

UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA
LICENCIATURA EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES

**Diseño y construcción
de un sistema domótico interoperable,
de bajo costo con integración IoT utilizando
el protocolo MQTT como lenguaje de comunicación**

TESIS

QUE PARA OPTAR POR EL TÍTULO DE
**LICENCIADOS EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS Y DE TELECOMUNICACIONES**

PRESENTAN:

**DIEGO JUÁREZ LUNA
MIRIAM MONSERRAT REYES SÁNCHEZ**

DIRECTOR: **DR. MARCO ANTONIO GONZÁLEZ SILVA**
CODIRECTOR: **DR. EDER CONTRERAS GALLEGOS**

Ciudad de México, mayo de 2024.

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Agradecimientos

Diego Juárez Luna:

“Si he visto más, es poniéndome sobre los hombros de Gigantes” (Isaac Newton)

*A mis padres amorosos y pacientes que jamás me abandonan, sin ellos nada de lo que he
logrado en mi vida sería posible.*

*A mi hermanito Roy, que sin hablarme siempre tiene lo que mi corazón necesita para encontrar calma
y bondad.*

A mis profesores que me acompañaron y aconsejaron durante toda mi vida académica.

A esas personas que jamás me han dado la espalda y siguen creyendo en mí.

*Al Dr. Marco Antonio Gonzáles Silva por darme su voto de confianza y apoyarme sin
conocerme, en este proceso tan importante.*

*Al Dr. Eder Contreras Gallegos por su tiempo, apoyo y compromiso en todas y cada una de sus clases,
por no desistir en esta etapa tan importante para mí.*

*A la Universidad Autónoma de la Ciudad de México, que me devolvió la oportunidad de
convertirme en el primer ingeniero en mi familia.*

Miriam Reyes Sánchez:

*Hoy culmino una etapa muy importante de mi vida y veo realizada una de las más grandes metas que
me propuse, el camino no fue fácil pero también viví las mejores experiencias de mi vida que me llevaron
a crecer como persona, es por eso que quiero extender mis más sincero y profundo agradecimiento a
las personas que compartieron conmigo este proceso, que siempre aportaron a mi vida inspiración,
apoyo y fortaleza.*

*A la vida por ser tan bondadosa conmigo y permitirme continuar en este camino con salud y
paz en el alma.*

A mis padres y hermanos, por el apoyo incondicional y amor que me dan.

A mi abuelita, por enseñarme a ser una mujer decidida y fuerte, que debe trabajar duro para lograr sus metas.

A mi esposo, que es mi respaldo, por siempre creer en mí, por el apoyo, amor y comprensión que me da, pues son los pilares fundamentales que me han sostenido en todo momento para continuar.

A mis hijos, por ser la fortaleza de mi vida.

A los profesores que, en mi trayectoria como estudiante, me guiaron y enseñaron.

Al Dr. Eder Contreras Gallegos y el Dr. Marco Antonio Gonzáles Silva por su tiempo, confianza y compromiso para poder culminar este trabajo de tesis.

A la Universidad Autónoma de la Ciudad de México, por ser la institución donde pude continuar con mi formación académica y lograr mi objetivo de tener una licenciatura.

Dedicatoria

Diego Juárez Luna:

Con todo mi amor, respeto y profundo agradecimiento para mi mejor amiga, hermana y confidente.

Porque sin tu apoyo, esta meta no hubiese sido posible, todos tus sacrificios valieron la pena.

Aún desde el cielo; nunca dejas de apoyarme. Mi única forma de honrar tu memoria es ser todo eso que un día prometimos ser.

Hasta que nos volvamos a encontrar mi querida “Lunita”

En memoria de mi hermana Alejandra Juárez Luna.

Miriam Reyes Sánchez:

Quiero dedicar esta tesis a mi esposo, ya que su apoyo fue fundamental en este proceso, a mis hijos que me tuvieron paciencia cuando tuve que quitarles tiempo porque tenía que cumplir con actividades escolares. No lo hubiera podido lograr sin su apoyo y comprensión, siempre me llenaron de fortaleza para seguir adelante. Este logro, no es solo mío, es de nuestra hermosa familia.

Resumen

En este trabajo se desarrolla un Sistema Domótico con integración del Internet de las Cosas (IoT), mediante el diseño de prototipos que emplean el protocolo MQTT como lenguaje de comunicación.

Este sistema está integrado por diversos sensores y actuadores que recopilan información e interactúan con el entorno del hogar para ser analizados y procesados mediante un panel o “Dashboard” de control, de forma local y remota a través de Internet.

El sistema se enfoca en brindar una alternativa accesible y de bajo costo al mundo de la automatización del hogar, aportando seguridad, confort y ahorro energético.

Mediante el uso de programación visual a través de flujos basada en JavaScript es posible crear aplicaciones para IoT que simplifican el problema de interoperabilidad entre protocolos y fabricantes.

Gracias al editor de flujos de NodeRed es posible integrar los prototipos y consolidarlos en un sistema de gestión y automatización del hogar, además de dotarlo de capacidades de red, donde el usuario final puede ejecutar de forma remota comandos que le permiten tener control de su hogar desde un navegador web en tiempo real.

El sistema se construyó con dispositivos de bajo costo, tales que garanticen su viabilidad económica y sea una opción accesible para el mercado domótico en el hogar de los habitantes de la Alcaldía Gustavo. A. Madero en la Ciudad de México.

Abstract

This work develops a Home Automation System with integration of the Internet of Things (IoT), through the design of prototypes that use the MQTT protocol as a communication language. This system is comprised of various sensors and actuators that gather information and interact with the home environment to be analyzed and processed through a control panel or "Dashboard," both locally and remotely via the Internet.

The system focuses on providing an accessible and low-cost alternative to home automation, contributing to security, comfort, and energy savings. Through the use of visual programming via JavaScript-based flows, it is possible to create IoT applications that simplify the problem of interoperability between protocols and manufacturers.

Thanks to the NodeRed flow editor, it is possible to integrate the prototypes and consolidate them into a home management and automation system, as well as providing network capabilities, where the end user can remotely execute commands that allow them to control their home from a web browser in real-time.

The system was built using low-cost devices, ensuring its economic viability and making it an accessible option for the home automation market in the Gustavo A. Madero borough in Mexico City.

Índice

Agradecimientos.....	2
Dedicatoria	4
Resumen	5
Abstract	6
Índice de Figuras.....	10
Índice de Tablas	12
Lista de Acrónimos.....	13
Glosario de Términos	15
Introducción.....	20
Exposición General	21
CAPITULO 1. ESTADO DEL ARTE.....	23
1.1 Objetivos	24
1.1.1 Objetivo General	24
1.1.2 Objetivos Específicos.....	24
1.2 Planteamiento del Problema	26
1.2.1 Problema Interoperabilidad.....	26
1.2.2 Problema de costo	27
1.2.3 Problema inseguridad	27
1.3 Justificación.....	29
1.4 Historia de la Domótica	31
1.4.1 Servicios que ofrece la Domótica.....	32
1.4.2 La domótica y el IoT.....	34
1.4.3 Interoperabilidad.....	35
CAPITULO 2. MARCO TEÓRICO Y MARCO METODOLÓGICO.....	36
2.1 MARCO TEÓRICO.....	37
2.1.1 Componentes de un Sistema Domótico	37
2.1.2 Medios de Transmisión de un Sistema Domótico.....	38
2.1.3 Arquitectura de un Sistema Domótico	39
2.1.4 Topologías físicas de un Sistema Domótico.....	40
2.1.5 Internet de las Cosas (IoT: Internet of Things)	40
2.1.6 Arquitecturas, Estándares y Protocolos para IoT	43
2.1.7 Comparación de Modelos de Referencia.....	48
2.1.8 Microcomputadoras y Microcontroladores	49

2.1.9 Raspberry Pi 4B	50
2.1.10 SoC ESP32	51
2.1.11 Protocolo MQTT (Message Queuing Telemetry Transport protocol).....	52
2.1.12 Servicios Web	56
2.1.13 NodeRed	57
2.2 MARCO METODOLÓGICO	59
2.2.1 Metodología para el diseño y construcción del sistema domótico.....	59
2.2.2 Fase 1: Identificación del problema	61
2.2.2.1 Paso 1: Análisis de la situación actual a nivel nacional e internacional	62
2.2.2.2 Paso 2: Investigación, desarrollos y aplicaciones de los sistemas domóticos	63
2.2.2.3 Paso 3: Análisis del enfoque social en la alcaldía G.A.M. de la Ciudad de México acerca de los sistemas domóticos y el IoT	63
2.2.3 Fase 2. Diseño del sistema domótico	65
2.2.3.1 Paso 1: Selección y descripción del hardware y software	66
2.2.3.2 Paso 2: Selección y descripción del servidor.....	67
2.2.3.3 Paso 3: Esquemas y descripción de los prototipos IoT	68
2.2.3.4 Paso 4: Mapeo y programación de los dispositivos.....	69
2.2.4 Fase 3. Construcción del sistema domótico	70
2.2.4.1 Paso 1: Diagrama de Procesos del Sistema	71
2.2.4.2 Paso 2: Servidor y Bróker MQTT	71
2.2.4.3 Paso 3: Ensamblado de los dispositivos.....	72
2.2.4.4 Paso 4: Configuración de la Red	72
2.2.4.5 Paso 5: Implantar los Dispositivos en el sistema.....	73
2.2.5 Fase 4. Evaluación y mejoramiento del sistema domótico	73
2.2.5.1 Paso 1: Evaluación de la Interoperabilidad.....	74
2.2.5.2 Paso 2: Diagnóstico de la eficiencia energética	74
2.2.5.3 Paso 3: Evaluación de Costos	74
2.2.5.4 Paso 4: Identificación de Áreas de Oportunidad.....	75
2.2.5.5 Paso 5: Mejoramiento	75
CAPITULO 3. IMPLEMENTACIÓN DE LA METODOLOGÍA	77
3.1 Fase 1: Identificación del problema	78
3.1.1 Análisis del enfoque social y económico en la alcaldía Gustavo A. Madero de la Ciudad de México, acerca de los sistemas domóticos y de IoT.....	78
3.1.2 Planeación de la encuesta	79
3.1.3 Calculo de la muestra	79

3.1.4 Diseño del Cuestionario	82
3.1.5. Distribución de la encuesta	83
3.1.6. Obtención y tratamiento de los datos.....	84
3.1.7. Análisis e interpretación de resultados	88
3.2 Fase 2. Diseño del sistema domótico	90
3.2.1 Paso 1: Selección y descripción del hardware y software	92
3.2.2 Paso 2: Selección y descripción del servidor.....	93
3.2.3 Paso 3: Esquemas y descripción de los prototipos IoT	95
3.3 Fase 3. Construcción del Sistema domótico.....	98
3.3.1 Paso 1: Diagrama de Procesos del Sistema	98
3.3.2 Paso 2: Servidor y Bróker MQTT	100
3.3.3 Paso 3: Ensamblado de los dispositivos.....	104
3.3.4 Paso 4: Configuración de la red.....	105
3.3.5 Paso 5: Implantar los Dispositivos en el sistema.....	108
3.4 Fase 4. Evaluación y mejoramiento del sistema domótico	130
3.4.1 Paso 1: Evaluación de la Interoperabilidad.....	130
3.4.2 Paso 2: Diagnóstico de la eficiencia energética	132
3.4.3 Paso 3: Evaluación de Costos	137
3.4.4 Paso 4: Paso 4: Identificación de Áreas de Oportunidad	141
3.4.5 Paso 5: Mejoramiento	145
Conclusiones.....	151
Bibliografía	153
ANEXO.....	158
DATASHEET'S.....	159

Índice de Figuras

Figura 1.1.Eventos más relevantes en el desarrollo de la domótica. (Elaboración propia, 2023).	32
Figura 1.2.Servicios que ofrece la domótica. (Elaboración propia, 2023).	33
Figura 2.1.Elementos de un sistema domótico. (Elaboración propia, 2023).	38
Figura 2.2.Arquitectura domótica centralizada. (Elaboración propia, 2023).	39
Figura 2.3.Evolución del IoT. (Elaboración propia, 2023).	42
Figura 2.4.Estructura general de la arquitectura del IoT. (Adaptado de [27]).	43
Figura 2.5.Arquitectura de 3 capas. (Elaboración propia, 2023).	44
Figura 2.6.Transmisiones y recepciones de datos pila de protocolos TCP/IP. (Fuente: [29]).	47
Figura 2.7.Funcionamiento de MQTT. (Fuente: [31]).	54
Figura 2.8.Ejemplo de Arquitectura MQTT. (Adaptada de: (Ine4C, 2023)).	54
Figura 2.9.Formato de un mensaje MQTT (Adaptada de: [32]).	55
Figura 2.10.Nodos predeterminados de NodeRed. (Fuente: [34]).	58
Figura 2.11.Entorno de uso de la plataforma NodeRed. (Fuente: [34]).	59
Figura 2.12.Fases de la metodología propuesta. (Elaboración propia, 2023).	60
Figura 2.13.Diagrama de bloques de la metodología propuesta (Elaboración propia, 2023).	61
Figura 2.14.Fase 1: Identificación del problema (Elaboración propia, 2023).	62
Figura 2.15.Mapa de procesos de la realización de la encuesta. (Elaboración propia, 2023).	63
Figura 2.16.Fase 2: Diseño del sistema domótico. (Elaboración propia, 2023).	66
Figura 2.17.Fase 3. Construcción del Sistemas Domótico. (Elaboración propia, 2023).	70
Figura 2.18.Fase 4. Evaluación y mejoramiento del Sistema. (Elaboración propia, 2023).	73
Figura 3.1.Demarcación geográfica de la Alcaldía Gustavo A. Madero. (Fuente: Google Maps, 2023).	79
Figura 3.2.Código del script para cálculo de muestras. (Elaboración propia, 2023).	81
Figura 3.3.Cálculo del número de muestras. (Elaboración propia,2023).	81
Figura 3.4.Margen de error: calculadora SurveyMonkey (Fuente: Extraído de [105]).	82
Figura 3.5.Presentación cuestionario Domótica & IoT. (Elaboración propia, 2023).	83
Figura 3.6.Ingresos promedio de la población encuestada. (Fuente: Encuesta Google Forms,2023).	84
Figura 3.7.Consideración del costo de dispositivos de sistemas domóticos. (Fuente: Encuesta Google Forms,2023).	85
Figura 3.8.Consideración para invertir en un sistema domótico el hogar. (Fuente: Encuesta Google Forms,2023).	86
Figura 3.9.Intercomunicación de dispositivos. (Fuente: Encuesta Google Forms,2023).	87
Figura 3.10.Diagrama de bloques del sistema propuesto. (Elaboración propia, 2023).	96
Figura 3.11.Distribución general de prototipos IoT. (Elaboración propia, 2023).	97
Figura 3.12.Diagrama de flujo de la comunicación con el Broker. (Elaboración propia,2023).	99
Figura 3.13.Distribución de pines RBP4B. (Fuente: raspberrypi.com).	101
Figura 3.14.Obtención de la herramienta Raspberry Pi Imager. (Fuente: raspberrypi.com).	101
Figura 3.15.Instalación de Mosquito. (Elaboración propia, 2023).	102
Figura 3.16.Ejecución de NodeRed activa. (Elaboración propia,2023).	104
Figura 3.17.Gestor de tarjetas. (Elaboración propia, 2023).	105
Figura 3.18.Port Forwarding NodeRed. (Elaboración propia, 2023).	107
Figura 3.19.Configuración DDNS. (Elaboración propia, 2023).	107
Figura 3.20.Flow para el control de Foco. (Elaboración propia, 2023).	110
Figura 3.21.Dashboard control de Luminaria. (Elaboración propia, 2023).	110
Figura 3.22.Modulo Relé. (Elaboración propia, 2023).	112
Figura 3.23.Diagrama Control de Luminaria. (Elaboración propia, 2023).	112
Figura 3.24.Código tomacorriente (Elaboración propia, 2023).	114
Figura 3.25.Control de cargas. (Elaboración propia, 2023).	115
Figura 3.26.Diagrama Tomacorriente. (Elaboración propia, 2023).	116
Figura 3.27.Adquisición de datos, sensor DHT22. (Elaboración propia, 2023).	117

<i>Figura 3.28.Flow para sensor de temperatura y humedad. (Elaboración propia).</i>	118
<i>Figura 3.29.Lectura de valores del sensor DHT22. (Elaboración propia, 2023).</i>	119
<i>Figura 3.30.Sensor Temperatura y Humedad. (Elaboración propia, 2023).</i>	120
<i>Figura 3.31.Programación Sensor MQ2. (Elaboración propia, 2023).</i>	121
<i>Figura 3.32.Publicación del tópico /gas. (Elaboración propia, 2023).</i>	121
<i>Figura 3.33.Diagrama detección de gas. (Elaboración propia, 2023).</i>	122
<i>Figura 3.34.Detección de flama. (Elaboración propia, 2023).</i>	123
<i>Figura 3.35.Flow Detección de flama. (Elaboración propia, 2023).</i>	124
<i>Figura 3.36.Correo Alarma de Incendio. (Elaboracion propia, 2023).</i>	125
<i>Figura 3.37.Sensor de Ventana. (Elaboración propia, 2023).</i>	126
<i>Figura 3.38.Diagrama Control de Acceso: Cerradura. (Elaboración propia, 2023).</i>	127
<i>Figura 3.39.Interfaz de Visualización de la ESP32-Cam. (Elaboración propia, 2023).</i>	129
<i>Figura 3.40.Prototipos del Sistema. (Elaboración propia, 2023).</i>	129
<i>Figura 3.41.Monitor de energía. (Fuente: Amazon,2023).</i>	133
<i>Figura 3.42.Graficas de corriente y potencia. (Elaboración propia, 2023).</i>	137
<i>Figura 3.43.Prueba seguridad MQTT. (Elaboración propia, 2023).</i>	142
<i>Figura 3.44.Análisis MQTT con MQTT Explorer. (Elaboración propia, 2023).</i>	143
<i>Figura 3.45.Análisis paquetes MQTT mediante Wireshark. (Elaboración propia, 2023).</i>	144
<i>Figura 3.46.Análisis MQTT, mediante Wireshark. (Elaboración propia, 2023).</i>	145
<i>Figura 3.47.Mapa de procesos del sistema. (Elaboración propia, 2023).</i>	146
<i>Figura 3.48.Construcción de maqueta. (Elaboración propia, 2023).</i>	149
<i>Figura 3.49.Distribución de prototipos. (Elaboración propia,2023).</i>	149
<i>Figura 3.50.Prototipo Sistema domótico. (Elaboración propia,2023).</i>	150
<i>Figura 3.51.Prototipo Sistema domótico. (Elaboración propia,2023).</i>	150

Índice de Tablas

<i>Tabla 1.1. Tipos de topologías en sistemas domóticos. (Adaptado de [85]).</i>	40
<i>Tabla 1.2. Modelo de referencia OSI. (Extraído de [88]).</i>	46
<i>Tabla 1.3. Pila de protocolos TCP/IP. (Extraído de [89]).</i>	48
<i>Tabla 1.4. Comparación entre modelos. (Elaboración propia, 2023).</i>	48
<i>Tabla 1.5. Microcomputadora vs Microcontrolador. (Extraído de [90]).</i>	49
<i>Tabla 1.6. Especificaciones Raspberry Pi4B. (Extraído de [91]).</i>	51
<i>Tabla 1.7. Especificaciones ESP32. (Extraído de [92]).</i>	51
<i>Tabla 1.8. Manejo de la Información. (Elaboración propia, 2023).</i>	62
<i>Tabla 1.9. Consideraciones de Software y Hardware. (Elaboración propia, 2023).</i>	67
<i>Tabla 1.10. Análisis FODA Broker MQTT. (Elaboración propia, 2023).</i>	67
<i>Tabla 1.11. Consumo promedio distintas plataformas. Fuente: CyberPuerta, 2023).</i>	94
<i>Tabla 1.12. Tarifas CFE mes de enero. Fuente: CFE, 2023.</i>	132
<i>Tabla 1.13. Medición de parámetros eléctricos del sistema. (Elaboración propia, 2023).</i>	133
<i>Tabla 1.14. Costos gastos comunes x mes, precios a fecha de 23/01/24. (Fuente: Internet, 2024).</i>	138
<i>Tabla 1.15. Costo Elementos del Sistema. (Fuente: Tiendas online especializadas, 2024).</i>	139
<i>Tabla 1.16. Comparación de Dispositivos IoT. (Fuente: Steren y Amazon, 2024).</i>	140

Lista de Acrónimos

A: Amper

ADC: Analog to Digital Converter - Convertidor Análogo a Digital

ADSL: Asymmetric Digital Subscriber Line - Línea de Abonado Digital Asimétrica

AMQL: Analytics metrics centric query language - Lenguaje centrado en métricas

AMQP: Advanced Message Queuing Protocol - Protocolo Avanzado de Cola de mensajes

API: Application Programming Interfaces - Interfaz de Programación de Aplicaciones

BLE: Bluetooth Low Energy - Bluetooth de baja energía

CoAP: Constrained Application Protocol - Protocolo de aplicaciones restringidas

CPU: Central Processing Unit - Unidad Central de Procesamiento

DAC: Diseño Asistido por Computadora

DAQ: Adquisición de Datos

DDNS: Sistema Dinámico de Nombres de Dominio

DDS: Servicio de Distribución de Datos

DDoS: Distributed Denial of Service – Ataque de denegación de servicio

DTLS: Seguridad de la Capa de Transporte de Datos

E/S: Entrada y Salida

FGJCDMX: Fiscalía General de Justicia de la Ciudad de México

FTDI: Future Technology Devices International

FTTX: Fiber To The X

GAM: Gustavo A. Madero

GB: Gigabyte

GHz: Gigahertz

GND: Ground - Tierra

GPIO: Entrada/Salida de Propósito General

GPS: Sistema de Posicionamiento Global

GUI: Interfaz Gráfica de Usuario

HDMI: Interfaz Multimedia de Alta Definición

HTTP: Protocolo de transferencia de hipertexto

HVAC: Calefacción, Ventilación y Aire Acondicionado

Hz: Hertz

IA: Inteligencia Artificial

ID: Identificación

IETF: Internet Engineering Task Force

IFT: Instituto Federal de Telecomunicaciones

INEGI: Instituto Nacional de Estadística y geografía

IoT: Internet of Things - Internet de la Cosas

IP: Protocolo de Internet

ISO: Organización Internacional para la Estandarización

ISP: Internet Services Provider - Proveedor de Servicios de Internet

ITU: International Telecommunications Union – Unión Internacional de Telecomunicaciones

KB: Kilobyte

LAN: Red de Área Local

LED: Diodo Emisor de Luz

M2M: Machine to Machina - Máquina a Máquina

MB: Megabyte

mCU: Microcontroller unit - Microcontrolador

MDF: Medium Density Fibreboard - Fibra de Densidad Media

MHz: Megahertz

ML: Machine Learning

MOM: Middleware Orientado a Mensajes

MQTT: Transporte de Telemetría de Cola de Mensajes

NFC: Comunicación de Campo Cercano

Nmap: Network Mapper

NOM: Norma Oficial Mexicana

OASIS: Organización para el Avance de Estándares de Información Estructurada

OCDE: Organización para la Cooperación y Desarrollo Económico

OCF: Fundación para la Conectividad Abierta

OPC-UA: Arquitectura Unificada de Comunicaciones de Plataforma Abierta

OSI: Interconexión de Sistemas Abiertos

PC: Personal Computer- Computadora Personal

QoS: Calidad de Servicio

RAM: Random Access Memory - Memoria de Acceso Aleatorio

REST: Transferencia de Estado de Representación

RF: Radiofrecuencia

RFID: Identificación de Radiofrecuencia

ROM: Read Only Memory - Memoria de Solo Lectura

SCADA: Control de Supervisión y Adquisición de Datos

SD: Digital Segura

SIM: Módulo de Identidad del Suscriptor

SO: Sistema Operativo

SOAP: Protocolo simple de acceso a objetos

SoC: Sistema en Chip

SPI: Interfaz de periféricos en serie

SRAM: Memoria de Acceso Estática

SSL: Secure Socket Layer - Capa de Sockets Seguros

TCP: Protocolo de Control de Transmisión

TIC: Tecnologías de la Información y Comunicaciones

TLS: Seguridad de la Capa de Transporte

TSMC: Taiwan Semiconductor Manufacturing Company Limited

UART: Receptor/Transmisor Asíncrono Universal

UDP: Protocolo de datagrama de usuario

ULP: Enchufe Lógico Universal

UMB: Bloque de Memoria Superior

USD: dólar estadounidense

V: Volt

VCA: Voltage Controlled Amplifier - Amplificador controlado por tensión

W: Watt

WSAN: Redes inalámbricas de sensores y actuadores

WSN: Sensor de redes inalámbricas

WWW: World Wide Web

XMPP: Protocolo extensible de mensajería y comunicación de presencia

Glosario de Términos

Actuadores: Dispositivos capaces de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía [1].

Algoritmo: Conjunto de instrucciones definidas y ordenadas que permite hallar la solución de un problema, o bien controlar determinados procesos [2].

Asistente Virtual: Agente de software que ayuda a usuarios de sistemas computacionales, automatizando y realizando tareas con la mínima interacción hombre-máquina, usando como medio de comunicación la voz [3].

Autenticación: Es el proceso que se utiliza para verificar que solo los usuarios, aplicaciones y/o servicios, con los permisos adecuados, puedan acceder a recursos y sistemas digitales, utilizando identificaciones como nombre de usuario y contraseña [4].

Automatización: Se refiere a la acción de utilizar tecnología para realizar tareas manuales repetitivas para reducir la intervención humana [5].

Broker MQTT: Es el sistema de back-end que coordina los mensajes entre los diferentes clientes. Las responsabilidades del agente incluyen recibir y filtrar mensajes, identificar a los clientes suscritos a cada mensaje y enviarles los mensajes [6].

C++: Es un lenguaje de programación compilado, multiparadigma, principalmente de tipo imperativo (conjunto de instrucciones que indican al computador cómo realizar una tarea) y orientado a objetos, incluyendo también programación genérica y funcional [7].

Cifrado de Datos: El cifrado de datos es una forma de traducir datos de texto sin formato (sin cifrar) a "ciphertext" (cifrado). Los usuarios pueden acceder a los datos cifrados con una clave de cifrado y a los datos descifrados con una clave de descifrado [8].

Circuito integrado: Es un chip que contiene una combinación de elementos que trabajan en

conjunto para simplificar funciones eléctricas [9].

Coohom: Empresa Software Service dedicada al diseño de planos 3D en el ámbito de la arquitectura y la industria inmobiliaria [10].

Consola de Comandos: También conocida como terminal, línea de comandos o interfaz de línea de comandos (CLI), es una interfaz de usuario basada en texto que se utiliza para ingresar comandos a un sistema operativo o software [11].

Control Inteligente: El control inteligente se refiere a un enfoque de la ingeniería de control que utiliza técnicas de inteligencia artificial (IA) para desarrollar sistemas capaces de adaptarse y optimizar su comportamiento en tiempo real [12].

Dashboard: Es la interfaz de usuario interactiva o panel de control de un sitio web, aplicación, juego, etc [13].

Diagrama de flujo: Es la representación gráfica y secuencial de todos los pasos, tareas y decisiones de un proceso o flujo de trabajo para lograr un objetivo común, estableciendo la relación entre ellas mediante símbolos [14].

Dispositivo IoT: Se refiere a un objeto físico que está conectado a internet, capaz de recopilar y transmitir datos sin intervención humana directa. Estos dispositivos aprovechan sensores integrados, software y otras tecnologías para intercambiar datos con otros dispositivos y sistemas a través de la red [15].

Dispositivo Inteligente: Es un aparato electrónico que está equipado con capacidades de procesamiento avanzadas, incluyendo la habilidad de conectarse a internet o a redes inalámbricas y la capacidad de interactuar con otros dispositivos o sistemas [16].

Domótica: También conocida como automatización del hogar, se refiere al conjunto de tecnologías aplicadas para el control y la automatización inteligente de la casa, lo que permite mejorar la calidad de vida de sus

habitantes, aumentar la eficiencia energética y proporcionar mayor seguridad [17].

Escalabilidad: Es la capacidad que tiene un sistema para ser ampliado ya sea agregando hardware adicional o actualizar el hardware existente sin modificar demasiado la aplicación [18].

ESP32: Es un microcontrolador de bajo costo y bajo consumo de energía con conectividad Wi-Fi y Bluetooth integrada, desarrollado por Espressif Systems. Se ha convertido en una opción popular para proyectos de electrónica e IoT [19].

Estándar: Son normas establecidas previamente aprobadas y redactadas con la finalidad de garantizar el acoplamiento entre elementos que se construyen de forma independiente [20].

Forwarding port: El forward de puertos, también conocido como mapeo de puertos, es una técnica utilizada en redes de computadoras para permitir que dispositivos externos accedan a servicios en una red privada a través de puertos específicos. Esto se logra redirigiendo las comunicaciones de red desde un puerto de dirección IP específico hacia otro puerto en otra dirección IP, generalmente a través de un dispositivo de red como un router o firewall [21].

Fritzing: Es una herramienta de software de diseño electrónico de código abierto que facilita la creación de esquemas, prototipos y placas de circuito impreso (PCB) para proyectos electrónicos [22].

Gmail: Es un servicio de correo electrónico gratuito proporcionado por Google [23].

Google Forms: Es una herramienta de encuestas en línea proporcionada por Google que permite a los usuarios crear formularios personalizados para recopilar información de manera fácil y eficiente [24].

Hardware: Es el conjunto de las partes físicas que integran una computadora [25].

Internet: Es una red global de computadoras interconectadas que utilizan el protocolo de comunicación estándar de Internet (TCP/IP) para compartir información y recursos a través

de una variedad de servicios, como la “World Wide Web”, el correo electrónico, transferencia de archivos, mensajería instantánea, entre otros. Fue desarrollada originalmente por la Agencia de Proyectos de Investigación Avanzada (ARPA) del Departamento de Defensa de los Estados Unidos en la década de 1960, y desde entonces ha evolucionado para convertirse en una infraestructura fundamental que conecta a personas, organizaciones y dispositivos en todo el mundo [26].

Inteligencia Artificial: También llamado IA, es el conjunto de aplicaciones que realizan tareas complejas para las que antes eran necesaria la intervención humana, como la comunicación en línea con los clientes o jugar al ajedrez. El término a menudo se usa indistintamente junto con los nombres de sus subcampos, el aprendizaje automático y el aprendizaje profundo [27].

Interfaz de usuario: Es la parte de un producto de software que el cliente realmente ve, puede incluir el diseño de pantallas de visualización o salida impresa, texto visualizado o impreso, comandos, ayuda en línea y mensajes. Una interfaz de usuario es también la parte de un producto de software que debe traducir o realizar cambios culturales para los usuarios de otros países o culturas [28].

Interfaz Gráfica de Usuario: Es una interfaz visual e interactiva que permite al usuario comunicarse ya sea con una PC, tabletas y otros dispositivos para poder controlarlos [29].

Interoperabilidad: Se refiere a la capacidad de diferentes sistemas, dispositivos o aplicaciones para intercambiar datos y trabajar juntos de manera efectiva, sin importar las diferencias en sus tecnologías subyacentes, arquitecturas o entornos operativos [30].

IoT: Se refiere a la interconexión de dispositivos físicos (objetos cotidianos) a través de internet, que les permite recopilar y compartir datos. Estos dispositivos pueden ser cualquier cosa, desde electrodomésticos, vehículos, dispositivos médicos, sensores industriales, hasta máquinas de fábrica, que están equipados con sensores, software y

conectividad de red para recopilar y intercambiar datos. El IoT permite la comunicación entre estos dispositivos y la toma de decisiones autónomas [31].

Java Script: Lenguaje de programación que permite el desarrollo de páginas web interactivas [32].

Lenguaje: En informática, el término "lenguaje" se refiere a un conjunto de reglas y símbolos utilizados para escribir programas de computadora. Estos lenguajes están diseñados para que los humanos puedan comunicar instrucciones a una computadora de manera clara y precisa [33].

Matlab: Es una plataforma de programación utilizada para el desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico que permite al usuario realizar el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y array's directamente [34].

Medio de Transmisión: Es el medio físico a través del cual se transmiten las señales de un dispositivo emisor a un receptor. Estos medios pueden ser cables de cobre, cables de fibra óptica, ondas de radio, microondas, o incluso el espacio libre en el caso de comunicaciones por satélite. Cada medio de transmisión tiene sus propias características y limitaciones en términos de ancho de banda, distancia de transmisión, atenuación de la señal, interferencia y seguridad [35].

Middleware: Es el software que funciona como una capa de traducción oculta que permite la comunicación y administración de datos entre aplicaciones distribuidas y el sistema operativo [36].

NOM: "Norma Oficial Mexicana", es un sistema regulador en México que establece estándares obligatorios en diversos campos para proteger la salud, seguridad, medio ambiente y los derechos del consumidor. Las NOM son emitidas por diversas dependencias gubernamentales mexicanas, como la Secretaría de Salud, la Secretaría de

Economía, la Secretaría de Medio Ambiente y Recursos Naturales, entre otras [37].

NodeRed: Es una herramienta de código abierto utilizada para la programación visual de flujos de datos. Proporciona un entorno de desarrollo basado en navegador web que permite a los usuarios conectar dispositivos de hardware, API web y servicios en la nube utilizando nodos predefinidos y arrastrar y soltar para crear flujos de trabajo automatizados. Node-RED es especialmente popular en el ámbito del Internet de las Cosas (IoT) y la automatización del hogar, aunque también se utiliza en una variedad de otros contextos, como el desarrollo web y la ciencia de datos [38].

Objeto Inteligente: En el contexto de las telecomunicaciones se refiere a dispositivos o sistemas que están equipados con capacidades de procesamiento y comunicación, así como sensores y actuadores, para recopilar, procesar y transmitir datos de manera autónoma. Estos objetos inteligentes suelen formar parte del Internet de las Cosas (IoT) [39].

Open Source: "Open Source" (código abierto) se refiere a un modelo de desarrollo de software en el que el código fuente del programa es accesible para cualquier persona y puede ser utilizado, modificado y distribuido libremente. Este enfoque fomenta la colaboración y la transparencia en el desarrollo de software, permitiendo que una comunidad global de desarrolladores contribuya a mejorar y perfeccionar el software [40].

Ontología Semántica: La ontología semántica en informática se refiere al estudio y la definición formal de conceptos y relaciones en un dominio específico de conocimiento, utilizando un lenguaje formal que permita la representación y el razonamiento sobre la información de manera precisa y semánticamente coherente. Las ontologías semánticas son utilizadas en áreas como la web semántica, la inteligencia artificial y la recuperación de información para facilitar la interoperabilidad entre sistemas, el descubrimiento de conocimiento y la integración de datos [41].

Paint: En informática, "Paint" se refiere a un software de edición de imágenes básico que permite a los usuarios crear, editar y modificar imágenes de forma sencilla [42].

Plataforma de desarrollo: Es el ambiente o entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Funciona como sistema plataforma o multiusuario, también sirven como base para que ciertos módulos de hardware o de software funcionen, siempre y cuando tengan compatibilidad [43].

Programación visual: Es un tipo de lenguaje de programación que permite ilustrar de manera más sencilla los procesos que se están programando, ya que se utilizan palabras comprensibles por el usuario y no se requiere de grandes conocimientos en lenguajes de programación [44].

Prototipo: Es un primer modelo que sirve como representación o simulación del producto final que permite verificar el diseño y confirmar que cuenta con las características específicas planteadas [45].

Protocolo: En informática, un protocolo se refiere a un conjunto de reglas y convenciones que especifican el formato, la secuencia y la interacción de mensajes entre dispositivos o sistemas de una red de computadoras. Estas reglas son necesarias para que los dispositivos puedan comunicarse entre sí de manera efectiva y comprender los datos transmitidos. Los protocolos pueden abordar diferentes aspectos de la comunicación, como la transmisión de datos, el enrutamiento de paquetes, el control de errores, la autenticación y la seguridad [46].

Raspberry Pi: Es una serie de computadoras de placa única de bajo costo y tamaño reducido desarrolladas por la Fundación Raspberry Pi en el Reino Unido. Estas computadoras están diseñadas para promover la enseñanza de la informática y la programación, así como para fomentar la creación de proyectos de hardware y software de bricolaje. Las Raspberry Pi son ampliamente utilizadas en una variedad de aplicaciones, incluyendo proyectos de IoT [47].

Relevadores: También llamados relés, son dispositivos electromagnéticos accionados eléctricamente que actúan como interruptores [48].

Router: Es un dispositivo de red que se utiliza para conectar redes informáticas y dirigir el tráfico de datos entre ellas. Los routers operan en la capa de red del modelo OSI y utilizan tablas de enrutamiento para determinar la mejor ruta para reenviar los paquetes de datos desde su origen hasta su destino [46].

Script: Es un conjunto de instrucciones de programación que se ejecutan dentro un entorno específico, son sencillos y funcionan como herramientas para extender o mejorar las funcionalidades de otros programas, además de depender de dichos programas para poder ejecutarse [49].

Sensor: Es un dispositivo de entrada que provee una salida manipulable de la variable física medida, que no solo transforma la energía, sino que además debe poseer el tipo de dominio requerido ya que es un intermedio entre la variable física y el sistema de medida [1].

Servidor Web: En el contexto de las telecomunicaciones, un servidor web es un dispositivo o software que proporciona servicios de acceso a páginas web y otros recursos a través de Internet o una red privada. Este servidor responde a las solicitudes de los clientes, como navegadores web, enviando los archivos solicitados, como páginas HTML, imágenes, scripts, etc. Un servidor web en telecomunicaciones puede ser utilizado para alojar sitios web públicos, aplicaciones web, servicios de correo electrónico, almacenamiento en la nube y otros servicios en línea [50].

Sistema: En ingeniería, un sistema se define como un conjunto de elementos interrelacionados que interactúan entre sí para lograr un objetivo específico. Estos elementos pueden ser componentes físicos, como máquinas, dispositivos electrónicos o estructuras, así como procesos, software, personas u otros recursos. La interacción entre

los elementos del sistema produce un comportamiento conjunto que es diferente de la suma de los comportamientos individuales de cada elemento por separado [51].

Sistema Domótico: Es un conjunto de dispositivos electrónicos interconectados que automatizan y controlan diversas funciones dentro del hogar, como la iluminación, la climatización, la seguridad, los electrodomésticos y otros dispositivos, con el objetivo de mejorar la comodidad, la seguridad, la eficiencia energética y la calidad de vida de los residentes [52].

Sistema Operativo: Es un software que actúa como intermediario entre el hardware de una computadora y el usuario, proporcionando una interfaz que permite la interacción con el sistema y la ejecución de programas. Además, gestiona los recursos del sistema, como la memoria, el procesador, los dispositivos de entrada y salida, y coordina la ejecución de procesos y la asignación de recursos para garantizar el funcionamiento eficiente del sistema [53].

Smart Home: Una Smart Home, u hogar inteligente, es un entorno residencial equipado con dispositivos electrónicos y sistemas de automatización que permiten la gestión remota y automática de diversas funciones del hogar, como la iluminación, la climatización, la seguridad, los electrodomésticos, el entretenimiento y otros dispositivos conectados. Estos dispositivos están interconectados a través de una red de comunicación, como Wi-Fi, Zigbee o Z-Wave, y pueden ser controlados y monitoreados a través de aplicaciones móviles, comandos de voz u otros dispositivos inteligentes [54].

Software: Es la parte lógica o intangible de una computadora [25].

Telemetría: Es el proceso de recopilación, medición y transmisión remota de datos sobre variables físicas, como temperatura, presión, nivel de líquidos, vibración, ubicación, entre otros, utilizando dispositivos de telemetría instalados en ubicaciones remotas. Estos datos se transmiten a través de redes de comunicación, como satélites, redes de telefonía móvil, o Internet, y se utilizan para monitorear y controlar dispositivos, procesos o sistemas en tiempo real [55].

Ubicuidad: Se refiere a la capacidad de estar presente en todas partes o en múltiples lugares al mismo tiempo. En el contexto de la tecnología y la informática, la ubicuidad se refiere a la disponibilidad y accesibilidad constante de los servicios digitales y la información, independientemente del lugar o del dispositivo utilizado para acceder a ellos [56].

Visio: Es un software de diagramación y visualización desarrollado por Microsoft. Microsoft Visio permite a los usuarios crear diagramas profesionales y técnicos, como diagramas de flujo, organigramas, diagramas de red, diagramas UML y muchos otros tipos de diagramas. Es ampliamente utilizado en el ámbito empresarial y técnico para visualizar procesos, estructuras y relaciones de datos de manera intuitiva y efectiva [57].

Wireshark: Es una herramienta de software de código abierto utilizada para el análisis de redes y la resolución de problemas. Permite a los usuarios capturar y analizar el tráfico de red en tiempo real y también permite la inspección de archivos de captura previamente guardados [58].

Introducción

La tecnología surge de la necesidad del hombre de mejorar su vida, es una herramienta capaz de ser adaptada a su entorno y ritmo de vida, brindando múltiples beneficios que le ha permitido convertirse en algo indispensable para la vida y desarrollo del ser humano.

El término IoT o Internet de las Cosas, fue utilizado por primera vez en 1999 y se le atribuye al informático Kevin Ashton. Este término se le asigna a cualquier “cosa” o dispositivo que pueda ser conectado a internet, por lo que se le asocia el concepto de “inteligente” [59].

Previo al surgimiento del IoT, se encuentra la Domótica, cuya invención se le atribuye al norteamericano Joel Spira en la década de los 70; la domótica es el conjunto de sistemas enfocados a la automatización de una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación; además de poder estar integrados por medio de redes de comunicación, cableadas o inalámbricas, cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar [60].

Hipótesis: *La integración de la Domótica y el Internet de las cosas (IoT) en los hogares conducirá a una mejora significativa en la seguridad, eficiencia energética y la comodidad de los usuarios. De tal forma que, al integrar la Domótica y el IoT, es posible crear sistemas de automatización y gestión del hogar para mejorar la vida de las personas y al mismo tiempo poder comunicar los dispositivos que integran dichos sistemas, dotándolos de inteligencia y con ello hacer más fácil la extracción de datos que ayuden a mejorar la calidad de vida de los usuarios.*

A pesar de que existen varias organizaciones internacionales que proponen el uso de protocolos, estándares y normas universales, no se ha logrado que todos los fabricantes se adhieran a ellos, por lo tanto, no existe hasta el momento ninguna normatividad internacional que regule la forma en que se comunican los dispositivos del IoT.

Al no existir interoperabilidad, las opciones disponibles en el mercado se vuelven bastante costosas e inaccesibles para ciertos sectores de la población, donde incluso este tipo de sistemas son considerados como “lujos innecesarios”.

Es por ello que la motivación principal de este proyecto es desarrollar un Sistema Domótico interoperable y de bajo costo para que sea accesible y se ajuste a los requisitos y/o necesidades de la población. El sistema se desarrolla tomando el protocolo de comunicación MQTT (“Message Queuing Telemetry Transport”), el cual es frecuentemente utilizado en el IoT, ya que permite la comunicación M2M (máquina a máquina) y es viable en la mayoría de sistemas integrados con recursos limitados, además de ser un protocolo abierto, de implementación ligera y funciona con un ancho de banda limitado [61].

Se aborda la problemática del proyecto desde un enfoque tecnológico, teniendo como objetivo el diseño y construcción de una solución domótica con base en conocimientos en sistemas de telecomunicaciones y electrónica.

Exposición General

Esta tesis se divide en 3 capítulos, con el fin de tener una metodología de desarrollo ordenada. En el primer capítulo se analizan los factores y circunstancias que enmarcan el proyecto para así conocer las condiciones del contexto para la investigación y desarrollo del mismo.

En el segundo capítulo se desarrolla el marco teórico y metodológico, en el cual se documenta la información técnica acerca de los elementos que conforman un sistema domótico, la evolución del IoT y su funcionamiento, además de plantear el diseño metodológico para el desarrollo del proyecto.

En el tercer capítulo se lleva a cabo la metodología planteada en el capítulo anterior. Se realiza el diseño, la construcción y la implementación del sistema con base en las características, los requisitos y la información teórica. Además, se evalúan el funcionamiento y los resultados para

determinar el cumplimiento de los objetivos del proyecto. Por último, se analizan aspectos importantes desde la perspectiva del usuario como la interoperabilidad, la accesibilidad y los costos del sistema.



CAPÍTULO 1

ESTADO DEL ARTE

Este capítulo contextualiza el mundo de la Domótica y el Internet de las Cosas, cuyos conceptos técnicos serán analizados más detalladamente en el Capítulo 2.

1.1 Objetivos

1.1.1 Objetivo General

Diseñar y construir un sistema domótico interoperable y de bajo costo con integración IoT utilizando el protocolo MQTT como lenguaje de comunicación, basado en las necesidades de un sector de la población de la Alcaldía Gustavo A. Madero (GAM).

1.1.2 Objetivos Específicos

- Analizar el nivel de conocimiento, necesidades y el poder adquisitivo de un sector de la población de la Alcaldía Gustavo A. Madero acerca de los sistemas de domótica e IoT por medio de una encuesta.
- Determinar las características, la funcionalidad y los costos de las tecnologías de comunicación IoT y dispositivos a utilizar para garantizar la interoperabilidad y bajo costo del sistema.
- Diseñar y construir prototipos de dispositivos IoT mediante la tarjeta de desarrollo ESP32 para monitorear y controlar la apertura y cierre de puertas, encendido y apagado de luminarias, censado de temperatura, gas y humedad, y control de tomacorrientes.
- Implementar una arquitectura de patrón publicación/suscripción usando el protocolo MQTT.
- Llevar a cabo la instalación del Broker Mosquitto en una microcomputadora Raspberry Pi para el control del sistema.
- Programar los dispositivos que integran el sistema utilizando el IDE de Arduino y la API de NodeRed.

- Diseñar la interfaz de control mediante el uso de una página web utilizando la plataforma de programación NodeRed.
- Establecer comunicación entre los dispositivos y el servidor IoT bajo un esquema cliente-servidor.
- Implementar el sistema domótico en una maqueta para pruebas de funcionamiento y detección de errores.
- Analizar y evaluar el desempeño, funcionalidad, interoperabilidad y coste del sistema domótico.

1.2 Planteamiento del Problema

El mundo se encuentra en una constante transformación tecnológica, debido a las grandes necesidades y demanda de servicios por parte de la población, por ello en el desarrollo de sistema inteligente es de gran importancia poder mantener comunicados los dispositivos que lo conforman y así recolectar información que permita predecir y moldear el comportamiento de la población y generar soluciones que ayuden a mejorar sus vidas.

A partir de esto, surge la domótica, que es un conjunto de sistemas capaces de automatizar una vivienda, aportando al usuario beneficios como el confort, seguridad, comunicación y gestión energética. Actualmente, su popularidad ha aumentado debido a que además de aportar beneficios para la vida cotidiana, ofrecen beneficios para el cuidado del medio ambiente [62].

1.2.1 Problema Interoperabilidad

Para alcanzar todo el potencial del IoT, donde tiene lugar la comunicación entre las personas y los dispositivos, no basta con que estén conectados a Internet, es necesario que estos dispositivos tengan interoperabilidad, es decir, que tengan la capacidad de comunicarse, compartir y utilizar servicios, datos e información entre sí [63]; sin embargo el contar con ésta presenta un gran reto debido al creciente número de dispositivos, aproximadamente 50 billones en el año 2020; de tecnologías, tanto antiguas como nuevas y el surgimiento de miles de fabricantes que desarrollan sus propios dispositivos con protocolos propietarios o cerrados.

En este sentido, se han planteado diversas soluciones como el uso de ontologías semánticas que consisten en proporcionar estructura en el manejo de la información que permita a las computadoras entender el significado del contenido que se procesa y aplicarlo en el diseño de dispositivos desde cero, el desarrollo y establecimiento de estándares, y una extensa lista de soluciones innovadoras propuestas por la industria, la academia y organismos de interés en los sistemas de IoT [64].

No obstante, el problema de interoperabilidad continua presente, aunado a los retos propios del IoT en almacenamiento, análisis, costos, seguridad, gestión de ancho de banda, gestión energética, etc., limitando su avance e implementación.

1.2.2 Problema de costo

En el mercado hay una amplia gama de tecnologías, dispositivos y protocolos para realizar la construcción e implementación de un sistema domótico. La finalidad es que los distintos dispositivos que integran el sistema tengan comunicación y reciban instrucciones que sean capaces de ejecutarse en cualquier plataforma.

Para que esto sea posible los dispositivos deben tener intercomunicación, y es aquí donde está el principal problema que enfrenta la domótica, ya que aún no se cuenta con estándares globales que lo hagan posible, y con la existencia de miles de proveedores y fabricantes aumenta más el problema. Los fabricantes desarrollan dispositivos con protocolos y estándares cerrados, que solo se comunican entre sí, entonces, cuando se quiere montar un sistema domótico, todos los dispositivos deben ser de la misma marca, negándole al consumidor la oportunidad de acceder a soluciones con mejores costos.

1.2.3 Problema inseguridad

La integración de tecnología puede contribuir a la reducción de delitos, con ello disminuir la inseguridad presente en los hogares mediante el despliegue de sistemas que permitan a los usuarios vigilar sus hogares y controlar el acceso a las mismas.

La percepción de inseguridad se define como “la perturbación angustiosa del ánimo que se deriva de la diferencia entre el riesgo percibido de ser víctima de un crimen y la victimización de hecho”, es decir, la percepción que una persona tiene de ser víctima de un delito, independientemente de la probabilidad de serlo [65].

Según cifras de la base de datos global de información sobre la calidad de vida “Numbeo”, México ocupa el lugar número 20 en ranking de Criminalidad en América; con un 54.08% de tasa de criminalidad y una tasa de seguridad del 45.92% [66].

A nivel nacional, en marzo de 2023, 62.1% de la población de 18 años y más, consideró inseguro vivir en su ciudad [67].

Los resultados de la ENSU (Encuesta Nacional de Seguridad pública Urbana) revelan que, del porcentaje de la población que mencionó haber visto o escuchado conductas delictivas o antisociales en los alrededores de su vivienda durante el cuarto trimestre de 2023, 60.0% se relacionó con consumo de alcohol en las calles; 50.2%, con robos o asaltos; 39.1%, con venta o consumo de drogas; 39.1%, con vandalismo en las viviendas o negocios; 38.0%, con disparos frecuentes con armas; 24.2%, con bandas violentas o pandillerismo.

Específicamente hablando, en la Alcaldía Gustavo A. Madero, CDMX el 57.0% considera que vivir actualmente en su ciudad es inseguro, donde 19.8% considera sentirse inseguro dentro de su hogar, según datos de INEGI [68].

Según datos de la Fiscalía General de Justicia de la Ciudad de México (FGJCDMX), la Alcaldía Gustavo A. Madero, es una de las dos alcaldías donde más robos a casa habitación con violencia se registran, se registraron 288 robos a casa habitación en Gustavo A. Madero, 265 sin violencia y 23 con violencia [69].

1.3 Justificación

Actualmente, el mundo está en una constante transformación tecnológica impulsada por las necesidades y demanda de servicios por parte de la población, tanto en el ámbito doméstico como en el ámbito laboral.

El desarrollo de sistemas inteligentes requiere entre otros aspectos, del rápido avance de las tecnologías, de las características y desempeño de los elementos que los componen, y de la capacidad de comunicación de los distintos dispositivos interconectados [70].

En este aspecto, el IoT tienen una importancia significativa, ya que no se limita a comunicar dispositivos entre sí, sino que también ofrece la posibilidad de recopilar información y analizarla.

La falta de interoperabilidad representa uno de los grandes retos del mercado IoT; ya que, al tener una gran variedad de dispositivos de diferentes fabricantes, el intercambio de información se ve limitado, lo cual pone en peligro a sus usuarios [71]. Justamente en países como México donde el desarrollo tecnológico presenta un rezago considerable es donde mayor problemática se tiene a la hora de explorar el mercado del IoT, ya que según la Organización para la Cooperación y Desarrollo Económico (OCDE); la penetración de dispositivos conectados en México solo es del 6.3%, y además datos del IFT revelan que solo el 56.4% de hogares en el país cuentan con conexión a internet [72]. Sumando esta problemática al hecho de que el mercado IoT está dominado por fabricantes con protocolos propietarios (cerrados), se observa que la domótica y el propio IoT es inaccesible para habitantes de bajos recursos, ya que además de enfrentarse a la falta de conectividad deben lidiar con la falta de interoperabilidad entre dispositivos IoT. Aun cuando la popularidad de ciertos dispositivos domésticos IoT aumenta, el no tener una solución que permita garantizar interoperabilidad entre dispositivos ha frenado que sean más los usuarios que opten por

adquirirlos. Es por esto que la interoperabilidad dentro del IoT surge como una necesidad primordial, esta brinda el valor económico total potencial que el IoT permite [73].

De tal forma que el presente proyecto de tesis, se propone poder desarrollar un sistema domótico para automatizar tareas de una vivienda, el cual garantice la interoperabilidad de todos los dispositivos IoT presentes y así poder brindar una alternativa económica al mercado de la domótica y el IoT en México. El sistema contará con la capacidad de automatizar una vivienda, aportando servicios de ahorro energético, seguridad, bienestar y comunicación, a través de una red alámbrica e inalámbrica que permita la gestión y control de los datos, información y dispositivos, dentro y fuera del hogar. Este sistema ayudará a los usuarios a monitorear y controlar de forma remota varias condiciones en el hogar: apertura y cierre de puertas, encendido y apagado de luminaria, sensado de temperatura, gas y humedad, además del control de tomacorrientes.

Para mantener una comunicación confiable en el sistema y controlar los dispositivos se emplea el módulo ESP32 y se utiliza el protocolo de comunicación de Transporte de Telemetría de Cola de Mensajes (MQTT) el cual es un protocolo abierto de bajo consumo, esta característica lo vuelve ideal para los propósitos deseados.

Para que los mensajes se publiquen o suscriban en el modelo MOM, se utiliza la aplicación Mosquito, la cual funciona como un Broker o servidor MQTT, el cual se encarga de dispersar los mensajes a los clientes (dispositivos) dentro del sistema. Dicho Broker es instalado en una mini computadora Raspberry Pi, con el objetivo de reducir el consumo energético, el costo de la inversión y el espacio físico para su instalación dentro del hogar, aspectos que se verían incrementados si se utiliza otro tipo de tecnología convencional.

Por lo que este enfoque de diseño e implementación integra, además de la interoperabilidad y reducción de costos; requisitos como la confiabilidad, la disponibilidad, el rendimiento en tiempo real, la escalabilidad y la seguridad.

1.4 Historia de la Domótica

El origen etimológico de la palabra domótica proviene de la unión de dos palabras: “Domus” del latín que significa casa y “Tica” del griego que significa automática, la domótica se refiere a la integración de tecnologías que se aplican en una vivienda mediante un conjunto de sistemas que son capaces de automatizarla [74].

Ubicar temporalmente el nacimiento de la Domótica es algo, complejo ya que es gracias al proceso evolutivo de la tecnología que surgen aportaciones para la automatización de los hogares, que permiten el desarrollo de la misma.

La creación e implementación de la domótica comenzó con pruebas piloto de los primeros dispositivos de automatización en edificios, lo cual, en los años 80 fue un éxito pues estaba en su apogeo la creación de rascacielos con oficinas comerciales, sus sistemas brindaban eficiencia y bajo consumo energético en los sistemas de control.

A partir de esto, el desarrollo de la domótica ha sido continuo, y más aún, favorecido con la creación de las TIC (Tecnologías de la Información y las Comunicaciones), ofreciendo la posibilidad de ser implementada en casas habitación, particularmente en productos del hogar como los electrodomésticos y otros dispositivos, surgiendo así el concepto de “Smart Home” u hogares inteligentes, que se refiere a una casa en la que se utilizan dispositivos electrónicos interconectados para automatizar y controlar diferentes funciones del hogar. Además, con la llegada del Internet, en los últimos años, su crecimiento ha sido exponencial [75].

En la Figura 1.1 se observa algunos de los eventos más importantes en el desarrollo de la domótica a través de los años.

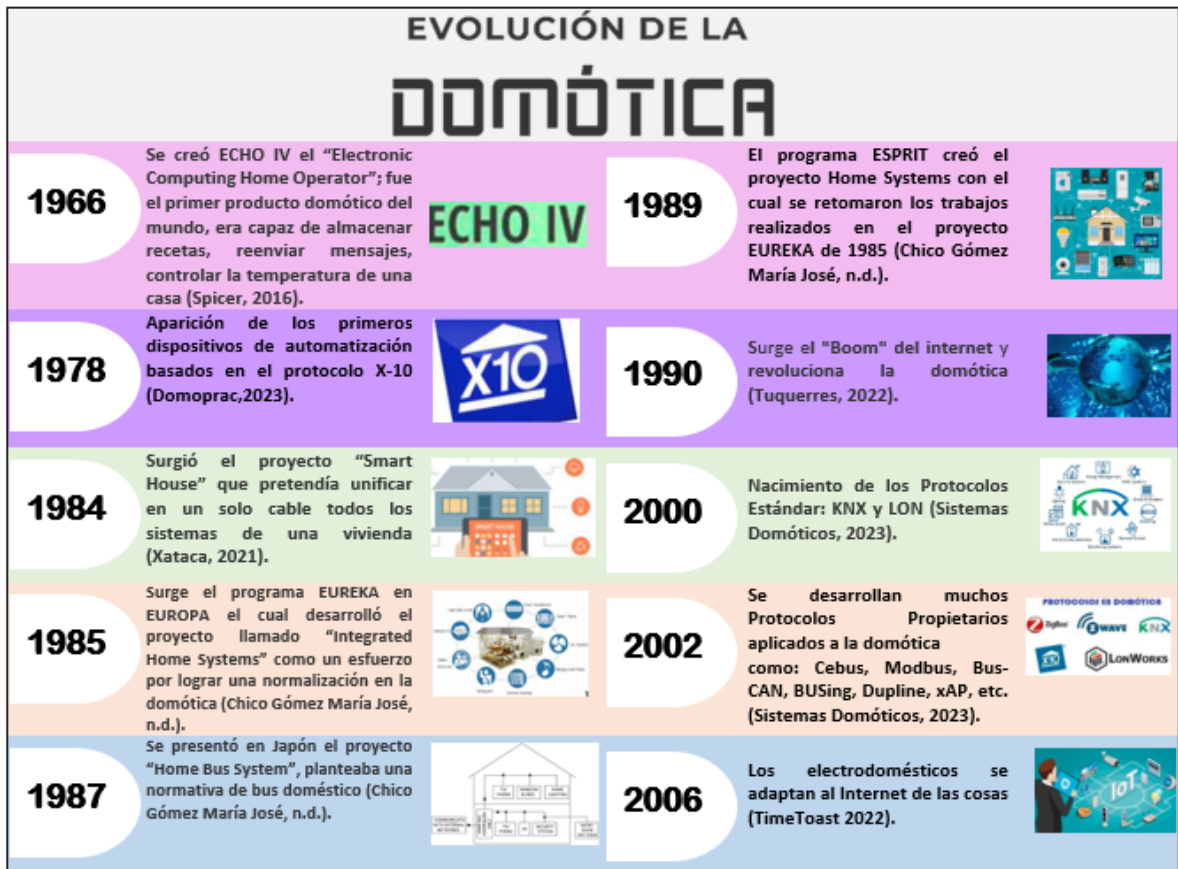


Figura 1.1. Eventos más relevantes en el desarrollo de la domótica. (Elaboración propia, 2023).

Inicialmente, los Sistemas Domóticos eran simplemente una red de sensores y actuadores, con arquitectura centralizada, donde la escalabilidad era bastante limitada, sumado a la falta de normatividad, la baja capacidad de procesamiento de sus elementos técnicos y dependencia de interconexiones cableadas limito bastante el desarrollo de la misma.

1.4.1 Servicios que ofrece la Domótica

Los servicios que ofrece la domótica se pueden agrupar en cuatro aspectos principales, como se muestra en la Figura 1.2 [62].

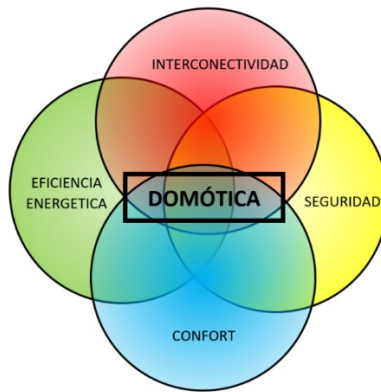


Figura 1.2. Servicios que ofrece la domótica. (Elaboración propia, 2023).

Tanto las ventajas y desventajas de la domótica se miden de acuerdo a las facultades que tienen los usuarios para utilizarla como una alternativa de comunicación y operación siendo una herramienta que puede mejorar la calidad de vida, pero sin llegar a la exclusión social [76].

Seguridad

El servicio de seguridad es el más importante, pues es la principal razón de los usuarios para adquirir estos sistemas debido a que se obtienen beneficios en cuanto a seguridad integral del hogar, mediante la implementación de alarma, sistemas de monitoreo, etc.

Confort

El confort es otra parte fundamental que ofrece la domótica, para la mayoría de los usuarios este servicio representa un lujo en sus vidas y para otros una gran ayuda en sus actividades cotidianas. Mejorar la calidad de vida de los usuarios mediante el uso de tecnología es parte fundamental de un sistema domótico.

Eficiencia Energética

La población ha crecido de manera considerable, y con ello el aumento en el consumo de energía, actualmente éste representa una gran parte del consumo de energía mundial, ya que es el lugar donde permanecen conectados a la red eléctrica diversos electrodomésticos.

La domótica permite al usuario hacer un seguimiento del uso de energía en el hogar estando en él o de manera remota, y así tomar acciones necesarias para reducir el consumo de energía. Dichas funciones pueden permitir la reducción de los costos por consumo de energía eléctrica, beneficiando la economía de los usuarios, a la vez que ayuda a mitigar los impactos negativos al medioambiente.

Interconectividad

Esta parte de la domótica se refiere a la capacidad de conexión entre dispositivos para intercambiar información con el resto del sistema y con el usuario, por lo general, esta comunicación se establece mediante un dispositivo de control permitiendo al usuario administrar y manejar todos los dispositivos conectados al sistema desde una sola ubicación.

1.4.2 La domótica y el IoT

La domótica busca automatizar las funciones de ciertos dispositivos electrónicos de uso cotidiano con la finalidad de que faciliten y mejoren la vida de los usuarios, mientras que el principal objetivo de IoT es la comunicación de los dispositivos electrónicos, sensores, actuadores y de casi cualquier dispositivo. Favoreciendo así la evolución de la domótica tradicional, pues ofrece dispositivos que incorporan agrupamiento y análisis de datos dotándolos de conectividad que les permite que se adapten mejor a las tareas domésticas [77].

El IoT dentro de la automatización de los hogares ha ido en aumento gracias a los avances de las TIC y la comunicación entre equipos de trabajo dedicados a la investigación y desarrollo de la automatización. En los últimos años, el IoT ha evolucionado gradualmente hasta convertirse en el grupo de investigación más importante para el desarrollo de las Smart Home, pues sus paradigmas influyen significativamente en su diseño e implementación.

Gracias a esto, el IoT se convertirá en el núcleo de la estructura de conocimiento de la casa inteligente [78], [79].

1.4.3 Interoperabilidad

La interoperabilidad se refiere a la capacidad que tienen distintos elementos que tienen sistemas operativos y marcas diferentes para funcionar conjuntamente y compartir información sin obstáculos técnicos o jurídicos.

El empleo de estándares abiertos es una condición previa para la neutralidad tecnológica ya que promueve la interoperabilidad y la integración de múltiples sistemas de información y conjuntos de datos. Gracias al establecimiento de normas comunes se reducen las diferencias entre especificaciones técnicas que disminuyen las barreras para el desarrollo tecnológico y la prestación de servicios competitivos para los usuarios estimulando la aceptación e innovación y al mismo tiempo, reduciendo costos [80].

Respecto a las tendencias para futuras soluciones se ha buscado desarrollar un estándar universal para el protocolo de comunicación de sistemas basados en IoT. A pesar de que diferentes organizaciones privadas, asociaciones, comités y alianzas en colaboración entre empresas han creado varias normas, ninguna ha sido adoptada para convertirse en un estándar para el hogar inteligente o la automatización del hogar [81].



CAPÍTULO 2

MARCO TEÓRICO Y MARCO METODOLÓGICO

En este capítulo se plasman en una primera parte el Marco Teórico, el cual contiene los conceptos e información teórica sobre los temas de interés que se analizan en esta tesis.

El Marco Metodológico describe las técnicas y pasos a seguir para el desarrollo de la tesis.

2.1 MARCO TEÓRICO

2.1.1 Componentes de un Sistema Domótico

La Domótica ha sido implementada desde hace varias décadas, permitiendo la automatización y el control inteligente de diferentes elementos en el hogar dirigidos a tareas cotidianas relacionadas con la seguridad, el bienestar y el confort brindando además eficiencia energética. Los elementos básicos e imprescindibles que conforman una instalación domótica son los siguientes:

- ***Central de gestión***

La central de gestión es el núcleo, distribuye la información a los elementos que conforman todo el sistema, actuando como el cerebro que lo controla.

- ***Sensores***

Son los elementos encargados de recabar información concreta del entorno o de la instalación, transforman los estímulos y las señales del entorno en señales que son procesadas por la central de gestión.

- ***Actuadores***

Son los mecanismos encargados de realizar la acción determinada por la central de gestión, en función de las señales que recibe del entorno mediante los sensores.

- ***Bus (Soportes de comunicación)***

Son los elementos que permiten el acceso a internet del sistema, ya sea de manera remota o local, así se dispone de un acceso remoto a la instalación para que el usuario tenga control de los elementos de interacción del sistema permitiéndole realizar acciones como activar o desactivar escenas, centralizar notificaciones, monitoreo y

gestión del sistema. Los elementos que conforman al sistema deben estar conectados entre ellos mediante un bus de datos que es el medio por el que viaja la información, puede ser por bus (cableado), por radiofrecuencia (inalámbrico) o mixto (bus y radiofrecuencia). Para que estos elementos conformen un sistema integral deben tener interoperabilidad y ser compatibles mediante un protocolo [82].

- **Interfaces (Aparatos terminales)**

Son todos los dispositivos físicos que procesan la información recibida de los puntos de terminación de la red para inter funcionar a través de ella, con el objetivo de enviar, procesar o recibir información y ejecutarla.

En la Figura 2.1 se muestran algunos ejemplos de cada uno de los dispositivos mencionados en una arquitectura centralizada para comprender la manera en que interactúan.

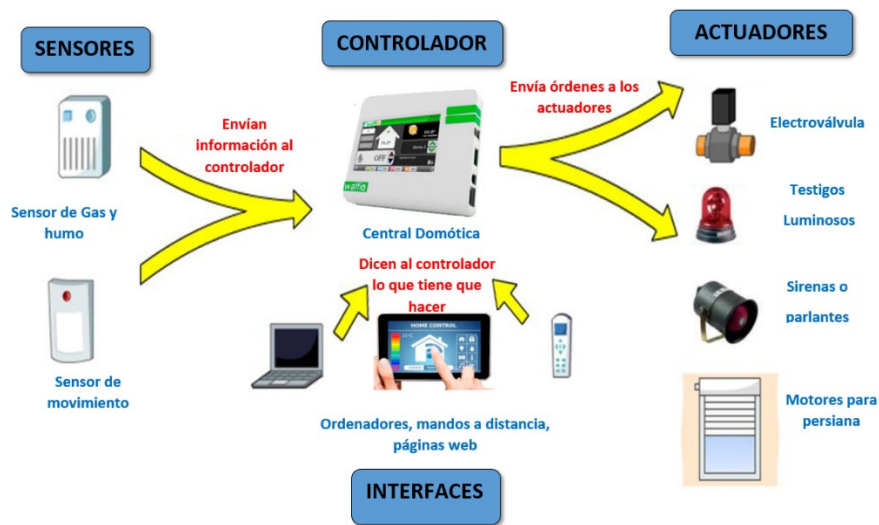


Figura 2.1. Elementos de un sistema domótico. (Elaboración propia, 2023).

2.1.2 Medios de Transmisión de un Sistema Domótico

Los principales medios de transmisión de la información o bus, entre los dispositivos de los sistemas domóticos son [83]:

- **Cableado Compartido:** Se utilizan cables compartidos con redes existentes para la transmisión de la información, por ejemplo, la red eléctrica, la red telefónica o la red de datos (internet), los mas usados pueden ser par de cobre, cable coaxial o fibra óptica.
- **Cableado Propio:** La transmisión por un cableado propio es el medio más común para los sistemas de domótica, principalmente son del tipo: par trenzado, coaxial o fibra óptica.
- **Inalámbrica:** Muchos sistemas de domótica utilizan soluciones de transmisión inalámbrica entre los distintos dispositivos, principalmente tecnologías de radiofrecuencia o infrarrojo.

2.1.3 Arquitectura de un Sistema Domótico

La arquitectura es la información de la distribución, ubicación y control de los elementos del sistema domótico y de cómo éstos se relacionan entre sí [84].

Arquitectura Centralizada

En esta arquitectura, la información de los sensores se envía directamente al controlador central, este a su vez interpreta los datos y los procesa, mediante la interface se interactúa con el usuario y finalmente modifica el estado de los actuadores. La desventaja radica en que si el nodo central falla, todo el sistema falla. En la Figura 2.2 se ilustra la distribución de la arquitectura centralizada.

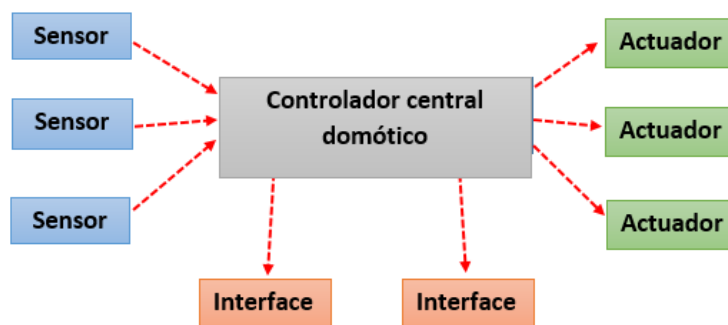
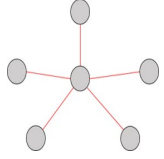
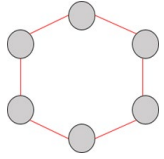
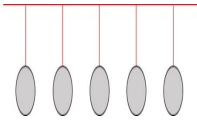
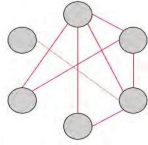


Figura 2.2. Arquitectura domótica centralizada. (Elaboración propia, 2023).

2.1.4 Topologías físicas de un Sistema Domótico

La topología es la forma en que está diseñada una red domótica y la forma en la que se interconectan los dispositivos del sistema, se pueden clasificar de acuerdo con su hardware o software [85], en la Tabla 1.1 se describen las clasificaciones de topología para Sistemas Domóticos.

Tabla 1.1. Tipos de topologías en sistemas domóticos. (Adaptado de [85]).

Tipo de topología	Descripción	Esbozo de la distribución
Topología en estrella	En este tipo de topología los dispositivos de entrada (sensores) y los de salida (actuadores) van cableados hasta el controlador central. Este tipo de conexión se utiliza generalmente en los sistemas centralizados donde existe un solo controlador sobre el que pasa toda la información.	
Topología en anillo	Aquí los nodos se conectan en un bucle cerrado, los datos se transmiten de nodo a nodo alrededor del bucle, siempre en la misma dirección, hasta que los datos llegan a su destino.	
Topología en bus	En este tipo de topología todos los elementos del sistema (sensores, actuadores y nodos) comparten la misma línea de comunicación. Cada elemento posee una dirección única permitiendo la comunicación.	
Topología en malla	Para esta topología existen diferentes nodos que permiten el envío de la información por diversos caminos. Cada nodo tiene la capacidad de recibir y enviar información, además de poder reenviar la información de los nodos vecinos. Esta conexión se utiliza generalmente en los sistemas domóticos distribuidos, donde todos los dispositivos se encuentran intercomunicados entre sí.	

2.1.5 Internet de las Cosas (IoT: Internet of Things)

El Internet de las cosas (IoT) es la agrupación o interconexión de distintos dispositivos a través de una red de internet y los usuarios pueden interactuar a través de dicha red con los

dispositivos, cualquier tipo de objeto puede formar parte del Internet de las Cosas el único requisito es que puedan ser conectados a Internet o una red privada para que se pueda producir lo que se conoce como una interacción M2M, es decir, de máquina a máquina (machine to machine) y sin intervención humana [86].

El IoT depende de un servidor del cual depende su velocidad y todas sus capacidades de funcionamiento.

El primer electrodoméstico conectado a Internet, fue creado en 1982 por el Departamento de Computer Science del Carnegie Mellon; se trataba de una máquina dispensadora de refrescos de Coca-Cola. Con esto los clientes podían consultar desde su computadora si había o no refrescos en la máquina [87].

La evolución del IoT prosiguió con la integración de sensores, con lo cual se buscó dotar de inteligencia a los objetos. En la Figura 2.3 se muestra la evolución del IoT con sus eventos más representativos.

EVOLUCIÓN DEL IOT

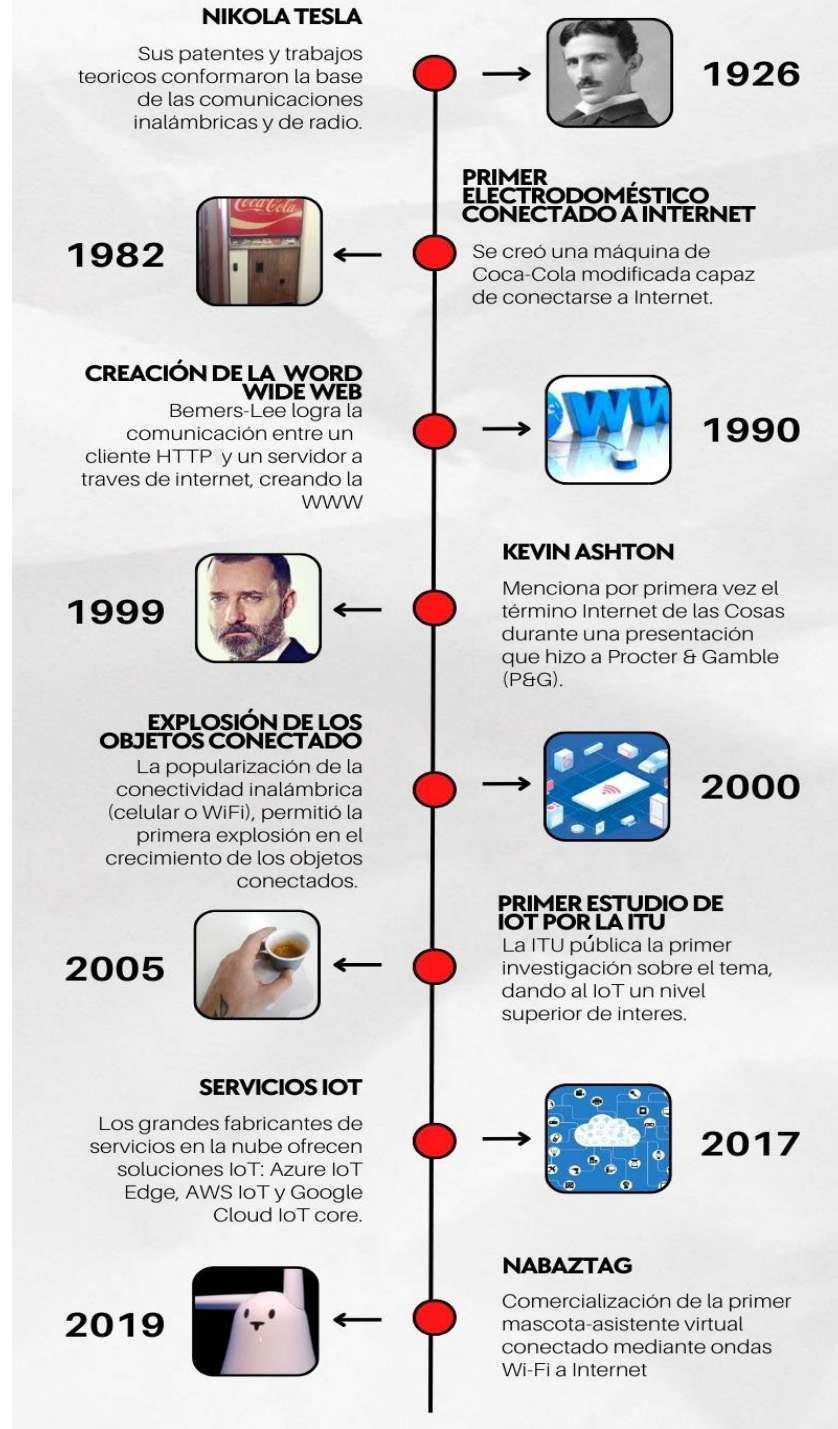


Figura 2.3. Evolución del IoT. (Elaboración propia, 2023).

2.1.6 Arquitecturas, Estándares y Protocolos para IoT

Los protocolos son las reglas que se deben establecer para poder lograr un proceso de comunicación efectiva sin que existan conflictos o pérdidas de información en dicho proceso. Aunque existen recomendaciones como la “UIT-T Y.2060” por parte de la ITU (Unión Internacional de Telecomunicaciones) la cual es idéntica a la Norma Mexicana “NMX-I-321-NYCE-2020 TELECOMUNICACIONES” por parte de la Secretaria de Salud en México (ambas solo brindan la descripción en cuanto a términos y modelos de referencia para el Internet de las Cosas de forma general), aun no existen normativas o estándares que establezcan regulaciones en cuanto a protocolos para IoT, son los mismos fabricantes los que en conjunto proponen diferentes formas de organizar y caracterizar el proceso de comunicación.

Dentro de estas propuestas existe la iniciativa de separar en capas al IoT, adoptando el modelo de referencia TCP/IP como una guía donde referenciar cada elemento que integra la comunicación entre dispositivos inteligentes. En la Figura 2.4 se muestra una estructura generalizada de arquitectura para IoT.

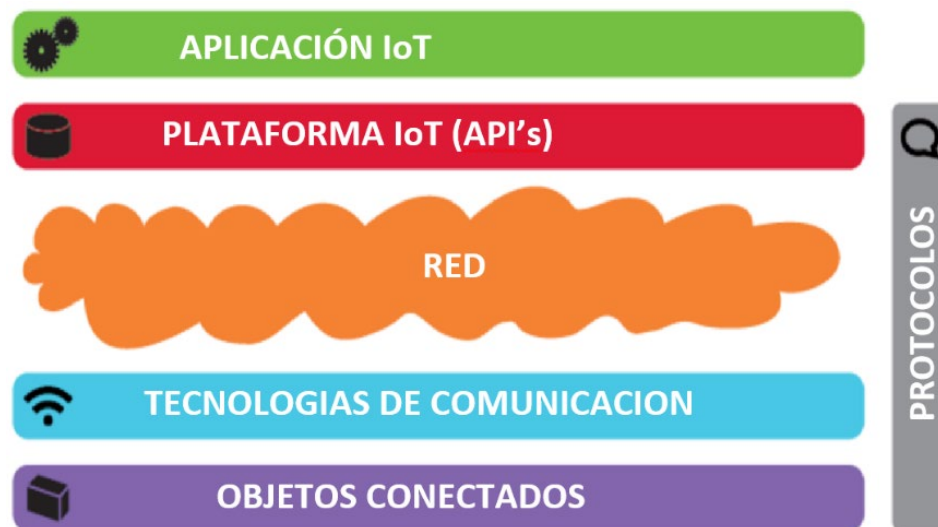


Figura 2.4. Estructura general de la arquitectura del IoT. (Adaptado de [27]).

Gracias a los modelos de referencia, se tiene una mejor comprensión de procesos de comunicación y la arquitectura que utiliza un sistema para poder interconectarse con otros sistemas, por lo general un modelo de referencia está organizado en capas, donde cada una se encarga de sus propias funciones y se comunica directamente con sus capas inmediatas superiores. Las capas inferiores están directamente relacionadas con el hardware, es decir; con los elementos físicos mediante los cuales se reciben y envían señales para su procesamiento. Por otro lado, las capas superiores son las encargadas de traducir estas señales para ser representadas y poder interactuar con el usuario final.

Arquitectura de 3 capas

La arquitectura de 3 capas se considera la más sencilla y las más utilizada por su simpleza de operación, está basada en una topología de estrella, donde se tiene un nodo central que organiza y coordina los elementos de la red IoT. En la Figura 2.5 se representa esta arquitectura.

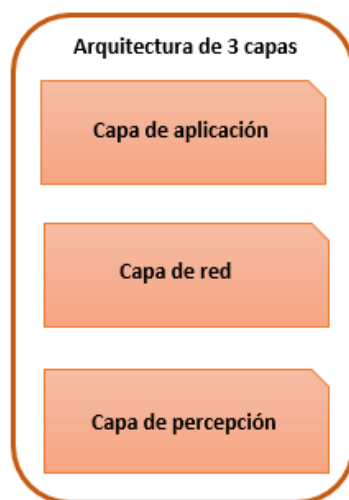


Figura 2.5.Arquitectura de 3 capas. (Elaboración propia, 2023).

Capa de Percepción

La capa inferior se denomina “capa de percepción”, en ella los “objetos inteligentes” (objeto que tiene la capacidad de conectarse a internet para intercambiar información propia de su funcionamiento) o cosas interactúan con el mundo físico y transmiten información valiosa del entorno, los elementos que componen esta capa son: sensores, actuadores, cámaras, dispositivos de radiofrecuencia, entre otros. La capa de percepción, recoge estas señales y las acondiciona para poder ser enviadas a la siguiente capa.

Capa de Red

Es la capa intermedia, su función es transmitir y procesar la información proveniente de la capa inferior. Esta capa también se encarga de interconectar con más dispositivos IoT presentes en otras redes, es decir poder entablar comunicación con servidores dentro y fuera de Internet. Este proceso se realiza mediante el uso de los protocolos TCP/IP los cuales dictan la forma en que los datos son enviados y recibidos.

Capa de Aplicación

Esta capa es la responsable de facilitar los servicios y aplicaciones al usuario final. Esta capa es la encargada de poder interactuar con el usuario y proveerlo de los mecanismos necesarios para la interpretación y manipulación de los dispositivos IoT.

La existencia de arquitecturas con más capas supone mayor complejidad y diversificación de procesos en el intercambio de información dentro del mundo del IoT, sin embargo, al no existir regulaciones, estandarizar IoT aún es una tarea difícil de lograr entre fabricantes.

Modelo OSI

El Modelo de Interconexión de Sistemas abiertos (Open Systems Interconnection, OSI), es el fundamento universal propuesto por la Organización Internacional para la

Normalización (ISO) en la década de los 80's, dicho modelo define las pautas para el establecimiento de comunicación entre sistemas. En la Tabla 1.2 se da una breve descripción de cada capa del modelo OSI [88].

Tabla 1.2. Modelo de referencia OSI. (Extraído de [88]).

NIVEL	Nombre	DESCRIPCION
7	Aplicación	Se compone de servicios y aplicaciones de comunicación que usa el usuario final. Algunos ejemplos de aplicaciones son los programas de hojas de cálculo, de procesamiento de texto y los de las terminales bancarias.
6	Presentación	La capa de presentación garantiza que la información que envía la capa de aplicación de un sistema pueda ser leída por la capa de aplicación de otro. De ser necesario, la capa de presentación traduce entre varios formatos de datos utilizando un formato común.
5	Sesión	La capa de sesión establece, administra y finaliza las sesiones entre dos hosts que se están comunicando. Además, proporciona sus servicios a la capa de presentación. También sincroniza el diálogo entre las capas de presentación de los dos hosts y administra su intercambio de datos.
4	Transporte	Segmenta los datos originados en el host emisor y los reensambla en una corriente de datos dentro del sistema del host receptor. El límite entre la capa de transporte y la capa de sesión puede imaginarse como el límite entre los protocolos de aplicación y los protocolos de flujo de datos.
3	Red	Proporciona conectividad y selección de ruta entre dos sistemas de hosts que pueden estar ubicados en redes geográficamente distintas.
2	Enlace de Datos	Proporciona tránsito de datos confiable a través de un enlace físico. Al hacerlo, la capa de enlace de datos se ocupa del direccionamiento físico, la topología de red, el acceso a la red, la notificación de errores, entrega ordenada de tramas y control de flujo.
1	Física	La capa física define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales. Las características tales como niveles de voltaje, temporización de cambios de voltaje, velocidad de datos físicos, distancias de transmisiones máximas, conectores físicos y otros atributos similares son definidos por las especificaciones de la capa física.

Actualmente el Modelo OSI sirve únicamente como referencia en el proceso de comunicación entre sistemas, sin embargo, es la pila de protocolos TCP/IP quien establece las reglas de comunicación entre sistemas.

Protocolo TCP/IP

La pila de protocolos TCP (Transmission Control Protocol) /IP (Internet Protocol) está compuesta por un conjunto de protocolos abiertos que soportan una gran cantidad de sistemas, ya que no depende ni del hardware ni del software que se estén utilizando. TCP/IP funciona sobre cualquier medio de transmisión.

Gracias a TCP/IP se define como es el proceso de intercambio de información desde el remitente hasta el destinatario. Durante este recorrido, cada capa añade o quita información de control a los datos para garantizar su correcto envío.

Los datos que se envían por internet se mueven por toda la pila del protocolo TCP/IP desde la capa superior hasta la capa más baja, cuando los datos son recibidos, el proceso se invierte. En la Figura 2.6 se ilustra este proceso.

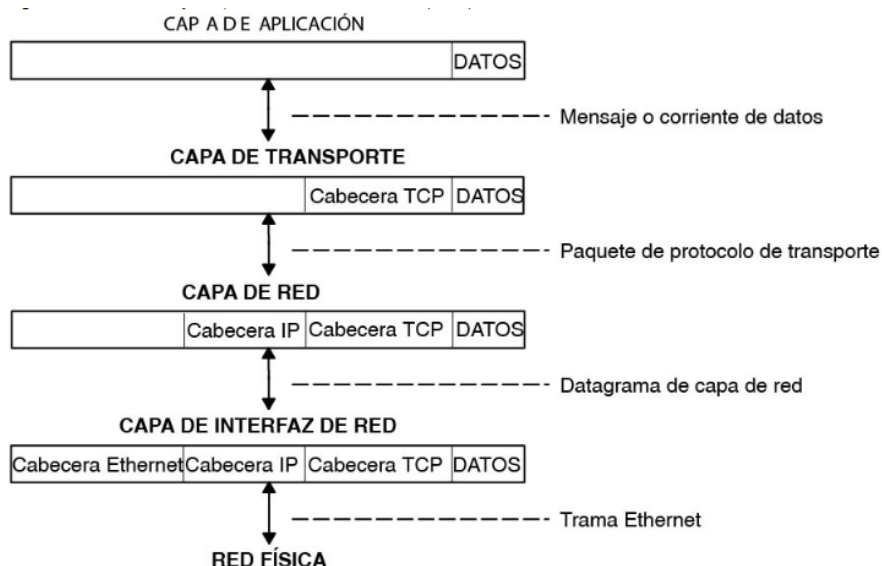


Figura 2.6. Transmisiones y recepciones de datos pila de protocolos TCP/IP. (Fuente: [29]).

En la Tabla 1.3 se muestra la pila de protocolo TCP/IP y se describe brevemente su función por capa [89].

Tabla 1.3. Pila de protocolos TCP/IP. (Extraído de [89]).

NIVEL	Nombre	DESCRIPCION
4	Aplicación	Representa y codifica los datos para el usuario.
3	Transporte	Permite la comunicación entre dispositivos ubicados en redes diferentes.
2	Internet	Determina la ruta más optima a través de la red.
1	Acceso a la Red	Controla los dispositivos físicos (hardware) y los medios que constituyen la red.

2.1.7 Comparación de Modelos de Referencia

Utilizar protocolos nos permite garantizar la comunicación de forma confiable entre los diversos dispositivos IoT.

Los protocolos se van a representar por medio de una arquitectura de capas, donde cada una de ellas brinda una función que es condicionada por un conjunto de reglas para el intercambio de datos entre los dispositivos que se conectan a una red.

El desarrollo de internet ha sido posible gracias a la utilización de la pila de protocolos TCP/IP, se puede decir que es la piedra angular en la comunicación de Internet y es utilizado en muchas áreas, una de ellas el IoT. En el caso de redes de baja potencia se utiliza el protocolo MQTT el cual es una adaptación del protocolo TCP/IP, con lo cual se solventan grandes problemas como la interoperabilidad y facilitar la transmisión eficiente de datos en las redes IoT. En la Tabla 1.4 tenemos la comparación por capas entre los modelos antes mencionados.

Tabla 1.4. Comparación entre modelos. (Elaboración propia, 2023).

CAPA	MODELO OSI	MODELO TCP/IP	MODELO IoT
7	APLICACIÓN	APLICACIÓN	APLICACIÓN
6	PRESENTACIÓN		
5	SESIÓN		

4	TRANSPORTE	TRANSPORTE	RED
3	RED	INTERNET	
2	ENLACE DE DATOS	ACCESO A LA RED	PERCEPCIÓN
1	FÍSICA		

2.1.8 Microcomputadoras y Microcontroladores

Una microcomputadora es una unidad de procesamiento que está contenida dentro de una unidad de procesamiento central (CPU), es un encapsulado que realiza diversas funciones aritméticas y lógicas. Incluye un microprocesador como unidad central, memoria y controladores de entrada y salida para poder ser conectada con periféricos externos.

Los microcontroladores son circuitos integrados que pueden ser programados para ejecutar acciones o instrucciones grabadas en su memoria de acuerdo a las necesidades del usuario, está constituido por varios bloques que tienen asignada una tarea específica, pero sus tres principales unidades funcionales son la unidad central de procesamiento, memoria y periféricos de entrada/salida. Al ser chips o dispositivos de tamaño pequeño son de gran utilidad para circuitos que requieren de un espacio físico reducido, son económicos y brindan fiabilidad de funcionamiento y disminución en el consumo. En la tabla 1.5 se muestran las características más relevantes de estos dispositivos.

Tabla 1.5. Microcomputadora vs Microcontrolador. (Extraído de [90]).

	Microcomputadora	Microcontrolador
Memoria	No cuentan con módulos de memoria interna para almacenar los datos de las aplicaciones. Es necesario utilizar almacenamientos de memoria externos, como la ROM y la RAM, con un bus externo.	Cuentan con memorias ROM y RAM internas integradas internamente, utiliza un bus interno para interactuar con los módulos de memoria integrados.

Periféricos	No cuenta con periféricos integrados internos, Es necesario conectar los periféricos de manera externa para ampliar los casos de uso del microprocesador más allá del procesamiento matemático y lógico.	Se conectan a los periféricos del chip mediante un bus de control interno. Esto permite que el microcontrolador controle los dispositivos electrónicos con un mínimo de piezas adicionales o incluso sin ellas.
Capacidad computacional	Son potentes chips de computadora capaces de realizar tareas matemáticas y computacionales complejas.	Tienen una potencia de procesamiento comparativamente menor a las microcomputadoras ya que son limitados a la lógica de la aplicación específica.
Consumo de energía	Alto consumo de energía debido a la gran cantidad de componentes adicionales., requieren una fuente de alimentación externa.	Consumen una cantidad mínima de energía ya que, diseñados para funcionar de manera eficiente con una potencia mínima, la mayoría cuentan con características de ahorro de energía.
Sistema operativo	Requieren la instalación de un sistema operativo que proporcione las funcionalidades necesarias.	No requieren un sistema operativo para funcionar, aunque existen sistemas operativos específicos que permiten a los microcontroladores de gama media y alta a funcionar de manera más eficiente.
Conectividad	Gestiona la transferencia de datos a alta velocidad. Soporta USB 3.0 y Gigabit Ethernet.	Soporta comunicaciones de baja a moderada velocidad. Interfaz periférica serial (SPI) e I ² C. Receptor-transmisor asíncrono universal (UART).
Costo	Es un circuito integrado que consta de la CPU, la unidad aritmética lógica (ALU) y los registros, por ello, su costo de fabricación se reduce. Pero al tener un sistema basado en microcomputadoras se hace más costoso debido a los componentes adicionales que se requieren.	Tiene una arquitectura interna más compleja, lo que lo hace más caro que una microcomputadora. Los sistemas basados en microcontroladores son más económicos ya que requiere menos componentes adicionales ya que un solo circuito integrado proporciona múltiples funcionalidades.
Casos de uso	Computación genérica o sistemas que requieren una capacidad computacional sólida.	Sistemas compactos, dispositivos alimentados por baterías o de procesamiento lógico.

2.1.9 Raspberry Pi 4B

Raspberry PI es una microcomputadora de placa de base reducida de 85mm x54 mm, y de bajo costo, fue desarrollada en Reino Unido por la Fundación Raspberry PI. Funciona con el sistema operativo Linux que la hace apropiada para cualquier persona que quiera aprender a programar lenguajes como Scratch y Python; es capaz de ejecutar sistemas

complejos, ofreciendo un amplio abanico de posibilidades. Tienen un consumo por debajo de los 7W, lo cual la hace ideal para desarrollar aplicaciones de IoT. En la Tabla 1.6 se muestran sus especificaciones.


Tabla 1.6. Especificaciones Raspberry Pi4B. (Extraído de [91]).

Modelo	CPU	Memoria RAM	Conectividad	Puertos	Precio aproximado
RASPBERRY PI 4B	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	2/4/8GB	802.11ac / Bluetooth 5.0 Ethernet	2x USB 3.0, 2x USB 2.0, 1x Gigabit Ethernet, 2x micro-HDMI	35/55/75 USD

2.1.10 SoC ESP32

El ESP32 pertenece a la familia de chips SoC (System On Chip) debido a que es una mezcla entre microcontrolador y microcomputadora ya que su construcción incluye a ambos, fue diseñado por Espressif Systems y fabricado por la empresa multinacional Taiwanesa TSMC (Taiwan Semiconductor Manufacturing Company Limited), es de bajo costo, bajo consumo de energía y además es capaz de ejecutar sus propias aplicaciones de tiempo real. Es utilizado como puente para el acceso a la red o a las soluciones IoT y domótica para microcontroladores que no cuentan con la capacidad para conectarse, gracias a que integra las tecnologías Wi-Fi y modo dual con Bluetooth. En la tabla 1.7 se listan sus especificaciones.

Tabla 1.7. Especificaciones ESP32. (Extraído de [92]).

ESP32		
	Procesador principal	Tensilica Xtensa LX6 de 32 bits.
	Wi-Fi	802.11 b / g / n / e / i (802.11n @ 2.4 GHz hasta 150 Mbit / s).

	Bluetooth	v4.2 BR / EDR y Bluetooth Low Energy (BLE).
	Frecuencia de Clock	Programable, hasta 240MHz.
	Rendimiento	Hasta 600DMIPS.
	ROM	448KB, para arranque y funciones básicas.
	SRAM	520KiB, para datos e instrucciones

2.1.11 Protocolo MQTT (Message Queuing Telemetry Transport protocol).

El protocolo de Transporte de Telemetría de Cola de Mensajes está enfocado en la transmisión de datos a nivel de byte, es uno de los protocolos más populares para la implementación de IoT debido a su bajo consumo de recursos. Fue creado en 1999 por IBM y liberado en 2010.

Para el año 2013, fue presentado formalmente por la Fundación OASIS y finalmente se estandarizó en 2016 ante la ISO.

El protocolo MQTT proporciona un método extremadamente ligero para enviar mensajes mediante un modelo de publicación /suscripción. Con una topología en estrella y tiene capacidad de conectar hasta 10,000 clientes [93].

Arquitectura MQTT

Está compuesta por los siguientes elementos:

- **Broker:** Es el nodo central, se encarga de recibir y distribuir la información a los clientes.
- **Cliente:** Es cualquier dispositivo que se conecta al Broker para enviar o recibir información. Se encarga de realizar dos funciones: publicar/suscribir.
- **Publicar:** Es el envío de información al Broker para distribuirla a los clientes interesados en base al nombre del "topic".

- **Suscribirse:** Los clientes le dicen al Broker que “topics” les interesan. Cuando un cliente se suscribe a un “topic”, cualquier mensaje publicado al Broker es distribuido a los suscriptores de ese “topic”. Los clientes también pueden cancelar la suscripción para dejar de recibir mensajes del Broker sobre ese “topic”.
- **Topic:** Es la manera de identificar el mensaje. Los clientes publican, se suscriben hacen ambas cosas a un “topic”.

El protocolo MQTT dispone de 3 niveles de QoS (calidad de servicio), que son los que establecen la manera de entrega del mensaje al cliente y la confirmación de la entrega [94].

El protocolo MQTT se basa en “topics” que se organizan de forma jerárquica para gestionar los mensajes que son enviados y recibidos por cada cliente, estos “topics”, son la base para obtener y publicar la información en un Broker MQTT, esto permite separar la información en niveles y además, se tiene la opción de acceder a los datos de varios “topics” a la vez.

Los clientes inician una conexión TCP con el Broker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza. Por defecto, MQTT emplea el puerto 1883 y el 8883 cuando funciona sobre TLS (Seguridad de la capa de Transporte) [95], en la Figura 2.7 se muestra lo explicado anteriormente.

En la Figura 2.8 se muestra la topología básica de la arquitectura MQTT con tres clientes, donde el cliente de la izquierda (Raspberry Pi) hace la función de publicador en un instante de tiempo para que el mensaje sea procesado por NodeRed, se envía el mensaje asociado a un “topic”, donde el Broker es el encargado de distribuirlo a todos los clientes de la derecha que están suscriptos a dicho a dicho “topic”. Este proceso se realiza en ambas direcciones ya que cada cliente es un publicador y a la vez suscriptor dependiendo de dónde se origina el mensaje.



Para enviar los mensajes el cliente emplea mensajes PUBLISH, que contienen el topic y el payload.



Para suscribirse y desuscribirse se emplean mensajes SUBSCRIBE y UNSUBSCRIBE, que el servidor responde con SUBACK y UNSUBACK.



Por otro lado, para asegurar que la conexión está activa los clientes mandan periódicamente un mensaje PINGREQ que es respondido por el servidor con un PINGRESP. Finalmente, el cliente se desconecta enviando un mensaje de DISCONNECT.

Figura 2.7. Funcionamiento de MQTT. (Fuente: [31]).

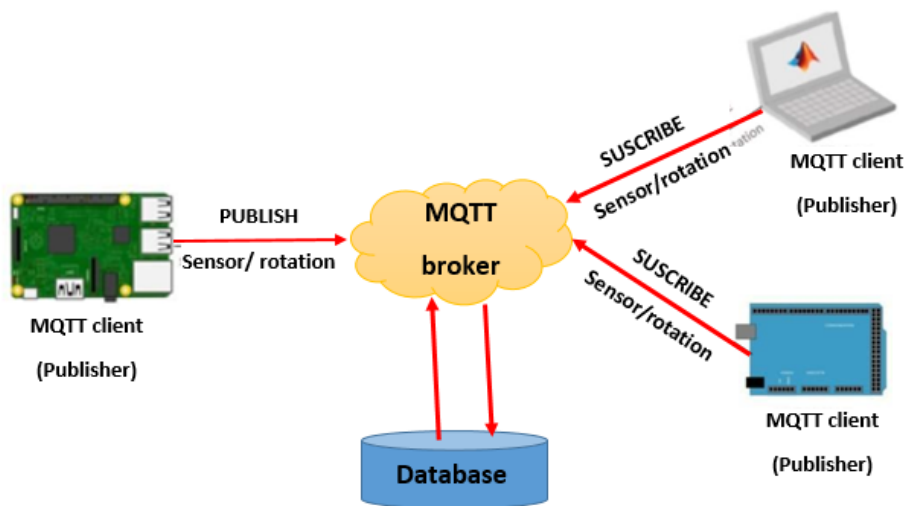


Figura 2.8. Ejemplo de Arquitectura MQTT. (Adaptada de: (Ine4C, 2023)).

Mensaje MQTT

El formato de un mensaje MQTT se encuentra dividido en tres grandes bloques, encabezado fijo, encabezado variable y carga útil, en la Figura 2.9 se muestra la estructura completa del mensaje MQTT [96].

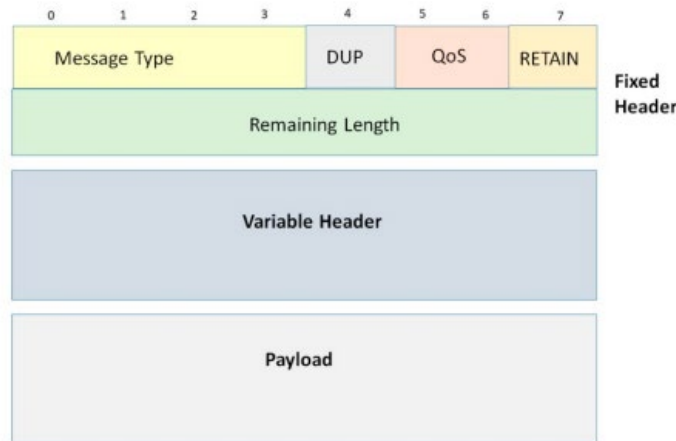


Figura 2.9. Formato de un mensaje MQTT (Adaptada de: [32]).

- **Encabezado fijo:** Es obligatorio y debe contener los siguientes campos:
- **Message Type:** Identifica el tipo de paquete. Los tipos más comunes de mensaje son CONNECT que realiza la conexión entre el publicador o suscriptor y el Broker o “PUBLISH”, se utiliza para recibir datos o almacenarlos en el Broker y el mensaje tipo “SUBSCRIBE”, mediante el cual se realiza la petición al Broker para acceder al contenido del “topic” al que se quiere acceder.
- **Duplication:** Es un bit con dos estados: activado o desactivado. Indica que un mensaje ya ha sido recibido.
- **QoS:** Son 2 bits por los cuales se codifican tres tipos diferentes de calidades de servicio.
- **Retain:** Es un bit con dos estados: activado o desactivado. Indica al Broker si se desea mantener el mensaje y la QoS indicada para futuros suscriptores.

- **Remaining Length:** Indica el número de bytes que faltan en el mensaje.
- **Encabezado variable:** Se utiliza para añadir información de control.
- **Carga útil:** Contiene la información que se quiere transmitir o recibir de un “topic” a través de un Broker.
- **Contenido (payload):** Es el contenido real del mensaje. Puede tener un máximo de 256 Mb, aunque en implementaciones reales el máximo es de 2 a 4 kB.

2.1.12 Servicios Web

La WWW (“World Wide Web”), o internet, junto con el correo electrónico, se convirtieron en la herramienta más utilizada por la población para el intercambio de información. Con ello la popularización de lo que se conocen como servicios web; los cuales se refieren a la unión de funciones de aplicación que se ejecutan usando programación en internet. Permiten el intercambio de información entre entidades, se conectan dinámicamente y ejecutan actividades en tiempo real con la mínima intervención humana.

Protocolo HTTP

El protocolo de transferencia de Hipertexto (“Hypertext Transfer Protocol”), es el protocolo base para el uso de la web, este protocolo está orientado a conexión por lo que emplea el protocolo TCP para el envío de datos por la red, además el protocolo HTTP, crea conexiones independientes para el envío de datos por lo que no existe relación entre las mismas, por lo que para acceder a una página web es necesario hacer el envío de todo el código HTML del texto y los archivos multimedia que contenga el sitio.

Existe una variante de HTTP llamada HTTPS (S por “secure”) que utiliza el protocolo de seguridad SSL (“secure socket layer”) para cifrar y autenticar el tráfico entre cliente y servidor.

Lenguaje HTML

El lenguaje de marcado de hipertexto (“HyperText Markup Language”), es un conjunto de elementos que definen la estructura del contenido de los sitios web y junto a HTTP son los encargados de poder presentar los datos de la capa de Aplicación.

En general, el funcionamiento principal de un servidor web o servidor HTTP es almacenar y transmitir el contenido solicitado de un sitio web al navegador del usuario, cuando el navegador requiere el acceso a un archivo que está almacenado en el servidor web, el navegador lo solicitará al servidor mediante el protocolo HTTP. Cuando la petición alcanza al servidor web correcto (hardware), el servidor HTTP (software) acepta la solicitud, encuentra el documento requerido y lo envía de regreso al navegador, también a través de HTTP, pero si el servidor no encuentra el documento requerido, devuelve una respuesta 404 de error.

Para que el servidor web funcione correctamente debe tener la capacidad de permanecer encendido todo el tiempo para no interrumpir el servicio o servicios que les ofrecen a sus clientes, ya que si el servidor falla o se apaga no se puede acceder al sitio web solicitado [97].

2.1.13 NodeRed

Es una plataforma de programación basada en flujos, que trabaja por eventos; es decir, la estructura y ejecución de lo que se programa se determina por sucesos definidos por el usuario o por sucesos provenientes del mismo sistema.

Fue creado en 2013 por Nick O’Learly y Dave Conway, su objetivo es crear una herramienta que permita la manipulación y visualización de tópicos del protocolo MQTT, permite crear aplicaciones escalables y soportar múltiples conexiones simultáneas.

Una de sus características más importantes es que permite al usuario crear sus propios nodos, de acuerdo a sus necesidades e instalarlos de una manera muy sencilla [98].

Funciona a través de una plataforma basada en Node.js, que es un entorno de ejecución de JavaScript.

Cuenta con nodos predeterminados divididos en ocho categorías: entradas, salidas, funciones, social, almacenamiento, análisis, avanzado. Si se desea emplear nuevos nodos, se permite agregar directamente desde la paleta de nodos del mismo entorno de NodeRed. En la Figura 2.10 se muestran algunos de los nodos mencionados.

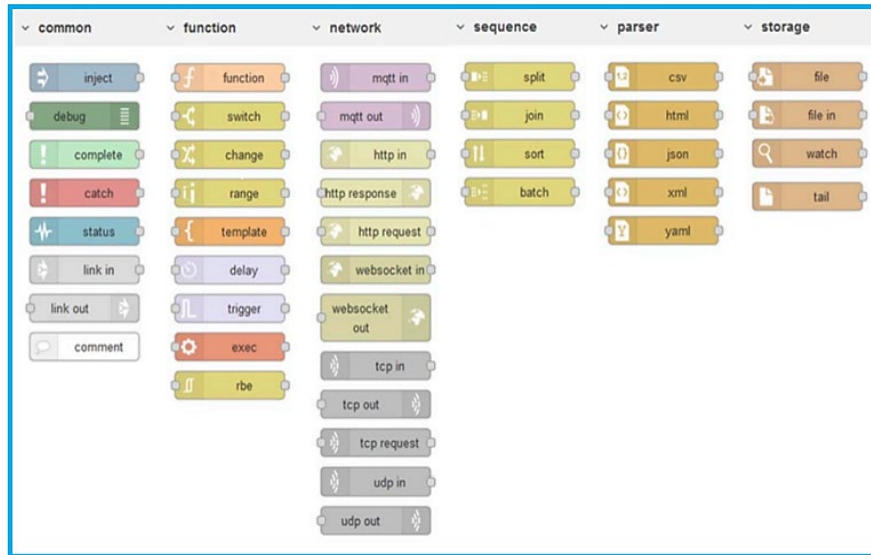


Figura 2.10. Nodos predeterminados de NodeRed. (Fuente: [34]).

En la Figura 2.11 se muestra la interfaz de la plataforma, para una mejor comprensión de su funcionamiento, se enumeran sus principales partes, para enseguida dar una breve explicación de cada una de ellas.

1. Se encuentran todos los nodos que se pueden utilizar, se organizan por tipos.
2. Se tienen las pestañas para navegar entre flujos o “flows”
3. Es la parte central es el espacio de trabajo, en donde se arrastran y ordenan los nodos que se van a utilizar.
4. Es el apartado de las herramientas, aquí se le proporciona al usuario guía de cada uno de los nodos, mostrando la respuesta que se obtiene por consola y las variables almacenadas junto a su valor.

5. Se encuentra el botón de “deploy” que funciona como compilador para ejecutar la programación de los nodos.

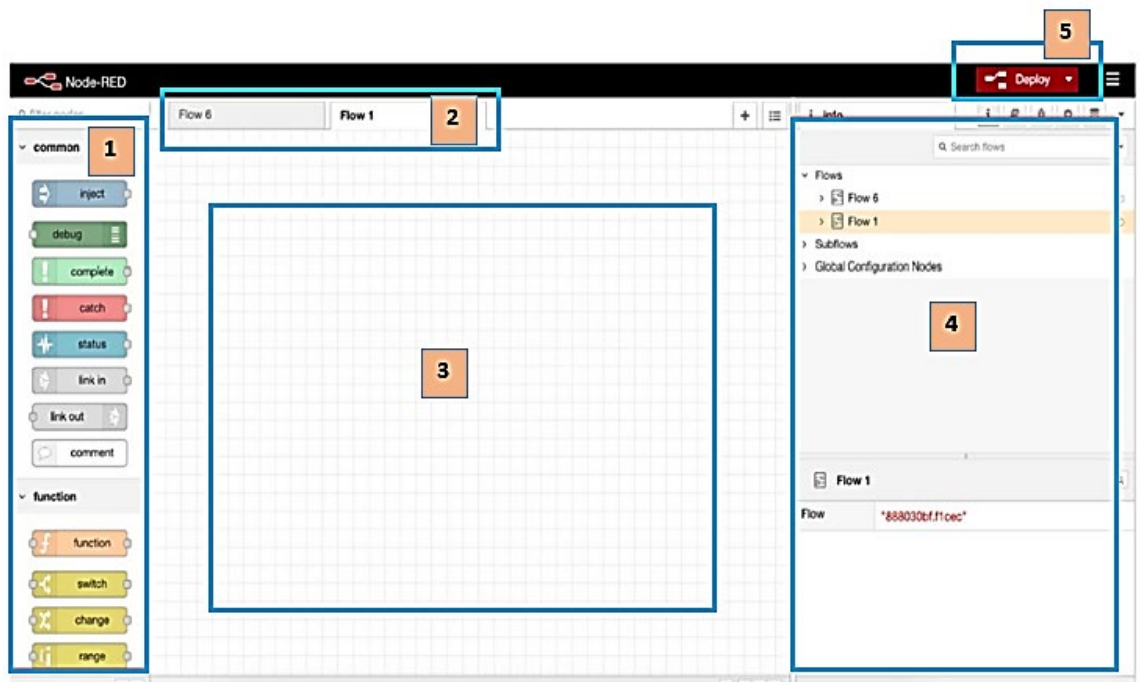


Figura 2.11. Entorno de uso de la plataforma NodeRed. (Fuente: [34]).

2.2 MARCO METODOLÓGICO

El marco metodológico es el conjunto de métodos, herramientas y técnicas de recolección de datos, utilizado para dirigir el desarrollo de un proyecto de manera eficiente para lograr los objetivos planteados [99].

2.2.1 Metodología para el diseño y construcción del sistema domótico

El desarrollo de la tesis se basa en una metodología por prototipos, concretamente “Metodología por prototipos modular”. Este tipo de metodología es dinámica, es decir, facilita que todo el sistema, o algunas de las partes que lo constituyen puedan ser construidas de manera individual y ser integradas al resto del desarrollo mismo, lo cual permite ir agregando

modificaciones de acuerdo a las necesidades que surjan para dar un cumplimiento óptimo de los objetivos [100].

La metodología propuesta consta de cuatro fases con capítulos y actividades que están interrelacionados entre ellos, en la Figura 2.12 se muestra la estructura general de la metodología: Fase 1. Identificación del problema; Fase 2. Diseño del sistema domótico; Fase 3. Construcción del sistema domótico y Fase 4. Evaluación y mejoramiento del sistema domótico.

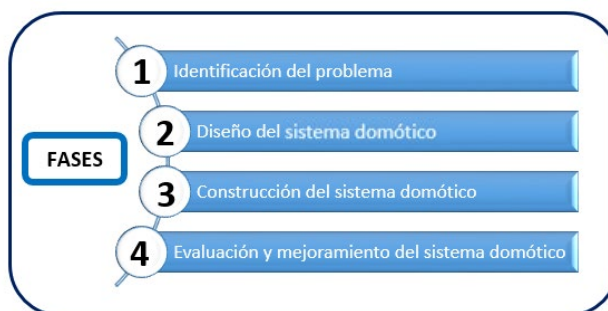


Figura 2.12. Fases de la metodología propuesta. (Elaboración propia, 2023).

Considerando los diferentes cambios y factores que pueden afectar de manera positiva o negativa el desarrollo del proyecto, se opta por una metodología ágil o adaptativa, además de apoyarse de la retroalimentación en puntos específicos, de modo que se abre la posibilidad de agregar nuevos métodos, procesos o herramientas para optimizar o mejorar los resultados parciales obtenidos. En la Figura 2.13 se muestra el diagrama de bloques de la metodología propuesta con sus fases y actividades planteadas para cada una de ellas.

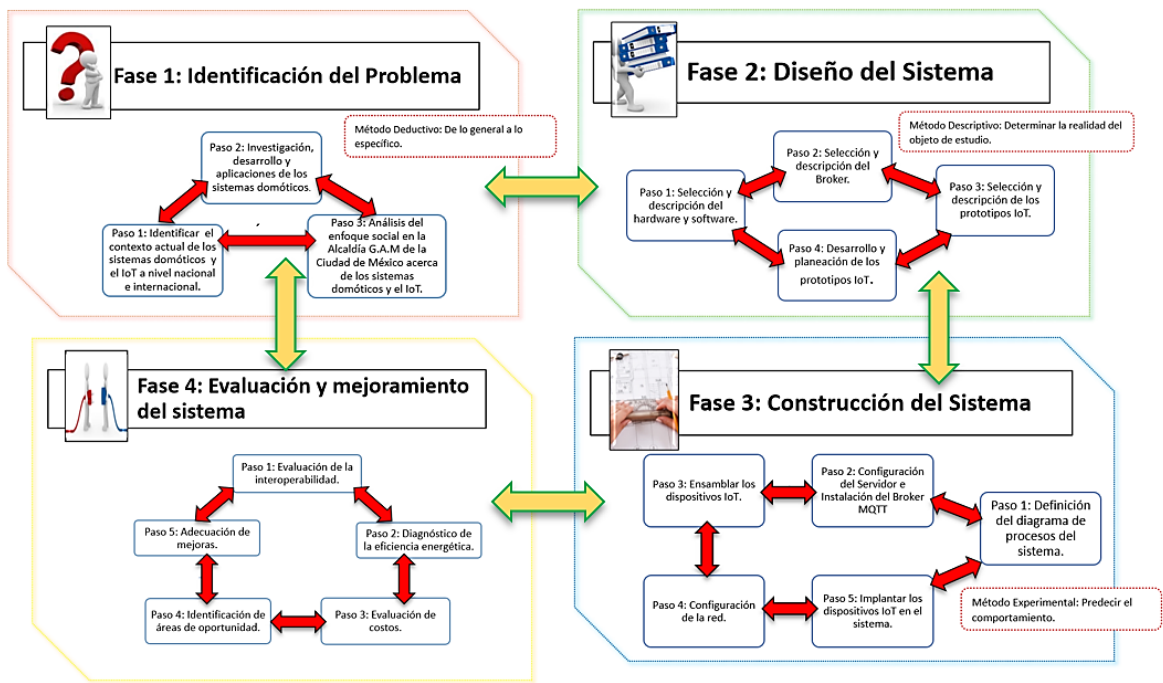


Figura 2.13. Diagrama de bloques de la metodología propuesta (Elaboración propia, 2023).

A continuación, se describe de manera breve cada una de las fases:

2.2.2 Fase 1: Identificación del problema

La finalidad de esta fase es analizar las necesidades del usuario y proponer un sistema domótico con integración IoT que sea capaz de satisfacerlas, partiendo de lo general a lo específico, con la ayuda de diversas fuentes de información para contextualizar la propuesta de solución. En esta fase se realiza la recopilación de datos históricos, conceptuales y teóricos que permiten contextualizar el desarrollo del mercado IoT. Paralelamente se realiza una encuesta que permite analizar las necesidades de los usuarios ya que la implementación del sistema domótico está orientada a ello. En la Figura 2.14 se muestran los pasos a seguir para dar un cumplimiento satisfactorio a esta fase.

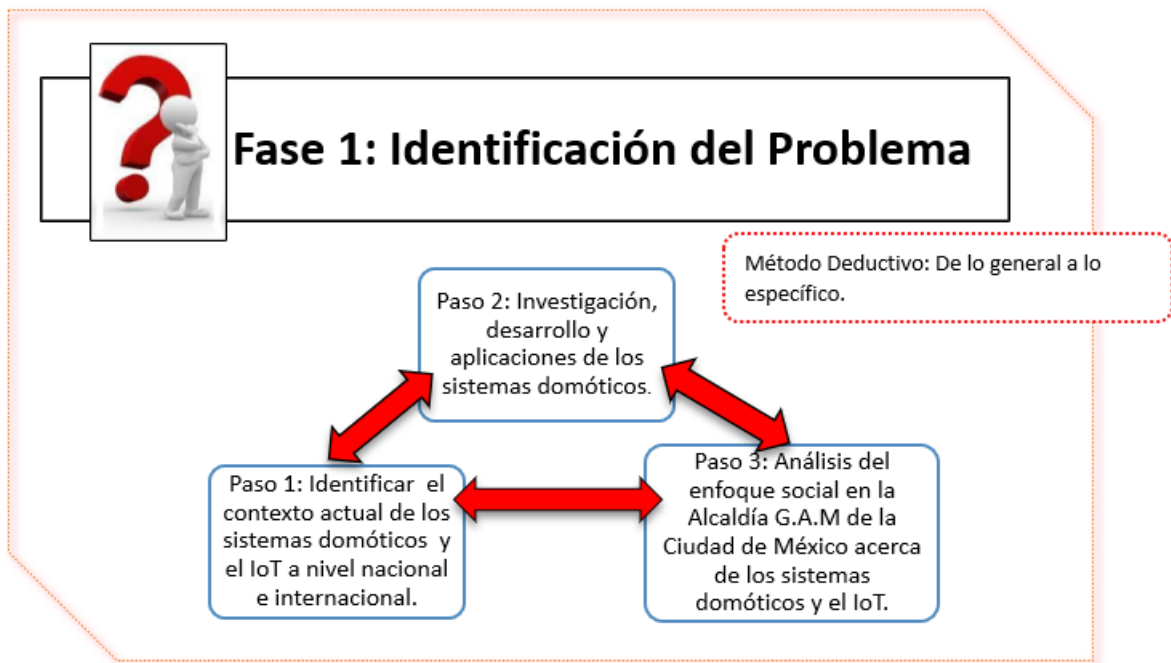


Figura 2.14. Fase 1: Identificación del problema (Elaboración propia, 2023).

2.2.2.1 Paso 1: Análisis de la situación actual a nivel nacional e internacional

Es necesario tener un contexto del estado tecnológico, social y de mercado de los sistemas domóticos y el IoT para el correcto desarrollo del proyecto de tesis, para ello se realiza una investigación siguiendo la metodología mostrada en la Tabla 1.8, se observan los pasos que se llevan a cabo para un manejo óptimo de la información recabada, y así obtener únicamente la información útil para el proyecto.

Tabla 1.8. Manejo de la Información. (Elaboración propia, 2023).

Búsqueda de información	Para la realización de esta etapa de la investigación se hace una búsqueda de recopilando información de distintas fuentes utilizando un navegador Web especializado (Google Académico) a través de palabras clave. La información se obtiene de bases de datos (estadísticas), libros especializados en el tema de interés, páginas web ya sea de contenido académico o en páginas de proveedores que ofrecen servicios relacionados al tema de interés, revistas, artículos, tesis, etc. Todo se hace a mediante Google académico para obtener referencias científicas y de investigación que garanticen la veracidad de la información.
-------------------------	--

Análisis de la información	Teniendo una recopilación de información, debe leerse cuidadosamente para seleccionar y extraer únicamente la información valiosa para el tema del proyecto.
Síntesis de la Información.	Teniendo un conjunto de información, que ayuda a sustentar el proyecto, se hace una síntesis de la misma. Se extraen ideas principales, se realizan diagramas, y lo más importante, parafrasear los contenidos agregando una perspectiva personal. Después la información se categorizarla y ubica según corresponda dentro de la estructura propuesta para el proyecto de tesis. Todo esto con el fin de que la información sea fácil de entender, pero que aporte el contenido necesario para lograr el fin deseado.

2.2.2.2 Paso 2: Investigación, desarrollos y aplicaciones de los sistemas domóticos

Se realiza una investigación para conocer el contexto actual de la domótica y el IoT en el mercado siguiendo la misma metodología de la Tabla 1.8.

2.2.2.3 Paso 3: Análisis del enfoque social en la alcaldía G.A.M. de la Ciudad de México acerca de los sistemas domóticos y el IoT

Para dar cumplimiento con este paso, se utiliza como método de investigación una encuesta, la cual está diseñada como se representa en la Figura 2.15.

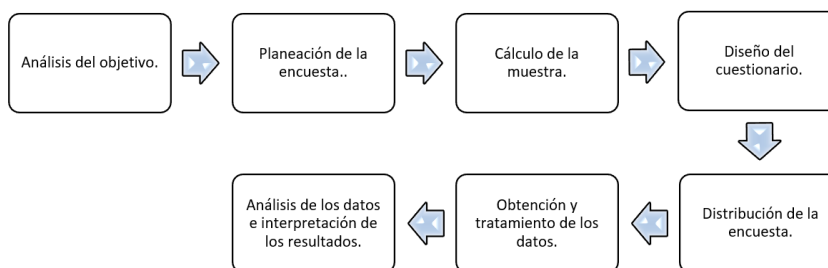


Figura 2.15. Mapa de procesos de la realización de la encuesta. (Elaboración propia, 2023).

Como primer paso se define ¿el para qué? de la encuesta; La encuesta se realiza con la finalidad de analizar el nivel de conocimiento, necesidades y poder adquisitivo de los usuarios acerca de los sistemas de domótica e IoT.

- a) ¿Una vez que se sabe el para qué?, se determina el ¿Cómo? en Donde y a ¿Quién? se le realizara la encuesta; El estudio estadístico se realiza mediante el uso de un formulario de “Google Forms” dentro de alcaldía Gustavo A. Madero, que es una de las dieciséis demarcaciones territoriales de la Ciudad de México. Dicha encuesta se distribuye compartiendo la URL de forma virtual para su llenado correspondiente.
- b) Para realizar el cálculo de la muestra se utiliza la técnica de muestreo aleatorio simple, pues no hay restricciones respecto a características específicas de las personas a encuestar. Partiendo del número de habitantes en la G.A.M. obtenido directamente del Instituto Nacional de Estadística y Geografía (INEGI) se toma como referencia el número de muestras mínimas a obtener para que el resultado sea significativo. Para ello se utiliza un software desarrollado en MATLAB el cual permite obtener resultados matemáticos confiables, además de utilizar una herramienta especializada de cálculo del sitio “SURVEY MONKEY”, la cual es una plataforma especializada en aplicación de encuestas, con la finalidad de obtener datos contundentes y veraces de nuestro objeto de estudio.
- c) De la síntesis de información de la Tabla 1.8 se seleccionan los conceptos más coloquiales, es decir aquellos que no contengan gran información técnica y puedan ser identificados por la población a encuestar. Se utilizan preguntas cerradas con un abanico de respuestas específicas que permiten obtener concretamente la información que se desea, además facilita el tratamiento estadístico de las respuestas.
- d) Posterior a tener el formulario completo se distribuye de manera virtual mediante un link, ya que es más fácil hacerla llegar a una cantidad significativa de la población de manera más eficiente. Este link se comparte por correo electrónico y redes sociales como WhatsApp, Facebook e Instagram con la población seleccionada y se les solicita contesten con la mayor sinceridad posible sin hacer una búsqueda previa de los

conceptos, esto con la finalidad de tener un resultado fidedigno de la situación actual de la Domótica y el IoT en su vida cotidiana.

- e) Una vez que la encuesta es resuelta y entregada dentro de la plataforma “Google Forms”, se hace una selección, síntesis y evaluación de los resultados obtenidos. Google Forms permite interpretar las respuestas mediante la generación de gráficas y datos concluyentes que facilitan el análisis de los resultados. La aplicación de la encuesta comprende desde el mes de noviembre del 2022 al mes de febrero del 2023.
- f) Del análisis de los resultados se hace una evaluación integral de las tendencias e indicadores correspondientes a cada pregunta y con ello conocer mejor a los sujetos de estudio; esta información permite responder acertadamente a sus necesidades y expectativas.

2.2.3 Fase 2. Diseño del sistema domótico

En la Fase 2, se analizan y comparan distintos tipos de tecnologías para seleccionar la que mejor se adecue a las necesidades del sistema basado en las respuestas de la encuesta por parte de los usuarios y con ello generar la planeación adecuada en la construcción del mismo, además de generar la documentación necesaria que describa el comportamiento del sistema para que sea posible su materialización en la Fase 3. Consta de 4 pasos que se muestran en la Figura 2.16.

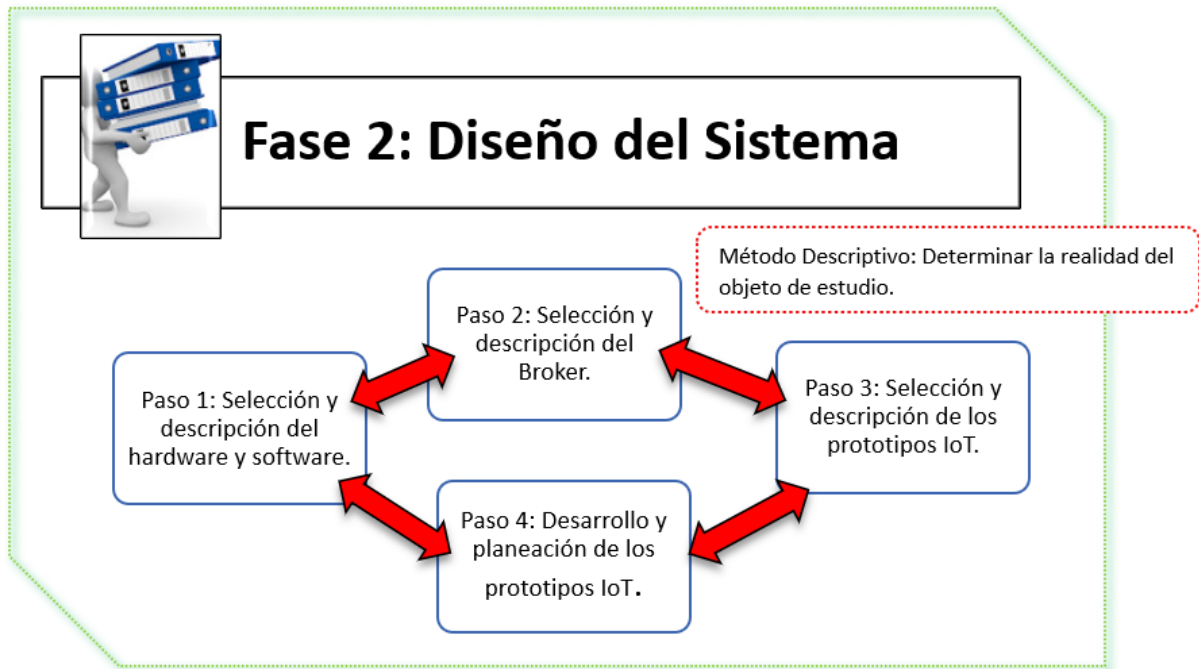


Figura 2.16. Fase 2: Diseño del sistema domótico. (Elaboración propia, 2023).

2.2.3.1 Paso 1: Selección y descripción del hardware y software

En el marco teórico, se plasma la información acerca de diversas tecnologías tanto de transmisión como de procesamiento y control a nivel de hardware y software existentes. Partiendo de esa base y sumado a las consideraciones derivadas del resultado de la encuesta se determina el uso de protocolos, medios de transmisión y demás componentes necesarios para el diseño del sistema Domótico con integración IoT.

Teniendo claros los requisitos que el hardware y software deben cumplir para el adecuado desarrollo del proyecto, se realiza la comparación y selección de las tecnologías a utilizar para garantizar que el hardware y software utilizado sea seguro, robusto y fiable.

En la Tabla 1.9 se muestran las consideraciones requeridas para seleccionar el hardware y software.

Tabla 1.9. Consideraciones de Software y Hardware. (Elaboración propia, 2023).

	Características	Hardware	Software
Tipo de vivienda	La arquitectura de la vivienda no es relevante para la selección de hardware y software.	Se considera la existencia previa tanto de una instalación eléctrica como de una conexión a Internet.	No hay requisitos previos.
Elementos a automatizar	Derivado de los resultados de la encuesta, se determinan los dispositivos que el sistema domótico incluye.	De acuerdo a la investigación realizada en el marco teórico, donde se hace un análisis de distintas tecnologías, se determinan los elementos adecuados para la construcción del sistema domótico.	De acuerdo a la investigación realizada en el marco teórico, donde se hace un análisis de distintas tecnologías, se determinan los elementos adecuados para la construcción del sistema domótico.
Protocolos de comunicación	Se requiere de un protocolo de comunicación que sea capaz de comunicar los dispositivos dentro del sistema con un alto nivel de desempeño consumiendo los menos recursos posibles.	Se requiere de dispositivos compatibles.	Bajo el objetivo de tener un sistema de bajo costo, se requiere de software libre de licencias.

2.2.3.2 Paso 2: Selección y descripción del servidor

La selección del servidor (Bróker MQTT) del sistema mantiene el objetivo de optimizar costos, lo cual conlleva la utilización de un Broker local que no incremente los costos de implementación. Por ello se selecciona un Broker de código abierto (“Open Source”) y se le realiza un análisis FODA que se muestra en la Tabla 1.10 para corroborar que cumple con los requisitos del proyecto.

Tabla 1.10. Análisis FODA Broker MQTT. (Elaboración propia, 2023).

Fortalezas	Debilidades
<ul style="list-style-type: none"> • Open Source • Simplicidad de uso • Gran eficiencia a bajo costo 	<ul style="list-style-type: none"> • Tiene que ser ejecutado de forma local. • Poca integración con servicios de terceros en la nube. • Bajo nivel de encriptación por defecto.
Oportunidades	Amenazas
<ul style="list-style-type: none"> • Utilización de certificados de seguridad • Utilización de puertos TCP seguros 	<ul style="list-style-type: none"> • Bajo nivel de seguridad • Vulnerabilidad ante ataques DD2 • Depende totalmente del software donde se está ejecutando (Raspberry)

2.2.3.3 Paso 3: Esquemas y descripción de los prototipos IoT

Es necesario tener una referencia física del espacio donde se van a implementar los dispositivos del sistema, para ello se realiza un plano de dos dimensiones que describe la localización geográfica de los elementos que conforman la vivienda y los dispositivos IoT mediante el uso del software de diseño arquitectónico “Coohom” en su versión Web y un editor de fotografía.

En dicho plano se representa una versión genérica de una vivienda la cual cuenta con habitaciones para dormir, sala/comedor, baños, cocina, cochera y jardín.

Si bien es cierto, que esta vivienda no representa una versión auténtica de los hogares de las personas entrevistadas, únicamente se utiliza a modo de poder representar la distribución de los prototipos IoT con que cuenta el sistema Domótico, ya que como se estableció en los objetivos de la tesis, el sistema pretende poder ser implementado en cualquier tipo de vivienda, sin necesidad de hacer modificaciones arquitectónicas.

Con la herramienta de diseño arquitectónico “Coohom” se diseña cada uno de los espacios dentro de la vivienda y se utiliza un plano de vista superior para visualizar mejor dichos espacios en una representación 2D.

Posteriormente se utiliza un editor de fotografía básico (Paint) para agregar los símbolos que representan los dispositivos IoT dentro del plano.

Para la creación de dichos prototipos IoT, se consideró la información generada por la encuesta, donde la participación de la gente encuestada nos indica qué tareas son más relevantes en la implementación de un sistema domótico.

Los dispositivos IoT que considera el Sistema Domótico propuesto son:

- Control de Cargas; encendido y apagado de luminaria y electrodomésticos.
- Control de accesos; control de apertura y cierre de puertas y ventanas.

- Monitoreo de temperatura, humedad, detección de fugas de gas.
- Monitoreo en tiempo real de la vivienda mediante el envío y transmisión de imágenes mediante una cámara de vigilancia.

El sistema debe ser capaz de realizar todas estas tareas de forma remota o local, además de no necesitar instalar ninguna aplicación de terceros para su control. Por ello se emplean diversas herramientas de programación para el desarrollo de los prototipos; como hardware se utiliza una tarjeta de desarrollo económica, dotada de la tecnología inalámbrica necesaria para las tareas que el sistema requiere, la cual utiliza los elementos del código de Arduino para su programación, el cual se basa en C++. Además, se hace uso del lenguaje de programación JavaScript para la creación de la plataforma web y la comunicación con el Broker MQTT.

Para cada dispositivo es necesario crear un algoritmo que describa el funcionamiento y la lógica bajo la cual trabaja, para visualizar este algoritmo se utilizan diagramas de flujo, los cuales establecen los procesos de dicho algoritmo. Para la realización de los diagramas de flujo se utiliza el software Visio de Microsoft y abstracción lógica con lo cual se puede representar el proceso básico que sigue el funcionamiento de cada uno de los dispositivos IoT que se desarrollan.

2.2.3.4 Paso 4: Mapeo y programación de los dispositivos

Una vez que se ha descrito el proceso de comunicación de los dispositivos y se tiene definido su funcionamiento, se procede a crear su esquema electrónico, para ello se emplea el software de código abierto "Fritzing"; al cual se puede acceder directamente desde su página web: <https://fritzing.org/> , con esta herramienta es posible diseñar de forma virtual los prototipos IoT y tener una noción teórica del comportamiento de dichos dispositivos, tomando en cuenta las especificaciones de los fabricantes según las hojas de datos de cada componente electrónico que se utilizará.

2.2.4 Fase 3. Construcción del sistema domótico

En esta fase, se parte de las consideraciones tomadas en las fases anteriores: para iniciar el diseño de la propuesta del sistema domótico, lo primero a considerar es el área geográfica donde el sistema puede ser desplegado. Para ello se propone una pequeña casa genérica cuya distribución se describe en la fase anterior. En la Figura 2.17 se muestra los pasos que conforman esta fase.

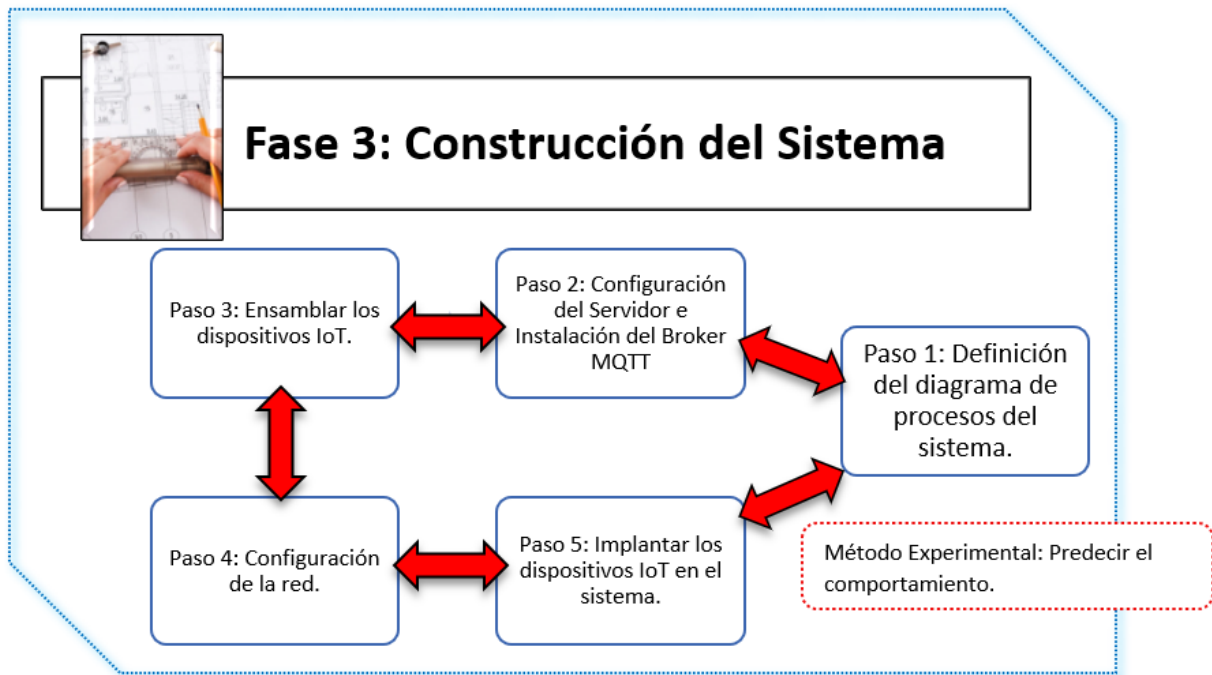


Figura 2.17. Fase 3. Construcción del Sistemas Domótico. (Elaboración propia, 2023).

De acuerdo a los resultados obtenidos en la encuesta, se determinan los dispositivos que el sistema debe incluir, sabiendo que la principal necesidad a cubrir es la seguridad, se propone la implementación de sensores que apoyen al monitoreo y toma de acciones ante situaciones de riesgo. Los sensores realizan la medición de niveles de temperatura, humedad y niveles de gas, los cuales pueden ser monitoreados en tiempo real desde el Dashboard del sistema. Sumado al apartado de seguridad, se contempla un mecanismo de cerradura eléctrica auxiliar a las cerraduras tradicionales previamente instaladas en la vivienda, esta cerradura se activa

por medio de Internet desde la misma página web, además de enviar notificaciones vía correo electrónico (Gmail) con lo cual el usuario se asegura que el acceso a su vivienda está controlado y monitoreado de manera remota.

Se contempla el control de luminaria y cargas eléctricas, con lo cual se pueden accionar diversos electrodomésticos, motores, o cualquier dispositivo eléctrico que funcione con corriente alterna. Se pretende que el sistema sea un complemento a las actividades diarias del usuario, facilitando el monitoreo y control remoto de sus equipos eléctricos.

2.2.4.1 Paso 1: Diagrama de Procesos del Sistema

En este paso ya se tiene claro cómo es que será la comunicación entre dispositivos, cuál será el proceso y los protocolos con los cuales se realiza el envío de datos por ello es importante realizar un esquema general del modelo de comunicación que permite visualizar completamente todo este proceso, es decir definir la topología física y lógica que se implementó y poder correlacionarla con la arquitectura IoT seleccionada.

2.2.4.2 Paso 2: Servidor y Bróker MQTT

Lo primero a realizar en este paso es llevar a cabo la instalación del sistema operativo de Raspberry Pi, el cual permite utilizar la tarjeta de desarrollo (Raspberry Pi 4, modelo B) propia de Raspberry como un micrordenador. Una vez instalado el sistema operativo de Raspberry, mediante el uso de la consola de comandos se descarga y actualiza el Bróker previamente seleccionado (Eclipse Mosquitto). Es necesario realizar la configuración del puerto TCP utilizado por Mosquitto ya que es necesario para poder levantar los servicios del Broker, para ello se hace “forwarding ports” (redirigir conexiones a través del router), directamente en el router del ISP lo cual permite el correcto envío y recepción de datos mediante el protocolo MQTT.

Utilizando la consola de comandos se realiza la instalación de Node.js y NodeRed, lo cual permite acceder al entorno de programación Web, donde se configura cada uno de los nodos que componen al sistema.

De igual forma se debe hacer “forwarding port” en el router del ISP.

2.2.4.3 Paso 3: Ensamblado de los dispositivos

En este paso, se realiza la conexión física entre los sensores, actuadores y la tarjeta de desarrollo ESP32, mediante el uso de cable de cobre se conectan las GPIO previamente designadas en el programa realizado en el IDE de Arduino.

También es en este paso donde es necesario tener la maqueta donde se distribuirán los prototipos para realizar las conexiones eléctricas necesarias.

2.2.4.4 Paso 4: Configuración de la Red

A cada uno de los dispositivos IoT se le asigna un cliente MQTT, este cliente MQTT posee una dirección IP estática la cual es asignada en el proceso de comunicación desde la ESP32 hasta el Broker.

Usando una tarjeta ESP32 es posible controlar diferentes prototipos IoT, por tanto, tendremos asignadas diferentes direcciones IP las cuales son de vital importancia para poder controlar el sistema desde internet.

Para poder controlar y tener acceso al sistema de forma remota por internet se utiliza un servicio de DNS dinámico, el cual provee los servicios necesarios para mantener la IP estática del Broker y los clientes MQTT con el router y ligarlo a una URL que dará acceso al sistema desde cualquier parte del mundo de manera automática y sin tener que escribir direcciones IP en el navegador, esto es posible al usar el servicio de DDNS del proveedor No-Ip, el cual es gratuito en su versión más básica.

2.2.4.5 Paso 5: Implantar los Dispositivos en el sistema

Haciendo uso de la plataforma NodeRed y teniendo ya programados los clientes MQTT se configura el Dashboard que contine la interfaz del sistema.

Este es el paso final antes de poder controlar los prototipos IoT por internet mediante el protocolo de comunicación MQTT.

2.2.5 Fase 4. Evaluación y mejoramiento del sistema domótico

En esta fase es necesario someter al sistema a ciertas pruebas de desempeño para verificar que se cumplan los objetivos iniciales del proyecto de tesis. En la figura 2.18 se muestra cada uno de los pasos a seguir en esta fase.

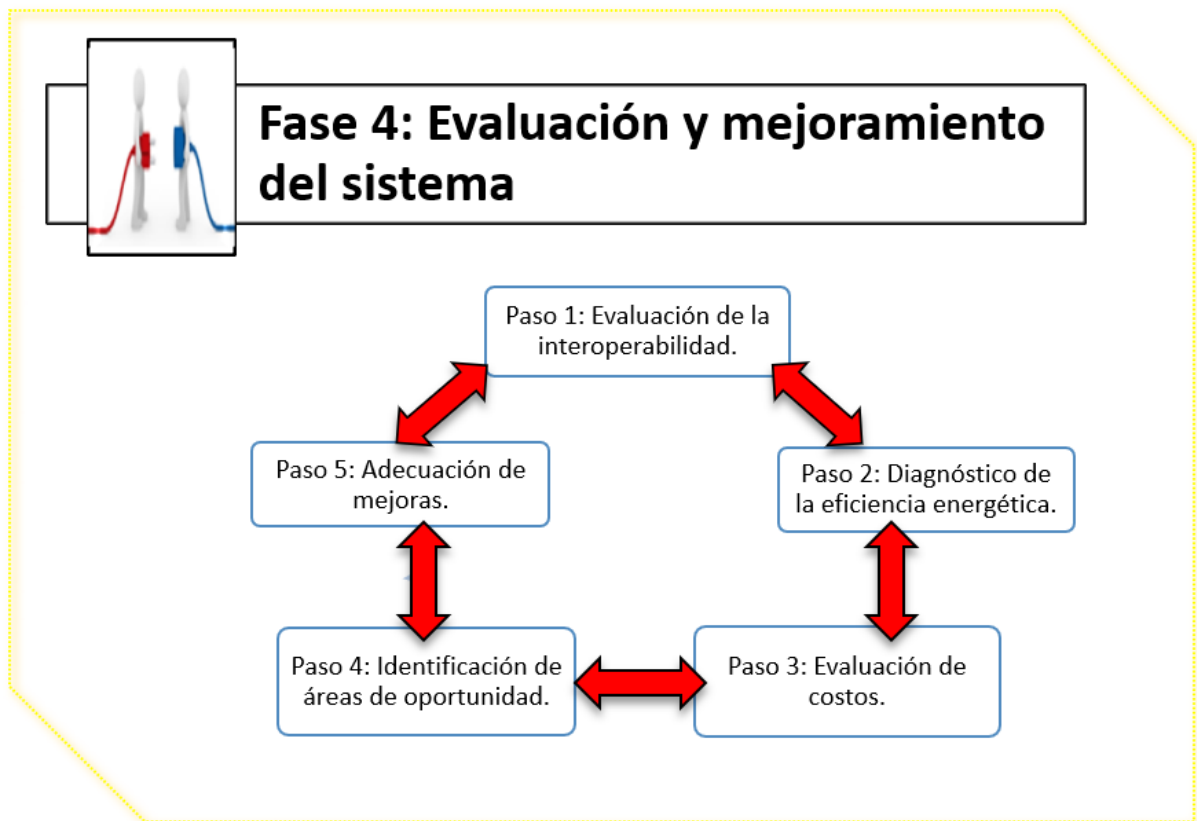


Figura 2.18. Fase 4. Evaluación y mejoramiento del Sistema. (Elaboración propia, 2023).

Siguiendo la metodología propuesta, que en un inicio es dinámica y modular, permite ir y venir entre fases lo cual da la posibilidad de realizar las adecuaciones necesarias para garantizar un correcto funcionamiento del sistema.

2.2.5.1 Paso 1: Evaluación de la Interoperabilidad

Para comprobar que el sistema cuenta con esta propiedad, basta con añadir nuevos dispositivos previamente programados de la misma forma que los dispositivos que conforman al sistema originalmente ya que no existe la restricción del uso de protocolos propietarios o cerrados. Gracias al uso de NodeRed es posible llevar a cabo esta tarea, ya que sus nodos permiten la integración de dispositivos compatibles con los asistentes virtuales más comerciales del mercado como lo son Amazon Alexa o Google Assistant.

2.2.5.2 Paso 2: Diagnóstico de la eficiencia energética

Para comprobar que el sistema cumple con el objetivo de ahorro energético, se utiliza un Vatímetro o medidor de consumo eléctrico con el cual se hacen mediciones durante lapsos determinados de tiempo para compararlas con el consumo energético habitual y con ello calcular el costo eléctrico de la implementación del sistema en el hogar.

2.2.5.3 Paso 3: Evaluación de Costos

El consumo eléctrico evaluado en el paso anterior nos permite hacer un cálculo de cuánto cuesta desplegar y mantener en funcionamiento el sistema, también se realiza un análisis de costo del mercado Domótico y de IoT en el país, donde se compara el costo de dispositivos comerciales disponibles y el costo de producción de los prototipos propuestos.

Para este análisis se verifica el precio de focos inteligentes, cerraduras, sensores, interruptores y demás artículos en venta en tiendas de tecnología de los fabricantes más populares y se genera una tabla comparativa que muestre la diferencia económica que representa adquirir cada uno de estos productos.

2.2.5.4 Paso 4: Identificación de Áreas de Oportunidad

Bajo la misma premisa de tener una metodología que nos permite desplazarnos entre fases, resulta evidente que la experimentación es la única que nos permite adquirir este conocimiento empírico derivado del funcionamiento de los prototipos, es por esto que se realiza un análisis de los aspectos que pueden ser mejorados en el diseño, construcción e implementación del sistema domótico propuesto.

2.2.5.5 Paso 5: Mejoramiento

Después de analizar el funcionamiento del sistema, es posible determinar qué elementos no funcionan como se deseaba en su concepción y cuales otros pueden trabajar mejor y depurar ciertos errores que se presentan en su funcionamiento, si bien el alcance de la tesis no es llevar el sistema a un ámbito comercial, si es posible determinar qué características pueden ser añadidas y que otras no validan su permanencia dentro del sistema.

Para esto, una vez más se recurre a los resultados de la encuesta y se analizan las respuestas; las cuales marcan una tendencia en las preferencias de los usuarios y con ello poder se generar propuestas de mejoramiento para trabajos futuros. Dado que la naturaleza del sistema permite un alto grado de escalabilidad.

Mediante un diagrama de flujo de mejora continua de los procesos dentro de la construcción del sistema podemos evaluar y determinar qué puntos son los que el sistema puede mejorar para trabajos futuros.

Finalmente, se utiliza una maqueta como herramienta auxiliar para presentar un prototipo a escala del Sistema Domótico planteado que contiene todos los componentes electrónicos y que permite comprobar su funcionalidad.

Para la elaboración de la maqueta debemos considerar lo siguiente:

- Decidir el tamaño de la maqueta: Para ello, se hace una estimación previa para saber el tamaño final que tendrá la maqueta.

- Elegir los materiales: Las maquetas suelen construirse de un solo material o de varios, en función también del resultado que se quiera conseguir, dependiendo del material elegido será el tipo de fijación de los elementos.
- Corte de piezas: Para realizar este paso, lo mejor es utilizar máquinas de corte láser, para obtener piezas lo más limpias posibles.
- Preparar todas las piezas: Esto permite tener una visión preliminar de como quedara la maqueta y determinar si se debe realizar alguna modificación.
- Unión de las piezas: En este punto, ya se tiene una distribución general de todas las piezas de la maqueta, por lo que unir todos sus elementos será una tarea más fácil.

La maqueta del sistema no tiene que ser un modelo a escala de una vivienda, por lo cual basta con representar un tablero donde se pueda distribuir todos los componentes que conforman al sistema. De esta manera se tiene una representación visual del sistema funcionando.



CAPÍTULO 3

IMPLEMENTACIÓN DE LA METODOLOGÍA

En este capítulo se materializa la metodología propuesta en el Capítulo 2, se describe objetivamente cada una de las tareas realizadas en el desarrollo del Sistema Domótico para garantizar el cumplimiento de los objetivos propuestos.

De forma experimental se analiza el funcionamiento del sistema y finalmente se generan resultados evaluando su desempeño.

Finalmente se plasman los resultados para tener evidencia de la culminación satisfactoria de este proyecto de tesis.

3.1 Fase 1: Identificación del problema

Para comenzar el desarrollo de esta fase, se parte del análisis e interpretación de la información recabada en el marco teórico, en el cual se visualizan las tendencias y paradigmas de los sistemas Domóticos y el Internet de las Cosas.

El siguiente paso es analizar el nivel de conocimiento, necesidades y poder adquisitivo de los usuarios acerca de los sistemas de Domótica e IoT. Se desea que, con los resultados obtenidos, se pueda determinar el sector de la población en el que el sistema puede tener mayor adhesión, si están dispuestos a invertir, de ser así, la cantidad de dinero que estarían dispuestos a gastar y, además, cuáles dispositivos les interesan estén presentes en un Sistema Domótico.

3.1.1 Análisis del enfoque social y económico en la alcaldía Gustavo A. Madero de la Ciudad de México, acerca de los sistemas domóticos y de IoT

El estudio estadístico se realiza dentro de alcaldía Gustavo A. Madero, que es una de las dieciséis demarcaciones territoriales de la Ciudad de México. Es la segunda alcaldía, después de Iztapalapa, con mayor número de habitantes, de acuerdo al INEGI, en el año 2022 el número de habitantes era de 1,173,351 [101]. Su demarcación geográfica se muestra en la Figura 3.1.

Cuenta con una extensión territorial de 94.07 Km², tiene colindancias con algunos de los municipios del Estado de México (Coacalco de Berriozábal, Tlalnepantla de Baz, Ecatepec de Morelos, Nezahualcóyotl y Tultitlán) y con las alcaldías Venustiano Carranza, Azcapotzalco y Cuauhtémoc. Su rango de temperatura está entre los 12 y 18°C con un clima templado subhúmedo con lluvias en verano de menor humedad 72.80% y seco-semiseco templado 27.20% [102].

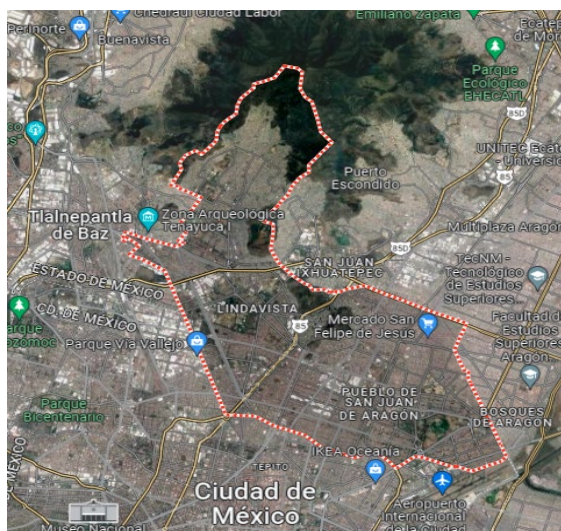


Figura 3.1. Demarcación geográfica de la Alcaldía Gustavo A. Madero. (Fuente: Google Maps, 2023).

3.1.2 Planeación de la encuesta

Para esta investigación cuantitativa se utiliza la técnica de muestreo aleatorio simple, pues no hay restricciones respecto a características específicas de las personas a encuestar, lo que quiere decir que todos los componentes de la población tienen las mismas posibilidades de formar parte de la muestra y a su vez cada una de las posibles muestras del mismo tamaño tiene la misma probabilidad de ser escogida [103].

3.1.3 Calculo de la muestra

De acuerdo con los datos del INEGI [101] en el año 2022, en la alcaldía Gustavo A. Madero se registran 1,173,351 habitantes, partiendo de esta cantidad se calcula el número de

muestras que se deben obtener para que el resultado sea significativo. Para el cálculo del tamaño muestral, se utiliza la fórmula de muestreo aleatorio para una población finita y conocida [104](Ecuación 3.1.1).

$$n = \frac{(p)(q)(N)(Z^2)}{E^2(N - 1) + Z^2(p)(q)}$$

Ecuación 3.1.1. Fórmula de muestreo aleatorio para una población finita y conocida [104].

Donde:

n = Tamaño de la muestra

N = Población total a encuestar

p = Porcentaje estimado de la variabilidad positiva (50%)

q = 100-p (variabilidad negativa)

E = Error o precisión de estimación permitido (5%)

Z = Nivel de confianza en la distribución Z (95%)

Para el cómputo de la ecuación 3.1.1, se utiliza un programa informático llamado MATLAB, en donde se diseñó un script, en el cual se ingresan los valores de las variables correspondientes y se obtiene un resultado confiable y exacto, tal como se muestra en la Figura 3.2.

Con este resultado, se hace la consulta en la calculadora en línea del sitio web SurveyMonkey, el cual es un sitio especializado en diseño e implementación de encuestas [105]. En la Figura 3.4 se muestran los datos obtenidos mediante la calculadora de SurveyMonkey. Lo anterior tiene el propósito de comparar los resultados obtenidos con el script de MATLAB y así tener una referencia sólida en los resultados obtenidos.

Calcula tu margen de error

Tamaño de la población ①
1173351

Nivel de confianza (%) ①
90

Tamaño de la muestra ①
225

Margen de error
5%

Figura 3.4. Margen de error: calculadora SurveyMonkey (Fuente: Extraído de [105]).

De esta manera se garantiza que con al menos **225** personas que respondan la encuesta, representan un porcentaje significativo dentro de la población de la alcaldía G.A.M. de acuerdo a la fórmula de muestreo aleatorio para una población finita y conocida.

3.1.4 Diseño del Cuestionario

El cuestionario es el instrumento básico utilizado en una investigación por encuesta, capaz de suscitar respuestas fiables, válidas cuantificables. Para el diseño del cuestionario es necesario partir del objetivo y variables previamente establecidos, las preguntas deben ser elaboradas de acuerdo a las características de la población objetivo (lenguaje, nivel cultural, edad, etc.) [103].

Se utilizan preguntas cerradas con un abanico de respuestas específicas que permiten obtener concretamente la información que se desea, además facilita el tratamiento estadístico de las

respuestas. Las preguntas son claras, cortas y sencillas, para que el encuestado las entienda fácilmente y que puedan ser respondidas en menos tiempo.

El cuestionario fue diseñado y elaborado mediante la herramienta Formularios de Google, lo cual permite que la encuesta sea atractiva y fácil de responder. Se realizó una breve explicación en la parte inicial del cuestionario para la población encuestada como se muestra en la Figura 3.5.



Figura 3.5. Presentación cuestionario Domótica & IoT. (Elaboración propia, 2023).

La primera sección del cuestionario se enfoca en conocer el sector de la población a que pertenece el individuo que responde. Con esto se clasifica de mejor manera los resultados.

La segunda parte del cuestionario se enfoca en preguntas meramente conceptuales que permiten identificar el nivel de conocimiento y familiarización con el mundo de la domótica y el IoT.

3.1.5. Distribución de la encuesta

Derivado de la emergencia sanitaria producida por el COVID-19 en el país, se decidió distribuir de manera virtual mediante un link, ya que es más fácil hacerla llegar a una cantidad significativa de la población de manera más eficiente y segura. Este link se comparte por correo electrónico y redes sociales como WhatsApp, Facebook e Instagram, dejando a los

encuestados la libertad de responderla en el momento que ellos consideren más pertinente. Por otro lado, “Google Forms” permite interpretar las respuestas mediante la generación de gráficas y datos concluyentes que facilitan el análisis de los resultados. La aplicación de la encuesta comprende desde el mes de octubre del 2022 al mes de febrero del 2023.

3.1.6. Obtención y tratamiento de los datos

La identidad de los individuos que respondieron la encuesta permaneció anónima, ya que al momento de diseñarse el cuestionario no era relevante conocer sus identidades, basto con conocer su nivel de conocimiento respecto a los conceptos de Domótica & IoT, como se mencionó anteriormente solo se les clasifico por sus respuestas en la primera parte de la encuesta (estudio socioeconómico).

Derivado de la participación de la población encuestada se pudo obtener la siguiente información:

Del total de personas que participaron en la encuesta el 57.5% fueron mujeres y el 41.2% hombres con un rango de edad entre los 20 y 40 años. Además, el 61.5% manifiesto tener estudios a nivel licenciatura, el 56.6% vive en casa propia y el 42% tiene un ingreso mayor a \$10,000 mensuales, lo cual se observa en la Figura 3.6.

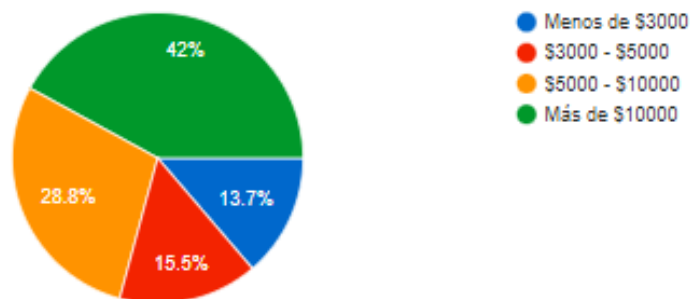


Figura 3.6. Ingresos promedio de la población encuestada. (Fuente: Encuesta Google Forms, 2023).

De acuerdo a la información recabada se sabe que la mayoría de la población encuestada cuenta con vivienda propia, el 52.7% tienen entre 3 y 5 habitaciones, el 46% con garaje propio y la respuesta con más porcentaje acerca de los focos con los que cuenta su vivienda es de 51.8% para el rango de entre 6 y 10 focos, el 47.8% dispone de entre 4 a 6 ventanas.

Una de las preguntas más significativas de esta parte de la encuesta es la oportunidad de conocer los servicios con los que cuenta la vivienda, el 92.9% del total de muestras cuenta con el servicio de internet, el 98.7% con energía eléctrica y solo el 28.3% con cámaras de seguridad, donde este resultado final en concreto, nos muestra que es bastante bajo el porcentaje de personas que tienen un sistema de seguridad en sus hogares.

Al cuestionar a las personas encuestadas si consideran que los sistemas domóticos son costosos, se encuentra que un 57.1% de los encuestados consideran que es costoso y un 20.4% que es muy costoso, esto se visualiza en la Figura 3.7.

¿Cómo consideras que es el costo de estos dispositivos?

226 respuestas

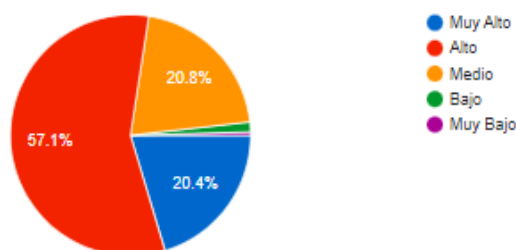


Figura 3.7. Consideración del costo de dispositivos de sistemas domóticos. (Fuente: Encuesta Google Forms, 2023).

Esta consideración respecto al costo en los sistemas domóticos hace que las personas vean este tipo de sistemas como un lujo y no como una necesidad, ya que el 55.8% de los encuestados lo afirmaron, el 38.9% dijeron que depende del contexto en el que se adquieran

los sistemas y solo el 5.3% dieron respuestas que podrían englobarse dentro del mismo criterio del contexto.

Los dispositivos inteligentes con los que mayormente se cuenta en el hogar son: Smart TV, asistente virtual y lavadora con un 85%, 35% y 23.5% de popularidad respetivamente.

Dichos dispositivos se encuentran dentro de una gama de marcas tales como Apple, Steren, Amazon, TP-Link, Huawei y las más populares Samsung y LG.

Otro punto importante es que un gran número de los encuestados consideran que la seguridad es el aspecto más importante en su hogar, seguido por la tecnología, el ahorro energético y el confort, lo cual ofrece información relevante de las necesidades específicas que debe satisfacer el sistema.

Por otro lado, el 76.1% del total de encuestados dijo estar interesado en invertir en un sistema de automatización, de los cuales el 50.9% estaría dispuesto a invertir hasta \$5,000 y un 16.8% hasta \$10,000 sin embargo, el 27.4% muestra su rechazo a este tipo de sistemas, tal como lo muestra la Figura 3.8

¿Cuánto estarías dispuesto a pagar por un sistema de automatización de tu hogar?

226 respuestas

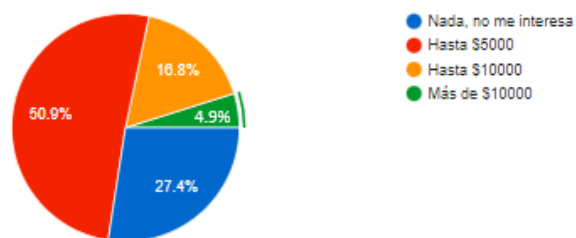


Figura 3.8. Consideración para invertir en un sistema doméstico el hogar. (Fuente: Encuesta Google Forms, 2023).

Para conocer cuáles son los dispositivos con más aceptación entre los usuarios, se cuestionó sobre cuáles de ellos consideran una necesidad para su hogar, siendo los que brindan seguridad y prevención de incidentes los más populares.

De acuerdo con la información obtenida de la encuesta, se sabe que el dispositivo con mayor preferencia para agregar y controlar dispositivos inteligentes en el hogar es el teléfono móvil (celular) con un 88.1%, seguido de la Tablet con un 3.5% y la computadora con un 3.1%.

El 49.6% de los encuestados que cuenta con dispositivos inteligentes, dice que para cada nuevo que deseen agregar necesitan una aplicación para controlarlo, el 19% lo agrega a una aplicación previamente instalada, el 8.4% lo hace mediante un asistente virtual y el 14.2% utiliza los 3 métodos, el resto de los encuestados dijeron no tener dispositivos inteligentes y por lo tanto no tienen que hacer esta tarea.

Relacionado a la pregunta anterior, el 30.1% de los usuarios afirma utilizar una aplicación para intercomunicar cada uno de sus dispositivos inteligentes, un 15.5% utiliza una sola aplicación y un 24.8% algún asistente virtual, el resto de los encuestados dijeron no tener dispositivos inteligentes o no utilizar ninguna de las opciones dadas para controlarlos. Esto se muestra en la Figura 3.9.

Para intercomunicar tus dispositivos, utilizas:
226 respuestas



Figura 3.9. Intercomunicación de dispositivos. (Fuente: Encuesta Google Forms, 2023).

Para finalizar, de las personas interesadas en invertir en un sistema domótico, un poco más del 77% esperan que este sistema favorezca en aspectos de ahorro energético, alrededor del 78% en aspectos de seguridad y el 85% en el confort.

3.1.7. Análisis e interpretación de resultados

Nivel de Conocimiento de los conceptos de domótica e IoT

De las 226 personas encuestadas, solo el 38.94% dijo saber a qué se refiere el término domótica, sin embargo, el 59.73% de estos seleccionó el enunciado correcto sobre el concepto de domótica. De manera que se puede observar que no existe congruencia con la afirmación de conocer el concepto de domótica y su respuesta al seleccionar el enunciado correcto, además de dar evidencia de que la población encuestada no cuenta con elementos sólidos que demuestren que realmente tienen conocimiento sobre el término de domótica y lo que ella engloba.

Por otro lado, el 57.52% afirmó conocer el término IoT “Internet de las cosas”, sin embargo, solo el 49.56% eligió la opción correcta para describir este concepto. Por lo que la población encuestada tiene conocimiento sobre IoT, pero existe confusión sobre lo que realmente significa y engloba el término.

La información recabada a través de la encuesta permitió determinar las bases del sistema domótico a desarrollar. La población a la que va dirigido el sistema son personas dentro de un rango de edad de 20 a 40 años, con un nivel de estudios promedio de licenciatura, ya que, estos tienen un mayor conocimiento acerca de los sistemas que permiten automatizar un hogar, el interés en implementar la tecnología en su vida, además de la economía para sustentarlo.

Para poder integrar un nuevo dispositivo inteligente en el hogar los usuarios deben agregar una aplicación nueva para gestionarlo, por lo que se propone un sistema interoperable que se pueda controlar desde una página web con una interfaz fácil de operar por el usuario, basado en protocolos que sean compatibles con el IoT, que tengan un bajo consumo de recursos y permitan un intercambio de información fiable y rápida entre dispositivos por internet.

El sistema se va a conectar a la red de internet residencial, el requisito de contar con internet para que el sistema pueda funcionar no debe representar un gasto extra, ya que, se espera que alrededor de un 90% cuente con este servicio en el hogar.

Se desea que el sistema sea asequible y pueda ser instalado en cualquier vivienda sin tener que modificar las condiciones arquitectónicas de los inmuebles. Con la información recabada se observó que la gran mayoría viven en casa propia de más de dos recámaras, con por lo menos 1 baño, garaje, cocina, comedor y sala. Además de contar con al menos 10 focos, por lo que el gasto energético es considerable debido a estas características en las viviendas.

La población espera que el sistema domótico brinde seguridad, por ello, se deben considerar sistemas que permitan detectar la intrusión, daño o cualquier tipo de actividad que represente un riesgo para las personas dentro y en el entorno de su vivienda. De igual forma contar con sistemas tanto de prevención como de acción ante una situación de esta índole. Por otra parte, la seguridad dentro de casa ante algún evento o siniestro se debe tomar en cuenta, ya que los accidentes son frecuentes y pueden poner en riesgo la integridad física y patrimonio de las personas. De manera que se hace necesario la implementación de sensores que apoyen al monitoreo y toma de acciones ante este tipo de situaciones.

La población encuestada considera que un sistema domótico es costoso y por ende un lujo en sus vidas, sin embargo, sus respuestas manifiestan que sí estarían dispuestos a invertir en sistema de esta naturaleza, necesitando que el sistema les brinde principalmente seguridad, seguida por ahorro energético y confort. Esta información es relevante para poder iniciar con el planteamiento del sistema, ya que, esta respuesta da fiabilidad en que hay interés por adquirir un sistema que cubra las necesidades tanto personales como de la vivienda.

Uno de los principales objetivos es hacer un sistema económico y accesible en comparación a los que se ofertan actualmente en el mercado por grandes fabricantes. El diseño del sistema, así como su implementación debe estar dentro de un rango aproximado de \$5,000 a \$10,000

pesos mexicanos sin incluir el servicio y mantenimiento futuro; lo cual permitiría a gran parte de la población poder adquirir dicho sistema.

“Hoy en día, el Smartphone es uno de los dispositivos más utilizados a nivel global, su uso condiciona el día a día de los usuarios, en México, en el 2021 más del 78% de la población mayor a seis años de edad son usuarios de un teléfono celular o Smartphone [106], es importante mencionar que, además el 84.5% accede a él a través de un Smart Phone”; lo cual reafirma el resultado de la encuesta, donde se aprecia la tendencia del uso de dispositivos móviles como el “SmartPhone”, como el predilecto de la población encuestada para el control de un sistema domótico.

Aunque la mayoría de las personas, cuentan con internet en su vivienda; no están familiarizados con los sistemas domóticos y del internet de las cosas, sin embargo, consideran importante invertir en un sistema que mejore su calidad de vida, brinde seguridad y confort, pero sobre todo que sea una opción que les permita ahorrar energía.

En conclusión, la población encuestada considera interés en acceder a tecnologías domóticas que les permitan cubrir las necesidades de seguridad, confort y ahorro energético en sus hogares; por lo que este proyecto representa una alternativa viable, alcanzable y económica que permita al público ser parte del avance tecnológico y sobre todo mejorar su calidad de vida a través de un sistema eficiente, económico y funcional.

3.2 Fase 2. Diseño del sistema domótico

Después de analizar la información obtenida de la encuesta, en donde las principales necesidades del público son hacer un uso más eficiente del consumo energético, brindar seguridad y aportar confort en sus hogares, se propone un Sistema Domótico con integración al Internet de las Cosas que cumpla las siguientes características:

- Que tenga un bajo costo de implementación; con un rango promedio de \$7,500 pesos mexicanos.

- Que se pueda controlar desde cualquier dispositivo sin necesidad de instalar aplicaciones de terceros.
- Que el consumo energético del sistema no represente gastos considerables y que por el contrario aporte significativamente al ahorro energético del hogar.
- Que el sistema aporte seguridad en el hogar.
- Que el sistema proporcione confort en la realización de tareas dentro de la vivienda.

Por lo cual, se hace la propuesta de desarrollar un sistema que integre diferentes dispositivos capaces de monitorear y controlar diversas tareas dentro del hogar:

Seguridad

1. En el ámbito de la seguridad se propone un sistema de alarma capaz de obtener lecturas provenientes de sensores de gas y detección de flama que permita el envío de alertas en tiempo real con lo cual se minimice el riesgo de incendios dentro de la vivienda.
2. Un sistema de alertas en tiempo real que gestione el control de acceso a la vivienda por medio del desarrollo de una cerradura que notifique al usuario cuando sea abierta.
3. Un sistema de alertas en tiempo real que notifique la apertura de ventanas cuando sean abiertas.
4. Un mecanismo de monitoreo por medio del uso de un módulo de cámara, el cual sea capaz de transmitir imagen en tiempo real y tomar fotografías cuando sucedan ciertas acciones dentro de la vivienda (captura de fotografías al abrir la cerradura o apertura de la ventana).

Confort y ahorro energético

1. Monitoreo de temperatura y humedad que permita conocer el estado de ciertas zonas dentro de la vivienda por medio de la lectura de sensores. Con lo cual se puedan tomar

decisiones que modifiquen el entorno a conveniencia del usuario. (Modificar la temperatura de una habitación encendiendo o apagando un sistema de calefacción).

2. Mecanismo de control de cargas eléctricas, mediante el cual se pueda controlar el encendido de luminaria, energizar y accionar electrodomésticos de forma remota, con lo cual el consumo eléctrico de ciertos electrodomésticos (que permanecen enchufados a la red eléctrica) este controlado. Como agregado, el poder controlar el encendido de luces dentro de la vivienda, aporta simulación de presencia en la vivienda con lo cual se inhibe la presencia de intrusos dentro de la misma.

3.2.1 Paso 1: Selección y descripción del hardware y software

Es evidente que el avance tecnológico actual permite tener una amplia gama de opciones al momento de seleccionar los componentes necesarios para el desarrollo del sistema, por ello siguiendo la metodología propuesta, mediante el uso de la Tabla 1.9 presente en el Capítulo 2 se determinan los elementos necesarios.

- a) El primer requisito indispensable para el sistema es que la vivienda cuente con servicio de internet y suministro de energía eléctrica. Es evidente que sin energía eléctrica difícilmente puede tenerse acceso a internet, por lo que esta consideración no puede modificarse.
- b) Para el Hardware necesario en la creación de los prototipos IoT, es necesario decantarse por aquellos componentes electrónicos cuya tecnología permita cumplir con los objetivos propuestos. Por ello se seleccionan aquellos cuyo costo y compatibilidad no representen inconvenientes en su uso.
- c) Se requiere de un protocolo de comunicación que sea capaz de comunicar los dispositivos, con un alto nivel de desempeño (consumiendo los menos recursos posibles) y que brinde interoperabilidad entre todos los componentes del sistema.
- d) La directiva de tener un sistema de bajo costo es imperativa, por ello el uso de software de código abierto facilita disminuir costos en su implementación.

ESP32

Esta tarjeta es una opción muy popular para proyectos IoT, debido a su conectividad inalámbrica Wifi y BLE, además de presentar un costo bajo y una gran eficiencia energética, sumado a su capacidad de procesamiento y el amplio soporte de desarrollo que presenta, por lo que la convierte en un dispositivo ideal para este sistema.

3.2.2 Paso 2: Selección y descripción del servidor

Mantener interoperabilidad entre todos los elementos que integran el sistema requiere seleccionar un protocolo de comunicación que sea capaz de intercomunicar los prototipos de forma eficiente. En el marco teórico se encuentra el protocolo MQTT, que como ya mencionamos es uno de los protocolos de transmisión más populares en IoT ya que mejora la eficiencia energética y proporciona un método ligero para enviar mensajes, haciéndolo adecuado para la mensajería de IoT, como con sensores de baja potencia o dispositivos móviles como teléfonos, computadoras integradas o microcontroladores.

Dado que el Broker es el nodo central del sistema tal como se muestra en la Figura 2.11 se opta por la utilización de un Broker local que no incremente los costos de implementación. Por ello se selecciona el Broker de código abierto **Eclipse Mosquitto**, al cual se le aplicó un análisis FODA que se realizó en la Tabla 1.10 donde se determina que es ideal para cumplir los objetivos propuestos.

Para poder disponer de un servidor MQTT con Mosquitto es necesario tener previamente instalado el Sistema Operativo Raspberry PiOS. Como se describe en el marco teórico, Eclipse Mosquitto es un Broker que se ejecuta de forma local, por lo cual es necesario utilizar una computadora que funcione como servidor web.

En este punto existen distintas alternativas bajo las cuales se puede desplegar un servidor web, lo primero es seleccionar la plataforma de desarrollo con la cual trabajaremos.

Como se mencionó anteriormente, mantener un bajo costo en el sistema es una de las consideraciones más importantes, por ello a partir de este momento se manejará este requisito como principal criterio para la elección de los componentes del sistema.

En la Tabla 1.11 se muestran los consumos promedio de 3 plataformas donde se puede instalar Eclipse Mosquitto, estos valores pueden variar dependiendo de los dispositivos (valores tomados de la web), pero de forma general y después de revisar las especificaciones de consumo de diferentes modelos se establecieron estos valores.

Tabla 1.11. Consumo promedio distintas plataformas. Fuente: CyberPuerta, 2023).

	LAPTOP	PC De Escritorio	RBPI4B
Consumo promedio en W	60 W	300 W	6.4W
Consumo en kWh	0.06 kWh	0.3kWh	0.0064kWh
Especificaciones	14.1" HD, Intel Celeron N3350 1.10GHz, 4GB, 64GB, Windows 10	PC Gaming CM-99913, AMD E1-6010 1.35GHz, 8GB, 240GB SSD, Windows 10	Raspberry Pi 4B, Broadcom BCM2711 Cortex-A72 (ARM v8) 1,8 GHz, 2GB RAM,
Costo (\$ MXN)	\$2500.00	\$2830.00	\$1300.00

De esta manera es necesario disponer de una microcomputadora que sea capaz de soportar los servicios necesarios para la ejecución de Mosquitto. La plataforma elegida es Raspberry Pi, concretamente Raspberry Pi 4B, la cual es una tarjeta de desarrollo ampliamente utilizada no solo en IoT si no en el mundo de las Telecomunicaciones.

Dado que la idea de contar con un sistema domótico que integre IoT está básicamente pensada en tener un bajo costo de implementación, consumo energético y facilidad de programación, podemos decantarnos por elegir montar el Broker en una mini computadora como Raspberry Pi, ya que tener una computadora convencional como servidor, incrementa su consumo energético, el costo de la inversión y el espacio físico para su instalación dentro del hogar.

Las ventajas que la utilización de una placa Raspberry en este proyecto son las siguientes

- Bajo consumo (a pleno rendimiento menos de 7W).
- Bajo costo.
- Multitud de posibles proyectos, al basarse en una distribución Debian (LINUX).
- Reducidas dimensiones; gracias a su diminuto tamaño (85x54mm) es posible instalarlo en múltiples sitios sin que ocupe demasiado espacio y mejorando la estética del diseño.

3.2.3 Paso 3: Esquemas y descripción de los prototipos IoT

3.2.3.1 Arquitectura para el desarrollo del proyecto

El funcionamiento del sistema está basado en una serie de sensores que envían pulsos eléctricos a una tarjeta de desarrollo, esta los procesa y transforma en una secuencia de bits, dicha tarjeta cuenta con un chip de comunicación inalámbrica con la cual se envían paquetes mediante una red Wifi, haciendo uso del protocolo MQTT. El Broker es el encargado de gestionar las peticiones de conexión cliente-servidor, traducir estos datos y determinar su comportamiento de acuerdo a su programación en NodeRed y la ESP32.

Posteriormente se publican estos datos en una página web. Una vez en la página web, se hace uso del protocolo HTML y JavaScript para obtener una interfaz gráfica donde el usuario interactúa con el sistema. Finalmente, el usuario puede controlar y monitorear los dispositivos dentro del sistema, en la Figura 3.10 se muestra el diagrama de bloques que representa el funcionamiento del sistema.

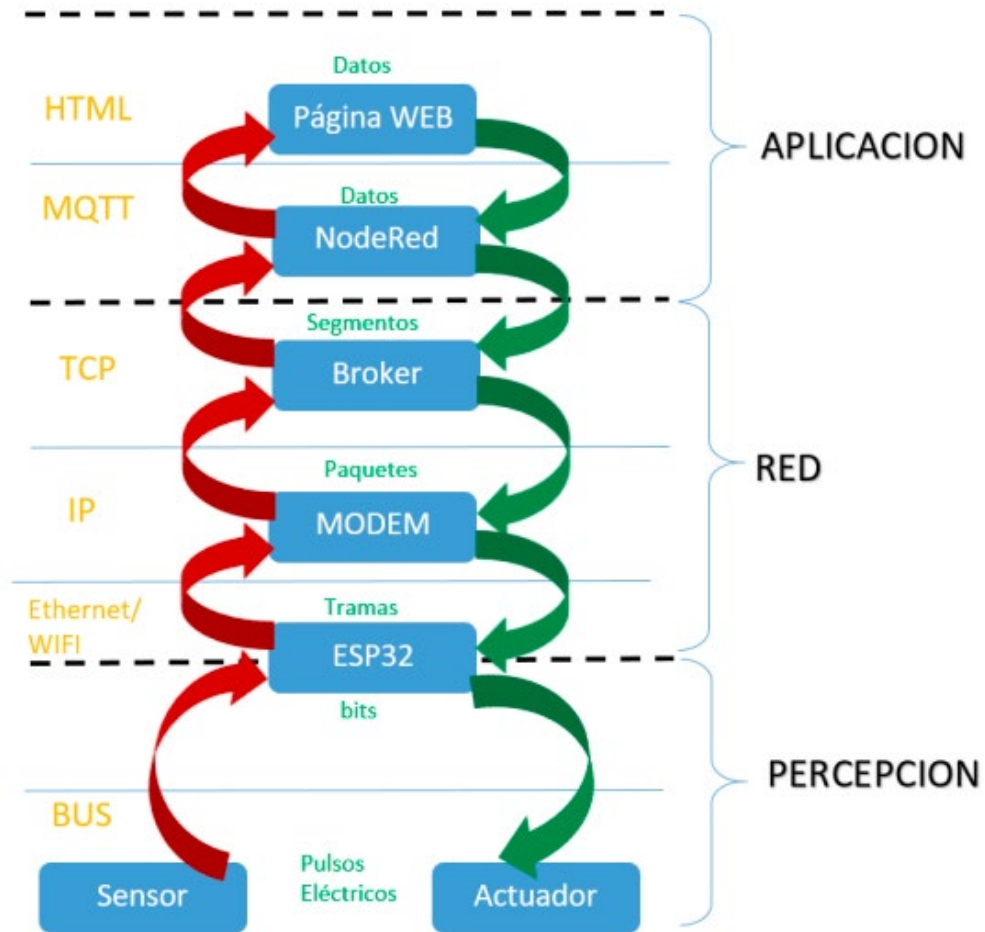


Figura 3.10. Diagrama de bloques del sistema propuesto. (Elaboración propia, 2023).

3.2.3.2 Distribución de los dispositivos IoT

Es necesario tener una referencia física del espacio donde se implementarán los dispositivos del sistema, para ello se realiza un plano de dos dimensiones que describe la localización geográfica de los elementos que conforman la vivienda y los dispositivos IoT mediante el uso del software de diseño arquitectónico “Coohom”, como se muestra en la Figura 3.11.

En este plano se ubica de forma ilustrativa la distribución de los prototipos IoT a desarrollarse y la estructura general del funcionamiento del sistema.

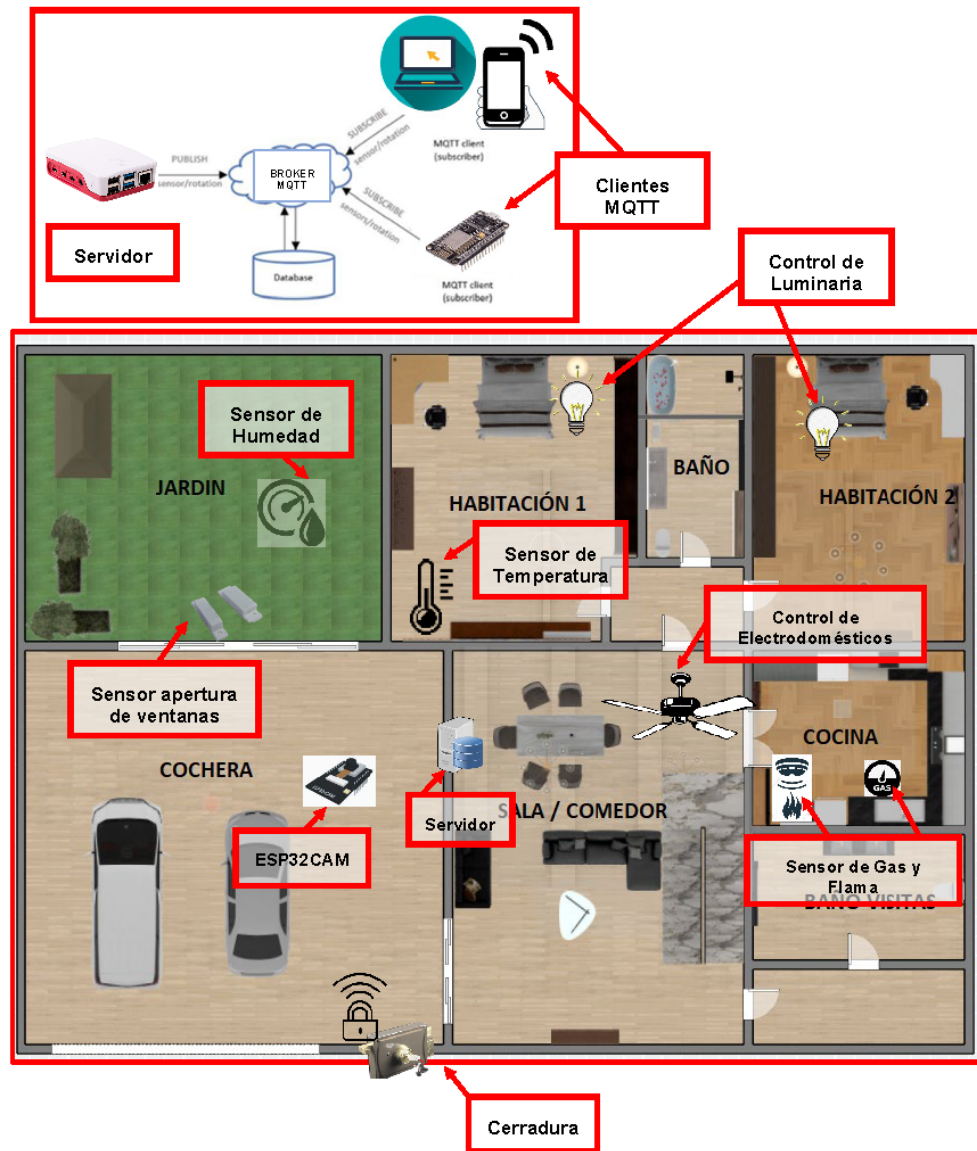


Figura 3.11. Distribución general de prototipos IoT. (Elaboración propia, 2023).

3.2.3.3 Paso 4: Mapeo y programación de dispositivos

En este paso, se procede a crear su esquema electrónico, para ello se emplea el software de código abierto Fritzing, con esta herramienta es posible diseñar de forma virtual los prototipos IoT. Cada prototipo será descrito detalladamente en la Fase 3.

3.3 Fase 3. Construcción del Sistema domótico

Como se mencionó en el marco metodológico, esta fase del proyecto parte de las consideraciones tomadas en las fases anteriores. Es decir, ya se tiene seleccionada la tecnología, el hardware y el software que se usará para implementar el sistema. Sin embargo, aún no se conoce como es que funcionará el sistema, para ello primero se debe crear un diagrama que defina el proceso que sigue el sistema para establecer comunicación y tomar decisiones.

3.3.1 Paso 1: Diagrama de Procesos del Sistema

Para comprender el proceso de comunicación se realizó el siguiente diagrama, el cual expone cual es el proceso lógico bajo el cual se realiza el establecimiento de publicación en el Broker MQTT, se muestra en la figura 3.12.

En la Capa de Percepción

Como se muestra en la Figura 3.10 el intercambio de información inicia como envió de pulsos eléctricos en la capa de percepción, en dicha capa se encuentran los sensores y actuadores, que a su vez son leídos por la placa ESP32.

La ESP32 recibe los pulsos eléctricos provenientes de los sensores usando gpio de entrada y los traduce en forma de bits en donde el microcontrolador, determina las acciones a realizar dependiendo de la programación realizada y establece del estado de sus gpio de salida.

En la Capa de Red

Después, la placa ESP32 se comunica vía WiFi hacia internet por medio del router del ISP como puerta de enlace, para llevar paquetes cuyo contenido es el estado de las gpio de salida donde se tiene un “/topic” asociado a cada una de ellas haciendo uso del protocolo TCP/IP para transportarlo hacia la capa superior.

En la Capa de Aplicación

Al Broker llegan segmentos de información que son procesados para poder ser utilizados por el protocolo MQTT, el cual se encarga de llevar los datos a la página web, con lo cual NodeRed los puede interpretar y representar por medio del Dashboard.

Finalmente, el usuario tiene la información que el sensor está entregando en tiempo real, este proceso se realiza en forma inversa para controlar a los actuadores asociados a cada “/topic” desde el Dashboard en la página web

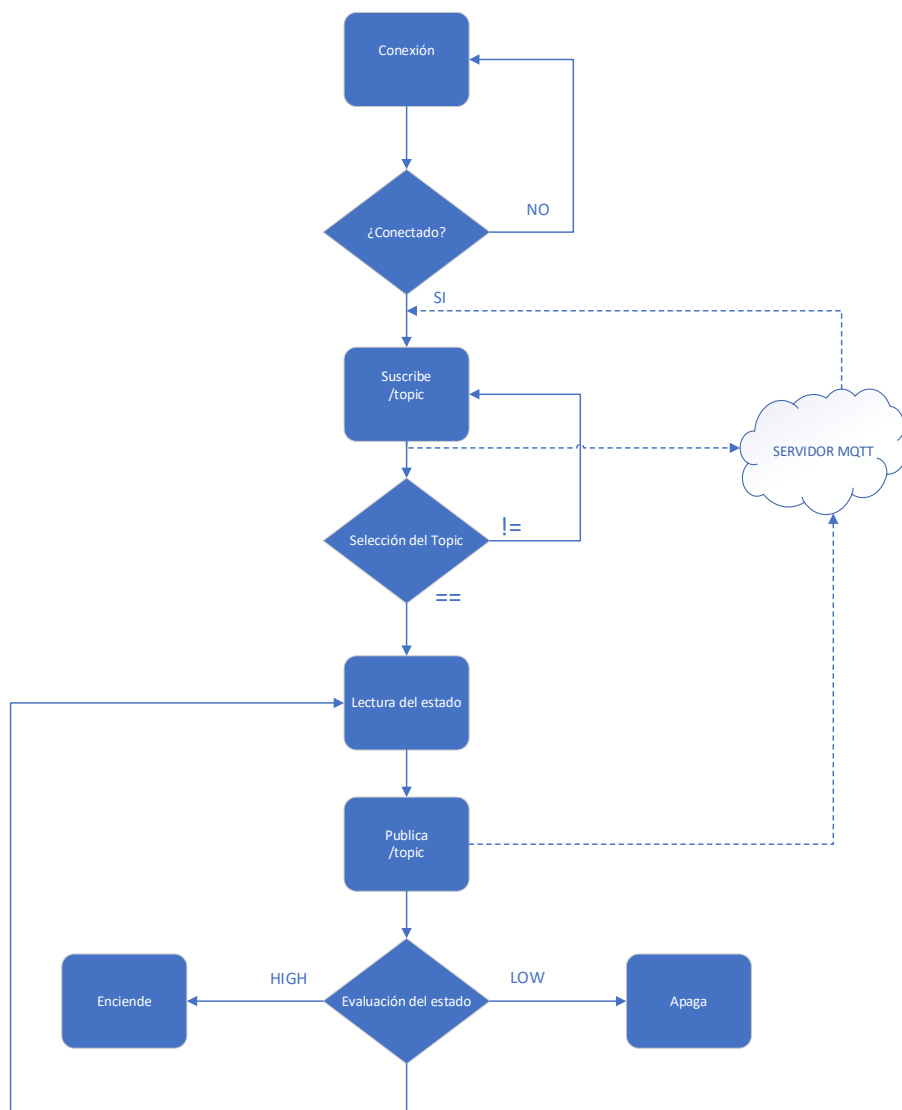


Figura 3.12. Diagrama de flujo de la comunicación con el Broker. (Elaboración propia, 2023).

3.3.2 Paso 2: Servidor y Bróker MQTT

Teniendo claro como es el proceso que se usa para el envío de información desde el sensor hasta la página web, se comienza con la parte práctica del desarrollo del proyecto. Por lo cual es necesario en primera instancia crear el servidor web que contendrá el Broker MQTT.

3.3.2.1 Instalación de Raspberry PiOS

A continuación, se describe la instalación del sistema operativo Raspberry PiOS antes conocido como Raspbian. Dicho software es una distribución de Linux modificada específicamente para ejecutarse en la plataforma Raspberry Pi.

Elementos Necesarios:

- Raspberry Pi 4 Model B memoria RAM 2GB (especificaciones completas: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>)
- Tarjeta MicroSD velocidad de lectura es de 100 MB/s y de escritura 60 MB/s. 32 GB de almacenamiento
- Conexión a Internet
- Software Raspberry Pi Imager obtenido en el siguiente enlace: <https://www.raspberrypi.com/software/>

Antes de comenzar con la instalación de Raspberry PiOS, es necesario acondicionar la placa Raspberry Pi 4 Modelo B, previamente se instalan disipadores de calor en los integrados que conforman la placa, adicionalmente para tener un mejor rendimiento de la tarjeta se instala un mini ventilador que se conecta los pines 4 y 6. En la Figura 3.13 se muestra la distribución de pines de la RBPi4B.

Posteriormente se descarga el software “Raspberry Pi Imager” directamente de la página oficial de Raspberry. El cual permite crear una imagen del sistema operativo en la memoria MicroSD. Enlace: <https://www.raspberrypi.com/>, situándonos en el apartado Software. En Figura 3.14 visualizamos el sitio web.

Al abrir la pestaña Software se puede descargar el asistente de grabación de Raspberry PiOS, seleccionando el tipo de sistema operativo con el cual se esté trabajando. En este caso se selecciona el botón “Descargar para Windows”.

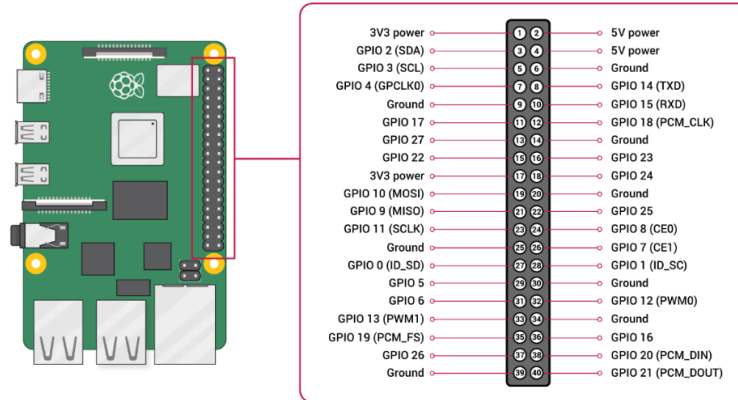


Figura 3.13. Distribución de pines RBP4B. (Fuente: raspberrypi.com).

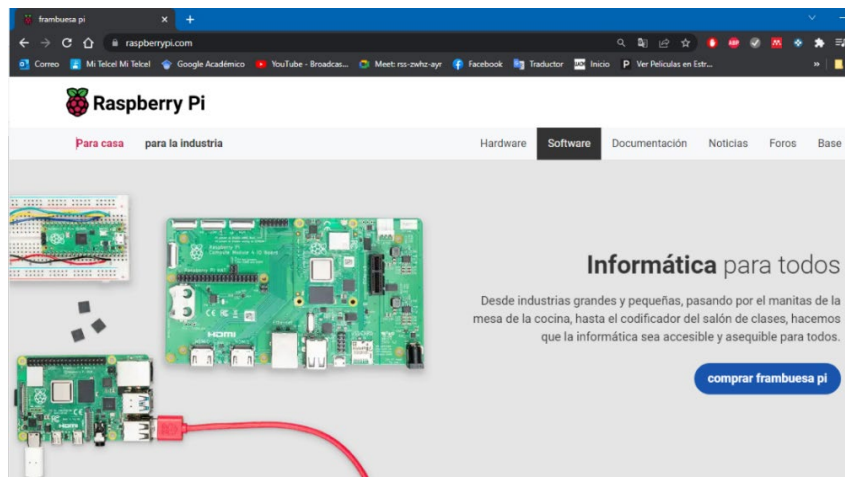


Figura 3.14. Obtención de la herramienta Raspberry Pi Imager. (Fuente: raspberrypi.com).

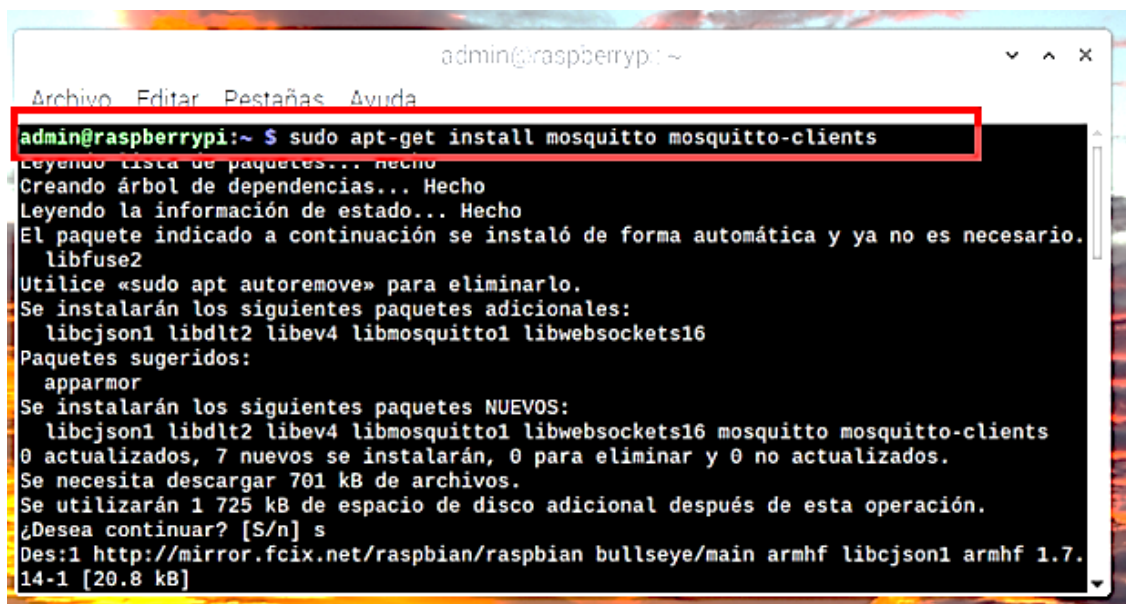
Al finalizar la instalación de la imagen del sistema operativo, se inserta la tarjeta dentro de la Raspberry y se comienza la configuración del sistema Raspberry PiOS. Se reiniciará un par de veces y se prosigue con la configuración inicial del sistema. Los primeros aspectos a configurar son: el idioma, región y la distribución del teclado que se desea usar.

Inmediatamente se configura la seguridad básica de la Raspberry, estableciendo una contraseña y un usuario para acceder al sistema.

Todo el proceso de instalación está perfectamente detallado y documentado en la página oficial de Raspberry Pi, en el siguiente enlace se puede consultar la documentación detallada a este proceso: <https://www.raspberrypi.com/documentation/computers/getting-started.html>

3.3.2.2 Servidor y Bróker MQTT

Después de Instalar el SO, se procede a instalar Eclipse Mosquitto que servirá de Broker MQTT. Este proceso se hace directamente desde la consola de comandos, como se muestra en la Figura 3.15.



```
admin@raspberrypi: ~
Archivo Editor Pestañas Ayuda
admin@raspberrypi:~ $ sudo apt-get install mosquitto mosquitto-clients
Leyendo lista de paquetes... hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
 libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
 libbson1 libdlt2 libev4 libmosquitto1 libwebsockets16
Paquetes sugeridos:
 apparmor
Se instalarán los siguientes paquetes NUEVOS:
 libbson1 libdlt2 libev4 libmosquitto1 libwebsockets16 mosquitto mosquitto-clients
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 701 kB de archivos.
Se utilizarán 1 725 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://mirror.fcix.net/raspbian/raspbian bullseye/main armhf libbson1 armhf 1.7.
14-1 [20.8 kB]
```

Figura 3.15. Instalación de Mosquito. (Elaboración propia, 2023).

Por defecto, el Broker MQTT no tiene configurado ningún tipo de autenticación o encriptación de seguridad; sin embargo, es posible modificar estos parámetros para garantizar la integridad del sistema, ya que Mosquitto permite agregar usuarios y contraseñas con diferentes permisos

según su categoría, además de agregar cifrado TTL, todo esto realizando la modificación del fichero de configuración que se encuentra en la siguiente ruta:

```
/etc/mosquitto/mosquitto.conf.
```

Al modificar el archivo mosquitto.conf se permiten las conexiones de los distintos clientes MQTT que sean necesarios. Desde la terminal de consola se introduce el comando:

```
sudo nano /etc/mosquitto.conf
```

y se agregan las siguientes líneas de código:

```
Allow_anonymous true
```

```
Listener 1883
```

3.3.2.3 Instalación de NodeRed

Para instalar NodeRed, es necesario que previamente se instale Node.js, esto se realiza desde consola con los comandos:

```
sudo su curl -sL https://deb.nodesource.com/setup_19.x | bash -
```

```
sudo apt-get install -y nodejs
```

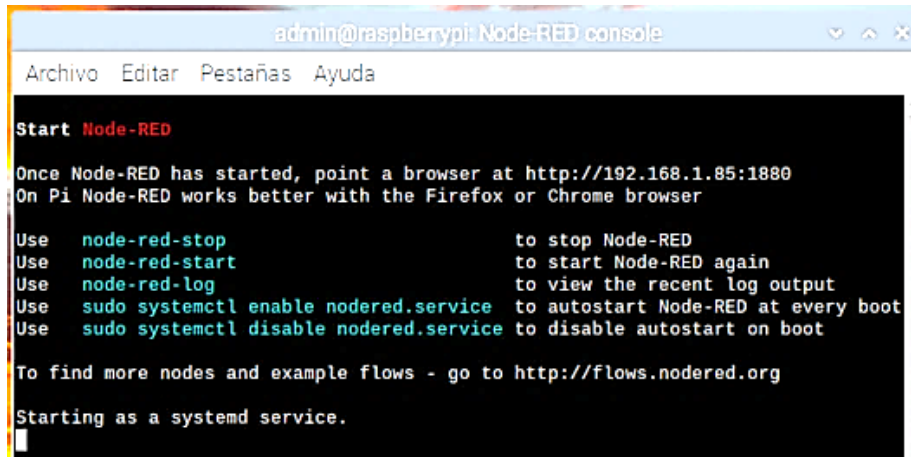
Después de instalar Node.js, se procede a instalar NodeRed con el siguiente comando:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux  
installers/master/deb/update-nodejs-and-nodered)
```

Al terminar este proceso, se configura para que inicie en automático cada que encienda la RBPi, con el comando: *node-red-start*

Se mostrará la dirección IP correspondiente donde el servidor de NodeRed se está ejecutando localmente en la red, en nuestro caso la dirección IP es 192.168.1.85:1880, tal como se muestra en la Figura 3.16.

Con esta dirección IP se accede directamente en la API de NodeRed para comenzar a trabajar con los flujos.



```
admin@raspberrypi: Node-RED console
Archivo  Editar  Pestañas  Ayuda

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.1.85:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop           to stop Node-RED
Use  node-red-start          to start Node-RED again
Use  node-red-log            to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use  sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
█
```

Figura 3.16. Ejecución de NodeRed activa. (Elaboración propia, 2023).

Después de instalar NodeRED, se deben instalar los módulos necesarios para la programación de los prototipos, directamente desde la paleta de nodos. Toda la documentación de NodeRed se encuentra en su página web [107].

3.3.3 Paso 3: Ensamblado de los dispositivos

Para programar el código de los prototipos es necesario utilizar la interfaz de programación de Arduino, ya que las tarjetas ESP32 no son hechas por Arduino, es necesario agregar las bibliotecas necesarias según corresponda al modelo, en la Figura 3.17 se ilustra este proceso.

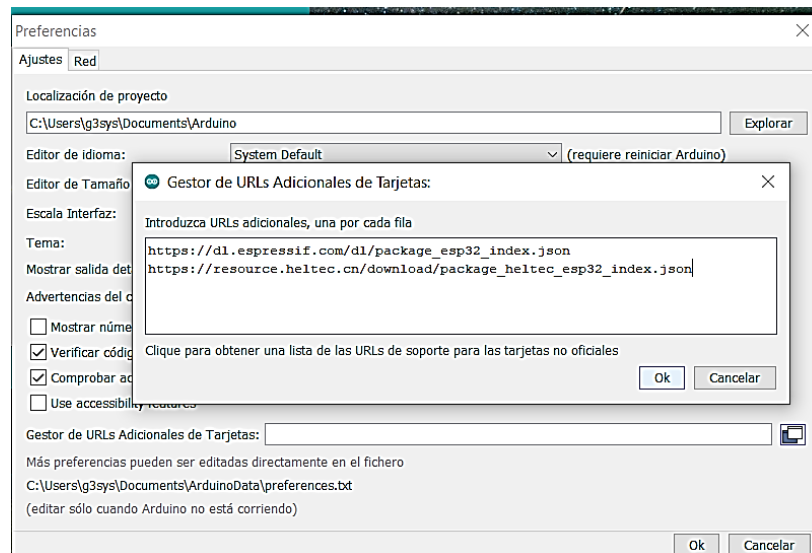


Figura 3.17. Gestor de tarjetas. (Elaboración propia, 2023).

Adicionalmente, es necesario instalar la biblioteca PubSubClient, la cual está especialmente diseñada para trabajar con el protocolo MQTT. La documentación de dicha biblioteca está disponible en el enlace: <https://www.arduino.cc/reference/en/libraries/pubsubclient/>

3.3.4 Paso 4: Configuración de la red

Para ingresar al Dashboard, el usuario deberá ingresar por medio de un navegador web, sin importar el tipo de dispositivo desde el cual se quiera conectar.

De forma local, es decir en la misma red LAN en donde se ejecuta el servidor MQTT, únicamente deberá ingresar al enlace: <http://192.168.1.82:1880/ui>

Al ingresar por primera vez, en un navegador será necesario realizar un proceso de autenticación.

Para ingresar de forma remota, es decir desde fuera de la Red LAN donde se ejecuta el servidor MQTT (Internet), es necesario publicar el sistema en la Web.

Para lograr esto debemos realizar las siguientes configuraciones:

- Abrir los puertos con los que trabaja NodeRed y Mosquitto.

- Llevar NodeRed a internet mediante un dominio de internet.
- Llevar el servidor Web de la ESP32-Cam a internet mediante un dominio de internet.

Para realizar estas configuraciones se explicará brevemente cada uno de los mecanismos y procedimientos usados.

Iniciaremos con la apertura de puertos tcp/udp necesarios.

- Broker MQTT: puerto 1883
- NodeRed: 1880
- ESP32-Cam: 80 y 81

La configuración se realizará bajo el ISP: TELMEX (“Internet Service Provider”) la cual es la empresa que brinda la conexión a internet por medio de un modem de fibra óptica.

Estas configuraciones pueden variar dependiendo el modelo de modem y el ISP con que se esté trabajando.

Se ingresa a la interfaz del modem por medio de la dirección IP 192.168.1.254.

Es necesario crear reglas de “port forwarding” (el “port forwarding” es una forma de redirigir conexiones a través del router). Al habilitar el “port forwarding”, se permite que un dispositivo remoto conecte de forma directa con nuestro propio equipo a través de un redireccionamiento que se realiza designando puertos específicos del router para establecer esta conexión. La configuración utilizada se muestra en la Figura 3.18.

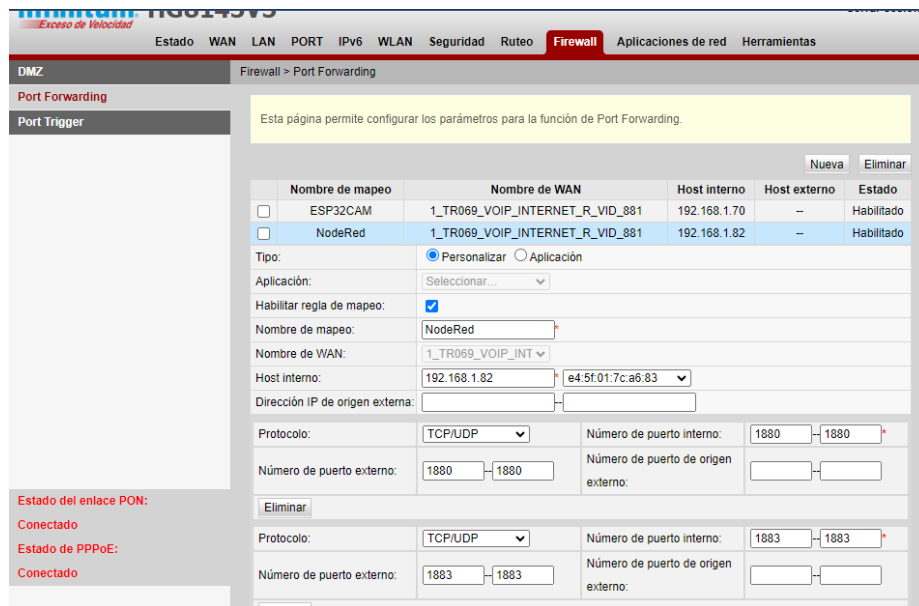


Figura 3.18. Port Forwarding NodeRed. (Elaboración propia, 2023).

Esta configuración se aplica al Broker y al Modulo ESP32Cam para poder ser alcanzados desde internet. El servicio de DDNS se realiza mediante el proveedor “No-Ip” desde su sitio web: <https://www.noip.com/es-MX> y se configuran los accesos en el router del ISP, tal como se muestra en la Figura 3.19.

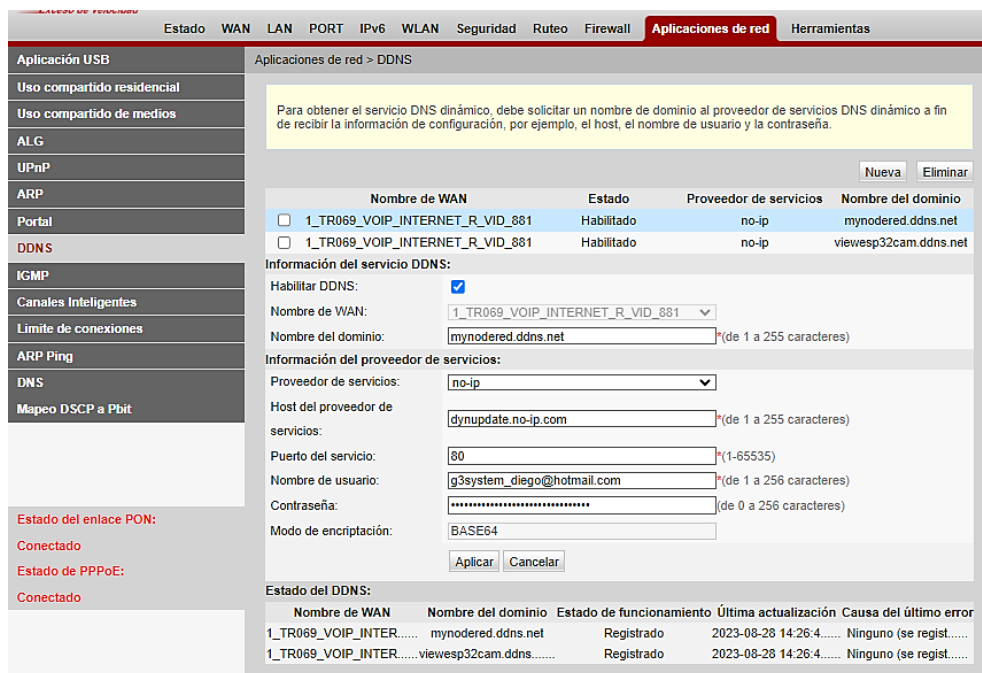


Figura 3.19. Configuración DDNS. (Elaboración propia, 2023).

3.3.5 Paso 5: Implantar los Dispositivos en el sistema

Una vez que ya se tiene configurado el servidor, la comunicación con el Broker es exitosa y ya podemos acceder al Dashboard desde internet es momento de describir el proceso de creación de cada uno de los prototipos que integran el sistema.

3.3.5.1 Control de Luminaria

Para poder controlar las luces de cualquier habitación, se utilizan focos comunes y relevadores. Cada lámpara o foco se controla mediante botones de encendido y apagado programados mediante NodeRed utilizando Raspberry como Broker-servidor MQTT.

La primera parte del código es la función `“setup_wifi()”`, la cual se encarga de hacer la conexión wifi entre la ESP32 y la red LAN que se está usando.

Como se mencionó anteriormente, debemos incluir la librería `WiFi.h` y la librería `PubSubClient.h`, las cuales permiten crear la comunicación inalámbrica y el uso del protocolo MQTT desde la ESP32.

Lo siguiente es tener un mecanismo para poder suscribirnos a los “topics” y para ello se utiliza la función `“reconnect()”`; la cual asigna un ID de forma aleatoria a la ESP32, esto para no tener conflicto a la hora de tener más placas ESP32 conectadas al servidor MQTT. Lo primero es evaluar si estamos conectados al servidor MQTT, de ser “verdadero” se crea el “topic conexión”, cuyo payload es “Conectado” para indicar que la ESP32 estableció conexión de forma exitosa con el servidor MQTT(Mosquitto).

Inmediatamente, el cliente, (la ESP32) se suscribe a los “topics” correspondientes a cada uno de los dispositivos.

En caso de que no se esté conectado, no se suscribe a ningún “topic” y se evalúa la sentencia “else” y se procede a hacer un nuevo intento de conexión al servidor MQTT en un tiempo específico determinado por un “delay” de 5 segundos.

Se escribe la función **“callback()”**, la cual se encarga de definir el tamaño y la naturaleza del **“topic”** y el **“payload”** (mensaje) para poder enviar e interpretar los datos usados para el control del encendido y apagado de cada foco.

Dentro de esta función se creó una estructura condicional mediante el uso de las sentencias **“if”** y **“else”**, donde primero se evalúa el **“topic”** y posteriormente se evalúa el valor del contenido del **“payload”**.

El valor del **“payload”** debe ser un carácter **“1”** o **“0”**, el cual es evaluado y dictamina si la salida del gpio asignado toma un valor alto (HIGH) o un valor bajo (LOW).

Esta evaluación se realiza por cada uno de los dispositivos que se desean controlar, es decir, para tres focos es necesario evaluar tres veces estas sentencias y realizar una instrucción u otra dependiendo del resultado de dicha evaluación.

La siguiente instrucción es el llamado de la función **“setup ()”**, la cual permite establecer las gpio como salidas, dichas gpio se definen en la parte inicial del código y cada una corresponde a un dispositivo a controlar.

Además evalúa los datos de la función **“setup_wifi()”**, y se pasan los parámetros del servidor MQTT y la función **“callback()”**.

Finalmente tenemos la función **“loop ()”**, en donde se evalúa si el cliente está conectado al servidor MQTT, en caso de ser verdadero se establece el mecanismo de envío de datos usando la función **“millis”** la cual es una función que se usa para medir tiempo. Esta instrucción da el tiempo en milisegundos desde que se encendió la tarjeta ESP32. Uno de los principales usos es el determinar el tiempo que ha pasado entre dos o más procesos del programa principal.

Dado que la extensión del código empleado es bastante amplia, en la sección de Anexo se detalla como poder consultarlo completamente.

El Flow de NodeRed es bastante sencillo, para controlar el estado del foco se utiliza el nodo “Switch”, el cual es unido a un nodo “MQTT Out” y a un nodo “led” que sirve para tener una referencia visual del estado del Foco, tal como se representa en la Figura 3.20.

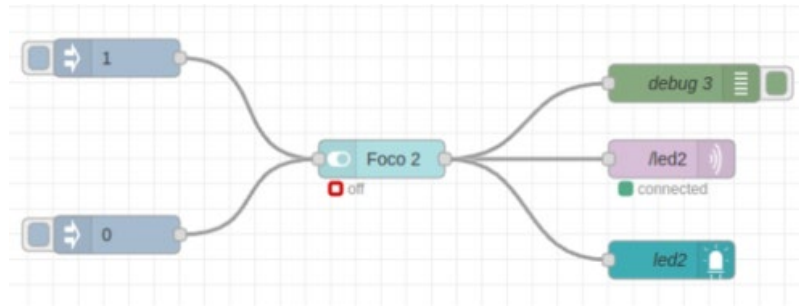


Figura 3.20.Flow para el control de Foco. (Elaboración propia, 2023).

Mediante el Dashboard se visualizan los switches pertenecientes a los focos que queremos controlar. Gracias al nodo Led, es posible visualizar el estado actual de cada foco; donde el color rojo indica que están apagados y el cambio a color verde indica que los focos están encendidos como se muestra en la Figura 3.21.

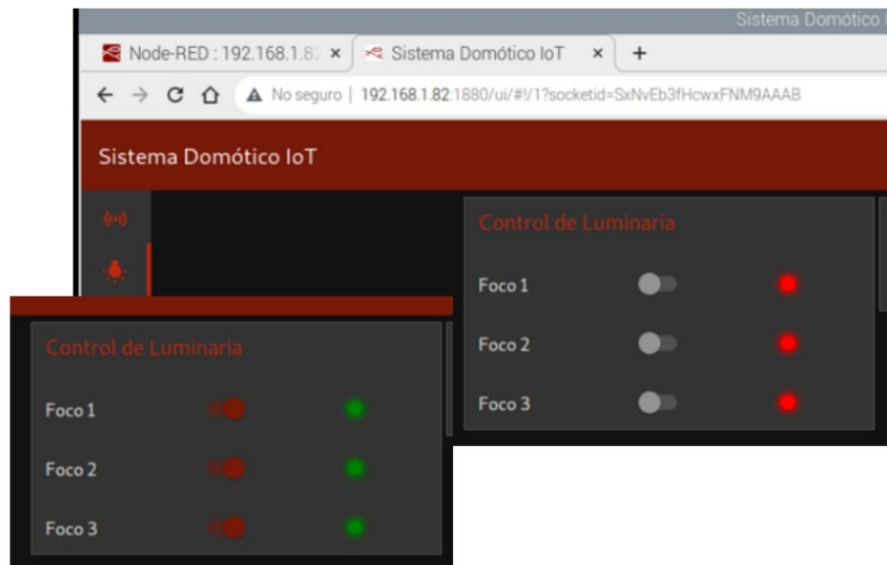


Figura 3.21.Dashboard control de Luminaria. (Elaboración propia, 2023).

Después de tener la parte del software configurada, pasamos al hardware:

Al estar controlando cargas, la señal de control proveniente del ESP32, puede no ser suficientemente estable para producir niveles lógicos aceptables usados en dispositivos TTL (Transistor–Transistor–Logic).

- Un valor lógico 0 es cualquier voltaje entre 0 y 0.8 volts.
- Un valor lógico 1 es cualquier valor entre 4 y hasta 5 volts.

Para ello usaremos el ULN2003 el cual es un arreglo de 7 drivers contenido en un circuito integrado, el cual es utilizado para conmutar cargas. El módulo relevador recibe la señal desde la salida del ULN2003, cada driver está formado por un arreglo de transistores Darlington con los cuales podemos controlar cargas cuya corriente de entrada es muy pequeña, en este caso la señal de control proveniente del ESP32 tiene un valor de 500 mA, con esto se garantiza la activación del relevador permitiendo el paso de corriente alterna (AC) que se encuentra alimentando la terminal común y cerrando la bobina con la terminal NO (normalmente abierta).

El módulo relevador utilizado es un dispositivo electromagnético cuya función es comportarse como un interruptor, que por medio de un electroimán y una bobina accionan una serie de contactos o terminales que permiten o cortan el flujo de corriente a los circuitos conectados en sus terminales.

El relevador es activado con una señal de $0 V_{CD}$, es decir; con un “cero” lógico.

Es importante mencionar que el módulo relevador que se está utilizando para el ensamblaje de los prototipos es un módulo que ya tiene integrado un transistor, una resistencia de base y un diodo que impiden el retorno de corrientes parasitas hacia la ESP32 las cuales podrían dañarla.

Además, cuenta con un led que nos indica el estado de activación del mismo y terminales tipo bloque para las terminales Común, Normalmente Abierto y Normalmente Cerrado. Todo montado sobre una PCB tal cual se muestra en la Figura 3.22.

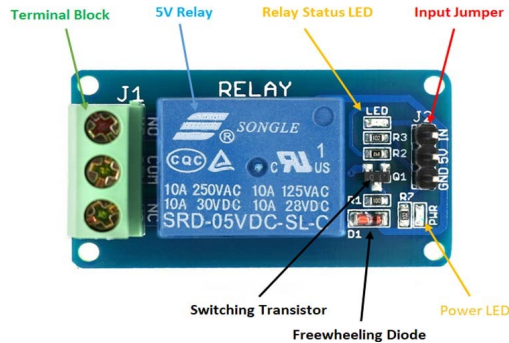


Figura 3.22. Modulo Relé. (Elaboración propia, 2023).

Una vez que se contemplan todos estos detalles en la construcción del prototipo, procedemos a realizar un esquema virtual que nos permita visualizar el comportamiento, para ello se utilizó el software “Fritzing”, el cual es un software de código abierto que nos ayuda en el diseño y creación de PCB’s, cuya documentación podemos obtener directamente en su sitio web [108]. En la Figura 3.23 se visualiza el diagrama de conexión.

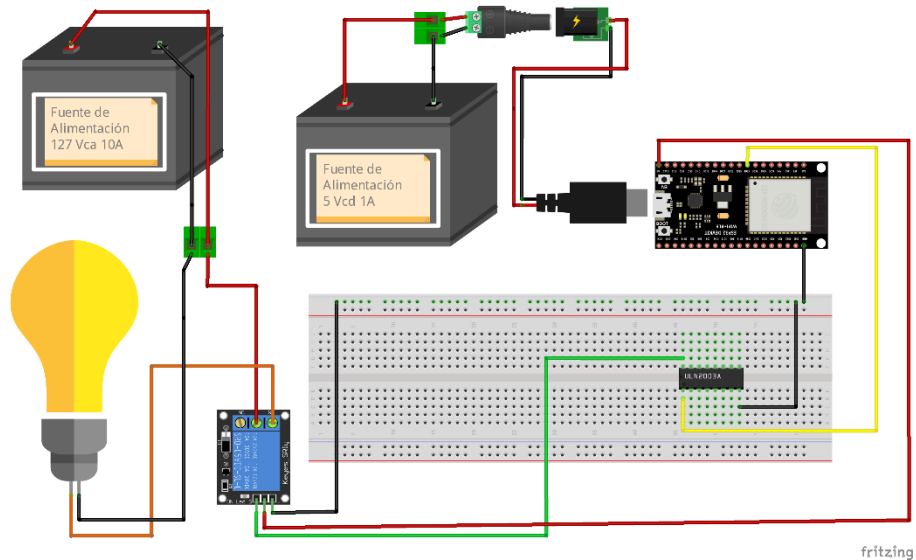


Figura 3.23. Diagrama Control de Luminaria. (Elaboración propia, 2023).

La tarjeta ESP32 es alimentada con una fuente de corriente directa de 5 volts a 1 Ampere. De la ESP32 tenemos una salida digital (cable amarillo) la cual entra en el pin 1 del ULN2003; la salida, pin 16 (cable verde) se conecta al pin de conexión IN del módulo relevador, el cual es conectado a 5 volts (cable rojo) provenientes del pin VIN de la tarjeta ESP32 al pin VIN del relé y a GND (cable negro).

La terminal Normalmente Abierto se conecta a una de las terminales del socket del foco (cable naranja), el cual actúa como retorno en la activación del mismo. La terminal Común es conectada directamente a la fase de 127 VCA (cable rojo). Por último, la terminal restante en el socket es conectada al neutro (cable negro).

Dada la complejidad que existe de simular el prototipo por la naturaleza de su funcionamiento se decidió no realizar simulaciones de ningún tipo y realizar directamente las conexiones, comprobando de forma experimental a prueba y error el funcionamiento del prototipo. Tanto en su parte de hardware como en su parte de software.

3.3.5.2 Control de cargas

La lógica empleada para la elaboración de un tomacorriente inteligente es la misma que se empleó en el desarrollo del prototipo anterior, únicamente se sustituye el socket del foco por un contacto de 127 VCA de 10 A.

Es importante mencionar que el módulo relé solo puede manejar cargas con 10 A, como máximo por lo tanto debemos hacer el cálculo de cuanta potencia es posible manejar con esta configuración.

- Del propio relevador se sabe que como máximo se puede tener una corriente máxima de 10 A
- El voltaje correspondiente es de 127 VAC

Por tanto, para conocer la potencia máxima que este arreglo puede manejar, se usa la siguiente expresión:

$$P = V * I$$

Donde:

- P es potencia expresada en Watts
- V es el voltaje expresado en Volts
- I es la corriente eléctrica expresada en Amperes

Sustituyendo valores se tiene:

$$P = 127 \text{ v} * 10 \text{ A}$$

$$P = 1270 \text{ W}$$

De tal forma que también se calcula el consumo por hora:

$$\text{Consumo} = \text{Potencia} * \text{Tiempo}$$

$$\text{Consumo} = 1270 \text{ W} * 1 \text{ h}$$

$$\text{Consumo} = 1.27 \text{ kW} * 1 \text{ h}$$

$$\text{Consumo} = 1.27 \text{ kWh}$$

De tal modo que para poder controlar cargas más grandes (con mayor consumo energético) es necesario sustituir este módulo relevador por uno que sea capaz de soportar más Amperios.

La lógica para programar el contacto corresponde también a la usada para el foco, lo único que cambia es la creación de un nuevo “topic”, el cual se le dio el nombre de “/enchufe”, tal como se muestra en la Figura 3.24

```
else if (topico == "/enchufe") {
  //digitalWrite(led3, mensaje[0]);
  if ((char)payload[0] == '1') { //si se re
    digitalWrite(toma, HIGH);
  }
  else if ((char)payload[0] == '0') { //
    digitalWrite(toma, LOW);
  }
}
```

Figura 3.24. Código tomacorriente (Elaboración propia, 2023).

La configuración de nodos en NodeRed es exactamente la misma que se empleó en la configuración del foco, el único cambio que se realizó fue el cambio del “topic” y la creación de un nuevo “Group” donde se localizara la sección de control de cargas. Visualmente, en el Dashboard se mantiene la misma esencia; el uso de un switch que te permita activar o desactivar el flujo de corriente en el prototipo, acompañado de un led indicador de estado y con el mismo “letrero” desplegable al dejar el cursor sobre el switch que te indica la acción, tal cual se muestra en la Figura 3.25

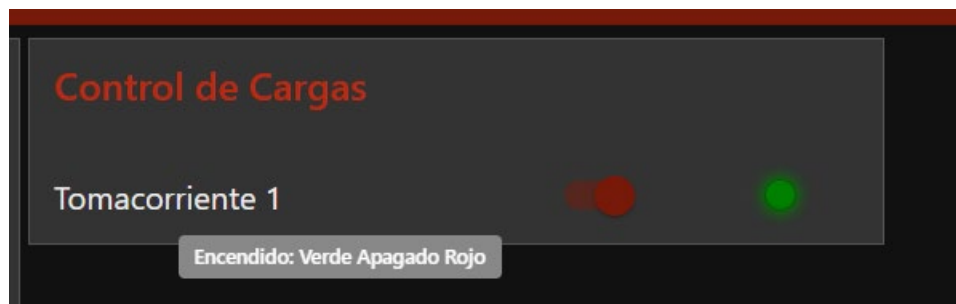


Figura 3.25. Control de cargas. (Elaboración propia, 2023).

El diagrama de conexión es prácticamente igual al utilizado en el prototipo anterior, solo se cambió el socket del foco por un tomacorriente comercial.

Dicho tomacorriente está diseñado para soportar cargas de hasta 15A, pero como se mencionó anteriormente, el diseño de este prototipo está limitado a cargas que manejen como máximo 10A.

La tarjeta ESP32 es alimentada con una fuente de corriente directa de 5 volts a 1 Ampere. Desde la ESP32 tenemos una salida digital (cable amarillo) la cual entra en el pin 1 del ULN2003; la salida, pin 16 (cable verde) se conecta al pin de conexión IN del módulo relevador, el cual es conectado a 5 volts (cable rojo) provenientes del pin VIN de la tarjeta ESP32 al pin VIN del relé y a GND (cable negro).

La terminal Normalmente Abierto se conecta a una de las terminales del tomacorriente (cable naranja), el cual actúa como retorno en la activación del mismo. La terminal Común es conectada directamente a la fase de 127 VCA (cable rojo). Por último, la terminal restante en el tomacorriente es conectada al neutro (cable negro), como se muestra en la Figura 3.26.

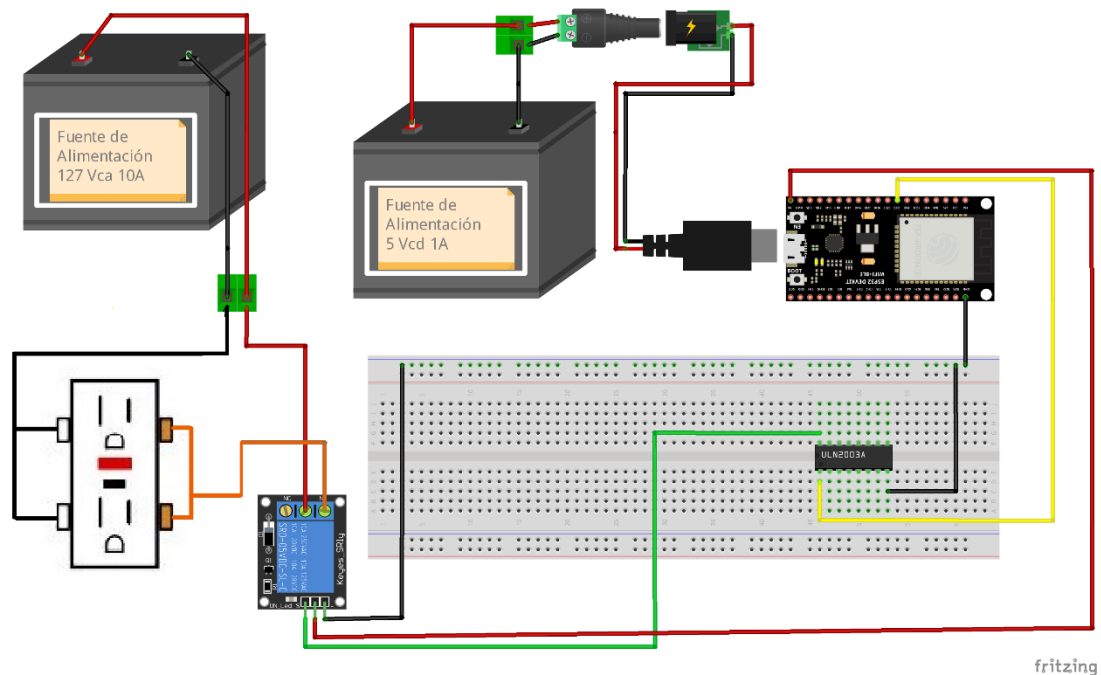


Figura 3.26. Diagrama Tomacorriente. (Elaboración propia, 2023).

3.3.5.3 Sensor de Humedad y Temperatura

Para el sensado de temperatura y humedad se usa el Sensor DHT22 el cual dispone de un sensor capacitivo para la humedad y un termistor para medir el aire a su alrededor. La salida que este sensor brinda una salida digital.

Las especificaciones de trabajo más relevante se enlistan enseguida, mismas que pueden ser consultadas en la hoja del fabricante que se adjuntara en la sesión de anexo.

- Humedad de 0-100% con 2-5% de precisión
- Temperatura de -40 a 80 °C con $\pm 0,5^{\circ}\text{C}$ Precisión
- Muestreo máximo de 0,5 Hz (una vez cada 2 segundos)

Para poder utilizar este sensor con el IDE de Arduino es necesario incluir la librería <DHT.h>. La documentación puede ser consultada directamente en la página web del desarrollador [109].

La programación para la adquisición de datos se muestra en la Figura 3.27.

```
    } else {  
    //Cliente Conectado  
    if (millis() - timerEnvioDatos > 5000) {  
        timerEnvioDatos=millis();  
  
        //-----  
        //delay(2000);  
        int temperatura=dht.readTemperature(); //guar  
        int humedad=dht.readHumidity(); //guardamos e  
        //-----  
  
        snprintf (msg, 16, "%d", temperatura); //escri  
        client.publish("/temperatura", msg); //el pri  
        Serial.begin(115200); //  
        Serial.println("Temperatura: "); //  
        Serial.print(temperatura);  
        Serial.println(" °C");  
  
        snprintf (msg, 16, "%d", humedad);  
        client.publish("/humedad", msg);  
        Serial.println("Humedad: "); //  
        Serial.print(humedad);  
        Serial.println(" %");  
    }  
    client.loop();
```

Figura 3.27. Adquisición de datos, sensor DHT22. (Elaboración propia, 2023).

La función “`millis()`” nos permite saber la cantidad de mili segundos que pasan desde que se ejecuta por primera vez el código, es decir, toma el registro de pulsos generados por el contador y los emplea para calcular el tiempo en milisegundos.

De tal forma que si la resta entre “`millis()`” y “`timmerEnvioDatos`” es menor a 5000 milisegundos, se ejecutan todas las instrucciones dentro del “`if`”.

En otras palabras, estamos condicionando a que se envíen muestras cada 5 segundos, este tiempo es controlado por la condición >5000 , pero se debe tener en cuenta que lo máximo que el sensor DHT22 puede enviar datos es cada 2000 milisegundos, por lo tanto, si disminuimos

el tiempo en el argumento del “if”, las lecturas que se obtienen del DHT22 pueden ser erróneas.

Para poder leer los datos del DHT22, creamos una matriz o buffer la cual se usa para almacenar estos datos, por ello empleamos la función “snprintf()” la cual le da formato a los datos provenientes de la gpio del ESP32. De tal forma que almacena estos datos en la variable “msg” como una cadena de caracteres (char), con un tamaño específico. Posteriormente se usa el método “read” para devolver los valores guardados en el búfer y asignarlos a la variable que corresponde.

Finalmente con “client.publish”, se crea el tópico “/temperatura” y el tópico “/humedad”. Cada tópico recibe el valor de “payload” almacenado en la variable “msg” según corresponde.

Finalmente se ejecuta “client.loop()” la cual permite al cliente procesar los mensajes, actualizar los datos de publicación y actualizar el estatus de la conexión con el servidor MQTT.

Para recibir todos los datos provenientes de la ESP32, se usa el protocolo MQTT; para ello se emplea el nodo “mqtt in”, el cual permite entablar conexión a los datos publicados por el cliente y el Broker MQTT como se muestra en la Figura 3.28.

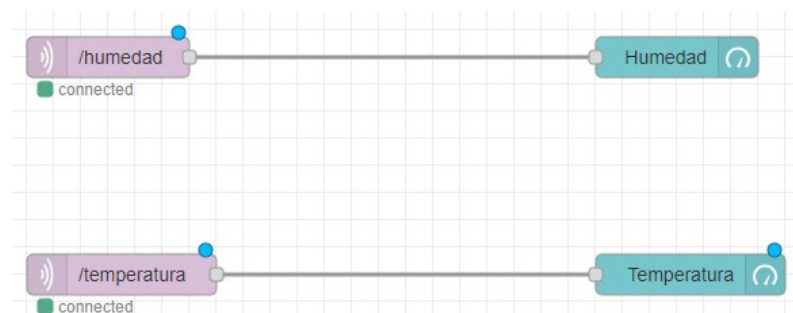


Figura 3.28. Flow para sensor de temperatura y humedad. (Elaboración propia).

Finalmente, para visualizar los datos en el Dashboard es necesario utilizar el nodo “Gauge”, el cual nos permite desplegar un gráfico para indicar los valores provenientes del ESP32.

La configuración de dicho nodo es bastante simple:

- En el apartado “Group”, debemos escribir en que parte del Dashboard queremos ubicar el indicador.
- Size; sirve para definir el tamaño del indicador en la interfaz del Dashboard.
- Label; permite darle un nombre al indicador.
- Los demás apartados son opcionales y sirven para personalizar más el indicador.
- Type; nos da a elegir el tipo de indicador a usar.

En el Dashboard de NodeRed, el usuario visualiza en tiempo real el estado del sensor DHT22 con una actualización de muestreo cada 5 segundos, en la Figura 3.29 se muestra el aspecto de los nodos en el Dashboard.

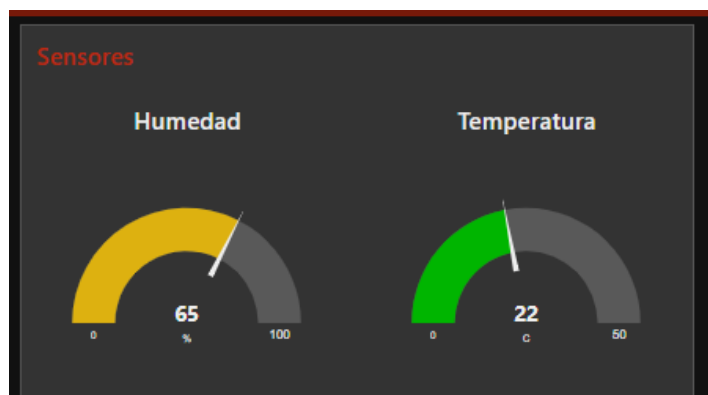


Figura 3.29. Lectura de valores del sensor DHT22. (Elaboración propia, 2023).

En la Figura 3.30 se observa el diagrama de conexión. El cable amarillo es la señal de entrada que envía los datos desde el DHT22 hacia la ESP32. El cable negro es GND y el cable rojo es VIN 5 volts.

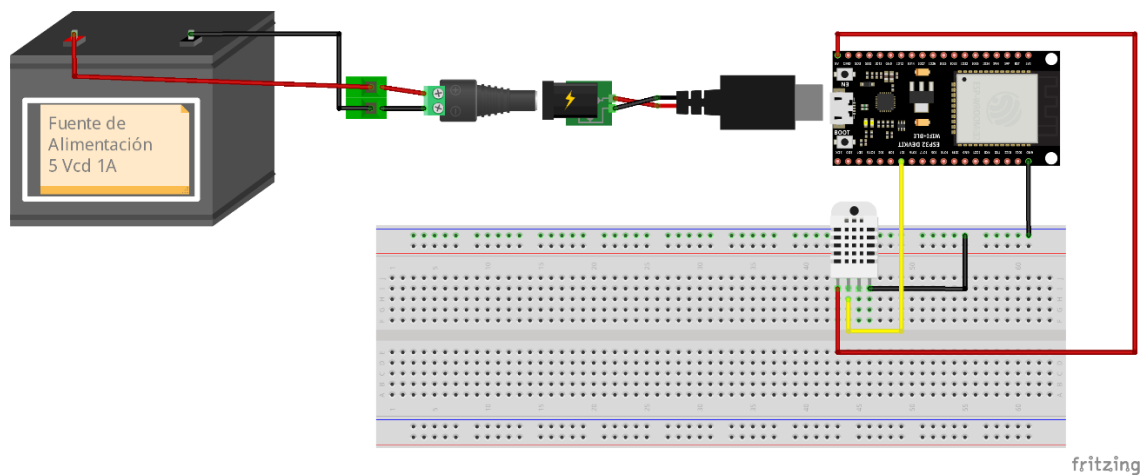


Figura 3.30. Sensor Temperatura y Humedad. (Elaboración propia, 2023).

3.3.5.4 Sensor de Gas

Para la construcción del prototipo indicador de gas, se usó el sensor MQ2; el cual es sensible a GLP, i-butano, propano, metano, alcohol, hidrogeno y humo. La resistencia del sensor cambia de acuerdo a la concentración del gas en el aire, para nuestro caso, el sensor se programó para mandar un carácter de tipo “char 1” cuando se detecta presencia de gas LP y un carácter “0”, de tipo “char” cuando no se detecte presencia de gas.

Cuando el sensor MQ2 detecta la presencia de gas, envía un valor de 4095 a su resistencia. Previamente se declara un pin gpio de la ESP32 como entrada y se procede a leer el valor.

El valor proveniente del sensor MQ2 se almacena en una variable llamada gas, a su vez el valor de la variable gas es asignado a la variable de tipo entero llamada “gasVa1or”.

La variable “gasVa1or” se usa para ser comparado con el valor de la resistencia del sensor y se procede a ser comparada dentro de un condicional if. Si el valor de la resistencia es idéntico al valor contenido en “gasVa1or”, se ejecuta todo lo que este contenido dentro de la condicional if y una nueva variable llamada “gaz” recibe un valor tipo “char” de “1” y se envía el mensaje “gas detectado” por el puerto serial. En caso contrario se recibe un “0”. En la Figura 3.31 se muestra lo antes mencionado.

```

int gasValor = analogRead(gas);

char fuego;
char gaz;

if (fireValor == 4095){
  fuego = 0;
} else {
  fuego = 1;
  Serial.println("Fuego detectado: ");
}

if (gasValor == 4095){
  gaz = 0;
} else {
  gaz = 1;
  Serial.println("Gas detectado: ");
}

```

Figura 3.31. Programación Sensor MQ2. (Elaboración propia, 2023).

Posteriormente mediante el uso de la función “snprintf()” se formatea se almacena y formatea el valor de la variable “gaz”, para que pueda ser publicado con el “topic /gas” mediante la función “client.publish()” tal como se muestra en la Figura 3.32

```

snprintf (msg, 16, "%d",gaz ); //escrib
client.publish("/gas", msg); //el prime
Serial.begin (115200);
Serial.println(gaz); //Envío del valor

```

Figura 3.32. Publicación del tópico /gas. (Elaboración propia, 2023).

El mecanismo de conexión a la red LAN, al servidor Broker MQTT y las funciones de reconexión y asignación de “ID-client” son exactamente las mismas que se han descrito anteriormente, por lo cual ya no se entra en detalles de su configuración.

Para visualizar el indicador de presencia de gas se emplea un nodo “Gauge” dentro del Dashboard de NodeRed y para acceder a los datos enviados se usa el nodo MQTT in. Tal cual se describió en la sección anterior del sensor DHT22, la configuración no cambia mucho, lo único es la asignación del “topic /gas” al nodo MQTT in y la configuración del nodo Gauge que ahora solo recibirá valores de cero cuando no detecta presencia de gas y valores de 1 cuando

condicional `if` y una nueva variable llamada `fuego` recibe un valor tipo “char” de “1” y se envía el mensaje “flama detectada” por el puerto serial. En caso contrario se recibe un “0”.

Se usa la misma forma de publicar los datos que en el sensor de gas, mediante la función “`snprintf()`” y “`client.publish()`”. Solo cambia el “topic”, ahora se usa “/fire”.

Para la visualización del estatus de detección de flama, se emplea el mismo mecanismo que con la detección de gas, se recibe un carácter “1” si existe presencia de flama o fuego, por otro lado, se recibe un carácter “0” si no hay presencia de flama, todo esto por medio del nodo MQTT in que ya se describió anteriormente.

Para la detección de flama, además de poder detectar la presencia de fuego es necesario crear una manera de alertar al usuario cuando este evento suceda, por ello es que se adiciona el envío de notificaciones vía correo electrónico.

El esquema de conexión para el sensor de flama se muestra en la Figura 3.34, donde el cable amarillo es la entrada de datos desde el sensor hacia la ESP32.

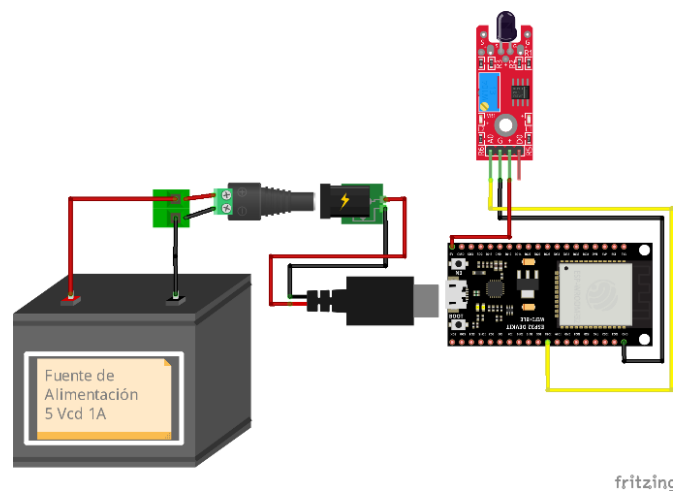


Figura 3.34. Detección de flama. (Elaboración propia, 2023).

Además del nodo MQTT in y el nodo gauge, ahora se utilizan nodos extras para poder enviar las notificaciones, como se muestra en la Figura 3.35.

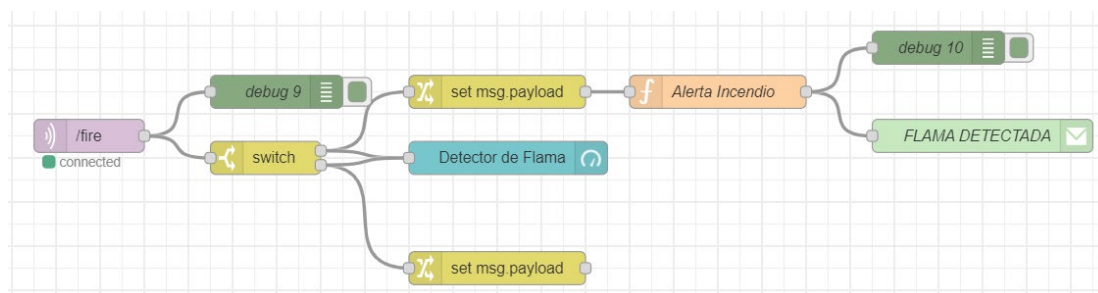


Figura 3.35.Flow Detección de flama. (Elaboración propia, 2023).

El nodo email, el cual es el encargado de enviar el correo electrónico a una dirección email específica. La configuración del nodo email es la siguiente:

- To: Se debe colocar la dirección email del destinatario.
- Server: Se coloca la dirección del servidor SMTP de Google. SMTP son las siglas de “Simple Mail Transfer Protocol”, un protocolo de correo estándar de Internet dedicado al intercambio de correo electrónico.
- Port: Es el puerto TCP/UDP donde se ejecuta el Servidor SMTP de Google.
- Auth Type: Es la forma de autenticarse con el servidor, en este caso se usa la autenticación básica.
- Userid: Aquí se coloca la dirección de correo electrónico del remitente.
- Password: Aquí se coloca la contraseña del remitente, más adelante se establecerá como debe ser esta contraseña.
- TLS option: Agrega un certificado TLS al servidor. El cual proporciona seguridad de la capa de transporte (en inglés: “Transport Layer Security”) ya que es un protocolo criptográfico, que proporciona comunicaciones seguras por una red, comúnmente Internet.

De tal forma que cuando el sensor detecta presencia de flama, se envía un correo electrónico, en la Figura 3.36 se muestra el formato de dicho correo.

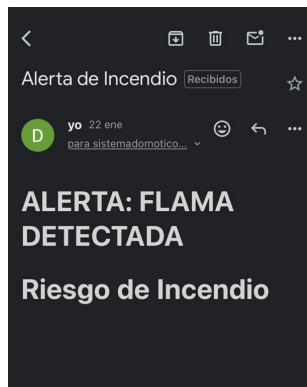


Figura 3.36. Correo Alarma de Incendio. (Elaboracion propia, 2023).

3.3.5.6 Sensor de Ventana

Para la construcción del sensor de ventana se emplea un interruptor magnético de tipo switch que se activa con la presencia de un campo magnético.

Se compone de dos laminas ferromagnéticas hechas de Níquel (Ni) y Hierro (Fe), las cuales están selladas herméticamente en una capsula de vidrio. Estas laminas dejan un pequeño espacio entre ellas, mientras que son rodeadas de un gas inerte que permanece sellado en la capsula de vidrio.

Para programar este sensor de ventana, se declaran dos gpio del ESP32, uno será para recibir la lectura proveniente del interruptor electromagnético y el otro para colocar un led indicador del estado de activación del sensor.

El mecanismo y configuración para el envío de notificaciones vía correo electrónico es exactamente igual al descrito en la sección anterior.

El diagrama de conexión para este prototipo es el siguiente: Tenemos una señal de entrada en la gpio 13, la cual se configuró como entrada y además se activó la resistencia de "pullup" o bien se puede usar una resistencia conectada en la gpio 13 y una de las terminales del sensor electromagnético (cable verde). Después tenemos un led conectado a la gpio 14, la cual toma un valor en alto (el led enciende) cuando la ventana se abre, cuando la gpio se

encuentra en estado bajo quiere decir que la ventana se ha cerrado (el led se apaga). Tal como se muestra en la Figura 3.37.

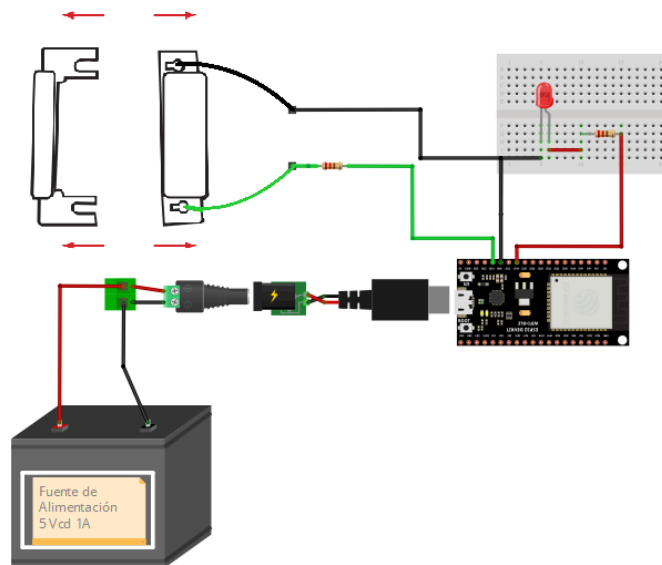


Figura 3.37. Sensor de Ventana. (Elaboración propia, 2023).

3.3.5.7 Control de Acceso: Cerradura

Para la cerradura se optó por utilizar una cerradura eléctrica de 12v compuesta por un solenoide, el cual es un electroimán que consta de una bobina de alambre de cobre con una armadura (un lingote de metal) en el medio. Cuando se energiza la bobina, el centro de metal se mueve en la bobina. Esto hace que el solenoide sea capaz de tirar de un extremo y mover el pestillo que abre o cierra un cerrojo.

La cerradura eléctrica utilizada tiene las siguientes características:

- Voltaje: 12 V
- Corriente: 0.8 A
- Potencia: 9.6 W

La lógica de programación para la cerradura es muy similar a lo utilizado en prototipos anteriores, se asigna un topic: “/openlook”; se usa el condicional if y se evalúa el contenido

del payload, si el payload == 1 entonces se pone en alto la gpio de la ESP32 lo cual abre la cerradura, si el payload == 0, entonces se pone en valor bajo, lo cual se traduce en que la cerradura permanece cerrada.

La cerradura no solo se abre y cierra, además de esto se adicionó el sistema de envío de notificaciones vía correo electrónico y además se sumó el envío de una fotografía tomada desde una ESP32-CAM cada vez que la cerradura sea abierta. En esta sección solo se explicará la configuración de nodos usados para la cerradura, más adelante se detalla el funcionamiento del uso de la ESP32-CAM y como se envía la fotografía por correo electrónico.

Para controlar la activación del solenoide, se sigue usando un módulo relevador y un driver (ULN2003) para lograr hacer la conmutación de la señal proveniente de la tarjeta ESP32.

La señal de control (cable amarillo) entra al ULN2003, de allí se conecta al módulo relevador (cable verde). El pin común del relevador se conecta a 12VCD, el terminal NA (normalmente abierto) va conectado a una de las terminales de la cerradura (cable naranja), finalmente el otro terminal de la cerradura es conectado a GND de la fuente de 12VCD. En la Figura 3.38 se ilustra las conexiones.

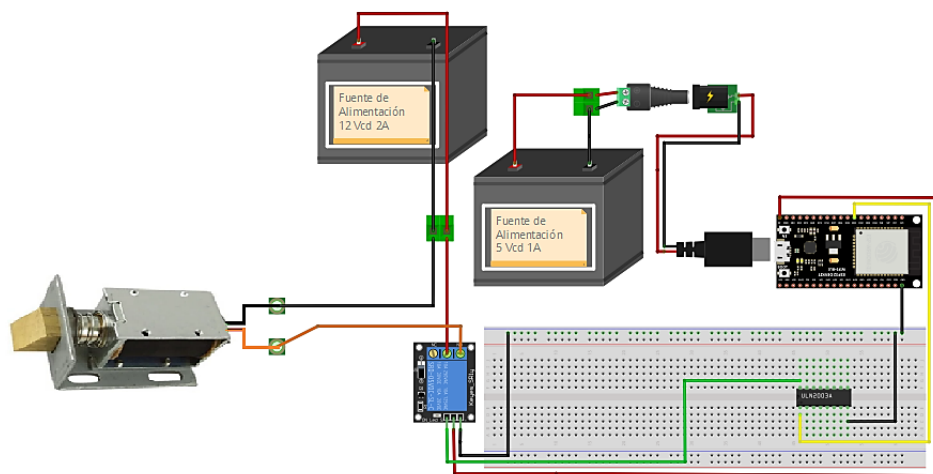


Figura 3.38. Diagrama Control de Acceso: Cerradura. (Elaboración propia, 2023).

3.3.5.8 ESP32-CAM: Envío de Imágenes y Streaming de Video

Para la construcción de este prototipo se emplea el uso de la ESP32-CAM AI-Thinker, la cual integra todas las prestaciones de la plataforma ESP32 y adiciona una cámara OV2640, le permiten hacer “streaming” de vídeo e imágenes y servirlos a la red creando un servidor local en el mismo chip, la posibilidad de añadir una tarjeta microSD de hasta 4 GB, más la opción de antena Wi-Fi externa cierran las características más importantes de este dispositivo. Para programar el ESP32-CAM es necesario usar un programador FTDI (“Future Technology Devices International Limited”) para conectarlo a la computadora. Luego debemos agregar la familia de tarjetas ESP32 desde el Gestor de tarjetas del Arduino IDE que fue descrito anteriormente.

Para utilizar este módulo, se debe seleccionar en el gestor de tarjetas, el Código para levantar el servidor local de video se puede consultar directamente en el IDE de Arduino o en la página del fabricante.

Una vez que se carga el código en la ESP32-Cam mediante el uso del programador FTDI, podemos visualizar en el monitor serie la información de donde está corriendo el servidor Web.

- El servidor web está trabajando sobre el puerto 80 tcp/udp
- Podemos visualizar el “streaming” de video en el puerto 81 tcp/udp
- La dirección IP asignada es: 192.168.1.70

La configuración de los nodos usados es bastante extensa, por ello se puede consultar el Flow completo en el área de Anexo. Finalmente, en el Dashboard la cámara puede visualizar video en formato VGA o QVGA en tiempo real, desde dentro y fuera del hogar por medio de internet. En la figura 3.39 se muestra el Dashboard.

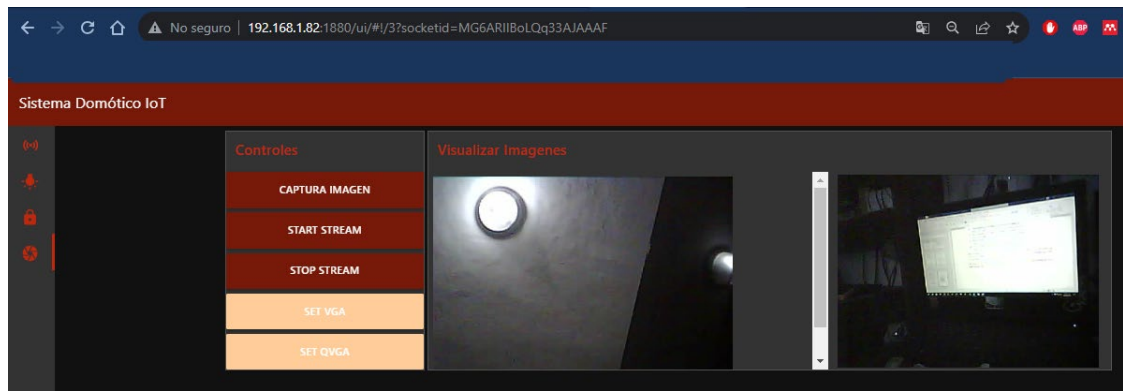


Figura 3.39. Interfaz de Visualización de la ESP32-Cam. (Elaboración propia, 2023).

Finalmente se integran todos los prototipos para conformar el sistema domótico tal cual se muestra en la Figura 3.40.

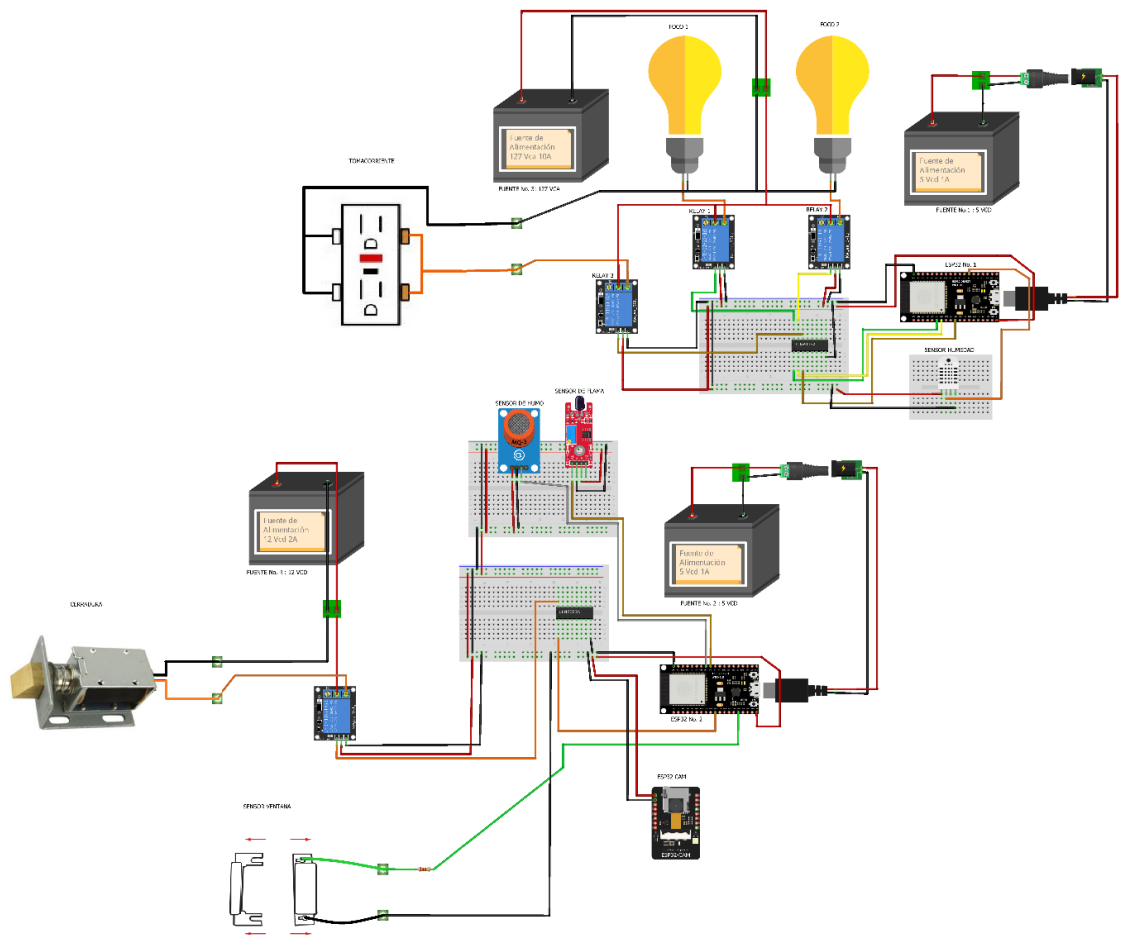


Figura 3.40. Prototipos del Sistema. (Elaboración propia, 2023).

3.4 Fase 4. Evaluación y mejoramiento del sistema domótico

3.4.1 Paso 1: Evaluación de la Interoperabilidad

Sin duda la interoperabilidad en un sistema Domótico con integración IoT es parte fundamental en su implementación y éxito comercial. De la encuesta realizada se identificó que el mercado IoT es bastante extenso, existe gran número de fabricantes y por ello el integrar todos los dispositivos en un ecosistema único, resulta ser una tarea bastante compleja.

Sumado a la ausencia de normas y estándares de interoperabilidad que obliguen a estos fabricantes a mantener protocolos de comunicación que permitan llevar esta tarea, se puede afirmar que encontrar un sistema interoperable al 100% es prácticamente imposible hasta el momento.

En el caso del sistema propuesto, la interoperabilidad es un tema de vital relevancia, ya que uno de los objetivos es poder tener un sistema interoperable, escalable y de fácil acceso.

Gracias al uso de NodeRed, esta tarea se simplifica de gran manera.

Es justo por esta característica que el usuario final tiene la facilidad de agregar dispositivos IoT de diferentes fabricantes, por medio de plataformas en la nube como “AWS” o “Home Assistant” si así lo requieren.

3.4.1.1 NodeRed con Amazon Alexa

Como se observó en el desarrollo del sistema, NodeRed contiene una paleta de nodos en la cual se encuentran algunos que hacen posible la comunicación con Amazon Alexa.

Los Flows de Node-RED pueden controlar muchos dispositivos domésticos diferentes como bombillas “Belkin WeMo” y “Philips Hue” o cualquier dispositivo y servicio artesanal o casero [110].

La documentación completa de estos nodos la podemos consultar directamente desde la biblioteca de desarrollo de NodeRed:

<https://flows.nodered.org/node/node-red-contrib-alexa-home-skill>

3.4.1.2 NodeRed con Home Assistant

Home Assistant es un completo sistema operativo de código abierto, que permite integrar cientos de marcas de dispositivos domóticos, y miles de dispositivos, y todo ello de forma simultánea para tener el control total de la domótica en nuestro hogar. Una vez que hemos añadido todos los dispositivos al sistema (se llama integraciones), podemos realizar diferentes automatizaciones para que realicen ciertas acciones que nosotros podemos configurar en detalle.

Home Assistant no solamente nos permite unificar de manera centralizada todos los sensores, actuadores y dispositivos de domótica que tengamos en nuestro hogar, con una gestión local y una interfaz gráfica de usuario realmente intuitiva.

Para poder utilizar Home Assistant con NodeRed, debemos instalar los nodos correspondientes directamente desde la paleta de nodos o desde la consola de comandos:

```
npm node-red-contrib-home-assistant-websocket
```

La documentación completa de estos nodos la podemos consultar directamente desde la biblioteca de desarrollo de NodeRed:

<https://flows.nodered.org/node/node-red-contrib-home-assistant-websocket>

3.4.1.3 NodeRed con Google Assistant

Google Assistant o Asistente de Google, es un asistente virtual creado por Google, el asistente de Google puede ser utilizado en varios sitios. Por ejemplo, vas a poder encontrarlo en un teléfono móvil Android, o incluso en iOS dentro de la aplicación de Google. También está integrado en otros tipos de dispositivos, desde televisores a altavoces inteligentes, incluyendo auriculares, y todos los dispositivos conectados a Internet en los que veas que en la caja hay una insignia con el logo del asistente.

Al igual que Alexa, es posible utilizarlo para controlar dispositivos de IoT, basta con instalar los nodos necesarios desde la paleta de nodos o por medio de la consola de comandos:

```
npm install node-red-contrib-google-smarthome
```

Se pueden usar los nodos para controlar sus dispositivos domésticos inteligentes a través del Asistente de Google o la aplicación Google Home.

Esto se hace implementando un proveedor de hogar inteligente que se ejecuta en su propio host. Por lo tanto, no tiene que depender de ningún servicio de terceros (aparte de Google) que pueda desconectarse, cobrarle o filtrar sus datos.

La documentación completa de estos nodos se puede consultar directamente desde la biblioteca de desarrollo de NodeRed, en el siguiente enlace:

<https://flows.nodered.org/node/node-red-contrib-google-smarthome>

Gracias a la robustez de NodeRed y la facilidad con que cualquier persona puede desarrollar nodos dentro de la plataforma, es que el tema de interoperabilidad encuentra una alternativa ante la ausencia de estándares como ya se comentó en secciones anteriores. Por tanto, la interoperabilidad del sistema propuesto no está comprometida, por el contrario, la escalabilidad de este sistema dependerá del usuario y del desarrollo existente.

3.4.2 Paso 2: Diagnóstico de la eficiencia energética

El costo de la energía eléctrica en Ciudad de México está regulado por la Comisión Federal de Electricidad, la cual establece diferentes tarifas y costos con fecha de enero 2024 esta información se visualiza en la Tabla 1.12.

Tabla 1.12. Tarifas CFE mes de enero. Fuente: CFE, 2023.

Tarifa	Temperatura media en verano (°C)	Costo en \$ Consumo Básico kWh	Costo en \$ Consumo Intermedio kWh	Costo en \$ Consumo Excedente kWh	DAC kWh/mes
1	<25	1.015	1.239	3.62	250
1A	25	0.909	1.051	3.62	300
1B	28	0.909	1.051	3.62	400
1C	30	0.909	1.051 -1.355	3.62	850

1D	31	0.909	1.051 - 1.355	3.62	1000
1E	32	0.755	0.943 - 1.224	3.62	2000
1F	33	0.755	0.943 - 2.290	3.62	2500

Para consultar las tarifas completas es posible hacerlo directamente en la página web de CFE:

<https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/TarifasCRECasa/Casa.aspx>

Por lo que para calcular el costo de la implementación del sistema es necesario hacer el cálculo correspondiente según el consumo del mismo.

Para poder hacer una estimación del consumo energético total del sistema se utiliza un medidor de voltaje como el que se muestra en la Figura 3.41.



Figura 3.41. Monitor de energía. (Fuente: Amazon, 2023).

Se utilizó el monitor de energía para registrar las variaciones de potencia y consumo mostradas por el sistema. En la Tabla 1.13 se registran los datos de cada muestra bajo ciertas condiciones en un lapso de 60 minutos (Se hicieron dos periodos de muestreo, cada uno de 60 minutos).

Tabla 1.13. Medición de parámetros eléctricos del sistema. (Elaboración propia, 2023).

CONDICIONES	LECTURAS	LECTURAS PROMEDIO
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (Todo apagado: Solo funcionan sensores y las 2 ESP32).	Corriente: 0.74 -0.84 A	Corriente: 0.79 A
	Power Factor: 0.48-0.60	Power Factor: 0.54
	Potencia: 4.7-5.3 W	Potencia: 5 W

	Voltaje: 119.5-120.01 V 60HZ	Voltaje: 119.8 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE(Todo apagado: Solo funcionan sensores y las 2 ESP32, tomacorriente, sp32cam encendida sin transmitir ni capturar).	Corriente: 0.95-0.105 A	Corriente: 0.5275 A
	Power Factor: 0.48-0.56	Power Factor: 0.52
	Potencia: 6.4-6.7 W	Potencia: 6.55 W
	Voltaje: 119.5-120.01 V 60HZ	Voltaje: 119.8 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (2 focos encendidos, funcionan sensores y 2 las ESP32, tomacorriente, sp32cam encendida sin transmitir ni capturar).	Corriente: 0.423-0.434 A	Corriente: 0.4285 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 36.8-42.1 W	Potencia: 39.45 W
	Voltaje: 119.5-120.01 V 60HZ	Voltaje: 119.8 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa capturando imagen (foto), tomacorriente.	Corriente: 0.402-0.413 A	Corriente: 0.4075 A
	Power Factor: 0.80-0.81	Power Factor: 0.805
	Potencia: 38.3-42.4 W	Potencia: 40.35 W
	Voltaje: 119.5-120.01 V 60HZ	Voltaje: 119.8 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo Stream video, tomacorriente).	Corriente: 0.402-0.413 A	Corriente: 0.4075 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 38.3-38.6 W	Potencia: 38.45W
	Voltaje: 118.7-120.0 V 60HZ	Voltaje: 119.35 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo	Corriente: 0.561-0.572 A	Corriente: 0.5665 A
	Power Factor: 0.77-0.79	Power Factor: 0.78

Stream video, tomacorriente, cerradura activa(abierta).	Potencia: 38.7-52.6 W	Potencia: 45.65 W
	Voltaje: 118.7-120.0 V 60HZ	Voltaje: 119.35 V
MEDICIONES SIN PANTALLA, SIN TECLADO Y SIN MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo Stream video, cerradura activa(abierta), sensor ventana abierta, tomacorriente).	Corriente: 0.529-0.532 A	Corriente: 0.5305 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 49.5-52.7 W	Potencia: 51.1 W
	Voltaje: 118.7-120.0 V 60HZ	Voltaje: 119.35 V
Potencia maxima: 52.7W Potencia minima: 4.1 W	Eficiencia promedio x hora: 0.03236428 kWh	Costo x hora: \$0.0705540
MEDICIONES CON PANTALLA, TECLADO Y MOUSE (Todo apagado: Solo funcionan sensores y las 2 ESP32).	Corriente: 0.84-0.95 A	Corriente: 0.895 A
	Power Factor: 0.48-0.55	Power Factor: 0.515
	Potencia: 5-5.3 W	Potencia: 5.15 W
	Voltaje: 118.9-121.3 V 60HZ	Voltaje: 120.1 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE (Todo apagado: Solo funcionan sensores y las 2 ESP32, tomacorriente, sp32cam sin transmitir ni capturar).	Corriente: 0.84-0.95 A	Corriente: 0.895 A
	Power Factor: 0.48-0.55	Power Factor: 0.515
	Potencia: 5-5.3 W	Potencia: 5.15 W
	Voltaje: 118.9-121.3 V 60HZ	Voltaje: 120.1 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE(2 focos encendidos, funcionan sensores y las 2 ESP32, tomacorriente, sp32cam sin transmitir ni capturar).	Corriente: 0.434-0.44 A	Corriente: 0.439 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 39.2-39.8 W	Potencia: 39.5 W

	Voltaje: 119.5-120.5 V 60HZ	Voltaje: 120 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE(2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa capturando imagen, tomarcorriente).	Corriente: 0.423-0.550 A	Corriente: 0.4865 A
	Power Factor: 0.79-0.80	Power Factor: 0.795
	Potencia: 38.9-39.2 W	Potencia: 39.05 W
	Voltaje: 119.5-120.1 V 60HZ	Voltaje: 119.8 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo Stream video, tomarcorriente).	Corriente: 0.423-0.555 A	Corriente: 0.489 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 38.9-39.5 W	Potencia: 39.2 W
	Voltaje: 118.7-120 V 60HZ	Voltaje: 119.35 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE(2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo Stream video, tomarcorriente, cerradura activa(abierta)).	Corriente: 0.561-0.572 A	Corriente: 0.5665 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 52.6-53.6 W	Potencia: 53.1 W
	Voltaje: 118.7-120 V 60HZ	Voltaje: 119.35 V
MEDICIONES CON PANTALLA, TECLADO Y MOUSE (2 focos encendidos, funcionan sensores y las 2 ESP32, esp32 cam activa, haciendo Stream video, cerradura activa(abierta), sensor ventana abierta, tomarcorriente).	Corriente: 0.550-0.575 A	Corriente: 0.5625 A
	Power Factor: 0.77-0.79	Power Factor: 0.78
	Potencia: 49.8-53.6 W	Potencia: 51.7 W
	Voltaje: 118.9-120 V 60HZ	Voltaje: 119.45 V
Potencia maxima: 53.6 W Potencia minima: 4.7 W	Eficiencia promedio x hora: 0.03326428 kWh	Costo x hora: \$0.07251614

Mediante el uso de MATLAB se graficaron estas muestras y se compararon entre ellas, tal como se muestra en la Figura 3.42.

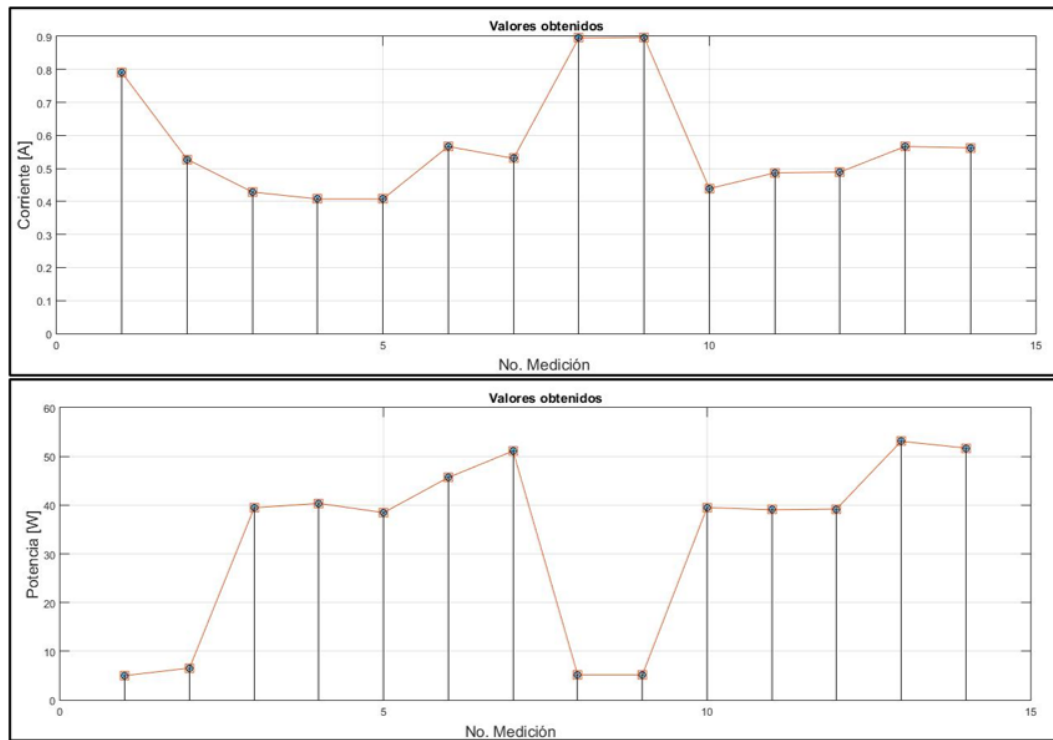


Figura 3.42. Graficas de corriente y potencia. (Elaboración propia, 2023).

De dichas graficas se encuentra que el mayor consumo energético se tiene cuando la cerradura se activa, tanto la corriente como la potencia del sistema se incrementan significativamente respecto a las otras mediciones. Con esto se observa una tendencia de consumo en el sistema. Cabe mencionar, que se registra un mayor consumo eléctrico al conectar un monitor, mouse y teclado.

3.4.3 Paso 3: Evaluación de Costos

3.4.3.1 Gasto Eléctrico

Evaluar los costos del sistema es una tarea donde se considera factores como el costo energético y el costo de los consumibles.

Lo primero a evaluar será el consumo eléctrico. Con base en la información obtenida en secciones anteriores podemos calcular el costo de operación del sistema.

Con la información del costo del kWh en ciudad de México de la Tabla 1.12 se obtiene el costo promedio de la energía, usando el costo máximo y mínimo:

$$\frac{\$0.755 + \$3.62}{2} = \$2.1875$$

De acuerdo con los datos de la Tabla 1.13, la eficiencia x hora, no teniendo conectado un monitor, un mouse y un teclado fue de fue de 0.032364287 kWh. Multiplicando este número por el costo del kWh para la ciudad de México se tiene:

$$\$2.1875 \times 0.03236428 \text{ kWh} = \$0.07079686 \text{ kWh}$$

Multiplicando por 24, se obtiene el costo por día: \$1.699124

Multiplicando el costo por 365, se calcula el **costo anual: \$620.1805 MXN**

Este gasto se compara con algunos otros gastos contenidos en la Tabla 1.14 que muy comunes entre la población, con el fin de evidenciar que la implementación del sistema es muy accesible económicamente.

Tabla 1.14. Costos gastos comunes x mes, precios a fecha de 23/01/24. (Fuente: Internet,2024).

Servicio/Producto	Costo Mensual	Costo Anual
Netflix Básico	\$139	\$1668
Amazon Prime Video	\$99	\$899
Spotify Individual	\$129	\$1548
Plan Telcel Básico	\$249	\$2988
GYM Básico	\$300	\$3600
Consumo de 1 refresco 600 ml diario.	\$19x7x4 = \$532	\$6384
Consumo de 1 cigarro diario.	\$5x7x4 = \$140	\$1680

Comparando el costo por la operación del sistema con alguno de estos gastos, podemos notar que el costo es menos de la mitad que algunos de ellos.

Además, el poder energizar o des energizar un electrodoméstico de forma remota ayuda a combatir fugas de consumo eléctrico.

Aparatos como televisores, decodificadores, cargadores celulares entre otros, tienen consumos en modo de reposo, lo cual produce un gasto extra.

De igual manera, poder encender o apagar las luces en nuestro hogar representa un ahorro energético, de esta manera si salimos de casa y olvidamos apagarlas, el sistema ayuda a resolver esta situación y no generar gastos innecesarios.

Por todos estos motivos, es que el uso del sistema domótico con integración IoT, resulta como una alternativa viable y con un costo de operación muy bajo.

3.4.3.2 Costo Inicial (Consumibles)

Para poder calcular el costo para desarrollar e implementar el sistema se toman en cuenta diferentes factores.

- Cambio de precios en componentes eléctricos y electrónicos
- Costos mensuales por servicios indispensables (Internet, Electricidad)

En la Tabla 1.15 se registraron los precios de cada uno de los componentes utilizados en el desarrollo del sistema, sin contar los gastos realizados para realizar la maqueta. Los precios fueron tomados al día 23/01/24.

Tabla 1.15. Costo Elementos del Sistema. (Fuente: Tiendas online especializadas, 2024)

Elemento	Precio Individual \$	Cantidad	Precio \$
Raspberry Pi 4B	1309	1	1309
Tarjeta ESP32	115	2	130
Tarjeta ES32CAM	280	1	280
Soket porta Lampara	19	2	38
Foco Led	34	2	68
Placa 1 Contacto	75	1	75
Caja chalupa	19	1	19
Cerradura	136	1	136
Sensor Contacto magnético	89	1	89
Sensor MQ2	59	1	59
Sensor DHT22	95	1	95
Sensor de Flama	45	1	45
Relevador 5VCC	38	5	152
Cargador 12 Vcc	94	1	94

Cargador 5 Vcc	49	2	98
Cable Dupont	64	1	64
Transistor ULN2003	56	1	56
Tablilla Fenólica	14	2	28
TOTAL			\$2835

De tal forma que, para conocer el costo Total, de la implementación del sistema se deben tomar en cuenta lo siguiente:

- Costo Internet
- Costo Electricidad
- Costo de los consumibles

El costo del servicio de internet lo podemos omitir para hacer el cálculo del costo del sistema, ya que como se observó en la encuesta, se infiere que es un servicio con el que ya cuenta la vivienda donde se va a implementar.

En la tabla 1.16 se muestra la comparación de costos de los dispositivos considerando dispositivos económicos populares en el mercado y que son capaces de interactuar en un solo ecosistema.

Tabla 1.16. Comparación de Dispositivos IoT. (Fuente: Steren y Amazon, 2024).

Dispositivo x Función	Dispositivo IoT Comercial Ecosistema Amazon Alexa	Costo	Dispositivo IoT Comercial Ecosistema Google Assistant	Costo
Control de carga: Foco	Foco Inteligente Wi-Fi Steren	\$249	Foco Inteligente Wi-Fi tp-Link	\$200
Control de Carga: Tomacorriente	Contacto Wi-Fi Steren	\$199	Enchufe Wi-Fi tp-link	\$247
Cerradura	Cerradura digital Wi-Fi Steren	\$2390	Cerradura Electrónica Inteligente (China)	\$1599
Sensor Ventana	Sensor Wi-fi Para Puerta O Ventana Steren	\$299	TP-Link Tapo T110 Sensor de contacto	\$499
Detector de Humo y Flama	Detector de Fugas de Gas	\$698	Monitor inteligente de humedad de	\$499

	LP, Alarma, Wi-Fi, Tuya Smart)		temperatura WiFi: TUYA	
Detector Temperatura y Humedad	Sensor Wi-Fi* de temperatura y humedad con pantalla digital	\$499	TP-Link Tapo T315 Sensor de Temperatura y Humedad	\$569
	TOTAL	\$4344	TOTAL	\$3613

Aunque en el mercado existen opciones más económicas, pero no son totalmente integrables en un solo ecosistema, lo cual es la finalidad de este Sistema Domótico, por ello es que se descartan esos dispositivos.

Con la información de las tablas 1.15 y 1.16, se sostiene que el sistema propuesto en esta tesis; ofrece una solución IoT bajo un mismo ecosistema, que logra interoperabilidad y ofrece una interfaz universal para el control de todos los dispositivos que lo integran, además resulta económicamente más accesible que adquirir distintos dispositivos IoT existentes en el mercado.

3.4.4 Paso 4: Paso 4: Identificación de Áreas de Oportunidad

3.4.4.1 Seguridad en el Protocolo MQTT

Uno de los pasos más importantes en el desarrollo del sistema es la identificación de puntos críticos en su funcionamiento y seguridad.

Dentro de la documentación de NodeRed, se recomienda no exponer el servicio a internet, esto se debe a que por defecto la comunicación entre NodeRed y el Broker no tiene ningún método de encriptación que proteja el tráfico de datos.

Para comprobar dicha información emplearemos la herramienta MQTT Explorer, el cual es un software grafico que nos permite monitorear en tiempo real los tópicos que se publican en nuestro servidor. Paralelamente emplearemos el software de análisis de redes Wireshark, el cual nos permite analizar paquete por paquete las comunicaciones en cada interfaz de nuestra red LAN.

Ambas herramientas están disponibles para su descarga y se puede consultar su documentación en sus respectivas páginas web:

- MQTT Explorer: <http://mqtt-explorer.com/>
- Wireshark: <https://www.wireshark.org/>

Para empezar este análisis de seguridad, empezaremos por enviar datos desde NodeRed al cliente ESP32. Para ello se realizó un “flow” el cual permite encender y apagar leds conectados a los gpio de ESP32 mediante el uso del “nodo Switch”.

Posteriormente, en NodeRed inyectamos manualmente el contenido del mensaje; mediante el uso del “nodo Inject”.

Para hacer la prueba de seguridad al enviar datos estaremos inyectando la frase: “sin_seguridad” a nuestro switch que controla el led1.

Al mismo tiempo este dato puede ser visualizado tanto en el monitor serie de Arduino IDE como en el monitor de debug de NodeRED. En la Figura 3.43 visualizamos el último mensaje enviado en el topic “/led1” el cual contiene la frase: “sin_seguridad”.



Figura 3.43. Prueba seguridad MQTT. (Elaboración propia, 2023).

Para visualizar la carga del protocolo MQTT empleamos MQTT Explorer, el cual nos permite visualizar en una interfaz gráfica, los topic que están siendo publicados y suscritos por nuestros

clientes en tiempo real. Basta con configurar los datos de nuestro servidor el Broker Mosquitto que corre en la Raspberry Pi con la IP: 192.168.1.85 en el puerto 1883.

Una vez realizada esta configuración podemos visualizar los “topic” existentes en ese momento y visualizar su contenido, como se muestra en la Figura 3.44.

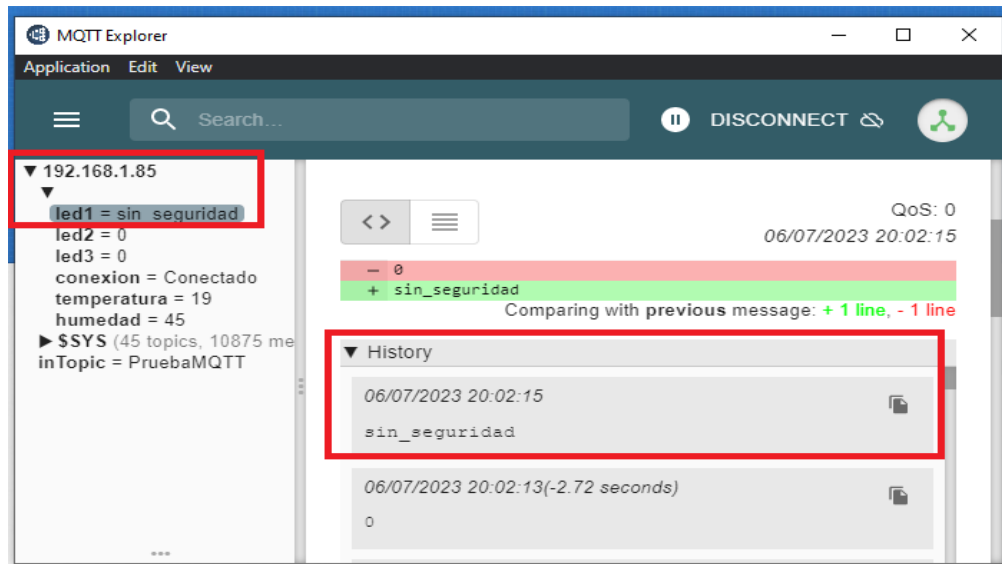


Figura 3.44. Análisis MQTT con MQTT Explorer. (Elaboración propia, 2023).

Se observó que en las Figuras 3.42 y 3.43, en el topic “/led1” se envió correctamente la frase “sin_seguridad”, con lo cual se puede asegurar que tanto el cliente ESP32, NodeRed y el Broker Mosquitto están conectados de forma correcta y la comunicación entre ellos no presenta problemas de conectividad o pérdida de datos.

Sin embargo, ahora corresponde realizar un análisis de seguridad al enviar estos datos por internet, ya que, al estar expuesto nuestro servidor en una conexión remota, algún individuo externo podría capturar esta información y comprometer la seguridad de nuestro sistema.

Para ello se empleó la herramienta Wireshark que permite analizar los protocolos de red implicados en dicha comunicación. Para este análisis se filtraron los protocolos, únicamente analizó el protocolo MQTT.

Al analizar el paquete que contiene el último mensaje publicado en el topic “/led1” se encontró un problema de seguridad grave, ya que el contenido del mensaje o “payload” se transmite en texto plano, es decir no está cifrado.

Esto representa un problema que compromete la seguridad del sistema, ya que cualquier persona podría manipular los datos y afectar el funcionamiento de los clientes y el propio servidor al no poseer ningún tipo de encriptación que proteja de algún ataque.

En la Figura 3.45 se analiza detalladamente el contenido del paquete que contiene dicha información y constatar que la frase “sin_seguridad” es enviada en texto plano y se especifica el “topic” por el cual fue enviado dicho mensaje.

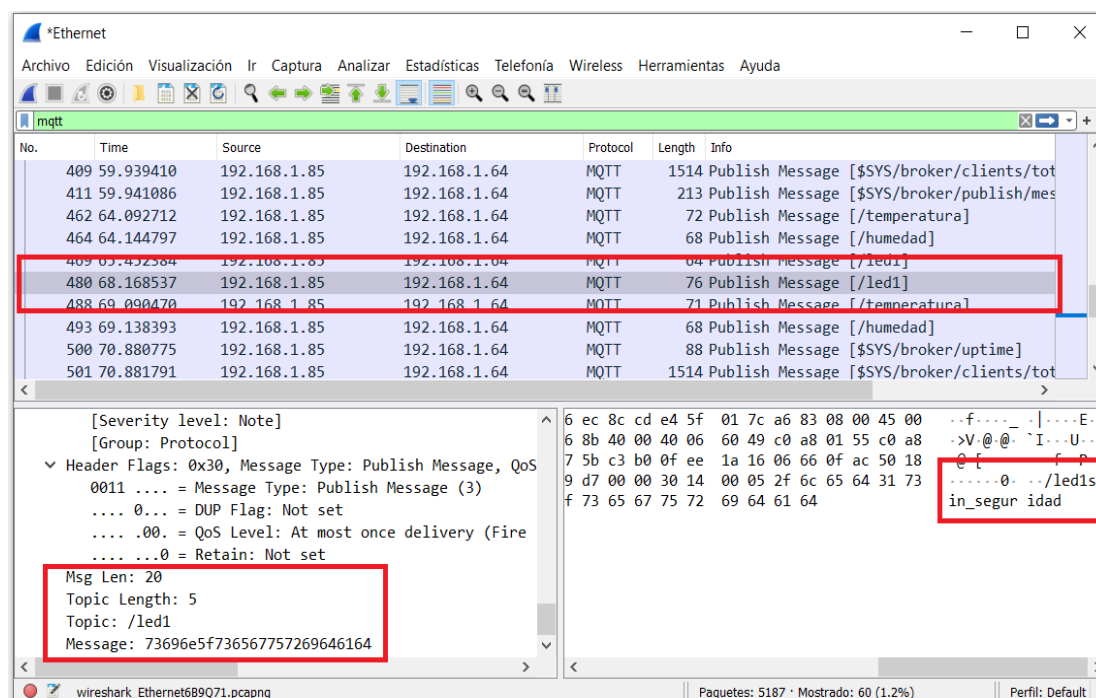


Figura 3.45. Análisis paquetes MQTT mediante Wireshark. (Elaboración propia, 2023).

Otro punto a considerar es que, con esto, un atacante podría vulnerar nuestra red LAN ya que conocería la IP del servidor y los puertos por los cuales se comunica como se muestra en la

Figura 3.46; con ello podría inyectar paquetes maliciosos, redirigir el tráfico o simplemente tirar nuestro servidor con un ataque de denegación de servicio (DoS).

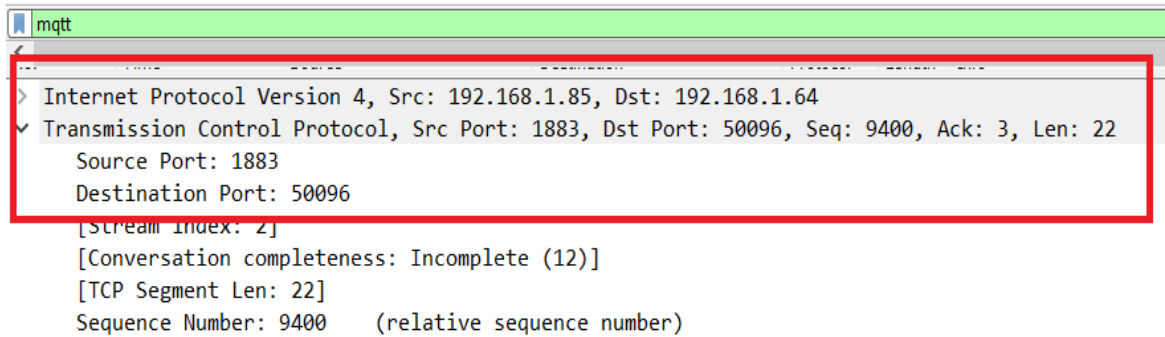


Figura 3.46. Análisis MQTT, mediante Wireshark. (Elaboración propia, 2023).

Por ello, antes de exponer el servidor local a Internet es conveniente implementar medidas de seguridad que nos permitan brindar protección tanto al servidor como a nuestros clientes MQTT.

Afortunadamente, Mosquitto proporciona soporte TLS para conexiones de red cifradas y autenticación, además cambiamos el puerto de comunicación al 8883, el cual es un puerto cifrado y gracias a esto dejaremos de enviar texto plano.

3.4.5 Paso 5: Mejoramiento

Para poder representar el sistema completo se realiza un diagrama de flujo que nos permite ubicar mejor cada una de sus partes e identificar la existencia de aspectos a mejorar. Figura 3.47.

Con azul, se describe el proceso de intercambio de información desde la capa de percepción rumbo a la capa de aplicación. El proceso inverso está marcado con las flechas de color rojo.

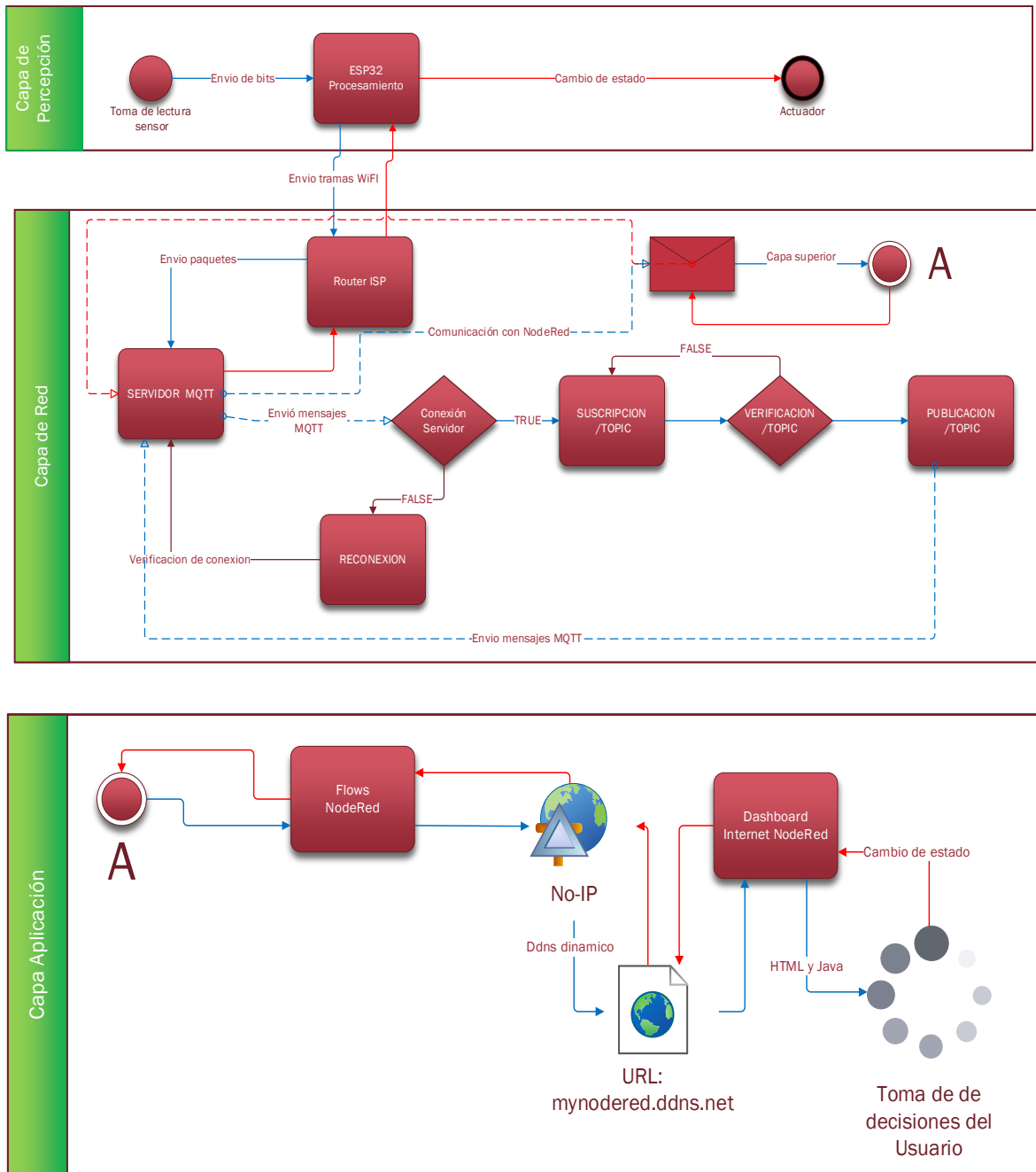


Figura 3.47. Mapa de procesos del sistema. (Elaboración propia, 2023).

Al ubicar cada elemento del sistema, es más sencillo identificar la existencia de problemas en su funcionamiento.

En este sentido, pudimos identificar algunos problemas de hardware.

- El primero radica en que el buffer de la ESP32CAM, llega a saturarse en ocasiones, esto derivado de la propia programación del servidor para la cámara.
- El segundo es el envío de Falsos positivos en la alarma de detección de humo del sensor MQ, esto se debe a que el sensor necesita hacer un proceso de calibración (se calienta), detectamos que este proceso le toma alrededor de 12 minutos, tiempo en el que envía una alerta de detección de humo. Para esto se optó por utilizar un “nodo trigger” que permita retardar el envío de la alerta en el primer encendido del sistema.

A nivel software se modificaron algunos parámetros de los sensores de flama y humo, para ajustar la sensibilidad con que estos perciben la información del medio ambiente.

También se añadieron nodos independientes para la captura de imágenes por medio de la ESP32-Cam, para cuando se activen o desactiven ciertas condiciones en el sensor ventana y en la apertura y cierre de la cerradura.

En cuanto al uso del servidor y del Broker, no se presentaron anomalías que afecten el desempeño del sistema.

Dentro de los parámetros a mejorar dentro del sistema, se consideraría el diseño de PCB's en donde se pueda montar las placas ESP32 y la fabricación de gabinetes individuales para poder ubicarlos en las habitaciones.

También se considera para trabajos futuros, el cambio de adaptadores de corriente, por el uso de baterías de Litio, esto con la finalidad de disminuir el cableado eléctrico necesario para alimentar las ESP32.

Quizá también se puede mejorar la información proveniente de los sensores, utilizando otros de mejor calidad, aunque esto representaría un aumento del costo de los insumos.

Toda la programación, tanto de las tarjetas ESP32, ESP32-Cam y la configuración de nodos usados por NodeRed se puede encontrar en el apartado de Anexos.

Construcción de Maqueta

La maqueta es una herramienta utilizada en ingeniería para representar un proyecto antes de ser implementado el sitio, por ello se decidió, construir un prototipo del tablero donde se instalaría el sistema.

Para la construcción de esta maqueta se empleó los siguientes materiales.

- Madera MDF 5mm
- Pegamento epóxico
- Clavos
- Bisagras metálicas
- Pestillos de cerradura
- Cajas plásticas transparente 9x6x3 cm
- Tornillos diferentes medidas
- Taladro
- Barra multicontacto
- Silicon
- Cable de cobre calibre 16
- Cable DuPont
- Papel tapiz autoadherible

Lo primero fue crear una estructura donde fijar cada componente del sistema, para ello se decidió usar madera MDF de 5mm cuyo costo es bastante accesible y fácil de cortar con láser.

- a) Se cortaron 8 rectángulos de 5x50 cm
- b) Se cortaron 2 cuadrados de 50x50 cm
- c) Se unieron las piezas para confeccionar dos charolas.
- d) Una vez que se tienen ambas charolas, se cubrieron con el vinil autoadherible para tener una mejor presentación visual.

- e) Lo siguiente fue utilizar cada charola como cara de una caja y unir las por medio de bisagras metálicas y utilizar pestillos de cerradura para evitar que la caja se abra.
- f) Lo siguiente fue distribuir los elementos para determinar la zona en que deberían quedar fijados como se muestra en la Figura 3.48.



Figura 3.48. Construcción de maqueta. (Elaboración propia, 2023).

- g) Posteriormente utilizando las cajas plásticas como gabinetes se fijaron los componentes con tornillos.
- h) Después de tener todos los componentes fijos, se realizaron las conexiones eléctricas, las cuales quedan ocultas dentro del tablero. Figura 3.49.



Figura 3.49. Distribución de prototipos. (Elaboración propia, 2023).

- i) El paso siguiente fue realizar las conexiones eléctricas entre todos los elementos. Para esto se usó cable calibre 16 para conectar los relevadores con los focos y la toma de corriente, se utilizó cable DuPont para las conexiones entre los componentes

electrónicos. En este punto, la creación de PCB's, en trabajos futuros ayudara a gestionar mejor el cableado

- j) Posteriormente, energizamos el tablero para comprobar su funcionamiento. Como se muestra en la Figura 3.50.



Figura 3.50. Prototipo Sistema domótico. (Elaboración propia, 2023).

El proceso de armado de la maqueta fue una tarea extra en el desarrollo del sistema, por lo que los gastos generados para ello no se contemplaron en el costo del desarrollo del mismo.

Disponer de una maqueta permite tener un mecanismo didáctico, para poder interactuar con el sistema en tiempo real. Figura 3.51.



Figura 3.51. Prototipo Sistema domótico. (Elaboración propia, 2023).

Conclusiones

Según los resultados de la encuesta, solo población de entre 20 y 40 años considera importante contar con un sistema que les permita automatizar tareas en su hogar.

Dentro de esta población que, si considera necesario tener un sistema que les permita monitorear y controlar sus hogares, la mayoría coincide en que aspectos como la seguridad, la prevención de accidentes y el ahorro energético, son los temas más interesantes dentro de sus necesidades.

De esta manera es que se planteó el sistema Domótico con integración IoT, como una alternativa viable para satisfacer estas necesidades antes mencionadas, cuyo costo debe ser accesible y no sobrepasar un costo promedio de implementación de \$7500 MXN.

El costo del sistema resulto ser mucho menor a estos \$7500, por lo cual consideramos es una opción muy accesible para el público.

Después de un análisis teórico de las tecnologías disponibles para su implementación, se determina que el uso de un protocolo ligero, confiable y eficiente como el protocolo MQTT, resulta ideal para desarrollar dispositivos IoT de bajo consumo, ya que es uno de los protocolos más utilizados en el desarrollo del internet de las cosas a nivel mundial.

Combinar el protocolo MQTT, con herramientas de desarrollo como lo es NodeRed, nos permite tener una plataforma para el Internet de las Cosas en la cual, problemas conocidos como lo la interoperabilidad no representan grandes inconvenientes. Es precisamente la robustez de NodeRed la que nos permite interactuar con protocolos y dispositivos de distintos fabricantes saltando estas limitaciones de interconexión.

Si bien, en Cyberseguridad no hay sistemas seguros, el sistema propuesto permite al usuario controlar su hogar de forma segura sin exponer sus datos personales a recopilación por parte de terceros, el sistema puede utilizar métodos de cifrado y certificados de seguridad siempre y cuando sean configurados previamente.

En cuanto a consumo energético, el sistema brinda la posibilidad al usuario de controlar remotamente sus electrodomésticos y luminaria, con lo cual se evitan gastos innecesarios al permanecer encendidos cuando no se están utilizando.

Precisamente poder encender o apagar diferentes dispositivos en el hogar, permiten simular presencia, con lo cual podemos disuadir a los delincuentes de ingresar a los hogares. Bajo esta línea de protección del hogar, poder tener un sistema de alertas en tiempo real que te avise de eventualidades en el hogar, refuerza la seguridad de la misma.

Estas notificaciones sirven no solo para evitar la delincuencia, también nos permiten monitorear la calidad del entorno físico del hogar, gracias a la detección de fugas es posible tener medidas de prevención de incendios.

Finalmente, agregamos confort a nuestro hogar teniendo mecanismos que nos ayuden a realizar tareas en el hogar aun cuando nos encontramos fuera.

Poder controlar el clima de los hogares remotamente sin duda resulta bastante confortable.

Finalmente, retomando la encuesta, el poder controlar el sistema mediante un Smartphone, resulta bastante atractivo, ya que tal cual arrojó la encuesta, es el dispositivo predilecto por la población mexicana.

Por todo lo anterior, concluimos que se cumplen los objetivos propuestos en esta tesis, dejando abierta la posibilidad de generar mejoras al sistema, gracias a su escalabilidad y bajo costo de desarrollo.

Si bien se planteó como un sistema orientado a los hogares de una población específica, el uso de Internet de las Cosas lo hace atractivo para poder llevarlo a otras áreas.

Bibliografía

- [1] L. G. Corona Ramírez, G. S. Abarca Jiménez, y J. Mares Carreño, *Sensores y Actuadores*, Primera ed. Ciudad de México, 2014.
- [2] Real Academia Española, “Diccionario de la lengua Española”. Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.rae.es/>
- [3] j. López, “Estudio de rendimiento de asistentes virtuales de voz en condiciones ruidosas”, Universidad Autónoma de Madrid, España, 2020. Consultado: el 1 de abril de 2024. [En línea]. Disponible en: https://repositorio.uam.es/bitstream/handle/10486/692896/lopez_lopez_javier_tfg.pdf?sequence=1&isAlloved=y
- [4] Microsoft, “¿Qué es la autenticación?” Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.microsoft.com/es-es/security/business/security-101/what-is-authentication>
- [5] Inc. Red Hat, “¿Qué es la automatización?” Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.redhat.com/es/topics/automation/whats-it-automation>
- [6] I. Amazon Web Services, “¿Qué es MQTT?”, 2023. Consultado: el 1 de abril de 2024. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/mqtt/>
- [7] Angel Robledano, “Qué es C++: Características y aplicaciones”, OpenWebinars. Consultado: el 1 de abril de 2024. [En línea]. Disponible en: <https://openwebinars.net/blog/que-es-cpp/>
- [8] IBM, “¿Qué es el cifrado?”. Consultado: el 1 de abril de 2024. [En línea]. Disponible en: <https://www.ibm.com/mx-es/topics/encryption>
- [9] HARDWARE, “Circuitos integrados e impresos.” Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://sites.google.com/prod/view/laramblahardware/circuitos-integrados-e-impresos>
- [10] COOHOM, “COOHOM”. Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.coohom.com>
- [11] Christine Bresnahan y Richard Blum, *Linux Essentials*, Wiley. 2015.
- [12] K. Madani y A. Dourado, *Intelligent Control Systems with an Introduction to System of Systems Engineering*. CRC Press, Taylor & Francis Group., 2022.
- [13] ARIMETRICS, “Qué es Dashboard”. Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.arimetrics.com/glosario-digital/dashboard>
- [14] TEAM ASANA, “¿Qué es un diagrama de flujo y cómo hacerlo?” Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://asana.com/es/resources/what-is-a-flowchart>
- [15] Samuel Greengard, “The Internet of Things”, The MIT Press, 2015.
- [16] Rolf H. Weber, “Internet of Things – New security and privacy challenges”, *Springer*, 2020.
- [17] Alexandra Deschamps-Sonsino, *Smarter Homes: How Technology Will Change Your Home Life*. Apress, 2018.
- [18] BizTalk Server, “¿Qué es la escalabilidad?”, 2023. Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/biztalk/core/what-is-scalability>
- [19] Kolban N., *Kolban’s Book on ESP32*. CreateSpace Independent Publishing Platform., 2018.
- [20] J. F. Samaniego, “ISO/IEC 30141 abrió el camino: así es la nueva familia de estándares para internet de las cosas”. [En línea]. Disponible en: <https://blog.orange.es/innovacion/estandar-iot/>
- [21] Stewart J., *CompTIA Network+ Guide to Networks*. Cengage Learning, 2020.
- [22] Monk S., *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education, 2016.
- [23] Lanier H., *Gmail For Dummies*. For Dummies, 2019.

- [24] Damon A., *Google Forms for Surveys: An Easy Guide to Creating Customizing, and Sharing Surveys*. Amazon Digital Services LLC., 2020.
- [25] M.A. Cuauhtémoc Vélez Martínez, "Hardware y Software ", Instituto de Ingeniería UNAM. Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.iingen.unam.mx/es-mx/AlmacenDigital/CapsulasTI/Paginas/hardwareysoftware.aspx>
- [26] Kleinrock L., *The Birth of the Internet*. Springer, 2012.
- [27] Oracle, "¿Qué es la IA? Conoce la inteligencia artificial". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>
- [28] IBM Corporation, "Interfaces de usuario". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.ibm.com/docs/es/i/7.5?topic=applications-user-interfaces>
- [29] D. Guide. IONOS, "¿Qué es una interfaz gráfica de usuario (GUI)?" [En línea]. Disponible en: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-una-gui/>
- [30] Gruber J., *Interoperability*. Oxford University Press, 2019.
- [31] Atzori L., Iera A., y Morabito G., "'The Internet of Things: A survey'. In: *Computer Networks*", 2010.
- [32] Amazon Web Services, "¿Qué es JavaScript?" Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/javascript/>
- [33] Sebesta R. W., *Concepts of Programming Languages*. Pearson, 2018.
- [34] Inc. The MathWorks, "Matemáticas. Gráficas. Programación." Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://la.mathworks.com/products/matlab.html>
- [35] Stallings W., *Data and Computer Communications 10th ed.* Pearson, 2019.
- [36] I. Solutions Company, "¿QUÉ SON LOS MIDDLEWARE? SOLUCIONES DE INTEGRACIÓN". [En línea]. Disponible en: <https://www.imagar.com/consultoria-informatica-especializada/que-son-los-middleware-soluciones-de-integracion/>
- [37] Secretaría de Economía, "Normas Oficiales Mexicanas". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.gob.mx/se/acciones-y-programas/normas-oficiales-mexicanas-14424>
- [38] O'Hanlon C. y Rees J., *Node-RED: Programming for the Internet of Things*. Packt Publishing Ltd., 20169.
- [39] Lopez G., Blasco J., Marco A., y Casas R., "'Intelligent Object Architecture for the IoT: The F-Interop Approach'. In: *Internet of Things – IoT Infrastructures*", Springer, 2019.
- [40] Raymond E. S., *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media., 1999.
- [41] Gruber T. R., "A translation approach to portable ontologies"., *Knowledge Acquisition*, 1993.
- [42] Microsoft, "Paint - Microsoft Support". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://support.microsoft.com/en-us/topic/paint-3e3424e1-c5e2-4d7c-91c1-3b3a9000d034>
- [43] O. Molina, "Plataforma de desarrollo", 2016, [En línea]. Disponible en: <https://es.slideshare.net/osvaldomolina11/plataformas-de-desarrollo-64318009>
- [44] MCAD, "Programación visual y sus aplicaciones en BIM". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://mcaad.co/programacion-visual-y-sus-aplicaciones-en-bim/>
- [45] PROTOTIPO0, "¿Qué es un prototipo y para qué sirve?" Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://prototip0.com/disenio-de-prototipos/>
- [46] Tanenbaum A. S. y Wetherall D. J., *Computer Networks*, 6a ed. Pearson, 2018.
- [47] Upton E. y Halfacree G., *Raspberry Pi User Guide: Getting Started with Raspberry Pi*. Wiley, 2019.
- [48] S. Industrial, "Actuadores". [En línea]. Disponible en: <https://sindustrial.com.mx/blog/actuadores/>

- [49] ARIMETRICS, "Qué es Script". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.arimetrics.com/glosario-digital/script>
- [50] Goleniewski L., *Telecommunications Essentials: The Complete Global Source*. Pearson Education, 2019.
- [51] Budynas R. G. y Nisbett J. K., *Shigley's Mechanical Engineering Design*. McGraw-Hill Education., 2015.
- [52] Domingo J. G., *Domótica e inmótica: Edificios inteligentes*. Ediciones Díaz de Santos., 2017.
- [53] Tanenbaum A. S. y Boss H., *Modern Operating Systems*. Pearson., 2014.
- [54] Al-Fuqaha A., Guizani M, Mohammadi M., Aledhari M., y Ayyash M., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys & Tutorials*, 2015.
- [55] Barnes J. y Jones M., *Understanding telemetry systems*. Artech House, 2000.
- [56] Weiser M., "The computer for the 21st century.", *Scientific American*, 1991.
- [57] Microsoft, "Microsoft Visio". Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.microsoft.com/en-us/microsoft-365/visio/compare-visio-plans>
- [58] Wireshark Community., "Wireshark - Official Website." Consultado: el 2 de abril de 2024. [En línea]. Disponible en: <https://www.wireshark.org/>
- [59] V. Tirupathi y K. Sagar, "A Research on Interoperability Issues in Internet of Things at Application Layer", 2019, [En línea]. Disponible en: <https://www.ijrte.org/wp-content/uploads/papers/v8i1s4/A11440681S419.pdf>
- [60] R. Orguidan, "Domótica", *Ing. Inf. Universidad de Girona*, pp. 1–15, 2019, [En línea]. Disponible en: https://sistemamid.com/panel/uploads/biblioteca/2014-06-14_07-37-03105052.pdf
- [61] F. J. Bellido Outeiriño, J. M. Flores Arias, E. J. Palacios Garcia, V. Pallares Lòpez, y D. Matabuena Gómez Limón, "M2M Home Data Interoperable Management System Based on MQTT", 2017.
- [62] M. I. Yáñez Collado, "La Domótica, Un Bien Para Todos", *Innovacion Y Experiencias Educativas*, núm. TECNOLOGÍA, EL FUTURO, 2010, [En línea]. Disponible en: https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_30/ISABEL_YAN EZ_1.pdf
- [63] M. Noura, M. Atiquzzaman, y M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges", *Mobile Networks and Applications*, , 2019.
- [64] V. M. Tayur y R. Suchithra, "Application Layer of Internet of Things", *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017.
- [65] Carmina Jasso López, "Percepción de inseguridad en México", *Revista Mexicana de Opinión Pública*, 2013.
- [66] Numbeo, "América: Índice de Criminalidad por País 2023 Mitad de año". Consultado: el 20 de enero de 2024. [En línea]. Disponible en: <https://es.numbeo.com/criminalidad/clasificaciones-por-pa%C3%ADs?region=019&title=2023-mid>
- [67] INEGI, "ENCUESTA NACIONAL DE SEGURIDAD PÚBLICA URBANA PRIMER TRIMESTRE DE 2023", México, 2023.
- [68] INEGI, "ENCUESTA NACIONAL DE SEGURIDAD PÚBLICA URBANA (ENSU) Cuarto trimestre de 2023", México, 2024.
- [69] Fiscalía General de Justicia de la Ciudad de México, "BOLETÍN ESTADÍSTICO DE LA INCIDENCIA DELICTIVA EN LA CIUDAD DE MÉXICO DEL MES DE NOVIEMBRE 2023", CDMX, 2023.
- [70] Y. Guaman, G. Ninahualpa, G. Salazar, y T. Guarda, "Comparative Performance Analysis between MQTT and CoAP Protocols for IoT with Raspberry PI 3 in IEEE 802.11 Environments", *Iberian Conference on Information Systems and Technologies, CISTI*, 2020.

- [71] Pardo Dimas, "Interoperabilidad del IoT; la clave para su desarrollo", ¿Qué necesita el IoT para crecer? La interoperabilidad entra en escena. Consultado: el 4 de octubre de 2021. [En línea]. Disponible en: <https://pandorafms.com/blog/es/interoperabilidad-del-iot/>
- [72] IFT, "En México hay 80.6 millones de usuarios de internet y 86.5 millones de usuarios de teléfonos celulares: ENDUTIH 2019. 17 de febrero | Instituto Federal de Telecomunicaciones - IFT". Consultado: el 5 de octubre de 2021. [En línea]. Disponible en: <http://www.ift.org.mx/comunicacion-y-medios/comunicados-ift/es/en-mexico-hay-806-millones-de-usuarios-de-internet-y-865-millones-de-usuarios-de-telefonos-celulares>
- [73] P. Lara, "Telcel Empresas", ¿Que se necesita para activar el IoT en México? [En línea]. Disponible en: <https://www.telcel.com/empresas/tendencias/notas/activar-iot-en-mexico>
- [74] R. Hernández Balibrea, "Tecnología domótica para el control de una vivienda.", 2012. [En línea]. Disponible en: <https://repositorio.upct.es/bitstream/handle/10317/2793/pfc4381.pdf>
- [75] L. VIVE, "Domótica e inmótica: de la fantasía a la realidad". [En línea]. Disponible en: <https://www.vivelibre.es/domotica-e-inmotica-de-la-fantasia-a-la-realidad>
- [76] M. I. Yáñez Collado, "La Domótica, Un Bien Para Todos", *Innovacion Y Experiencias Educativas*, núm. TECNOLOGÍA, EL FUTURO, 2010, [En línea]. Disponible en: https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_30/ISABEL_YAN_EZ_1.pdf
- [77] EDteam, "Internet of Things y la domótica." [En línea]. Disponible en: <https://ed.team/blog/internet-things-y-la-domotica>
- [78] W. Choi, J. Kim, S. Lee, y E. Park, "Smart home and internet of things: A bibliometric study", *J Clean Prod*, 2021.
- [79] M. Kim, K. L. Man, y N. Helil, "Advanced Internet of Things and Big Data Technology for Smart Human-Care Services", *J Sens*, 2019.
- [80] Barcelona, "Interoperabilidad y estándares abiertos". [En línea]. Disponible en: <https://www.barcelona.cat/digitalstandards/es/tech-sovereignty/0.1/interoperability>
- [81] N. Balta-Ozkan, R. Davidson, M. Bicket, y L. Whitmarsh, "The development of smart homes market in the UK", *Energy*, 2013.
- [82] S.A. Simón, "Cómo elegir un sistema de control domótico para mi hogar", 2023, [En línea]. Disponible en: <https://bricoladores.simonelectric.com/bid/270846/C-mo-elegir-un-sistema-de-control-dom-tico-para-mi-hogar>
- [83] C. Téllez Ortega, "Sistema De Prevencion Y Sistemas De Seguridad Para Casas Haptacion Utilizando Domotica a Base De Microcontrolador", 2017.
- [84] Pablo Castillo García, "Diseño de un sistema de control domótico para una casa de dos viviendas", 2019, [En línea]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/144331>
- [85] C. Téllez Ortega, "Sistema De Prevencion Y Sistemas De Seguridad Para Casas Haptacion Utilizando Domotica a Base De Microcontrolador", 2017.
- [86] I. E. de Posgrado, "¿Qué es el Internet de las Cosas?" Consultado: el 15 de mayo de 2021. [En línea]. Disponible en: <https://www.iep.edu.es/que-es-internet-de-las-cosas-iot/>
- [87] K. Rose, S. Eldrige, y L. Chapin, "La Internet de las Cosas Una breve- reseña", *Internet Society*, 2015.
- [88] Eunice Echeverría Portillo y Ma. del Carmen Celis Ortega, "TCP/IP e Interconexión de Redes. Introducción a la Arquitectura Internet OSI. Un Modelo de Referencia", *Polibits*, 1998, Consultado: el 1 de abril de 2024. [En línea]. Disponible en: <https://polibits.cidetec.ipn.mx/ojs/index.php/polibits/article/viewFile/3671/2989>
- [89] IBM Corporation, "Protocolos TCP/IP". Consultado: el 1 de febrero de 2024. [En línea]. Disponible en: <https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-protocols>

- [90] A. Amazon, “¿Cuál es la diferencia entre microprocesadores y microcontroladores?” [En línea]. Disponible en: <https://aws.amazon.com/es/compare/the-difference-between-microprocessors-microcontrollers/>
- [91] raspberrypi.c, “¿Qué es Raspberry Pi?” [En línea]. Disponible en: <https://raspberrypi.c/que-es-raspberry/>
- [92] Espressif Systems, “ESP32 Series Datasheet”, *Espressif Systems*, 2019. [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [93] I. Rodríguez, “Arquitectura orientada a eventos sobre protocolo MQTT Luciano Chambers Silvina Maneglia Licenciatura en Sistemas”, 2021.
- [94] Fundación Eclipse, “Eclipse Mositto”. [En línea]. Disponible en: <https://mosquitto.org/>
- [95] L. Llamas, “¿QUÉ ES MQTT? SU IMPORTANCIA COMO PROTOCOLO IOT”. [En línea]. Disponible en: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- [96] M. Á. Moral Pérez, “NODE-RED COMO HERRAMIENTA VISUAL DE DISPOSITIVOS IOT”, 2021.
- [97] C. MDN, “Que es un servidor WEB?”, 2023, [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server
- [98] SINELEC, “¿QUÉ ES NODE-RED Y PARA QUÉ SIRVE?” [En línea]. Disponible en: <https://gruposinelec.com/que-es-node-red-y-para-que-sirve/>
- [99] Á. E. Azuero Azuero, “Significatividad del marco metodológico en el desarrollo de proyectos de investigación”, *Revista Arbitrada Interdisciplinaria Koinonía*, 2019.
- [100] F. Hostingplus, “Modelo de prototipos: ¿qué es y cuáles son sus etapas?” [En línea]. Disponible en: <https://www.hostingplus.mx/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/>
- [101] INEGI, “INEGI”, 2023.
- [102] C. De Información y G. Municipal, “Compendio de información geográfica municipal 2010 BENITO JUÁREZ”, *Instituto Nacional de Estadística y Geografía*, 2010.
- [103] J. Casas Anguita, J. R. Repullo Labrador, y J. Donado Campos, “La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I)”, *Aten Primaria*, 2003.
- [104] “METODOLOGÍA ESTADÍSTICA SIMPLE”, 2012.
- [105] Momentive, “surveymonkey.com”. [En línea]. Disponible en: <https://es.surveymonkey.com/>
- [106] Statista, “Porcentaje de usuarios de teléfonos celulares sobre la población en México de 2015 a 2021”. [En línea]. Disponible en: <https://es.statista.com/estadisticas/1092097/tasa-penetracion-telefonos-celulares-mexico>
- [107] OpenJS Foundation & Contributors, “Node-RED”. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://nodered.org>
- [108] “Bienvenido a Fritzing”. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://fritzing.org/>
- [109] “DHT-sensor-library/DHT.h at master · adafruit/DHT-sensor-library · GitHub”. Consultado: el 19 de enero de 2024. [En línea]. Disponible en: <https://github.com/adafruit/DHT-sensor-library/blob/master/DHT.h>
- [110] Ben Hardill, “Puente de habilidades para el hogar Alexa de Node-RED”. Consultado: el 20 de enero de 2024. [En línea]. Disponible en: <https://alexa-node-red.bm.hardill.me.uk/docs>

ANEXO

El código de la programación del sistema puede ser consultada directamente en archivo Anexo contenido en el CD de la Tesis.

mynodered_cargas_temperatura.ino

mynodered_control_acceso.ino

El cuestionario realizado para la elaboración de la tesis puede ser consultado en el archivo Anexo contenido en el CD de la Tesis.

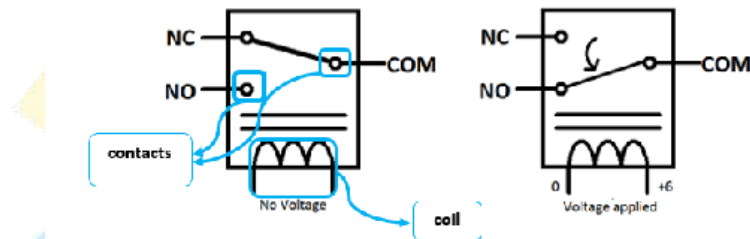
DATASHEET'S



RELAY MODULES

RELAY WORKING IDEA

Relays consist of three pins normally open pin, normally closed pin, common pin and coil. When coil power on magnetic field is generated the contacts connected to each other.

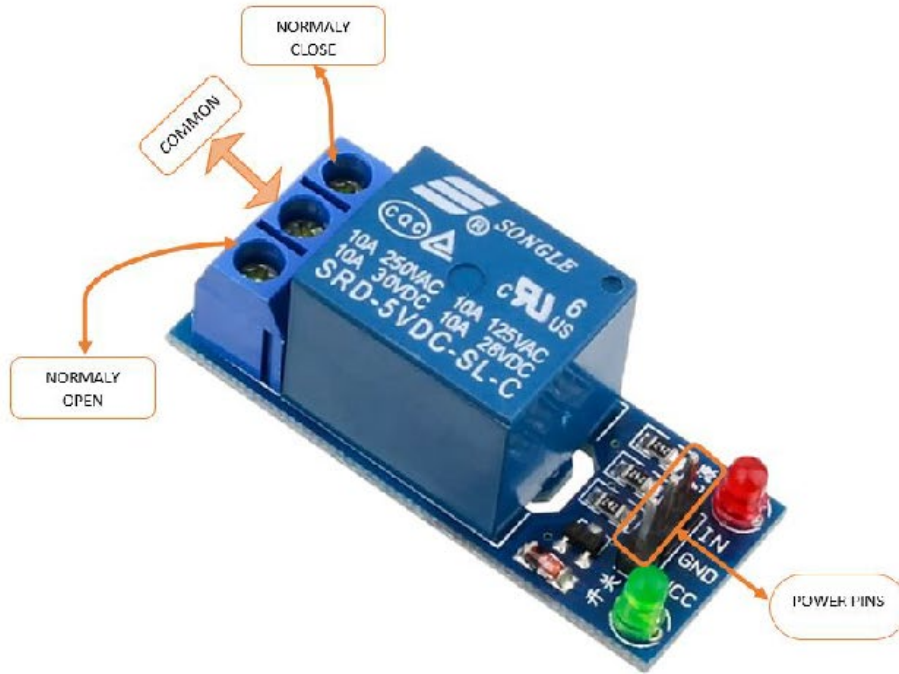


Relay modules 1-channel features

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power isolation between the coil and the switching pins so coils can be easily powered from Arduino by connecting VCC and GND pins from Arduino kit to the relay module kit after that we choose Arduino output pins depending on the number of relays needed in project designed and set these pins to output and make it out high (5 V) to control the coil that allow controlling of switching process.



NOTE : whatever was the relay channels number the pinconfiguration is the same for every channel except the power pins (VCC and GND) are for the board itself. The input signal (IN) pin for every relay.



ESP32 Series

Datasheet

2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC

Including:

ESP32-D0WD-V3

ESP32-D0WDR2-V3

ESP32-U4WDH

ESP32-S0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6 – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6-V3 – [Not Recommended for New Designs \(NRND\)](#)



Version 4.3
Espressif Systems
Copyright © 2023



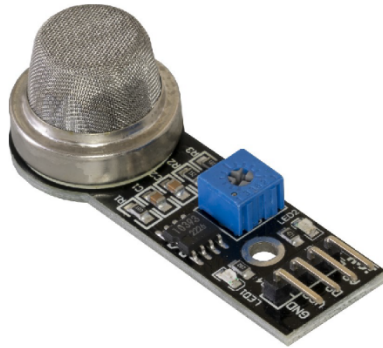
www.espressif.com

Su contenido completo puede ser consultado en:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

SEN-MQ2

Analog sensor for combustible gas and smoke on module



This analog gas sensor has a small heating part with an electronic chemical sensor. It is suitable for indoor usage. The sensor can output exact values only after warm-up phase.

Caution: sensor gets hot while usage!



Published: 26.09.2023

MAIN FEATURES

Measurement range	100 - 10'000 ppm
Measurable substances	LPG, <i>i</i> -butan (C4H10), propane (C3H8), methane (CH4), hydrogen(O2), alcohol, smoke
Application areas	Detecting gas leaks, gas alarm, robotic, microcontroller projects
Compatible with	Raspberry Pi (with AD-converter), Arduino. etc.
Special features	High sensitivity which can be adjusted by potentiometer, quick response time, wide detection range
Dimensions	51,5 x 20 x 17mm
Items delivered	SEN-MQ2

FURTHER SPECIFICATIONS

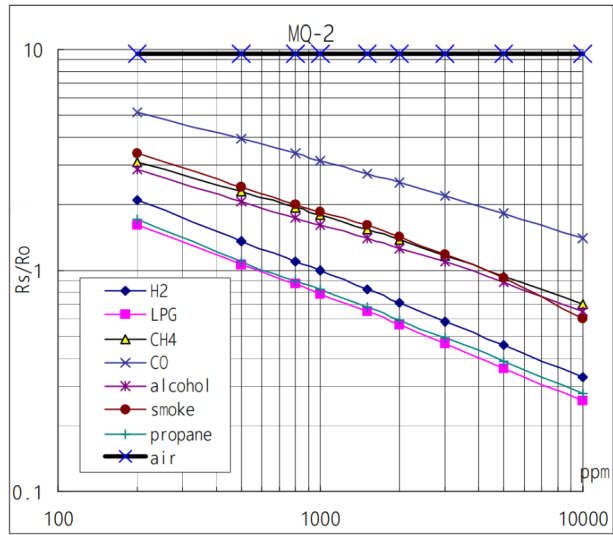
Analog Output	values will be processed by microcontroller
Digital Output	thresholds can be set
Pins:	
VCC	Voltage supply (5 V)
GND	Ground
AOUT	Analog output (0 V - 5 V)
DOUT	Digital output (0 V / 5 V)

FURTHER DETAILS

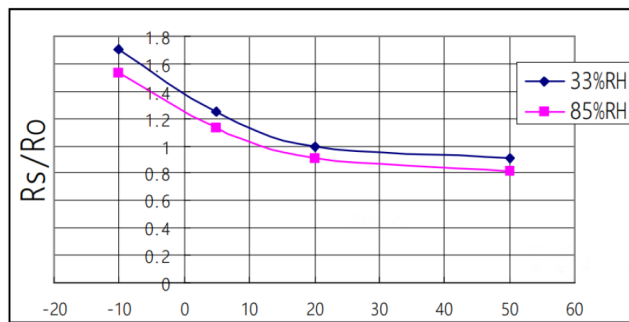
Article No.	SEN-MQ2
EAN:	4250236819938
Customs Tariff No.	90269000

SEN-MQ2

Analog sensor for combustible gas and smoke on module



This shows the typical sensitivity characteristics of the MQ-2. Rs means resistance of the sensor in different gases, Ro means resistance of sensor in 1000ppm H2.



Correlation between sensor resistance(Rs) and ambient temperature and humidity

The resistance of the sensor can be calculated with the following formula:

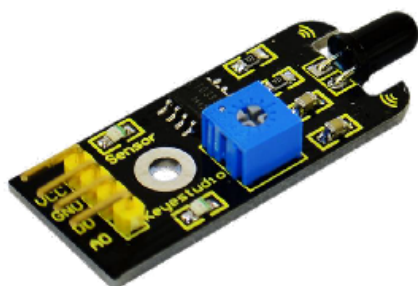
$$Rs = (Vc / VRL - 1) \times RL$$

VC= Supply voltage; VRL= Analog pin voltage; RL= Load resistance (1,5k)

Published: 26.09.2023

MODULO SENSOR DETECTOR FLAMA

ESTE MODULO PERMITE DETECTAR Y MEDIR LA LUZ PRODUCIDA POR UNA FLAMA, PUEDE IMPLEMENTARE EN SISTEMAS DE CONTRA INCENDIO, DISEÑO MODULAR DE FÁCIL COLOCACIÓN.

**ATRIBUTOS**

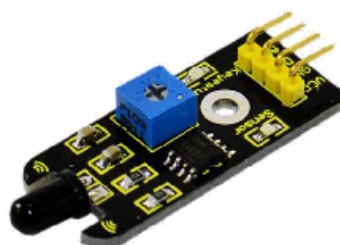
- El modulo AR-FLAME (KS-0036) es un sensor que permite detectar la flama producida por el fuego u otras fuentes de luz, está diseñado para se implementado en circuitos y proyectos Arduino.
- Este sensor de flama puede ser utilizado para detectar fuego u otras fuentes de luz cuya longitud de onda se encuentre en el rango de los 760nm hasta los 1100nm.
- Se dispone de un potenciómetro que permite ajustar la sensibilidad del sensor para adecuarlo a las necesidades del proyecto, así como dos salidas una digital y una analógica para utilizar la que más se adapte a el uso establecido.
- Como ejemplo este sensor puede implementarse en un robot para combate contra incendios. Incluso puede usarse como un medio de control de flamas para cortar el suministro de gas si no se produce una flama en sistemas de calefacción, boiler, estufas y hornos.
- La flama juega un papel muy importante en este sensor ya que se utiliza la luz que se despiden de esta para encontrar la fuente que provoca o ocasiona el fuego.
- Cuenta con una perforación con diámetro para un tornillo M3 para facilitar su colocación y montaje, permitiendo su uso en cualquier aplicación que se requiera.

ESPECIFICACIONES TÉCNICAS

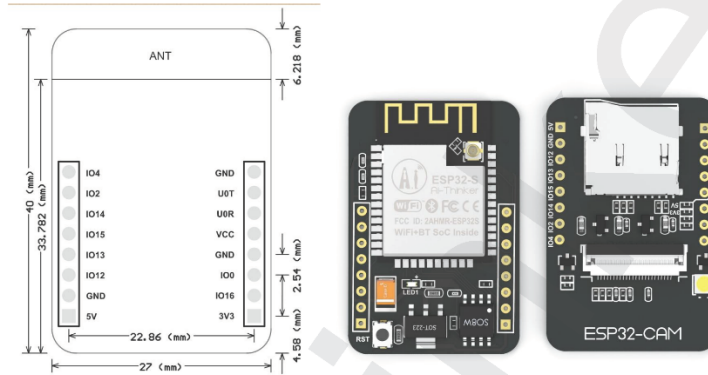
DIMENSIONES:	44 mm x 17 mm x 7 mm
MATERIAL:	FR-4
PESO:	4 gramos
VOLTAJE DE OPERACIÓN:	3.3 – 5.0 Vcc
LONGITUD DE ONDA:	760nm – 1100nm
RANGO DE DETECCIÓN:	20cm (4.8V) – 100cm (1V)
TIPO DE SENSOR:	Digital
TEMPERATURA DE OPERACIÓN:	-25°C +85°C
PINO OUT:	1. GND 2. VCC 3. SEÑAL

ACCESORIOS

- 1 x AR-FLAME



ESP32-CAM Module



Features

- The smallest 802.11b/g/n Wi-Fi BT SoC Module
- Low power 32-bit CPU, can also serve the application processor
- 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)

Overview

The ESP32-CAM has a very competitive small-size camera module that can operate independently as a minimum system with a footprint of only 27*40.5*4.5mm and a deep sleep current of up to 6mA.

ESP-32CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.

ESP-32CAM adopts DIP package and can be directly inserted into the backplane to realize rapid production of products, providing customers with high-reliability connection mode, which is convenient for application in various IoT hardware terminals.

Documento completo disponibile en: https://docs.ai-thinker.com/media/esp32/docs/esp32-cam_product_specification_zh.pdf

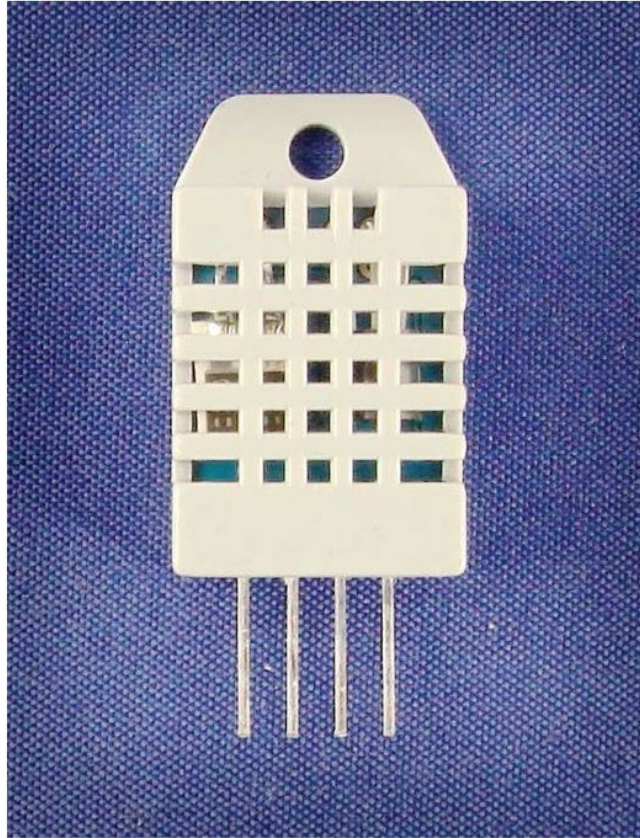
http://www.ai-thinker.com/Uploads/file/20231220/20231220100532_99224.pdf

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

Digital-output relative humidity & temperature sensor/module

DHT22 (DHT22 also named as AM2302)



Capacitive-type humidity and temperature module/sensor

1

Thomas Liu (Business Manager)

Email: thomasliu198518@yahoo.com.cn

Documento completo disponibile en:

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>



Manual de Instrucciones

INSTRUCCIONES

Los interruptores magnéticos pueden ser utilizados en puertas o ventanas.

1. Tienda el cableado del sistema de alarma hasta el lugar donde instalará el interruptor.

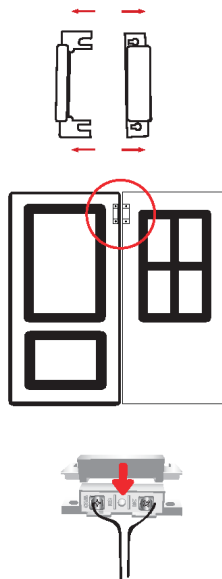
2. Retire la cubierta.

3. Ponga las dos piezas del interruptor sobre la superficie de la puerta o ventana donde desea instalarlo, y realice las marcas de perforación. Asegúrese de que la distancia entre ambas piezas no exceda los 2 cm y de que estén alineadas, como se muestra en la imagen.

4. Fije las piezas con tornillos (no incluidos).

5. Coloque los cables del sistema de alarma en las terminales y apriete los tornillos.

6. Coloque nuevamente la cubierta. Cuando la puerta o ventana se abra, se pondrá en marcha el sistema de alarma.



Antes de utilizar el producto, lea el siguiente instructivo para evitar cualquier mal funcionamiento.

PRECAUCIONES

- Este aparato no se destina para utilizarse por personas (incluyendo niños), cuyas capacidades físicas, sensoriales o mentales sean diferentes o estén reducidas, o carezcan de experiencia o conocimiento.
- Los niños deben supervisarse para asegurar que no empleen el aparato como juguete.
- No exponga el producto a temperaturas extremas.
- No lo use ni almacene en lugares donde existan goteras o salpicaduras de agua. Puede causar un mal funcionamiento

ESPECIFICACIONES

- Distancia máxima de operación: 25 mm
- Voltaje de operación: ≤ 24 V
- Voltaje de ruptura: 250 V máx.
- Corriente: 100 mA
- Tiempo de operación: 0,6 milisegundos
- Tiempo de liberación: 0,1 milisegundos

El diseño del producto y las especificaciones pueden variar sin previo aviso.

IMPORTADO POR

ELECTRÓNICA STEREN S.A. DE C.V.
 Biólogo Maximino Martínez, núm. 3408, col. San Salvador Xochimanca, Azcapotzalco, Ciudad de México, C.P. 02870, RFC: EST850628-K51

STEREN PRODUCTO EMPACADO S.A. DE C.V.
 Autopista México-Querétaro, Km 26.5, sin número, Nave Industrial 3-A, col. Lomas de Boulevares, Tlalnepantla de Baz, Estado de México, C.P. 54020, RFC: SPE-941215-H43

En caso de que tu producto presente alguna falla, o si tienes alguna duda o pregunta, por favor, llama a nuestro Centro de Atención a Clientes, en donde con gusto te atenderemos en todo lo relacionado con tu producto Steren.

Centro de Atención a Clientes
 (55) 15 16 60 00

Para el solenoide usado en la cerradura, no se encontró una hoja de fabricante, por lo que las especificaciones de la misma pueden ser consultadas en:

<https://www.adafruit.com/product/1512>

ULN200x, ULQ200x High-Voltage, High-Current Darlington Transistor Arrays

1 Features

- 500-mA-Rated Collector Current (Single Output)
- High-Voltage Outputs: 50 V
- Output Clamp Diodes
- Inputs Compatible With Various Types of Logic
- Relay-Driver Applications

2 Applications

- Relay Drivers
- Stepper and DC Brushed Motor Drivers
- Lamp Drivers
- Display Drivers (LED and Gas Discharge)
- Line Drivers
- Logic Buffers

3 Description

The ULx200xA devices are high-voltage, high-current Darlington transistor arrays. Each consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads.

The collector-current rating of a single Darlington pair is 500 mA. The Darlington pairs can be paralleled for higher current capability. Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. For 100-V (otherwise interchangeable) versions of the ULx2003A devices, see the SLRS023 data sheet for the SN75468 and SN75469 devices.

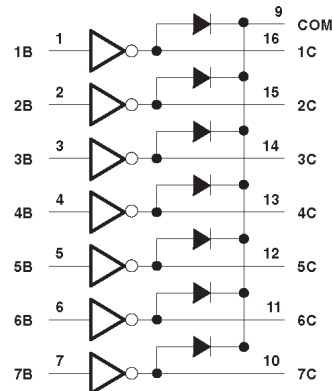
The ULN2002A device is designed specifically for use with 14-V to 25-V PMOS devices. Each input of this device has a Zener diode and resistor in series to control the input current to a safe limit. The ULx2003A devices have a 2.7-kΩ series base resistor for each Darlington pair for operation directly with TTL or 5-V CMOS devices.

The ULx2004A devices have a 10.5-kΩ series base resistor to allow operation directly from CMOS devices that use supply voltages of 6 V to 15 V. The required input current of the ULx2004A device is below that of the ULx2003A devices, and the required voltage is less than that required by the ULN2002A device.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
ULx200xD	SOIC (16)	9.90 mm × 3.91 mm
ULx200xN	PDIP (16)	19.30 mm × 6.35 mm
ULN200xNS	SOP (16)	10.30 mm × 5.30 mm
ULN200xPW	TSSOP (16)	5.00 mm × 4.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.



Simplified Block Diagram

▲ An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Documento completo disponible en:

https://www.ti.com/lit/ds/symlink/uln2003a.pdf?ts=1706254758505&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FULN2003A