



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA ELECTRONICA Y AUTOMATIZACION**

**DISEÑO E IMPLEMENTACION DE UN BANCO DE PRUEBAS EMPLEANDO EL
NodeMCU (ESP8266) PARA APLICACIONES DOMOTICAS CON IoT Y APPLE HOMEKIT.**

**Trabajo de titulación a la obtención del
Título Ingeniero Electrónico y Automatización**

**AUTORES: JAMYE EDUARDO MENDOZA MENDOZA Y RONNY ALEXANDER
QUEVEDO CELI**

TUTOR: PhD. BREMNEN MARINO VELIZ NOBOA

Guayaquil – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jamye Eduardo Mendoza Mendoza con número de identificación N° 094287357-1 y Ronny Alexander Quevedo Celi con número de identificación N°1104368087; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 9 de septiembre del año 2022

Atentamente,



Jamye Eduardo Mendoza Mendoza

094287357-1



Ronny Alexander Quevedo Celi

1104368087 **CERTIFICADO DE**

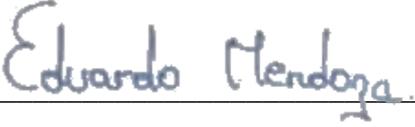
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITECNICA SALESIANA

Nosotros, Jamye Eduardo Mendoza Mendoza con número de identificación No 0942873571 y Ronny Alexander Quevedo Celi con número de identificación 1104368087 expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Trabajo de Titulación: **“DISEÑO E IMPLEMENTACION DE UN BANCO DE PRUEBAS EMPLEANDO EL NodeMCU (ESP8266) PARA APLICACIONES DOMOTICAS CON IoT Y APPLE HOMEKIT”**, el cual ha sido desarrollado para optar por el título de Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 9 de septiembre del año 2022

Atentamente,



Jamye Eduardo Mendoza Mendoza

094287357-1



Ronny Alexander Quevedo Celi

1104368087

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, BREMNEN MARINO VELIZ NOBOA con documentación N° 0703865139, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación “**DISEÑO E IMPLEMENTACION DE UN BANCO DE PRUEBAS EMPLEANDO EL NodeMCU (ESP8266) PARA APLICACIONES DOMOTICAS CON IoT Y APPLE HOMEKIT**”, realizado por Jamye Eduardo Mendoza Mendoza con documento de identificación N° 094287357-1 y por Ronny Alexander Quevedo Celi con documento de identificación N° 1104368087, obteniendo como resultado final de trabajo de titulación bajo la opción Trabajo de titulación que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 9 de septiembre del año 2022

Atentamente,



PhD. Bremnen Marino Veliz Noboa

0703865139

AGRADECIMIENTO

A Dios por la ayuda que me ha brindado para terminar mi carrera profesional dentro de la universidad UPS Guayaquil, agradecerles a mis tutores por formar parte de mi proceso formativo. Le agradezco a mis papas y hermanos por el apoyo a lo largo de la carrera.

Ronny Alexander Quevedo Celi

AGRADECIMIENTO

A Dios, por bendecirme en todo momento, darme sabiduría y fuerzas para seguir adelante en este camino.

A mi madre, que siempre me ha apoyado en cada momento de mi vida.

A mis abuelos por confiar y creer en mí.

A todas las personas que me apoyaron en algún momento de la carrera universitaria.

Jamye Eduardo Mendoza Mendoza

RESUMEN

El presente proyecto de titulación muestra el desarrollo de un prototipo de seguridad automatizado domótico para viviendas, compuesto por sensores de humedad, temperatura, gas, el tipo de gas que mide es dióxido de carbono y sus unidades son micro-gramo/metro cubico ($\mu\text{g}/\text{m}^3$), luz, movimiento y actuadores de luces led que son controlados desde un dispositivo iPhone o desde un ordenador mediante una conexión a internet por WIFI. Para la conexión WIFI se ha usado el sistema en chip ESP8266 que es programado en el lenguaje C de Arduino que cumple la función de escribir, editar y grabar nuestro programa, su objetivo principal es conectarse a una red, requiere de un ordenador y un cable USB para conectar el chip con el ordenador. Se hace uso de la aplicación Apple Homekit y la pagina Adafruit. La primera permite controlar las luces y verificar los datos de cada sensor que envía al prototipo, es instalada en un dispositivo iPhone y se debe configurar ingresando a la aplicación, digitamos un código general del iPhone e iremos agregando los dispositivos. La segunda realiza las mismas funciones de la primera aplicación (Homekit) en controlar las luces y recibir cada dato de los sensores, se usa un ordenador y se configura creando un usuario y un código que me facilita la página principal de Adafruit y se lo ingresa a la Plataforma de la Raspberry. Un Raspberry Pi 3 se conecta con la ESP8266 por medio de puerto serial para alimentar, enviar y recibir información, fue usado para que a través del internet le envié los datos a Adafruit, también se configuro y se utilizó un Router independiente para el prototipo.

El presente trabajo de titulación está enfocado al aprendizaje de los estudiantes de la carrera de electrónica de la UPS-GUAYAQUIL, mediante practicas realizadas en el banco de pruebas.

PALABRAS CLAVES

Apple Homekit, ESP8266, IoT, Adafruit.

ABSTRACT

This degree project shows the development of a home automation automated security prototype for homes, composed of humidity, temperature, gas sensors, the type of gas it measures is carbon dioxide and its units are micrograms/cubic meter ($\mu\text{g}/\text{m}^3$), light, movement and LED light actuators that are controlled from an iPhone device or from a computer through a WIFI internet connection. For the WIFI connection, the ESP8266 chip system has been used, which is programmed in the Arduino C language that fulfills the function of writing, editing and recording our program, its main objective is to connect to a network, it requires a computer and a cable USB to connect the chip with the computer. Use is made of the Apple Homekit application and the Adafruit page. The first allows to control the lights and verify the data of each sensor that it sends to the prototype, it is installed on an iPhone device and must be configured by entering the application, we enter a general code of the iPhone and we will add the devices. The second performs the same functions of the first application (Homekit) in controlling the lights and receiving each data from the sensors, a computer is used and it is configured by creating a user and a code that the Adafruit home page provides me and enter it to the Raspberry Platform. A Raspberry Pi 3 connects to the ESP8266 through a serial port to power, send and receive information, it was used to send the data to Adafruit through the internet, an independent Router was also configured and used for the prototype.

The present degree work is focused on the learning of the students of the electronics career of the UPS-GUAYAQUIL, through practices carried out in the test bench.

- **KEYWORDS**

Apple Homekit, ESP8266, IoT, Adafruit.

2. Revisión de la Literatura o Fundamentos Teóricos	17
2.1. Internet de la Cosas IOT	17
2.2. Qué es una Plataforma IOT	17
2.3. Qué es Adafruit	17
2.4. Para que sirve Adafruit	18
2.5. Raspberry Pi	18
2.6. Para que sirve la plataforma Arduino	18
2.7. Apple Homekit	18
2.8. Cómo funciona Apple Homekit.....	18
3. Marco Metodológico	20
3.1. Metodología Experimental	20
3.2. Exposición del Modelo Esquemático	20
3.3. Diagrama Esquemático	20
3.4. ESP 8266	21
3.5. Raspberry Pi 3	21
3.6. Sensores	22
3.7. Sensor de Luz KY-018	22
3.8. Sensor de Temperatura y Humedad DHT11.....	23
3.9. Sensor de Gas MQ-135	24
3.10. Sensor de movimiento HC-SR501	24
3.11. Interfaz	25
3.12. Actuadores	25
3.13. Spot Led	25
3.14. Router TP-LINK WR840N	26
3.15. Micro SD 32GB	26
3.16. Fuente de Alimentación EBC-HQ 12VDC 4.2AMP	27
3.17. Breaker 2X25 AMP EBC	27
3.18. Ventilador 120X120X38MM 10V-220V	27
3.19. Filtro plástico para ventilador 120MM	28
3.20. Protector Metálico para ventilador 120MM	29
3.21. Toma doble polarizado 15A-110V	29
3.22. Bornera Riel	30
3.23. Luz Piloto Tipo Led	30
3.24. Cable de poder	31
3.25. Gabinete T/Gris fondo de metal 500X700X250CM.....	31

3.26.	Regulador de voltaje Lm2596	32
3.27.	Selector Eléctrico EMAS 2P - ON/OFF	32
3.28.	Switch ojo de cangrejo 1P - EBC	33
3.29.	Configuración de Red	33
	3.29.1 Acceso al Router para la conexión wifi de la universidad	34
	3.29.2 Acceso al Router para la conexión wifi de la casa	37
3.30.	Configuración Raspberry Pi3.....	40
	3.30.1 Instalación Sistema Operativo Raspbeery Pi 3	40
	3.30.2 Instalación de Balenaetcher	41
3.31.	Ensamblaje y Montaje	42
	3.31.1 Diagrama Eléctrico	44
4.	Resultados	45
4.1.	Practica#1	45
4.2.	Practica#2	46
4.3.	Practica#3	47
4.4.	Practica#4	48
4.5.	KY-018 Sensor de Luz	49
4.6.	DHT11 Sensor de Temperatura y Humedad.....	50
4.7.	MQ-135 Sensor de Gas	50
4.8.	HC-SR Sensor de Movimiento	51
4.9.	Luz de Entrada (Luz 1)	51
4.10.	Luz Cocina (Luz 2).....	52
4.11.	Luz Dormitorio (Luz 3)	52
4.12.	Luz Baño (Luz 4)	53
4.13.	Configuración del ESP8266	53
4.14.	Sensor de Luz KY-018	54
4.15.	Sensor de Temperatura y Humedad(DHT11)	54
4.16.	Sensor de Gas (MQ-135)	54
4.17.	Sensor de Movimiento (HC-SR)	54
5.	Cronograma	55
6.	Presupuesto	56

7. Conclusiones .	57
8. Recomendaciones .	58
9. Referencias Bibliográficas .	59
10. Anexos .	60
10.1. Anexos 1. Código ESP8266	60
10.2. Accesorios	66
10.3. Wifi	71

Índice de Figuras

Figura 1. Internet de las cosas (IOT) [1]	17
Figura 2. Diagrama de Raspberry [4].	18
Figura 3. Apple Homekit	19
Figura 4. Elementos de un sistema domótico	20
Figura 5. ESP 8266.	21
Figura 6. Tarjeta Raspberry Pi 3	22
Figura 7. Sensores	22
Figura 8. Sensor de luz KY-018	23
Figura 9. Sensor de temperatura y Humedad DHT11 .	23
Figura 10. Sensor de Gas MQ-135.	24
Figura 11. Sensor de movimiento HC-SR501.	24
Figura 12. Actuadores	25
Figura 13. Spot led.	25
Figura 14. Router TP-Link WR840N.	26
Figura 15. Memoria Sd Micro 32gb	26
Figura 16. Fuente de alimentación EBH-HQ 12 VDC 4.2 AMP.	27
Figura 17. Breaker 2X25 AMP EBC	27
Figura 18. Ventilador 120X120X38MM 10V-220V.	28
Figura 19. Ventilador 120X120X38MM 10V-220V.	28
Figura 20. Protector metálico para ventilar 120MM.	249
Figura 21. Toma doble polarizado 15A – 110V.	249
Figura 22. Bornero riel	30

Figura 23. Luz piloto led.....	30
Figura 24. Cable de poder	31
Figura 25. Gabinete t/gris fondo metal 500x700x250cm	31
Figura 26. Módulo regulador de voltaje	32
Figura 27. Selector eléctrico EMAS 2 posiciones	32
Figura 28. Switch de cangrejo 1P-EBC	33
Figura 29. Página del Router TL-WR840N.	34
Figura 30. Página de Configuración del Router TL-WR840N.....	34
Figura 31. Modo de operación del Router TL-WR840N.....	35
Figura 32. Modo de operación del Router TL-WR840N	35
Figura 33. Configuración de la red Router TL-WR840N.....	36
Figura 34. Finalizada la Configuración del Router TL-WR840N	36
Figura 35. Página del Router TL-WR840N.	37
Figura 36. Página de Configuración del Router TL-WR840N.....	37
Figura 37. Modo de operación del Router TL-WR840N.....	38
Figura 38. Modo de operación del Router TL-WR840N	38
Figura 39. Configuración de la red Router TL-WR840N.....	39
Figura 40. Finalizada la Configuración del Router TL-WR840N.....	39
Figura 41. Panel de Raspberry Pi OS	40
Figura 42. Panel principal Balenaetcher.....	41
Figura 43. Panel guardado de Balenaetcher	41
Figura 44. Dispositivos instalados dentro del Gabinete	42
Figura 45. Dispositivos instalados en la parte frontal del gabinete.....	42
Figura 46. Dispositivos instalados en la parte lateral del gabinete.	43
Figura 47. Conexiones y cableado del gabinete.....	43
Figura 48. Diagrama eléctrico de fuerza.....	44
Figura 49. Diagrama eléctrico de control	44
Figura 50. Elementos instalados.....	49
Figura 51. Sensor de Luz ky-018	49

Figura 52. Sensor DHT11.....	49 figura
53. Sensor de Gas	50 Figura 54.
Sensor HC-SR501.....	50
Figura 55. Sensor HC-SR501.....	50
Figura 56. Sensor HC-SR501.....	50

Índice de Tablas

Tabla 1. Conexión de sensor KY-018	54
Tabla 2. Conexión de sensor DHT11	54
Tabla 3. Conexión de Gas MQ-135	54
Tabla 4. Conexión de sensor HC-SR501	54
Tabla 5. Cronograma de proyecto	55
Tabla 6. Presupuesto de proyecto	56

Introducción

El uso de tecnología en tiempo real, la conexión de dispositivos y la automatización de procesos se denomina Internet de las cosas (IoT). Este concepto está ganando popularidad porque puede ayudar a mejorar los procesos de producción al proporcionar información, realizar tareas de forma remota y automatizar procesos que requieren poca o ninguna participación humana. La manera en que se desarrolla la tecnología está teniendo un gran golpe en la forma en que se crean los procesos industriales, lo que lleva a la idea de IoT.

Presenta el problema, su historia, su importancia, lo que cubrirá el proyecto y el objetivo general del proyecto. También hay una lista de objetivos más específicos que el proyecto logrará.

Introduce los conceptos subyacentes y las herramientas utilizadas en el desarrollo del proyecto. Se incluyen detalles sobre los módulos, sensores y otros temas.

Detalla el método experimental empleado para desarrollar este prototipo y todas las diferentes topologías de red, diseños eléctricos, métodos de comunicación y más.

Nos focalizamos en los resultados, explicando los resultados obtenidos tras implementar e investigar el prototipo didáctico, y cómo se consiguieron todos los objetivos propuestos inicialmente.

Analiza los resultados y brinda una interpretación ordenada de los efectos y datos obtenidos en la prueba de desempeño después de implementar el prototipo.

Finalmente, se extraen conclusiones y recomendaciones, indicando el propósito y las consecuencias de este proyecto después de su desarrollo, así como las opiniones para las posteriores investigaciones.

1. Problema

1.1. Antecedentes

En los laboratorios de Electrónica y Automatización de la UPS-Guayaquil, se pudo observar la falta de equipos electrónicos con los cuales los estudiantes puedan realizar los talleres de la materia de domótica, ya que muchos de estos trabajos han quedado en recursos teóricos por la falta de equipos.

El presente trabajo de titulación plantea una solución. El utilizar plataformas convencionales como Arduino, Raspberry Pi, acceden a eliminar variable limitante del precio, por lo tanto, la mayoría de estos códigos abiertos son "Open Source", incluso es muy versátil su programación y por lo tanto de múltiples maneras, para la personalización completa del servicio domótico.

Finalmente, a los estudiantes tendrán una aplicación instalada desde su teléfono o también otro dispositivo permitiendo conectarse y que se le otorgara la Carrera de Electrónica y Automatización UPS - Guayaquil para pueda conectarse al servidor de domótica.

1.2. Importancia y Alcances

Se busca aportar y utilizar las nuevas tecnologías, también al decir el beneficio que proporciona las nuevas tecnologías IoT es importante estar bien experto en el conocimiento técnico y poder ofrecer buenos resultados en esta área.

El presente trabajo de titulación brinda un mejor acercamiento al dispositivo digital que brindan las tecnologías IoT, para favorecer la enseñanza de los estudiantes, se entregara un prototipo didáctico para ser empleado para tener prácticas en los laboratorios y aportar en la enseñanza de los estudiantes.

1.3. Delimitación

1.3.1. Temporal

El proyecto de Titulación se desarrolló e implemento en la ciudad de Guayaquil, Ecuador para un periodo máximo de 6 meses a partir de su aprobación.

1.3.2. Espacial

Las pruebas y desarrollo del prototipo didáctico se realizaron a cabo en la ciudad de Guayaquil - Ecuador. Además, este prototipo será entregado en la UPS sede Guayaquil en los laboratorios de Electrónica para uso de los estudiantes.

1.3.3. Académica

Se aplico conocimientos teóricos, técnicos, prácticos a lo largo del periodo de la carrera de Ingeniería Electrónica, resaltando conocimientos de Automatización, Programación de sistemas embebidos, entre otros.

1.4. Objetivos Generales y Específicos

1.4.1. Objetivo General

Desarrollar un sistema domótico didáctico empleando NodeMCU (ESP8266) y plataformas de desarrollo convencionales como controladores y **Apple Home Kit para el laboratorio de domótica.**

1.4.2. Objetivos Específicos

- Diseñar un tablero para instalar los dispositivos domóticos.
- Crear una aplicación web a través de una interface grafica.
- Configurar un buen servidor domótico propio a través de una plataforma web utilizando la plataforma exclusiva de Apple Homekit y la Raspberry pi 3.

- Configurar la programación del control domótico utilizando la plataforma NodeMCU.
- Realizar 5 prácticas de sistemas domóticos.

2. Revisión de la Literatura o Fundamentos Teóricos.

2.1. Internet de la Cosas IOT

IOT se define como la conexión de muchos dispositivos y plataformas inteligentes que juntos permiten un "mundo inteligente" del medio ambiente. Mantener una red compleja conectando a miles de personas y dispositivos, interconectando cualquier dispositivo o aparato electrónico, sin importar donde se encuentre una persona, social, académica, etc.

Es un microordenador su tamaño es parecido a una tarjeta de crédito, la placa dispone con un puerto RJ45, un mini Jack de audio, con un puerto de video HMI, dos puertos USB 2.0.y necesita una memoria SD desde 8 a 128GB. Posee un sistema operativo Linux.

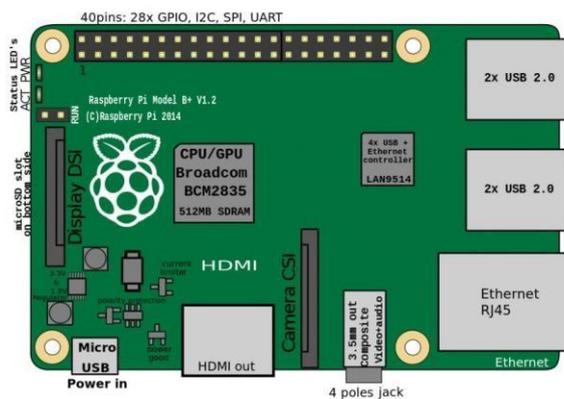


Figura 2. Diagrama de Raspberry [4]

2.6. Para que sirve la plataforma Arduino

Se puede utilizar para desarrollar elementos autónomos o también interactuar con otros dispositivos para interactuar tanto con el hardware como con el software. [4]

2.7. Apple Homekit

Es una plataforma que pertenece a Apple, su propósito es de domotizar Luces, enchufes y otros dispositivos. [6]

2.8. Cómo funciona Apple Homekit

Apple HomeKit viene a, como decía al principio, integrar en una misma plataforma las distintas soluciones de domótica que existen en el mercado. Bien, todas aquellas las marcas que decidan adaptarse a HomeKit. Y lo hace de la forma más sencilla que podía hacer: hablándole a tu iPhone, dándole instrucciones a Siri con tu propia voz, instrucciones como "Encender las luces", "Apaga la luz de la habitación de Pablo" o "Pon el termostato de la primera planta a 22 grados". [6]

HomeKit viene también a simplificar el proceso de configuración de los distintos dispositivos de distintas marcas. Bastará con descargar la aplicación del fabricante, abrirla y escanear el código de configuración con la cámara del iPhone o iPad, código similar a este: [6]

Pero, además, HomeKit nos permite crear y configurar diferentes escenas, zonas, habitaciones, casas o situaciones. [6]



Figura 3. Apple Homekit [6]

3. Marco Metodológico

3.1. Metodología Experimental

El desarrollo del proyecto de titulación se usó el método experimental, se realizó diversas pruebas entre dispositivos empleando protocolos de comunicación.

Se inicio con este prototipo, los sensores que se conectaron al esp8266, se adquirió información y fue enviada a la plataforma homekit.

Para escoger la plataforma IoT, se recibió varias opiniones, que tengan pocas limitaciones y que sean gratuitas, y que se acople a las necesidades del proyecto.

3.2. Exposición del Modelo Esquemático

El modelo didáctico esta implementado en un gabinete plástico eléctrico con fondo de metal y sus medidas son 500X700X250CM, con todos nuestros dispositivos.

Los sensores utilizados son: sensor de luz LDR (KY-018), sensor de temperatura y humedad (DHT11), sensor de gas (MQ-135), sensor de movimiento (HC-SR501).

El prototipo posee un breaker sensible general de 2 polos, una fuente de alimentación de 12v a 4.2A que alimenta a los reguladores de voltaje que se ajusta de 12VDC a 5VDC para alimentar al NodeMCU (ESP8266) y los sensores.

3.3. Diagrama Esquemático

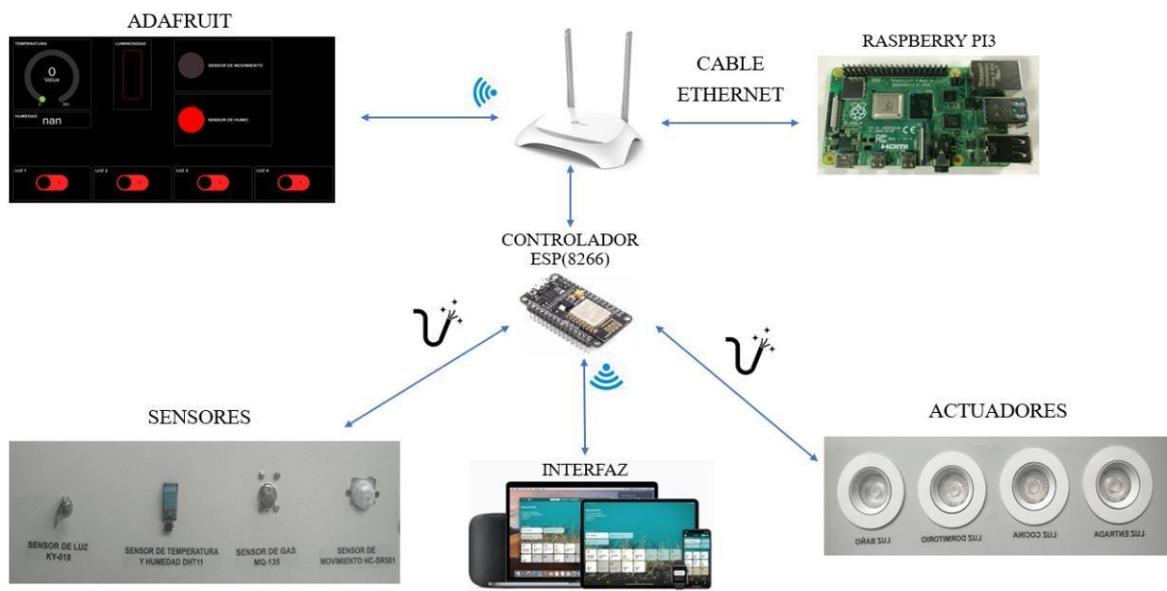


Figura 4. Elementos de un sistema domótico

3.4. ESP 8266

Se conoce como un chip compatible con el protocolo de comunicación TCP/IP y con conexión WIFI, el objetivo primordial es permitir a cualquier microcontrolador a una red. [3]



Figura 5. ESP 8266

3.5. Raspberry Pi 3

La Raspberry Pi es una pequeña computadora de bajo costo que es lo suficientemente pequeña como para caber en la palma de una mano, y puede conectar su televisor y teclado e interactuar con ella como cualquier otra computadora.

La Raspberry Pi 3 es una placa integrada, está diseñado y construido para ser utilizado junto con el sistema operativo, tiene un procesador ARM Cortex de 1.4 Ghz con 1 GB de RAM, WiFi, BLE y Ethernet. La última versión de la familia de tarjetas Raspberry PI es la versión 3 B plus.

Una de las principales mejoras respecto a la versión anterior es la posibilidad de conectarse a redes Wi-Fi en la banda de 5Ghz. También podemos mencionar cómo aumenta la velocidad de procesamiento de 1.2Ghz a 1.4Ghz. Finalmente, se han mejorado las capacidades de red, con versiones de Bluetooth de 4.1 a 4.2 y Ethernet con capacidad de 300 Mbps. [7]

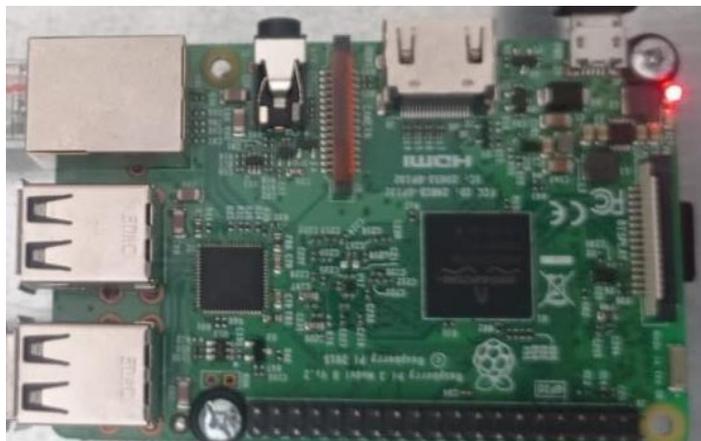


Figura 6. Tarjeta de Raspberry 3

3.6. Sensores

Son dispositivos en detectar acciones y en consecuencia en activar una o varias salidas que se lo conocen como actuadores.



Figura 7. Sensores

3.7. Sensor de Luz KY-018

El módulo KY-018 es un dispositivo electrónico, su resistencia necesita de la intensidad de la luz, sea mayor cantidad de luz será menor resistencia presentada, determina la ausencia de luz o presencia.[7]



Figura 8. Sensor de luz KY-018 [6]

3.8. Sensor de Temperatura y Humedad DHT11

Este sensor detecta humedad y temperatura del entorno. El sensor DHT11 mide un rango de humedad entre 20% y 95%. Su temperatura de medición está entre 0 y 50 °.



Figura 9. Sensor de temperatura y humedad DHT11

3.9. Sensor de Gas MQ- 135

Este sensor de control eficiente del aire se utiliza para localizar la contaminación del lugar, se suele utilizar en circuitos de control, como alarmas en casas, donde es necesario prevenir altos niveles de contaminación con niveles de oxígeno, como en su proceso industrial. También pueden ser compuestos peligrosos, especialmente para la salud.



Figura 10. Sensor de Gas MQ-135

3.10. Sensor de movimiento HC-SR501

El sensor **HC-SR501 es un sensor de movimiento**, se lo conoce como sensor PIR que consta de dos elementos. Por una parte, tiene un dispositivo emisor de la señal diferencial entre él y otro sensor que será el que active la alarma.



Figura 11. Sensor de movimiento HC-SR501

3.11. Interfaz

Una interfaz establece una función entre varios dispositivos o sistemas, funcionan individualmente uno del otro.

3.12. Actuadores

Es un dispositivo que también se lo puedo conocer como un receptor, convierte la energía en un movimiento normalmente en aplicar fuerza.



Figura 12. Actuadores

3.13. Spot Led

Transmiten una luz direccional de calidad increíble, ilumina un área o un objeto específico. También se lo conoce foco concentrado.



Figura 13. Spot led.

3.14. Router TP-LINK WR840N

Es un dispositivo combinado con cable/inalámbrico, permite conectar redes con diferentes prefijos en su dirección IP, con una velocidad inalámbrica de 300 Mbps. Su servicio es de conectar el mejor camino que enviara datos a cada paquete para llegar al dispositivo de destino.



Figura 14. Router TP-Link WR840N

3.15. Micro SD 32GB

Las tarjetas microSD de Kingston están diseñadas para brindar confiabilidad y son ideales para capturar fotos de una buena resolución en su teléfono inteligente. Esta tarjeta está probada y es duradera, puede llevarla a cualquier lugar sus fotografías, videos y otros datos que estarán protegidos.



Figura 15. Memoria Sd Micro 32gb [5]

3.16. Fuente de Alimentación EBC-HQ 12 VDC 4.2 AMP

Este dispositivo su función es de transformar una corriente alterna AC a una corriente continua DC, cuenta de una entrada de voltaje máximo de 240VAC y con una entrada de voltaje mínima de 110VAC.

Cuenta con un voltaje máximo de 12 VDC de salida, con una corriente de 4.2A y una salida máxima de 50W de potencia.



Figura 16. Fuente de alimentación EBH-HQ 12 VDC 4.2 AMP.

3.17. Breaker 2X25 AMP EBC

Un breaker protege un circuito eléctrico por su excesiva corriente o de un cortocircuito. Su propósito es la de cortar la corriente después de que se detecte un fallo.



Figura 17. Breaker 2X25 AMP EBC.

3.18. Ventilador 120X120X38MM 10V-220V

Es un ventilador negro de material de aluminio pensado para refrigerar carcasas de ordenador y de equipos



Figura 18. Ventilador 120X120X38MM 10V-220V.

3.19. Filtro plástico para ventilador 120MM

El filtro de ventilador de plástico de espuma, evita que la basura y el polvo ingresen en el sistema. Mejorar el rendimiento de refrigeración. Fácil de limpiar y lavable.



Figura 19. Ventilador 120X120X38MM 10V-220V.

3.20. Protector Metálico para ventilador 120MM

Esta es una rejilla de metal que se puede conectar al ventilador, cubre bien el ventilador y no tiene ningún efecto sobre el enfriamiento, protege las aspas y los dedos del ventilador.

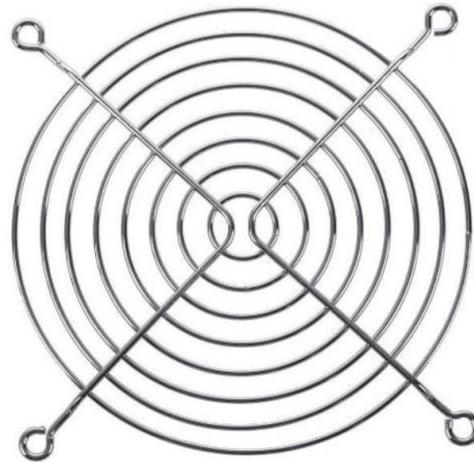


Figura 20. Protector metálico para ventilador 120MM.

3.21. Toma doble polarizado 15A-110V

Es un dispositivo eléctrico, también se lo conoce como enchufe tipo hembra, cuyo objetivo es establecer una buena conexión con un enchufe tipo macho, permitiendo así que la corriente circule de manera eficiente. Los receptáculos generalmente se colocan en una superficie plana o en una pared que se puede empotrar o montar en la superficie.

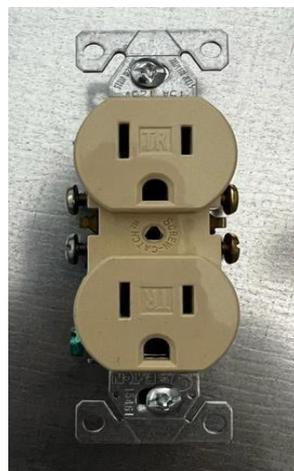


Figura 21. Toma doble polarizado 15A-110V.

3.22. Bornera Riel

Borneras para Riel, facilita asegurar la conexión eléctrica entre 2 conductores de cobre rígido o flexible con o sin terminal.



Figura 22. Bornero riel.

3.23. Luz Piloto Tipo Led

El único propósito de una luz indicadora es advertirnos que un dispositivo eléctrico o electrónico está funcionando. Mientras el dispositivo esté funcionando, la luz indicadora estará encendida para mostrar que la corriente está circulando.



Figura 23. Luz piloto led.

3.24. Cable de poder

El cable de poder o alimentación de un equipo de cómputo es el que nos permite vincularnos directamente con la corriente eléctrica, es primordial que nuestro

equipo reciba electricidad pues si el cable no esté enchufado a la corriente nuestro equipo no encenderá.



Figura 24. Cable de poder

3.25. Gabinete T/Gris fondo de metal 500X700X250CM

Es un plástico que tiene alta resistencia, también es resistente al golpe o impacto, tiene protección anticorrosión, tiene clasificación IP65 y tiene certificación inglesa. Mide 500mmx700mmx250mm e incluye una chapa galvanizada.



Figura 25. Gabinete t/gris fondo metal 500x700x250cm

3.26. Regulador de Voltaje Lm2596

Es un regulador y soporta una corriente de 3A (Step-down), tiene una eficiencia de conversión, excelente regulación, cuenta entrada de voltaje de 4.5 VDC a 40 VDC y una salida de voltaje de 1.23VDC a 37 VDC.

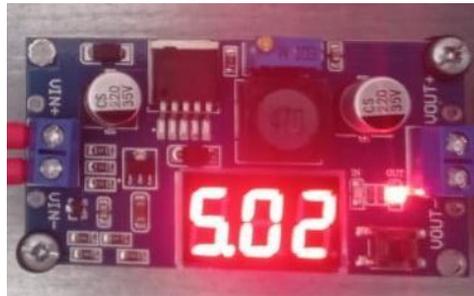


Figura 26. Módulo regulador de voltaje.

3.27. Selector Eléctrico EMAS 2P - ON/OFF

Se utiliza para interrumpir la corriente eléctrica, también se lo conoce conmutador eléctrico y tiene una palanca giratoria, permitiendo situarla en 2 ubicaciones, dependiendo del uso.



Figura 27. Selector eléctrico EMAS 2 posiciones

3.28. Switch ojo de cangrejo 1P – EBC

Es utilizado para encender y apagar cualquier tipo de proyecto electrónico por su gran resistencia a los cambios de corriente y cambios de voltaje.



Figura 28. Switch de cangrejo 1P-

EBC

3.29. Configuración de Red

Se realizaron configuraciones en la red designada al uso del prototipo, la red puede ser vía inalámbrica y vía cable de internet.

Características:

- Marca: TP-LINK
- Modelo: TL-WR840N

El acceso vía WIFI es:

- Nombre de red: TESIS_MENDOZA_QUEVEDO
- Contraseña: 123456789

3.29.1. Acceso al Router para la conexión wifi de la universidad

Ingresar a la dirección de la marca del Router utilizada para el proyecto <http://tplinkwifi.net>.

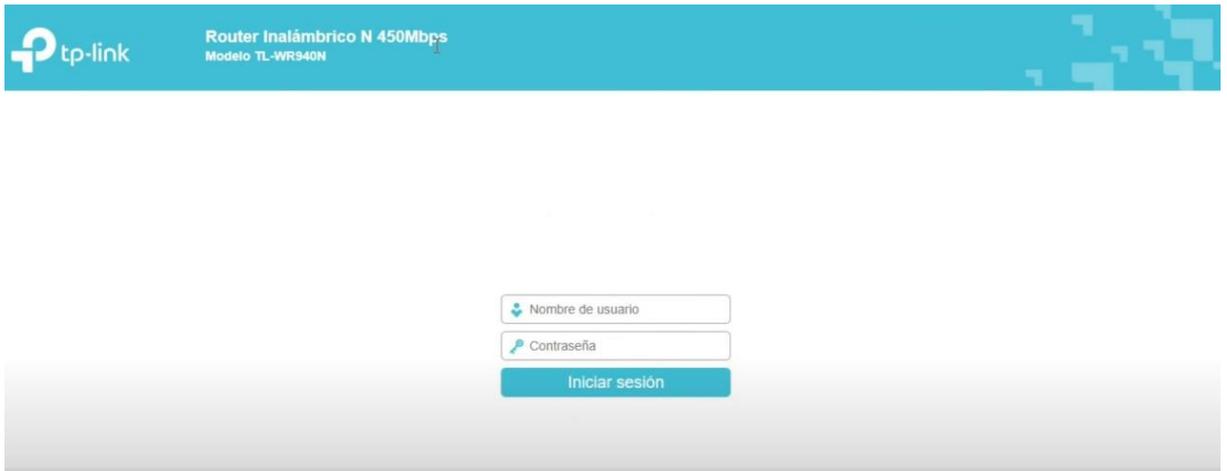


Figura 29. Página del Router TL-WR840N

Para ingresar accedemos las credenciales en la página principal.

- Usuario: admin
- Contraseña: admin

Para el modo de operación del router seleccionamos la opción WISP la cual nos permite hacer uso de la red de la universidad.

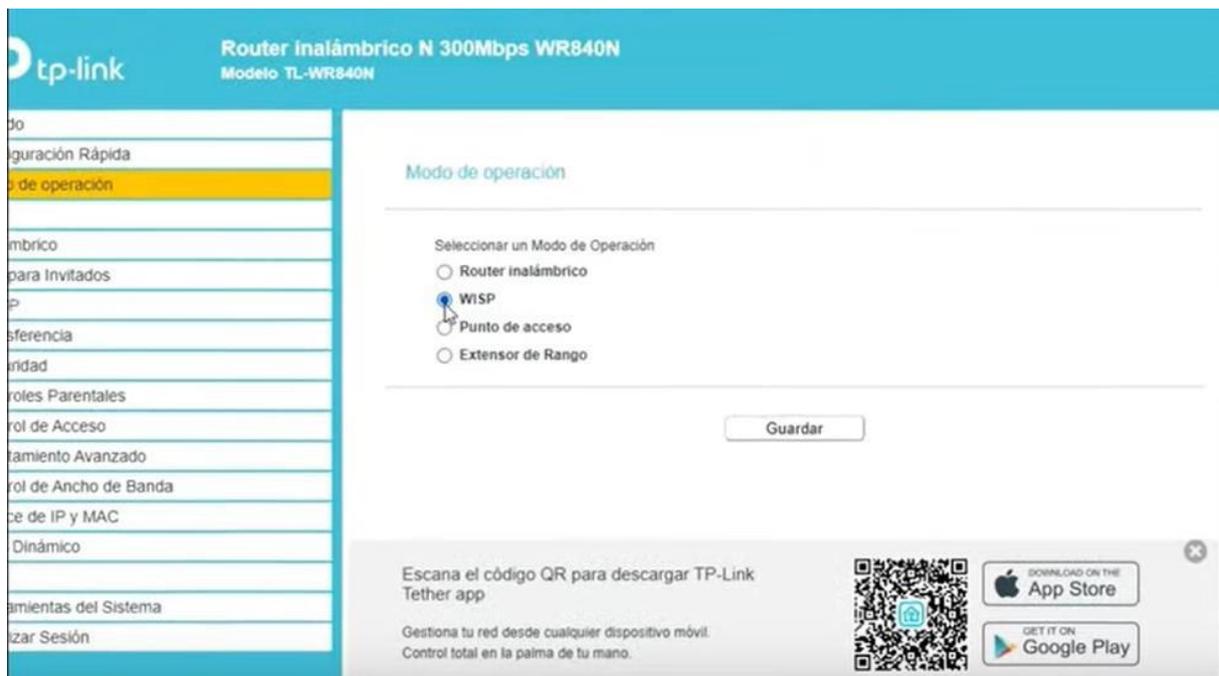


Figura 30. Página de configuración Router TL-WR840N

Después de acceder los datos, se muestra el status vigente en el Router.

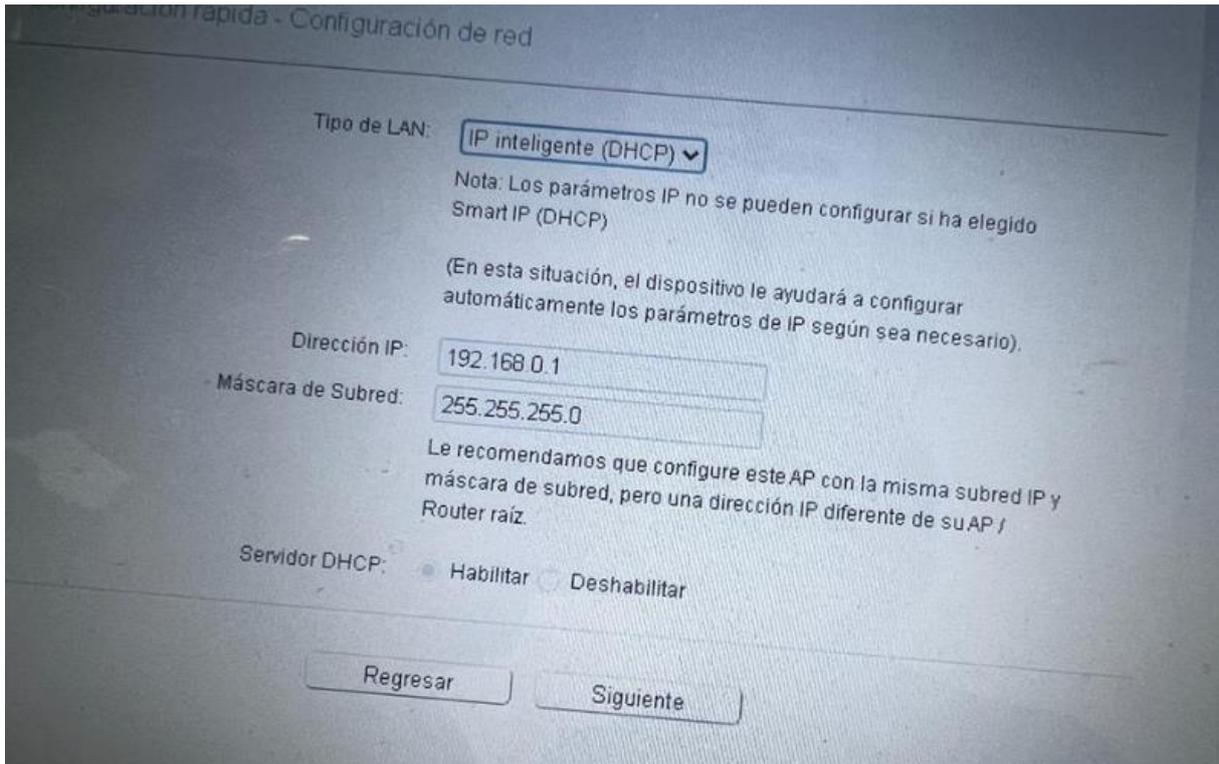


Figura 31. Modo de operación del Router TL-WR840N

Configuración de las variables de la red LAN, para acceder acceso a internet.

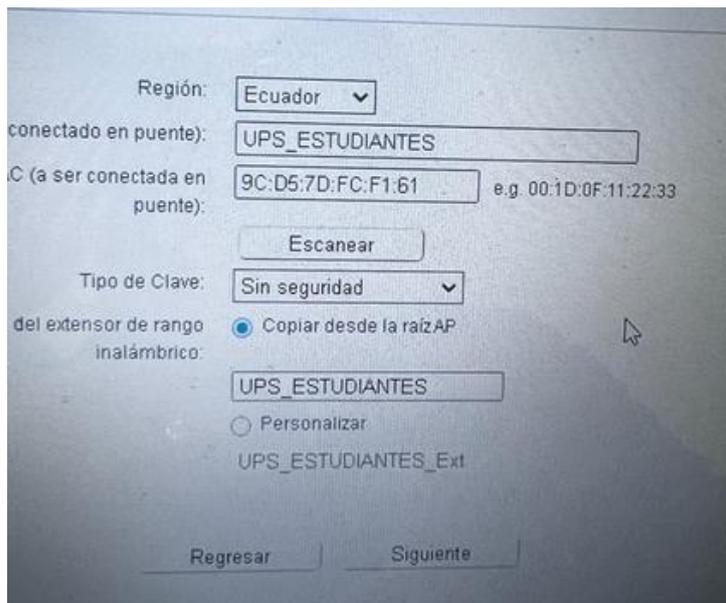


Figura 32. Modo de operación del Router TL-WR840N

Se solicita a la dirección IP con el acceso del internet.

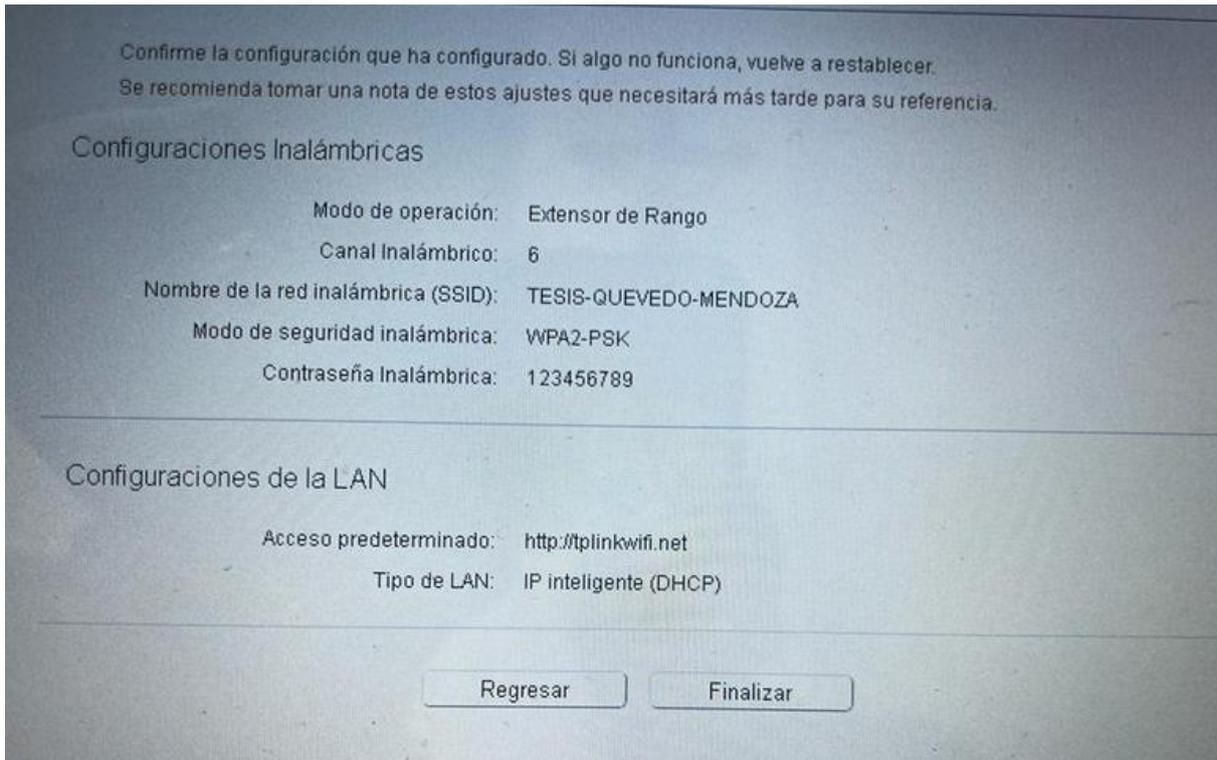


Figura 33. Configuración de la red del Router TL-WR840N

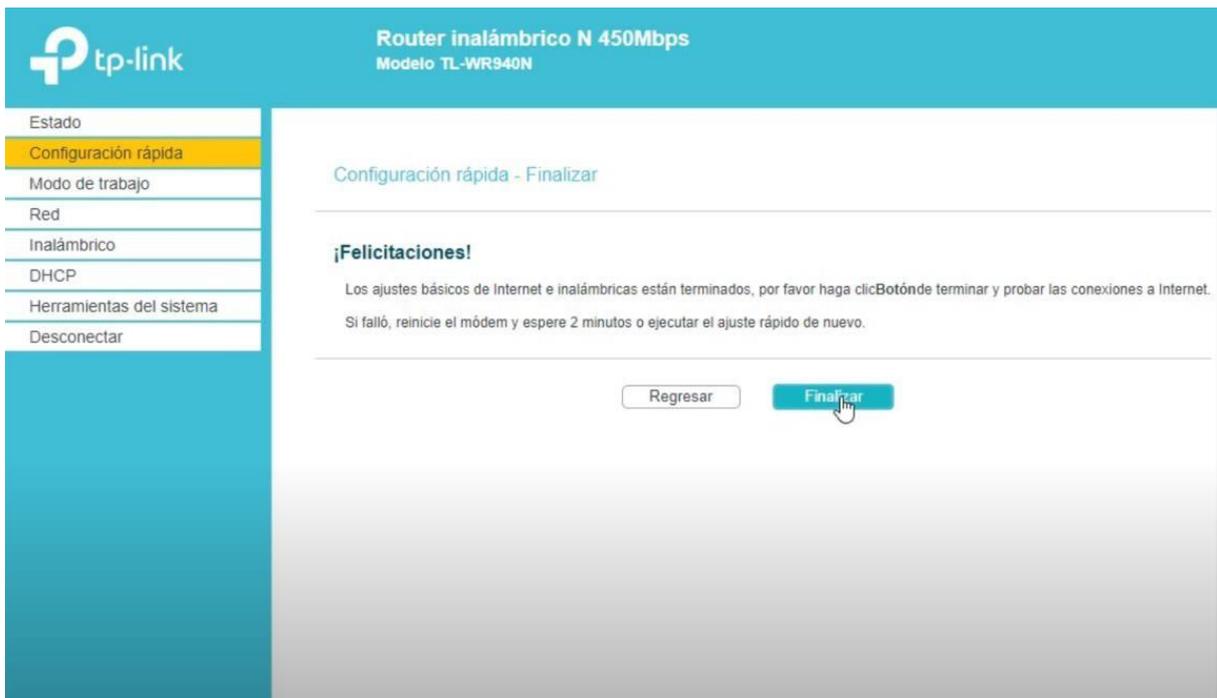


Figura 34. Finalizada la configuración del Router TL-WR840N

3.29.2. Acceso al Router para la conexión wifi de la casa

Ingresar a la dirección de la marca del Router utilizada para el proyecto <http://tplinkwifi.net>.

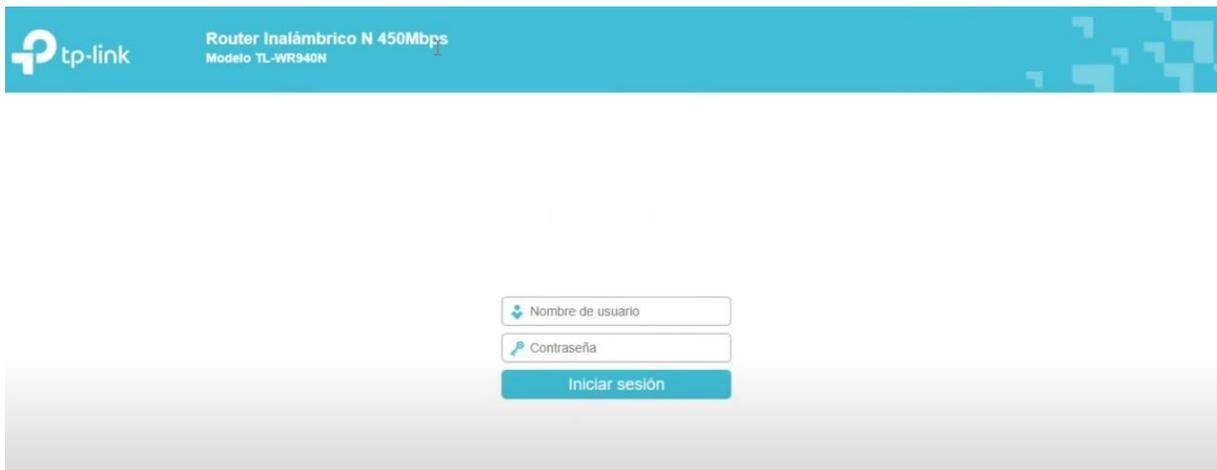


Figura 35. Página del Router TL-WR840N

Para ingresar accedemos las credenciales en la página principal.

- Usuario: admin
- Contraseña: admin

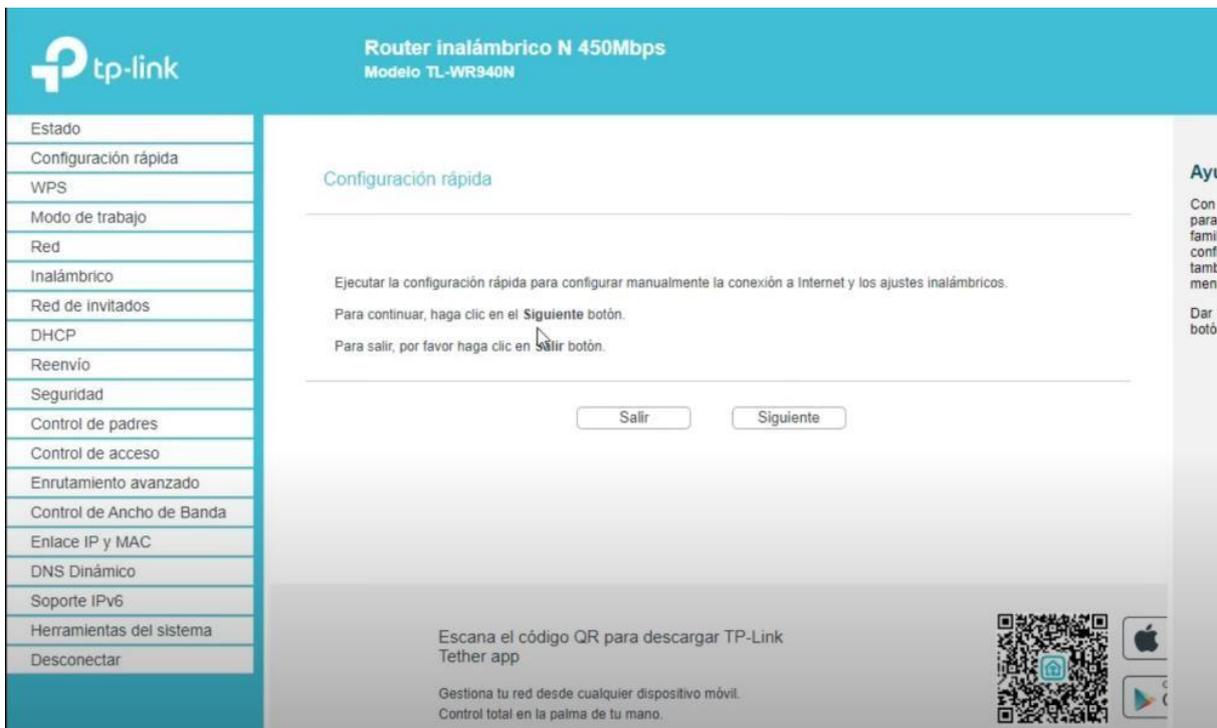


Figura 36. Página de configuración Router TL-WR840N

Después de acceder los datos, se muestra el status vigente en el Router.

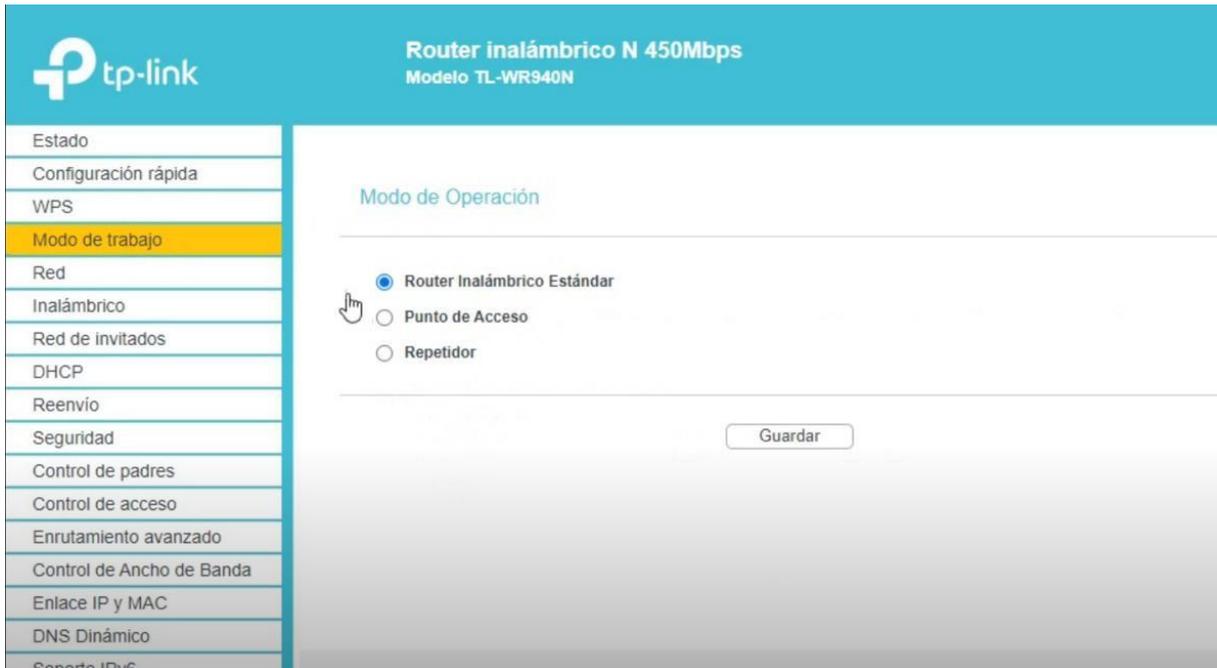


Figura 37. Modo de operación del Router TL-WR840N

Configuración de las variables de la red LAN, para acceder acceso a internet.

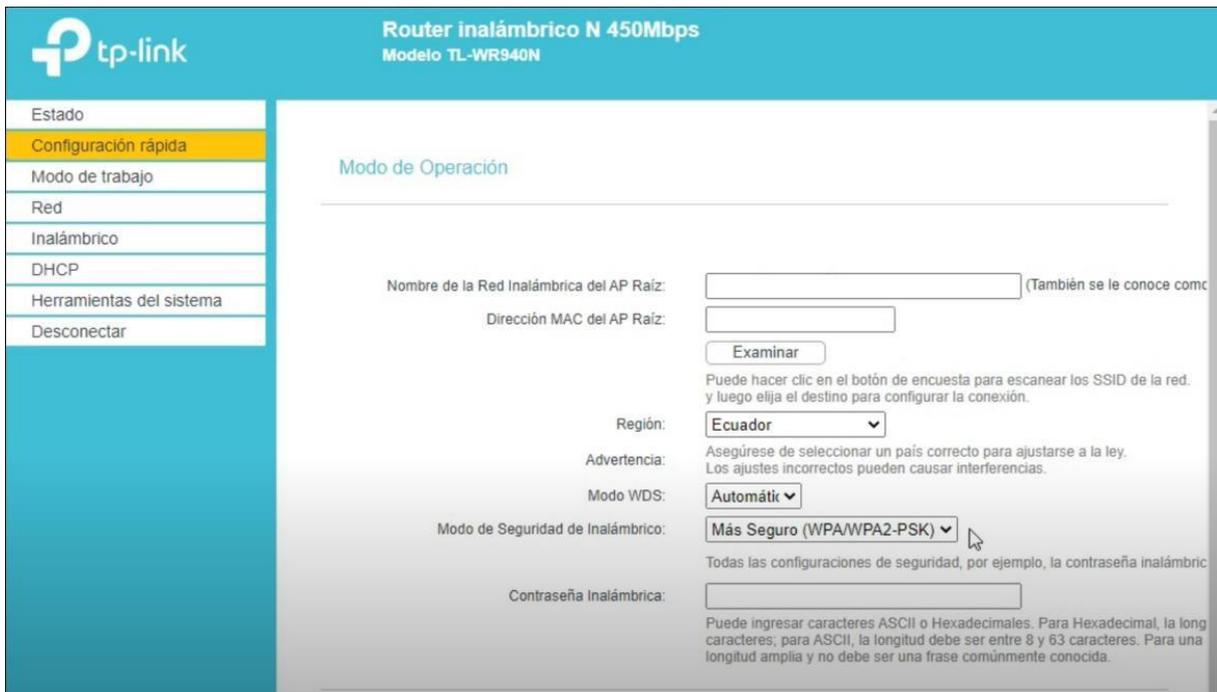


Figura 38. Modo de operación del Router TL-WR840N

Se solicita a la dirección IP con el acceso del internet.

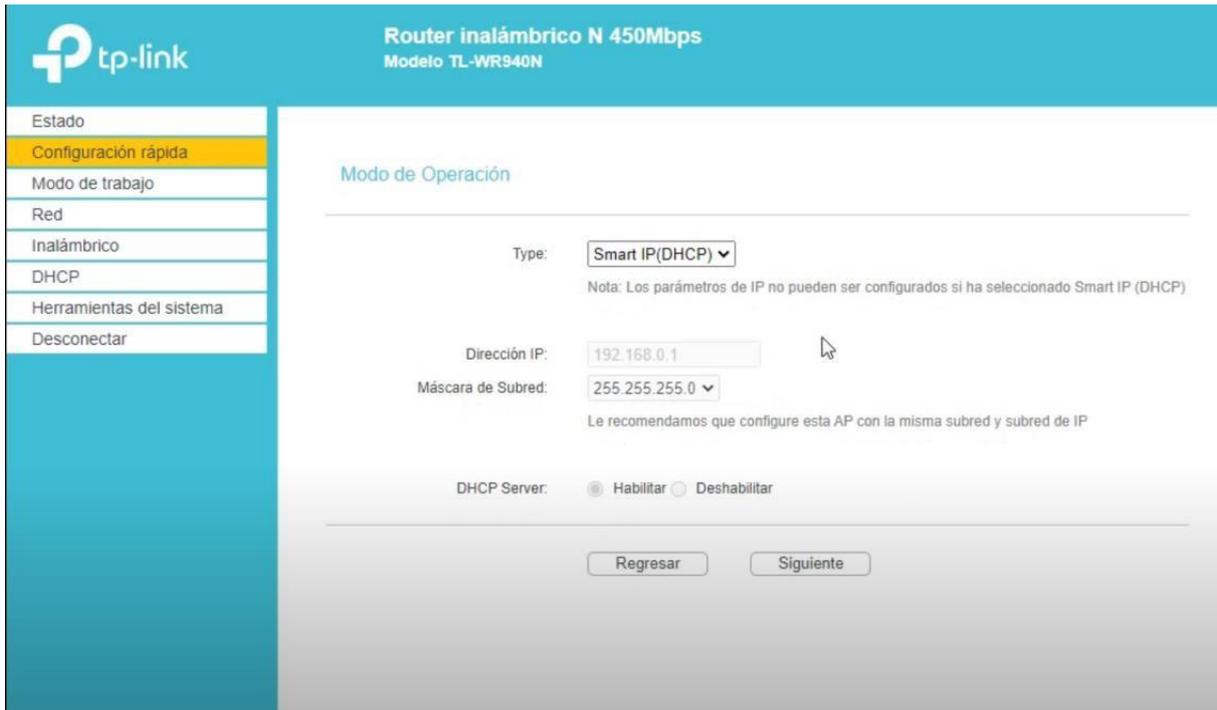


Figura 39. Configuración de la red del Router TL-WR840N

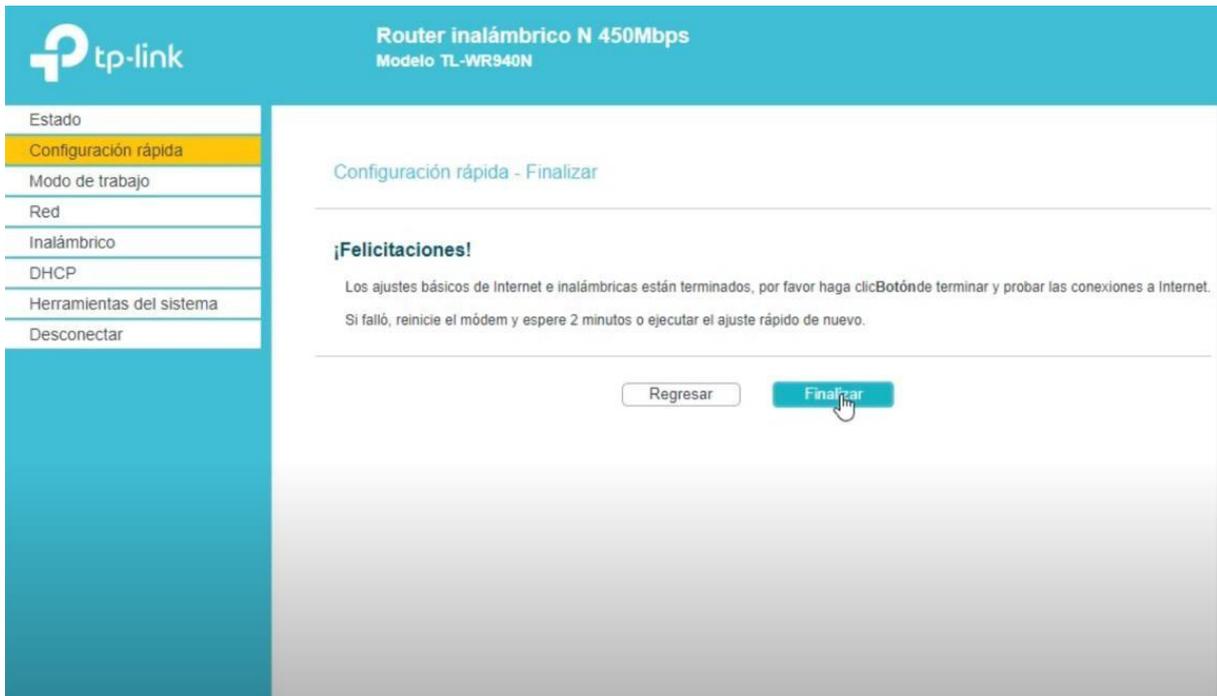


Figura 40. Finalizada la configuración del Router TL-WR840N

3.30. Configuración Raspberry Pi 3

El microcontrolador que se utilizó es el Raspberry Pi 3 con una computadora ARM CórteX de 1,4 Ghz con 1 GB de RAM, WiFi, Bluetooth y Ethernet, es un sistema operativo fácil de trabajar el cual se usó para crear una plataforma de ayuda para controlar el tablero electrónico, haciendo uso del servidor de Adafruit.

3.30.1. Instalación Sistema Operativo Raspberry Pi 3

El método que se utilizó para el funcionamiento del microcontrolador se llama RASPERRY PI OS, el cual es un sistema ligero con muchas prestaciones al momento de trabajar.

Usa una tarjeta SD de 32 GB para el almacenamiento.

Ingresar a la página de Raspberry pi al siguiente enlace <https://www.raspberrypi.com/software/operating-systems/>

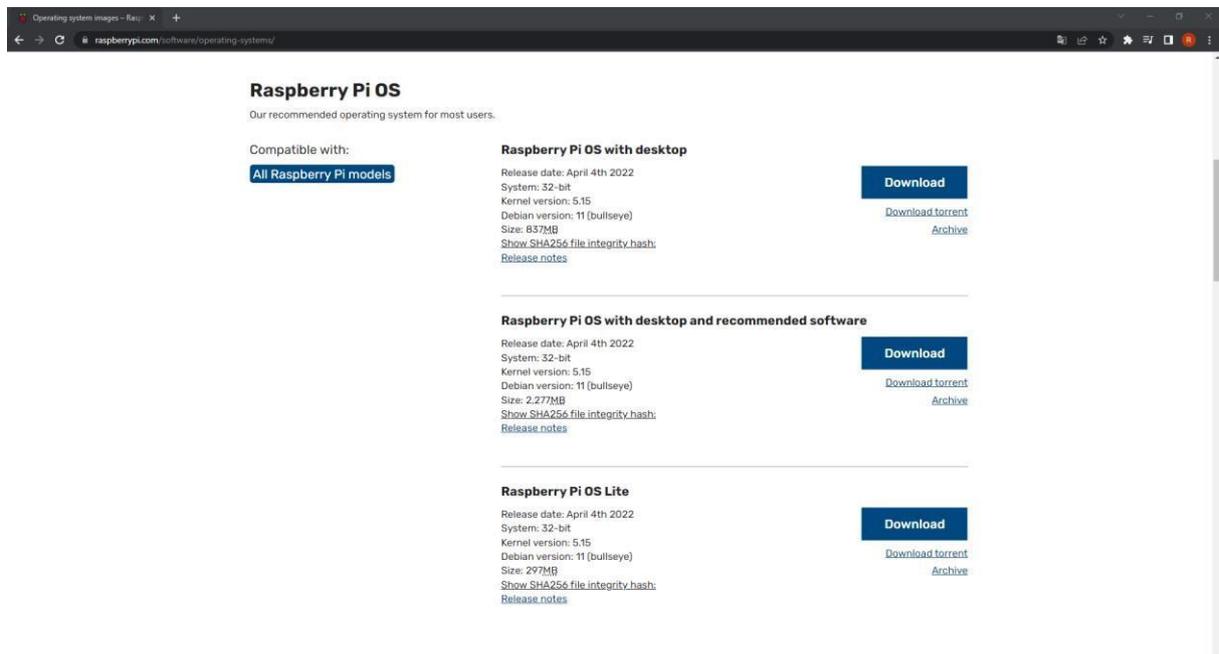


Figura 41. Panel de Raspberry Pi OS

3.30.2. Instalación de Balenaetcher

El servidor de BALENAETCHER es una aplicación que nos ayuda a flashear el software del Raspberry en la tarjeta SD.

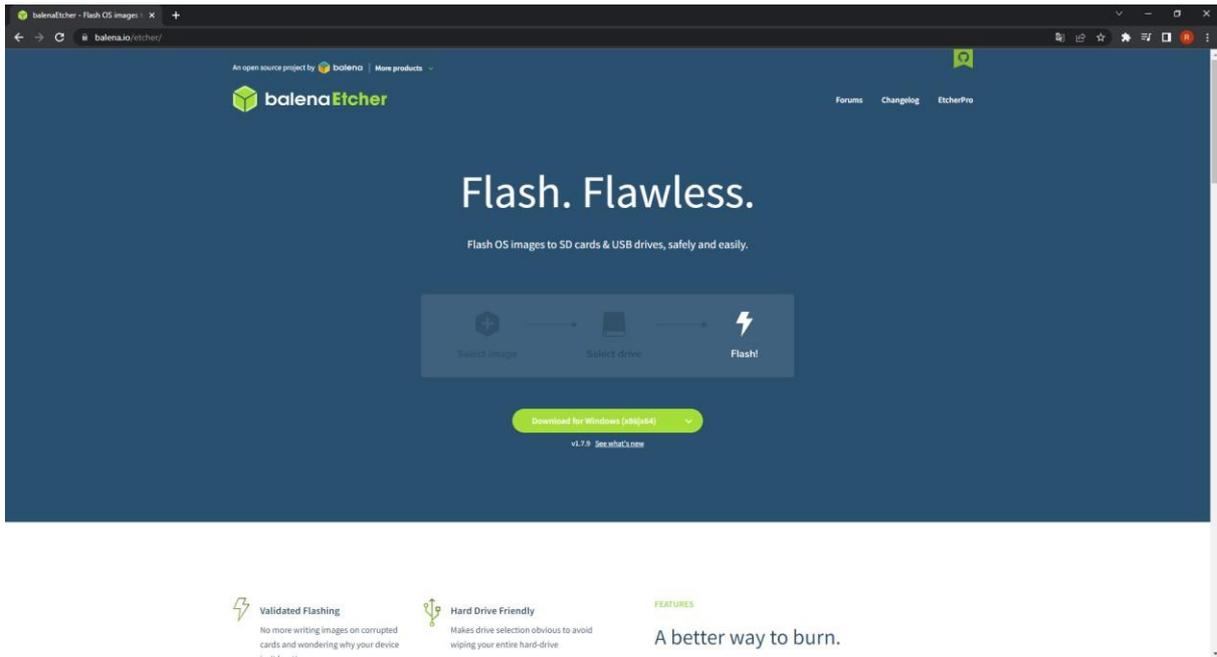


Figura 42. Panel principal Balenaetcher

Lo que hace aplicación es guardar el sistema operativo en la tarjeta SD.

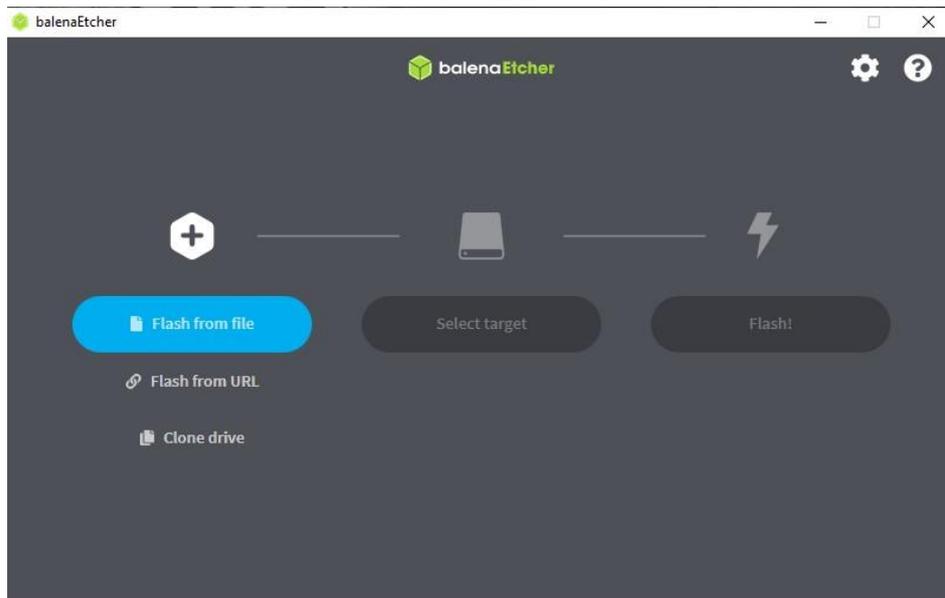


Figura 43. Panel guardado de Baleanetcher

3.31. Ensamblaje y Montaje

Se realizo el montaje de los dispositivos sobre el gabinete fondo de metal 500x700x250mm.



Figura 44. Dispositivos instalados dentro del Gabinete

En la parte frontal del gabinete se instaló los spots led, sensores, luz piloto y se colocó nombre de cada dispositivo.



Figura 45. Dispositivos instalados en la parte Frontal del gabinete

En la parte lateral izquierda se instaló un selector, un switch de ojo de cangrejo y un ventilador.



Figura 46. Dispositivos instalados en la parte lateral izquierda del gabinete

Se realizó las conexiones de todos los dispositivos y cableado en el gabinete.



Figura 47. Conexiones y cableado del gabinete.

3.30.1. Diagrama Eléctrico

El diagrama Eléctrico tiene el propósito de presentar los dispositivos de un circuito eléctrico y nos permite saber las posiciones y las conexiones de una forma muy fácil y sencilla.

El diagrama Eléctrico de fuerza es la alimentación general del circuito, donde van a ir conectado todos los dispositivos instalados del circuito.

El diagrama Eléctrico de control nos permite controlar todo el circuito eléctrico normalmente se utiliza botones, sensores y consume poca corriente.

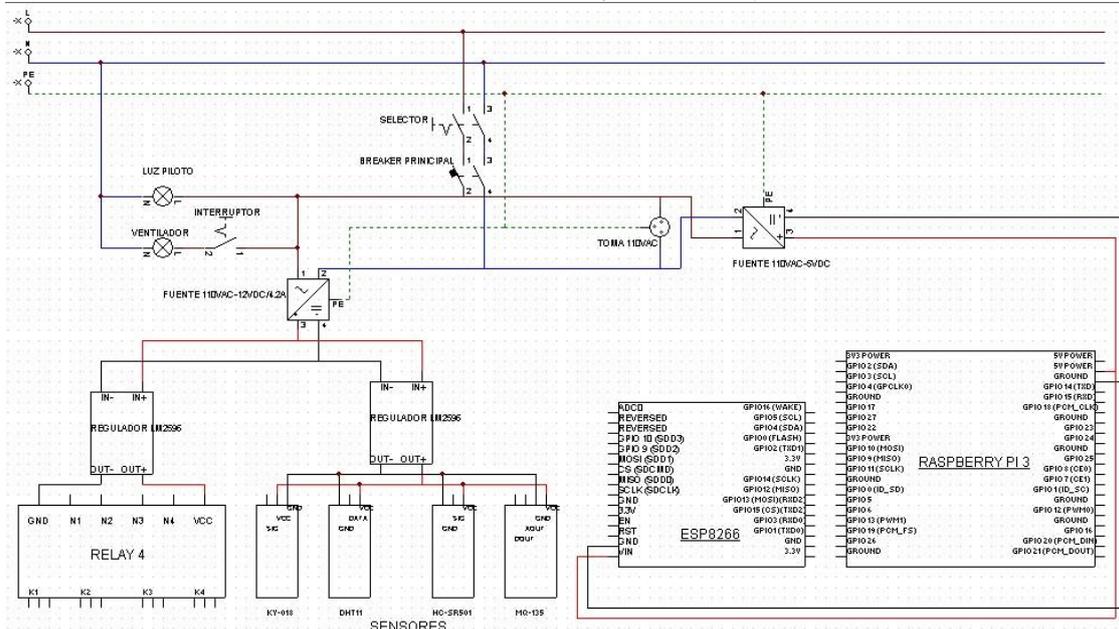


Figura 48. Diagrama eléctrico de fuerza

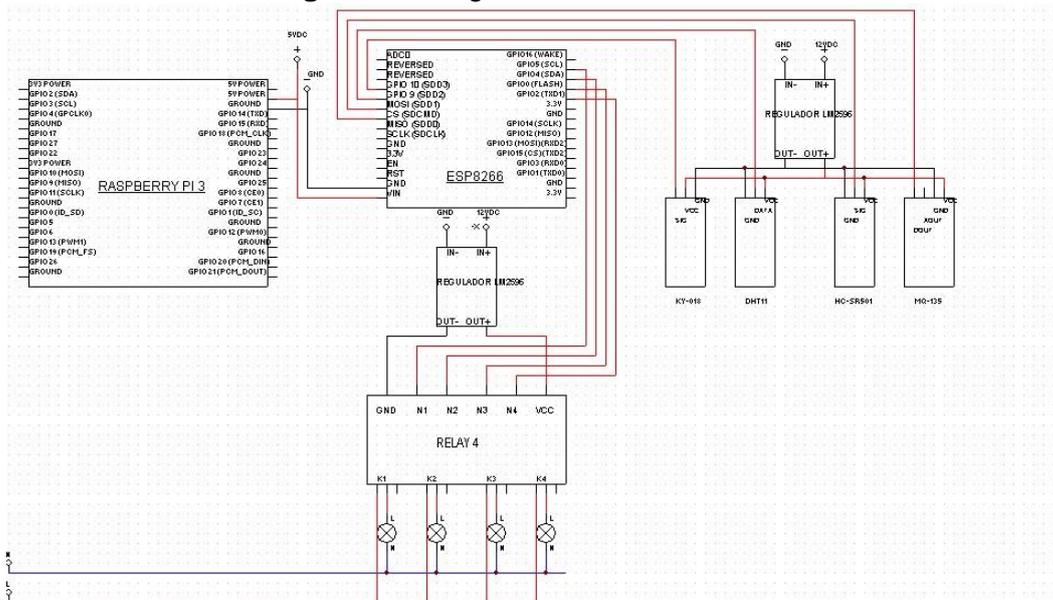


Figura 49. Diagrama eléctrico de Control

4. Resultados

4.1. Practica#1

-

CARRERA:		ASIGNATURA: Domótica
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Taller 1.0 – Comunicaciones de datos a la ESP8266
OBJETIVO: OBJETIVO GENERAL. Implementar un algoritmo en un lenguaje de programación Arduino.		
OBJETIVOS ESPECÍFICOS: - Implementar un programa en Arduino para la implementación al Esp8266		
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	<ol style="list-style-type: none"> 1. Acceder a la carpeta donde se ubicado el código. 2. Ubicar el código al final del documento. 3. Implementar el código puesto en el Anexo del documento. 4. Verificar en el programa Arduino el funcionamiento del código. 	
	<ol style="list-style-type: none"> 5. Responder la tarea a través del AVAC, adjuntando un archivo en Arduino file (extensión. rap) para el ejercicio. Los archivos deben ser almacenados en una carpeta con su NOMBRE_APELLIDO, y adjuntados como respuesta al taller. 	
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)		
<ol style="list-style-type: none"> 1. Realizar el análisis del problema y el diseño de la solución algorítmica. Verificar que se haya tomado en cuenta todas las entradas y salidas que intervienen en la solución. 		
<ol style="list-style-type: none"> 2. Implementar el programa en el lenguaje de programación Arduino. Las entradas, salidas, toma de decisiones y repeticiones, deben ser visibles al momento de la interacción, provisto en el lenguaje. Realizar la revisión del resultado para realizar mejoras, sea que haya presentado errores o no. 		
RESULTADO(S) OBTENIDO(S): Verificación de los resultados a través de la salida por pantalla del programa.		
CONCLUSIONES: Los estudiantes estarán en capacidad de realizar la implementación de un programa, en un lenguaje de programación en Arduino, utilizando apropiadamente el lenguaje previamente utilizado.		
RECOMENDACIONES: Verificar en la ayuda del lenguaje la correcta utilización de variables y expresiones, para poder llevar a cabo la implementación.		

4.2. Practica#2

-

CARRERA:		ASIGNATURA: Domótica
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Taller 1.0 – Conexión con los sensores al ESP8266
OBJETIVO: OBJETIVO GENERAL. Conexión del Router TL-WR840N OBJETIVOS ESPECÍFICOS: - Realizar las conexiones del Router TL-WR840N		
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	1. Ingresar a la plataforma del Router http://tplinkwifi.net Ingresar las credenciales para el acceso a la plataforma. 3. Seguir los pasos de la figura 29. A la figura 34. Para la correcta conexión al wifi de la universidad	
	4. Responder la tarea a través del AVAC, adjuntando un archivo en Arduino file (extensión. rap) para el ejercicio. Los archivos deben ser almacenados en una carpeta con su NOMBRE_APELLIDO, y adjuntados como respuesta al taller.	
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)		
1. Verificar la correcta conexión al internet de la red de la universidad.		
2. Ingresar el nombre con el cual esta configurado el Router en la programación dada.		
RESULTADO(S) OBTENIDO(S): Verificación de los resultados mediante la conexión de los dispositivos a la red del banco de pruebas.		
CONCLUSIONES: Los estudiantes estarán en capacidad de acceder a las diferentes acciones del Router según el uso el cual se le valla a dar.		
RECOMENDACIONES: Verificar la correcta conexión a la red abierta de la universidad.		

4.3. Practica#3

CARRERA:		ASIGNATURA: Domótica
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Taller 4.0 – Interacción del control remoto
OBJETIVO: OBJETIVO GENERAL. Comprobar la comunicación. OBJETIVOS ESPECÍFICOS: - Comprobar la comunicación de la esp8266 hacia la plataforma Adafruit IO. - Comprobar la comunicación de la esp8266 hacia la aplicación Apple Homekit. (opcional)		
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	3. Adjuntar imágenes junto a una descripción de cada uno de los pasos a seguir en la práctica.	
	4. Responder la tarea a través del AVAC, adjuntando un archivo Rar para el ejercicio. Los archivos deben ser almacenados en una carpeta con su NOMBRE_APELLIDO, y adjuntados como respuesta al taller.	
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)		
5. Verificar la conexión mediante la plataforma Adafruit IO siguiendo los pasos en el manual de configuración.		
6. Verificar la conexión mediante la aplicación Apple Homekit siguiendo los pasos en el manual de configuración. (opcional)		
RESULTADO(S) OBTENIDO(S): - Verificación de los resultados a través de la plataforma Adafruit IO. - Verificación de los resultados a través de la aplicación Apple Homekit. (opcional)		
CONCLUSIONES: Los estudiantes estarán en capacidad de realizar la implementación de las aplicaciones seleccionadas.		
RECOMENDACIONES: Verificar la correcta interacción de las plataformas y el módulo domótico.		

4.4. Practica#4

CARRERA:		ASIGNATURA: Domótica
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Taller 1.0 – Conexión de luces y sensores al ESP8266
OBJETIVO: OBJETIVO GENERAL. Conexión y comprobaciones con las luces al ESP8266 Conexión y comprobaciones con los sensores al ESP8266 - OBJETIVOS ESPECÍFICOS: - Realizar las conexiones de las luces mediante el programa Arduino - Realizar las conexiones de los sensores mediante el programa Arduino		
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	1. Comprobar el correcto funcionamiento del código para las luces 2. Comprobar el correcto funcionamiento del código para los sensores 3. Acceder a las plataformas para el acceso de las configuraciones realizadas.	
	4. Responder la tarea a través del AVAC, adjuntando un archivo en Arduino file (extensión. rap) para el ejercicio. Los archivos deben ser almacenados en una carpeta con su NOMBRE_APELLIDO, y adjuntados como respuesta al taller.	
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)		
7. Realizar la conexión de las plataformas al banco de pruebas.		
8. Comprobar la programación de Arduino. Mediante la interacción de el Esp8266 y las plataformas utilizadas.		
RESULTADO(S) OBTENIDO(S): Verificación los resultados a través de las plataformas utilizadas.		
CONCLUSIONES: Los estudiantes estarán en capacidad de interactuar con el banco de pruebas.		
RECOMENDACIONES: Si la conexión del banco de pruebas a las plataformas no se realiza, se recomienda borrar la programación del Esp8266 o a la vez volver a reiniciar todo el banco de pruebas apagando y preniendo el sistema.		

Para instalación de los dispositivos se procedió a empotrar cada uno de ellos en la lámina de metal los dispositivos de control y fuerza y en la parte frontal de la caja los sensores y actuadores del banco de pruebas.

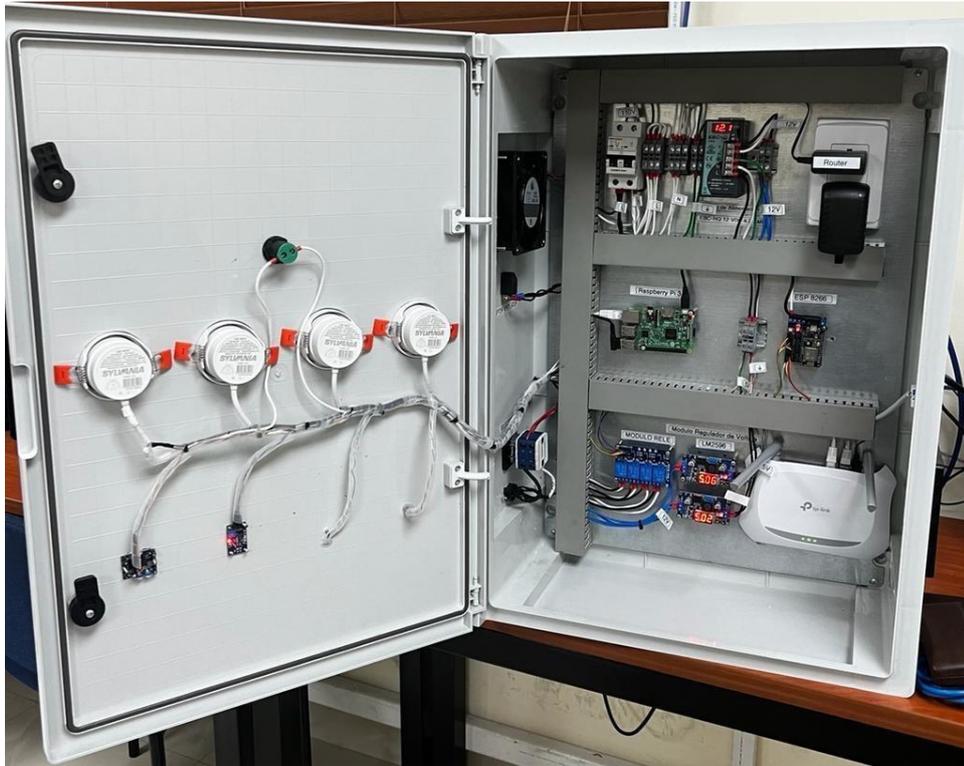


Figura 50. Elementos instalados

4.5. KY-018 Sensor de Luz

Este sensor óptico incluye una foto resistor que es un dispositivo semiconductor fotosensible, tiene una resistencia variable de 1M a 50M aproximadamente.



Figura 51. Sensor de Luz ky-018



Figura 52. Recibiendo datos en Homekit

4.6. DHT11 Sensor de Temperatura y Humedad

Este sensor permite leer valores de temperatura y humedad, opera en un intervalo de medición de temperatura 0 a 50°C y un intervalo de medición de humedad de 20% a 90% RH, la distancia máxima recomendada para la longitud de cable es de 20m.



Figura 53. Sensor DHT11

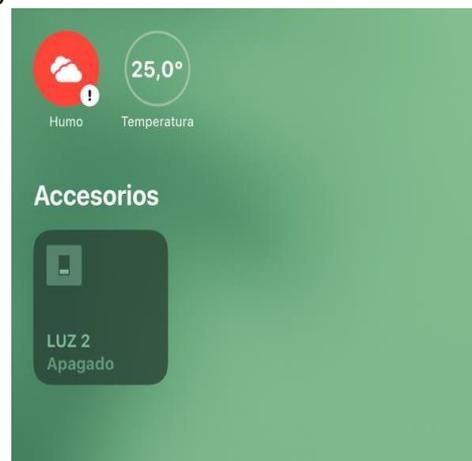


Figura 54. Recibiendo datos en Homekit

4.7. MQ-135 Sensor de Gas

El sensor (MQ-135) nos permite detectar los gases contaminantes como benceno, alcohol, CO2, esto se da por manejo de químicos, detecta concentraciones de gases desde 10ppm hasta 10000 ppm.



Figura 55. Sensor de Gas



Figura 56. Notificación de activación

4.8. HC-SR501 Sensor de Movimiento

El rango que detecta es de 2 metros, nos permite detectar el movimiento.



Figura 57. Sensor HC-SR501



Figura 58. Se detecto movimiento

4.9. Luz de Entrada (Luz 1).

El control domótico de luz de entrada nos permite distribuir la luz artificial en el espacio y momento necesario, son utilizados en espacios residenciales tanto en interiores y exteriores. Como podemos observar en la figura 49 y 50.



Figura 59. Luz entrada



Figura 60. Luz entrada activada

4.10. Luz Cocina (Luz 2).

Como podemos observar en la figura 51 su estado en reposo y en la figura 52 cuando esta activada.



Figura 61. Luz Cocina



Figura 62. Luz Cocina activada

4.11. Luz Dormitorio (Luz 3).

Como podemos observar en la figura 53 su estado en reposo y en la figura 54 cuando esta activada.



Figura 63. Luz Dormitorio

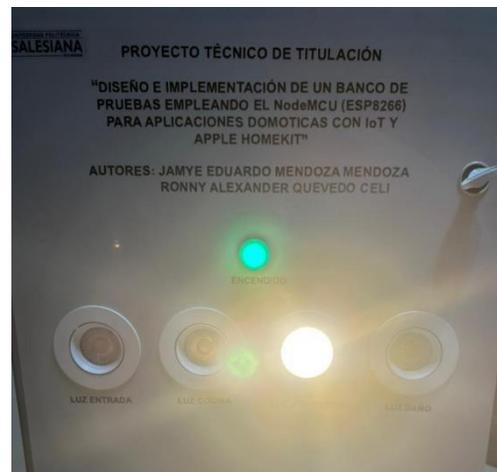


Figura 64. Luz Dormitorio activada

4.12. Luz Baño (Luz 4).

Como podemos observar en la figura 55 su estado en reposo y en la figura 56 cuando esta activada.



Figura 65. Luz baño



Figura 66. Luz baño activada

4.13. Configuración del ESP 8266

El ESP 8266 es usado como microcontrolador programable que nos permite establecer comunicación vía wifi.



Figura 67. Configuración ESP 8266

4.14. Sensor de Luz KY-018

Se uso el sensor KY-018, el cual detecta la luz.

SENSOR	CONEXIÓN
--------	----------

+VCC	+Vout
GROUND	-Vout
DATA	10

Tabla 1. Conexión del sensor KY-018

4.15. Sensor de Temperatura y Humedad (DHT11)

El uso del sensor el cual detecta la humedad del ambiente y la temperatura en cual se está aplicando.

SENSOR	CONEXIÓN
+VCC	+Vout
GROUND	-Vout
DATA	5

Tabla 2. Conexión del sensor DHT11

4.16. Sensor de Gas (MQ-135)

Se uso el sensor, el cual detecta gases contaminantes o perjudiciosos como el amonio, humo y alcohol.

PIN SENSOR	PIN CONEXIÓN
+VCC	+Vout (LM2596)
GROUND	-Vout (LM2596)
DATA	11

Tabla 3. Conexión del sensor MQ-135

4.17. Sensor de Movimiento (HC-SR501)

Se uso el sensor, el cual nos sirve para medir distancias dentro de un rango establecido.

PIN SENSOR	PIN CONEXIÓN
+VCC	+Vout
GROUND	-Vout
DATA	7

Tabla 4. Conexión del sensor HC-SR501

5. Cronograma

Cronograma del proyecto	
Nombre del Proyecto	DISEÑO E IMPLEMENTACION DE UN BANCO DE PRUEBAS EMPLEANDO EL NodeMCU (ESP8266) PARA APLICACIONES DOMOTICOS CON IoT YAPPLE HOMEKIT.

Descripción de actividades	1	2	3	4	5	6
Estudio y diseño del banco de pruebas						
Configuraciones de los equipos a usar						
Armado de banco de prueba.						
Prueba y realización del proyecto.						
Desarrollo de las 5 prácticas.						
Desarrollo de la documentación (TESIS).						

Tabla 5: Cronograma del proyecto

6. Presupuesto

Descripción	Cantidad	Precio de cantidad	Total
Gabinete T/Gris Fondo Metal 500X700X250cm	1	119.00\$	119.00\$

Canaleta ranurada plástica 40X40mm	1	7.30\$	7.30\$
Riel metálico perforada	1	3.00\$	3.00\$
Raspberry Pi 3	1	125.00\$	125.00\$
Micro SD 32GB	1	8.00\$	8.00\$
Adaptador	1	10.00\$	10.00\$
Fuente de alimentación EBCHQ 12 VDC 4.2 AMP	1	55.97\$	55.97\$
Breaker 2x25 AMP EBC	1	8.95\$	8.95\$
Ventilador 120x120x38mm 10v220v	1	18.55\$	18.55\$
Filtro plástico para ventilador 120mm	1	2.40\$	2.40\$
Protector metálico para ventilador 120mm	1	1.90\$	1.90\$
Toma doble polarizado 15A-110V	1	2.65\$	2.65\$
Bornera Riel	10	0.83\$	8.30\$
Luz piloto tipo led	1	3.18\$	3.18\$
Spot led	4	2.67	10.68\$
Cable de poder	1	3.00\$	3.00\$
Modulo regulador voltaje LM2596 (STEP-DOWN)	2	5.50\$	11.00\$
Interruptor EMAS 3P – On-Off	1	14.80\$	14.80\$
Switch ojo de cangrejo 1P - EBC	1	3.18\$	3.18\$
Total			416.86\$

Tabla 6: Presupuesto de proyecto

7. Conclusiones

- Un sistema de sensores y actuadores con la plataforma NodeMCU ESP8266, al momento de diseñar un prototipo es una opción viable, por lo que aporta en sus costos accesibles y beneficiosos.

- Se verifico que el Raspberry Pi 3 logro una buena ejecución de tarea como conectarse al internet que haría cualquier ordenador dentro de su sistema operativo.
- Se logró establecer comunicación al ESP8266 con la Aplicación Homekit.
- Se comprobó que la pagina Adafruit es una buena opción para la comunicación con el sistema de sensores y actuadores.
- Se configuro y utilizo un Router independiente para evitar cualquier tipo de problema en la red.
- Este prototipo didáctico es una buena herramienta factible para un monitoreo sencillo y en tiempo real.

8. Recomendaciones

- Para realizar conexiones o desconexiones eléctricas en el prototipo se deberá trabajar sin energía eléctrica para seguridad y con las protecciones debidas.
- Utilizar en el Raspberry Pi 3 una memoria SD de 32 GB o mayor para no tener problemas de almacenamiento.
- Mantener una ventilación adecuada al prototipo para así evitar temperaturas elevadas.
- Dar mantenimiento periódico a los dispositivos instalados en el prototipo para que no tenga fallas cuando esté funcionando.
- Tener una conexión estable de internet para tener una excelente comunicación.

9. Referencias Bibliográficas

- [1] F. Andrade, «Cablecom,» 5 2 2020. [En línea]. Available: <https://www.cablecom.com.ec/post/el-internet-de-las-cosas-iot-y-c%C3%B3mofunciona>. [Último acceso: 7 6 2022].
- [2] «Babara,» 16 4 2021. [En línea]. Available: <https://barbaraiot.com/blog/plataformas-iot-que-son-y-como-pueden-beneficiar-a-tu-empresa/>. [Último acceso: 12 6 2022].
- [3] P. Domingo, «Didactronica,» 13 3 2019. [En línea]. Available: <https://didactronica.com/2200-2/>. [Último acceso: 12 9 2022].
- [4] «El Blog de Chito de Tecnología,» [En línea]. Available: <https://chitoraspberrypi.blogspot.com/p/diagramas.html>. [Último acceso: 30 6 2022].
- [5] «Aquae fundacion,» [En línea]. Available: <https://www.fundacionaquae.org/wiki/sabesarduinosirve/#:~:text=%C2%BFPara%20qu%C3%A9%20sirve%20un%20Arduino,hardware%20como%20con%20el%20software..> [Último acceso: 12 9 2022].
- [6] M. López, «Applesfera,» 17 3 2020. [En línea]. Available: <https://www.applesfera.com/accesorios/homekit-trucos-consejos-que-puedes-aplicartienes-tiempo-para-domotizar-tu-hogar>. [Último acceso: 12 9 2022].
- [7] «Macnificos,» 10 10 2018. [En línea]. Available: <https://www.macnificos.com/blog/apple-homekit/reviews/domotica>. [Último acceso: 13 9 2022].
- [8] L. d. V. Hernández, «Programarfacil,» [En línea]. Available: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>. [Último acceso: 20 6 2022].
- [9] R. Solé, «ProfesionalReview,» 18 7 2021. [En línea]. [Último acceso: 1 7 2022]. [1 «Unit Electronics,» [En línea]. Available: <https://uelectronics.com/producto/modulo-ky018-sensor-foto-resistor/>. [Último acceso: 15 7 2022].
- [1 «IDC,» [En línea]. Available: <https://www.idcmayoristas.com/producto/memoria-sd1-micro-32gb-kingston-clase-10/>. [Último acceso: 2 7 2022].

10. Anexos

10.1. Anexo 1. Código ESP 8266

```
Tesis my_accessory.c wifi_info.h
#include <arduino_homekit_server.h>
#include "wifi_info.h"
//#include <DHT.h>

#define LOG_D(fmt, ...) printf_P(PSTR(fmt "\n"), ##__VA_ARGS__);

//DECLARACION PARA LOS PINES DEL NODEMCU
#define pin_mq 12 //D6
#define sensor_luz A0 //A0
#define pin_mov 13 //D7
//#define DHTTYPE DHT11
//#define DHTPin 14 //D5
//DHT dht(DHTPin, DHTTYPE);
int L1;
int L2;
int L3;
int L4;

void setup() {
  wifi_connect(); // in wifi_info.h
  pinMode(pin_mq, INPUT);
  pinMode(pin_mov, INPUT);
  //pinMode(DHTPin, INPUT);
  //dht.begin();
  my_homekit_setup();
}

void loop() {
  my_homekit_loop();
  delay(10);
}
```

Declaración de pines del NodeMCU

```

//LLAMADO DE VARIABLES ANTES DECLARADAS
extern "C" homekit_server_config_t config;
extern "C" homekit_characteristic_t cha_switch_on;
extern "C" homekit_characteristic_t cha_switch_on2;
extern "C" homekit_characteristic_t cha_switch_on3;
extern "C" homekit_characteristic_t cha_switch_on4;
extern "C" homekit_characteristic_t cha_temperature;
extern "C" homekit_characteristic_t cha_humidity;
extern "C" homekit_characteristic_t cha_motion;
extern "C" homekit_characteristic_t cha_light;
extern "C" homekit_characteristic_t cha_smoke;

static uint32_t next_heap_millis = 0;
static uint32_t next_report_millis = 0;
static uint32_t t_report = 0;

```

Llamado de variables antes declaradas.

```

//DECLARACION PARA LOS PINES DEL NODEMCU
#define PIN_SWITCH 16
#define PIN_SWITCH2 5
#define PIN_SWITCH3 4
#define PIN_SWITCH4 0

// Called when the value is read by iOS Home APP
homekit_value_t cha_programmable_switch_event_getter() {
    // Should always return "null" for reading, see HAP section 9.75
    return HOMEKIT_NULL_CPP();
}

```

Declaración para los pines del NodeMCU.

```

//ACTIVACION DE LA SALIDA LUZ 1 MEDIANTE LA APLICACION
void cha_switch_on_setter(const homekit_value_t value) {
    bool on = value.bool_value;
    cha_switch_on.value.bool_value = on; //sync the value
    LOG_D("LUZ1: %s", on ? "ON" : "OFF");
    digitalWrite(PIN_SWITCH, on ? LOW : HIGH);
}

//ACTIVACION DE LA SALIDA LUZ 2 MEDIANTE LA APLICACION
void cha_switch_on_setter2(const homekit_value_t value) {
    bool on = value.bool_value;
    cha_switch_on2.value.bool_value = on; //sync the value
    LOG_D("LUZ2: %s", on ? "ON" : "OFF");
    digitalWrite(PIN_SWITCH2, on ? LOW : HIGH);
}

```

```

//ACTIVACION DE LA SALIDA LUZ 3 MEDIANTE LA APLICACION
void cha_switch_on_setter3(const homekit_value_t value) {
    bool on = value.bool_value;
    cha_switch_on3.value.bool_value = on; //sync the value
    LOG_D("LUZ3: %s", on ? "ON" : "OFF");
    digitalWrite(PIN_SWITCH3, on ? LOW : HIGH);
}
//ACTIVACION DE LA SALIDA LUZ 4 MEDIANTE LA APLICACION
void cha_switch_on_setter4(const homekit_value_t value) {
    bool on = value.bool_value;
    cha_switch_on4.value.bool_value = on; //sync the value
    LOG_D("LUZ4: %s", on ? "ON" : "OFF");
    digitalWrite(PIN_SWITCH4, on ? LOW : HIGH);
}

```

Activación de salida de luces.

```

void my_homekit_setup() {
    //DECLARACION DE LAS LUCES COMO SALIDA
    pinMode(PIN_SWITCH, OUTPUT);
    digitalWrite(PIN_SWITCH, HIGH);
    pinMode(PIN_SWITCH2, OUTPUT);
    digitalWrite(PIN_SWITCH2, HIGH);
    pinMode(PIN_SWITCH3, OUTPUT);
    digitalWrite(PIN_SWITCH3, HIGH);
    pinMode(PIN_SWITCH4, OUTPUT);
    digitalWrite(PIN_SWITCH4, HIGH);

    cha_switch_on.setter = cha_switch_on_setter;
    cha_switch_on2.setter = cha_switch_on_setter2;
    cha_switch_on3.setter = cha_switch_on_setter3;
    cha_switch_on4.setter = cha_switch_on_setter4;
    arduino_homekit_setup(&config);
}

```

Declaración de las luces como salida.

```

    //LECTURA DE LOS VALORES DE LOS SENSORES CADA 5 SEGUNDOS
    next_report_millis = t + 5 * 1000;
    my_homekit_report();
}
if (t > next_heap_millis) {
    next_heap_millis = t + 5 * 1000;
    LOG_D("Free heap: %d, HomeKit clients: %d",
        ESP.getFreeHeap(), arduino_homekit_connected_clients_count());
}
}
}

void my_homekit_report() {
    float t = 25;//dht.readTemperature(); //temperatura
    float h = 30;//dht.readHumidity(); //humedad
    float l = map(analogRead(sensor_luz),1023,0,0,100); //luz
    bool m = digitalRead(pin_mov); //movimiento
    bool s = !digitalRead(pin_mq); //humo
    bool L1 = digitalRead(PIN_SWITCH);
    bool L2 = digitalRead(PIN_SWITCH2);
    bool L3 = digitalRead(PIN_SWITCH3);
    bool L4 = digitalRead(PIN_SWITCH4);

```

Lectura de los valores de los sensores

```

//ENVIA EL VALOR DEL SENSOR DE TEMPERATURA A LA APLICACION HOMEKIT
cha_temperature.value.float_value = t;
homekit_characteristic_notify(&cha_temperature, cha_temperature.value);

//ENVIA EL VALOR DEL SENSOR DE HUMEDAD A LA APLICACION HOMEKIT
cha_humidity.value.float_value = h;
homekit_characteristic_notify(&cha_humidity, cha_humidity.value);

//ENVIA EL VALOR DEL SENSOR DE MOVIMIENTO A LA APLICACION HOMEKIT
cha_motion.value.bool_value = m;
homekit_characteristic_notify(&cha_motion, cha_motion.value);

//ENVIA EL VALOR DEL SENSOR DE LUZ A LA APLICACION HOMEKIT
cha_light.value.float_value = l;
homekit_characteristic_notify(&cha_light, cha_light.value);

//ENVIA EL VALOR DEL SENSOR DE GAS A LA APLICACION HOMEKIT
cha_smoke.value.bool_value = s;
homekit_characteristic_notify(&cha_smoke, cha_smoke.value);

```

Envío de los valores de los sensores.

```

Serial.begin(115200);
//MUESTRA TODOS LOS DATOS DE LOS SENSORES Y LAS SALIDAS
LOG_D("t %.1f , h %.1f , l %.1f , m %u , s %u , L1 %u , L2 %u , L3 %u , L4 %u", t, h,l, (uint8_t)m,(uint8_t)s,L1,L2,L3,L4);

t = millis();
t_report = t + 3 * 1000;
//LECTURA DE LOS DATOS RECIBIDOS POR LA RASPBERRY
while (t < t_report){
  String dato = Serial.readStringUntil('\n');
  if (dato == "LUZ1 1"){
    L1 = 1;
  }
  if (dato == "LUZ1 0"){
    L1 = 0;
  }
  if (dato == "LUZ2 1"){
    L2 = 1;
  }
  if (dato == "LUZ2 0"){
    L2 = 0;
  }
  if (dato == "LUZ3 1"){
    L3 = 1;
  }
  if (dato == "LUZ3 0"){
    L3 = 0;
  }
  if (dato == "LUZ4 1"){
    L4 = 1;
  }
  if (dato == "LUZ4 0"){
    L4 = 0;
  }
  t = millis();
}
Serial.end();

```

Lectura de los datos del Raspberry.

```

//ACTIVACION DE LAS SALIDAS SEGUN LOS DATOS RECIBIDOS POR LA RASPBERRY
if (L1 == 1){
    digitalWrite(PIN_SWITCH, HIGH);
    cha_switch_on.value.bool_value = false;
    homekit_characteristic_notify(&cha_switch_on, cha_switch_on.value);
}
if (L1 == 0){
    digitalWrite(PIN_SWITCH, LOW);
    cha_switch_on.value.bool_value = true;
    homekit_characteristic_notify(&cha_switch_on, cha_switch_on.value);
}
if (L2 == 1){
    digitalWrite(PIN_SWITCH2, HIGH);
    cha_switch_on2.value.bool_value = false;
    homekit_characteristic_notify(&cha_switch_on2, cha_switch_on2.value);
}
if (L2 == 0){
    digitalWrite(PIN_SWITCH2, LOW);
    cha_switch_on2.value.bool_value = true;
    homekit_characteristic_notify(&cha_switch_on2, cha_switch_on2.value);
}
if (L3 == 1){
    digitalWrite(PIN_SWITCH3, HIGH);
    cha_switch_on3.value.bool_value = false;
    homekit_characteristic_notify(&cha_switch_on3, cha_switch_on3.value);
}
if (L3 == 0){
    digitalWrite(PIN_SWITCH3, LOW);
    cha_switch_on3.value.bool_value = true;
    homekit_characteristic_notify(&cha_switch_on3, cha_switch_on3.value);
}

if (L4 == 1){
    digitalWrite(PIN_SWITCH4, HIGH);
    cha_switch_on4.value.bool_value = false;
    homekit_characteristic_notify(&cha_switch_on4, cha_switch_on4.value);
}
if (L4 == 0){
    digitalWrite(PIN_SWITCH4, LOW);
    cha_switch_on4.value.bool_value = true;
    homekit_characteristic_notify(&cha_switch_on4, cha_switch_on4.value);
}

}

int random_value(int min, int max) {
    return min + random(max - min);
}

```

Activación de las salidas según los datos recibidos por la RASPBERRY.

10.1.1. Accesorios

```
#include <homekit/homekit.h>
#include <homekit/characteristics.h>

void my_accessory_identify(homekit_value_t _value) {
    printf("accessory identify\n");
}

homekit_characteristic_t cha_switch_on = HOMEKIT_CHARACTERISTIC_(ON, false); //DECLARACION DE LA VARIABLE LUZ 1
homekit_characteristic_t cha_switch_on2 = HOMEKIT_CHARACTERISTIC_(ON, false); //DECLARACION DE LA VARIABLE LUZ 2
homekit_characteristic_t cha_switch_on3 = HOMEKIT_CHARACTERISTIC_(ON, false); //DECLARACION DE LA VARIABLE LUZ 3
homekit_characteristic_t cha_switch_on4 = HOMEKIT_CHARACTERISTIC_(ON, false); //DECLARACION DE LA VARIABLE LUZ 4
homekit_characteristic_t cha_temperature = HOMEKIT_CHARACTERISTIC_(CURRENT_TEMPERATURE, 1); //DECLARACION DE LA VARIABLE SENSOR TEMPERATURA
homekit_characteristic_t cha_humidity = HOMEKIT_CHARACTERISTIC_(CURRENT_RELATIVE_HUMIDITY, 1); //DECLARACION DE LA VARIABLE SENSOR HUMEDAD
homekit_characteristic_t cha_motion = HOMEKIT_CHARACTERISTIC_(MOTION_DETECTED, false); //DECLARACION DE LA VARIABLE SENSOR MOVIMIENTO
homekit_characteristic_t cha_light = HOMEKIT_CHARACTERISTIC_(CURRENT_AMBIENT_LIGHT_LEVEL, 1); //DECLARACION DE LA VARIABLE SENSOR LUZ
homekit_characteristic_t cha_smoke = HOMEKIT_CHARACTERISTIC_(SMOKE_DETECTED, 1); //DECLARACION DE LA VARIABLE SENSOR GAS

homekit_characteristic_t cha_name = HOMEKIT_CHARACTERISTIC_(NAME, "TESIS");

homekit_accessory_t *accessories[] = {
    // INICIALIZACION DE VARIABLE LUZ 1 EN LA APLICACION HOMEKIT
    HOMEKIT_ACCESSORY(.id=1, .category=homekit_accessory_category_switch, .services=(homekit_service_t*[]) {

        HOMEKIT_SERVICE(AccessoryInformation, .characteristics=(homekit_characteristic_t*[]) {
            HOMEKIT_CHARACTERISTIC(NAME, "LUZ 1"),
            HOMEKIT_CHARACTERISTIC(MANUFACTURER, "Arduino HomeKit"),
            HOMEKIT_CHARACTERISTIC(SERIAL_NUMBER, "0123456"),
            HOMEKIT_CHARACTERISTIC(MODEL, "ESP8266/ESP32"),
            HOMEKIT_CHARACTERISTIC(FIRMWARE_REVISION, "1.0"),
            HOMEKIT_CHARACTERISTIC(Identify, my_accessory_identify),
            NULL
        }
    }
},
```

Inicialización de variables Luz 1.

```
// INICIALIZACION DE VARIABLE LUZ 2 EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=2, .category=homekit_accessory_category_switch, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "LUZ 2"),
        HOMEKIT_CHARACTERISTIC(MANUFACTURER, "Arduino HomeKit"),
        HOMEKIT_CHARACTERISTIC(SERIAL_NUMBER, "0123456"),
        HOMEKIT_CHARACTERISTIC(MODEL, "ESP8266/ESP32"),
        HOMEKIT_CHARACTERISTIC(FIRMWARE_REVISION, "1.0"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }),
    HOMEKIT_SERVICE(SWITCH, .primary=true, .characteristics=(homekit_characteristic_t*[]){
        &cha_switch_on2,
        &cha_name,
        NULL
    }),
    NULL
}),
}),
```

Inicialización de variables Luz 2.

```
// INICIALIZACION DE VARIABLE LUZ 3 EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=3, .category=homekit_accessory_category_switch, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "LUZ 3"),
        HOMEKIT_CHARACTERISTIC(MANUFACTURER, "Arduino HomeKit"),
        HOMEKIT_CHARACTERISTIC(SERIAL_NUMBER, "0123456"),
        HOMEKIT_CHARACTERISTIC(MODEL, "ESP8266/ESP32"),
        HOMEKIT_CHARACTERISTIC(FIRMWARE_REVISION, "1.0"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }),
    HOMEKIT_SERVICE(SWITCH, .primary=true, .characteristics=(homekit_characteristic_t*[]){
        &cha_switch_on3,
        &cha_name,
        NULL
    }),
    NULL
}),
}),
```

Inicialización de variables Luz 3.

```

// INICIALIZACION DE VARIABLE LUZ 4 EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=4, .category=homekit_accessory_category_switch, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "LUZ 4"),
        HOMEKIT_CHARACTERISTIC(MANUFACTURER, "Arduino HomeKit"),
        HOMEKIT_CHARACTERISTIC(SERIAL_NUMBER, "0123456"),
        HOMEKIT_CHARACTERISTIC(MODEL, "ESP8266/ESP32"),
        HOMEKIT_CHARACTERISTIC(FIRMWARE_REVISION, "1.0"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }
}),
HOMEKIT_SERVICE(SWITCH, .primary=true, .characteristics=(homekit_characteristic_t*[]){
    &cha_switch_on4,
    &cha_name,
    NULL
}),
    NULL
}),

```

Inicialización de variables Luz 4.

```

// INICIALIZACION DEL SENSOR DE TEMPERATURA EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=5, .category=homekit_accessory_category_sensor, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Temperature Sensor"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }
}),
HOMEKIT_SERVICE(TEMPERATURE_SENSOR, .primary=true, .characteristics=(homekit_characteristic_t*[]) {
    HOMEKIT_CHARACTERISTIC(NAME, "Temperature"),
    &cha_temperature,
    NULL
}),
    NULL

```

Inicialización del sensor de temperatura.

```

// INICIALIZACION DEL SENSOR DE HUMEDAD EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=6, .category=homekit_accessory_category_sensor, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Humidity Sensor"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }
}),
HOMEKIT_SERVICE(HUMIDITY_SENSOR, .primary=true, .characteristics=(homekit_characteristic_t*[]) {
    HOMEKIT_CHARACTERISTIC(NAME, "Humidity"),
    &cha_humidity,
    NULL
}),
NULL
}),

```

Inicialización del sensor de humedad.

```

// INICIALIZACION DEL SENSOR DE MOVIMIENTO PIR EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=7, .category=homekit_accessory_category_sensor, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Motion Sensor"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }
}),
HOMEKIT_SERVICE(MOTION_SENSOR, .primary=true, .characteristics=(homekit_characteristic_t*[]) {
    HOMEKIT_CHARACTERISTIC(NAME, "Motion"),
    &cha_motion,
    NULL
}),
NULL
}),

```

Inicialización del sensor de movimiento.

```

// INICIALIZACION DEL SENSOR DE LUZ EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=10, .category=homekit_accessory_category_sensor, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Light Sensor"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }),
    HOMEKIT_SERVICE(LIGHT_SENSOR, .primary=true, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Light"),
        &cha_light,
        NULL
    }),
    NULL
}),
};

```

Inicialización del sensor de luz.

```

// INICIALIZACION DEL SENSOR DE HUMO Y GAS EN LA APLICACION HOMEKIT
HOMEKIT_ACCESSORY(.id=11, .category=homekit_accessory_category_sensor, .services=(homekit_service_t*[]) {
    HOMEKIT_SERVICE(ACCESSORY_INFORMATION, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Smoke Sensor"),
        HOMEKIT_CHARACTERISTIC(IDENTIFY, my_accessory_identify),
        NULL
    }),
    HOMEKIT_SERVICE(SMOKE_SENSOR, .primary=true, .characteristics=(homekit_characteristic_t*[]) {
        HOMEKIT_CHARACTERISTIC(NAME, "Smoke"),
        &cha_smoke,
        NULL
    }),
    NULL
}),
};

homekit_server_config_t config = {
    .accessories = accessories,
    .password = "111-11-111"
};

```

Inicialización del sensor de humo y gas.

10.1.2. Wifi

```
#ifndef WIFI_INFO_H_
#define WIFI_INFO_H_v

#if defined(ESP8266)
#include <ESP8266WiFi.h>
#elif defined(ESP32)
#include <WiFi.h>
#endif

const char *ssid = "TESIS-QUEVEDO-MENDOZA"; //NOMBRE DE LA RED
const char *password = "123456789"; //CONTRASEÑA DE LA RED

void wifi_connect() {
    WiFi.persistent(false);
    WiFi.mode(WIFI_STA);
    WiFi.setAutoReconnect(true);
    WiFi.begin(ssid, password);
    Serial.println("WiFi connecting...");
    while (!WiFi.isConnected()) {
        delay(100);
        Serial.print(".");
    }
    Serial.print("\n");
    Serial.printf("WiFi connected, IP: %s\n", WiFi.localIP().toString().c_str());
}

#endif /* WIFI_INFO_H_ */
```

Configuración Wifi.