 VAC presente inconvenientes, las principales características son: voltaje de operación 110 VAC con un respaldo de energía de 1 hora, una potencia de carga de 960 Watts y una corriente de carga de 8 A, para mayor información diríjase al Anexo 20.

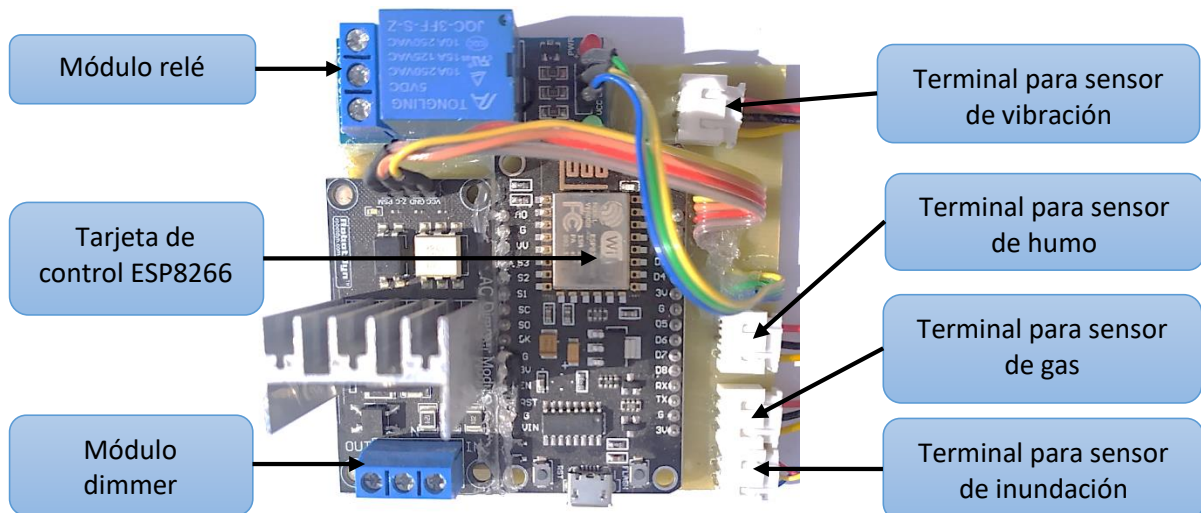
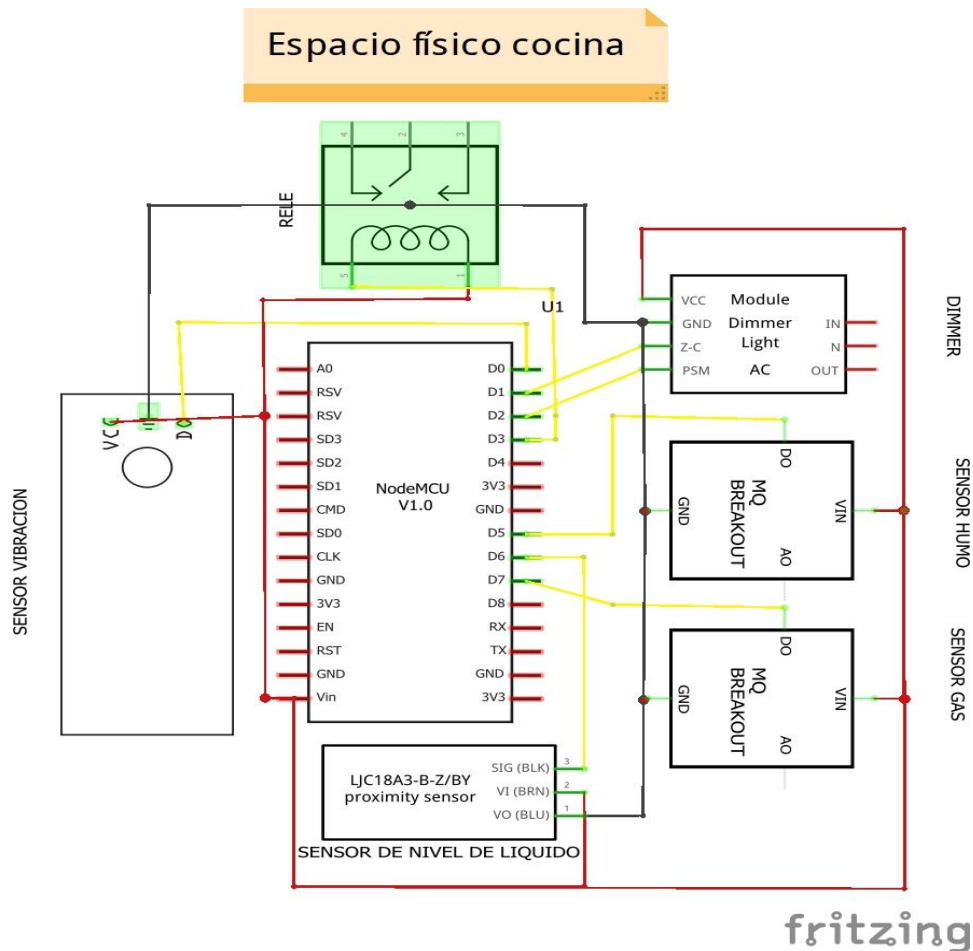


**Importante:** La mayoría de estos sensores y actuadores se los puede reemplazar por nuevas versiones, teniendo en cuenta las mismas características de operación y algunas características propias de estos elementos que se detallan en cada uno de sus Anexos. Además, para su compra se puede encontrar libremente en tiendas electrónicas físicas o en línea.

### 3. Conexión de las placas electrónicas de las áreas del sistema domótico

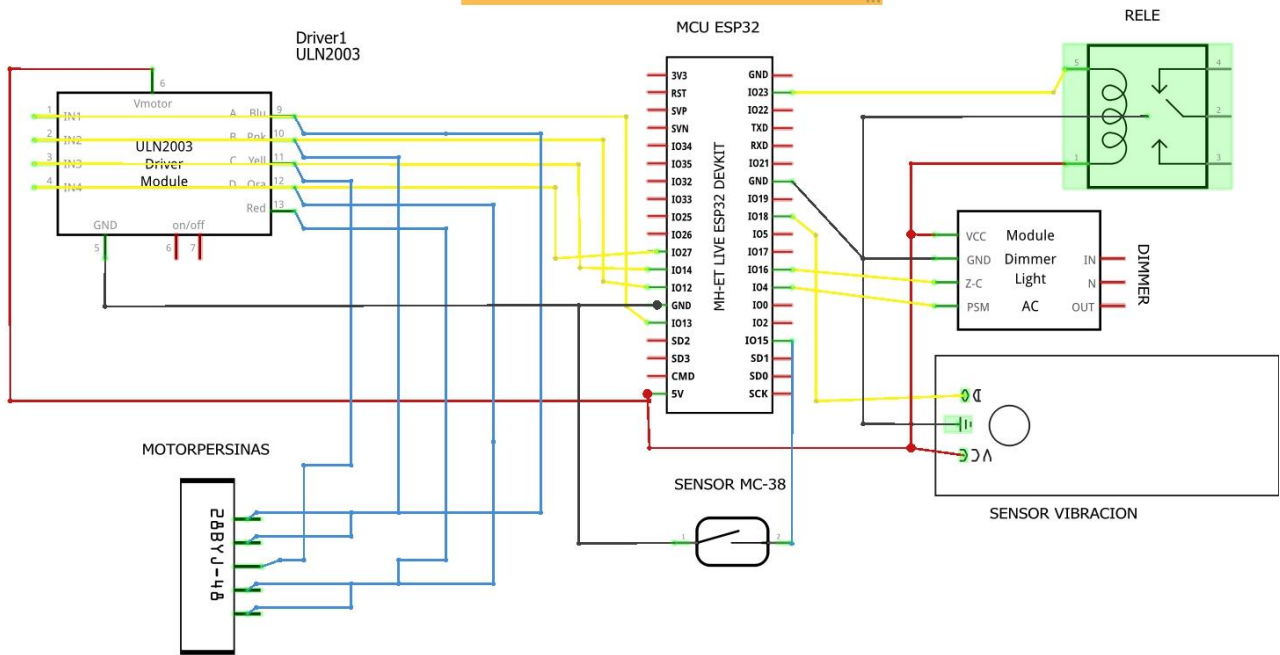
A continuación, se detallan los esquemas de conexión de cada una de las áreas de aplicación del sistema domótico.

#### 3.1. Esquema de conexión espacio físico cocina

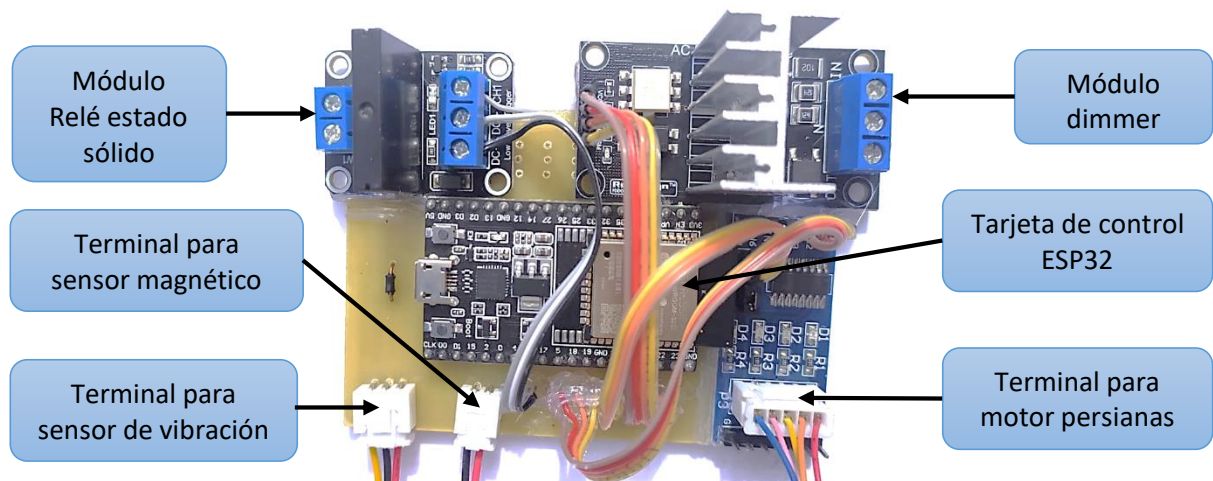


### 3.2. Esquema de conexión espacio físico dormitorios

#### Espacio físico dormitorios

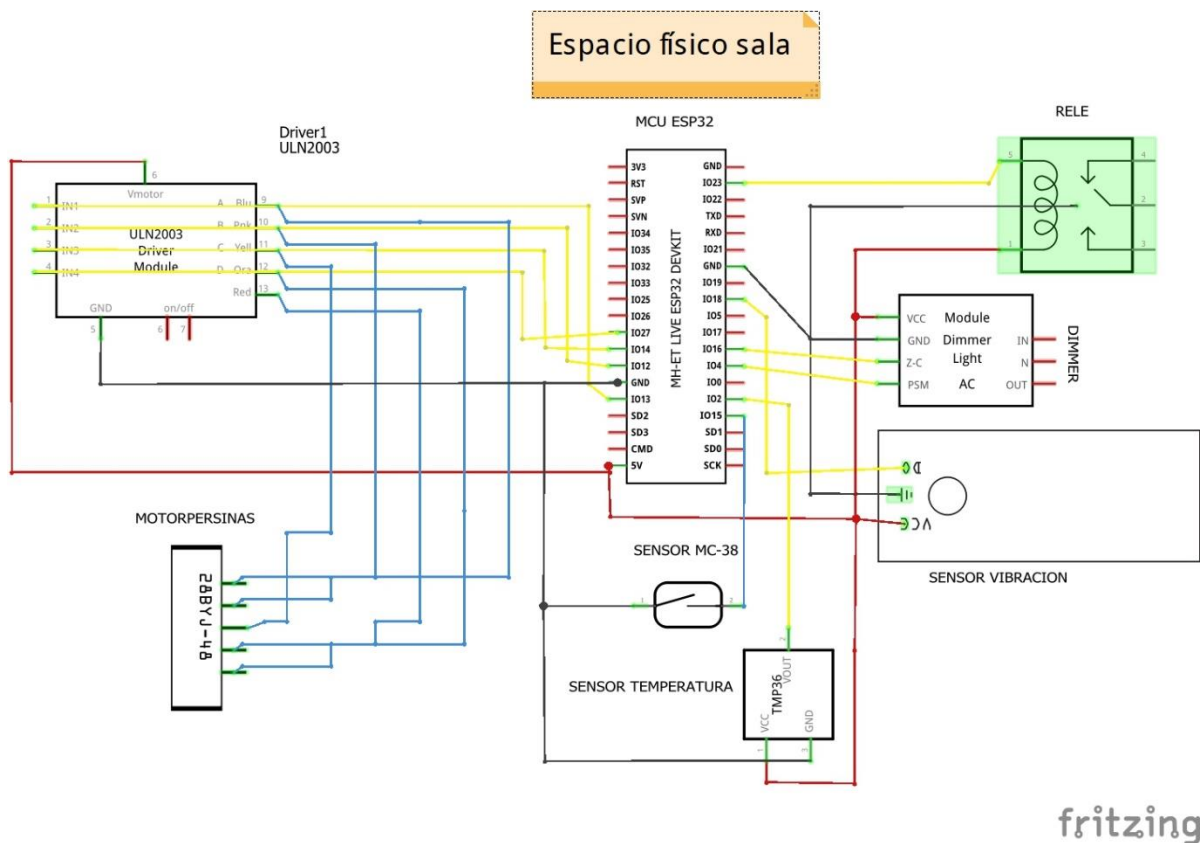


fritzing

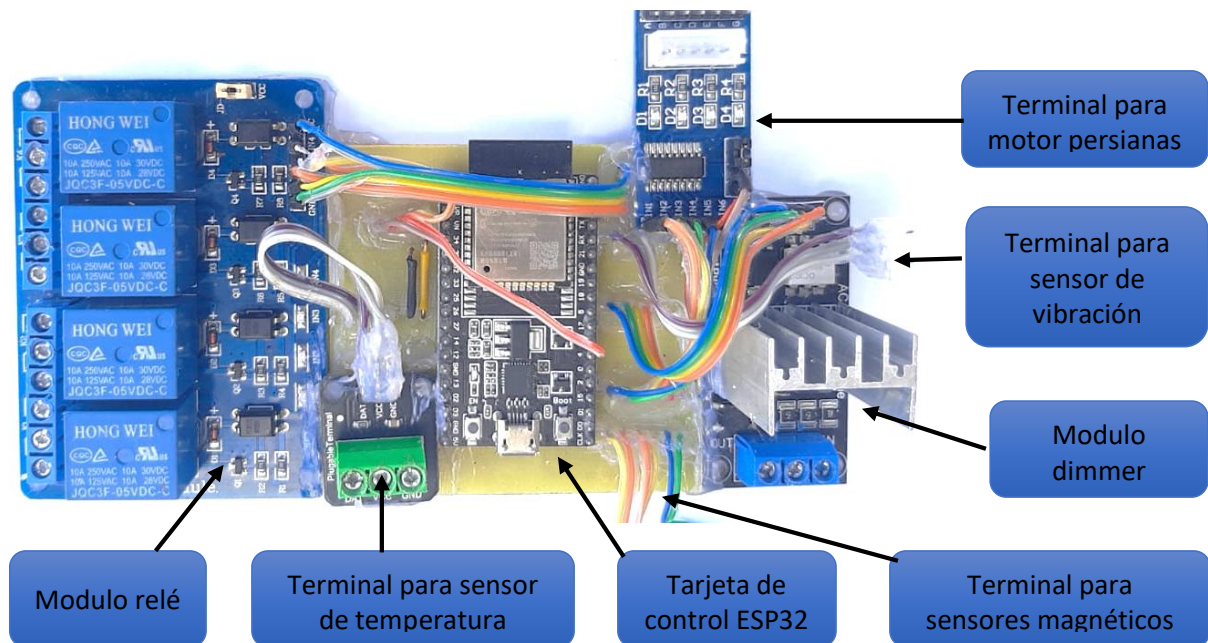




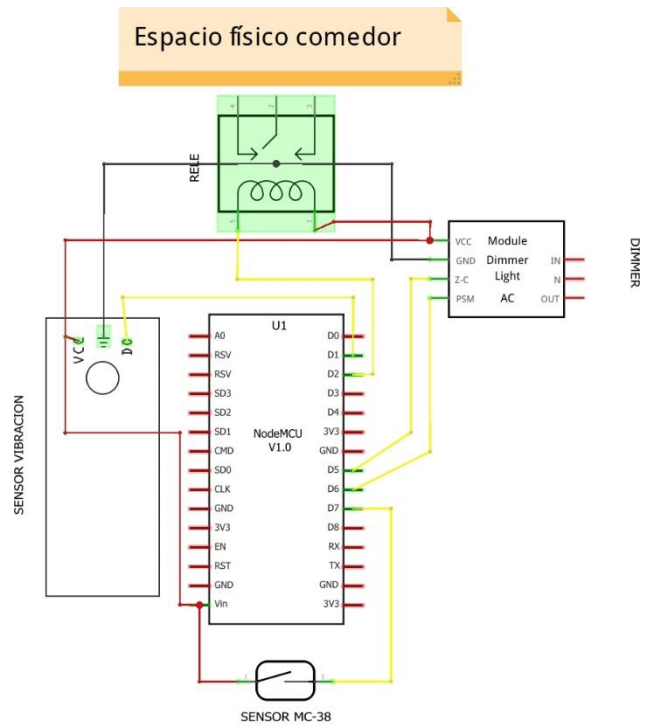
### 3.3. Esquema de conexión espacio físico sala



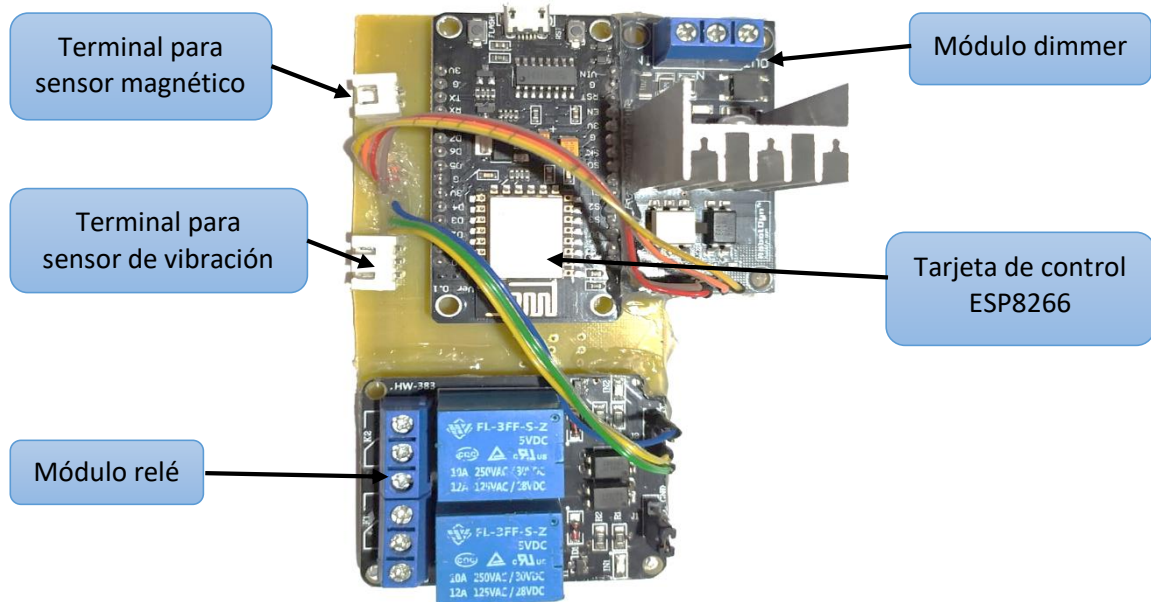
fritzing



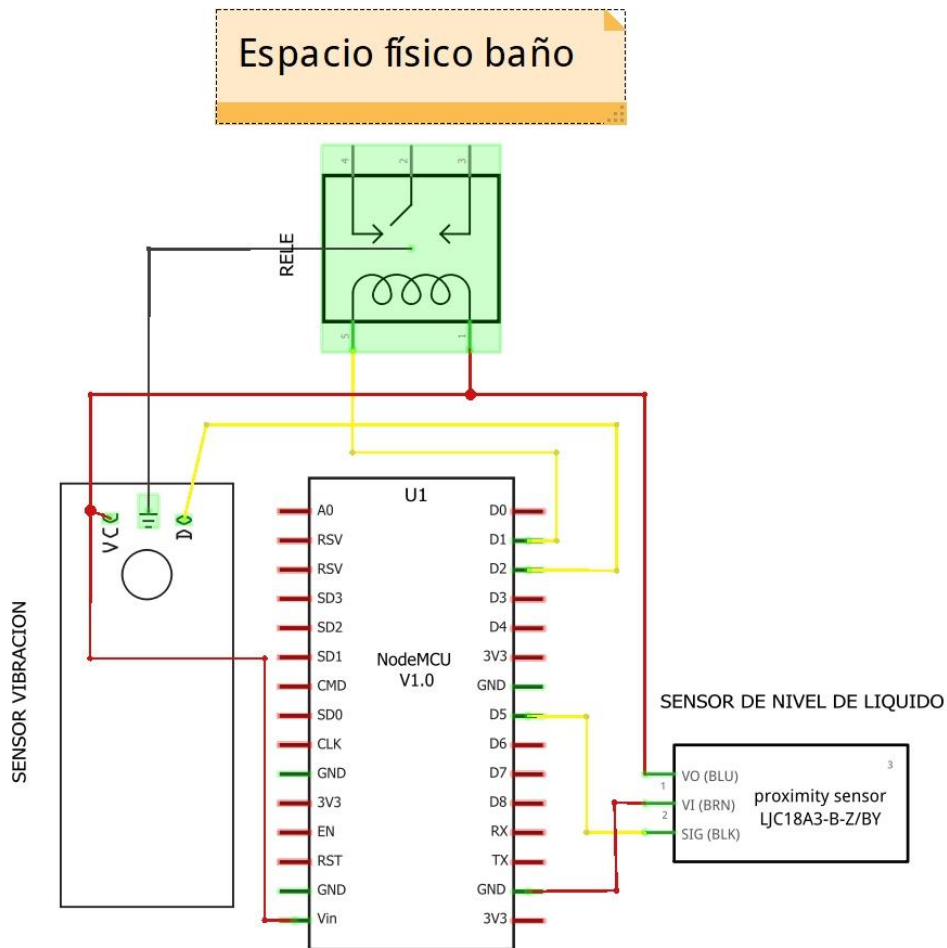
### 3.4. Esquema de conexión espacio físico comedor



fritzing



### 3.5. Esquema de conexión espacio físico baño

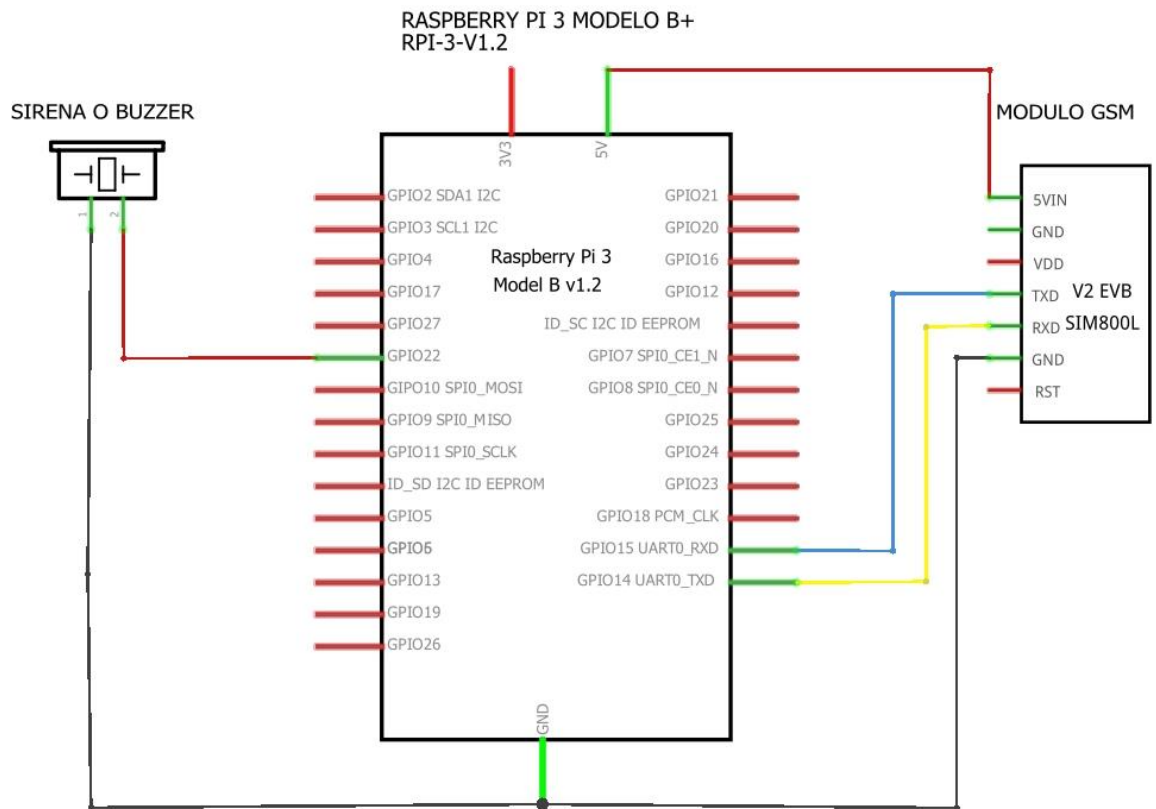


fritzing



### 3.6. Esquema de conexión Raspberry pi 3 modelo B+

## CENTRAL DE CONTROL RASPBERRY PI 3 MODELO B+



fritzing

### 4. Mantenimiento de los elementos del sistema domótico

El desarrollo de un plan de mantenimiento para un sistema domótico puede ser bastante útil para asegurar el correcto funcionamiento a largo plazo de los dispositivos y componentes involucrados. A continuación, se detalla una estructura básica para crear un plan de mantenimiento:

- Realizar un inventario completo de todos los dispositivos y componentes presentes en el sistema domótico, incluyendo su ubicación y funciones específicas.
- Establecer una frecuencia para llevar a cabo inspecciones periódicas del sistema domótico. Durante estas inspecciones, verifica visualmente el estado físico de los dispositivos, busca señales de desgaste o daños y realiza pruebas básicas de funcionamiento.

- Mantener actualizado el software de control y gestión del sistema domótico. Esto incluye los controladores de dispositivos, las aplicaciones móviles y cualquier otro software relacionado.

### **5. Mantenimiento preventivo**

Implementación de un programa de mantenimiento preventivo para los dispositivos específicos, de acuerdo con las recomendaciones del fabricante. Esto puede incluir la limpieza regular de sensores, verificación y ajuste de conexiones, reemplazo de baterías, entre otros. Con el fin de alargar la vida útil del sistema domótico.

### **6. Mantenimiento del sistema eléctrico**

El prototipo domótico está conformado por un sistema eléctrico bastante extenso por este motivo es importante tomar en cuenta las siguientes tareas de mantenimiento:

- Estado del aislamiento del sistema electrónico.
- Revisión de la conexión de los terminales.
- Verificación de los dispositivos de control y accionamiento.
- Reajuste del sistema eléctrico que comprende las partes móviles del sistema.
- Medición de consumo de corriente de todo el sistema.
- Limpieza de las placas electrónicas.
- Limpieza