Sistema Centralizado de Medición y Monitoreo de Variables Utilizando IoT Para la Prevención de Accidentes en el Hogar

Juan David Torres Bonilla, ✉ bonilla7401@gmail.com
Jorge Eliecer Largacha Hinestroza, ✉ jorge.larhin@hotmail.com

Trabajo de Grado presentado para optar al título de Ingeniero Electrónico
Asesor: Erika Sarria Navarro, Magister en Ingeniería (M. Sc.)
con Énfasis en Ingeniería Electrónica

Institución Universitaria Antonio José Camacho
Facultad de Ingenierías
Ingeniería Electrónica
Cali - Colombia
2022

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Institución Universitaria Antonio José Camacho para optar al título de Ingeniería Electrónica

Jurado

Jurado

Santiago de Cali,11 de agosto de 2022

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Glosario

Aplicación: Programa preparado para una utilización específica, como el pago de nóminas, el tratamiento de textos, etc. (RAE, 2022)

Bróker: es un intermediario que sirve de enlace entre los clientes y los publicadores dentro de una red MQTT.

Base de datos: Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático (ORACLE, 2022).

Base de datos: Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático (ORACLE, 2022).

Protocolo: Conjunto de reglas que se establecen en el proceso de comunicación entre dos sistemas (RAE, 2022).

Monitoreo: Observar mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías (RAE, 2022).

Escalable: Es una característica de los sistemas que le permiten ajustarse o adaptarse a situaciones que pueden cambiar o variar.

Nube: La Nube en informática se refiere a un servicio de computación que procesa y almacena datos por medio de una red de servidores (Significados, 2022).

PPM: Proporción de la concentración de una sustancia con respecto a la concentración de otra, como una unidad de soluto disuelta en un millón de unidades de disolvente. Se puede

expresar también en términos de peso-peso, volumen-volumen o en cualquier otra relación de unidades de medida (GreenFacts, 2022).

Station: El modo “station” o cliente en un modo de configuración de los dispositivos Wifi en cual se permite conectarse a un punto de acceso y comunicarse con otros dispositivos.

AP: Son dispositivos para establecer una conexión inalámbrica entre equipos y pueden formar una red inalámbrica externa (local o internet) con la que interconectar dispositivos móviles o tarjetas de red inalámbricas (Martinez, 2021).

PWA: Es un software de aplicación web basada en la web tradicional que incorpora algunas particularidades que le hacen parecerse a una app nativa para teléfonos móviles y tabletas (IEBS, 2019).

SSH: es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de Internet a través de un mecanismo de autenticación. (Hostinger, 2022)

Topic: Es un término usado en el protocolo MQTT para referirse al tema o dirección en la que se envían los datos por parte de los suscriptores.

Resumen

El presente proyecto es un sistema centralizado escalable de medición de variables que permite monitorear el estado de la vivienda para prevenir accidentes mediante el uso de una aplicación web. Este sistema de medición cuenta con cuatro sensores los cuales se encargan de medir el estado de puertas y ventanas, presencia o fugas de gas natural y LPG, y temperatura y humo para detectar incendios, estos sensores se conectan a módulos Wifi ESP-01 y NodeMCU los cuales envían datos de las variables a una unidad central Raspberry Pi Zero W, usando el protocolo de comunicación MQTT, donde la Raspberry hace la función de bróker y con un script escrito en Python sube los datos a Firebase, que posteriormente son consultados por la aplicación la cual esta desarrollada en ReactJs.

En la aplicación se pueden visualizar los datos de las variables como gráficas, datos numéricos e imágenes, además de generar alertas que notifiquen al usuario cuando las variables sobrepasen el límite establecido, y al igual que el sistema de medición es escalable al permitir agregar y configurar nuevos sensores mediante una interfaz gráfica amigable con el usuario.

Palabras claves: IoT, Raspberry pi, Wifi, Domótica, Monitoreo, Firebase, Sensores, ESP8266.

Abstract

This project is a scalable centralized system for measuring variables that allows monitoring the state of the home to prevent accidents using a web application. This measurement system has four sensors which are responsible for measuring the status of doors and windows, presence or leaks of natural gas and LPG, and temperature and smoke to detect fires, these sensors are connected to Wi-Fi modules ESP-01 and NodeMCU which send variable data to a Raspberry Pi Zero W central unit, using the MQTT communication protocol, where the Raspberry acts as a broker and with a script written in Python uploads the data to Firebase, which is later consulted by the application which is developed in ReactJs.

In the application you can visualize the data of the variables as graphs, numerical data, and images, in addition to generating alerts that notify the user when the variables exceed the established limit, and like the measurement system, it is scalable by allowing adding and configuring new sensors through a user-friendly graphical interface.

Keywords: IoT, Raspberry pi, Wi-Fi, Home automation, Monitoring, Firebase, Sensors, ESP8266.

1. Planteamiento del problema.

En los hogares se permanece gran parte del tiempo y deberían por lo tanto ser lugares seguros para toda la familia. Sin embargo, hay muchos riesgos de accidentes domésticos, con consecuencias a corto, mediano o largo plazo, ocasionadas por la falta de atención que se les da a algunos factores medibles como lo son calidad del aire, temperatura o fuego por cortocircuito que pueden causar un incendio o intoxicación, por ejemplo, en caso del gas natural.

Según una investigación del Instituto Nacional de Medicina Legal y Ciencias Forenses en Colombia se encontró que para el 2020 se registraron un total de 2.812 muertes accidentales y que la tasa estimada fue aproximadamente de 5,58 casos por cada 100.000 habitantes (Forensis, 2020).

Por otro lado, enfocando el problema en el hogar, según Min Salud el hogar es el lugar donde se presentan la mayor cantidad de muertes accidentales (38,55%) con 1131 casos y luego están los accidentes en escenarios abiertos: en donde se destacan dos categorías una de ellas es los espacios acuáticos al aire libre (17,31 %) con 508 casos y las vías públicas (11,21 %) con 329 casos. (Ministerio de Salud y Protección Social, 2020).

Según FORENSIS en los grupos de edades entre 0 a 19 años tienen un porcentaje de muertes accidentales del 19,21% siendo el grupo de las edades entre 20 a 34 años quienes tienen la mayor probabilidad de muerte accidental con un porcentaje del 29,27% (Forensis, 2020).

Los accidentes domésticos se pueden presentar en el hogar aun cuando no se encuentran las personas dentro de la residencia, lo que pueden generar pérdidas materiales o accidentes más

graves, debido a que se desconoce lo que sucede en los hogares cuando no se encuentra nadie dentro de la vivienda, haciendo uso de las tecnologías IoT se buscan soluciones para mitigar este problema.

¿Cómo diseñar e implementar un sistema de medición de variables utilizando IoT para la prevención de accidentes en el hogar?

2. Justificación

Los accidentes en el hogar son fenómenos que ocurren con mucha frecuencia y que pueden poner en riesgo la salud de las personas que viven dentro de la residencia. Las acciones que se toman a menudo no son rápidas, muchos de estos accidentes suceden por falta de prevención y cuidado de las mismas personas dentro de la residencia. Según el observatorio de salud de Bogotá hubo un incremento de los casos de accidentes en el hogar en los niños menores de 11 años, desde el 2015 al 2019, en donde en el 2019 se presenta un incremento del 50,1% con respecto al 2018, siendo los accidentes por quemaduras el 20,1% y otros el 8,13% (Observatorio de Salud de Bogotá, 2020).

Existen sistemas de medición de variables convencionales análogos que son implementados en empresas y en viviendas de recursos económicos altos, sin embargo, estos sistemas son de un costo elevado y los prototipos de sistemas de medición de variables que utilizan IoT se han orientado en el monitoreo de la cocina o en exteriores. Por lo que en este proyecto se diseñará un sistema de medición de variables que pueda ser implementado en todo el hogar utilizando módulos, sensores, y una interfaz remota que les permitirá a las personas prevenir estos accidentes al informarse del estado de sus hogares. Además, este sistema de monitoreo podría llegar a adaptarse y ajustarse a otros sectores como pueden ser los lugares público y la industria.

Este proyecto va integrado al semillero SELECT junto con otros proyectos de grado, que compartirán una misma unidad central de procesamiento y una interfaz común de usuario.

3. Objetivos

3.1 Objetivo General

Desarrollar un sistema centralizado de medición y monitoreo de variables utilizando IoT para la prevención de accidentes en el hogar.

3.2 Objetivos Específicos

- Investigar sobre los factores principales de riesgo de accidentes que existen dentro del hogar que puedan ser monitoreados

- Implementar un sistema para la medición al menos tres variables seleccionadas según el riesgo de accidentes en el hogar

- Diseñar una interfaz de usuario remoto para la comunicación entre el usuario y el sistema de medición y monitoreo de las variables seleccionadas dentro del hogar.

4. Metodología

4.1 Tipo de Investigación

Esta investigación será de tipo aplicada, ya que se desarrolla un sistema que permite monitorear las variables para prevenir accidentes, con lo que se pretende ayudar a las personas a conocer el estado y condiciones en las que se encuentra su hogar.

4.2 Enfoque de la investigación

Este estudio responde a un enfoque cuantitativo utilizando una investigación documental y experimental, ya que el desarrollo responde a recolección de datos, el análisis de estos, utilización de teorías y conocimientos adquiridos, para luego ser aplicados en el proyecto.

4.3 Fuentes y Técnicas

Se usaron técnicas como la investigación documental que da la base para el desarrollo del proyecto, se consultaron fuentes primarias como libros, y fuentes secundarias que comprenden tesis, artículos, y páginas web, estos documentos nos darán una guía de técnicas, procesos y tecnologías utilizados por otros investigadores, para así hacer la correcta selección de métodos e instrumentos que nos ayudarán a el desarrollo de este, también se combinara junto a técnicas cuantitativas experimentales.

4.4 Fases del proyecto:

4.4.1 Investigación y consulta de antecedentes

En esta fase se hace la investigación sobre las variables ambientales que deben ser supervisadas para evitar accidentes dentro del hogar, y se seleccionan las variables a medir. También se realizará una investigación de las tecnologías, lenguajes de programación, protocolos, y sistemas de bajo costo que permitan el desarrollo del problema en cuestión.

4.4.2 Selección de tecnologías

En esta etapa con base en la fase de investigación se seleccionan las tecnologías a utilizar, que nos permitirán medición y procesamiento de los datos, y así como las aplicaciones, lenguajes entre otros, para realizar el monitoreo y visualización de estos.

4.4.3 Diseño e implementación del sistema de medición

Se realiza el diseño, construcción y programación de los dispositivos de medición y sistemas de procesamiento, también se hace la conexión y la comunicación entre ellos.

4.4.4 Diseño e implementación de interfaz de usuario

En esta fase se hace el diseño de la interfaz gráfica y se hace uso los lenguajes de programación para crear la interfaz de usuario con la que el usuario pueda visualizar la información, estando al tanto de las condiciones de su hogar con respecto a las variables medidas.

4.4.5 Comunicación entre el sistema y la interfaz de usuario

Se implementa la comunicación entre el sistema de medición y procesamiento, y la interfaz de usuario, en esta parte se verifica la conexión, él envió de datos, y la visualización de estos en la interfaz.

4.4.6 Pruebas y ajuste

En esta etapa se hacen las pruebas correspondientes para que todo el sistema funcione y en caso de haber algún problema se hacen los ajustes correspondientes y se corrigen errores.

4.4.7 Documento Final

En esta fase mientras se avanza en el proyecto se ira documentando el desarrollo de este según las fechas estipuladas en el marco de referencia

5. Marco de referencia

5.1 Antecedentes

La tecnología con los años ha dado un gran salto y con ella la manera en la que nos relacionamos entre nosotros y las maquinas, el IoT es una de estas tecnologías que está cambiando la manera de actuar ante distintas circunstancias y estas llegan para mejorar y hacernos más fácil algunas tareas. En este apartado se encuentran las investigaciones consultadas en libros, revistas, páginas y proyectos que fueron desarrollados previo a este documento:

En el trabajo de grado de la Institución Universitaria Antonio José Camacho titulado "Envility: Aplicación del internet de las cosas al monitoreo de la calidad ambiental de los salones de clase de la Uniajc", se presenta un sistema IoT para los salones de la UNIAJC, donde los sistemas de procesamiento utilizados fueron los NodeMCU para las conexiones de los sensores y los cálculos, haciendo uso del protocolo MQTT para el envío de datos a una plataforma IoT llamada ThingSpeak que permite guardar esta información en las bases de datos, esta información es presentada en una aplicación Android desarrollada utilizando App Inventor para monitorear la calidad ambiental. En este proyecto se evidencia como el IoT permite la comunicación entre los dispositivos, utilizando el NodeMCU como bróker, y el protocolo de comunicación MQTT para conectarse entre estos, de esta forma logran el monitoreo de variables como la humedad, y calidad de aire para prever y tomar decisiones frente a estas. (Grueso Carabali & Torres Orozco, 2020)

En el trabajo de grado de la Universidad de San Buenaventura titulado “Dispositivo de monitoreo centralizado y escalable para casas inteligentes”, se enfocan en un sistema centralizado por lo que los datos son enviados a la nube a una plataforma llamada Firebase y se toman decisión a partir de estos, en este proyecto se usan módulos de bajo costo como el NodeMCU para el envío de datos, sensores de temperatura, gas, movimiento, una Raspberry Pi junto con motores y servomotores para funcionar como actuadores, y una aplicación móvil desarrollada en Android Studio para que el usuario pueda monitorear y tomar las decisiones frente a las variables medidas. (Rojas Colunge, 2020)

En el trabajo de grado titulado “Diseño e implementación de un prototipo escalable de detección de gases inflamables, temperatura y alarmas contra incendios basado en tecnología IoT de bajo costo para cocinas en viviendas de Guayaquil” se presenta un prototipo para la medición de distintas variables en las que se usan tecnologías de bajo costo, basándose principalmente en Arduino para la construcción de este, se utilizaron sensores de diferentes fabricantes y una cámara IP. Para el almacenamiento de la información y la interfaz gráfica se usó la plataforma IoT Ubidots que permite mostrar estas variables de una manera amigable, esta plataforma permite visualización de la información en tiempo real y enviar alertas vía correo electrónico. (Bajaña Molina & Molina Sarco, 2020)

En el trabajo de grado titulado “Prototipo de artefacto IoT para la detección de riesgos y prevención de accidentes en la cocina del hogar” se propone un prototipo que está basado en un ordenador de placa reducida como lo es Raspberry pi, integrando una cámara térmica y sensores de calidad de aire y de temperatura, se usó como base de datos MySQL, Python para el

procesamiento de los datos y Swift para el desarrollo de la aplicación móvil, con la que las personas podrán ser notificadas y podrán observar las zonas donde la temperatura es mayor por medio de la cámara térmica. En este proyecto se usaron metodologías como lo es Scrum para un desarrollo ágil de proyecto, se obtuvo un prototipo funcional que afirma que los dispositivos IoT ayudan a prever accidentes en la cocina del hogar. (Mendez León & Vásquez Torres, 2020)

En el proyecto de investigación denominado "Plataforma IoT para el control y monitoreo de variables físicas con tecnología open hardware" se implementa una plataforma IoT para las aulas del instituto ITCA-FEPADE en el Salvador, en la que hacen una red PAN (Personal Área Network) utilizando a ZigBee (802.15.4) como protocolo de comunicación, diseñan las tarjetas de manejo y monitoreo de cargas eléctricas, y la tarjeta de medición de temperatura, utilizando como microcontrolador el Atmega328P, a los cuales conectan los sensores SCT13-30 y SCT13-100 para la medición de la corriente y el sensor DS18B20 para la medición de la temperatura, la información que toman estos sensores es procesada y enviada a un servidor local que almacenaba la información, para después presentarla a los clientes los cuales podían visualizar los datos a través de dispositivos como PC, Smartphone y Tablet, en el proyecto demuestran que los sensores pueden usar protocolos como ZigBee que benefician al generar un bajo consumo de energía, sin embargo este protocolo cuenta con una velocidad máxima de 250 kbps (Padilla & Lobos, 2019)

En el artículo nombrado "Sistema basado en internet de las cosas (IoT) para la monitorización en tiempo real de variables de temperatura y humedad en un equipo de refrigeración del área de farmacia de un hospital de cuarto nivel", se presenta un proyecto

realizado por un grupo de ingenieros de la Universidad del Rosario con el objetivo de monitorear las condiciones térmicas de una nevera de un hospital haciendo uso del sensor SHT11, el cual permite medir humedad y temperatura. Al sensor se le conectó el módulo Wifi ESP8266, el protocolo de comunicación que utilizaron es el MQTT, el procesamiento de los datos lo hicieron con la Raspberry Pi 3, una vez enviados los datos a la nube se publicaban en la plataforma Bluemix, para realizar un envió seguro de los datos implementaron una base de datos relacional implementada en MySQL, alojada en un servidor externo. Para el almacenamiento de los datos el bróker por medio del protocolo HTTP junto con un certificado de seguridad SSL se comunica y transfiere la información a un API el cual se encarga de almacenar los datos. La seguridad al momento de enviar y almacenar los datos es muy importante es por eso por lo que se debe hacer uso de protocolos como HTTP y SSL, además de usar recursos como los Api que permiten un mejor servicio con los clientes (Soto, y otros, 2019)

En el trabajo de grado de la Universidad Distrital Francisco José de Caldas titulado "Sistema de medición de calidad de aire e intensidad UV con sistema embebidos bolt iot y Arduino" se utilizó el sensor de medición de gases MQ-135, el sensor de intensidad UV ML8511 y el sensor temperatura y humedad DHT22, usaron como sistema de procesamiento el Arduino Mega y la placa Bolt IoT las cuales las conectaron por el puerto serial, el Arduino lee los datos enviados por los sensores y luego los envía a la placa Bolt IoT la cual se encarga de interpretar los datos y presentarlos de manera gráfica, Aunque el sistema Bolt IoT en su interfaz gráfica y programación es demasiado amigable con el usuario, cuenta con políticas de acceso y ejecución

de las Apis, lo que no les permitía hacer mediciones en todo momento ya que había un límite de solicitudes de tipo Api (Tinjacá, 2019)

En el artículo "Sistema de monitoreo de monóxido de carbono en tiempo real en el hogar como aplicación de Internet de las Cosas" se describe un proyecto donde se utilizan como sistema de procesamiento el Arduino UNO (Atmega328P), para medir el monóxido de carbono del hogar utilizan el sensor MQ-7 y hacen uso de la infraestructura AWS EC2, implementan un contenedor Docker como plataforma y dentro de ese contenedor tienen los servicios de Message Broker, utilizan como base de datos un servidor externo en la nube con MySQL, aunque el sistema es de bajo costo, requiere de un computador que se conecte con los Arduino, los cuales están conectados por medio de cable USB, lo que es poco eficiente y estético a la hora de comunicarlos por lo que sugieren hacer eso de conexión inalámbrica como módulos Wifi (Zárate & Romá, 2019).

En el trabajo de grado de maestría, "Sistema de Control y Monitoreo de Consumo Energético para Equipos de Climatización Orientado a Internet de las Cosas (IoT)" se tiene como objetivo reducir el consumo que generan los sistemas de refrigeración como los aires acondicionados, en este proyecto hace uso del NodeMCU V3.0 el cual se puede programar a través del entorno de programación Arduino IDE, para la medición de temperatura y humedad utilizaron el sensor DHT21, utilizan el sensor de corriente no invasivo ECS1030. Para este proyecto hicieron una aplicación web utilizando una plataforma en el software PHP alojado en la nube con dominio propio, el protocolo de comunicación que utilizaron fue el MQTT ya que se adaptaba mejor a las necesidades de su proyecto y para visualizar los datos de los sensores usaron

la plataforma IoT Ubidots, la cual les permitía visualizar los datos obtenidos a través de gráficas (Ariza, 2019)

Para la realización del proyecto de las lecturas consultadas se tomaron en cuenta fundamentalmente los proyectos presentados en (Grueso Carabali & Torres Orozco, 2020), (Rojas Colunge, 2020), (Mendez León & Vásquez Torres, 2020), (Soto, y otros, 2019), (Tinjacá, 2019) y (Ariza, 2019), como referencia al hacer uso del protocolo MQTT para la conexión de los dispositivos, además de informar sobre las cualidades y posibles usos de dispositivos de procesamiento como el NodeMCU, Raspberry Pi, Arduino, ESP8266 y algunos sensores como el DS18B20, DHT21, MQ-7, entre otros, los cuales son útiles para la medición de ciertas variables y plataformas IoT como Firebase y Ubidots (ver Tabla 1).

Tabla 1. Tabla comparativa del proyecto vs antecedentes

	Torres & Largacha, 2022	Grueso Carabali & Torres Orozco, 2020	Rojas Colunge, 2020	Mendez León & Vásquez Torres, 2020	Soto, y otros, 2019	Tinjacá, 2019	Ariza, 2019
Dispositivos	ESP-01, NodeMCU, MQ-9, DS18B20, Interruptor magnético, Raspberry Pi Zero W	DHT22, MQ-135, PMS5003, NodeMCU v2	DHT22, NodeMCU V2, MQ-2, Sensor PIR, Raspberry Pi 3 B, L293D	Raspberry Pi Cámara, Raspberry Pi 4, bme280, ccs811	SHT11, ESP8266, Raspberry Pi 3	Bolt Iot, Arduino Mega 2560, MQ-135, DHT22, ML8511	NodeMCU V3, DHT21, ECS1030, KY-005, KY-022
Protocolos	Wifi 802.11, MQTT	Wifi 802.11, MQTT	Wifi 802.11	I2C, Z-Wave	MQTT, HTTP, Wifi 802.11	Wifi 802.11, HTTPS	Wifi 802.11, MQTT
Base de datos	FireBase realtime	ThingSpeak	FireBase	MySQL	MySQL	Cloud Bolt IoT	IoT Ubidots

	Torres & Largacha, 2022	Grueso Carabali & Torres Orozco, 2020	Rojas Colunge, 2020	Mendez León & Vásquez Torres, 2020	Soto, y otros, 2019	Tinjacá, 2019	Ariza, 2019
Plataforma	FireBase	ThingSpeak	FireBase	No usa	BlueMix	Cloud Bolt IoT	IoT Ubidots
Aplicación	IoT Home (app desarrollada en ReactJs)	App desarrollada en App Inventor	App desarrollada en Android Studio	App desarrollada en Swift para Apple	Interfaz web que hace uso de BlueMix	App desarrollada en JavaScript	App web desarrollada en PHP

5.2 Marco Teórico

En el marco teórico se presentarán definiciones de conceptos, términos, y teorías que son necesarios para el desarrollo e implementación del proyecto realizado en este documento, con el fin de dar claridad y un mejor entendimiento sobre el contenido de este.

5.2.1 Accidentes en el hogar

Son hechos aleatorios que como consecuencia generan lesiones, pueden ocurrir en cualquier lugar de la residencia, como lo son las escaleras, las habitaciones, la cocina, el baño, la sala, entre otros. Las causas de estos accidentes son muy variables, pueden ser factores como la edad, el desconocimiento e incluso las malas condiciones del entorno. Es importante la caracterización de estos accidentes, ya que ayudara a la selección de variables a ser monitoreadas:

5.2.1.1 *Intoxicaciones*

Una intoxicación es la reacción del organismo a la entrada de una sustancia tóxica que causa lesión o enfermedad y en ocasiones la muerte. El grado de toxicidad varía según la edad, el

sexo, el estado nutricional, la vía de entrada y la concentración del tóxico. Las intoxicaciones que se presentan dentro del hogar pueden ser generadas por distintas circunstancias como ingerir medicamentos o alimentos en estado de descomposición, o por gases tóxicos como lo son aerosoles, fugas de gas, combustibles o derivados de este, e incluso productos de limpieza. (Escuela Cántabra de salud, 2019).

Los principales gases que pueden causar intoxicaciones son:

- **Monóxido de carbono (CO):** es un gas peligroso incoloro e inoloro que produce cualquier combustible, ya sea gas natural, carbón, madera o gasolina, al ser inhalado en niveles elevados puede causar incluso la muerte. (SoCalGas Company, 2022)
- **Gas natural y LPG:** el gas natural y el LPG son gases compuestos por varias moléculas como el gas metano (CH4), butano (C4H10) y propano (C3H8), según (Hospital Alemán Asociación Civil, 2018): “La principal consecuencia de la intoxicación con gas es la falta de oxígeno, que en el organismo impacta en todos los órganos”. Los principales síntomas que podrían experimentar las personas que se encuentran en los alrededores y han inhalado gas natural o LPG son náuseas, cefaleas, mareos, vómitos, dependiendo de la concentración del gas y el tiempo de exposición. Y cuando la concentración es mayor, pueden manifestarse pérdida de los reflejos y la conciencia o convulsiones. (Infobae, 2018)
- **Óxidos de nitrógeno:** es un gas que al igual que el CO es inoloro e incoloro, este no es inflamable, todos estamos expuestos a pequeñas cantidades de este, pero suelen ser más comunes en las cocinas, al quemar madera e incluso si se fuma. También son liberados al

aire por los escapes de los vehículos motorizados, y por la combustión del carbón, gas natural, y petróleo. La exposición a niveles altos de este gas puede dañar las vías respiratorias. (ATSDR, 2016)

- **Dióxido de azufre (SO2):** El dióxido de azufre es un gas incoloro que a altas concentraciones puede ser detectado por su sabor y por su olor cáustico e irritante, este gas se puede encontrar en las zonas urbanas y al ser disuelto en agua se puede convertir en acido sulfuroso el cual se oxida rápidamente con el aire y forma ácido sulfúrico. La OMS de calidad de aire para Europa recomiendan no superar concentraciones medias diarias de 125mg/m^3 de SO2, con máximos de 10 minutos de 500mg/m^3 y valores medios anuales de 50 mg/m^3. (Aránguez Ruiz, s.f)

5.2.1.2 *Quemaduras*

Estas son uno de los accidentes que generan las peores lesiones y una de las más dolorosas, estas se pueden dar por distintas circunstancias y en diferentes grados, usualmente el lugar en hogar donde se generan quemaduras es en la cocina, pero estas pueden ser provocadas por una plancha, el agua en los baños que poseen un calentador, un cortocircuito, el rose contra una superficie rugosa y otros. Estas lesiones se pueden presentar en todo tipo de edades incluso en los bebes con quemaduras leves que pueden ser provocadas por el rose del pañal, en edades más avanzadas las quemaduras son ocasionadas por el fuego o por superficies calientes (Revistas, 2021)

Entre los tipos de quemaduras según el grado se tienen los siguientes efectos:

- **Quemaduras de primer grado:** Estas son las más superficiales y solo afectan la superficie de la piel, pueden ser ocasionadas por roses, exposición al sol, y usualmente pueden ser tratadas en casa.
- **Quemaduras de segundo grado:** estas son más graves que las de primer grado afectan la siguiente capa de la piel (Dermis), por lo que son más profundas y con más complicaciones, pueden dejar cicatrices y necesitan más tiempo para sanar, para ser tratadas se debe ir con un especialista.
- **Quemaduras de tercer grado:** afectan las tres capas de la piel y estas pueden incluso afectar el musculo, estas quemaduras son las más graves de todas y estas pueden ser generadas por agentes químicos, electricidad, o una superficie a una temperatura muy elevada. En este caso se debe actuar de manera urgente, y estas deben ser atendidas en un hospital donde se tienen los conocimientos para tratar este accidente.

5.2.1.3 *Caídas o Golpes*

Los golpes y las caídas son de los accidentes que ocurren con mayor frecuencia en el hogar, y son un problema de importancia mundial, muchos de estos suceden por el descuido de las personas que residen en él, o por situaciones que no están bajo su control. La población que es más afectada por este tipo de accidentes es usualmente las personas mayores a sesenta años y se debe a su condición físicas ya que en esta etapa se es propenso a perder el equilibrio o el sentido de orientación, debilidad muscular, u otros que vienen al pasar los años. Las principales causas de caídas y golpes son ocasionadas por pisos mojados, Incorrecta ubicación de los objetos, poca

iluminación en el ambiente, deterioro de las instalaciones, trastornos médicos y efectos colaterales de los medicamentos.

Según la OMS más del 80% de las muertes suceden en países de bajos y medios recursos, y el 60% de esas muertes se ubican en el pacífico Occidental y Asia Sudoriental (Organizacion Mundial de la Salud, 2021).

5.2.1.4 *Ahogo o Asfixia*

Estos se presentan cuando la persona posee problemas para respirar puesto que algo obstruye la garganta y no permite que llegue el oxígeno a los pulmones, es muy común, se presenta en todas las edades, y es una de las causas de muerte de muchas personas al anualmente, los síntomas pueden ser dificultad para respirar, incapacidad para hablar, tos débil, piel morada y perdida del conocimiento. Para tratar problemas se deben administrar los primeros auxilios de manera inmediata, o consultar con un profesional o servicio médico. (Biblioteca Nacional de Medicina, s.f.)

Unas de las causas que ocasionan el ahogo o asfixia son el comer demasiado rápido, inhalar o ingerir objetos pequeños, lesiones en la cara o la cabeza, problemas de amígdalas u otros.

5.2.2 Sensores

Un sensor es todo dispositivo o artefacto sensible a alguna magnitud física y es capaz de convertir esa magnitud física en una señal eléctrica. Los sensores también conocidos como

transductores son componentes fundamentales en los sistemas modernos de adquisición de datos (Dewesoft, 2020). Son fabricados por diferentes fabricantes y en diferentes modelos, para permitir la medición de diferentes variables físicas constituyendo entonces distintos tipos de sensores como: sensores de gas, sensores de temperatura, sensores de calidad del aire, sensores llama, etc.

5.2.3 Tarjetas de desarrollo

Una tarjeta de desarrollo es una placa o circuito que contiene un microcontrolador principal que ejecuta una serie de instrucciones de un programa suministrado. Alrededor de este procesador o unidad principal se ha creado un diseño electrónico que permite, la programación del componente suministra el voltaje adecuado para el correcto funcionamiento del controlador, y proporciona acceso a las entradas y salidas del microcontrolador para la conexión de sensores y actuadores. Actualmente hay una gran variedad de tarjetas de desarrollo, cada una ofrece características especiales (Concepción, 2021). Algunos ejemplos son:

5.2.3.1 *Arduino*

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (Arduino, 2021). Existen varios modelos de Arduino los cuales tienen diferentes características, estos modelos son:

- Arduino UNO
- Arduino NANO
- Arduino Mega
- Arduino Yun
- Arduino Leonardo

5.2.3.2 *ESP8266 NodeMCU*

Es un sistema embebido de desarrollo de hardware y software libre que está basada en el chip ESP8266, esta placa de desarrollo es ideal para trabajar IoT de una manera económica ya que posee internamente un módulo Wifi (Ariza, 2019).

5.2.3.3 *Raspberry Pi*

La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un ratón y teclado estándar. Es un pequeño computador que corre un sistema operativo Linux capaz de permitirles a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python (Raspberry, 2021). Al igual que las anteriores placas es muy usada para proyectos de electrónica

5.2.4 IoT (Internet de las cosas)

El internet de las cosas o también conocido como IoT por sus siglas en inglés (Internet of Things), es un concepto que se refiere a la interconexión de objetos o dispositivos cotidianos con internet, conectándolos a una red en la que todos pueden interactuar entre sí (Gracia, 2021).

Respecto a que objetos se pueden conectar son básicamente cualquiera, puede ir desde sensores y dispositivos mecánicos hasta dispositivos de uso cotidiano como la nevera, el aire acondicionado, o incluso elementos como las prendas de vestir.

5.2.5 Protocolos usados en IoT

Se define como protocolo al conjunto de reglas o normas comunes que se deben cumplir para intercambiar mensajes entre dispositivos, así como a nivel humano hay reglas para la comunicación algunas más estrictas que otras, para que los dispositivos se puedan comunicar también hay unas reglas que deben seguir para conseguir una comunicación exitosa.

Entre el gran número protocolos que existen para las comunicaciones entre los diferentes dispositivos de hoy en día, encontramos protocolos para IoT, estos protocolos se caracterizan usualmente por el bajo consumo para transmisión de información (Cisco, 2016). Unos de los protocolos de comunicación más usados para IoT son:

5.2.5.1 *I2C*

El protocolo fue diseñado por Philips Semiconductors específicamente para aplicaciones de audio y video, Es un estándar que se basa en el uso de dos líneas que transportan datos de

forma bidireccional entre dos o más dispositivos o circuitos integrados, que facilita la comunicación entre microcontroladores, memorias y otros dispositivos la comunicación entre dispositivos presenta la jerarquía de maestro esclavo. Este protocolo posee un mecanismo para la detección de colisiones por lo que el maestro solo envía datos si detecta que las vías de comunicación están libres, se pueden dar velocidad de unos 100 Kbps, he incluso alcanzar velocidades de 3,4 Mbps bidireccionalmente. (Salas Arriarán, 2017).

5.2.5.2 *ZigBee*

Esta tecnología está diseñada para una amplia gama de aplicaciones y topologías de red, el estándar ZigBee soportan redes en mallas, en las que no hay un nodo principal si no que todos los nodos están conectados entre sí permitiendo así variados caminos para la información, las ventajas es que se pueden implementar aplicaciones con baja memoria y con muy poca potencia de procesamiento y es escalable. ZigBee se basa en el protocolo IEEE 802.15.4. (Quiñonez Muñoz, 2019)

5.2.5.3 *Wifi*

El wifi es un mecanismo que permite conectarse de forma inalámbrica a través de internet desde distintos dispositivos, como lo pueden ser computadores, dispositivos móviles, impresoras y otros. Nos permite la entrada a la red sin la necesidad del uso de cables.

Estas conexiones inalámbricas son posible gracias a la radiofrecuencia, es por medio de estas que se transmiten y se reciben la información. El estándar IEEE 802.11 es el que define los

protocolos para las comunicaciones entre los dispositivos, con el tiempo se han ido implementando nuevas versiones de este estándar por lo que recibe actualizaciones constantemente. (SoftwareLab.org, 2014-2021).

5.2.5.4 *MQTT*

El transporte de telemetría de cola de mensajes (MQTT, Queuing Telemetry Transport) está diseñado para proporcionar conectividad entre aplicaciones embebidas middleware en un lado y redes y comunicación en el otro. Está basado en publicaciones y suscripciones, donde el sistema está dividido en dos partes, clientes y agente intermediario o Bróker, los clientes pueden ser tanto publicadores o suscriptores o incluso ambos, los publicadores son los clientes que envían los datos al bróker sobre un tema o topic, los suscriptores son aquellos clientes que quieren acceder a la información enviada por los publicadores y se conectan con el bróker para que este les suministre la información publicada en el topic que deseado. (HiveMQ GmbH, 2022)

Las ventajas de este protocolo son su sencillez, escalabilidad y su ligereza, requiere de un ancho de banda mínimo y bajo consumo energético por lo que los dispositivos no necesitan mucha potencia.

5.2.5.5 *HTTP*

El protocolo HTTP (HyperText Transfer Protocol) es sobre todo un protocolo de transferencia de archivos, este permite realizar peticiones sobre los datos u otros recursos que se requieran. Este protocolo es la base de los intercambios de información en la web y usa una

estructura cliente servidor, lo que significa que el cliente hace una petición que puede ser presentada en el navegador web o cualquier otro programa, esta petición llega al servidor y devuelve un valor correspondiente a la petición.

A lo largo del tiempo HTTP ha ido creciendo y sus distintas versiones agregan a estas nuevas funcionalidades, su estructura cliente-servidor le ha permitido a este protocolo evolucionar permitiéndole a este hacer otros procesos como el cache o métodos de identificación o autentificación. (MDN Web Docs, 2005-2021).

5.2.5.6 *Z-Wave*

Es una tecnología de comunicaciones de RF de baja potencia diseñada principalmente para la automatización del hogar y pequeños centros comerciales, para productos como lo son electrodomésticos, puerta garajes, sensores, como ZigBee es un protocolo que soporta redes en malla, es escalable y permite conectar hasta 232 dispositivos, este protocolo ofrece bajan latencia con velocidades de datos de hasta 100 Kbit/s, pero este trabaja en la banda sub-1GHz, por lo que no es afectado por la interferencia de 2,4 GHz. (Quiñonez Muñoz, 2019)

5.2.6 Plataformas IoT

Las plataformas y herramientas de IoT se consideran el componente más importante del ecosistema de IoT. Los dispositivos de IoT permiten conectarse con a otros dispositivos y aplicaciones de IoT para transmitir información utilizando protocolos estándar de comunicación. Las plataformas de IoT llenan el vacío entre los sensores del dispositivo y las redes de datos.

Conecta los datos al sistema de sensores y brinda información utilizando aplicaciones de back-end, las cuales les permiten a los humanos la visualización de los datos enviados por los dispositivos o la interacción con los mismos (Geekflare, 2020). Entre las distintas plataformas tenemos: Zetta, Firebase, Node-red, Flutter, M2MLabs, ThingsBoard, Thinger, SiteWhere.

5.2.7 Arquitecturas de software para IoT

La arquitectura de software investiga métodos para determinar cómo dividir mejor un sistema, cómo los componentes se identifican y se comunican entre sí, cómo se comunica la información, cómo los elementos de un sistema pueden evolucionar de forma independiente y cómo se puede describir todo lo anterior utilizando notaciones formales e informales. La arquitectura de software de un sistema informático es una descripción del sistema que ayuda a comprender cómo se comportará el sistema (Barrera, 2018).

La arquitectura IoT tiene que cumplir ciertos requerimientos para que su aplicación sea viable. La arquitectura IoT debe permitir que los dispositivos y objetos puedan interactuar entre ellos, debe ser escalable, flexible, robusta, eficiente y segura (Enrique, 2021).

Es por estos requerimientos del IoT que las arquitecturas normalmente usadas en internet no son viables o eficientes. A continuación, veremos algunas arquitecturas IoT propuestas por algunos estudiantes en donde toman en cuenta todos estos aspectos.

La arquitectura propuesta en (Benítez, Anías, & Plasencia, 2016) se puede apreciar en la Figura 1, la cual se divide en las siguientes capas:

- Capa de dispositivos: están los dispositivos, que son aquellos que se encuentran en contacto con el mundo físico y permiten, por tanto, obtener datos y ejecutar acciones sobre el mismo.
- Capa de Gateways: está compuesta por los Gateways, cuya función principal es posibilitar la conexión a la Capa de Red.
- Capa de Red: encargada de transportar el tráfico de dispositivos y Gateways hacia o desde la nube o centro de datos.
- Capa de Nube / Centro de Datos: en la cual se procesan los datos que llegan desde los dispositivos, al igual que los comandos que se envían hacia estos, los cuales pueden generarse en la propia Capa de Nube / Centro de Datos o en las demás capas de la arquitectura.
- Capa de Aplicaciones: es a través de la cual los usuarios interactúan con el ecosistema IoT y sacan provecho de los procesamientos que se realizan, en su mayoría, en la capa inferior.
- Capas de Gestión y de Seguridad: si bien esta última es parte de la gestión en toda red, en el caso de IoT se hace necesario separarla para destacar su importancia. Estas dos capas garantizan un correcto funcionamiento de cualquier solución IoT, así como la protección de sus diferentes recursos, para lo cual, están presentes en las demás capas de la arquitectura, por lo que en cualquiera de estas pueden encontrarse funciones de las Capas de Gestión y Seguridad.

Figura 1. Arquitectura propuesta IoT

Nota: Fuente *(Benítez, Anías, & Plasencia, 2016)*.

La siguiente arquitectura IoT es propuesta por (Golondrino, Alarcón, & Ríos, 2020), el diseño de esta arquitectura tiene como objetivo el desarrollo de sistemas de monitorización y análisis de variables fisiológicas en el área de asistencia médica, el cual se podría adaptar a otros proyectos que sean de la misma temática. Como se puede ver en la Figura 2, al igual que en la arquitectura anterior se divide en capas las cuales son:

- **Capa de captura:** se obtienen un conjunto de variables fisiológicas desde los dispositivos ponibles comerciales, las cuales una vez obtenidas son enviadas a través del protocolo de comunicación inalámbrica bluetooth LE a los equipos receptores.
- **Capa de almacenamiento:** una vez los datos son obtenidos por los terminales o equipos receptores, son decodificados a partir de la trama de datos capturada en formato hexadecimal. Cada conjunto de datos capturados por paciente desde los

dispositivos ponibles es asociado a una sesión de captura, por lo cual los datos decodificados son almacenados teniendo un id de la sesión en una base de datos.

- **Capa de análisis:** se realizan cálculos estadísticos básicos a partir del historial de datos capturados y se aplican modelos de análisis de datos, para lo cual se cargan los datos al modelo, se ejecuta el modelo y se evalúa la precisión de este haciendo uso de librería de minería de datos.
- **Capa de visualización:** se realiza un seguimiento gráfico de las variables fisiológicas obtenidas en la capa de captura, del mismo modo en esta vista se presentan los resultados de las estadísticas halladas, así como los resultados de la ejecución y evaluación de los modelos de análisis considerados.

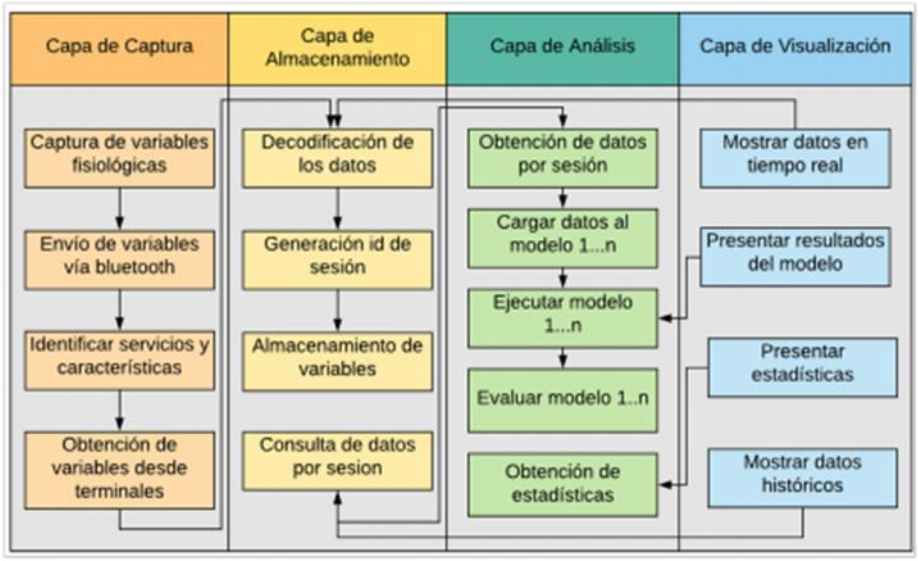


Figura 2. Prototipo arquitectura IoT

Nota: Fuente *(Golondrino, Alarcón, & Ríos, 2020)*.

5.3 Marco Conceptual

En este marco se definen algunos conceptos y términos que son necesarios tenerlos claros para el entendimiento y el desarrollo del proyecto.

5.3.1 Entornos de desarrollo

Los entornos de desarrollo o IDE (Integrated Development Environment) son programas informáticos las cuales cuentan con un conjunto de herramientas que permiten al programador el desarrollo de un programa o una aplicación. Los entornos de desarrollo pueden dedicarse a un solo lenguaje de programación o bien pueden utilizar varios.

Un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen autocompletado inteligente de código (IntelliSense) y algunos contienen un compilador, un intérprete, o ambos. Entre los IDE usados para programar las tarjetas de desarrollo están el Arduino IDE, STM32Cube IDE, CooCox CoIDE, KEIL – microVision y Mbed.

5.3.2 Placa electrónica

Es una superficie en la cual están impresas unos caminos o pistas hechas de un conductor eléctrico laminado (usualmente cobre) sobre una baquelita la cual es no conductora, las placas electrónicas se pueden encontrar en todos los dispositivos electrónicos.

5.3.3 Monitoreo

El monitoreo hace referencia a la acción de observar, supervisar en el ámbito de la seguridad monitorear se realiza por medio de un monitor, pero en el ámbito ambiental este se refiere a la observación del medio ambiente para recolectar datos relacionados con la contaminación, y distintos parámetros como temperatura, calidad de aire, etc. (Pérez Porto & Gardey, 2013)

5.3.4 Sistema centralizado

Son aquellos sistemas que dependen de una unidad de control central, donde cada uno de los dispositivos, sensores y actuadores están conectados a la misma unidad central, que es encargada de administrar y procesar estos datos recibidos por los sensores, y luego a los actuadores o dispositivos para la tarea correspondiente. (Núñez Sebastián, 2012).

5.3.5 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal, que mediante una serie de instrucciones que le permiten a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina, para decirlo de una manera sencilla un lenguaje de programación son una serie de símbolos, palabras claves y reglas semánticas que permiten la comunicación entre un programador y una máquina (Rock Content, 2021). Entre algunos lenguajes de programación tenemos: Lenguaje C, C++, Python, JavaScript, PHP, LUA, etc.

5.3.6 Interfaz de usuario

La interfaz de usuario es el medio por el que se comunican el usuario y la maquina o dispositivo, esta suele ser un programa acompañado por una interfaz gráfica, la cual le permite al usuario visualizar o comprender mejor el contenido que le presenta la máquina.

5.3.7 Servidor

Es un ordenador de gran potencia el cual se encarga de prestar servicios, permitiendo transferir información solicitada por el cliente, su función es almacenar los datos de las páginas web y transmitirlos al usuario por medio del navegador, haciendo uso del protocolo HTTP. (de Sousa, 2019).

5.3.8 Protocolos de comunicación

Un protocolo de comunicación es un sistema de reglas o pasos a seguir que permiten que los dispositivos pertenecientes a la red se comuniquen entre ellos para transmitir o recibir información.

5.3.9 API

API significa interfaz de programación de aplicaciones. Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber

cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos (Red Hat, 2021).

5.3.10 MySQL

Es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. Es una de las muchas opciones de software RDBMS. Suele pensarse que RDBMS y MySQL son lo mismo debido a la popularidad de MySQL. Para nombrar algunas aplicaciones web grandes como Facebook, Twitter, YouTube, Google y Yahoo!, todas usan MySQL para el almacenamiento de datos. Aunque inicialmente se creó para un uso limitado, ahora es compatible con muchas plataformas de computación importantes como Linux, macOS, Microsoft Windows y Ubuntu (Gustavo, 2020).

5.3.11 Almacenamiento en la nube

Es un servicio que permite almacenar información a través de internet a un sistema de almacenamiento externo que es propiedad de un tercero. En estos se pueden almacenar fotos, videos, documentos y otros archivos, las empresas usan la nube para las copias de seguridad de

sus datos, para poder tener un respaldo y poder compartirlos entre las diferentes ubicaciones. (Microsoft, s.f.).

5.4 Marco Contextual

El contexto en el que se va a realizar el proyecto es el lugar de residencia de las personas, al cual muchas le llaman hogar. La RAE define el hogar como: “Casa o domicilio, donde un grupo de personas emparentadas viven juntas” (RAE, 2021) y define residencia como: “Casa en la que se reside y en la que se vive” (RAE, 2021).

El hogar es el espacio que las personas más frecuentan e invierten su tiempo, por lo que es el lugar donde más pueden sufrir accidentes. Un reporte de Forensis revela que las muertes accidentales en viviendas representan el 38.55% de los casos en Colombia para el año 2018. Estos accidentes pueden variar dependiendo del estrato socioeconómico de las personas, siendo los estratos más bajos los más propensos a sufrir ciertos accidentes.

Un proyecto de investigación realizado por estudiantes de economía de la Universidad de La Salle, encontraron que la correlación entre el estrato socioeconómico y el nivel de educación de las personas es negativa (Aldana & Citelly, 2017), lo que se puede concluir en que las personas con menos recursos son quienes tienen un nivel de educación menor.

Según datos entregados por Forensis el mayor número fallecidos lo registra individuos con educación básica secundaria o secundaria básica con un porcentaje del 31.21%, seguidos por las personas que presentan educación básica primaria con una proporción del 27.60% (Forensis, 2020).

Este proyecto está integrado junto con otros proyectos pertenecientes al semillero de investigación Select, los cuales se desarrollarán para los hogares y viviendas colombianas, que no cuentan con los recursos suficientes para tener un sistema de medición y monitoreo de variables que pueden ocasionar accidentes dentro de sus hogares.

5.5 Marco Legal

A continuación, se encuentran las leyes y normas que hay que considerar al momento de implementar un sistema de medición de variables usando IoT y de esta forma se pueda evitar cualquier repercusión legal.

5.5.1 Ley 1273 del 2009

Denominado “de la protección de la información y de los datos” o ley de delitos informáticos hace referencia a la violación de los sistemas de información, las comunicaciones y otros, donde se penaliza a cualquier persona que sin autorización previa intercepte, modifique, borre, envié datos, o ingrese a un sistema sin poseer una orden judicial, o un permiso por la persona que poseen los derechos sobre este. También es meritorio de castigo la suplantación de identidad para recolectar información, hurto por medios electrónicos, y el uso de software malicioso en los sistemas de información (Republica de Colombia, 2009).

5.5.2 Ley 1581 de 2012

En esta ley se presentan los derechos que tienen las personas a conocer, modificar, y rectificar sus datos personales que se encuentren en una base de datos, esta ley aplica al tratamiento de los datos recogidos por empresas, o entidades públicas y privadas, aunque no aplica si los archivos están almacenados en un ámbito doméstico, pero una vez estos datos pasan a bases de datos de terceros o al hacer tratamiento de estos, el titular deberá ser notificado y solicitar su autorización (Republica de Colombia, 2012).

5.5.3 ISO/IEC 29161:2016

"Establece un esquema de identificación único para Internet de las cosas (IoT), basado en estructuras de datos existentes y en evolución. Esta Norma Internacional especifica las reglas comunes aplicables para la identificación única que se requieren para garantizar la compatibilidad total entre diferentes identidades. La identificación única es una construcción universal para cualquier objeto físico, objeto virtual o persona. Se utiliza en sistemas de información de IoT que necesitan rastrear o hacer referencia a entidades. Está diseñado para su uso con cualquier medio de IoT" (ISO/IEC JTC 1/SC 31 , 2016)

5.5.4 ISO/IEC 30141:2018

En este documento se presentan arquitecturas y modelos para unas buenas prácticas con IoT. Ya que es una tecnología que ha estado evolucionando con los años se pretende estos sistemas puedan ser fiables y poder adaptarse a las diferentes necesidades, permitiendo hacerles frente a los retos que aparecen con el avance de esta tecnología por ejemplo si una empresa

quisiera implementar un sistema de IoT, entonces se basarían en unas referencias que fueron implementadas con buenas prácticas garantizando el buen funcionamiento, seguridad y funcionalidad de este (ISO/IEC JTC 1/SC 41 , 2018).

5.5.5 Resolución 000964 de 2019: capítulo 7/sección 1 artículo 29

En este articulo presenta los tramites en línea relacionados con el servicio de radioaficionado, donde nos muestra que todos los trámites para el uso del espectro como radioaficionados pueden ser realizados en línea por medio del RABCA en la página web del MinTic, diligenciando el formulario y cumpliendo con los archivos y requisitos que se solicitan. (Ministerio de tecnologías de la información y las comunicaciones, 2019)

5.5.6 Resolución 000964 de 2019: capítulo 7/sección 4 artículo 32

Este articulo nos habla sobre la provisión de servicios de telecomunicaciones haciendo uso de bandas de frecuencia libre, donde las bandas de frecuencia de libre utilización ya están definidas por la agencia nacional del espectro, y para disponer de esta se deben cumplir con una serie de requisitos y obligaciones, propias de un proveedor de servicio de telecomunicaciones. (Ministerio de tecnologías de la información y las comunicaciones, 2019).

6. Selección de Tecnologías para el Proyecto

A continuación, se presentarán las tecnologías que se eligieron para el desarrollo del proyecto con su respectiva explicación.

6.1 Selección de las variables a monitorear

En el hogar existen situaciones que pueden poner en riesgo la integridad de las personas que viven en ellos, es por esta razón, que monitorear ciertas variables permite determinar situaciones que pueden poner en peligro la seguridad de todos:

Los incendios son accidentes comunes que ocasionan quemaduras y/o perdida de los bienes materiales por eso se escogió medir las variables temperatura y humo para determinar si se ha iniciado un incendio. El monitoreo de la lectura de los gases denominados como gas natural o gas propano permiten detectar fugas, que pueden llegar a intoxicar a las personas que son expuestas por accidente, o incluso provocar explosiones, y las variables que indican las aperturas de puertas y ventanas, las cuales permiten comprobar la seguridad e integridad de la residencia y así prevenir la entrada de intrusos para evitar robos o situaciones más peligrosas.

6.2 Selección de los sensores

Los sensores son los encargados de transducir las señales físicas en señales eléctricas para poder ser procesadas y leídas por las personas, estos tienen un papel fundamental en el monitoreo

de las variables, a continuación, se hará una comparación entre los sensores que se tomaron en cuenta para la realización del proyecto.

6.2.1 Sensores de temperatura y humedad

Los sensores de temperatura permiten medir la temperatura del lugar que se desea monitorear para así conocer el cambio de temperatura generado en ocasiones por los incendios, para la selección del sensor de temperatura se tomaron en cuenta los sensores presentados en la Tabla 2.

Tabla 2. Sensores de temperatura

Nombre	Imagen	Rango	Precisión	Alimentación	Output	Precio
SHT31	 **Nota:** Fuente https://acortar.link/koeAUF	-40º a 125ºC	±0.2ºC	+3.3V/+5VDC	Digital	US $7,48
DHT21	 **Nota:** Fuente https://acortar.link/b2uwHx	-40º a 80ºC	±0.5ºC	+3.3V/+5VDC	Digital	US $6,73

Nombre	Imagen	Rango	Precisión	Alimentación	Output	Precio
DS18B20	 **Nota:** Fuente https://acortar.link/X6piIU	-55º a 125ºC	±0.5ºC	+3.3V/+5VDC	Digital	US $2,50

Teniendo en cuenta que en la Tabla 2 se compararon los sensores, se decidió utilizar la sonda DS18B20, ya que permite medir temperaturas desde los -55º hasta los 125ºC con una precisión de ±0.5ºC. En Cali, Colombia las temperaturas mínimas no descienden a menos de 14ºC por lo que no es de interés medir temperaturas menores a esta, se descartó el DHT21 el cual mide temperaturas de -40ºC a 80ºC ya que tiene un rango de medición menor que el del DS18B20. En cuanto al SHT31, aunque cuenta con un rango máximo de temperatura igual al del DS18B20 con una mayor precisión, es más costoso y es más difícil de obtener que el sensor DS18B20 por lo que se optó no hacer uso de este sensor.

6.2.2 Sensor de humo y Sensor de Gas natural

El sensor de humo permite una detección más acertada en caso de un incendio y el sensor de gas natural permite medir o detectar fugas de gas natural para evitar accidentes como un incendio o intoxicación por el gas natural. Para la detección de estas fugas de gas natural el cual está compuesto por metano (CH4) en un 90% y el humo el cual está compuesto en su mayoría por monóxido de carbono (CO) se tomaron en cuenta los sensores de la Tabla 3.

Tabla 3. Sensores de humo *y Sensor de Gas natural*

Nombre	Imagen	Rango	Sensibilidad	Detección	Output	Precio
MQ-7	 **Nota:** Fuente https://acortar.link/Rwovej	10~500 ppm	$\frac{Rs_{(in\ air)}}{Rs_{(150ppm\ CO)}} \geq 5$	CO	Digital/ Análoga	US $3,24
MQ-9	 **Nota:** Fuente https://acortar.link/wpVpNI	10~1000 ppm CO 100~10000 ppm CH4, LPG	$\frac{Rs_{(in\ air)}}{Rs_{(100ppm\ CO)}} \geq 5$	CO, CH4, LPG	Digital/ Análoga	US $3,24
MQ-5	 **Nota:** Fuente https://acortar.link/55cRHT	300~10000 ppm	$\frac{Rs_{(in\ air)}}{Rs_{(2000ppm\ C_3H_8)}} \geq 5$	LPG, CH4	Digital/ Análoga	US $3,24
MQ-4	 **Nota:** Fuente https://acortar.link/uBlu4P	200~10000 ppm	$\frac{Rs_{(in\ air)}}{Rs_{(5000ppm\ CH_4)}} \geq 5$	CH4, natural gas	Digital/ Análoga	US $3,24

Para el sensor de humo y de gas natural se decidió hacer uso del sensor MQ-9, el cual como se puede observar en la Tabla 3 posee un rango de detección amplio y una sensibilidad mayor que la del MQ-7, MQ-4 y el MQ-5. El MQ-9 nos permite medir los gases como CO, CH4

y LPG lo que elimina la necesidad de tener un sensor de gas para medir CO y otro para medir gas natural reduciendo el costo y el espacio necesario para la detección de estos gases.

6.2.3 Sensor de apertura de puerta y ventana

Para la detección de apertura se decidió hacer uso de un interruptor magnético (Figura 3) , el cual funciona como un conductor cuando siente el campo magnético del imán, al separarse el interruptor se comporta como un circuito abierto, permitiendo de esta forma poder enviar señales digitales como si fuera un interruptor. Se consideró hacer uso de los interruptores magnéticos que cuentan con conexión a Wifi, pero se descartó esta idea debido a su costo y al problema de compatibilidad que podría tener con la unidad central.

Figura 3. Interruptor Magnético

Nota: Fuente https://acortar.link/8G55S4 Página web Tienda DELTASTORES.GR.

6.3 Conexión con la unidad central

Para la conexión de los sensores con la unidad central se decidió hacer uso de los módulos wifi ESP-01 (Figura 4) , el cual cuenta con las funciones que ofrece el microcontrolador ESP8266, permite conectarse a una red wifi de 2.4GHz y enviar datos a través de una red wifi, soporta WPA / WPA2 y configurar en 3 modos diferentes los cuales son AP, Station y AP + Station, este módulo wifi también cuenta con 2 pines GPIO y tiene la ventaja de ser de un tamaño relativamente pequeño y funciona con un voltaje de 3.3VDC. este módulo se usará para conectar los sensores de apertura magnética con la unidad central.

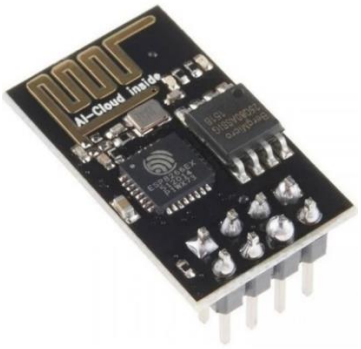


Figura 4. Modulo Wifi ESP-01

Nota: Fuente https://acortar.link/cgqkyC Pagina web Educativa DEVICE PLUS.

Para conectar los sensores de temperatura y los sensores de gas a la unidad central se hizo uso del módulo NodeMCU (Figura 5) el cual está fabricado usando el microcontrolador ESP8266, su diferencia con el módulo ESP-01, es que cuenta con muchos más pines y se puede alimentar con 5VDC, el inconveniente es que ocupa más espacio que el módulo ESP-01 pero es ideal para conectar varios dispositivos a la vez.

Figura 5. NodeMCU

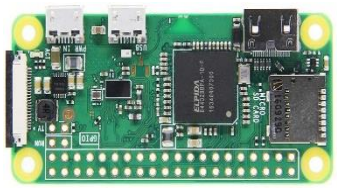
Nota: Fuente https://acortar.link/SRVsIQ Pagina web de Tienda ASK Electronics.

6.4 Unidad central

La unidad de procesamiento central es quien se encarga de procesar y subir los datos recibidos de los sensores a la base de datos para que los muestre la plataforma IoT, para la elección de la unidad central se tomaron en cuenta los dispositivos presentados en la Tabla 4.

Tabla 4. Unidades centrales de procesamiento

Nombre	Imagen	Memoria	Velocidad	Wifi	Consumo	Precio
Arduino Mega	 **Nota:** Fuente https://acortar.link/6YEfIL	SRAM 8KB	16 MHz	No posee	1W (5V)	US $33.30

Nombre	Imagen	Memoria	Velocidad	Wifi	Consumo	Precio
Raspberry Pi 4	 **Nota:** Fuente https://acortar.link/Vd2dYc	RAM 8GB	1.5 GHz (4 Cores)	Dual band Wifi, BT	12.5 W (5V)	US $209
Raspberry Zero W	 **Nota:** Fuente https://acortar.link/FLtxuO	RAM 512MB	1 GHz	Wifi, BT	0.8W (5V)	US $74.99

De acuerdo a la Tabla 4 se tomó la decisión de usar la Raspberry Pi Zero W, esta elección se realizó debido a sus características de hardware las cuales son superiores a las del Arduino Mega y proporciona conexión a wifi y bluetooth, las cuales soportan el protocolo 802.11 b/g/n y bluetooth 4.1, lo que permite una mayor flexibilidad al momento de su instalación, también cuenta con la característica de tener un bajo consumo de energía, un precio accesible y un tamaño pequeño (65mm x 30mm), contrario a la Raspberry Pi 4, la cual aunque cuenta con características de hardware superiores a la Raspberry Pi Zero W, esta cuenta con un precio más elevado y un tamaño más grande (85mm x 56mm), además sus características podrían estar sobre dimensionadas para las necesidades del proyecto.

6.5 Selección de Protocolos de comunicación

Un protocolo de comunicación es un conjunto de reglas y procedimientos, estos tienen una estructura, formatos definidos, y una serie de pasos que se deben realizar para que la comunicación sea establecida, como son procesados y transmitidos los datos.

Entre los múltiples protocolos de comunicación, para la comunicación entre los sensores y la unidad central usamos MQTT ya que este posee ciertas ventajas respecto a el protocolo HTTP para este caso específico, a continuación, en la Tabla 5 se hace una comparación de MQTT el protocolo HTTP.

Tabla 5. Protocolos de comunicación

Protocolo	Arquitectura	Tamaño y velocidad	Seguridad
MQTT	Se enfoca en trasmitir datos a nivel de byte, funciona con el modelo pub/sub, la conexión puede mantener la conexión, buen funcionamiento con dispositivos pequeños, comunicación full-dúplex	Es muy ligero ya que su cabecera más pequeña puede pesar 2 bytes. Mucho más rápido y es el consumo energético es mucho menor por lo que es adecuado para usarse en dispositivos de bajo consumo.	Acepta usuario y contraseña, y cifrado haciendo uso de certificados SSL/TLS
HTTP	Http funcionan mediante un modelo cliente-servidor, donde solo abre la crea la conexión cuando desea realizar una petición, orientada a documentos (XML, JSON), establece una comunicación Half-duplex	Usa más recursos para realizar cada petición, puede ser usado para enviar grandes volúmenes de datos.	Al igual que MQTT, hace uso de SSL/TLS, puede usar JWT

Como se muestra en la Tabla 5, se decidió usar el protocolo MQTT porque es más ligero y eficiente por lo tanto necesita recursos mínimos para funcionar en dispositivos de bajo costo,

facilita la transmisión de mensajes, y es un protocolo seguro al poder hacer uso de cifrado de mensajes con TLS y autenticación para la conexión.

6.6 Selección Bróker MQTT

Un bróker MQTT es un servidor encargado de organizar las transacciones entre las dos partes, este es el encargado de recibir los mensajes que han sido publicados por un cliente, estas publicaciones deben estar etiquetadas es decir debe ir dirigidas a un tema y los redirecciona a los clientes que estén suscriptos a dicho tema o tópico.

Hay diferentes brókeres MQTT algunos mejores que otros depende de la aplicación y la escalabilidad del proyecto, para esto se tomaron en cuenta distintos brókeres, como se muestra en la Tabla 6.

Tabla 6. Brókeres MQTT

Bróker	Características
Mosquitto	Open Source, implementa versiones las versiones 3.1, 3.1.1, 5.0 de MQTT, ligero, se puede instalar en dispositivos de baja potencia y servidores, fácil uso y configuración, basado en C, multiplataforma. *(Eclipse Foundation, s.f.)*
Emqttd	Multiplataforma, compatibilidad, con el protocolo MQTT en sus versiones 3.1, 3.1.1, escalable masivamente hasta 1 millón de conexiones, OpenSource, compatible con Websockets, CoAp, escrito en el lenguaje de programación Earlang. *(Lee, s.f.)*
HiveMQ	Bróker robusto diseñado para aplicaciones exigentes con un gran número de dispositivo, está basado en Java, compatible con versiones de MQTT 3.X y 5, Con versión gratuita se permite conectar hasta 100 dispositivos. *(HiveMQ, s.f.)*
Aedes/Mosca	Compatible con MQTT 3.1, 3.1.1, no posee una gran variedad de características, pero puede ejecutarse en Windows y Linux, Basado en Node.js. *(Collina, s.f.)*

Se decidió usar Mosquitto ya que es fácil de instalar y ligero al ser desarrollado en c, y no consume tantos recursos como algunos de los brókeres anteriormente mostrados, Mosquitto es un bróker completo, que nos proporciona seguridad, eficiencia, para este proyecto.

6.7 Firebase

Se seleccionó Firebase, porque es una plataforma en la nube que posee múltiples herramientas que serán muy útiles para el despliegue de nuestro proyecto, Firebase posee un apartado de autenticación, una base de datos NoSQL, una base de datos real time, un storage donde se pueden almacenar archivos y muchas otras funciones, tiene una fácil integración con distintas aplicaciones y lenguajes, y actuara como el backend del proyecto.

Firebase es una plataforma creada por Google que proporciona características para realizar una aplicación web o aplicaciones móviles, esta consta de diferentes servicios los cuales ayudan al desarrollo del proyecto.

6.7.1 Firebase Realtime

Este servicio permite almacenar datos y consultarlos en tiempo real, los datos se estructuran, y almacenan en formato JSON. Cada dato se sincroniza en tiempo real, por lo que si varios clientes comparten una instancia de la Base de Datos podrán recibir los datos al instante, esta herramienta nos permitirá realizar el monitoreo del estado de variables dentro de la residencia. (Google Developers, 2022)

6.7.2 Firestore Database

Firestore es una base de datos NoSQL, el almacenamiento de los datos es muy diferente a la forma tradicional de una base de datos SQL, ya que los datos no se almacenan en tablas ni filas, si no que se basa en documentos y colecciones.

Un documento es un registro liviano con campos y estos tienen una estructura “clave: valor”, un documento puede ser análogo a una fila en una tabla de SQL, pero no es del todo así, ya que un documento puede tener dentro de sí una colección de documentos y un campo puede tener más campos dentro de él (Google Developers, 2022). Este servicio permite almacenar rutas del proyecto, datos de los sensores, y datos relevantes para la aplicación.

6.7.3 Cloud Storage

Es el servicio de almacenamiento de objetos de Firebase en el que se puede guardar archivos, videos, audios y otros tipos de contenidos, puede ser usado para subir y descargar elementos, para la aplicación el Storage solo almacenara las imágenes que se mostraran por los diferentes apartados de la aplicación.

6.7.4 Cloud Messaging

Firebase Cloud Messaging es un servicio en la nube que permite el envío de mensaje y notificaciones, con el que se puede captar la atención del usuario, para así mantenerlo informado en caso de una emergencia u otro incidente. Esta herramienta al ser multiplataforma brinda flexibilidad a la hora de desarrollar aplicaciones.

6.7.5 Firebase Realtime

Este servicio permite almacenar datos y consultarlos en tiempo real, los datos se estructuran, y almacenan en formato JSON. Cada dato se sincroniza en tiempo real, por lo que si varios clientes comparten una instancia de la Base de Datos podrán recibir los datos al instante, esta herramienta nos permitirá realizar el monitoreo del estado de variables dentro de la residencia. (Google Developers, 2022)

6.8 Lenguajes de programación

Los lenguajes de programación son lenguajes que permiten escribir mediante instrucciones, un conjunto de ordenas, que deseamos que haga una máquina, con ellos podemos manipular datos, crear algoritmos, y mucho más, los lenguajes son la forma de cómo nos comunicamos con las maquinas, cada uno posee una estructura, forma de manipular los datos, pablaras clave y una reglas para que la maquina pueda interpretarlo; actualmente existe una gran variedad de estos y son usados para distintas cosas, desde hacer mover un mecanismo robótico, hasta crear una página web.

Los lenguajes de programación que se escogieron fueron Python para el procesamiento de los datos, C++ para la programación de los dispositivos embebidos, JavaScript para el desarrollo de la lógica de la aplicación, y librerías como React.js para la creación de la aplicación.

6.8.1 React.js

Es una biblioteca de JavaScript, creada por Facebook que nos ayuda a la creación de interfaces de usuario interactivas, es muy potente ya que esta ayuda a renderizar y actualizar de manera eficiente nuestra aplicación, funciona por medio de componentes, lo que hace que al cargar la página solo se carguen los datos, las vistas o los componentes que la aplicación necesita, para funcionar.

6.8.2 Python

Es un lenguaje de programación multiplataforma, de código abierto, que se caracteriza por ser de tipado dinámico y por la legibilidad de su código, es un lenguaje que se puede usar para un gran número de aplicaciones, usado para el procesamiento de datos, inteligencia artificial, e incluso para la web, Python al ser un lenguaje que puede correr en múltiples sistemas y de fácil instalación, cuenta con una gran comunidad y un gran número de librerías tanto propias, como creadas por terceros que nos ayudaran a solucionar y automatizar procesos.

7. Desarrollo del Proyecto

En este capítulo se presenta el diseño completo del proyecto, describiendo cada elemento y que función cumple dentro del mismo.

En la Figura 6 se presenta el diagrama de bloques del proyecto el cual se explicará a continuación.

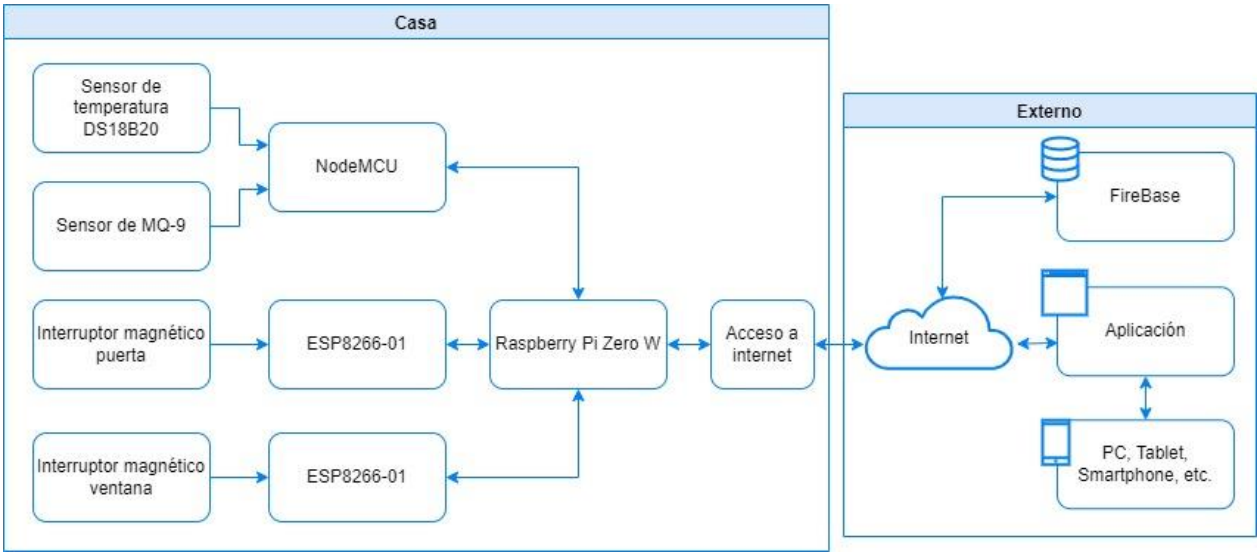


Figura 6. Diagrama de bloques

7.1 Sensor de temperatura DS18B20

El sensor de temperatura DS18B20 es un sensor tipo sonda (Figura 7) el cual se caracteriza por requerir un único pin puerto para comunicación, además de poder ser configurado en modo parasito lo que le permite alimentarse desde la línea de datos, este sensor se alimenta

con un rango de voltajes de entre 3.0V a 5.5V lo que lo hace ideal para conectarse a diversos tipos de microcontroladores los cuales en su mayoría trabajan con voltajes entre 3.3V y 5V. Como se mostró en la Tabla 2 este sensor permite medir las temperaturas entre -55ºC a 125ºC con una precisión de ±0.5ºC cuando la medición de la temperatura se encuentra entre los -10ºC a 85ºC, este sensor nos permite programar la resolución del termómetro de 9 a 12 bits.

Este sensor tiene la función de medir la temperatura de la cocina o de cualquier sección de la casa que se desee monitorear para poder detectar el incremento de temperatura generado por los incendios, este sensor en conjunto con el sensor de CO permite determinar si se ha iniciado un incendio.

Figura 7. Sensor DS18B20

Nota: Fuente https://acortar.link/1TywBD página web Tienda PCBoard.ca

La conexión del sensor DS18B20 se realiza como se puede observar en la Figura 8, el pin VDD del sensor se conecta directamente a +5V, el pin GND del sensor también se conecta directamente al pin GND del NodeMCU, el pin DQ el cual es el pin de datos se conecta al pin digital D1 del NodeMCU, se debe conectar una resistencia de pull-up de 4.7kΩ entre VDD y el

pin de datos DQ , el motivo de esta resistencia es debido a la electrónica para controlar el bus de comunicación, utiliza un FET de drenaje abierto que se comporta como una puerta AND.

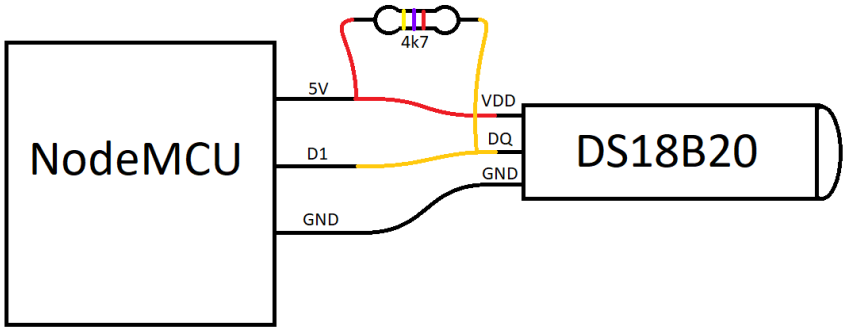


Figura 8. Conexión Sensor DS18B20 al NodeMCU

7.2 Sensor MQ-9

El material sensible del sensor de gas MQ-9 (Figura 9) es SnO2, que tiene una conductividad más baja en aire limpio. Realiza la detección por el método de ciclo de alta y baja temperatura, y detecta CO cuando la temperatura es baja (calentada por 1,5 V). La conductividad del sensor es más alta junto con el aumento de la concentración de gas.

Figura 9. Sensor de gas MQ-9

Nota: Fuente https://acortar.link/y5C9vo página web Tienda BIGTRONICA.

Cuando la temperatura es alta (calentada por 5,0 V), detecta metano, propano, gas combustible y limpia los otros gases absorbidos a bajas temperaturas. El sensor de gas MQ-9 tiene alta sensibilidad al monóxido de carbono, metano y GLP lo que podría usarse para detectar diferentes gases que contienen CO y gases combustibles, es de bajo costo y adecuado para diferentes aplicaciones. En la Figura 10 se muestran las curvas del sensor de gas, en esta gráfica se puede observar una curva logarítmica en donde hay una relación Rs/Ro que toma valores entre 0.1 y 10 y otra relación llamada ppm o partes por millón que mide la concentración de gas que se encuentra en el ambiente. Para que el sensor MQ-9 realice correctamente las mediciones se le debe hacer un proceso de "curado" en donde se debe dejar conectado el sensor por al menos 24h antes de usarlo.

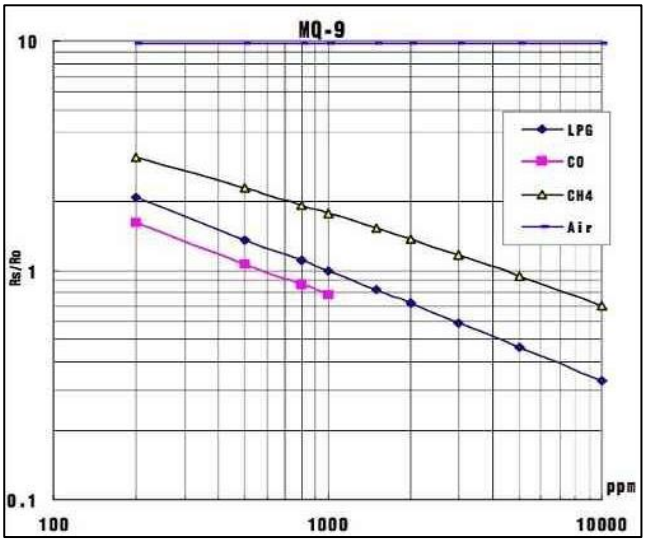


Figura 10. Curva MQ-9

Nota: Fuente (HANWEI, 2022)

A continuación, se mostrarán los cálculos que se realizaron para la calibración del sensor. Haciendo uso de la curva de CO de la Figura 10, se extraen los puntos para realizar una linealización de los datos. Haciendo uso de Excel se creó una tabla con los puntos de la curva CO como se puede observar en la Tabla 7, con estos puntos se grafica la curva CO como se muestra en la Figura 11.

Tabla 7. Puntos Curva CO

ppm	Rs/Ro
200	1,7
300	1,46
400	1,3
500	1,1
600	0,99
700	0,9
800	0,85
900	0,81
1000	0,78

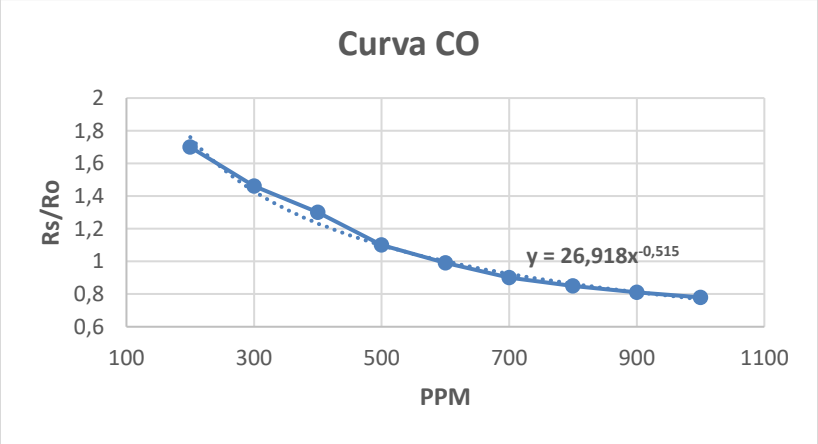


Figura 11. Curva CO

Se realiza una linealización de los datos de la Tabla 7 en donde a cada punto de la tabla se le saca el logaritmo, lo que da como resultado la Tabla 8. Con estos datos se forma la gráfica que se muestra en la Figura 12 de la cual obtenemos la ecuación de la recta, de esta ecuación obtenemos la pendiente de la recta y un punto que la conforma.

Tabla 8. Puntos curva CO linealizados

Log (ppm)	Log (Rs/Ro)
2,301	0,230
2,477	0,164
2,602	0,114
2,699	0,041
2,778	-0,004
2,845	-0,046
2,903	-0,071
2,954	-0,092
3,000	-0,108

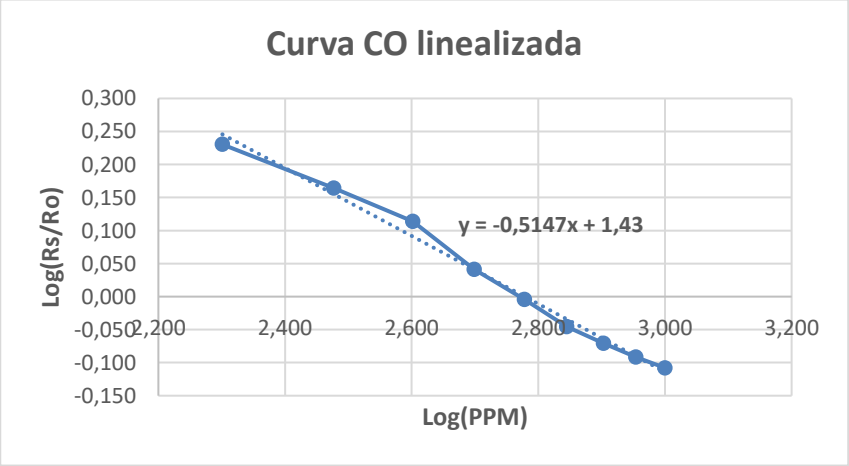


Figura 12. Curva CO Linealizada

El mismo procedimiento que se realizó para linealizar la curva CO, se realiza con la curva CH4 y LPG, en la Figura 13 y Figura 14 se pueden observar la ecuación de la recta para la curva LPG y CH4, respectivamente, una vez se obtienen las pendientes y un punto de cada recta se guardan en unas variables en el programa del NodeMCU.

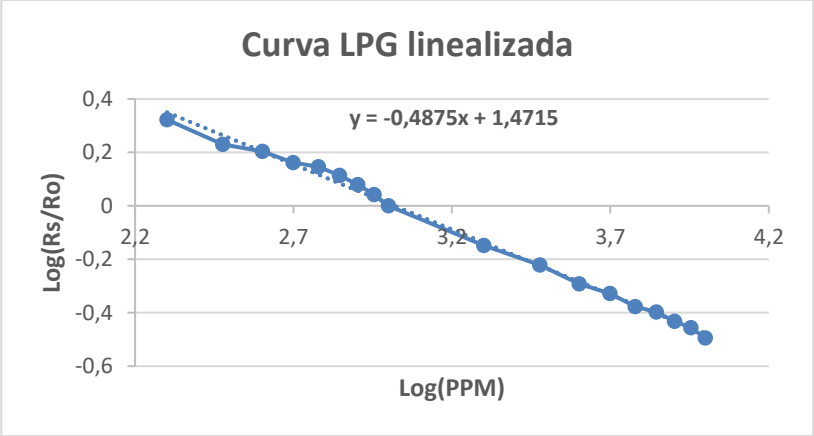


Figura 13. Curva LPG Linealizada

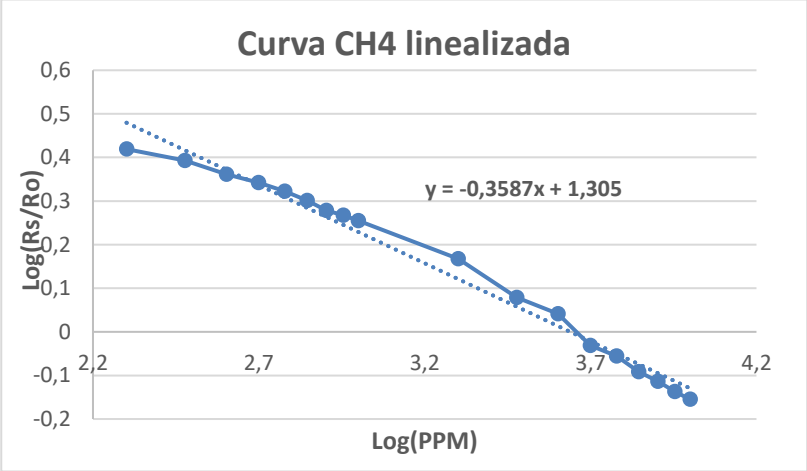


Figura 14. Curva CH4 Linealizada

En el programa se hace el cálculo de Rs (la resistencia generada por la detección del gas objetivo), Rs se calcula con el voltaje que entrega el sensor MQ-9 al ADC (analog digital convert) del NodeMCU y la Ecuación 1, el ADC del NodeMCU posee una resolución de 10 bits, lo que equivale a 1023 posibles combinaciones de bits, el valor de voltaje del MQ-9 que recibe el ADC lo convierte en una combinación de bits entre 0 y 1023, este valor se opera en la Ecuación 1 y así se obtiene el valor Rs, el valor RL es el valor de la resistencia de carga del sensor en el caso del sensor MQ-9 que se posee es de 1kΩ.

$$Rs = Rl\frac{(1023 - voltaje_{MQ-9})}{voltaje_{MQ-9}}$$

Ecuación 1. Calculo Rs

Luego el valor de Rs se divide entre la resistencia Ro, la cual es la resistencia del sensor en aire limpio, este valor Ro se obtiene con una función en el programa la cual hace un muestreo con los datos medidos por el sensor en ausencia de gas, este muestreo lo divide luego entre el intervalo de muestreo y lo divide entre el factor de resistencia del aire conocido, el cual se puede consultar en la Figura 10, en donde se puede observar que toma un valor de 9.87 aproximadamente.

Finalmente, el valor Rs/Ro se inserta en una función que realiza la operación en la Ecuación 2 la cual se obtiene al despejar ppm de la Ecuación 3, donde b es igual a la ecuación 4.

$$ppm = 10^{\frac{\log\left(\frac{Rs}{Ro}\right)-Y}{M}+X}$$

Ecuación 2. Calculo ppm

$$\frac{Rs}{Ro} = b.ppm^{M}$$

Ecuación 3. Ecuación de la curva potencial MQ-9

$$b = 10^{-MX+Y}$$

Ecuación 4. Cálculo de b con el punto y la pendiente

La conexión del sensor MQ-9 al NodeMCU se realiza de manera como se muestra en la Figura 15, en donde se puede observar que el pin VCC se conecta directamente a +5V, el pin GND del MQ-9 se conecta directamente al pin GND del NodeMCU, y por último se conecta el pin AO del MQ-9 al pin análogo A0 del NodeMCU.

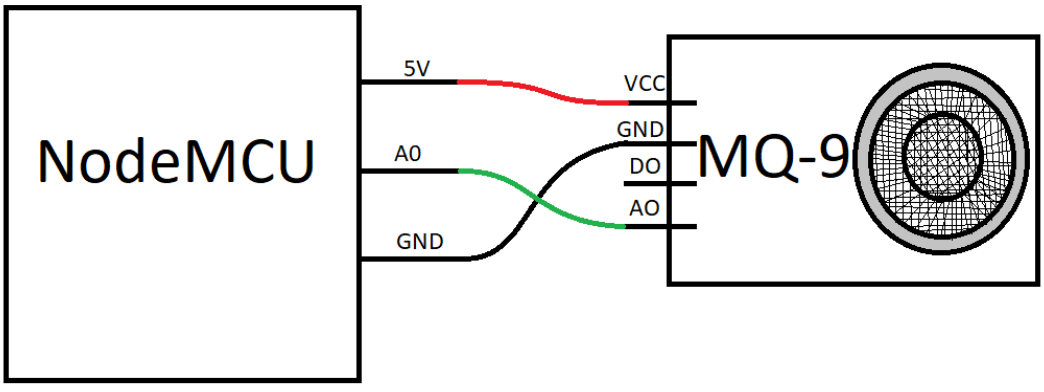


Figura 15. Conexión MQ-9 al NodeMCU

7.3 Interruptor magnético

El interruptor magnético es un interruptor conformado por dos láminas ferromagnéticas compuestas de Ni y Fe, selladas herméticamente en una capsula de vidrio. Las láminas están superpuestas internamente dejando un espacio entre ellas, las cuales en presencia de un campo magnético se juntan permitiendo el paso de corriente, de esta manera se pueden enviar señales de on/off con el uso de este interruptor como se puede observar en la Figura 16. Este sensor cumple la función de determinar si la puerta o la ventana están abiertas o cerradas esto lo hace enviando una señal digital de 1 o 0 al módulo wifi ESP-01.

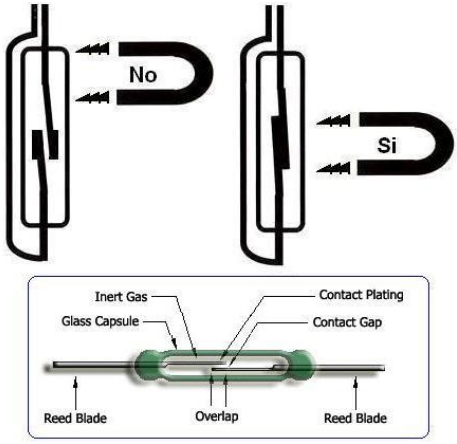


Figura 16. Funcionamiento Interruptor magnético

Nota: Fuente https://acortar.link/ppmxFL página web Tienda SHOPTRONICA.

La conexión del interruptor magnético se realiza como se puede observar en la Figura 17 en donde una punta del Interruptor se conecta a la entrada IO2 del ESP-01 y la otra entrada se conecta directamente a +3.3V, se conecta una resistencia de 330Ω entre el pin IO2 y GND, esto

se hace para que el ESP-01 detecte que se está enviando un nivel bajo de energía cuando el sensor este abierto.

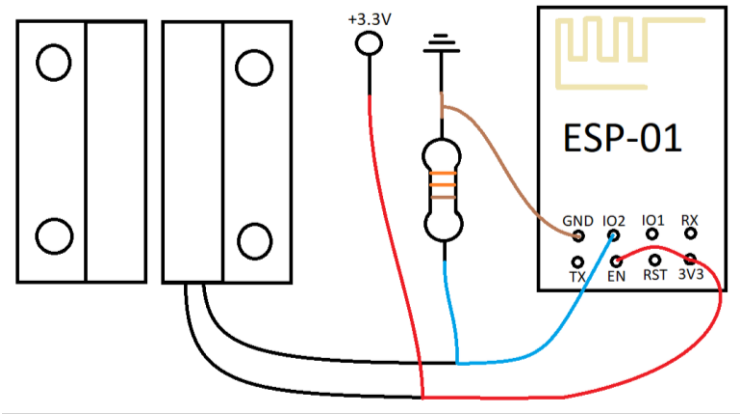


Figura 17. Conexión Interruptor Magnético con el ESP-01

7.4 NodeMCU

El NodeMCU es una placa de desarrollo la cual está basada en el microcontrolador ESP8266 el cual soporta el protocolo de comunicación Wifi y es compatible con el estándar IEEE 802.11 b/g/n, además de soportar seguridad WEP, WPA y WPA2, lo que lo hace muy versátil al permitir enviar datos a través de una red local o de internet de una manera segura, esta placa cuenta con varios pines digitales y solo cuenta con un pin analógico el cual cuenta con un ADC de 10 bits, esta placa se puede alimentar con un voltaje de entre 3.3V y 5V lo que lo hace ideal como modulo wifi para diferentes tipos de sensores como se puede observar en la Figura 18.

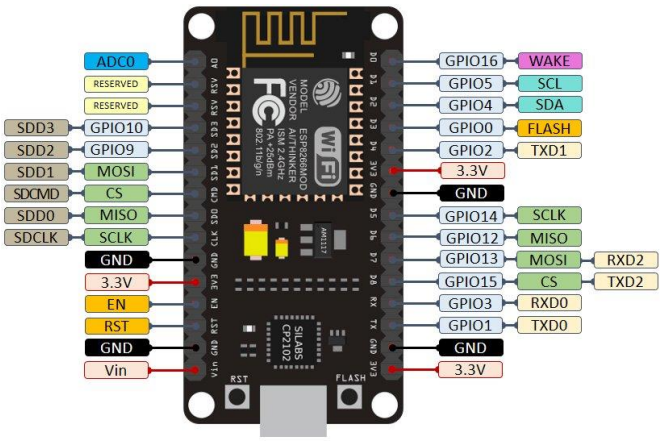


Figura 18. NodeMCU Pines

Nota: Fuente https://acortar.link/4aaEcQ Pagina web Educativa Robots Didácticos.

El NodeMCU tiene la función de recibir los datos del sensor de temperatura DS18B20 y el sensor MQ-9, y enviarlos a través de wifi hacia el bróker usando el protocolo MQTT. El NodeMCU debe procesar los datos recibidos de los sensores antes de enviarlos para que estos tengan sentido, y para esto se debe incluir varias librerías, entre las cuales están OneWire.h y DallasTemperature.h las cuales son necesarias para la lectura de los datos entregados por el DS18B20, el cual se conecta a un pin digital del NodeMCU. Otras librerías que son necesarias son ESP8266WiFi.h la cual nos permite conectarnos a una red y PubSubClient.h la cual permite usar el protocolo MQTT para conectarse al bróker, suscribirse en un tema y publicar los datos recibidos por los sensores. El NodeMCU tiene conectado en su pin analógico el sensor de gas MQ-9, el cual le entrega un voltaje dependiendo de la concentración del gas que se está midiendo, para poder determinar correctamente la concentración en partes por millón (ppm) del gas que se está midiendo se deben hacer unos cálculos para poder calibrarlo.

7.4.1 Programa del NodeMCU

La función reconnect de la Figura 19 permite realizar la conexión del NodeMCU al bróker.

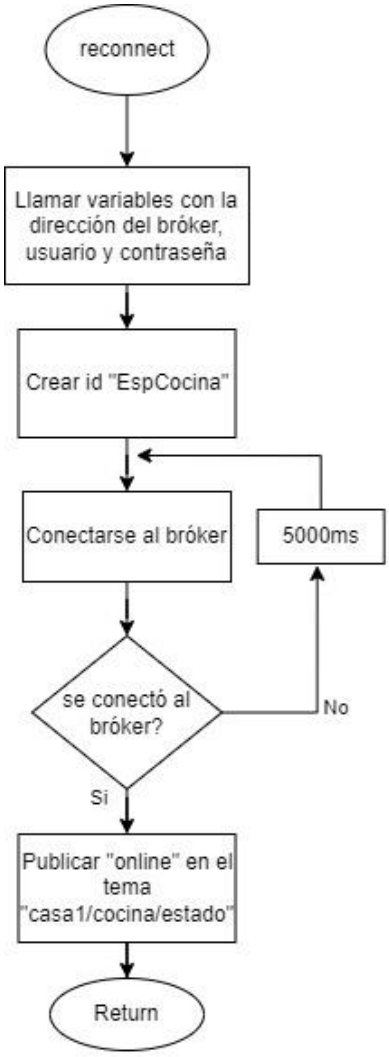


Figura 19. Función conexión al bróker

Para realizar la conexión con el bróker se llama la función client.connect() de la librería PubSubClient.h la cual recibe como parámetros la ID del dispositivo y las credenciales de acceso

del bróker y si se desea se puede establecer un mensaje Last Will and Testament, el cual el bróker enviará a los suscriptores en caso de que se pierda la conexión con el NodeMCU. Una vez se conecta el Node al bróker publica un mensaje de “online” al tema “casa/cocina/estado”. Si el NodeMCU no se logra conectar al bróker lo intenta de nuevo después de 5 segundos y lo sigue intentando hasta que logra conectarse.

La función MQ resistencia cálculo de la Figura 20 se encarga de leer los datos medidos por el sensor MQ-9 conectado en el pin análogo del NodeMCU, para luego procesarlos y calcular el valor de la resistencia MQr, la cual representa la concentración del gas que se está midiendo. Para realizar el cálculo de la resistencia el NodeMCU ingresa el voltaje que se está recibiendo en el pin análogo en la fórmula que se puede observar en la Figura 20 y luego retorna el valor de la resistencia calculado.

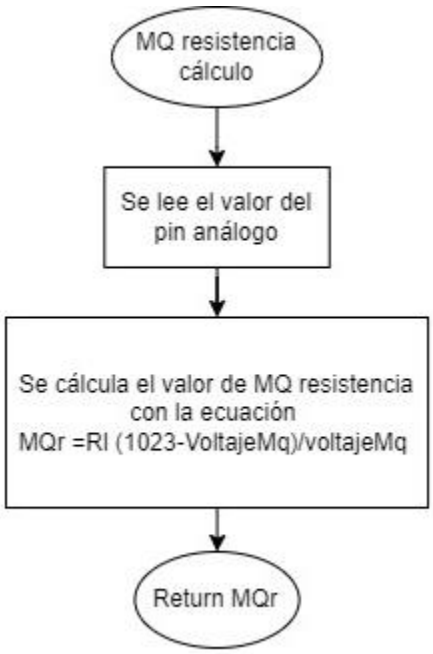


Figura 20. Función MQ resistencia calculo.

La función Calibración Ro MQ en la Figura 21 se encarga de calibrar el valor de la resistencia Ro.

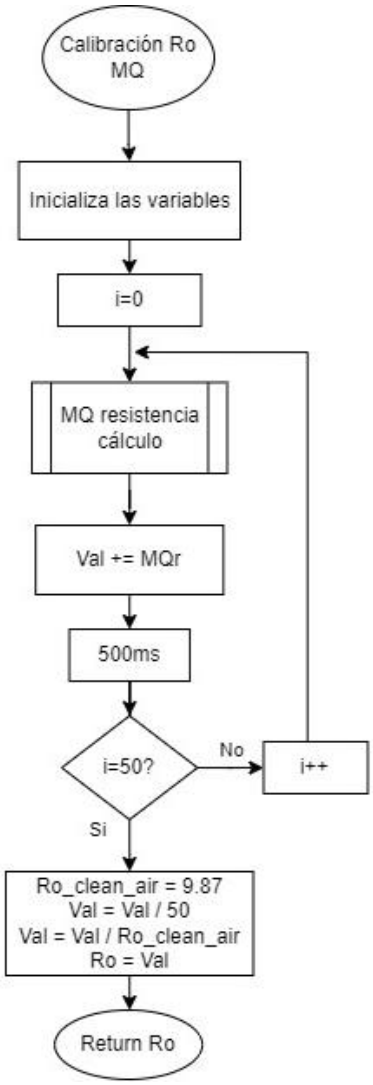


Figura 21. Calibración de la resistencia Ro

La resistencia Ro representa la concentración del gas medido por el sensor MQ-9 en presencia del aire limpio, lo que es necesario para realizar el cálculo de las partes por millón

(ppm), ya que las ppm se calculan por medio de la relación entre la resistencia entregada por el sensor en presencia del gas de interés con respecto a la resistencia entregada por el sensor en presencia del aire limpio, esta calibración del Ro se lleva a cabo llamando la función MQ resistencia calculo, la cual se mide 50 veces y se calcula el promedio de todas estas mediciones. Esta resistencia promedio luego se divide entre el factor Ro_clean_air, el cual es el valor aproximado que toma la relación Rs/Ro en presencia del aire limpio, este valor se puede tomar de la curva del MQ-9 . Una vez se hizo ese cálculo la función retorna el valor de la resistencia Ro.

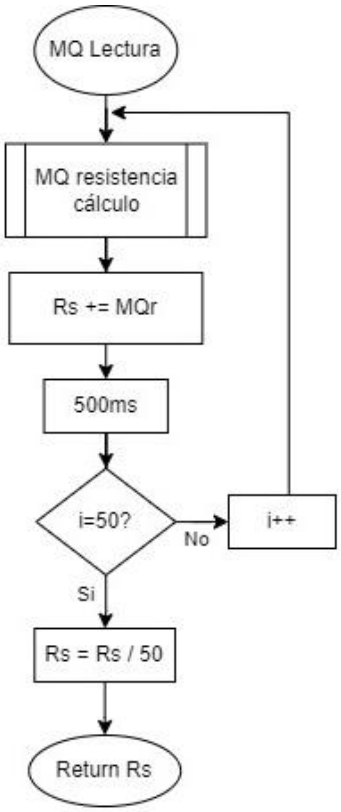


Figura 22. Lectura sensor

La función MQ Lectura de la Figura 22 se encarga de realizar el cálculo de la resistencia Rs promedio, la cual representa la concentración del gas medido por el sensor MQ-9, este cálculo

lo realiza llamando la función MQ resistencia calculo, el valor que entrega esta función se mide 50 veces cada medio segundo y se halla el promedio, el cual es el valor que se toma como Rs para luego retornarlo.

La función MQ PPM de la Figura 23 se encarga de realizar el cálculo de las ppm medidas por el sensor MQ-9, para realizar el cálculo se llama la función MQ Lectura, la cual retorna el valor de Rs el cual se divide por el valor Ro, el cual se calcula en la función Calibración Ro MQ, luego se llaman las variables Y, X y M las cuales contienen el punto y pendiente de la recta calculada en la sección 7.2 y se ingresa en la fórmula que se puede observar en la Figura 23. Una vez se calcula la ppm para la concentración del gas en el aire se retorna dicho valor.

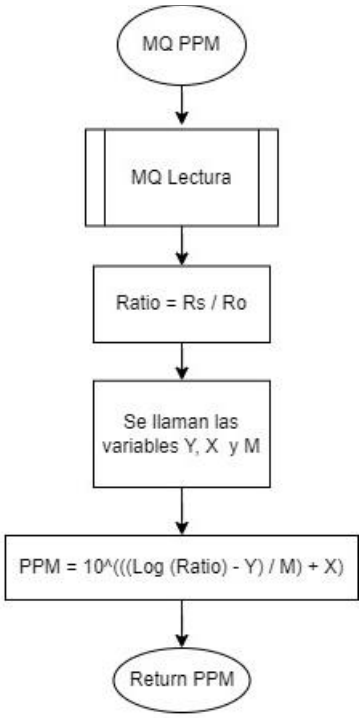


Figura 23. Cálculo de las ppm

En el diagrama de flujo general de la Figura 24 se puede observar el diagrama de flujo principal llamado NodeMCU.

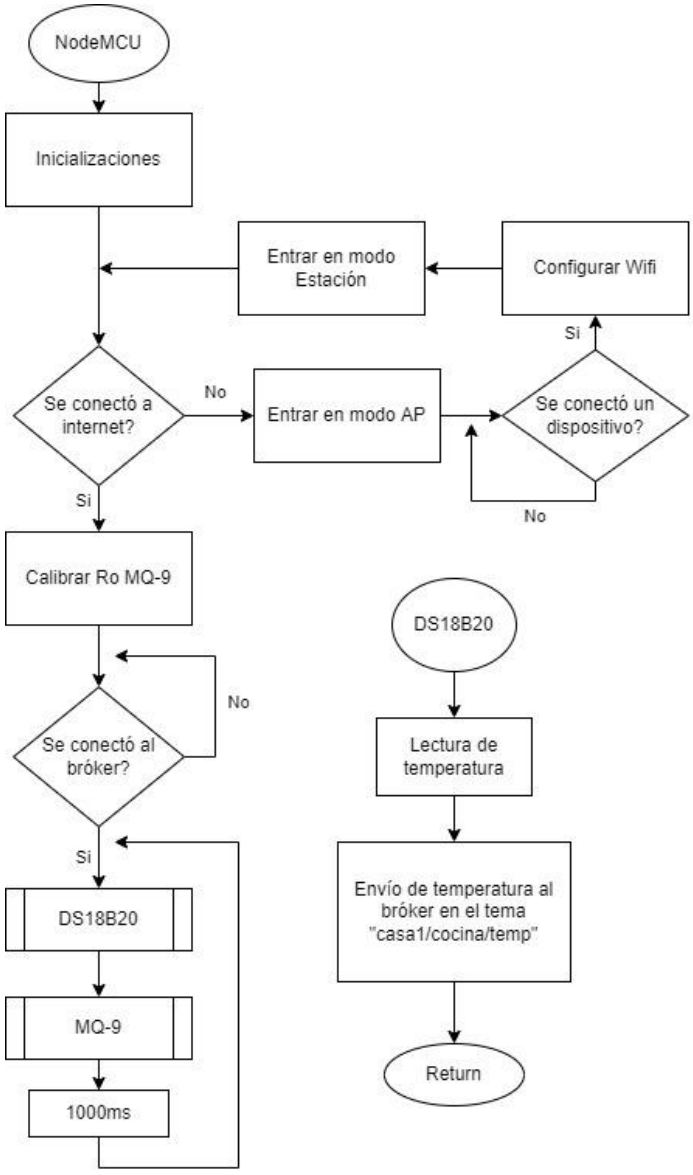


Figura 24. Diagrama de flujo general NodeMCU

El primer proceso que se realiza es inicializar las variables que se necesitan, luego se realiza la conexión a internet del NodeMCU, donde para realizar la conexión a la red Wifi se debe de disponer del SSID y Password de la red Wifi. Se le importo una librería llamada WiFiManager.h la cual permite realizar la conexión a la red Wifi, y en caso de no conectarse cambia el modo estación del NodeMCU a modo AP, donde se crea una red Wifi, en la cual conectándose desde cualquier dispositivo con Wifi se configura la SSID y el Password de la red a la cual se desea conectar el NodeMCU, una vez se hace esta configuración el NodeMCU se cambia de nuevo a modo estación y se intenta conectar nuevamente a la red.

Una vez el NodeMCU está conectado a una red Wifi continua con la calibración del Ro del sensor MQ-9 la cual llama la función Calibración Ro MQ, luego se inicia la conexión al bróker y para eso se llama la función reconnect mostrada en la Figura 19.

Luego se procede a llamar la función DS18B20 (Figura 25), la cual lee la temperatura enviada por el sensor al pin digital del NodeMCU y los publica en el bróker, para hacer esto se importan las librerías OneWire.h y DallasTemperature.h, se llama la función OneWire oneWire() en la cual se ingresa como parámetro el pin digital del cual se están leyendo los datos del sensor DS18B20, también se llama a la función sensorDS18B20.getTempCByIndex() la cual entrega el valor de la temperatura medido por el sensor para luego publicarlo en el bróker en el tema "casa1/cocina/temp".

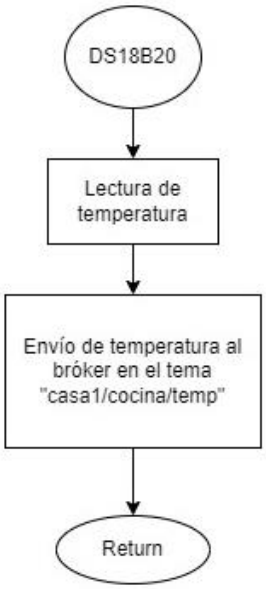


Figura 25. Función DS18B20

Por último se llama la función MQ-9 (Figura 26) en la cual se realiza la lectura, calculo y envío de las ppm hacia el bróker de los gases CO, CH4 y LPG, para realizar la lectura se llama la función MQ Lectura en la cual calcula el valor de Rs, luego se llama la función MQ PPM, en la cual se hace el cálculo de las ppm, esta función se llama 3 veces ya que el mismo sensor está midiendo 3 gases diferentes, el sensor envía un único valor de resistencia, con el cual se calcula la relación Rs/Ro, esta relación se evalúa en cada una de las curvas mostradas en la sección 7.2, lo que da como resultado el valor de ppm medido por el sensor, luego se envían los valores de ppm calculados al bróker en los temas "casa1/cocina/LPG", "casa1/cocina/CH4" y "casa1/cocina/CO".

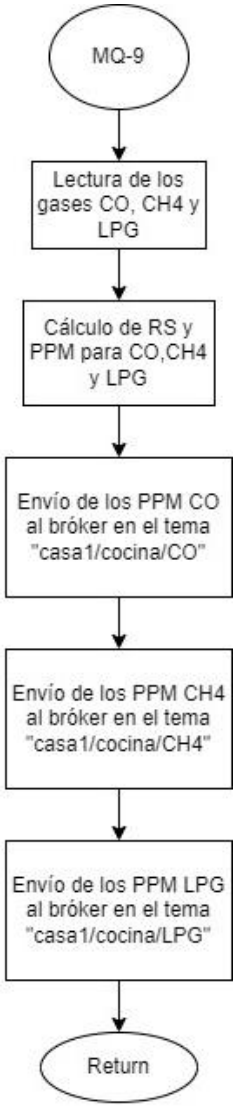


Figura 26. Función MQ-9

7.5 Modulo Wifi ESP8266-01

El ESP8266-01 o ESP-01 para abreviar es un módulo Wifi basado al igual que el NodeMCU en el microcontrolador ESP8266, lo que diferencia este módulo del NodeMCU es su tamaño reducido y el número de pines con los que cuenta, donde solamente cuenta con 2 pines

digitales, 2 pines para Rx y Tx que se pueden utilizar como pines digitales como se puede observar en la Figura 27, la desventaja de este módulo es que no cuenta con ningún pin analógico lo que limita un poco su uso.

Este módulo tiene la función de recibir la señal on/off del interruptor magnético y enviarla, esto lo hace conectándose a la red wifi y enviando los datos “open” y “close” hacia el bróker usando el protocolo MQTT. El interruptor magnético y el ESP-01 de la puerta y la ventana tienen el mismo funcionamiento, la diferencia es el tema en el cual estos publican los datos.

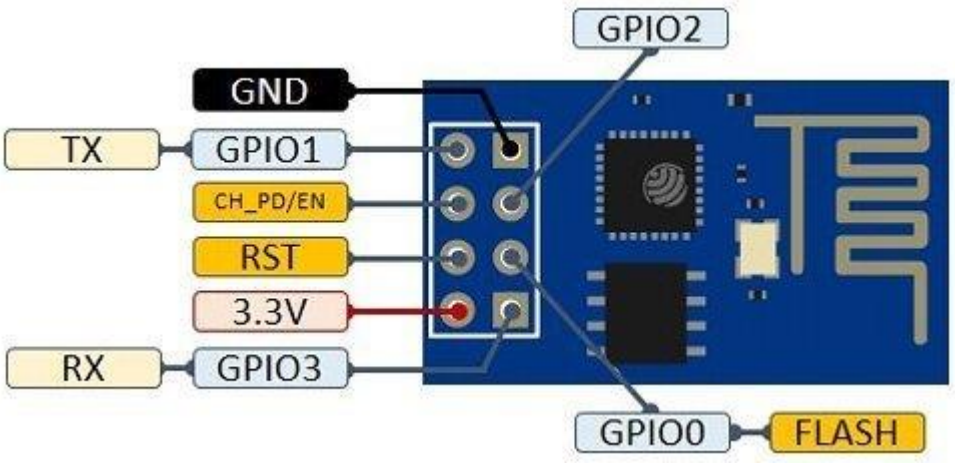


Figura 27. Pines ESP-01

Nota: Fuente https://acortar.link/4aaEcQ página web Educativa Robots Didácticos.

A continuación, se hará la explicación del diagrama de flujo que se utilizó para programar el módulo ESP8266-01 del sensor de apertura para la puerta y la ventana el cual se puede observar en la Figura 28 y Figura 29, Como se puede observar los diagramas de flujo son muy parecidos ya que ambos programas usan el mismo código, la única diferencia entre ambos

códigos son el tema donde se está publicando los datos, el ID y las credenciales para conectarse al bróker.

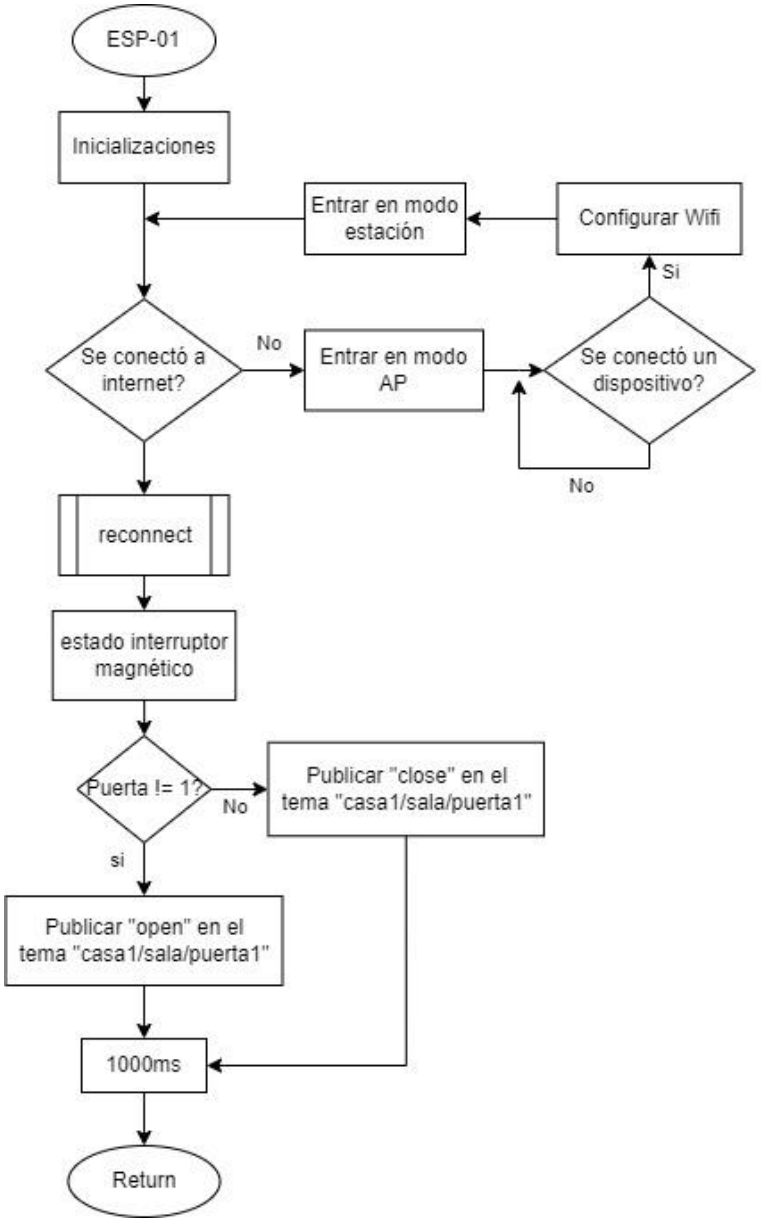


Figura 28. Diagrama de flujo ESP-01 Puerta

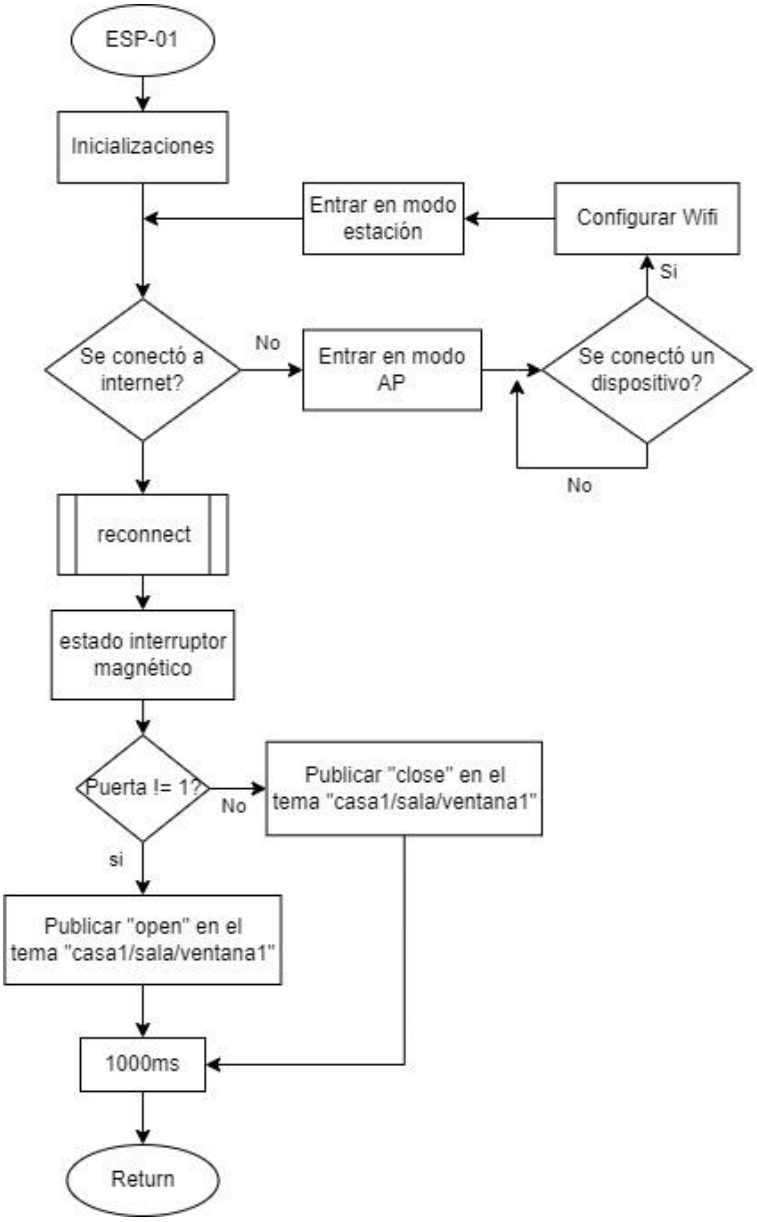


Figura 29. Diagrama de flujo ESP-01 Ventana

En el programa del ESP-01 primero se inicializan las variables y luego se realiza la conexión a internet del ESP-01, donde para realizar la conexión a la red Wifi se debe de disponer del SSID y Password de la red Wifi. Se le importó una librería llamada WiFiManager.h la cual

permite realizar la conexión a la red Wifi, y en caso de no conectarse cambia el modo estación del ESP-01 a modo AP, donde se crea una red Wifi, en la cual conectándose desde cualquier dispositivo con Wifi se configura la SSID y la Password de la red a la cual se desea conectar el ESP-01, una vez se hace esta configuración, el ESP-01 se cambia de nuevo a modo estación y se intenta conectar nuevamente a la red.

Una vez conectado a la red Wifi el ESP llama la función reconnect, la cual lo conecta al bróker MQTT, luego el módulo mide el estado del pin digital donde se encuentra conectado el interruptor magnético, y publica en el tema “casa1/sala/puerta1” un “close” si la puerta se encuentra cerrada y un “open” cuándo se encuentra abierta, esta publicación de los datos la realiza cada segundo.

7.6 Raspberry Pi Zero W

La Raspberry Pi Zero W es un ordenador de placa reducida el cual cuenta bluetooth y Wifi al igual que la Raspberry pi 3 y 4, la diferencia es el tamaño reducido de la Raspberry Pi Zero W con respecto a las otras Raspberry Pi como se puede observar en la Figura 30. Esta Raspberry es ideal para aplicaciones que no necesiten procesamiento robusto ya que sus características de hardware son reducidas en comparación a otras Raspberry como se puede observar en la Tabla 4, sin embargo, es muy dinámica al tener ese tamaño ya que ocupa poco espacio, es de bajo consumo y bajo costo.

La Raspberry pi cuenta con un software llamado Raspberry Pi OS el cual está basado en la distribución de Linux, sin embargo, se le pueden instalar otros sistemas operativos como Ubuntu, Windows 10 IoT Cores, Kali Linux entre otros.

Figura 30. Raspberry Pi Zero W vs Raspberry Pi 3

Nota: Fuente https://acortar.link/SodxZf Pagina web Tienda ARROW.

La función que cumple la Raspberry pi Zero W en el proyecto es de funcionar como bróker dentro de la red local en la que se encuentra conectados los demás dispositivos, para que la Raspberry funcione como bróker se debe descargar e instalar los paquetes de un bróker MQTT, en nuestro caso decidimos usar Mosquitto el cual es muy ligero, lo que lo hace perfecto para la Raspberry. Mosquitto se debe de configurar modificando o agregando el fichero de mosquitto.conf, el cual contiene información como la dirección del archivo con usuarios y contraseñas, el puerto en el que va a escuchar el bróker, entre otras configuraciones. Mosquitto es escalable por lo que se pueden agregar muchos más sensores y dispositivos si así se desea,

también se puede correr en segundo plano, lo que le permite a la Raspberry realizar otras acciones mientras el bróker sigue funcionando.

La función del bróker es de recibir los datos que se envían a un tema por un suscriptor y enviarlos a los dispositivos suscritos a ese tema, como se puede observar en la Figura 31, para que un dispositivo se suscriba al bróker debe contar con un usuario y contraseña valido dentro de los archivos de configuración del bróker, una vez se suscribe un dispositivo puede recibir y enviar datos a los temas que tengan permiso hacerlo.

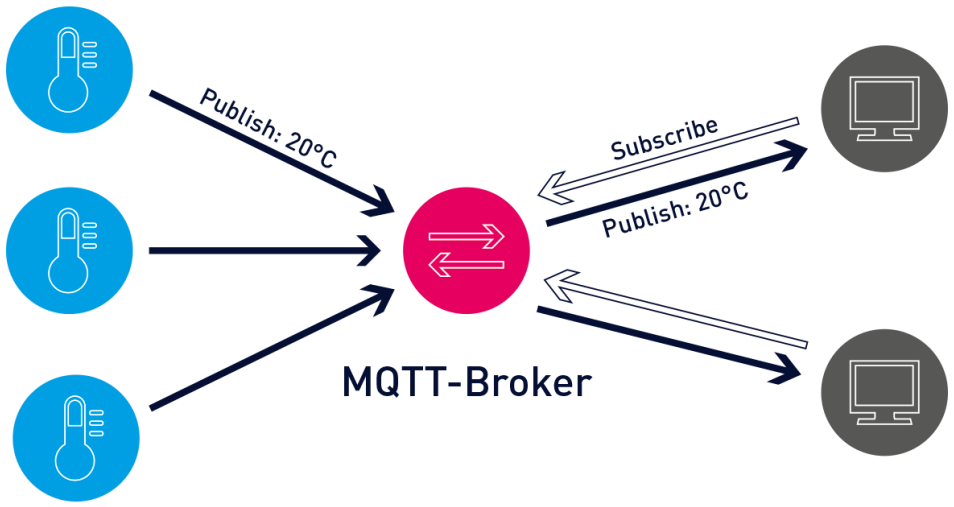


Figura 31. MQTT

Nota: Fuente https://www.paessler.com/es/it-explained/mqtt página web Empresa PAESSLER.

Cada módulo Wifi se suscribe al bróker usando un usuario y contraseña específico para cada uno, el cual solo les permite enviar datos al tema que está relacionado con el sensor que está midiendo los datos, por ejemplo, el sensor de apertura de puerta solo puede enviar datos en el tema “casa/sala/puerta” si intenta enviar algún dato por ejemplo al tema “casa/sala/ventana” el

bróker va a rechazar los datos enviados por el sensor. MQTT cuenta con algo que se le denomina calidad del servicio o QoS, esta característica indica la calidad de comunicación entre el bróker y los suscriptores, esta calidad de servicio va desde QoS 0 a QoS 2, donde:

- **QoS 0:** es una comunicación en la cual no se garantiza si los datos se reciben, lo que puede resultar en perdida de datos, sin embargo, no supone un problema en redes estables.
- **QoS 1:** se asegura de que los datos se reciban al menos una vez, esta calidad de servicio garantiza que el dato se reciba, pero puede haber datos duplicados.
- **QoS 2:** el bróker se asegura de recibir los datos solo una vez lo que garantiza que no haya datos perdidos ni duplicados, lo que lo hace ideal en redes inestables, sin embargo, este tipo de QoS genera mucho más tráfico y latencia en la comunicación.

Por limitaciones en las librerías MQTT de los módulos Wifi ESP8266 solo se permite hacer uso de la QoS 0, lo que realmente no supone ningún problema ya que no es indispensable garantizar la llegada de todos los datos, debido a que los sensores envían datos continua y constantemente.

Cabe destacar que los datos enviados y recibidos por el bróker están cifrados y las contraseñas en su archivo de usuarios también están cifradas, lo que le brinda seguridad a la comunicación entre los dispositivos, y protege el sistema de algún intruso no deseado.

7.7 Firebase

El primer paso es crear una cuenta en Firebase para esto se debe usar una cuenta de Google. Una vez se entra con ella al portal, se inicia un proyecto presionando en el botón agregar proyecto, desplegándose una pantalla donde se pedirá el nombre que se le dará al proyecto como se muestra en la Figura 32. Con este nombre se generarán las credenciales que se utilizarán para poder conectarse, leer, escribir, y utilizar cada uno de los servicios que ofrece esta plataforma en la nube.

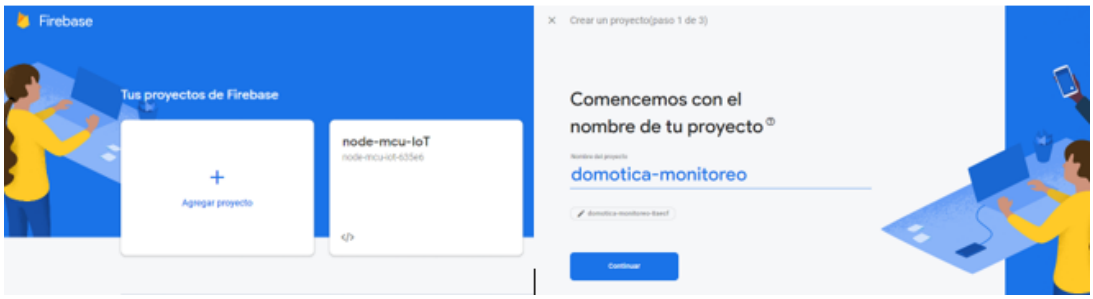


Figura 32. Creación de un proyecto en Firebase

Para este proyecto se usaron solo cuatro de los servicios, Firestore Realtime, Firestore Database, Cloud Storage, y Cloud Messaging, donde cada uno de ellos cumple una función importante el desarrollo del proyecto.

7.8 Desarrollo de la Aplicación

A continuación, se explicarán los pasos que se realizaron para el desarrollo de la aplicación.

7.8.1 Instalación de Programas y Dependencias

Para la construcción de la aplicación para el monitoreo de variables se usaron varias tecnologías, para crear la interfaz de usuario se usó ReactJs que es una biblioteca JavaScript, para el backend de la aplicación se usó Python y los servicios de Firebase.

Se inicia el proyecto instalado NodeJs, para esto hay que dirigirse a la página oficial nodejs.org donde se mostrara una pantalla de descarga como muestra la Figura 33.

Figura 33. Página de Inicio de NodeJs

Nota: Fuente https://nodejs.org/es/ Descarga NodeJs

Se selecciona la versión recomendada que es la versión más estable, y se descargará un ejecutable, se abre el archivo y se ejecuta hasta que finalice la instalación; Para comprobar que se instaló correctamente en la consola de la maquina se ejecuta el comando ‘node -v’ y se devolverá la versión instalada: para este proyecto se usó la versión 16.13.0 (ver Figura 34).

Figura 34. Verificación de instalación NodeJs

Una vez comprobada la instalación, se elige una carpeta donde estará el proyecto. Para moverse entre carpetas se usa el comando ‘cd’ en la consola, seguido del nombre de la carpeta “cd domotica-monitoreo”, una vez ubicados en la carpeta se procede a iniciar un proyecto de React con el comando npx create-react-app seguido del nombre del proyecto, como se muestra en la Figura 35, y luego se le pasará el parámetro ‘–template cra-template-pwa’ para crear una plantilla de React con las dependencias necesarias para convertir la aplicación en una PWA (Aplicación Web Progresiva), que permitirá tener una aplicación para distintas plataformas siempre y cuando estas cuenten con un navegador web.

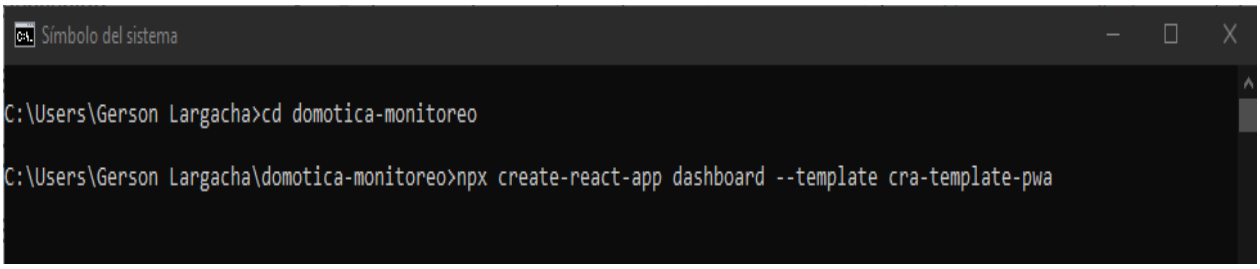


Figura 35. Creación de un proyecto en ReactJs

Con el comando de la Figura anterior se empezarán a descargar todos los paquetes para que un proyecto en React funcione, al terminar la descarga y configuración se procede a abrir un editor de código, para el proyecto se usó Visual Studio Code. Al abrir el proyecto aparecerán unas carpetas y archivos como se puede ver en la Figura 36: la carpeta node_modules contiene las dependencias y librerías del proyecto, en la carpeta public se encuentra un archivo index.html que es la plantilla de la página web, iconos y un archivo JSON que vienen por defecto al crear un nuevo proyecto.

En la carpeta src se encuentran archivos JavaScript, los archivos reportWebVitals, service-worker, serviceWorkerRegistration dentro contienen las configuraciones y demás instrucciones para que la aplicación pueda funcionar como una PWA, también archivos con extensión .css que contienen los estilos iniciales del proyecto, el archivo '.gitignore' es un archivo con el que se omiten carpetas o archivos que no se quieren subir a un repositorio.

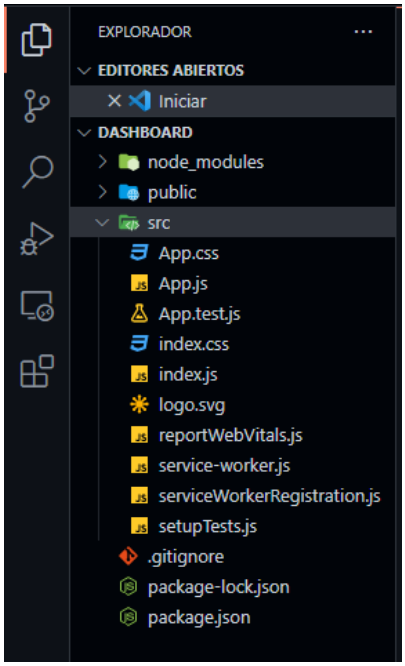


Figura 36. Capetas y archivos iniciales del proyecto

Los archivos package.json y package-lock.json como se ven en la Figura 37, son archivos que al igual que los anteriores se generan automáticamente y su función es mantener un registro de los paquetes instalados en el proyecto y que se necesitan para funcionar. Por ultimo los archivos postcss.config.js y tailwind.config.js que encontramos son dependencias que se instalaron para dar estilo a la aplicación.

Figura 37. Archivos de configuración

NodeJs posee una serie de comandos que automatizan tareas, pero para esto es esencial explicar que es ‘npm’; Node Package Manager (NPM) es la herramienta que usa NodeJs para instalar paquetes o distribuirlos, además es una herramienta para el manejo del terminal o consola. En el desarrollo de la interfaz de usuario se hizo uso de diferentes librerías, estas se instalaron usando el comando ‘npm install <nombre de paquete que se desea instalar>’ que descargara los archivos desde los repositorios de NodeJs.

Para los estilos se usó la librería TailwinCSS como se mencionó anteriormente. También se instaló la librería de Firebase para el uso de los servicios, almacenamiento, y manipulación de los datos, React-Router que proporciona ayuda con el manejo de las rutas, MomentJs un paquete para el manejo de fechas y horas, React-Chartjs-2 para la visualización de datos gráficos estadísticos, y React Toastify para la vista de las notificaciones dentro de la aplicación entre otras como se muestra en la Figura 38, estas librerías instaladas en el proyecto se encuentran en el archivo package.json.

Otro comando importante que vale la pena recalcar es ‘npm run start’, este comando le indica a NodeJ, que corra el comando start, que ejecuta un servidor local que muestra nuestra aplicación, una vez ejecutado el comando, se puede realizar cambios en el código y una vez guardado el archivo estos cambios se aplicarán automáticamente.

Se plantearon los bocetos de las vistas y pestañas de la aplicación para empezar a diseñar la interfaz, así tener claro como el usuario interactuara con cada aparatado de la aplicación.

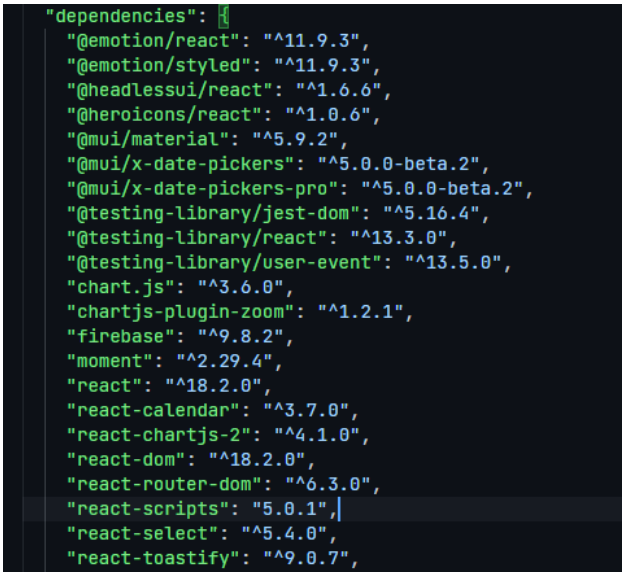


Figura 38. Dependencias de la aplicación

7.8.2 Interfaz Gráfica

La interfaz de la aplicación como ya se explicó anteriormente se desarrolló con ReactJs, y se diseñó una aplicación responsive para que el formato de la página se adaptara a los diferentes dispositivos como se muestra en la Figura 39, donde se presenta cómo se vería la aplicación en un computador o Tablet con un formato más horizontal, y como se vería en un dispositivo móvil en formato vertical.

La interfaz gráfica cuenta con cuatro pestañas que cada una de estas tiene propósitos específicos, para facilitarle al usuario el control y manejo de esta.

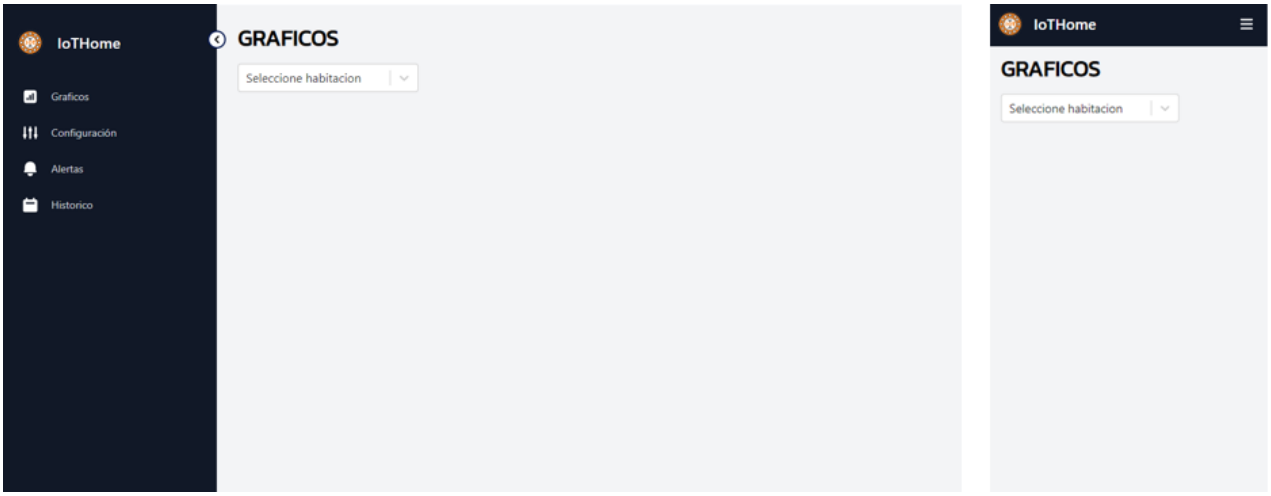


Figura 39. Diseño responsive de la interfaz

7.8.2.1 Pestaña Gráficos

Cuando se inicia la aplicación, se abrirá la primera pestaña que tiene por nombre gráficos como se muestra en la Figura 39, se encuentra un selector (Ver Figura 40) el cual permite seleccionar la habitación en la cual se desea consultar la medición de los sensores, estas mediciones son mostradas en diferentes gráficos, dependiendo del tipo de sensor, como se ve en la Figura 40, cada uno de estos son configurables en la pestaña de configuración de la aplicación, es pertinente resaltar que los datos mostrados hasta el momento son datos simulados.

Los datos que se ven en las gráficas son datos que se van actualizando cada vez que un sensor toma la medición, y estos son consultados en la nube por la aplicación, usando el servicio Firebase Realtime, lo que permite ver la actualización del dato en tiempo real.

Figura 40. Pestaña Gráficos

7.8.2.2 Pestaña Configuración

La siguiente pestaña corresponde a configuración, en el que hay dos secciones y cada sección cuenta con una tabla que da información sobre las habitaciones y sensores que se han añadido, desde ahí se puede agregar, editar y eliminar una habitación o dispositivo (ver Figura 41); esta pestaña cumple un rol muy importante ya que depende de esos datos que la aplicación se sincronice con los dispositivos, y reciba los datos.

Para configurar una habitación basta con presionar el botón ‘agregar habitación’ de la Figura 41 esto mostrará una ventana emergente que contiene un formulario (ver Figura 42) con solo dos campos: el primero para darle un nombre a la habitación, y el segundo es el tópico de la habitación; el tópico es la ruta en donde el sensor publicará los datos, como se observa en la figura anterior. También es importante mencionar que en la tabla habitaciones solo aparecen tres

columnas y solo se tienen dos campos en el formulario, esto es debido a que la columna número de dispositivos (# dispositivos) de la tabla, no es un valor de entrada, sino que es un campo que se actualiza solo cuando se ha agregado un dispositivo en esa habitación.

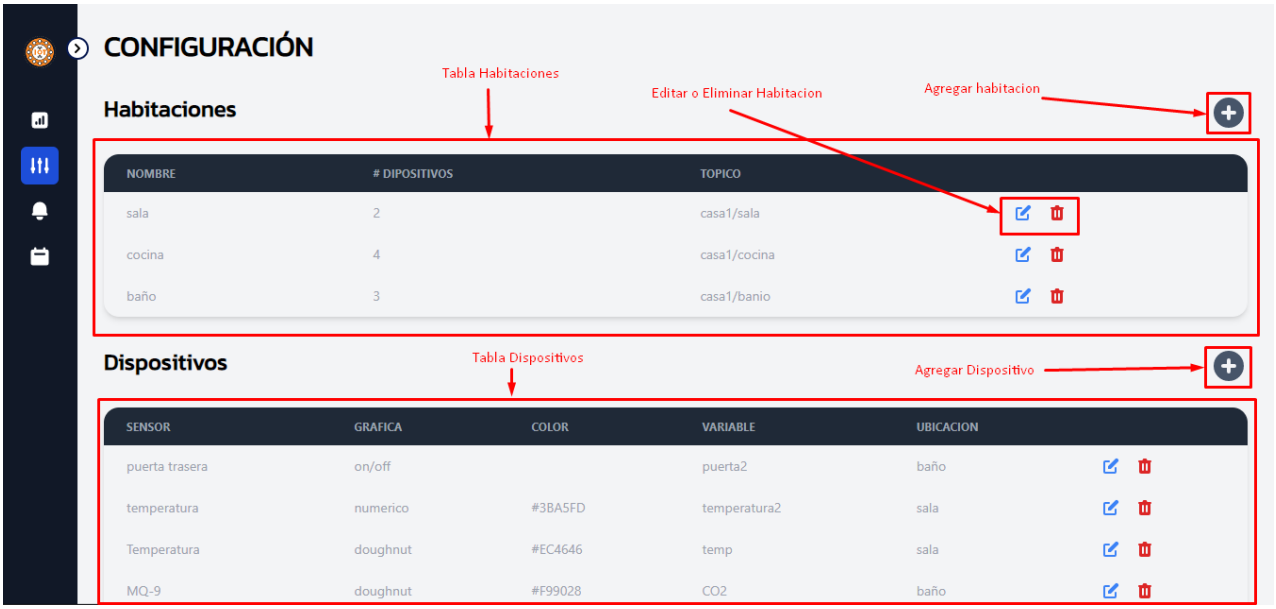


Figura 41. Pestaña de configuración

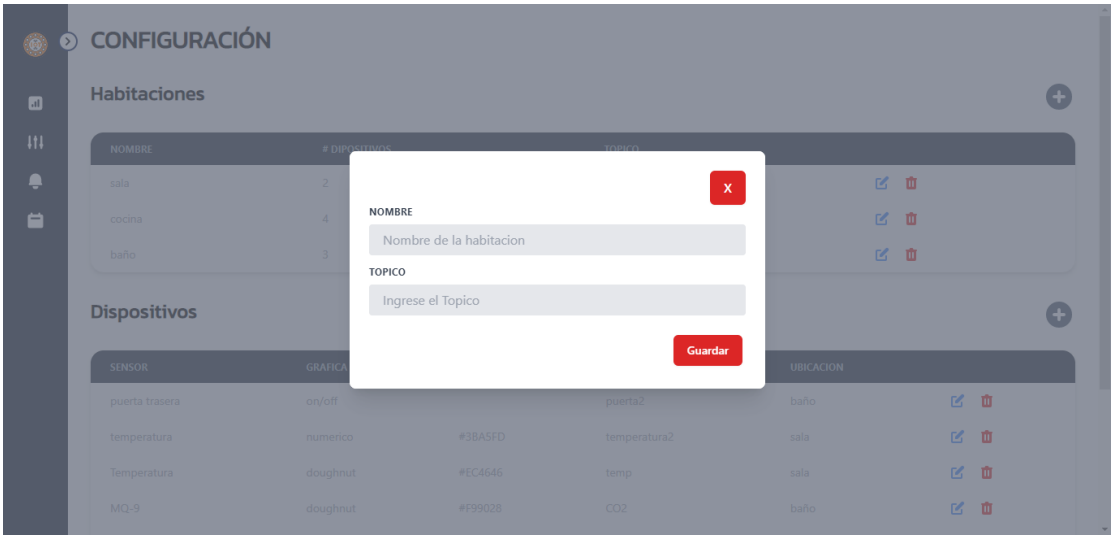


Figura 42. Formulario habitaciones.

En la misma sección de configuraciones al presionar agregar dispositivos (ver Figura 43) saldrá una ventana emergente parecida a la anterior, pero con más campos, para la configuración del dispositivo que se va a agregar.

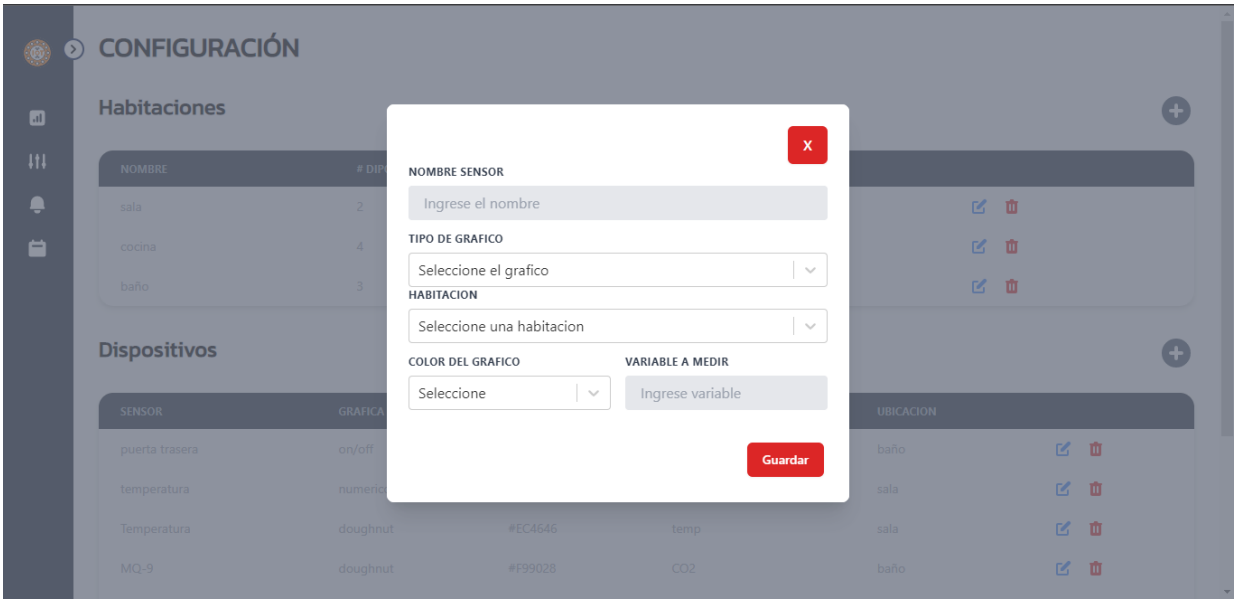


Figura 43. Formulario de Dispositivos

En el primer campo nombre del sensor, se puede poner un identificador del sensor, al presionar el campo tipo de gráfico, se despliega una lista que contienen los gráficos permitidos para mostrar en la aplicación, el usuario puede elegir entre cuatro diferentes opciones:

- Gráfico de tipo línea: estos gráficos representan un valor continuo en el tiempo de la medición que se está realizando (ver Figura 40).

- Gráfico de tipo dona: este tipo representan el ultimo valor leído por el sensor, donde la sección coloreada representa la medición del sensor, entre más color tenga, representa una medición mayor en el dispositivo (ver Figura 40).
- Gráfico de tipo numérico: este tipo solo representa un valor numérico con su correspondiente unidad de medida.
- Gráfico de tipo on/off: esta opción es diferente a las demás ya que al seleccionarla el campo llamado color gráfico cambiará a tipo de apertura donde se pueden seleccionar dos opciones, puerta o ventana.

El tercer campo permitirá seleccionar en que zona va a estar ubicado el dispositivo: con el cuarto campo se selecciona el color del gráfico. Esta propiedad solo aplica para gráficos de línea y gráficos tipo dona, y por último el campo variable a medir. Se debe tener en cuenta que este valor es complementario al tópico de la habitación, entonces si se desea medir temperatura y el sensor está ubicado en la cocina, el tópico de la habitación seria ‘casa/cocina/’ + el nombre de la variable; el resultado final sería ‘casa/cocina/temperatura’, para tener en cuenta, no es necesario incluir el ‘/’ al inicio de la variable a medir (ver Figura 43) o al final del tópico cuando se está creando la habitación.

En las configuraciones como antes se mencionó también se puede editar y eliminar tanto habitaciones, como dispositivos, para esto en la tabla de la Figura 41 se pueden observar dos iconos, uno de color azul con forma de lápiz sobre una hoja y otro color rojo con forma de papelera basurera. Si se presiona el icono azul entonces se abrirá el mismo formulario, con el que

se puede editar los campos de ese registro, al presionar el icono de papelera este eliminará el registro, pero antes preguntará si se desea eliminar el ítem. Cada ítem de la sección de habitaciones y dispositivos es almacenado en la base de datos de Firebase, Firebase Database de donde se consultan los datos para las otras pestañas de la interfaz.

7.8.2.3 Pestaña Alertas

La pestaña con icono de campana es la sección de alertas, en esta se encuentra un botón del pánico y un histórico de notificaciones. Al presionar el botón de emergencia este cambiará el estado de una variable dentro de Firebase RealTime, lo que generará un envío de un correo electrónico a los correos que estén agregados en la base de datos. Esta sección posee también un histórico de las notificaciones donde el usuario puede estar informado en caso de no haber visto esta alarma. Las alarmas también funcionan si se está fuera de la aplicación siempre y cuando se le da permiso de envío de notificaciones en el navegador. Cada mensaje del histórico puede ser eliminado presionando el botón X al final de cada mensaje como se muestra en la Figura 44.

Cada notificación o mensaje tiene un color, pues esto se debe a que hay diferentes niveles de notificaciones, el azul es el nivel más bajo, es un nivel tipo informativo (info), solo presenta datos no relevantes como, si se añadió un sensor o se modificó, el color amarillo representa un mensaje tipo advertencia (warn), es una notificación de segundo nivel, esto indica que algo está pasando y que se debería prestar atención. Un ejemplo de esto puede ser que la temperatura ha aumentado considerablemente o pasado el umbral, finalmente tenemos una notificación de tercer

nivel, color rojo, esta es una notificación critica (critical), indica que algo salió mal, puede ser que se ha detectado un incendio, o un paso de umbral de todos los dispositivos de medición.

Figura 44. Pestaña Alarmas y Notificaciones

Al lado derecho del botón de pánico como se muestra en la Figura 44 está ubicado un botón blanco con un lápiz azul este botón abrirá una ventana emergente donde se puede agregar los correos electrónicos a los que se enviará una alerta al momento en que se presiona el botón de emergencia.

En la ventana emergente de la Figura 45 se encuentra un campo con el nombre correo electrónico de emergencia. Aquí se pueden añadir los correos que sean deseen, solo es necesario darle agregar para que este correo quede registrado, en la parte central está ubicada una caja de texto más grande donde el usuario puede personalizar el mensaje de alerta, por último, en la parte

inferior de la ventana encontramos los correos electrónicos que han sido agregados previamente, estos pueden ser eliminados presionan la X que se encuentra en la parte derecha de cada correo.

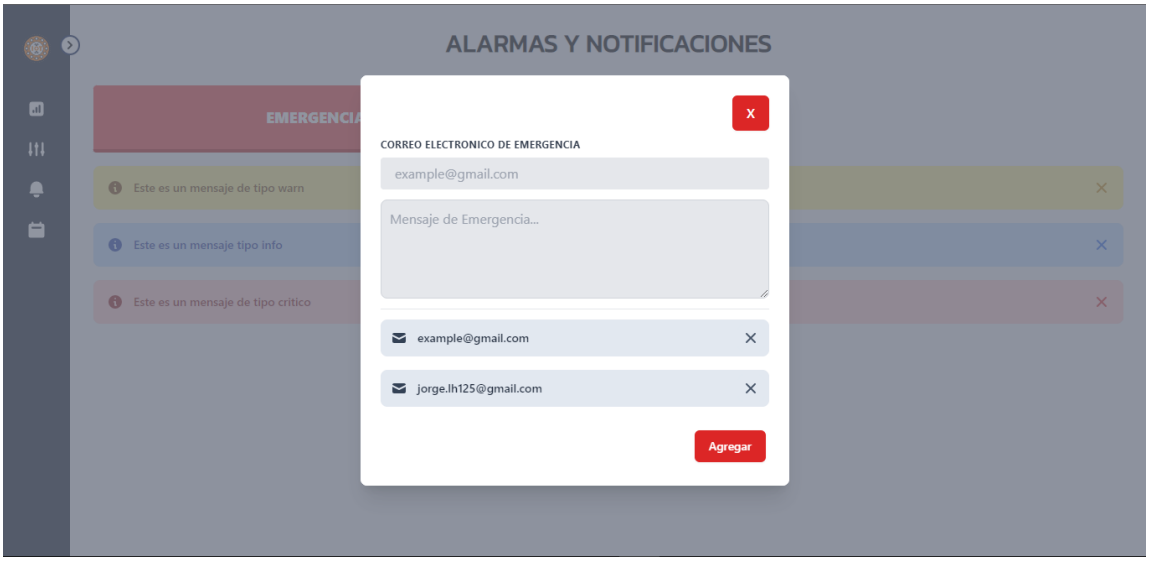


Figura 45. Ventana emergente e-mails de emergencia

7.8.2.4 Pestaña Histórico

La última pestaña representada con un calendario es la que tiene por nombre histórico, esta sección tiene un panel principal que posee controles para hacer la búsqueda de los datos y el panel secundario donde se encontrará el resultado de la búsqueda.

En la primera sección se encuentran dos selectores en los que se puede elegir la fecha inicial y la fecha final de la búsqueda, luego se encuentra otro selector que tiene la etiqueta sensor, aquí se puede elegir el sensor del cual queremos visualizar su histórico, al presionar el botón buscar los resultados aparecen en una segunda sección donde se muestran dos tipos de

gráficos el de línea como en muestra en la Figura 46, o un gráfico de barras solo si el sensor es de tipo on/off.

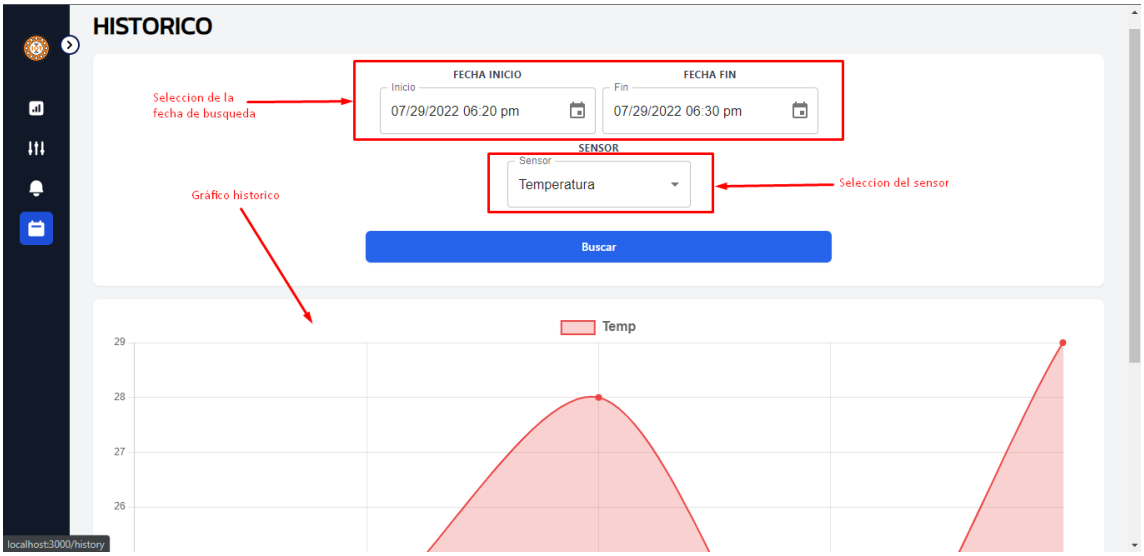


Figura 46. Pestaña Histórico

7.9 Backend con Python

Una vez hecha toda la interfaz gráfica se elaboró un programa que recogiera todos los datos enviados por los sensores al bróker, subirlos y sincronizarlos en Firebase para que la aplicación pudiese consultar estos datos. Para realizar dicha tarea se hizo uso de Python que es un lenguaje flexible y multiplataforma.

7.9.1 Instalación de Python

El primer paso fue usar el comando ‘sudo apt update’ en la Raspberry Pi. El comando ‘sudo’ permitirá ejecutar comandos como administrador, el comando ‘apt’ es el instalador de

paquetes del sistema y al pasarle el parámetro ‘update’ se le dice al sistema que se quiere actualizar la lista de repositorios del sistema, esto se hace para evitar tener algún conflicto con las versiones.

El siguiente paso es instalar Python con el comando “sudo apt install python3” usualmente este viene preinstalado en las distribuciones Linux, pero en caso de no estarlo se usa el comando anterior. Luego es necesario se digita el comando “sudo apt-get install python3-pip” para obtener el instalador de paquetes de Python, que pesa aproximadamente 245MB, y se comprueba con el comando ‘python3 --version’ con lo que se obtiene la versión que se tiene instalada, como se muestra en la Figura 47.

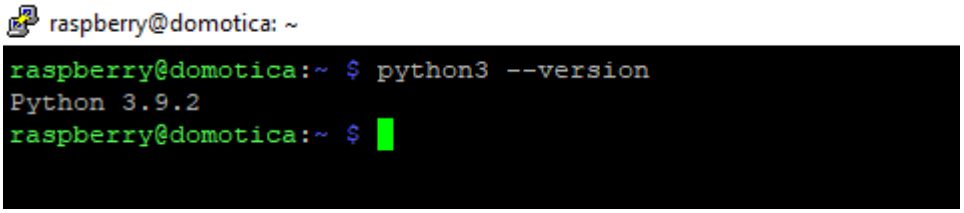


Figura 47. Verificación de instalación de Python

Antes de empezar a escribir el código, se instalan librerías necesarias para que el programa pueda conectarse al bróker mqtt mosquitto, y a Firebase. La librería usada para conectarse al bróker fue paho-mqtt y firebase-admin, y se obtienen con el comando “pip install paho-mqtt firebase-admin” como se ve en la Figura 48.

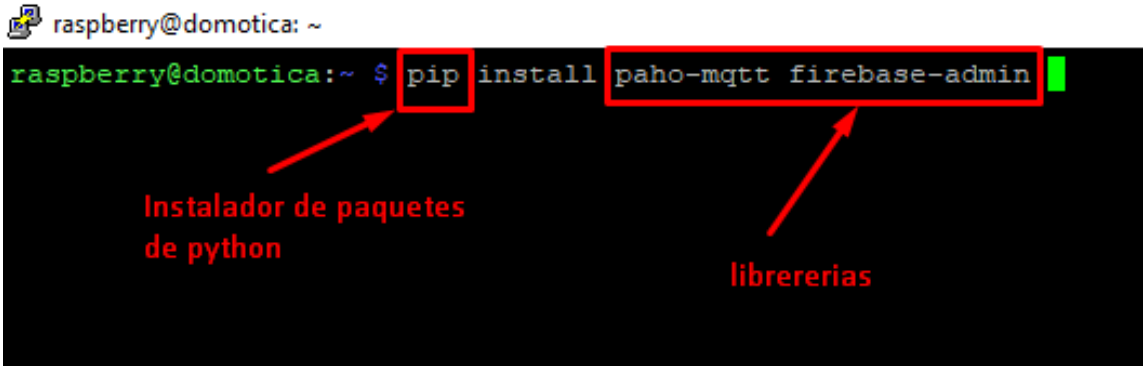


Figura 48. Instalación de librerías de Python

El siguiente paso es crear un archivo de Python, para lo cual se utilizó por comodidad Visual Studio Code en un ordenador Windows y luego se descargaban los archivos con el programa WinSCP que es un cliente que usa SSH para enviar los archivos.

7.9.2 Lógica del programa

A continuación, se explicará la lógica del programa, el cual cuenta de dos hilos como se puede observar en la Figura 49, y que puede observarse completa en el Anexo 1.

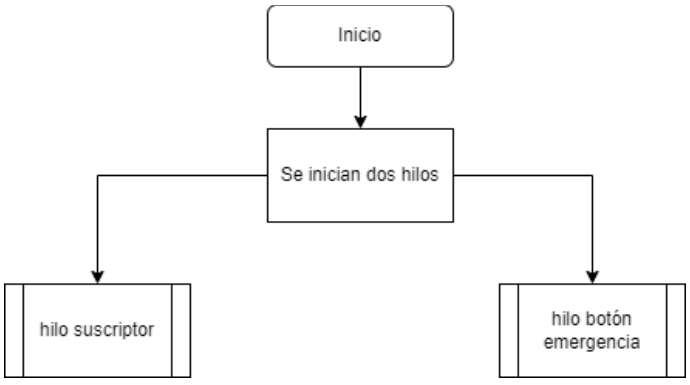


Figura 49. Diagrama de flujo creación Inicio del programa

En esta sección se explicarán los hilos por separado: un hilo para leer los datos del bróker y otro para enviar una alerta en caso de que se presione el botón de emergencia que se encuentra en la aplicación en la pestaña de alertas y notificaciones (ver Figura 44), para esto se usó la librería threading de Python, que permite ejecutar operaciones simultaneas en un mismo proceso.

En la Figura 50 cuando se inicia el hilo con el nombre suscriptor, lo que hace primero es conectarse al bróker MQTT, para esto hay que añadir algunas credenciales como lo son la dirección del bróker y el puerto por donde se está escuchando, al conectarse al bróker este queda a la espera de un mensaje, si hay un mensaje entonces pasamos a hacer validaciones, se valida si el dato está por debajo del umbral, en caso de que este dato sea mayor entonces se envía una alerta a la aplicación haciendo uso de la librería firebase-admin, por medio del servicio de Firebase, Cloud Messaging, y registrando esta alerta en la base de datos, para poder verla en caso de haberla omitido. En el caso contrario solo se guardará el dato, y volverá a esperar que haya una publicación por parte de los sensores.

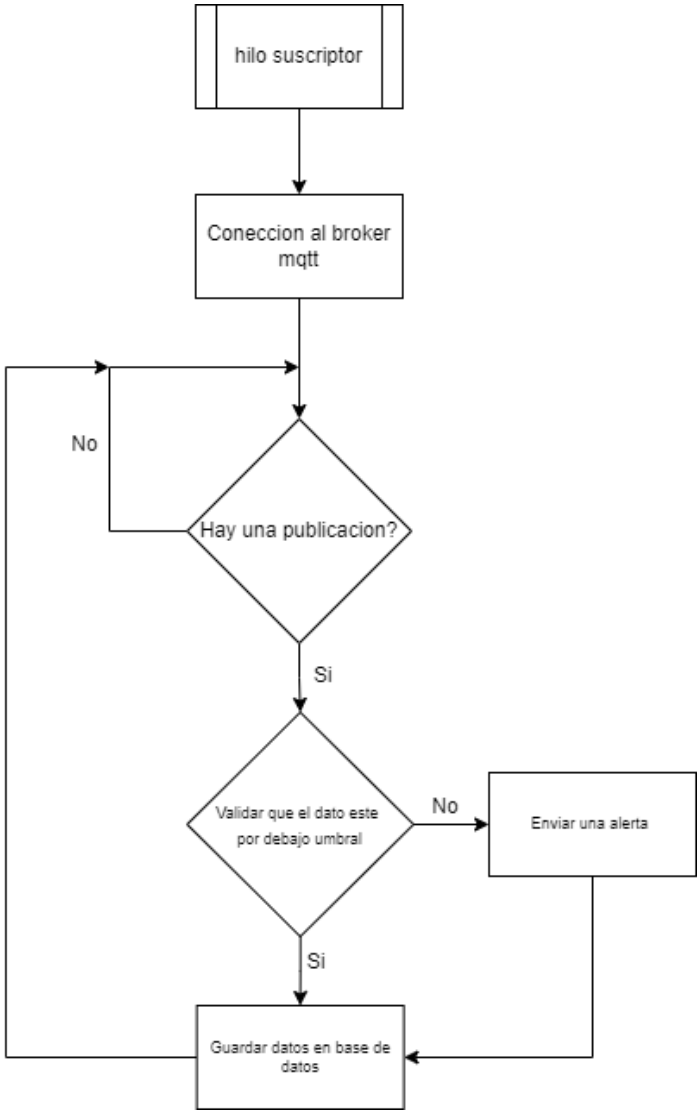


Figura 50. Hilo suscriptor

El hilo con el nombre botón emergencia de la Figura 51, escucha una variable en el servicio de Firebase RealTime que tiene como nombre “emergencia”, al presionar el botón de la aplicación en la Figura 44 este modificara la variable emergencia con el valor ‘yes’ entonces, este

valor será leído en este programa de Python y enviara un mensaje a los correos electrónico que fueron agregados por el usuario, avisándoles sobre el evento sucedido.

Para enviar este correo electrónico se usó la librería smtplib que viene por defecto en Python. Una vez el correo se envía entonces el estado de la variable emergencia vuelve a cambiar a ‘no’, y luego se seguirá consultando hasta que esa variable vuelta a cambiar su estado.

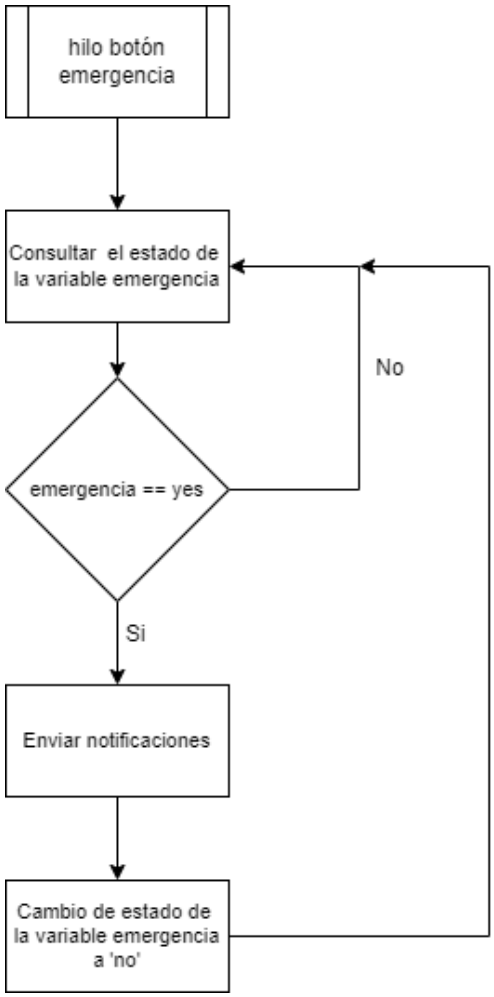


Figura 51. Hilo botón de emergencia

7.9.3 Creación de un servicio en Linux

Una vez terminado el script, para poder que este se ejecutará siempre y no tener que ingresar a la Raspberry, se crea un servicio Linux, un servicio es un programa que se puede ejecutar en segundo plano.

En la Raspberry se procede a ir la carpeta donde se encuentran los servicios del sistema, para esto se usa el comando “cd /lib/systemd/system”, al presionar enter se estará dentro de la carpeta. Con el comando ‘ls’ se devolverá una lista de archivos con la extensión ‘service’ con el que se verificará que se está dentro de la ruta deseada, con el comando ‘nano’ se crea un nuevo archivo con extensión ‘.service’ para este caso se le llamó postman como se ve en la Figura 52.

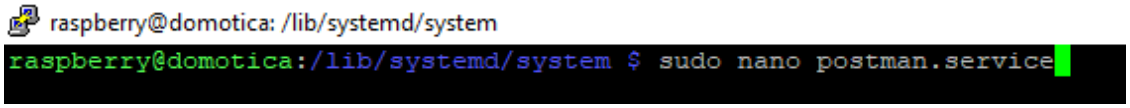


Figura 52. Creación de un servicio Linux

Lo anterior abrirá un archivo vacío, en el cual se deben escribir algunos parámetros para que el servicio funcione, el primer parámetro es una descripción, aquí se puede especificar el nombre o la tarea que se realizara, luego se especifica el valor after donde se le dice al servicio que se ejecute después de que algún servicio específico, en este caso después de que la red este activa como se muestra en la Figura 53, con StartLimitIntervalSec y StartLimitBurst, se configura el intervalo de comprobación y cuantos inicios por intervalos son permitidos, En la sección de service, se elige el tipo de servicio en este caso simple, con RestartSec se configurá el tiempo que debe esperar antes de reiniciar el servicio, con el Parámetro User se le dice al servicio

quien puede ejecutarlo, en este caso se añade el usuario raíz (root); se usa Enviroment para que busque las variables en la ruta definida, En el parámetro ExecStart se le indica la ruta exacta donde se encuentra el archivo o binario que se desea ejecutar junto con el comando, y parámetros a ejecutar en caso de tenerlos, para este caso solo le pasa el comando ‘python3’ más la ruta del archivo que se quiere ejecutar cuando se inicie el servicio como se presenta en la Figura 53, y por ultimo WantedBy para crear un enlace simbólico que apunta a un archivo o carpeta (Debian, 2022).

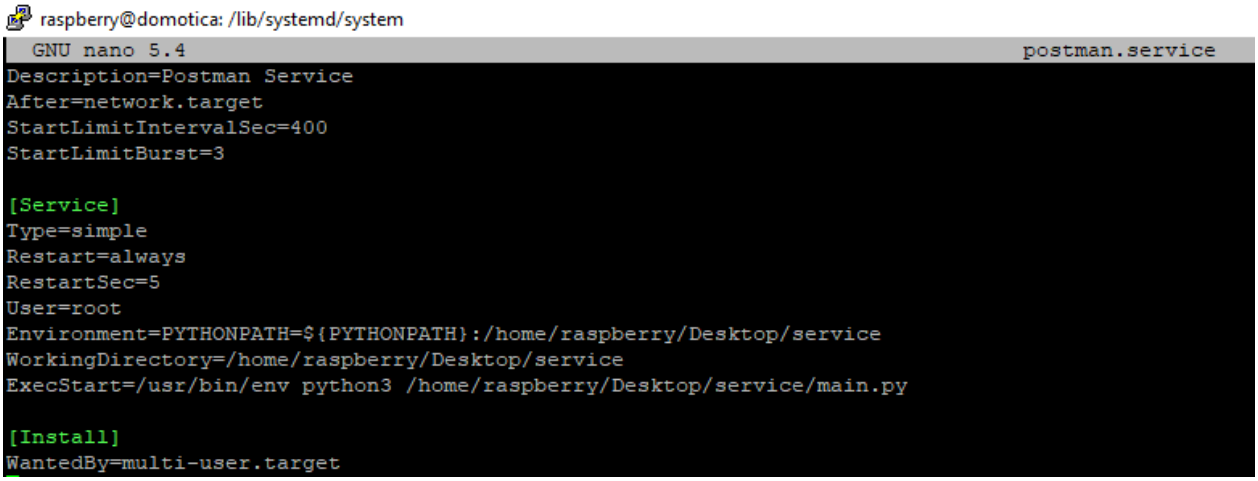


Figura 53. Configuración de un servicio Linux

Una vez se crea el archivo con la configuración del servicio, en la carpeta “/lib/systemd/system” se ejecuta el comando “systemctl enable postman.service” creándose un enlace simbólico como se evidencia en la Figura 54, que le permitirá al sistema buscar el archivo en la ubicación systemd donde se encuentran los archivos de arranque automático. En caso de no querer que el programa se inicie de forma automática cuando inicie el sistema, se usará el

comando “systemctl disable postman.service”. Luego se ejecuta el comando “systemctl daemon-reload” este comando volverá a cargar todos los archivos y dependencias necesarias, este comando es esencial para aplicar los cambios que se realizaron anteriormente.

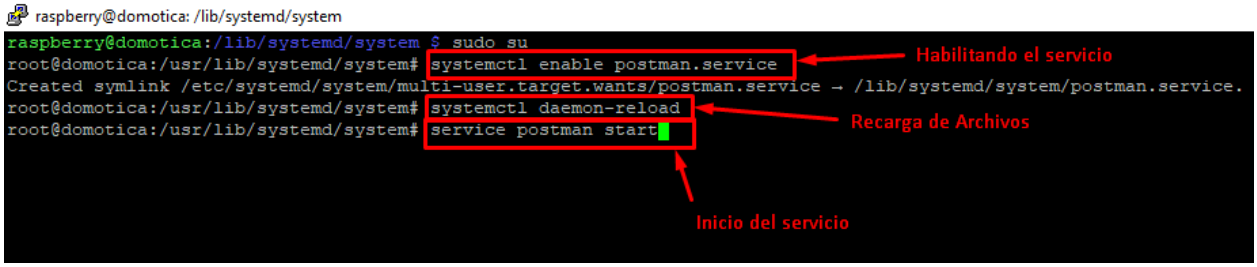


Figura 54. Habilitación y arranque del servicio

Después de cargar los cambios es necesario iniciar el servicio, para eso se usa el comando “service postman start", en la documentación de Debian se puede encontrar los comandos para la gestión de estos servicios, luego verificamos que todo esté funcionando, usando el comando “service postman status”, el resultado de ejecutar este comando se puede ver en la Figura 55 donde se muestra que el servicio se encuentra activo y corriendo.

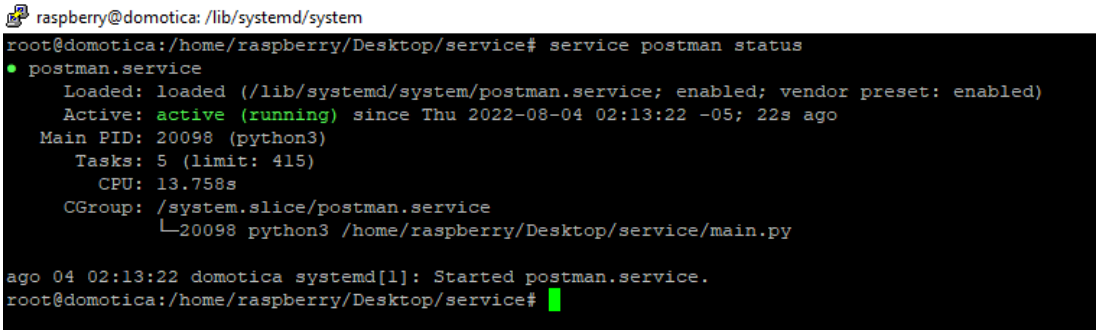


Figura 55. Estado del servicio

El uso de un servicio en Linux permite mantener el programa realizado en Python corriendo todo el tiempo en segundo plano, sin tener que entrar a la Raspberry y ejecutar este archivo, en caso de que la Raspberry Pi se apague o reinicie.

9. Resultados

En este capítulo se abordarán los resultados finales y PCB que se diseñaron para la conexión y alimentación de los sensores y los módulos Wifi, así como el funcionamiento del sistema de medición en conjunto con la aplicación.

9.1 Diseños y construcción de las PCB

A continuación, se presentarán los diseños que se realizaron para hacer la construcción de los circuitos de alimentación de los módulos Wifi y de los sensores.

9.1.1 PCB fuente ESP-01

El módulo ESP-01 funciona con un voltaje de 3.3V por lo que es necesario diseñar un circuito que se encargue de suministrar y regular el voltaje y corriente necesario para su funcionamiento.

En la Figura 56 se puede observar el circuito diseñado para la alimentación del ESP-01, en donde se usa el regulador ajustable LM317T, 2 condensadores que se encargan de estabilizar la corriente en la entrada y salida del regulador, unas resistencias y los diodos que se encargan de ajustar el voltaje que entrega el regulador de voltaje. Se decidió hacer uso del LM317T y con la ayuda de un diodo Zener 1N4371A y una resistencia de 330Ω se crea un divisor de voltaje que suministra un voltaje constante de 2.7V en el pin de ajuste del regulador, para que entregue un voltaje regulado de 3.3V cuando se alimenta el circuito con una fuente de 5V y que entrega un

voltaje de 4V cuando se alimenta con una fuente de 12V. Cuando se alimenta el circuito con la fuente de 12V se cambia el interruptor en la posición que conecta un diodo 1N4007 que reduce el voltaje a 3.3V ya que este diodo consume un voltaje de 0.7V. Como se puede observar se tiene conectado un bloque llamado J6 con una resistencia de 330Ω, este bloque es donde se conectan los cables del Interruptor magnético y el bloque J5 el cual son los pines en donde se conecta el ESP-01.

En la Figura 57 se puede observar el diseño del circuito PCB del circuito anterior.

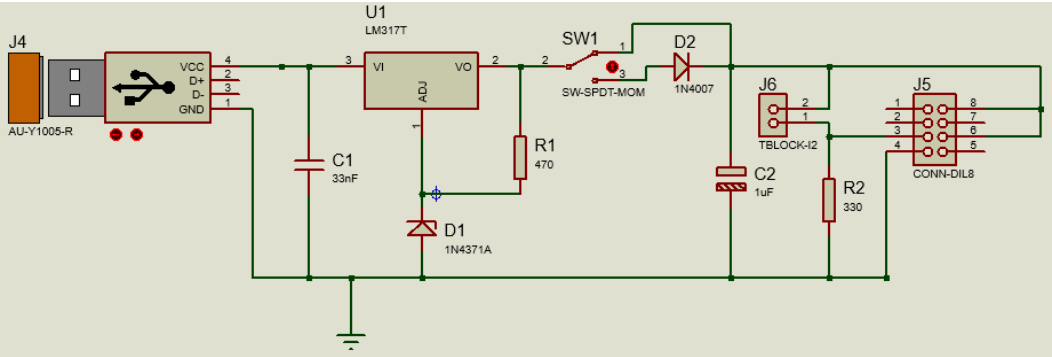


Figura 56. Circuito Fuente ESP-01

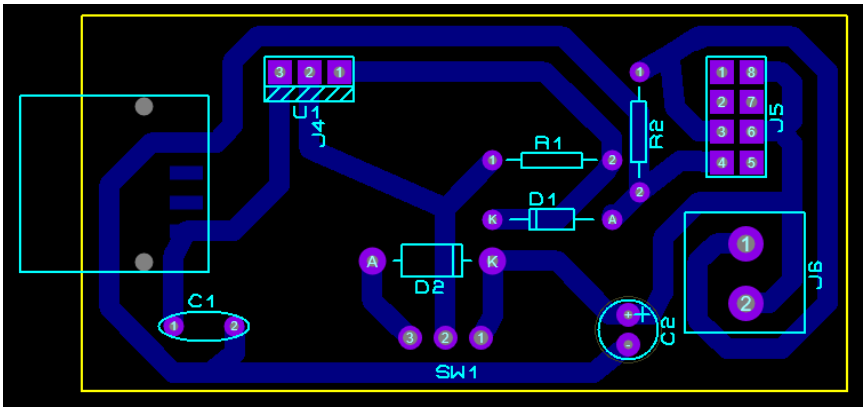


Figura 57. PCB Fuente ESP-01

9.1.2 PCB conexión entre el NodeMCU y los sensores

El NodeMCU tiene una ventaja frente al ESP-01 porque ya posee un regulador interno que le permite regular su voltaje, solo se necesita alimentar con una fuente que entregue entre 3.3 y 5V, es por esto por lo que se diseñó una PCB en donde se facilitará la conexión de los sensores con el NodeMCU el cual se puede observar en la Figura 58 y la Figura 59

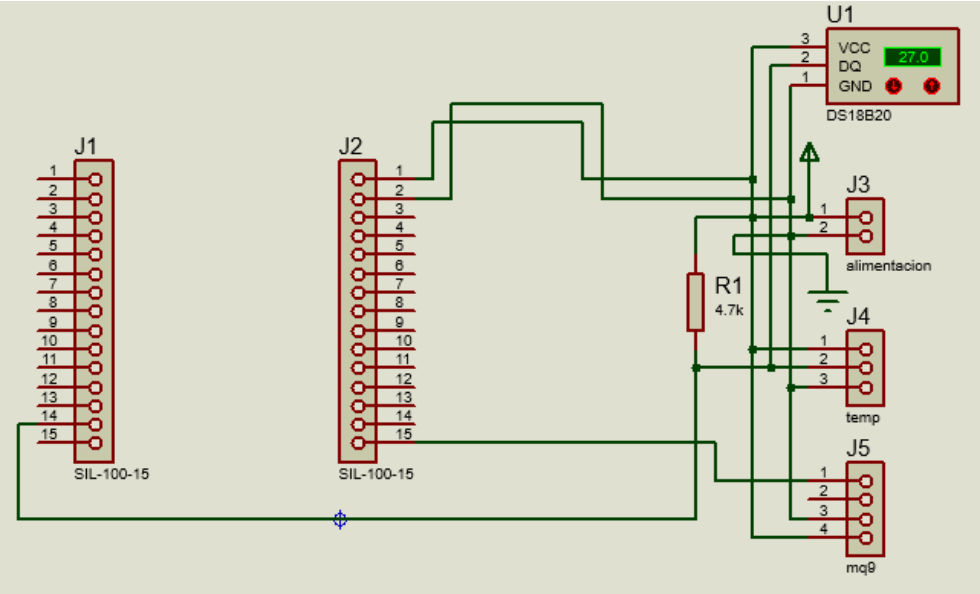


Figura 58. Circuito de conexión entre el NodeMCU y los sensores

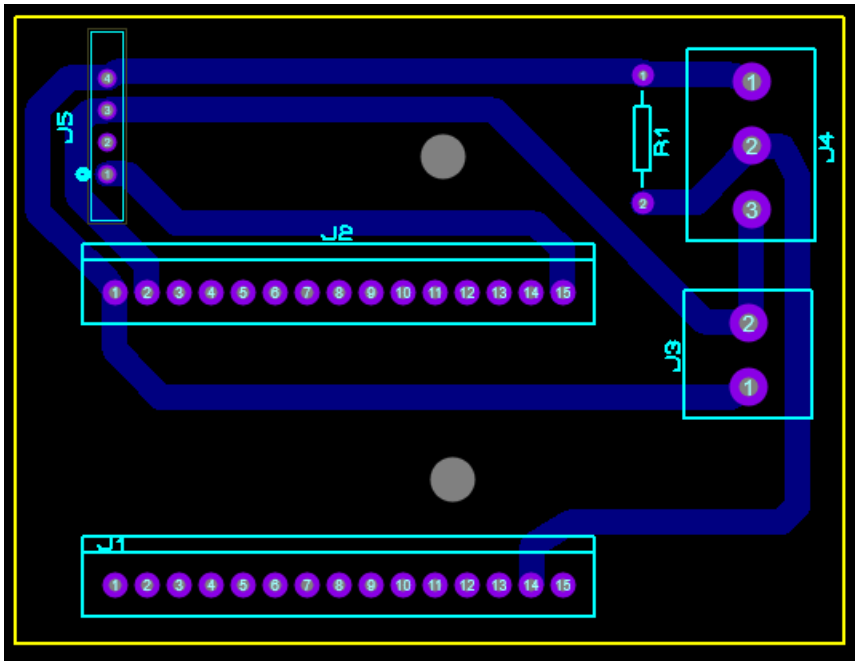


Figura 59. PCB del circuito de conexión entre el NodeMCU y los sensores

9.1.3 Construcción de los circuitos

Con los diseños PCB presentados en las figuras anteriores se hizo la construcción de los circuitos presentados en las Figura 60, Figura 61 y Figura 62, en los cuales se imprimieron, se grabaron los circuitos y se soldaron a mano cada componente. Haciendo uso de un multímetro se mide la corriente de consumo de los circuitos del ESP-01 en donde se midió que el consumo de corriente máxima era de 105mA a un voltaje de alimentación de 5V por lo que el consumo de potencia de los circuitos del ESP-01 es de 525mW, en cuanto al circuito del NodeMCU se mide la corriente de consumo la cual midió 200mA a un voltaje de 5V lo que se calcula una potencia de consumo de 1W, ya que se alimentan el sensor de temperatura, el sensor de gas y el NodeMCU, sumando el consumo de potencia de los tres circuitos da un consumo total de 1.5W.

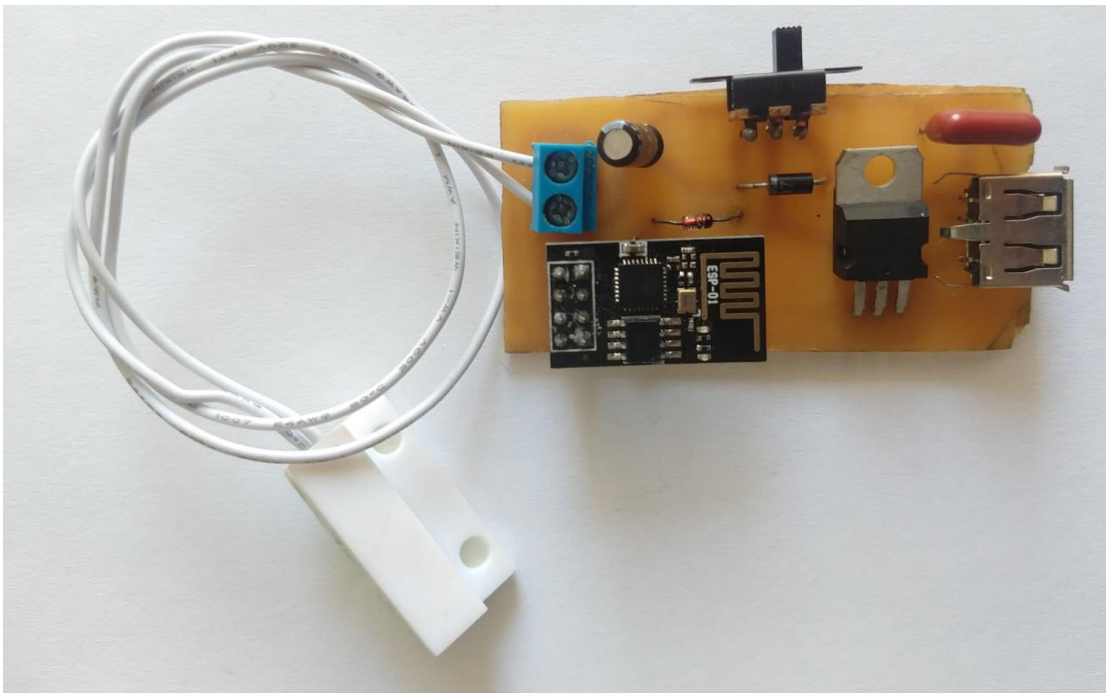


Figura 60. Circuito ESP-01 Puerta

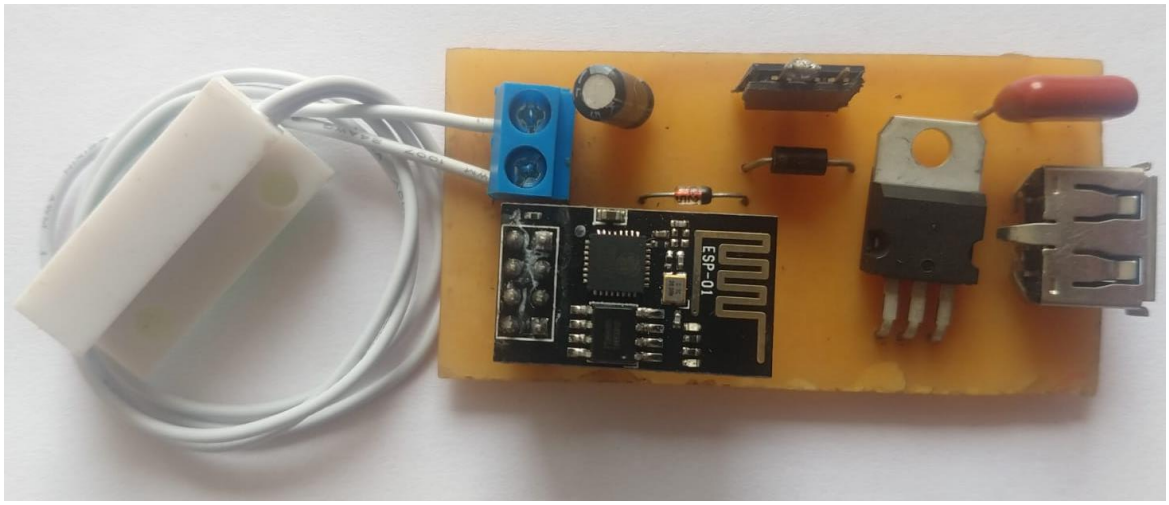


Figura 61. Circuito ESP-01 Ventana

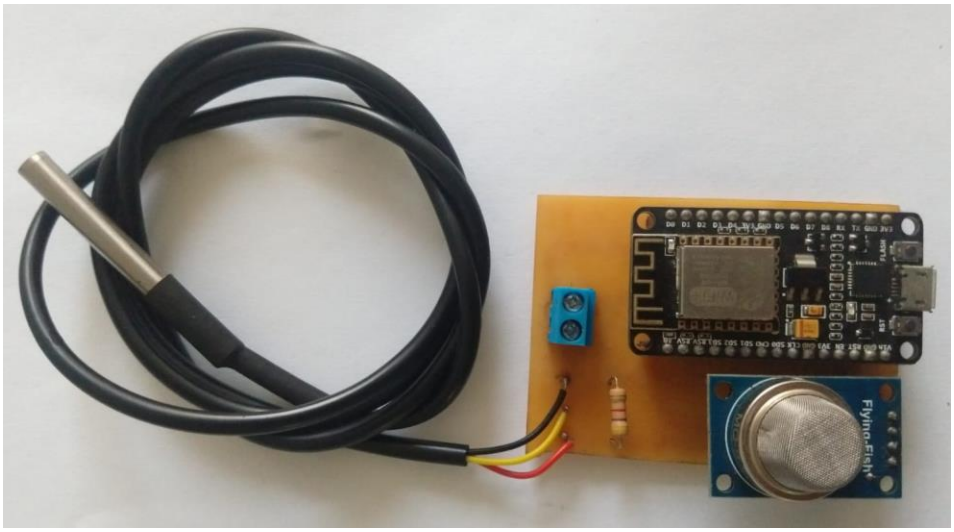


Figura 62. Circuito NodeMCU

9.2 Pruebas

Para realizar pruebas y permitir una manipulación sencilla de los circuitos se ajustaron y organizaron los circuitos con los sensores en una base como se puede observar en la Figura 63, los 3 circuitos se conectaron en paralelo a un cable con conector USB el cual se conectó a un cargador de +5V que puede proporcionar hasta 5W de potencia, en esta base se hicieron unas aberturas las cuales permiten el desplazamiento de abajo hacia arriba de los imanes que accionan el interruptor magnético. Para comprobar que los circuitos enviaran los datos al bróker se hizo uso de Putty para conectarse por medio de SSH al bróker y visualizar los datos recibidos.

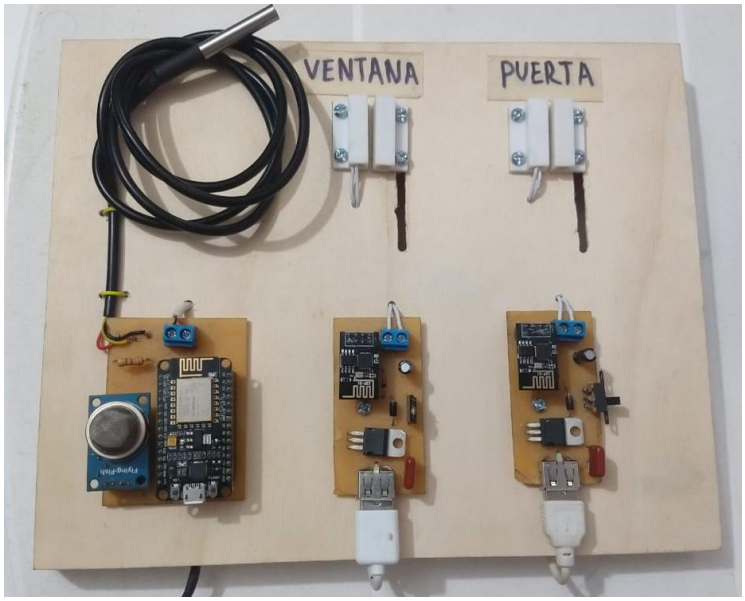


Figura 63. Base de pruebas

9.2.1 Pruebas de la aplicación

Para comprobar que la aplicación web funcionaba correctamente, al terminar el código del proyecto se creó una cuenta en netlify (ver Figura 64) que es una plataforma que permite alojar la aplicación web ver Anexo 2 donde se explica este proceso, una vez creada la cuenta, en el proyecto de React en la consola se ejecuta el comando ‘npm run build’ que creará una carpeta con el nombre build, donde estará la aplicación compilada, lista para subir a producción.

Al subir esta carpeta, se empezará al desplegar la aplicación con el nombre del dominio que se le asignó: domotica-monitoreo.netlify.app, al cargar la página, lo primero que se pedirá será habilitar las notificaciones, en caso de no pedirlo se debe hacer manualmente presionando el candado que aparece en la parte superior izquierda del navegador, se busca la parte de notificaciones y se presiona en permitir como se muestra en la Figura 65.

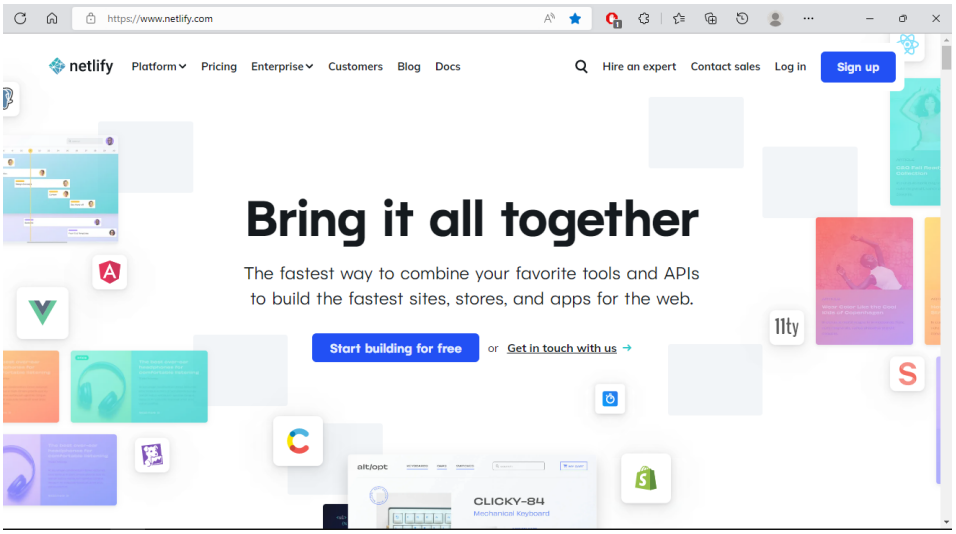


Figura 64. Netlify

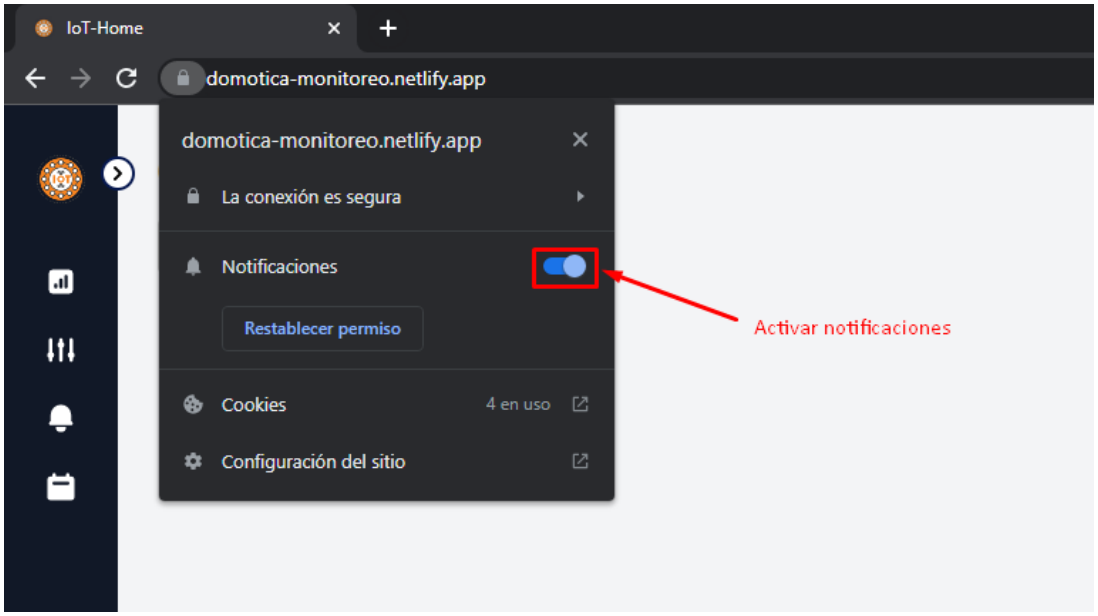


Figura 65. Activación de notificaciones

Activar las notificaciones es un paso importante, ya que en caso de no estar dentro de la aplicación aún se podrán recibir. La siguiente comprobación comprueba que la aplicación hubiese adquirido las características de una aplicación web progresiva, por lo tanto, desde un teléfono inteligente, se procedió a abrir el navegador, y buscamos la dirección donde se encuentra alojada la aplicación web. Al entrar nos pedirá activar las notificaciones y preguntará si desea instalar la aplicación en el teléfono, en caso de no aparecer esta opción, se busca en el navegador del dispositivo en la esquina superior derecha las opciones, y esto desplegará un menú como se muestra en la Figura 66, y seleccionaremos Instalar aplicación.

Al finalizar la descarga e instalación se mostrará un icono en la pantalla de inicio del dispositivo, y al abrirla se debe mostrar una versión de la aplicación ajustada a este tipo de formato de pantalla.

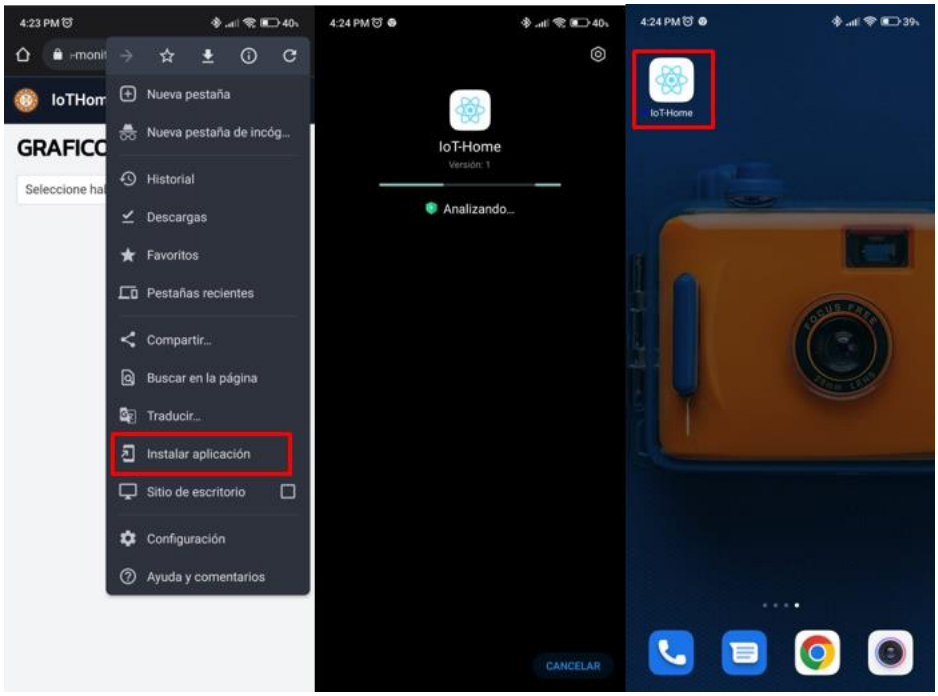


Figura 66. Instalación de la aplicación en un teléfono inteligente

Luego se pasó a configurar los sensores, para esto se abre la pestaña de configuración dentro de la aplicación, este proceso se puede realizar desde un computador o si la aplicación está instalada en el dispositivo móvil inteligente también será posible acceder desde este.

Como se muestra en la Figura 67 se crean dos habitaciones, la primera con el nombre ‘sala’ y la segunda con el nombre ‘cocina’, y se le agregan los tópicos correspondientes, el siguiente paso fue añadir los dispositivos. Se agregaron 4 dispositivos el primero de tipo numérico ubicado en la cocina, el segundo, tercero y quinto se agregó en la cocina y se indicó que fuera de tipo línea, luego se agregó el cuarto y el sexto ubicados en la sala y para estos se eligen que sean de tipo on/off ya que estos corresponden a sensores de puerta y ventana.

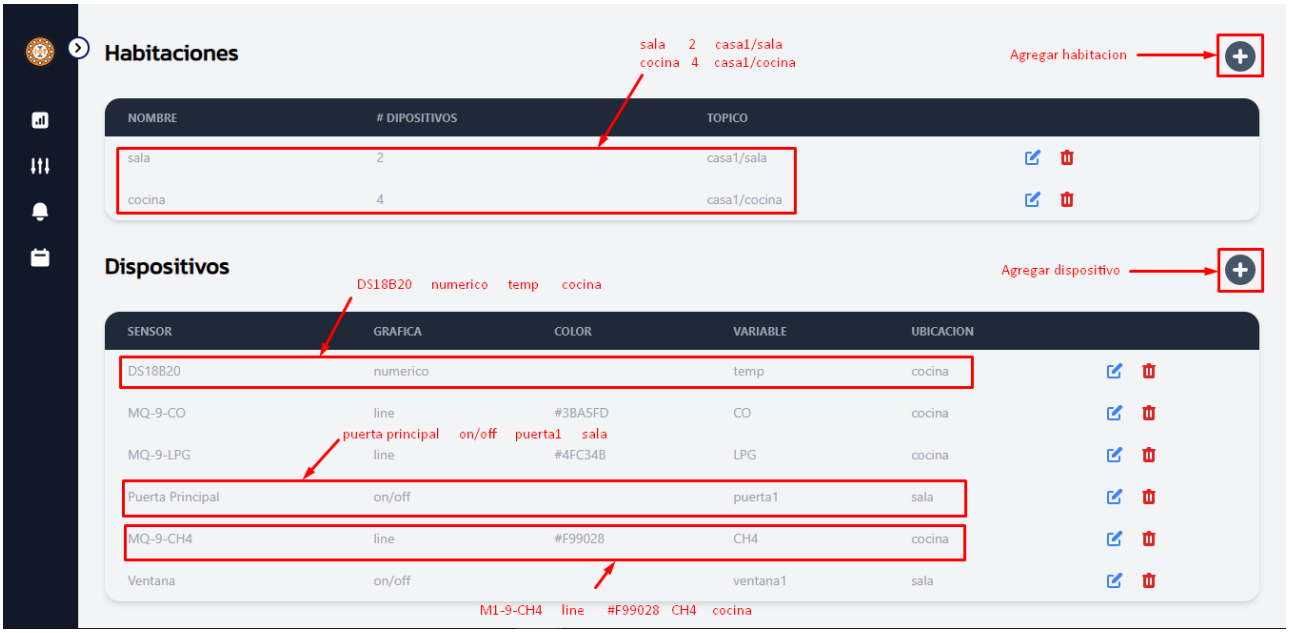


Figura 67. Prueba de configuración de sensores y habitaciones

Luego en la sección de alarmas, es donde se verán las alertas que envían a la aplicación, se incrementó la temperatura del sensor intencionalmente hasta que la temperatura pasara el umbral determinado previamente en la configuración de las alertas en el programa de Python que se encarga también de enviar los datos del bróker a Firebase, este umbral se determinó que cuando la temperatura alcanzara una temperatura mayor a los 35ºC enviara una notificación alertando sobre el incremento de temperatura y si la temperatura era mayor a 45ºC y se ha detectado CO por parte del MQ-9 enviara otra notificación alertando sobre el posible inicio de un incendio, cuando la temperatura paso el umbral mencionado anteriormente envía una notificación de temperatura en el dispositivo móvil como se muestra en la Figura 68, la notificación en la parte izquierda aparecerá cuando no se esté dentro de la aplicación, y la parte derecha de la figura es donde se encuentra el histórico de las alertas, lo que permite visualizar alguna alerta en caso de haberla omitido.

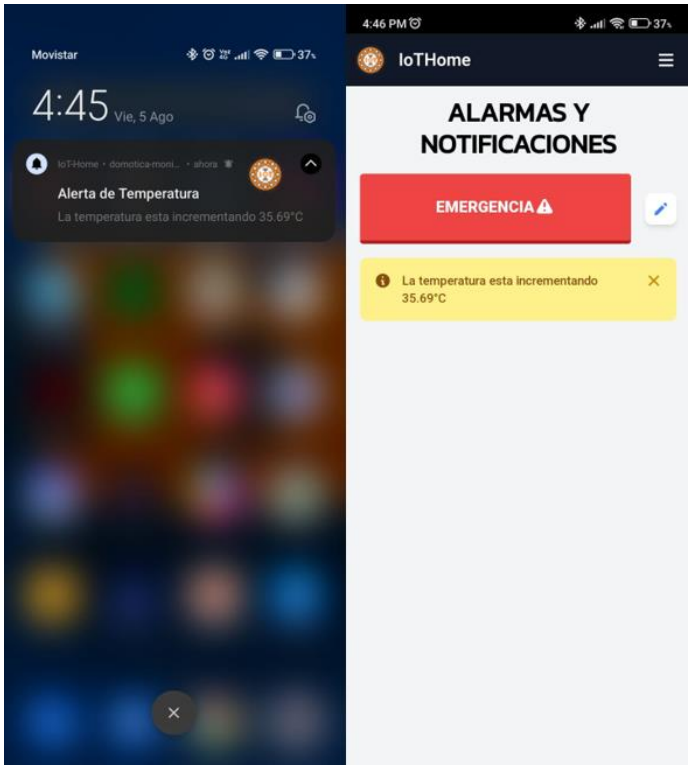


Figura 68. Prueba de notificaciones de la aplicación

Por último, se verificó que cada dato de los sensores se estuviese grabando en base de datos para poder ser consultados en algún momento, esto se puede realizar en la sección histórico en la aplicación, donde se eligió una fecha y hora inicial y final, se seleccionó el sensor al cual se desea consultar los datos, y se presiona el botón buscar, esto mostrará los datos en una gráfica como se puede visualizar en la Figura 69, donde hay datos con rangos de fechas entre las 9:24 am a las 11:48 am.

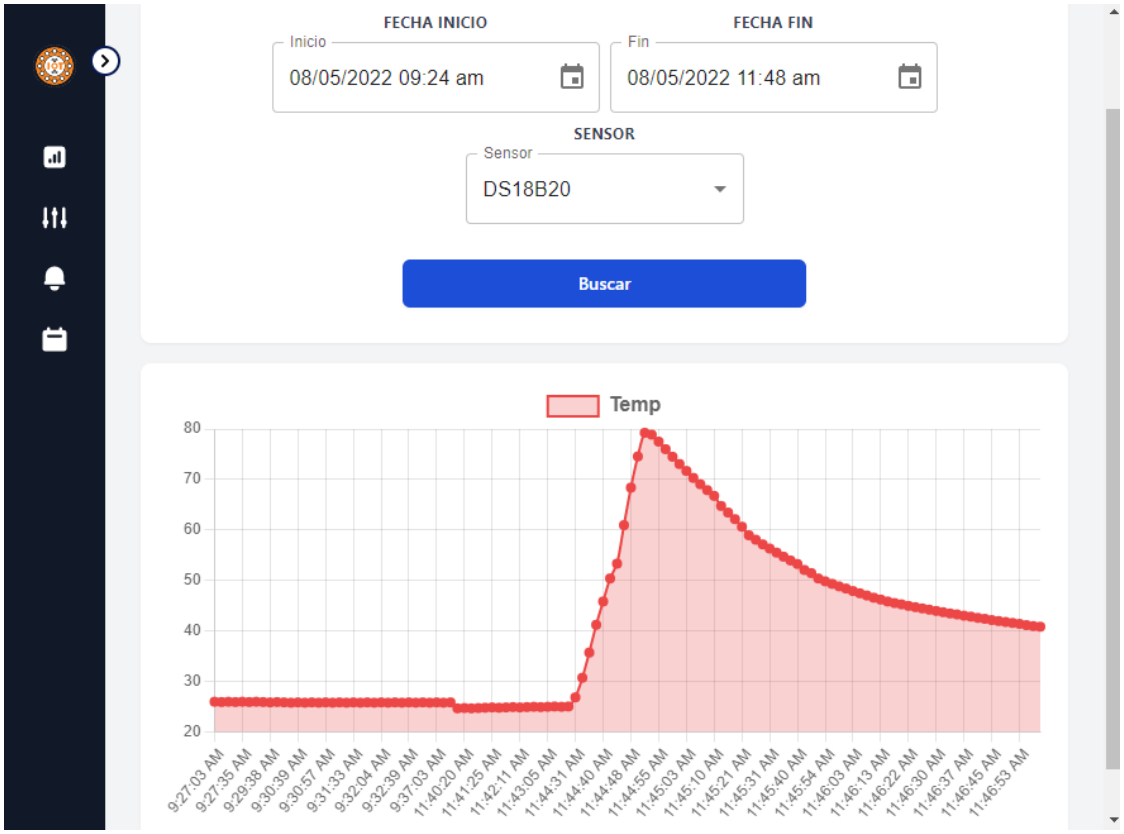


Figura 69. Prueba del histórico de la aplicación

9.2.2 Prueba sensor de apertura

Para la realización de esta prueba se posicionaron los imanes de los interruptores magnéticos que representan puerta y ventana en diferentes posiciones, primero ambos imanes junto a los interruptores como se puede observar en la Figura 70, donde se puede apreciar a la derecha de la figura la ventana del terminal del bróker, en la cual se muestran los datos que se están enviando al computador desde el bróker las cuales son el estado de la puerta y ventana, el cual está recibiendo el estado de “close” en ambas variables.

Figura 70. Prueba ventana y puerta cerrada

En la Figura 71 se puede observar que en la aplicación el estado de la puerta y ventana se encuentra cerradas.

Luego se hizo una prueba turnando el estado de ambos interruptores posicionando el estado del interruptor magnético de la ventana en abierto y el estado del interruptor magnético de la puerta en cerrado como se muestra en la Figura 72 y luego se invirtieron los estados como se puede observar en la Figura 73, en ambas figuras se puede apreciar el estado en que se encuentra los interruptores magnéticos y a su derecha la ventana del terminal de bróker recibiendo los datos de estos.

Figura 71. Prueba ventana y puerta cerrada aplicación

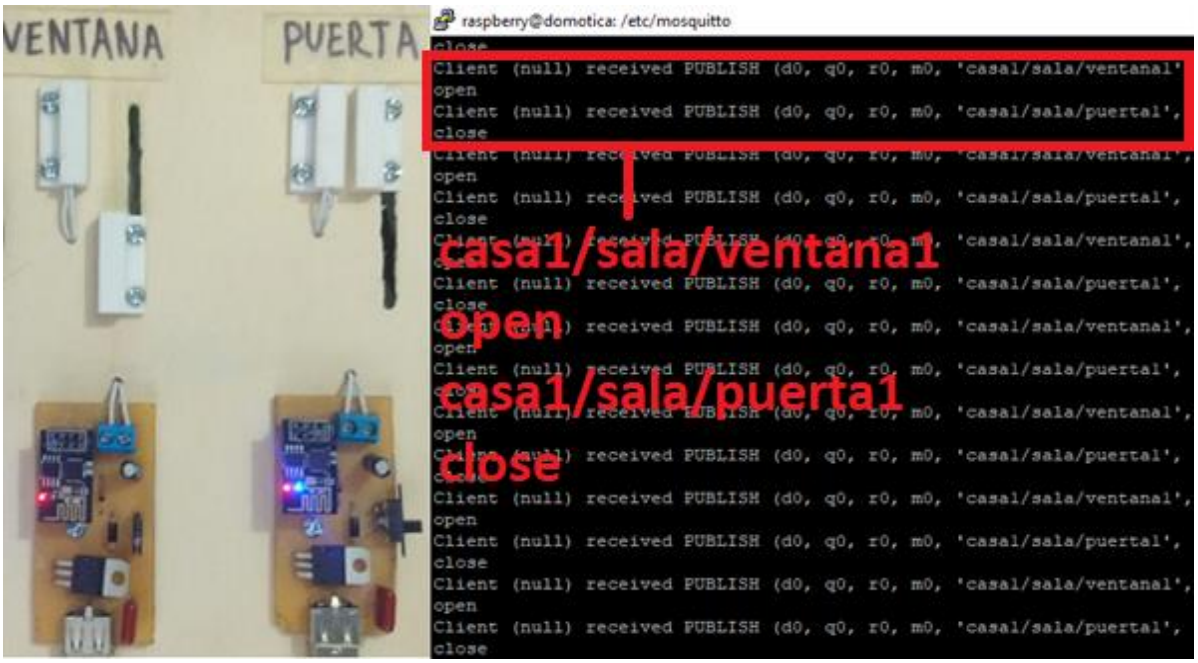


Figura 72. Prueba ventana abierta y puerta cerrada

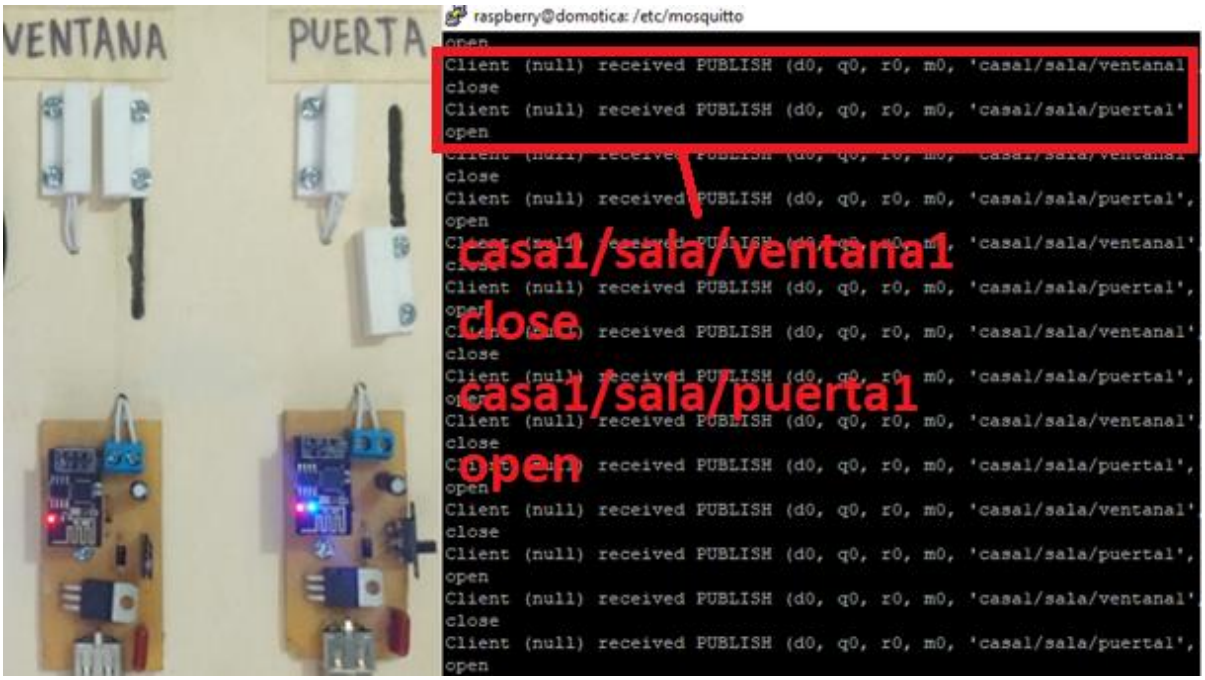


Figura 73. Prueba ventana cerrada y puerta abierta

En la Figura 74 se puede observar que la aplicación cambia el estado de las imágenes según el estado en que se encuentra la posición de los interruptores de puerta y ventana.

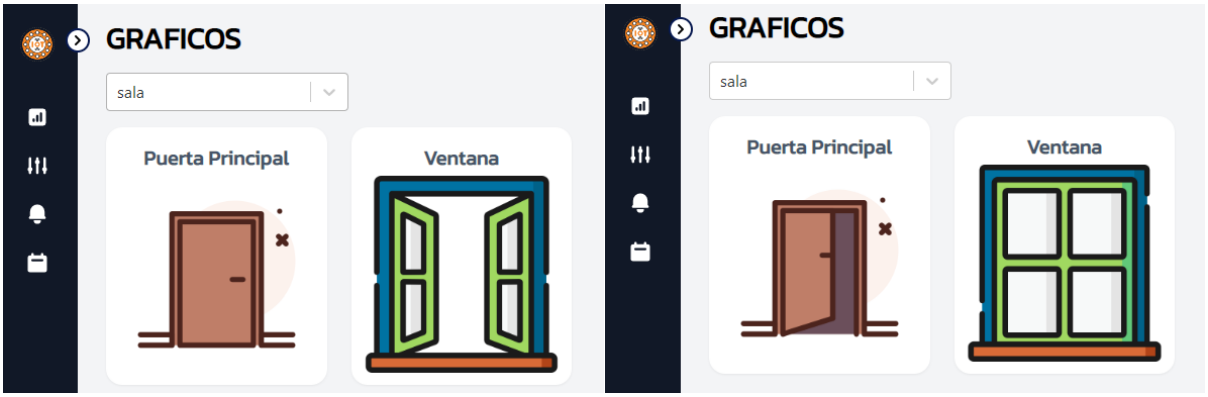


Figura 74. Prueba puerta y ventana alternando su posición

9.2.3 Prueba sensor de temperatura

Para la realización de la prueba del sensor de temperatura se hizo uso de un termómetro digital comercial de la marca Vick, el cual mide una temperatura entre 34 y 42ºC, agua caliente y al igual que con el interruptor magnético se observaron los datos enviados al bróker desde el módulo Wifi.

Primero se calentó un poco el agua hasta una temperatura que fuera capaz de medir el termómetro digital, para lograr esto se calentó el agua con el termómetro en su interior hasta que este no pudiera medir más, unas vez calentada el agua se procedió a introducir el termómetro y el sensor de temperatura DS18B20, como se puede observar en la Figura 75, el termómetro digital marcó una temperatura de 42.1ºC, a la derecha en el terminal de la Raspberry se puede observar que se recibe una temperatura de 42.13ºC lo que es una medida muy cercana a la medida por el termómetro digital, en la Figura 76 se puede observar la respuesta de la aplicación respecto a la temperatura que fue enviada por el sensor de temperatura. Este proceso de medición de temperatura del agua se repitió 15 veces utilizando el mismo termómetro dando como resultado la Tabla 9, el uso del termómetro Vick sirvió como referencia para la medición de la temperatura, pero en realidad no es un verdadero patrón que permita de manera exacta detectar la exactitud de la medida tomada por el sensor DS18B20.

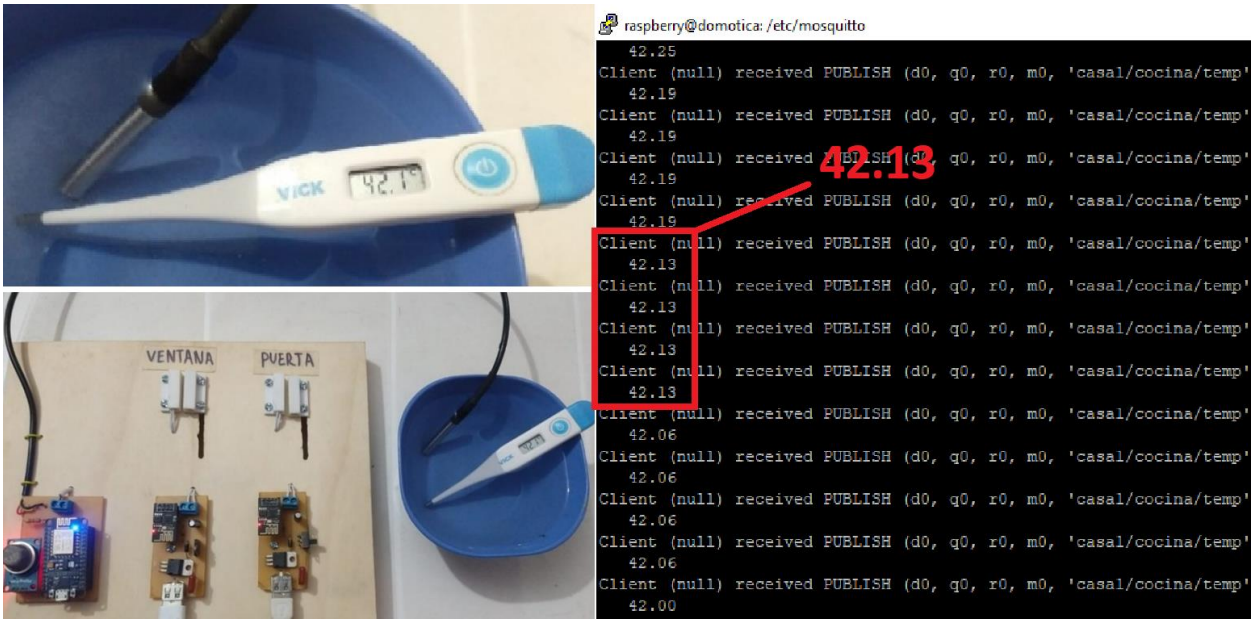


Figura 75. Prueba del sensor de temperatura

Figura 76. Prueba del sensor de temperatura aplicación

Tabla 9. Mediciones temperatura Termómetro VICK vs DS18B20

Termómetro VICK	*DS18B20*
42.3	*42.20*
42.1	*42.13*
41.6	*41.56*
40.8	*40.75*
39.8	*39.81*
39.5	*39.50*
39.1	*39.13*
38.7	*38.81*
38.5	*38.50*
38.1	*38.13*
37.9	*37.94*
37.7	*37.63*
37.3	*37.25*
37.1	*37.06*
36.9	*36.88*

9.2.4 Prueba sensor de gas

Para la realizacion de la prueba del sensor de gas MQ-9 se sometió el sensor a ciertas condiciones las cuales consistian en exponerlo al humo provocado por la quema de una hoja de papel, a la presencia del gas natural de las boquillas de una estufa y por último al gas contenido en los encendedores de cocina, para observar si podía detectar la presencia de los gases y su medida en ppm. Los resultados de las pruebas se pueden observar en la Figura 77, evidenciando que efectivamente el sensor mide la presencia de estos gases, los envía al broker y estos llegan a la aplicación, en la Figura 78 se puede observar la respuesta que tuvo el sensor cuando se midió el humo emitido por una hoja de papel quemandose, en la Figura 77 se puede observar que los datos señalados en rojo corresponden a puntos mostrados en las gráficas de la aplicación que se muestran en la Figura 78.

Figura 77. Prueba sensor de gas

Figura 78. Prueba sensor de gas aplicación

10. Conclusiones

Al investigar sobre los factores principales de riesgo de accidentes que existen dentro del hogar que puedan ser monitoreados, se logró determinar que los accidentes más propensos a ocurrir en los hogares son los incendios y las intoxicaciones, lo que los convierte en un tema de mucho cuidado, ya que en la mayoría de los casos estos accidentes se pueden prevenir. Otro caso vital para la seguridad del hogar es el ingreso de extraños a la vivienda, ya que el ingreso de personas no deseadas no solo supone perdida de bienes materiales, sino que puede resultar en una situación más peligrosa como lo puede ser el secuestro.

Se construyeron circuitos de medición de temperatura, humo, CH4, LPG y estado de puertas y ventanas haciendo uso del sensor DS18B20, sensor MQ-9 e interruptores magnéticos. Estos sensores son ideales para circuitos y diseños caseros debido a su costo, precisión y su amplio rango de medición, algunos de estos sensores se usan incluso en ámbitos industriales.

Se diseñó e implementó un sistema de medición inalámbrico haciendo uso de los Módulos Wifi ESP-01 y NodeMCU V3, Estos módulos son ideales para el uso en proyectos de domótica o IoT, ya que permiten la conexión inalámbrica de los sensores hacia algún otro dispositivo mediante el uso de Wifi con unas pocas líneas de código.

Se utilizó el protocolo MQTT para la comunicación entre los dispositivos hacia un dispositivo que cumple la función de bróker. Este protocolo de comunicación es muy utilizado en proyectos de IoT gracias a que permite una comunicación ligera entre los dispositivos, le brinda

escalabilidad al proyecto y es soportado por dispositivos con restricciones de recursos o ancho de banda de red limitado.

Se construyó un sistema de medición de variables central utilizando la Raspberry Pi Zero W como unidad central de datos y bróker. Esta Raspberry Pi cuenta conexión Wifi y bluetooth, es muy compacta y cuenta con recursos de procesamiento suficientes para instalar otros softwares para la construcción de proyectos de electrónica o de informática como Python, lo que la hace ideal para proyectos de IoT.

Se desarrolló una aplicación web en JavaScript haciendo uso de la biblioteca React que permitiera presentar la información de los sensores con una interfaz de usuario agradable, agregar gráficas, configurar los sensores que se deseen y además presentar alertas sobre el estado de las variables medidas. Los datos de los sensores se guardaron en la base de datos de Firebase la cual ofrece planes gratuitos y permite hacer uso de muchas de sus herramientas sin costos adicionales.

Se construyó un sistema centralizado escalable de medición y monitoreo de variables utilizando IoT para la prevención de accidentes en el cual los usuarios son notificados por alertas enviadas a los dispositivos a través de la aplicación web cuando las variables medidas por los sensores se acerquen a valores que puedan poner en riesgo la seguridad de la vivienda. Este sistema se puede implementar en cualquier tipo de casa o lugar que cuente con una red wifi local con conexión a internet y la aplicación se puede instalar y usar en cualquier dispositivo que cuente con un navegador de internet como los smartphones, tabletas y ordenadores.

11. Recomendaciones y Trabajos Futuros

En el trabajo presente se desarrolló un sistema de medición y monitoreo de variables, y a pesar de haberse logrado los objetivos de esta investigación e implementación, este sistema puede ser aún más dinámico, y configurable, quedan muchas áreas de trabajo para el proyecto.

Como recomendaciones se puede mejorar la aplicación haciendo esta aún más flexible permitiéndole al usuario agregar muchos más dispositivos en el sistema como cámaras de vigilancia, y sensores de movimiento, también es importante recomendar el uso de circuitos impresos por medio de máquinas que hagan un trabajo fino y preciso.

Mejorar el sistema con actuadores como rociadores en caso de detectarse un incendio, para que el sistema reaccione automáticamente, y evitar la mayor cantidad de incidentes posible.

12. Referencias

ISO/IEC JTC 1/SC 41 . (Agosto de 2018). *ISO*. Obtenido de ISO/IEC 30141:2018 - Internet of Things (IoT) — Reference Architecture: https://www.iso.org/standard/65695.html

Aldana, E. S., & Citelly, D. S. (2017). *Relación de los factores socioeconómicos con el rendimiento académico de los estudiantes de educación media para Colombia en el segundo semestre del 2017 : un enfoque geoeconómico.* Bogota D.C: Universidad de La Salle.

Aránguez Ruiz, E. (s.f). Contaminantes atmosféricos y su vigilancia. *Revista Española de Salud Pública*. Obtenido de https://www.scielosp.org/article/resp/1999.v73n2/123-132/es/

Arduino. (26 de Abril de 2021). *MCI Electronics*. Obtenido de https://arduino.cl/que-es-arduino/

Ariza, J. D. (2019). *Sistema de Control y Monitoreo de Consumo Energético para Equipos de Climatización Orientado a Internet de las Cosas (IoT).* Barranquilla: Universidad de la Costa, CUC.

ATSDR. (6 de Mayo de 2016). *Óxidos de nitrógeno (monóxido de nitrógeno, dióxido de nitrógeno, etc.) (Nitrogen Oxides) | ToxFAQ | ATSDR*. Obtenido de ATSDR: https://acortar.link/e7GER

Bajaña Molina, H. J., & Molina Sarco, J. C. (2020). *Diseño e implementación de un prototipo escalable de detección de gases inflamables, temperatura y alarmas contra incendios basado en tecnología iot de bajo costo para cocinas en viviendas de guayaquil.* Guayaquil, Ecuador: Universidad de Guayaquil.

Barrera, G. M. (2018). *Estilo arquitectónico para aplicaciones IoT.* Buenos Aires: Universidad del CEMA.

Benítez, M. D., Anías, C. C., & Plasencia, M. L. (2016). *Propuesta de arquitectura para Internet de las Cosas.* Habana: Universidad Tecnológica de La Habana José Antonio Echeverría.

Biblioteca Nacional de Medicina. (s.f.). *Asfixia en adulto o niño mayor de 1 año*. Recuperado el 2 de Abril de 2021, de MedlinePlus: https://acortar.link/LaAnn

Cárdenas, G. G., & Gómez, F. A. (2013). Diseño e implementación de una Tarjeta de Desarrollo con profundización en desarrollo de aplicación de Touch Sensing. *11th Latin American and Caribbean Conference for Engineering and Technology*, 14-16.

Carrillo Sampedro, C. S., & Villagrán Sánchez, B. A. (2008). *Implementación de un prototipo de tele - cuidado por medio de tele – ubicación de personas en riesgo (ancianos, discapacitados, epilépticos, enfermos del corazón, diabéticos, Alzheimer, etc.).* Quito: Escuela Politécnica Nacional.

Cisco. (2016). *Introduccion a redes.* Obtenido de Cisco Networking Academy: https://acortar.link/Er0Gu

Collina, M. (s.f.). *Aedes*. Obtenido de https://github.com/moscajs/aedes

Concepción, R. (26 de Abril de 2021). *rjconcepcion*. Obtenido de https://acortar.link/8r6E4

de Sousa, I. (14 de Junio de 2019). *rockcontent*. Obtenido de ¿Qué es un servidor web y cuáles son sus características?: https://rockcontent.com/es/blog/que-es-un-servidor/

Debian. (28 de 06 de 2022). *systemd.service - Service unit configuration: Debian Manpages*. Obtenido de Debian Manpages: https://manpages.debian.org/testing/systemd/systemd.service.5.en.html

Dewesoft. (09 de Marzo de 2020). *Dewesoft*. Obtenido de https://acortar.link/NEXcl

Eclipse Foundation. (s.f.). *Mosquitto*. Obtenido de https://mosquitto.org/

Enrique, C. J. (27 de Abril de 2021). *Aprendiendo Arduino*. Obtenido de https://acortar.link/sUu7A

Escuela Cántabra de salud. (2019). *Escuela cántabra de salud*. Obtenido de Intoxicaciones - Escuela Cántabra de Salud: https://acortar.link/Of5Ll

Firebase Realtime Database. (11 de 06 de 2022). Obtenido de https://firebase.google.com/docs/database?hl=es-419

Forensis. (2020). *Datos para la vida.* Bogota D.C: Instituto Nacional de Medicina Legal y Ciencias Forenses.

Geekflare. (21 de Agosto de 2020). *Geekflare*. Obtenido de 12 plataformas y herramientas de Internet de las cosas (IoT) de código abierto: https://geekflare.com/es/iot-platform-tools/

Golondrino, G. E., Alarcón, M. A., & Ríos, M. E. (2020). Arquitectura IoT para el desarrollo de sistemas de monitorización y análisis de variables fisiológicas en el área de asistencia médica. *Revista Investigación e Innovación en Ingenierías, vol. 8, n°3,* 1-13.

Google Developers. (29 de 07 de 2022). *Firebase.* Obtenido de https://firebase.google.com/docs/build?hl=es-419

Gracia, M. (27 de Abril de 2021). *Deloitte*. Obtenido de https://acortar.link/EheqS

GreenFacts. (5 de Agosto de 2022). *GreenFacts* . Obtenido de Partes por millón: https://www.greenfacts.org/es/glosario/pqrs/partes-million.htm

Grueso Carabali, R., & Torres Orozco, M. A. (2020). *Envility: Aplicación del internet de las cosas al monitoreo de la calidad ambiental de los salones de clase de la Uniajc.* Cali, Colombia: Institución Universitaria Antonio José Camacho.

Gustavo, B. (3 de Diciembre de 2020). *Hostinger Tutoriales*. Obtenido de Hostinger.co: https://www.hostinger.co/tutoriales/que-es-mysql

HANWEI. (29 de Agosto de 2022). *alldatasheet.com.* Obtenido de MQ-9 Datasheet: https://pdf1.alldatasheet.com/datasheet-pdf/view/1221388/HANWEI/MQ-9.html

HiveMQ GmbH. (23 de Junio de 2022). *HiveMQ*. Obtenido de MQTT Essentials: https://www.hivemq.com/mqtt-essentials/

HiveMQ. (s.f.). *HiveMQ*. Obtenido de https://www.hivemq.com

Hospital Alemán Asociación Civil. (18 de junio de 2018). *hospital aleman*. Obtenido de Intoxicación por inhalación de gas: cuáles son los riesgos para la salud frente a un escape: https://www.hospitalaleman.org.ar/nuestro-hospital/ha-en-los-medios/intoxicacion-por-inhalacion-de-gas-cuales-son-los-riesgos-para-la-salud-frente-a-un-escape/

Hostinger. (27 de Mayo de 2022). *Hostinger Tutoriales*. Obtenido de ¿Cómo funciona el SSH?: https://www.hostinger.co/tutoriales/que-es-ssh

IEBS. (5 de Noviembre de 2019). *IEBS Digital School*. Obtenido de ¿Qué son las Progressive Web Apps? ¿Por qué son tan importantes?: https://www.iebschool.com/blog/progressive-web-apps-analitica-usabilidad/

Infobae. (18 de Junio de 2018). *Infobae.* Obtenido de Intoxicación por inhalación de gas: cuáles son los riesgos para la salud frente a un escape: https://www.infobae.com/salud/2018/06/18/intoxicacion-por-inhalacion-de-gas-cuales-son-los-riesgos-para-la-salud-frente-a-un-escape/

ISO/IEC JTC 1/SC 31 . (Agosto de 2016). *ISO/IEC 29161:2016 - Information technology — Data structure — Unique identification for the Internet of Things*. Obtenido de ISO: https://www.iso.org/standard/45240.html

Lee, F. (s.f.). *Erlang MQTT Broker*. Obtenido de https://www.emqx.io/docs/en/v1.0/#license

Martinez, J. C. (15 de Septiembre de 2021). *YMANT*. Obtenido de ¿Qué es un AP (Access Point) y que usos y modos tiene?: https://www.ymant.com/blog/que-es-un-ap-access-point-y-que-usos-y-modos-tiene/

MDN Web Docs. (2005-2021). *Generalidades del protocolo HTTP*. Obtenido de MDN Web Docs: https://acortar.link/wYXMs

Mendez León , S. G., & Vásquez Torres, L. C. (2020). *Prototipo de artefacto iot para la detección de riesgos y prevención de accidentes en la cocina del hogar.* Bogota, Colombia: Universidad Piloto de Colombia.

Microsoft. (s.f.). *Qué es el almacenamiento en la nube y cómo se utiliza*. Recuperado el abril de 2020, de Microsoft Azure: https://acortar.link/J8xW1

Ministerio de Salud y Protección Social. (2020). *Orientaciones para medidas de seguridad y de prevención de accidentes en el hogar en el marco del estado de emergencia por SARS-COV-2 (covid-19).* Bogotá: Ministerio de Salud y Protección Social. Obtenido de

https://www.minsalud.gov.co/Ministerio/Institucional/Procesos%20y%20procedimientos/GIPG19.pdf

Ministerio de tecnologías de la información y las comunicaciones. (2019). *Resolucion 000964 de 2019.* Bogotá: Ministerio de tecnologías de la información y las comunicaciones. Obtenido de https://acortar.link/JHBBD

Núñez Sebastián, A. (2012). *KNX. Domótica e Inmótica: Guía Práctica para el instalador.* Ediciones Experiencia. Obtenido de https://books.google.com.co/books?id=UP-LDwAAQBAJ&printsec=frontcover&dq=inauthor:%22Antonio+N%C3%BA%C3%B1ez+Sebasti%C3%A1n%22&hl=es&sa=X&ved=2ahUKEwjT_IutxaTwAhVAFFkFHdkED0sQ6AEwAHoECAAQAg#v=onepage&q&f=false

Observatorio de Salud de Bogotá. (2020). *Tasa de notificación de accidentes domésticos en menores de 11 años en Bogotá D.C.* Obtenido de Saludata: https://acortar.link/2W69k

ORACLE. (5 de Agosto de 2022). *ORACLE*. Obtenido de Base de datos definida: https://www.oracle.com/co/database/what-is-database/

Organizacion Mundial de la Salud. (26 de 4 de 2021). *Caídas*. Obtenido de Organizacion Mundial de la Salud: https://acortar.link/9WkFj

Padilla, R. E., & Lobos, C. L. (2019). *Plataforma iot para el control y monitoreo de variables físicas con tecnología open hardware.* Santa Tecla: ITCA Editores.

Pérez Porto, J., & Gardey, A. (2013). *Definicion.de*. Obtenido de Monitoreo: https://definicion.de/monitoreo/

Pupo Rodríguez, G., Zunilda Leticia , B. F., Pavón Ramírez, M. A., Pacheco Pérez, Y., & Lluch-Silva, I. T. (2018). Brotes de intoxicación alimentaria ocurridos en los últimos diez años en Las Tunas. *Revista Electrónica Dr. Zoilo E. Marinello Vidaurreta*. Obtenido de http://www.revzoilomarinello.sld.cu/index.php/zmv/article/view/1562/pdf_537

Quiñonez Muñoz, O. (2019). *Internet de las Cosas (IoT).* Ibukku LLC. Obtenido de https://books.google.com.co/books?id=vnnEDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

RAE. (29 de Abril de 2021). *dle RAE*. Obtenido de https://dle.rae.es/hogar

RAE. (5 de Agosto de 2022). *Real Academia Española*. Obtenido de protocolo: https://dle.rae.es/protocolo#otras

RAE. (5 de Agosto de 2022). *Real Academia Española*. Obtenido de monitorear: https://dle.rae.es/monitorear#PecQiEQ

RAE. (5 de Agosto de 2022). *Real Academia Española*. Obtenido de aplicación: https://dle.rae.es/aplicación

Raspberry. (26 de Abril de 2021). *MCI Electronics*. Obtenido de https://raspberrypi.cl/que-es-raspberry/

Real Academica Española. (s.f.). *Monitorear.* En Diccionario de la lengua española (vigesimotercera edición). Obtenido de https://dle.rae.es/monitorear

Red Hat. (27 de Abril de 2021). *Red Hat*. Obtenido de https://acortar.link/HJWoG

Republica de Colombia. (5 de Enero de 2009). *Ley 1273 de 2009 Nivel Nacional*. Obtenido de Alcaldia Mayor de Bogotá: https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492

Republica de Colombia. (18 de Octubre de 2012). *Ley 1581 de 2012*. Obtenido de Gobierno de Colombia: https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981

Revistas, U. E. (2021). *Cómo curar quemaduras en niños*. Obtenido de CuidatePlus: https://acortar.link/emUxJ

Rock Content. (27 de Abril de 2021). *Rock Content blog*. Obtenido de https://acortar.link/wnkZN

Rojas Colunge, S. (2020). *Dispositivo de monitoreo centralizado y escalable para casas inteligentes.* Santiago de Cali: Universidad de San Buenaventura.

Salas Arriarán, S. (2017). *Todo sobre sistemas embebidos: Arquitectura, programación y diseño de aplicaciones prácticas con el PIC18F.* Universidad Peruana de Ciencias Aplicadas.

Significados. (5 de agosto de 2022). *Significados*. Obtenido de Significado de Nube: https://www.significados.com/nube/

SoCalGas Company. (23 de Junio de 2022). *Monóxido de carbono*. Obtenido de SoCalGas: https://acortar.link/j8ph0

SoftwareLab.org. (2014-2021). *¿Qué es WiFi, qué significa y para qué sirve?* Obtenido de SoftwareLab.org: https://acortar.link/2AhVk

Soto, A. B., Coral, D. S., AyaParra, P. A., Charry, Ó. J., Bernal, H. A., Torres, D. Q., & Rojas, J. S. (2019). *Sistema basado en internet de las cosas (iot) para la monitorización en tiempo*

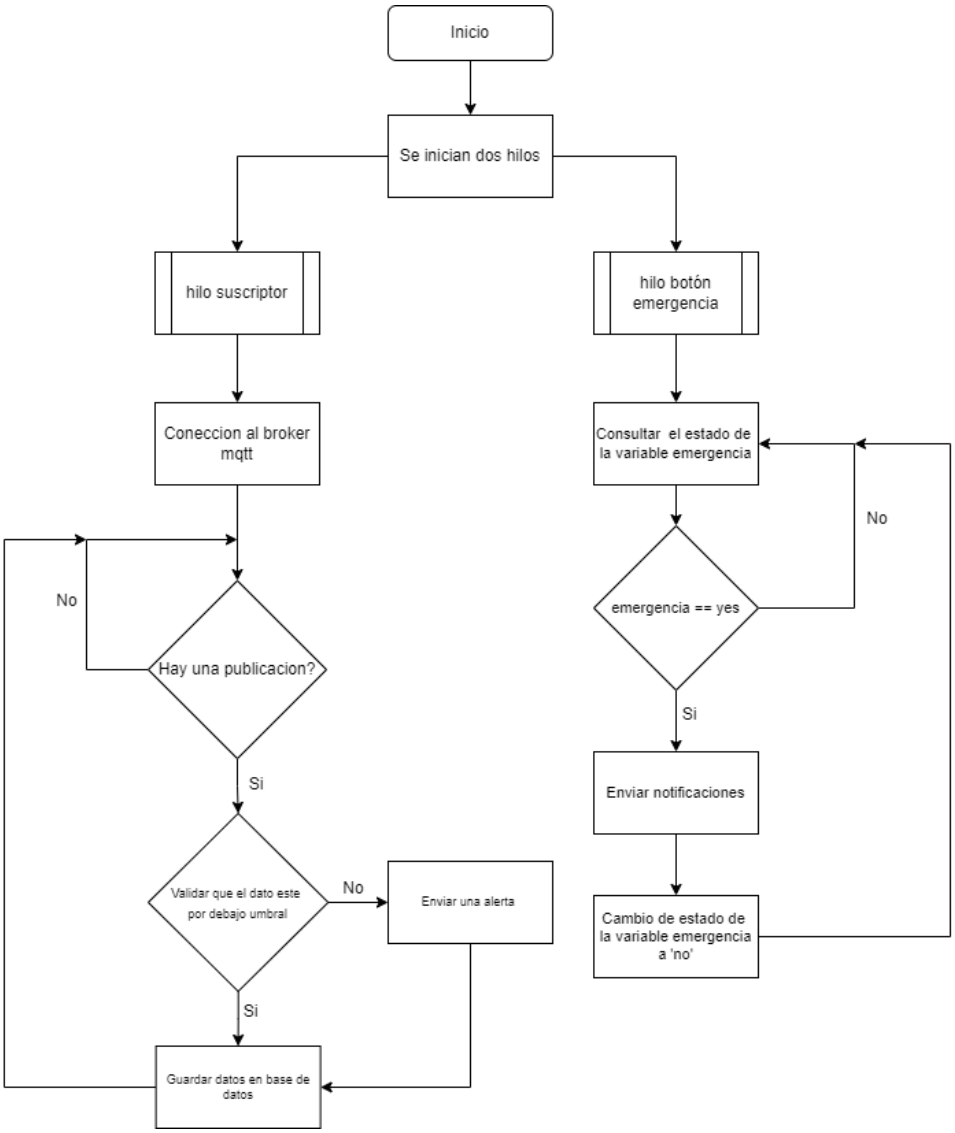
real de variables de temperatura y humedad en un equipo de refrigeración del área de farmacia de un hospital de cuarto nivel. Bogotá: Universidad del Rosario.

Tinjacá, D. L. (2019). *SISTEMA DE MEDICIÓN DE CALIDAD DE AIRE E INTENSIDAD UV CON SISTEMA.* Bogotá D.C: Universidad Distrital Francisco José De Caldas.

Zárate, J. C., & Romá, O. A. (2019). Sistema de monitoreo de monóxido de carbono en tiempo real en. *Revista CINTEX Vol 24(2)*, 25-32.

13. Anexos

13.1 Anexo 1



13.2 Anexo 2

Para subir la aplicación al alojamiento gratuito de netlify, el primer paso es crear una cuenta, en el navegador se procede a dirigirse a la ruta www.netlify.com , una vez ahí, en la esquina superior derecha encontraremos el botón Sign up (Registrarse) ver Figura 79.

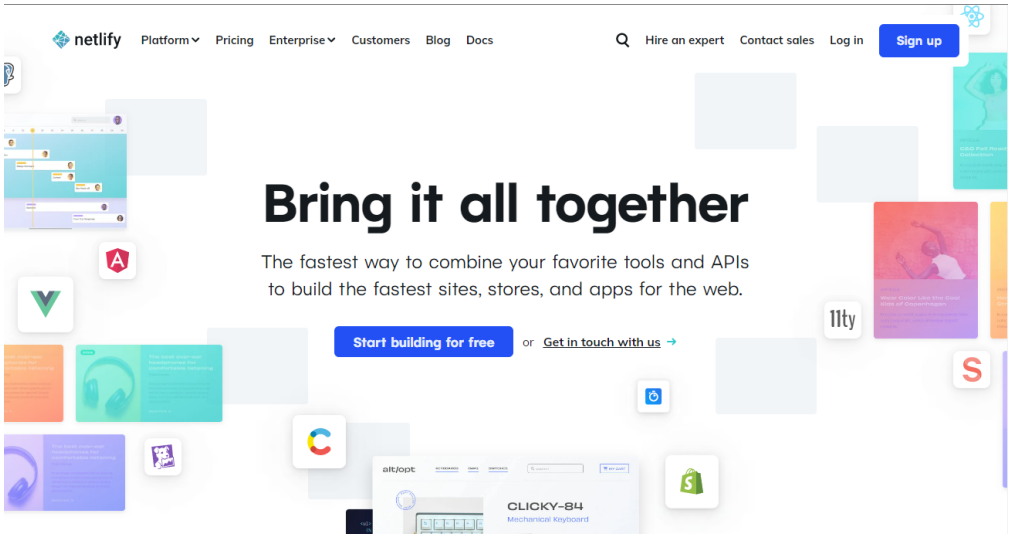


Figura 79. Página de inicio netlify

Al presionar el botón, se abrirá otra página en donde se pedirá la forma con la que se quiere registrar como se muestra en la Figura 80, se seleccionó la opción de Github, donde es necesario autenticarse para poder hacer uso de netlify.

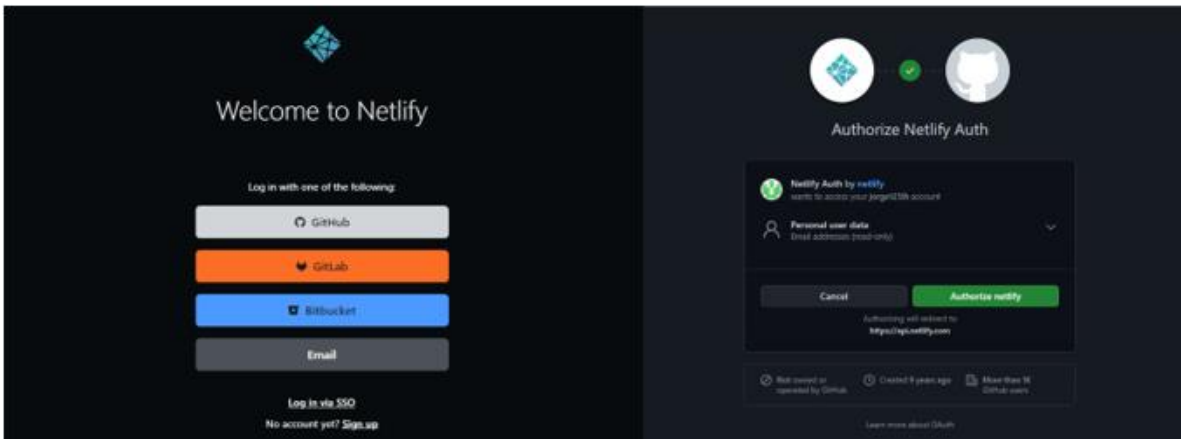


*Figura 80.*Registro y autenticación en netlify

Una vez creada la cuenta se abrirá la página donde se podrá subir la aplicación, en la sección Sites se podrá subir la carpeta build que se generó con el comando npm run build en nuestro proyecto de ReactJs, solo bastará con arrastrar la carpeta hasta el recuadro que se marca en la Figura 81.

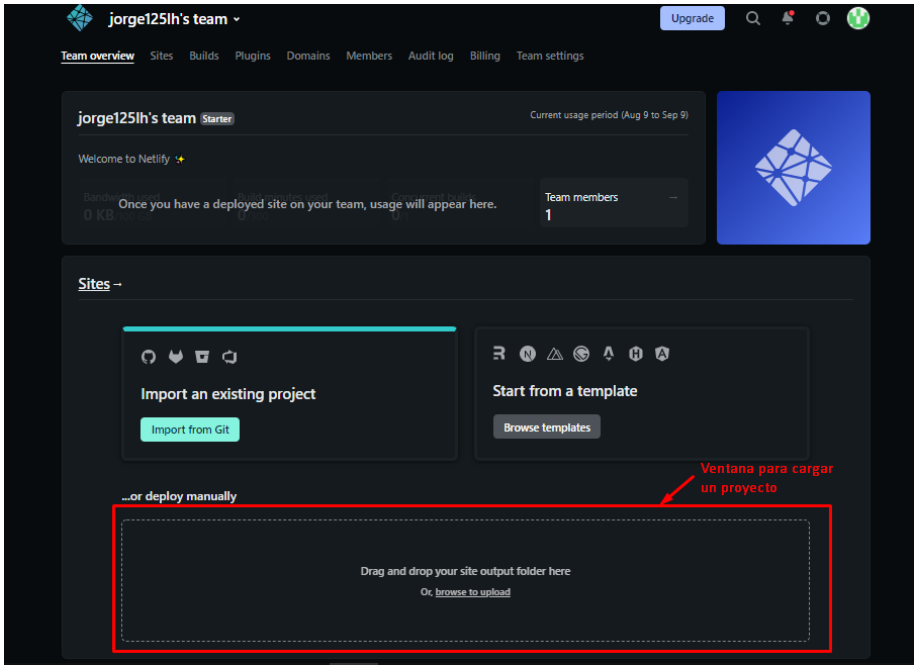


*Figura 81.*Carga y despliegue del proyecto en netlify

De esta manera se empezará a desplegar la aplicación y cuando esté listo, se generará un link con el cual se podrá consultar la página como muestra la Figura 82, en donde aparece una miniatura de como se verá la aplicación montada, y un vínculo en la parte izquierda.

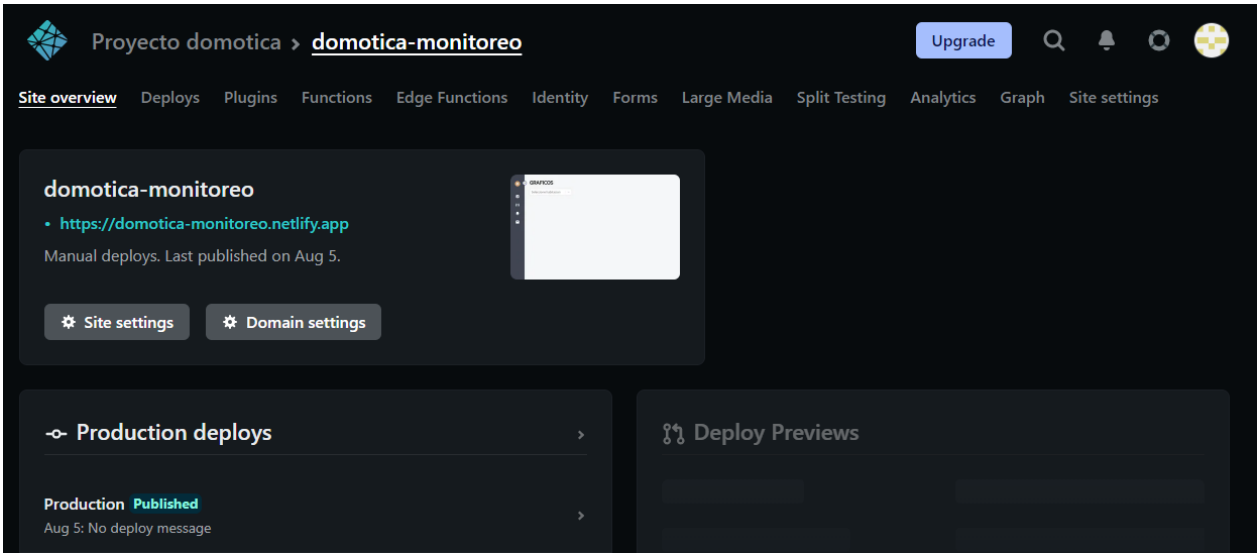
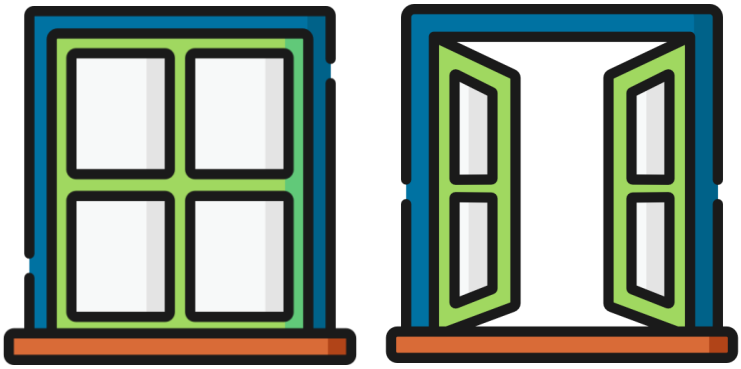


Figura 82. Publicación de la pagina

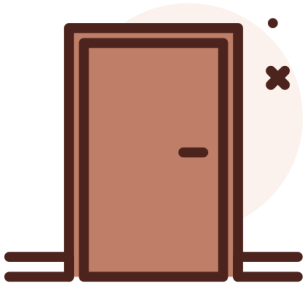
Presionando en el link https://domotica-monitoreo.netlify.app se abrirá una pestaña con la aplicación ajustándose al formato de pantalla desde donde se consulta.

13.3 Anexo 3

Las imágenes siguientes son usadas en la aplicación del proyecto y se referencian con su respectiva url de donde se sacaron.



Fuente: https://www.flaticon.com/free-icons/window



Fuente: https://www.flaticon.com/free-icons/closed-door

Fuente: https://www.flaticon.com/free-icons/open-door