

SISTEMA DE ALARMA PARA VIVIENDA CON TECNOLOGÍAS IoT

Andrés Felipe Gallo Gonzales

María Camila Rueda Cano



Universidad de Manizales

Facultad de Ciencias e Ingeniería

Programa de Ingeniería de Sistemas y Telecomunicaciones

Manizales, 2024

SISTEMA DE ALARMA PARA VIVIENDA CON TECNOLOGÍAS IoT

Andrés Felipe Gallo Gonzales

María Camila Rueda Cano

Director:

Ing. Diego Samir Melo Solarte

Trabajo de grado presentado como opción parcial para optar

al título de

Ingeniera de Sistemas y Telecomunicaciones

Universidad de Manizales

Facultad de Ciencias e Ingeniería

Programa de Ingeniería de Sistemas y Telecomunicaciones

Manizales, 2024

Agradecimiento

Este proyecto no era posible realizarse sin el granito de arena aportado por nuestros familiares los cuales siempre brindaron el impulso para poder finalizar nuestra carrera y el soporte económico para poder realizar el proyecto.

También queremos dar un agradecimiento especial al profesor Diego Samir Melo Solarte por su interés y ser guía fundamental en el desarrollo del proyecto, agradecemos por su dedicación, consejos y sobre todo por el conocimiento dado.

Tabla de Contenido

Introducción	4
1. Área Problemática	5
2. Objetivos	6
2.1. Objetivo General	6
2.2. Objetivos Específicos	6
3. Justificación	7
4. Marco Teórico	8
4.1. Marco Conceptual	8
4.1.1. <i>Arduino Esp32</i>	8
4.1.2. <i>Sensores</i>	10
4.1.3. <i>Bróker</i>	13
4.1.4. <i>Android Studio</i>	16
4.1.5. <i>Sistema de Alarma Para Vivienda</i>	18
4.1.6. <i>Fritzing</i>	21
4.2. Antecedentes	21
5. Metodología	27
5.1. Procedimiento.....	27
5.1.1. <i>Fase 1. Diseñar la arquitectura de un sistema de detección de intrusos y riesgos en una vivienda utilizando dispositivos de bajo costo.</i>	27

5.1.2.	<i>Fase 2. Establecer el mecanismo de comunicación entre los dispositivos de detección de intrusos o riesgos y los propietarios de la vivienda.</i>	31
5.1.3.	<i>Fase 3. Implementar una prueba piloto del sistema en la vivienda o en las viviendas de los autores.</i>	33
6.	Resultados	37
6.1.	Descripción de Resultados	37
6.2.	Discusión de Resultados	39
7.	Conclusiones	41
10.	Anexos	48

Lista de Figuras

Figura 1. Estructura ESP32.....	9
Figura 2. diagrama de bloques de esta tarjeta electrónica.	10
Figura 3. Módulo KY_026 Sensor de Flama.	11
Figura 4. Sensor de Gas Aire Mq2.	12
Figura 5. Sensor Infrarrojo de Movimiento Hc-sr501.....	13
Figura 6. Arquitectura de MQTT.	15
Figura 7. Sensor Infrarrojo Llama Flama Ky026.....	28
Figura 8. Sensor de Aire Gas MQ2 Frontal.....	29
Figura 9. Sensor de Aire Gas MQ2 Posterior.....	29
Figura 10. Sensor Infrarrojo De Movimiento Hc-sr501.....	30
Figura 11. Diseño de Sistema de Alarma.....	31
Figura 12. Diseño de la aplicación de Sistema de Alarma.....	33
Figura 13. Circuito del sistema de alarma.....	39
Figura 14. Prototipo de sistema de alarma.....	39
Figura 15. Módulo TP4056.....	42
Figura 16. Esquema de conexión batería (Stopa, 2019).....	44

Lista de Anexos

Anexo A. Código Arduino.....	48
Anexo B. Código Android Studio	55
Anexo C. Prender y Apagar el Sistema de Alarma por el Aplicativo Móvil.	66
Anexo D. Prueba Sensor de Gas	67
Anexo E. Prueba Sensor de Movimiento	69

Lista de Tablas

Tabla 1.. Tabla de Presupuesto.....	40
Tabla 2. Tabla Piezas necesarias para incluir batería	43
Tabla 3. Tabla de presupuesto futuro	44

Resumen

El objetivo de este proyecto es proporcionar un sistema de alarma para viviendas con tecnologías IoT (Internet de las cosas), diseñado para prevenir hurto o algún accidente doméstico, además identificar acontecimientos excepcionales sobre el estado de su vivienda notificando al propietario.

Se empleó un Arduino esp32 junto con varios sensores que sirven para la detección de movimiento, humo, fuego y gas, estos permiten identificar las anomalías en el espacio de la casa en el cual fue instalado. La información recibida por los sensores es enviada a una aplicación Android mediante el protocolo de mensajería mqtt, la cual está encargada de informar al usuario en tiempo real sobre cualquier situación inusual detectada en la vivienda. Además, brinda la opción que el usuario pueda desactivar la alarma y revisar detalladamente el estado actual de la vivienda.

Para la incorporación de este sistema se requirió de la configuración de un clúster en hiveMQ, permitiendo tener un bróker privado el cual coordina la mensajería entre el Arduino y la aplicación en Android. El sistema garantiza la seguridad de la vivienda en cuanto a la detección de incendio, posible incendio derivado de fuga de gas, posible incendio iniciado por fuego o cuando una persona no identificada irrumpe sin permiso al hogar. La notificación en tiempo real permitiendo dar opciones de control hacia una respuesta ágil ante cualquier situación que se pueda presentar.

En conclusión, este sistema de alarma brinda una solución hacia la seguridad de la vivienda con tecnologías IoT (Internet de las cosas). El sistema de alarma desarrollado da una

protección efectiva contra intrusos, fuga de gas, incendios y otros eventos adversos, brindando la tranquilidad y la capacidad necesaria para proteger su vivienda.

Palabras Claves: Sistema de Alarma, Seguridad, Vivienda, IoT (Internet de las cosas), Sensores, Protocolo MQTT, Clúster.

Abstract

The objective of this project is to provide an alarm system for homes with IoT technologies, designed to prevent theft or a domestic accident, in addition to identifying exceptional events on the state of your home notifying the owner.

An Arduino esp32 was used along with several sensors that serve to detect movement, smoke, fire and gas, these will identify anomalies in the space of the house in which it was installed. The information received by the sensors will be sent to an Android application using the mqtt messaging protocol, which will be responsible for informing the user in real time about any unusual situation detected in the home. In addition, it provides the option for the user to deactivate the alarm and review in detail the current state of the home.

To incorporate this system requires the configuration of a cluster in hiveMQ, allowing to have a private broker who coordinates the messaging between the Arduino and the application in Android. The system guarantees the safety of the house in terms of fire detection, possible fire derived from gas leak, possible fire initiated by fire or when an unidentified person breaks into the home without permission. The ability to notify in real time giving control options towards a quick and effective response to any eventuality.

In conclusion, this alarm system provides an integrated solution for housing security with IoT technologies. The developed alarm system provides effective protection against theft, home accidents and other adverse events, providing homeowners with the peace of mind and the ability to protect their home.

Key Words: Alarm System, Security, Housing, IoT, Sensors, MQTT Protocol, Cluster.

Introducción

Este Proyecto describe el trabajo realizado por los alumnos bajo la dirección del Ingeniero Diego Samir Melo Solarte con el fin de culminar el pregrado en la Universidad de Manizales. Su enfoque consiste en el diseño e implementación de un sistema de alarma para vivienda utilizando tecnologías de IoT (Internet de las Cosas).

Este proyecto se realizó bajo la necesidad que se tiene respecto a mejorar el cuidado del hogar, mejorar la seguridad de la vivienda, debido que en la actualidad existen varios factores que alteran la tranquilidad en el hogar y a su vez es difícil recuperarse de daños irreversibles como es hurto o un incidente en la vivienda. Por ello es necesario tener un sistema que permita detectar este tipo de situaciones en un tiempo corto para aminorar los daños que pueden ocasionar dichas alteraciones. Además, se implementaron medidas de seguridad para proteger la privacidad y la confidencialidad de la información, asegurando un acceso exclusivo para los residentes de la vivienda.

El objetivo principal de este proyecto fue diseñar un sistema de alarma para vivienda con IoT cuyo propósito es tener un monitoreo constante de diversos sensores encargados de procesar la información. Todo lo anterior pretende diseñar este sistema de alarma el cual tenga seguridad, comunicación y control de la vivienda.

1. Área Problemática

Según el Ministerio de Defensa enero del 2022 se identificaron 4866 robos a hogares, pero en enero del 2023 fueron 5583, aumentando en un 15% la cifra del año anterior en Colombia lo cual refleja que la inseguridad a la que se encuentran expuestos los hogares va creciendo y los siguientes años será mayor. Por otro lado, en 2020 se produjeron más de 18.000 incendios en Colombia, de los cuales más de 2.500 fueron estructurales, según datos del Servicio de Bomberos de Colombia. (Ministerio de Defensa Nacional de la Republica de Colombia, 2024).

Gracias a la falta de seguridad en los hogares ya sea por no disponer de un nivel económico óptimo para pagar el servicio de una compañía de seguridad el nivel de hurto se va elevando, los delincuentes implementan diferentes estrategias a la hora de irrumpir, se deben implementar nuevas estrategias para salvaguardar las pertenencias y la seguridad de los residentes.

Las viviendas de los ciudadanos la mayor parte del tiempo se encuentran deshabitadas, estas quedan expuestas a cualquier accidente ya sea por error humano o error de los sistemas de gas o eléctricos del hogar, también por intentos de hurto. los residentes de los hogares deben salir a realizar sus actividades diarias como lo son trabajar y o estudiar sin saber en qué condiciones encontrar el mismo.

¿Cómo beneficiará la implementación de un sistema de alarma con tecnología IoT para la seguridad de las viviendas?

¿Cómo desarrollar un prototipo de sistema de alarma que permita monitorear una vivienda y sea manejada por medio de una aplicación móvil?

2. Objetivos

2.1. Objetivo General

Desarrollar un sistema de alarmas para viviendas, permitiendo detectar situaciones de riesgo o acceso no autorizado en una vivienda.

2.2. Objetivos Específicos

- Diseñar un circuito de un sistema de detección de intrusos y riesgos en una vivienda utilizando dispositivos de bajo costo.
- Establecer el mecanismo de comunicación entre los dispositivos de detección de intrusos o riesgos y los propietarios de la vivienda.
- Implementar una prueba piloto del sistema en la vivienda o en las viviendas de los autores.

3. Justificación

Los sistemas de alarmas para viviendas han surgido de crear una solución tecnológica accesible a los usuarios para abordar este tipo de situaciones. Este tipo de sistemas ofrecen ventajas significativas cuando de la seguridad se habla, debido a que son programas con mayor simplicidad al momento que los usuarios lo utilicen en su vida cotidiana permitiendo una adaptación de cada hogar según las necesidades de cada uno.

Este sistema de alarmas incluye una configuración para notificar al usuario ante cualquier actividad inusual que llegue a detectar brindando la protección y respuesta inmediata a las situaciones de emergencia que se puedan presentar. Teniendo presente que este sistema contará con un acceso exclusivo para los usuarios de la vivienda salvaguardando la privacidad y la seguridad de la información.

Finalmente, proporciona un interfaz de usuario que permita la interacción efectiva con el sistema, esta interfaz facilita la configuración, el monitoreo y el control del sistema de alarmas, garantizando una experiencia de usuario optima y contribuyendo a la simplicidad en su manejo con el sistema de manera efectiva y sencilla logrando esa simplicidad de su manejo.

En resumen, este sistema de alarmas para vivienda en IoT representa una solución eficaz y completa para mejorar la seguridad en los hogares. Permitiendo ofrecer una instalación rápida, comunicación segura y facilidad de uso que para cualquier hogar pueda obtenerlo fortaleciendo la protección de sus habitantes.

4. Marco Teórico

Dentro del marco teórico se explica y detalla los elementos fundamentales para el desarrollo del proyecto, también diferentes tecnologías que se usaron.

4.1. Marco Conceptual

4.1.1. *Arduino Esp32*

ESP32 es la denominación de una familia de chips SoC (System on a chip / Sistema en un Chip) de bajo costo y consumo de energía, con tecnología WiFi y Bluetooth. El ESP32 Dev Kit V4.0 es parte de esta familia amplia; esta tarjeta es un microcontrolador de bajo costo y consumo de energía, cuenta con tecnología Wi-Fi y Bluetooth de modo dual integrada que permite controlar todo tipo de sensores, módulos y actuadores.

Es un módulo que tiene WIFI+BT+BLE para aplicaciones con sensores de baja potencia hasta tareas exigentes, como codificación de voz, transmisión de música y decodificación MP3. Contiene además una corriente de reposo 5uA, velocidad de hasta 150 Mbps y potencia de salida 20dBm. (Carranza, TodoMaker, 2021).

4.1.1.1. Estructura del ESP32. Presentamos ahora la estructura de los pines que se tiene en la tarjeta electrónica ESP32 DEVKIT V4.0. Esta cuenta con 30 pines los cuales tienen distintas funciones, distintos usos y demás. Se puede ver en la Figura 1.

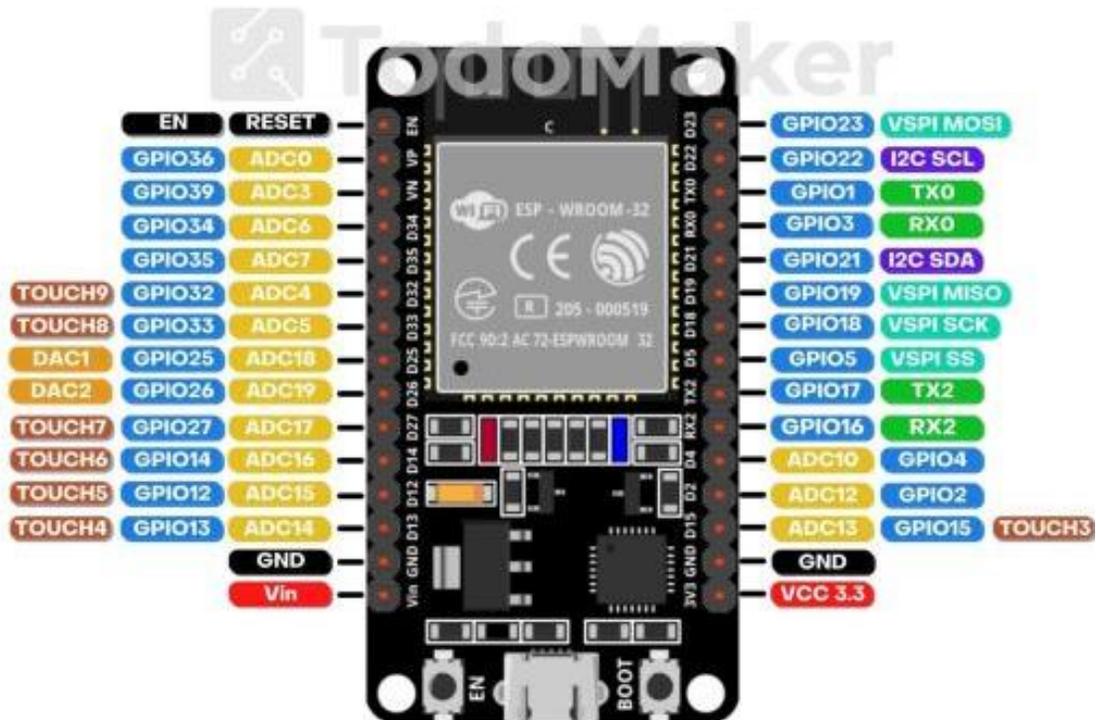


Figura 1. Estructura ESP32.

(Carranza, TodoMaker, 2021).

- 19 canales de convertidor analógico a la digital (ADC)
- 3 interfaces SPI
- 3 interfaces UART
- 2 interfaces I2C
- 16 canales de salida PWM
- 2 convertidores de digital a analógico (DAC)
- 2 interfaces I2S
- 10 GPIO de detección capacitiva

Además, se presenta también el diagrama de bloques de esta tarjeta electrónica. Se puede ver en la Figura 2.

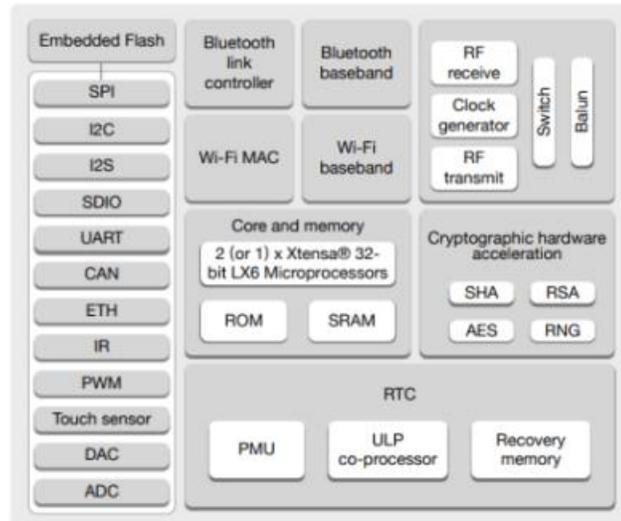


Figura 2. diagrama de bloques de esta tarjeta electrónica.

(Carranza, TodoMaker, 2021).

4.1.1.2. Entorno de desarrollo. IDE – entorno de desarrollo integrado, llamado IDE (Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware a través del puerto serie. (jecrespom, 2016).

4.1.2. Sensores

4.1.2.1. Módulo KY_026 Sensor de Flama. Es un Sensor de Flama que por medio de un LED receptor infrarrojo detecta longitudes de onda de llama en un rango de 760nm a 1100nm.

Las salidas de este sensor son digital y analógica e incluye un potenciómetro para ajuste de la sensibilidad del sensor. Se puede ver en la Figura 3.

El módulo KY-026 Sensor de Flama es útil para sistema de detección de incendios, como una medida de seguridad.

El sensor de llama KY-026 está equipado con un fotodiodo que es sensible al rango espectral de luz creado por una llama abierta. El sensor de llama detecta longitudes de onda que van desde 760 nm hasta 1100 nm en el espectro infrarrojo. Después de detectar una llama, la línea de salida digital (DO) se volverá ALTA. La salida analógica (AO) proporcionará una medición directa de la lectura.

No se recomienda que este dispositivo entre en contacto con una llama, ya que es probable que el plástico se derrita o se quemé. El sensor de llama debe mantenerse a una distancia razonable de la fuente de llama. (Electronics, 2024).

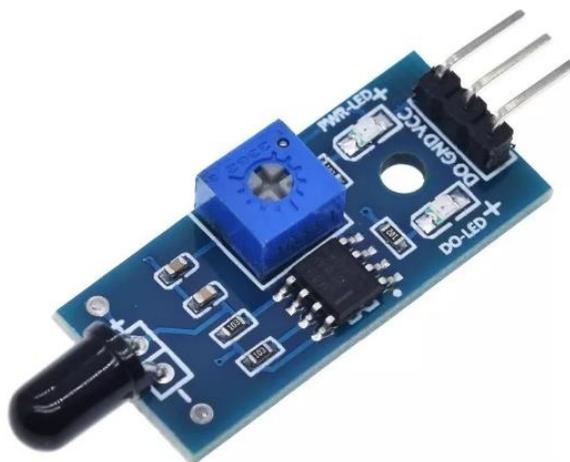


Figura 3. Módulo KY_026 Sensor de Flama.

(Electronics, 2024)

4.1.2.2. Sensor de Gas Aire Mq2. El módulo sensor de gas (MQ2). Detecta GLP, propano, metano, alcohol e hidrogeno. Voltaje de alimentación:5V. Interfaz analógica. El

módulo sensor de gas analógico (MQ-2) se utiliza en la detección de fugas de gas de equipos en los mercados de consumo y la industria, este sensor es adecuado para la detección de GLP, i-butano, propano, metano, alcohol, hidrógeno, tiene una alta sensibilidad, un tiempo de respuesta rápido Y dicha sensibilidad puede ser ajustada por el potenciómetro. Se puede ver en la Figura 4. (Módulo sensor de gas (MQ2), 2024)



Figura 4. Sensor de Gas Aire Mq2.

(Módulo sensor de gas (MQ2), 2024)

4.1.2.3. Sensor Infrarrojo de Movimiento Hc-sr501. El sensor de movimiento infrarrojo es un aparato electrónico creado con el objetivo de medir la radiación electromagnética infrarroja de los cuerpos en su eje de visión.

Todas las entidades contienen cierta cantidad de radiación, esta radiación es invisible a los ojos del humano. Contiene un fototransistor que es el lugar donde entran los rayos infrarrojos. Ahí se encuentra un material piroeléctrico, que en la mayoría de los aparatos está formado por una lámina delgada dentro del nitrato de galio (GaN), nitrato de Cesio (CsNO₃), derivados de la fenilpirazina, y ftalocianina de cobalto. El sensor de movimiento infrarrojo emite una señal al percibir un cambio de temperatura, bien sea a través de la presencia humana o por medio del cambio de temperatura ambiental. Ver figura 5. (admin, 2021)



Figura 5. Sensor Infrarrojo de Movimiento Hc-sr501.

(admin, 2021)

4.1.3. Bróker

Un message broker es un software que permite que las aplicaciones, los sistemas y los servicios se comuniquen entre sí e intercambien información. Para ello, el message broker convierte mensajes entre protocolos de mensajería formal. Esto permite que los servicios interdependientes "hablen" entre sí directamente, incluso si están escritos en diferentes lenguajes o implementados en plataformas distintas.

Los message brokers son módulos de software dentro de las soluciones de middleware de mensajería o middleware orientado a mensajes (MOM). Este tipo de middleware proporciona a los desarrolladores un medio estandarizado de manejo del flujo de datos entre los componentes de una aplicación para que puedan enfocarse en su lógica central. Puede servir como una capa de comunicaciones distribuidas que permite que las aplicaciones que abarcan múltiples plataformas se comuniquen internamente.

Los message brokers pueden validar, almacenar, enrutar y enviar mensajes a los destinos adecuados. Sirven como intermediarios entre otras aplicaciones, permitiendo a los remitentes

emitir mensajes sin saber dónde están los receptores, si están activos o no, o cuál es su número. Esto facilita el desacoplamiento de procesos y servicios en los sistemas.

Para proporcionar un almacenamiento de mensajes confiable y una entrega garantizada, los message brokers normalmente confían en una subestructura o componente denominado cola de mensajes, que almacena y ordena los mensajes hasta que las aplicaciones que los consumen puedan procesarlos. En una cola de mensajes, los mensajes se almacenan en el orden exacto en el que se transmitieron y permanecen en la cola hasta que se confirma la recepción. (IBM, 2023)

4.1.3.1. HivenMQ. MQTT son las siglas de Message Queuing Telemetry Transport. Se trata de un protocolo de mensajería ligero para usar en casos de clientes que necesitan una huella de código pequeña, que están conectados a redes no fiables o con recursos limitados en cuanto al ancho de banda. Se utiliza principalmente para comunicaciones de máquina a máquina (M2M) o conexiones del tipo de Internet de las cosas.

Arquitectura de MQTT

MQTT se ejecuta sobre TCP/IP utilizando una topología PUSH/SUBSCRIBE. En la arquitectura MQTT, existen dos tipos de sistemas: clientes y brókeres. Un bróker es el servidor con el que se comunican los clientes: recibe comunicaciones de unos y se las envía a otros. Los clientes no se comunican directamente entre sí, sino que se conectan con el bróker. Cada cliente puede ser un editor, un suscriptor o ambos.

MQTT es un protocolo controlado por eventos, donde no hay transmisión de datos periódica o continua. Así se mantiene el volumen de transmisión al mínimo. Un cliente sólo publica cuando hay información para enviar, y un bróker sólo envía información a los suscriptores cuando llegan nuevos datos. Ver Figura 6. (¿Que es MQTT?, 2019)

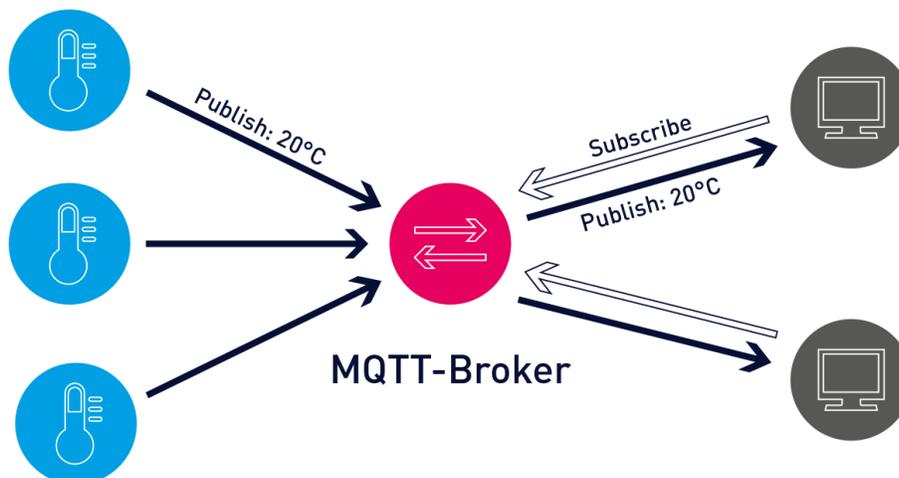


Figura 6. Arquitectura de MQTT.

(¿Que es MQTT?, 2019)

Mensajes en MQTT

Para mantener el protocolo al mínimo, sólo se pueden efectuar cuatro acciones en cualquier comunicación: publicar, suscribirse, cancelar suscripción o hacer ping.

Publicar: envía un bloque de datos que contiene el mensaje que se va a enviar. Estos datos son específicos de cada implementación, pero pueden ser algo tan simple como una indicación de encendido/apagado o un valor de un determinado sensor, como temperatura, presión, etc. En el caso de que el tema que se está publicando no exista, este se crea en el bróker.

Suscribirse: convierte a un cliente en suscriptor de un tema. Se puede suscribir a temas en concreto o mediante comodines, que permiten suscripciones a toda una rama de temas o a parte de ella. Para suscribirse, un cliente envía un paquete SUBSCRIBE y recibe un paquete SUBACK a cambio. Si hay un mensaje retenido para el tema, el nuevo suscriptor también lo recibe.

PING: un cliente puede hacer ping al bróker. El suscriptor envía un paquete PINGREQ y, como respuesta, se recibe un paquete PINGRESP. Se pueden utilizar pings para garantizar que la conexión siga funcionando y que la sesión TCP no haya sido cerrada inesperadamente por otro equipo de red, como un router o una puerta de enlace.

DESCONECTAR: un suscriptor o editor puede enviar un mensaje de DISCONNECT al bróker. Este mensaje informa al bróker de que ya no necesitará enviar o poner en cola mensajes para un suscriptor y que ya no recibirá datos de un editor. Este tipo de cierre permite al cliente volver a conectarse utilizando la misma identidad de cliente que en ocasiones anteriores. Cuando un cliente se desconecta sin enviar un mensaje de desconexión, se envía su última voluntad y testamento a los suscriptores. (¿Que es MQTT?, 2019)

4.1.4. Android Studio

Cuando hablamos de Android Studio, nos referimos a un entorno de desarrollo especializado. Evidentemente, hablamos de todo lo que tenga que ver con el desarrollo de herramientas y apps para sistemas operativos Android.

Para conocer exactamente qué es Android Studio, te traemos el siguiente artículo, en el que recabamos para ti toda la información con respecto a este entorno de desarrollo: desde sus funciones, características, qué tan bueno o potente puede llegar a ser y demás datos de interés.

Normalmente, toda aplicación, herramienta, página, o servidor digital que ofrece algún tipo de tarea en internet, posee lenguajes de programación o entornos de trabajo especializados. Por ejemplo, Python, que es un lenguaje muy utilizado en el desarrollo de Inteligencia Artificial.

Así, tal cual, pasa con el sistema operativo Android. Todas las aplicaciones y herramientas que se desarrollan para este SO en concreto, poseen su propia área o entorno de trabajo. Ese entorno es Android Studio, que permite una flexibilidad en cuanto al desarrollo de características y funciones que puede tener una herramienta o app de dicho sistema.

Este entorno sirve para que las aplicaciones que se estén desarrollando sean mucho más eficiente y autosuficientes. Esto permite, incluso, tener compatibilidades con otros sistemas o plataformas.

Características de Android Studio

Android Studio permite la integración de características y funciones bastante positivas para las aplicaciones que, con el tiempo, se perfeccionan. De esta forma, tenemos lo siguiente:

- El sistema de compilación es flexible, además de ser compatible con Gradle, la cual permite la automatización de compilaciones de forma flexible y con gran
- rendimiento. Groovy y Kotlin DSL son los lenguajes utilizados para los scripts de compilación.
- La intención de este entorno es la de permitir al usuario trabajar de forma fluida y con una gran cantidad de funciones prácticas y útiles.
- Esta plataforma te permite desarrollar aplicaciones para cualquier dispositivo Android.
- Contiene plantillas de compilación que te ayudan a otorgar funciones comunes de otras apps de forma mucho más rápida, además de importar códigos de muestra.
- Mayor cantidad de herramientas de prueba con marcos de trabajo.
- Modificar fragmentos de código y recursos de una app sin necesidad de que esta se reinicie.

- Proporciona compatibilidad con servicios en la nube tal como Google Cloud Platform.
- Compatibilidad con lenguajes como NDK y C++.

Lenguajes que se utiliza para programar en Android Studio

Desde siempre, el sistema operativo de Android se ha desarrollado a través del lenguaje de programación Java. No hay que confundir propiamente Android Studio con un lenguaje, ya que esto es solo el entorno para desarrollar el código Java, para que se puedan crear las aplicaciones propiamente dichas. Sin embargo, como se pudo observar en el punto anterior de las características, esta plataforma de trabajo también puede llegar a ser compatible con lenguajes como Kotlin (uno de los principales), NDK y C++. (Santaella, 2022)

4.1.5. Sistema de Alarma Para Vivienda

4.1.5.1. Sistemas. Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación.

El mundo puede abordarse desde una perspectiva sistemática o sistematicista, en la que todos los objetos forman parte de algún tipo de sistema, desde las partículas de un átomo hasta la corteza cerebral, la democracia representativa o los números enteros. Visto así, un sistema no es otra cosa que un segmento de la realidad que puede estudiarse de manera independiente del resto, pero en el cual sus componentes se hallan interconectados.

Los sistemas son objeto de estudio de la Teoría de Sistemas o Teoría General de Sistemas, una disciplina que los aborda sean cuales sean desde una perspectiva múltiple,

interdisciplinaria. Según ella cualquier sistema es reconocible dados sus límites y partes interrelacionadas e interdependientes (sus llamados subsistemas), a punto tal que la modificación de un elemento modifica necesariamente el funcionamiento del resto del sistema. (Equipo editorial, 2018)

4.1.5.2. Alarma. Una alarma es una señal o aviso que advierte sobre la proximidad de un peligro. El aviso de alarma informa a la comunidad en general o a una entidad específica (como los bomberos o la policía) que deben seguir ciertas instrucciones de emergencia dado que se ha presentado una amenaza.

Por ejemplo: “Creo que se ha producido un incendio cerca de aquí: oigo una alarma”, “La alarma comenzó a sonar y, cinco minutos más tarde, la policía ya se encontraba en el domicilio de la víctima, “La gente del pueblo se alborotó cuando la alarma que advierte sobre la inminencia de un tsunami entonó sus primeras notas”.

Las alarmas que constituyen los organismos que deben responder ante una emergencia, suelen formar parte de un sistema que incluye diversos estados. El primero de ellos es la prealerta, que avisa a los responsables del organismo sobre un incidente que puede tener lugar. El segundo estado es la alerta, que exige tomar las medidas y los recaudos necesarios. Finalmente llega la alarma, que es el llamado a la acción. (Julián Pérez Porto, & Gardey, 2010).

4.1.5.3. Vivienda. La vivienda está presente en la cotidianidad de la vida de las personas, es el lugar donde se lleva a cabo la gran mayoría de las actividades básicas de la vida diaria, es donde se duerme, se come, se guardan pertenencias, y el lugar donde se regresa al final de la jornada. Sin embargo, la vivienda tiene un significado psicológico profundo más allá del

puramente instrumental de cobijo y lugar donde se desempeñan las conductas domésticas.

(Pasca, 2013)

4.1.5.4. Sistema de Alarma. En las casas y los edificios, ya sean residenciales, comerciales o gubernamentales, el sistema de alarma es un elemento de seguridad pasiva (no puede evitar una situación de riesgo, pero advierte de ella y da aviso a las fuerzas de seguridad). Algunos sistemas pueden advertir de un peligro de forma automática (al detectar una intrusión por la noche a través de sensores de movimiento), mientras que otros requieren de una acción por parte del usuario (cuando se inicia un robo y un empleado activa la alarma para llamar a la policía).

Por lo general, los sistemas de alarma modernos cuentan con un panel numérico que debe utilizar el dueño de la propiedad protegida para crear una o más claves de seguridad, que deberán ser ingresadas en diversas situaciones bien definidas, como ser al entrar al inmueble y antes de abandonarlo. La detección de movimiento, que permite advertir automáticamente la actividad indebida en un edificio, no siempre resulta la opción adecuada, dado su grado de imprecisión.

Si bien los sensores de movimiento pueden ser calibrados para que ignoren un cierto rango de cambios en el ambiente, tales como las sombras de los muebles y las plantas proyectadas por las luces que provienen de la calle o el vuelo de un insecto, es imposible prevenir ciertos incidentes que requieran de acciones poco regulares por parte de los mismos propietarios. En dichos casos, a veces emergencias, la alarma se convierte en una molestia, en un obstáculo hacia la resolución del problema. (Julián Pérez Porto, & Gardey, 2010).

4.1.6. Fritzing

Fritzing no es más que un programa de automatización de diseño electrónico «gratuito» (esto lo explicaremos al final), diseñado para ayudar a los diseñadores y artistas en la transición de prototipos (utilizando, por ejemplo, placas de prueba) a productos finales. Fritzing se creó basándose en los principios de Processing y Arduino, lo que permite a los diseñadores, artistas, investigadores y aficionados registrar sus prototipos basados en Arduino y crear diagramas de circuitos impresos para su posterior fabricación. Además, tiene un sitio web complementario que puede ayudar a compartir y discutir bocetos y experiencias y reducir los costos de fabricación. (Perera, 2021)

4.2. Antecedentes

“Desde los inicios del siglo XIX la presencia de la electrónica y la automática en la vida cotidiana de las personas ha aumentado de manera exponencial, es difícil imaginar cualquier utensilio tradicional que no haya evolucionado hacia un producto de los que ahora denominamos inteligentes y que abarcan cualquier tipo de mercado, desde aspiradores, frigoríficos o lavavajillas hasta una simple bombilla. En este trabajo se aplicarán estos avances en el ámbito de la seguridad en el hogar y de la domótica. A día de hoy, la seguridad en el hogar es un asunto que preocupa a muchas personas, las cuales recurren a empresas especializadas en prestar servicios de vigilancia a cambio de pagar una determinada cuota mensual, sin embargo existen otras alternativas a estos métodos, que son las que vamos a tratar de encontrar en este trabajo, para ello estudiaremos los tipos de sistemas de alarma existentes, extrayendo de estos las características más interesantes, las cuales usaremos para diseñar un prototipo propio de sistema de seguridad con funciones de alarma, e incluso una serie de funcionalidades añadidas las

cuales son de utilidad para el usuario.

En este trabajo se va a hablar también de la domótica, la cual está íntimamente relacionada con el sistema que vamos a construir, nos adentraremos en la teoría de los sistemas domóticos, así como su aplicación práctica en el prototipo.

El prototipo será construido sobre un microprocesador el cual será el elemento central, debido a su versatilidad a la hora de programarlo es la opción más adecuada teniendo en cuenta que podemos añadir a este los periféricos que queramos para conseguir unas funciones determinadas como se explicará en los posteriores apartados. Dedicaremos también un capítulo para profundizar en el estudio de estos dispositivos.” (Blanco Huelmos, 2023)

“El objetivo de este proyecto es el diseño de un sistema de alarma, dicha instalación va a permitir controlar de forma remota y en tiempo real a través de la red IP una casa o pequeña empresa, la seguridad de las personas y de los bienes valiosos de su interior. El sistema ha sido pensado para reducir al máximo el cableado y de ésta manera facilitar la instalación, por lo que la comunicación entre los distintos dispositivos se hace de manera inalámbrica por medio de un protocolo descrito en la norma IEEE 802.15.4 llamado ZigBee. La red estará compuesta por varios sensores encargados de medir las variables del entorno como movimiento, gases, humo, vibración y temperatura, cada sensor se conectará a una placa Arduino encargada de procesar, transformar y transmitir dicha información al módulo Xbee. Para integrar dicho sistema se propone una placa de bajo costo llamado RaspBerryPi, en el que se encuentra alojado un servidor el cuál realizará el proceso de control y envío de los datos ya procesados al usuario a través de una aplicación para dispositivos móviles y computadores.” (Alexandra, 2015)

“En la actualidad, los robos a las viviendas no han hecho sino aumentar. Los ladrones se dan modos para ingresar a los domicilios de manera no autorizada, saqueando todo lo que pueden. Una buena solución sería contar con un Sistema de Alarma antirrobo que reducirá de manera considerable el riesgo de sufrir un robo en nuestro hogar. Con este Sistema, ya se puede prevenir de un robo o ingreso no autorizado. En el presente Trabajo de Aplicación se plantea una solución, la implementación de un Sistema de Alarma antirrobo para el hogar utilizando la plataforma Arduino, que lo hace un Sistema económico, fácil de armar y configurar. Este Sistema de Alarma antirrobo para el hogar contará con la plataforma Arduino UNO como central de Alarma, dos sensores de movimiento PIR que detectarán justamente el ingreso de una persona al domicilio, contará con un teclado y display LCD, que son periféricos que nos ayudarán a Configurar la Central de Alarma y ver los estados de configuración. También contará con una Sirena de Alarma para que, al sonar, cumpla con la función de avisar del posible robo.” (Patricio, 2021)

“Un sistema integral de seguridad de un objetivo, en nuestro caso una vivienda unifamiliar, podemos definirlo como el conjunto de elementos y sistemas de carácter físico y electrónico que, junto con la adecuada vigilancia humana, proporcionan un resultado armónico de seguridad relacionado directamente con el riesgo potencial que soporta. En el presente proyecto vamos a centrarnos en el desarrollo de los medios técnicos activos o de seguridad electrónica. De los numerosos agentes externos causantes de daños o pérdidas (naturales, químicos, antisociales, etc.) analizaremos sólo aquellos que van dirigidos contra los bienes y el patrimonio de forma intencionada, así como los causantes de incendios. El objeto del presente proyecto es el de desarrollar un medio técnico activo electrónico que ayude a mejorar la seguridad de un hogar, así

como su instalación y puesta en funcionamiento además de instruir en su manejo.” (Antonio, 2024)

“El internet de las cosas (IoT) se ha convertido en una parte integral de nuestra vida cotidiana, y está expandiéndose aún más a medida que el número de dispositivos inteligentes se incrementa. Esto plantea una serie de desafíos para la seguridad, ya que los dispositivos IoT se conectan a la red y comparten datos. Para abordar estos desafíos, un equipo de investigadores desarrolló un sistema de seguridad para viviendas urbanas basado en IoT. El sistema utiliza sensores inalámbricos para recopilar datos y almacenarlos en la nube. Además, una red inalámbrica de comunicación de datos se utiliza para transmitir datos de los sensores a un centro de control remoto. El sistema incluye un sistema de alerta que se puede configurar para enviar alertas al usuario si se detecta algún tipo de actividad anómala. El sistema también incluye un mecanismo de autenticación para garantizar que sólo los usuarios autorizados puedan acceder a los datos. Por último, el sistema proporciona una interfaz de usuario para que los usuarios puedan interactuar con el sistema. En conclusión, este sistema de seguridad para viviendas urbanas con IoT es una solución eficaz para mejorar la seguridad en entornos domésticos. El objetivo de este análisis es desarrollar un sistema de seguridad para viviendas urbanas conectadas a Internet de las Cosas (IoT). El sistema de seguridad consta de una variedad de dispositivos conectados a la red de IoT, como sensores de movimiento, cámaras de video vigilancia, sensores de presencia de humo y de CO, detectores de intrusos, sistemas de alarma, etc. Estos dispositivos se conectarán a una plataforma en la nube que estará configurada para recibir notificaciones en caso de detectar algún evento anómalo. La plataforma contará con un sistema de autenticación para proporcionar una seguridad adecuada y permitirá a los usuarios

acceder a los dispositivos desde cualquier lugar. El sistema también se diseñará para comprobar la integridad de los dispositivos conectados y permitir a los usuarios controlar el acceso a sus dispositivos. Se incluirá una aplicación móvil para permitir a los usuarios configurar y controlar los dispositivos de forma remota.” (Saa Ayala, 2023)

“En la actualidad los avances tecnológicos han permitido el desarrollo de productos y servicios que interactúan con personas y los objetos que los rodean. Estas tecnologías se orientan a simplificar y facilitar cuestiones de la vida diaria, sin embargo, en algunos casos, presentan una barrera. Especialmente para las personas de avanzada edad. Esta línea de investigación intenta encontrar una forma de derribar esas barreras tomando ventaja de las tecnologías actuales como sistemas embebidos con sensores de movimiento, sensores biométricos, conexión inalámbrica, geoposicionamiento, etc. Haciendo uso de estos dispositivos y de la computación en la nube, se busca brindar a las personas adultas mayores y de su entorno mayor comodidad, autonomía, independencia, reducción de costos en los cuidados, mejores controles de la salud y mayor celeridad en la respuesta ante emergencias. Se busca crear una solución que monitoree la salud del usuario y la reporte a los cuidadores, familiares y médicos a través de internet durante las 24hs los 7 días de la semana. Además, emite alertas en los casos que el usuario requiera atención inmediata.” (Barillaro, 2016).

“En este trabajo se detallan características de un sistema de automatización de una vivienda con el internet de las cosas (IoT), para tener control de cualquier objeto cotidiano que esté conectado a internet. La implementación de este proyecto se realizará mediante una aplicación cliente servidor, donde el servidor controlará el sistema ya sea de iluminación o activación de algún

equipo electrónico. La unidad de control de este sistema estará implementada en base a un NodeMCU ESP8266, que es parte principal de todos los procesos de control que se realizan, estará conectado a una red a través del WiFi, la información se la puede gestionar a través de una computadora, Tablet o celular, conectado a Internet, ya sea desde la aplicación o desde un navegador que esté. El sistema de control será centralizado, es decir que el controlador será el encargado de recepcionar las solicitudes del cliente y generar las órdenes que ejecutaran los actuadores. Y ya que es un proyecto con IoT necesariamente todos los dispositivos deben estar conectados a Internet.” (Rosas Cruz, 2021)

5. Metodología

La metodología que se desarrolló en el sistema de alarma para vivienda con IoT fue desde la planificación hasta la implementación y realización de pruebas para garantizar su eficacia y fiabilidad. La metodología utilizada es de tipo desarrollo tecnológico permitiendo usar diversas tecnologías.

El desarrollo propuesto, establece un sistema para el cuidado de la vivienda permitiendo tener una mejor seguridad y cuidado de las mismas.

5.1. Procedimiento

El proyecto estuvo presente en 3 fases principalmente, las cuales son las siguientes:

5.1.1. Fase 1. Diseñar el circuito de un sistema de detección de intrusos y riesgos en una vivienda utilizando dispositivos de bajo costo.

En esta fase se realizó un estudio de cada una de las variables que el sistema requiere, y a su vez también ver los costos de los implementos tecnológicos que se usarán; se realiza este estudio con el objetivo de poder brindar un sistema rentable para las viviendas.

Al cotizar los dispositivos necesarios para realizar este sistema de seguridad para vivienda, que los dispositivos fueran de bajo costo se tomó las siguientes decisiones de compra.

- **Arduino ESP32.** La placa de desarrollo ESP32 es más económica que la placa Arduino Uno. Esta placa es potente por un precio menor, adicional pesé a que tiene una arquitectura diferente a las de las placas arduino llega a soportar el framework de Arduino IDE, para realizar su configuración se realiza directamente en el framework por lo que programar su funcionalidad

se facilita a la hora de realizar el sistema por lo cual es una buena elección al momento de costos y funcionalidad.

- **Sensor Infrarrojo Llama Flama Ky026.** El sensor de flama KY026 se utiliza para la detección de fuego, enfocado en detectar longitudes de onda específicas entre 760nm y 110nm, frecuencias a las que se encuentran la mayoría de las llamas e incendios, incluso puede detectar algún tipo de luz que se emita con demasiada potencia. Este sensor trae un potenciómetro para cambiar la sensibilidad del sensor de llama, se ajustó según la necesidad del sistema, en este caso este sensor es ajustado y cuando entra a un valor igual a 0 genero la activación del sensor a un valor “True” por el resto del tiempo este sensor tuvo un valor superior lo cual hizo referencia a un valor inicial sin alteraciones catalogado como “False”. Ver Figura 7

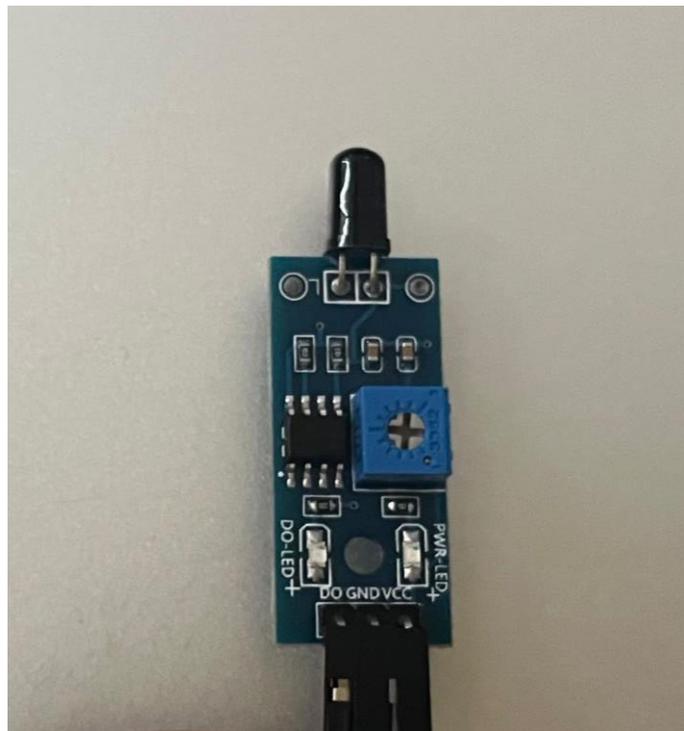


Figura 7. Sensor Infrarrojo Llama Flama Ky026

- **Sensor De Gas Aire Mq2.** El sensor MQ2 es un sensor de gas analógico que se utilizó para la detección de fugas de gas. Es idóneo para la detección de gas combustible y humo

concentrado por lo cual es la opción adecuada para el desarrollo de este sistema teniendo presente que por su alta sensibilidad causa que tenga un tiempo de respuesta rápida. Este sensor tiene una salida digital que es donde se calibra el potenciómetro, la salida del sensor cambio de acuerdo a la concentración de gas en el aire, en este proyecto se manejó un umbral del sensor de gas de 1100 teniendo presente que cuando el gas sea mayor a este valor su estado será “True” por lo cual la alarma será activada de lo contrario mientras el sensor este dentro del umbral tendrá un estado inicial y será “False”. Ver Figura 8 y Figura 9

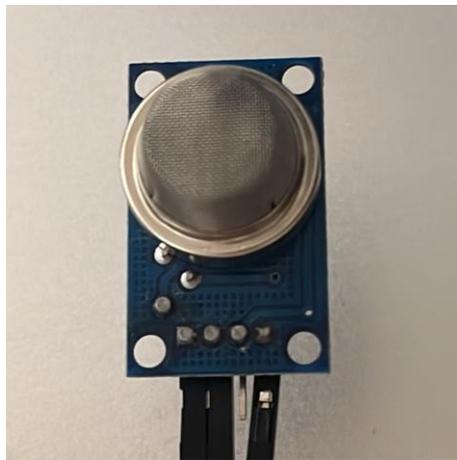


Figura 8. Sensor de Aire Gas MQ2 Frontal

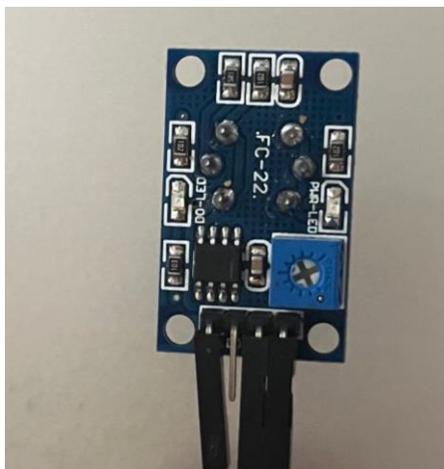


Figura 9. Sensor de Aire Gas MQ2 Posterior

- **Sensor Infrarrojo De Movimiento Hc-sr501.** Este Sensor infrarrojo de movimiento detecta la presencia de un cuerpo cuando este genera algún movimiento, tiene un campo amplio de detección, este módulo tiene la posibilidad de seleccionar el tiempo en que permanece activa por lo cual permitio un control del sistema de alarma en ese modulo. Este sensor es bastante sencillo y funciona de inmediato se necesitó una alimentación de 5V- 12V y tierra, la salida del sensor cuando es igual a HIGH se detecta movimiento pasa a un estado “True” y cuando sea igual a LOW se detecta como un estado inicial “False”; también conto con un potenciómetro para ajustar la sensibilidad el cual estableció el rango máximo de detección. Se ajustó aproximadamente 3m a 7m sin embargo se tuvo presente que al girar el potenciómetro en el sentido de las agujas del reloj aumento su sensibilidad y alcance. Ver Figura 10



Figura 10. Sensor Infrarrojo De Movimiento Hc-sr501.

Para realizar el diseño del sistema se usó un programa de automatización de diseño Fritzing Ver en la figura 11.

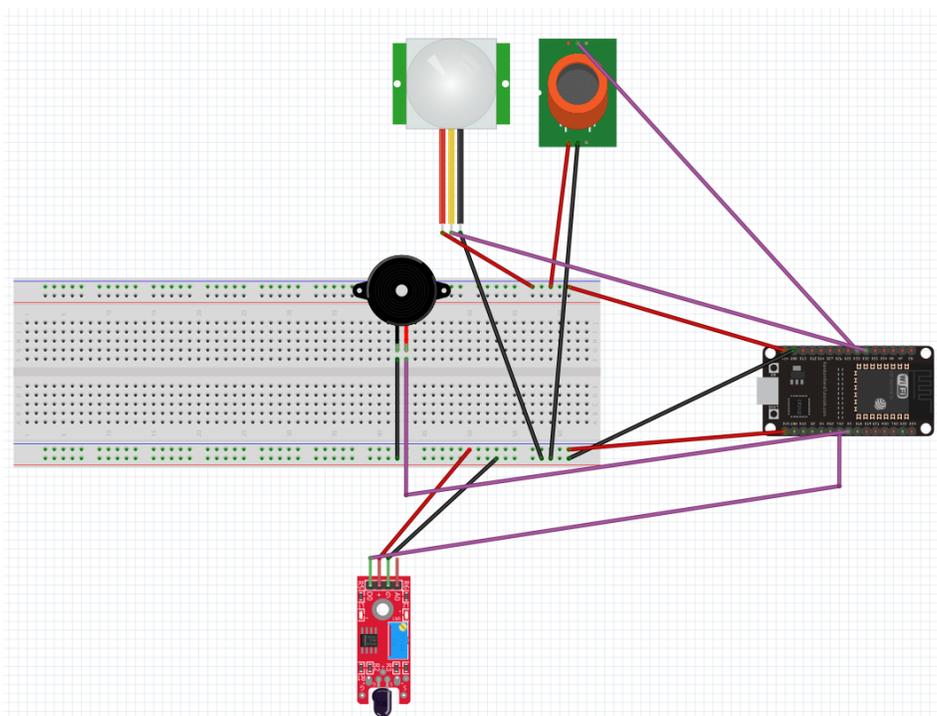


Figura 11. Diseño de Sistema de Alarma

5.1.2. Fase 2. Establecer el mecanismo de comunicación entre los dispositivos de detección de intrusos o riesgos y los propietarios de la vivienda.

Se estableció un mecanismo de comunicación que permita la interacción entre los dispositivos de detección de intrusos o riesgos y el propietario de la vivienda. Este proceso es fundamental para garantizar una respuesta rápida y efectiva ante cualquier situación que pueda surgir.

Se realizó un análisis exhaustivo de las tecnologías de comunicaciones disponibles para determinar cuál será la mejor opción, adaptándose a las necesidades que el sistema de alarma requiere. Como primera instancia se debe asegurar que los dispositivos tengan una buena compatibilidad y sincronización. Esto involucra la configuración de cada sensor de movimiento, sensor de gas y sensor de fuego que se utilizara en este sistema de alarma para vivienda que pueda enviar alertas de manera afectiva. Para ello, la configuración de dichos sensores se realiza

en el framework Arduino IDE. Los datos técnicos se encuentran en los documentos anexos.

(Anexo A).

En este sistema de alarma se realizó un protocolo de comunicación el cual permita la transmisión de datos entre los dispositivos de detección y los propietarios de la vivienda. Estos protocolos deberán garantizar la confidencialidad e integridad de la información transmitida es por ello que se usara un software que permite que las aplicaciones, sistemas y servicios se comuniquen entre si e intercambien información este software toma el nombre de bróker.

El bróker que se utilizó en este proyecto será HivenMQ permitiendo un protocolo de mensajería ligero. Un bróker es el servidor con el que se comunican los clientes, recibe comunicación de unos y envía a otros. Se utilizará para poder activar la alarma y así mismo una vez este activa puede ser apagada por medio de una interfaz gráfica.

La interfaz gráfica se desarrolló en Android Studio la cual permitió al propietario de la vivienda tener un acceso más amigable al sistema, para el diseño de esta interfaz se utilizará Figma la cual es una herramienta para diseñar páginas web y aplicaciones (Figura 12). Se cuenta con esta interfaz con el objetivo de crear un sistema de alarma más completo y seguro. El lenguaje de programación java fue el que se utilizó para el desarrollo de esta interfaz.

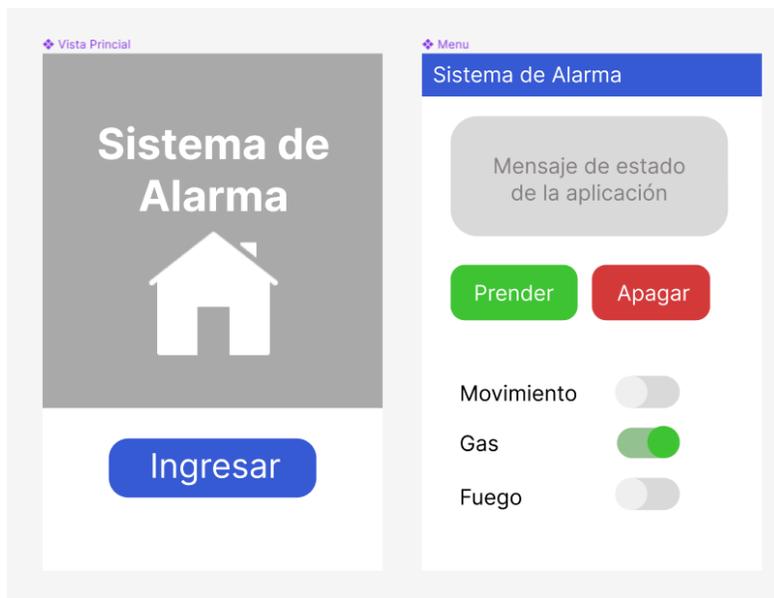


Figura 12. Diseño de la aplicación de Sistema de Alarma

Para el protocolo de comunicación se configuró tanto el framework Arduino IDE como el entorno de desarrollo Android Studio generando así la posibilidad de enviar y recibir respuesta desde el bróker. Los datos técnicos se encuentran en los documentos anexos. **(Anexo B)**.

Así mismo por medio de la aplicación móvil cuando se desee encender el sistema de alarma o apagar se enviará un mensaje al bróker para que permitan realizar la acción, también tendrá presente cuando no hay ningún tipo de estímulo que active la alarma para que esta permanezca en su estado inicial. Los datos técnicos se encuentran en los documentos anexos. **(Anexo C)**

5.1.3. Fase 3. Implementar una prueba piloto del sistema en la vivienda o en las viviendas de los autores.

Durante el transcurso de la implementación del proyecto se realizó una prueba piloto entre el correcto funcionamiento de los sensores y la comunicación del sistema como el aplicativo, estas pruebas han sido realizadas desde la interfaz de Android, con su producto físico terminado.

Para realizar las pruebas el Sistema de Alarma ha sido instalada en una vivienda durante aproximadamente 720 horas o un mes. Durante este tiempo se ha realizado continuamente pruebas de las diferentes detecciones que el sistema obtiene, como es realizar simulaciones de fuga de gas, incendios e incluso robo. Como resultado de estas pruebas se han corregido pequeños errores ya sea en el sistema como en su interfaz Android logrando que sea un Sistema de Alarma eficaz y fiable para el usuario.

Se simulo una fuga de gas en varios escenarios donde se pudo poner a prueba su funcionalidad; uno de los escenarios donde se probó fue directamente con el gas de la estufa y con una espera corta teniendo presente que por precaución se evitó una fuga de gas real que ocasionara un accidente, en este se evidencio como el sensor al sentir la concentración de monóxido de carbono en el aire genera la activación de la alarma de inmediato enviando al servidor un JSON indicando “Gas: True” que hace referencia a que el sensor de gas se encuentra activo y está generando una detección de fuga de gas, esta alerta llego al aplicativo móvil y por medio de una notificación se alertó lo que ocurrió, también al entrar en la aplicación móvil se vio reflejado el mensaje de alarma activa y el sensor que se activó. Los datos técnicos se encuentran en los documentos anexos. **(Anexo D)**.

Así mismo, se simulo un incendio el cual se realizó en varios escenarios, y donde más se puedo evidenciar la efectividad del sensor al detectar un incendio fue generando llama con velas directamente debajo del sensor pues al detectar incendio en una corta distancia genera la activación de la alarma inmediatamente enviando al servidor un JSON indicando “Llama: True” que hace referencia a que el sensor de llama se encuentra activo y está detectando un incendio, lo cual genera la alerta al aplicativo sobre un posible incendio y por medio de una notificación le indica al usuario lo que está ocurriendo, así mismo al entrar en la aplicación móvil se podrá ver

el mensaje de alarma activa y el sensor que fue activado. Los datos técnicos se encuentran en los documentos anexos. (**Anexo E**). También en una de las pruebas donde se simuló un incendio se realizó con candela y este sensor fue activado pero la candela también cuenta con gas propano y butano activando al tiempo el sensor de gas, pese a que esta prueba nos sirvió para la correcta funcionalidad de los dos sensores (Sensor de gas y Sensor de llama) se quiso en estas pruebas probar únicamente el sensor de llama, pues el objetivo también fue mostrar su función individual.

En las pruebas anteriores donde se probó gas e llama se tuvo desactivado el sensor de movimiento debido que al sentir una presencia de un cuerpo cerca ocasionaba su activación, como lo indicaba anteriormente se realizó esto con el objetivo de poder hacer las pruebas debidas de cada sensor para comprobar que su configuración y programación fueran las adecuadas. También la alarma activa constantemente por el sensor de movimiento no permitía centrar la concentración en poder hacer una correcta verificación de los sensores probados anteriormente mencionados.

Por último se realizó la prueba de detección de intrusos la cual en los escenarios que fue probada se debió salir de la vivienda y activar la alarma por medio de la aplicación móvil, pues es importante tener presente que una vez activa la alarma cualquier movimiento que se genere el sensor será activado por ende la alarma en sí, uno de los escenarios que se realizó fue poner el sistema de alarma en la habitación de uno de los autores ubicándola adecuadamente como una simulación real, hay que tener en cuenta que este sistema de alarma debe estar conectado por lo que como prototipo inicial no cuenta con batería, sin embargo, al tenerlo conectado en un puerto de carga (toma corriente, enchufe) cuenta con la energía necesaria para su funcionamiento. Una vez ubicado el sistema se procede a salir de la vivienda, apenas se sale de la vivienda por medio de la

aplicación móvil se activa la alarma, y luego de un tiempo se procede a entrar a la misma, en ese momento la alarma se activa inmediatamente y es ahí donde se pudo evidenciar como el sensor al sentir el movimiento por medio de la radiación infrarroja genera la activación de la alarma inmediatamente enviando al servidor un JSON indicando “Movimiento: True” que hace referencia a que el sensor de movimiento se encuentra activo y está detectando un intruso, lo cual genera la alerta al aplicativo sobre un posible intruso y por medio de una notificación le indica al usuario lo que está ocurriendo, así mismo al entrar en la aplicación móvil se podrá ver el mensaje de alarma activa y el sensor que fue activado. . Los datos técnicos se encuentran en los documentos anexos. (**Anexo F**).

6. Resultados

6.1. Descripción de Resultados

Se desarrolló un sistema de alarma el cual se activa dependiendo las anomalías identificadas en la vivienda; estas pueden ser: fuego, fugas de gas o movimiento, las mencionadas son recibidas por una caja creada con fines de seguridad y procede a dar la orden de activar una bocina para anunciar la situación de la vivienda.

Se codificó un sistema de detección de intrusos el cual mediante sensores de movimiento infrarrojo identifican la variación de temperatura en el ambiente permitiendo validar el ingreso de personal no autorizado a la vivienda informando al sistema de alarma y realizando la notificación a la aplicación creada y posteriormente al propietario de la vivienda.

Dentro de lo planteado se crea a su vez un sistema de detección de fuga de gas el cual está compuesto por un sensor mq-2, el cual posee un material sensible (SN02) que modifica la conductividad al haber presencia de estos gases, donde un circuito convierte este cambio de conductividad para así obtener la señal de salida analógica identificando la calidad del aire interno de la vivienda y así evalúa la situación informando al sistema de alarma y notificando al propietario de la vivienda en el aplicativo mencionado anteriormente.

Se formuló un sistema de detección de incendio el cual mediante modulos Ky026 por medio de un LED receptor infrarrojo detecta longitudes de onda de llama en un rango de 760nm a 1100nm, brindando una salida digital determinando si en la vivienda hay alguna emergencia a base de fuego e informará al sistema de alarma y notificará al propietario de la vivienda.

El sistema de seguridad fue desarrollado en código arduino el cual capturo la salida de los módulos digitales y analógicos, validando si existen variaciones en los mismos, activando el

sistema de alarma y publicando la información de los mismo en el broker para ser enviados a la aplicación móvil.

Se formalizó una aplicación Android la cual se encargó de brindar la experiencia máquina usuario dando la posibilidad de recibir información del sistema de seguridad, exponiendo la información al usuario, notificando de forma digital y permitiéndole al usuario desactivar el sistema de alarma.

Se implementaron módulos de bajos recursos funcionales los cuales pueden ser reemplazados fácilmente en caso de alguna falla.

Se diseñó un modelo 3d el cual fue impreso y contiene el circuito hecho a la medida para dar portabilidad y facilidad de uso al mismo.

Se usó el prototipo durante unas semanas para comprobar la validez del mismo, dando los resultados esperados, ya que cuando se montaba el ambiente no deseado este respondía sin ningún problema.

Para los ambientes no deseados se presentaron las siguientes situaciones:

-Cuando se presentaba el uso de un cilindro de gas propano el cual suministraba gas al ambiente, el sistema capturaba el cambio de los valores en el ambiente y publicaba mediante un json la información en el broker.

-Al presentarle el uso de un encendedor para simular la llama, el sistema tomaba la longitud de onda y publicaba en el mismo json la información en el broker.

-Cuando algún supuesto intruso accedió al área en la cual se encontraba el dispositivo este tomaba la información cada 5 segundos, confirmando lo detectado y anexando la información al json de salida.

El Json tiene las propiedades de cada uno de los factores presentados así informando las 3 situaciones al tiempo.

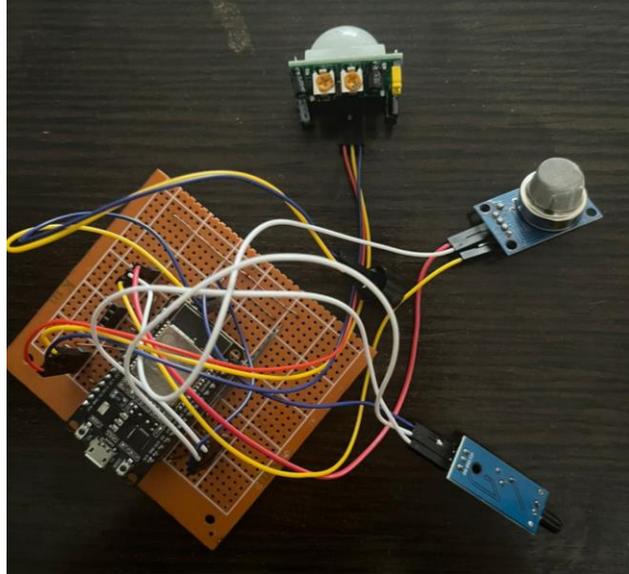


Figura 13. Circuito del sistema de alarma



Figura 14. Prototipo de sistema de alarma

6.2. Discusión de Resultados

En este proyecto , hemos diseñado y desarrollado un sistema de seguridad de vivienda para brindar más seguridad a la hora de dejar el hogar o al momento de estar descansando, a diferencia de los antecedentes del proyecto se ha desarrollado un sistema portable, en los

otros proyectos usan sistemas los cuales deben ser implementados por algún profesional ya que usan sensores y alarmas que deban ser instalados sitios específicos, El sistema de este proyecto puede ser utilizado en diversas ubicaciones como un hostel, hotel, apartamento y una casa. Este sistema permite desactivar y activar la alarma independientemente del lugar donde se encuentre el propietario, todo esto gracias a la comunicación entre arduino y el móvil mediante IoT recibiendo notificaciones sobre la seguridad de la misma a través de una aplicación propia, sin necesidad de desplazarse de su sitio, además mediante el uso de estas diferentes viviendas se han identificado un significativo aumento de seguridad, brindando paz.

Tabla 1.. Tabla de Presupuesto

PRESUPUESTO INICIAL			
Cantidad	Implementos	Valor unitario	Valor total
2	Buzzer	\$1.800	\$3.600
1	Arduino esp32	\$28.000	\$28.000
1	Sensor Infrarrojo Llama Flama Ky026 5cm 760-1100nm Arduino	\$14.000	\$14.000
1	Sensor De Gas Aire Mq2 Propano Metano Alcohol	\$16.300	\$16.300
2	Sensor infrarojo	\$5.500	\$11.000
1	Carcasa	\$50.000	\$50.000
TOTAL PRECIO PROYECTO			\$122.900

7. Conclusiones

Se realizó una amplia búsqueda de información acerca de los proyectos que tuvieran similitud con el presente proyecto, lo cual facilitó la elaboración y el desarrollo del sistema de alarma para vivienda.

Se desarrolló el software utilizando el entorno IDE de Arduino para cumplir con la función principal del Sistema de Alarma.

Al usar el sistema en diferentes hogares como se comentó en los resultados se comprobaron que se puede implementar un sistema de alarma para vivienda con IoT a bajo costo en comparación con algunos sistemas de seguridad que se encuentran en el mercado.

La investigación que se realizó para realizar el proyecto fue muy enriquecedora en cuanto a la obtención de conocimiento en áreas como la telecomunicación y la programación, de igual forma en la planeación y desarrollo de un proyecto de alta complejidad.

La configuración del bróker permite agregar nuevos módulos de seguridad al sistema.

El sistema puede ser utilizado en cualquier hogar desde que haya wifi.

El ser notificado de una situación de riesgo permite tomar medidas de forma inmediata.

Para construir un buen sistema de seguridad se requiere una buena elección de sensores y módulos.

En conclusión, se cumple el objetivo general de desarrollar un sistema de alarmas para viviendas, permitiendo detectar situaciones de riesgo o acceso no autorizado en una vivienda.

8. Recomendaciones

Tal como se ha expresado a lo largo del documento el propósito es diseñar un sistema de alarma para vivienda con tecnologías IoT de bajo costo para lo cual es necesario y recomendado tener en cuenta puntos tales como:

Se recomienda implementar video seguridad en la app como futura mejora al proyecto.

-Se recomienda la integración de la app en la app store para futuras mejoras.

-Implementar login al aplicativo móvil.

-Comprobar cada 6 meses el buen comportamiento de los sensores.

-Modificar el diseño para que permita integrar más sensores alcanzando todos los ángulos.

-Integrar una batería que permita ser recargada: para la integración de batería se deberá hacer con el módulo TP4056. “Se trata de un cargador programable lineal de corriente y tensión constante para baterías LiPo de célula única. El módulo además integra dos indicadores LED del estado de carga y dos chips adicionales encargados de la protección de la batería.

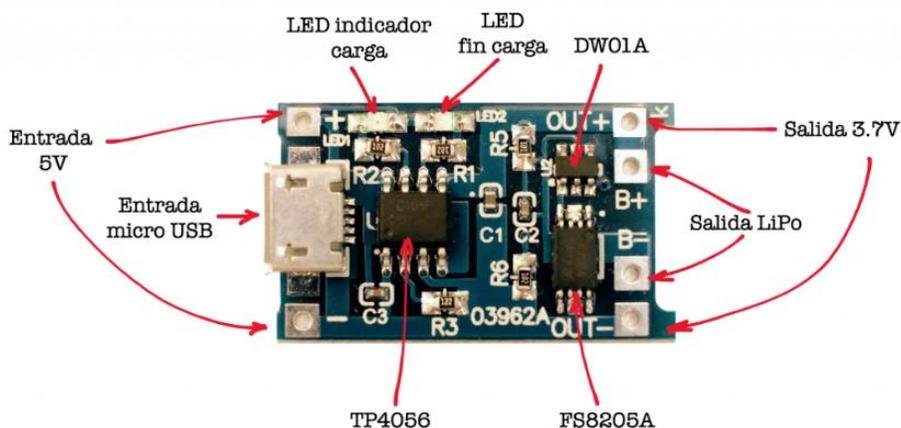


Figura 15. Módulo TP4056

El módulo facilita la realización del proceso de carga. Primero asegura una corriente constante hasta que la batería LiPo llega a los 4.2V. Después va reduciendo la intensidad de carga paulatinamente hasta completar la carga de la batería y finaliza el ciclo de carga cuando la corriente de carga desciende a una décima parte del valor programado. Soporta tensiones de entrada de hasta 8V por lo que puede utilizarse conectado directamente a un puerto USB o un cargador móvil de hasta 1.2A.

Piezas necesarias

A parte del módulo TP4056, necesitamos obviamente una batería LiPo y una placa Arduino o un clon. Además, como normalmente las placas Arduino trabajan con un voltaje de 5V es necesario añadir un módulo regulador de voltaje que aumente el voltaje de la batería LiPo.

Tabla 2. Tabla Piezas necesarias para incluir batería

Componente	Amazon.es	AliExpress
Módulo TP4056	(10 uds) <u>8,95 €</u>	<u>0,31 €</u>
Módulo step-up 5V	<u>7,29 €</u> (5 uds.)	<u>0,38 €</u>
Batería LiPo ~2000 mAh	<u>14,29 €</u>	<u>6,98 €</u>
Arduino Nano	(original) <u>14,40 €</u>	(clon) <u>2,93 €</u>
Interruptor	(50 uds.) <u>7,99 €</u>	<u>0,07 €</u>
Cable AWG 26 (-0.4 mm)	(6x10m) <u>17,89 €</u>	(6x10m) <u>10,60 €</u>

Esquema de conexión

Por suerte la conexión es bastante sencilla. Tan solo debemos conectar la batería y el módulo step-up junto con el interruptor al módulo TP4056. Después solo quedará conectar la salida del módulo step-up a la placa Arduino. Para ello es necesario conectar con al módulo step-up a los pins de tierra y 5V. Tened en cuenta que normalmente no se recomienda utilizar este

puerto pero como en este caso tenemos regulado el voltaje podemos utilizar este puerto sin peligro.” (Stopa, 2019)

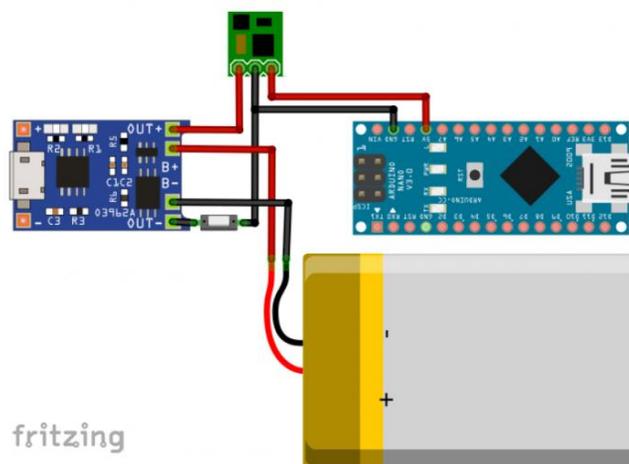


Figura 16. Esquema de conexión batería (Stopa, 2019)

Con las recomendaciones que se tiene para el proyecto a futuro se tiene una cotización estándar de lo que este sistema de alarma puede valer

Tabla 3. Tabla de presupuesto futuro

PRESUPUESTO FUTURO			
Cantidad	Implementos	Valor unitario	Valor total
1	Camara de vigilancia	\$42.000	\$42.000
1	Batería LiPo ~2000 mAh	\$59.482	\$59.482
2	Buzzer	\$4.200	\$8.400
1	Módulo TP4056	\$1.290	\$1.290
1	Módulo step-up 5V	\$1.581	\$1.581
1	Arduino esp32	\$28.000	\$28.000
1	Sensor Infrarrojo Llama Flama Ky026 5cm 760-1100nm Arduino	\$14.000	\$14.000
1	Sensor De Gas Aire Mq2 Propano Metano Alcohol	\$16.300	\$16.300
2	Sensor infrarojo	\$5.500	\$11.000
1	Carcasa	\$70.000	\$70.000
TOTAL PRECIO PROYECTO FUTURO			\$249.182

9. Bibliografía

- ¿Que es MQTT? (2019). *Paessler*. Obtenido de Paessler: <https://www.paessler.com/es/it-explained/mqtt#:~:text=MQTT%20son%20las%20siglas%20de,cuanto%20al%20ancho%20de%20banda>
- admin. (3 de Diciembre de 2021). *industrias GSL*. Obtenido de industrias GSL: <https://industriassgsl.com/blogs/automatizacion/sensor-de-movimiento-infrarrojo#:~:text=El%20sensor%20de%20movimiento%20infrarrojo,a%20los%20ojos%20del%20humano.>
- Alexandra. (2015). *repositorio*. Obtenido de repositorio: <https://repositorio.utp.edu.co/items/0fc5553d-2592-484e-8c7f-9d0dfcdd7370>
- Antonio. (2024). *Repositorio Digital*. Obtenido de Repositorio Digital: <https://repositorio.upct.es/handle/10317/1883>
- Barillaro, S. L. (2016). *sedici*. Obtenido de sedici: <http://sedici.unlp.edu.ar/handle/10915/53061>
- Blanco Huelmos, A. (2023). *uvadoc*. Obtenido de uvadoc: <https://uvadoc.uva.es/handle/10324/60695>
- Carranza. (27 de Octubre de 2021). Obtenido de TodoMaker: <https://todomaker.com/blog/conociendo-al-esp32/>
- Carranza. (27 de Octubre de 2021). *TodoMarker*. Obtenido de TodoMarker.
- Electronics, U. (2 de Mayo de 2024). *UNIT Electronics*. Obtenido de UNIT Electronics: <https://uelectronics.com/producto/modulo-ky-026-sensor-de-flama/>
- Equipo editorial. (22 de Febrero de 2018). *Concepto*. Obtenido de Concepto : <https://concepto.de/sistema/>

IBM. (7 de Noviembre de 2023). *ibm*. Obtenido de ibm: <https://www.ibm.com/mx-es/topics/message-brokers>

jecrespom. (29 de Marzo de 2016). *Aprendiendo Arduino*. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2016/03/29/entorno-de-programacion-de-arduino-ide/>

Julián Pérez Porto, & Gardey. (8 de Julio de 2010). *definicion*. Obtenido de definicion: <https://definicion.de/alarma/>

Ministerio de Defensa Nacional de la Republica de Colombia. (2024). *mindefensa*. Obtenido de mindefensa: <https://www.mindefensa.gov.co/irj/portal/Mindefensa>

Módulo sensor de gas (MQ2). (2024). *Didacticaselectronicas*. Obtenido de Didacticaselectronicas: [https://www.didacticaselectronicas.com/index.php/sensores/gases/SEN0127-detail#:~:text=El%20m%C3%B3dulo%20sensor%20de%20gas%20anal%C3%B3gico%20\(MQ%2D2\)%20se,r%C3%A1pido%20Y%20dicha%20sensibilidad%20puede](https://www.didacticaselectronicas.com/index.php/sensores/gases/SEN0127-detail#:~:text=El%20m%C3%B3dulo%20sensor%20de%20gas%20anal%C3%B3gico%20(MQ%2D2)%20se,r%C3%A1pido%20Y%20dicha%20sensibilidad%20puede)

Pasca. (2013). *ucm*. Obtenido de ucm: https://www.ucm.es/data/cont/docs/506-2015-04-16-Pasca_TFM_UCM-seguridad.pdf

Patricio. (2021). *Repositorio Institucional*. Obtenido de Repositorio Institucional: <https://repositorio.umsa.bo/xmlui/handle/123456789/30634>

Perera. (20 de Febrero de 2021). *automatismo*. Obtenido de automatismo: <https://automatismosmundo.com/que-es-el-software-fritzing/>

Rosas Cruz, A. (2021). *Repositorio Insitucional*. Obtenido de Repositorio Insitucional: <https://repositorio.umsa.bo/xmlui/handle/123456789/30674>

Saa Ayala, J. F. (2023). *dspace*. Obtenido de dspace:

<http://dspace.utb.edu.ec/handle/49000/14249>

Santaella. (26 de Abril de 2022). *talently*. Obtenido de talently: <https://talently.tech/blog/que-es-android-studio/>

Stopa. (10 de noviembre de 2019). Obtenido de <https://bikepixels.com/2019/11/10/anadir-una-bateria-lipo-en-proyectos-con-arduino/?lang=es>

10. Anexos

Anexo A. Código Arduino

```
#include <WiFi.h>

#include <PubSubClient.h>

#include <WiFiClientSecure.h>

#include <ArduinoJson.h>

const char* ssid = "Gallo";

const char* password = "andres123";

const char* mqtt_server = "cff7b28550d3405c9080c6c228b542a9.s1.eu.hivemq.cloud";

const int mqtt_port = 8883;

const char* mqtt_username = "Andres123";

const char* mqtt_password = "Andres123";

int var = 0;

int buzzPin=5;

const int flameSensorPin = 17;

const int gasSensorPin = 32;

const int motionSensorPin = 25;

const int gasSensorThreshold = 1100;

const int motionSensorThreshold = 4000;

bool alarmState = false;
```

```

bool llamaBool= false;

bool gasBool= false;

bool movimientoBool = false;

StaticJsonDocument<256> doc;

// Definición de los canales a utilizar

const char* CONTROL_ANDROID_TOPIC = "control-android";

const char* CONTROL_ARDUINO_TOPIC = "control-arduino";

const int ledPin = 10;

static const char* root_ca PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIFazCCA1OgAwIBAgIRAIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEEdyb3VwMRUwEwYDVQQDEwxCUjU1JHIFJvb3QgWDEwHhcNMTEwNjA0MTEwND
M4WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCCGA1UEChMgS
W50ZXJuZXQgU2VjdXJpdHkgUmVzZWZyY2ggR3JvdXAxFtATBgNVBAMTDElUkcgU
m9vdCBYMTCCAilwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm5
4rVygch77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+0TM8u
kj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfxwv60pIgbz5MDmgK7iS4+3mX6UA5/TR5d8mUgjU
+g4rk8Kb4Mu0UIXjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sWT8KOEUt+zwvo/7V3LvSye0r
gTBIIDHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyHB5T0Y3HsLuJvW5iB4YlcNHlsdu87
kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UCB5iPNgiV5+I3lg02dZ77DnKxHZu8A/IJBdiB3
)EOF"

```

QW0KtZB6awBdpUKD9jf1b0SHzUvKBds0pjBqAlkd25HN7rOrFleaJ1/ctaJxQZBKT5ZPt0m9
 STJEadao0xAH0ahmbWnOIFuhjuefXKNEgV4We0+UXgVCwOPjdAvBBI+e0ocS3MFEVzG6u
 BQE3xDk3SzynTnjh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaH
 WBfEbwrwbqHyGO0aoSCqI3Haadr8faqU9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
 rU7m2Ys6xt0nUW7/vGT1M0NPAgMBAAGjQjBAMA4GA1UdDwEB/wQEAWIBBjAPBgNV
 HRMBAf8EBTADAQH/MB0GA1UdDgQWBBR5tFnme7bl5AFzgAiIyBpY9umbbjANBgkqhK
 iG9w0BAQsFAAOCAgEA VR9YqbyyqFDQDLHYGmkGjYkIrGF1XIpu+ILlaS/V9IZLubhzEFn
 TIZd+50xx+7LSYK05qAvqFyFWwhfQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ3BebYhtF8GaV0n
 xvwu077x/Py9auJ/GpsMiu/X1+mvoiBOv/2X/qkSsisRcOj/KKNFtY2PwByVS5uCbMiogziUwth
 DyC3+6WVwW6LLv3xLfHTjuCvjHIIInNzktHCgKQ5ORAZI4JMPJ+GslWYHb4phowim57iazt
 XOOJwTdwJx4nLCgdNbOhdjsnvzqvHu7UrTkXWStAmzOVyyghqpZXjFaH3pO3JLF+I+/+sK
 Aluvtd7u+Nxe5AW0wdeRIN8NwdCjNPElpzVmbUq4JUagEiuTDkHszxHpFKVK7q4+63SM1
 N95R1NbdWhscdCb+ZAJzVcoyi3B43njTOQ5yOf+1CceWxG1bQVs5ZufpsMljq4Ui0/1lvh+wj
 ChP4kqKOJ2qxq4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d1
 1TPAmRGunUHBcnWEvgJBQ19nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
 emyPxgcYxn/eR44/KJ4EBs+IVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iltreGCc=-----END
 CERTIFICATE-----)EOF";

WiFiClientSecure espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

char msg[50];


```
}else if (strcmp(topic, CONTROL_ANDROID_TOPIC) == 0) {  
  
int estado = payload[0] - '0';  
  
var=estado;  
  
alarmState=false;  
  
digitalWrite(buzzPin, LOW);  
  
}  
  
}  
  
void reconnect() {  
  
while (!client.connected()) {  
  
Serial.print("Attempting MQTT connection...");  
  
if (client.connect("ESP32Client", mqtt_username, mqtt_password)) {  
  
Serial.println("connected");  
  
client.subscribe(CONTROL_ANDROID_TOPIC);  
  
} else {  
  
Serial.print("failed, rc=");  
  
Serial.print(client.state());  
  
Serial.println(" try again in 5 seconds");  
  
delay(5000);  
  
}  
  
}}  
  
void Seguridad(int flame,int gas,int movimiento)  
  
{  
  
if(movimiento==HIGH){
```

```
Serial.println("Entro a Movimiento");

Serial.println(movimiento);

alarmState = true;

movimientoBool=true;

}else{

movimiento=LOW;

}

if (flame==0){

Serial.println("Entro a Fuego");

Serial.println(flame);

alarmState = true;

llamaBool = true;

}else{

digitalWrite(buzzPin,LOW);

}

if(gas > gasSensorThreshold){

Serial.println("Entro a Gas");

Serial.println(gas);

alarmState = true;

gasBool = true;

}else{

}

}
```

```
void setup() {  
  
  pinMode(flameSensorPin, INPUT);  
  
  pinMode(gasSensorPin, INPUT);  
  
  pinMode(buzzPin, OUTPUT);  
  
  pinMode(motionSensorPin, INPUT);  
  
  Serial.begin(9600);  
  
  setup_wifi();  
  
  espClient.setCACert(root_ca);  
  
  client.setServer(mqtt_server, mqtt_port);  
  
  client.setCallback(callback);  
  
}  
  
void loop() {  
  
  StaticJsonDocument<256> doc;  
  
  if (!client.connected()) {  
  
    reconnect();  
  
  }  
  
  client.loop();  
  
  int flameSensorValue = digitalRead(flameSensorPin);  
  
  int gasSensorValue = analogRead(gasSensorPin);  
  
  int motionSensorValue = digitalRead(motionSensorPin);  
  
  if(var==0){  
  
    alarmState = false;  
  
  }else if(var==1){
```

```

Seguridad(flameSensorValue,gasSensorValue,motionSensorValue);
}

if(alarmState){
doc["Llama"]=llamaBool;

doc["Gas"]=gasBool;

doc["Movimiento"]=movimientoBool;

char out[128];

int L = serializeJson(doc,out);

digitalWrite(buzzPin, HIGH);

client.publish(CONTROL_ARDUINO_TOPIC,out);

}else{

digitalWrite(buzzPin, LOW);

client.publish(CONTROL_ARDUINO_TOPIC,"NOALARMA");

}

```

Anexo B. Código Android Studio

```

package com.example.sistemaseguridad;

import static android.graphics.Color.GRAY;

import static android.graphics.Color.GREEN;

import static android.graphics.Color.RED;

import org.json.JSONObject;

import androidx.appcompat.app.AppCompatActivity;

```

```
import android.annotation.SuppressLint;

import android.content.res.ColorStateList;

import android.media.Image;

import android.os.Bundle;

import android.os.Handler;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.Switch;

import android.widget.TextView;

import android.widget.Toast;

import org.eclipse.paho.android.service.MqttAndroidClient;

import org.eclipse.paho.client.mqttv3.IMqttActionListener;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;

import org.eclipse.paho.client.mqttv3.IMqttToken;

import org.eclipse.paho.client.mqttv3.MqttCallback;

import org.eclipse.paho.client.mqttv3.MqttConnectOptions;

import org.eclipse.paho.client.mqttv3.MqttException;

import org.eclipse.paho.client.mqttv3.MqttMessage;

import org.eclipse.paho.client.mqttv3.MqttSecurityException;

public class MainActivity extends AppCompatActivity {
```

```
private static final String BROKER_URL =
"ssl://cff7b28550d3405c9080c6c228b542a9.s1.eu.hivemq.cloud:8883";

private static String MQTTUSER = "Andres123456";

private static String MQTTPASS = "Andres123456";

static String TOPIC = "control-android";

static String TOPICDOS = "control-arduino";

static String MSG_ARDUINO = "ALARMA";

static String MSG_ON = "1";

static String MSG_OFF = "0";

private static final String CLIENT_ID = "AndroidClient";

private MqttHandler mqttHandler;

MqttAndroidClient client;

Handler handler = new Handler(); // En esta zona creamos el objeto Handler

MqttConnectOptions options;

String clienteID = "";

Boolean permisoPublicar= true;

private boolean alarmaActiva = false;

private String sensorActivo = ""; // Variable para almacenar el sensor activo
```

```
@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

mqttHandler = new MqttHandler();

mqttHandler.connect(BROKER_URL,CLIENT_ID);

connectBroker();

ejecutarsuscribe();

Button btnON = findViewById(R.id.btnON);

Button btnOFF = findViewById(R.id.btnOFF);

btnON.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

enviarMensaje(TOPIC,MSG_ON);

alarmaActiva = true;

});

btnOFF.setOnClickListener(new View.OnClickListener() {

@Override
```

```
public void onClick(View view) {  
  
    enviarMensaje(TOPIC,MSG_OFF);  
  
    alarmaActiva = false; // Establecer la alarma como desactivada cuando se presiona el botón de  
    apagado  
  
    apagarInterruptores();  
  
    });  
  
    }  
  
private void apagarInterruptores() {  
  
    Switch switchLlama = findViewById(R.id.switchLlama);  
  
    Switch switchGas = findViewById(R.id.switchGas);  
  
    Switch switchMovimiento = findViewById(R.id.switchMovimiento);  
  
  
    switchLlama.setChecked(false);  
  
    switchLlama.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.light_gray)));  
    switchLlama.setTrackTintList(ColorStateList.valueOf(getResources().getColor(R.color.gray)));  
  
    switchGas.setChecked(false);  
  
    switchGas.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.light_gray)));  
    switchGas.setTrackTintList(ColorStateList.valueOf(getResources().getColor(R.color.gray)));  
  
  
    switchMovimiento.setChecked(false);
```

```

switchMovimiento.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.li
ght_gray)));
switchMovimiento.setTrackTintList(ColorStateList.valueOf(getResources().getColor(R.color.gr
ay)));
}

```

```

private final int TIEMPO = 10000;

private void ejecutarsuscribe(){
handler.postDelayed(new Runnable() {

@Override

public void run() {
subscribeToTopic();
handler.postDelayed(this, TIEMPO);
}},TIEMPO);
}

private void enviarMensaje(String topic, String msg){
if (this.permisoPublicar){
try {
int qos=0;
Log.d("Entro a publish Encender",msg);
this.client.publish(topic,msg.getBytes(),qos,false);
} catch (MqttException e) {

```

```
e.printStackTrace();

}}

private void connectBroker(){

this.client = new

MqttAndroidClient(this.getApplicationContext(),BROKER_URL,this.CLIENT_ID);

this.options = new MqttConnectOptions();

this.options.setUserName(MQTTUSER);

this.options.setPassword(MQTTPASS.toCharArray());

try {

IMqttToken token = this.client.connect(options);

//Toast.makeText(getBaseContext(), "Llego", Toast.LENGTH_SHORT).show();

token.setActionCallback(new IMqttActionListener() {

@Override

public void onSuccess(IMqttToken asyncActionToken) {

Toast.makeText(getBaseContext(), "Conectado", Toast.LENGTH_SHORT).show();

subscribeToTopic();

}

@Override

public void onFailure(IMqttToken asyncActionToken, Throwable exception) {

Toast.makeText(getBaseContext(),"Error no conectado: "+ exception,

Toast.LENGTH_SHORT).show();

Log.d("Error MQTT",exception.toString());
```

```
}});  
  
} catch (MqttException e) {  
  
Log.d("Error MQTT",e.getMessage());  
  
e.printStackTrace();  
  
}}  
  
@Override  
  
protected void onDestroy(){  
  
mqttHandler.disconnect();  
  
super.onDestroy();  
  
}  
  
private void subscribeToTopic(){  
  
try {  
  
this.client.subscribe(TOPICDOS,0);  
  
  
} catch (MqttSecurityException e) {  
  
Log.d("ejecutarsuscribe",e.getMessage());  
  
e.printStackTrace();  
  
} catch (MqttException e) {  
  
e.printStackTrace();  
  
}  
  
  
  
this.client.setCallback(new MqttCallback() {
```

```
@Override

public void connectionLost(Throwable cause) {

    Toast.makeText(getBaseContext(),"Se desconecto",Toast.LENGTH_SHORT).show();

}

@SuppressLint("ResourceAsColor")

@Override

public void messageArrived(String topic, MqttMessage message) throws Exception {

    TextView txtinfo = findViewById(R.id.txtInfo);

    Switch switchLlama = findViewById(R.id.switchLlama);

    Switch switchGas = findViewById(R.id.switchGas);

    Switch switchMovimiento = findViewById(R.id.switchMovimiento);

    if (topic.matches(TOPICDOS)) {

        String msg = new String(message.getPayload());

        if (!msg.equals("NOALARMA")) {

            JSONObject jsonMsg = new JSONObject(msg);

            // Acceder a los campos del JSON

            Boolean llama = jsonMsg.getBoolean("Llama");

            Boolean movimiento = jsonMsg.getBoolean("Movimiento");

            Boolean gas = jsonMsg.getBoolean("Gas");
```

```
switchLlama.setChecked(false);

switchGas.setChecked(false);

switchMovimiento.setChecked(false);

if(llama || movimiento || gas){
    alarmaActiva = true;
    txtinfo.setText(R.string.suscribe_on);

    if(llama){
        switchLlama.setChecked(true);
        switchLlama.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.green))
        );
        switchLlama.setTrackTintList(ColorStateList.valueOf(getResources().getColor(R.color.AccentGreen)));
    }else if(gas){
        switchGas.setChecked(gas);
        switchGas.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.green)));
        switchGas.setTrackTintList(ColorStateList.valueOf(R.color.AccentGreen));
    }else if(movimiento){
        switchMovimiento.setChecked(movimiento);
```

```

switchMovimiento.setThumbTintList(ColorStateList.valueOf(getResources().getColor(R.color.g
reen)));

switchMovimiento.setTrackTintList(ColorStateList.valueOf(getResources().getColor(R.color.Ac
centGreen)));

}

}else{

alarmaActiva = false;

apagarInterruptores();

txtinfo.setText(R.string.suscribe_off);

}}

else{

alarmaActiva = false;

apagarInterruptores();

txtinfo.setText(msg);

txtinfo.setTextColor(GRAY);

Log.d("suscribe Off",msg);

}}

@Override

public void deliveryComplete(IMqttDeliveryToken token) {

}});

}

};

```

Anexo C. Prender y Apagar el Sistema de Alarma por el Aplicativo Móvil.

Mensaje que se envió de la aplicación móvil al servidor para prender el sistema de alarma.

The screenshot shows the MQTT Explorer interface. The 'control-arduino' topic is selected. A table of messages is displayed with the following data:

Seq #	Received Time	Topic	QoS
437	09:23:24.23804558	control-android	0
438	09:23:24.33804858	control-android	0
439	09:23:25.33805449	control-android	0

The 'control-android' topic is highlighted in the table. A red box highlights the 'control-android' topic filter and the 'Dump Messages (JSON)' button. A red arrow points from the 'control-android' topic in the table to the 'control-android' topic in the Topics Collector pane below, which is also circled in red.

Mensaje que se envió de la aplicación móvil al servidor para apagar el sistema de alarma.

The screenshot shows the MQTT Explorer interface. The 'control-arduino' topic is selected. A table of messages is displayed with the following data:

Seq #	Received Time	Topic	QoS
437	09:26:40.34000179	control-arduino	0
438	09:26:40.34000179	control-arduino	0
439	09:26:40.34000179	control-arduino	0
440	09:26:40.34000179	control-arduino	0
441	09:26:40.34000299	control-arduino	0
442	09:26:40.34000377	control-arduino	0
443	09:26:40.34000377	control-arduino	0
444	09:26:40.34000377	control-arduino	0

The 'control-arduino' topic is highlighted in the table. A red box highlights the 'control-arduino' topic filter and the 'Dump Messages (JSON)' button. A red arrow points from the 'control-arduino' topic in the table to the 'control-arduino' topic in the Topics Collector pane below, which is also circled in red.

Mensaje que se envía a la aplicación móvil mientras tenga un estado inicial continuo.

The screenshot shows the MQTT Explorer interface. The 'control-android' topic is selected. A table of messages is displayed with the following data:

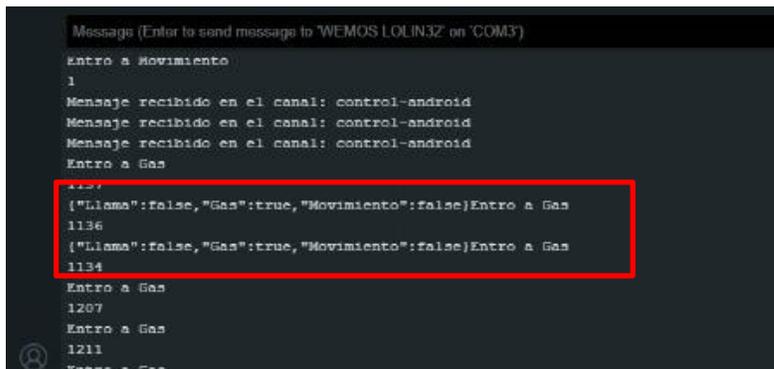
Seq #	Received Time	Topic	QoS
443	09:25:18.33918708	control-arduino	0
444	09:25:18.33918709	control-arduino	0
445	09:25:18.33918711	control-arduino	0
446	09:25:18.33918713	control-arduino	0
447	09:25:18.33918715	control-arduino	0
448	09:25:18.33918717	control-arduino	0
449	09:25:18.33918718	control-arduino	0
450	09:25:18.33918720	control-arduino	0

The 'control-arduino' topic is highlighted in the table. A red box highlights the 'control-arduino' topic filter and the 'Dump Messages (JSON)' button. A red arrow points from the 'control-arduino' topic in the table to the 'control-arduino' topic in the Topics Collector pane below, which is also circled in red.

Anexo D. Prueba Sensor de Gas

Información que envía arduino al bróker

```
{“Llama”:false,“Gas”:true,“Movimiento”:false}
```



Como se ve el mensaje llegado a la aplicación móvil



Indica que la alarma se encuentra activa y muestra que el sensor de gas está activo

Anexo E. Prueba Sensor de Llama.

Información que envía arduino al bróker

```
{"Llama":true,"Gas":false,"Movimiento":false}
```

```
1365
Entro a Gas
1366
Entro a Gas
1375
Mensaje recibido en el canal: control-android
Mensaje recibido en el canal: control-android
Entro a Fuego
0
{"Llama":true,"Gas":false,"Movimiento":false}Entro a Fuego
0
{"Llama":true,"Gas":false,"Movimiento":false}Entro a Fuego
0
Entro a Fuego
0
```

Como se ve el mensaje llegado a la aplicación móvil



Indica que la alarma se encuentra activa y muestra que el sensor de Fuego está activo

Anexo E. Prueba Sensor de Movimiento

Información que envía arduino al bróker

```
{“Llama”:false,“Gas”:false,“Movimiento”:true}
```

```
1365
Entro a Gas
1366
Entro a Gas
1375
Mensaje recibido en el canal: control-android
Mensaje recibido en el canal: control-android
Entro a Fuego
0
{"Llama":true,"Gas":false,"Movimiento":false}Entro a Fuego
0
{"Llama":true,"Gas":false,"Movimiento":false}Entro a Fuego
0
Entro a Fuego
0
```

Como se ve el mensaje llegado a la aplicación móvil



Indica que la alarma se encuentra activa y muestra que el sensor de Movimiento está activo