Escuela Politécnica Superior de Jaén

Máster Universitario en Ingeniería Mecatrónica

UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

PROTOTIPO DE SISTEMA DE CONTROL DE AFORO EN ESPACIOS CERRADOS

Alumno/a: Calero Brito, Pablo Alejandro

Tutores: Prof. D. Macarena Espinilla Estévez

Alicia Montoro Lendínez

Dpto.: Departamento de Informática

Septiembre, 2023

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña MACARENA ESPINILLA ESTEVEZ y Doña ALICIA MONTORO LEDINEZ, tutoras del Proyecto Fin de Máster titulado: PROTOTIPO DE SISTEMA DE CONTROL DE AFORO EN ESPACIOS CERRADOS, que presenta PABLO ALEJANDRO CALERO BRITO, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, septiembre de 2023

El alumno: Los tutores:

Pablo Calero Brito

Maarena Espinilla Estevez

Alicia Montoro Lendínez

Agradecimientos

Quiero expresar mi más sincero agradecimiento a las personas que han sido pilares fundamentales en mi trayecto hacia el éxito profesional. En primer lugar, quiero reconocer el inquebrantable apoyo brindado por mis padres, hermanos y mi compañera; su amor incondicional, respaldo emocional y sacrificio han sido los cimientos de mi desarrollo tanto personal como profesional.

Asimismo, quiero expresar mi profunda gratitud hacia mis dedicadas tutoras, cuya orientación experta, paciencia y dedicación incansable fueron vitales para dar forma a este logro. Sus conocimientos y experiencia en el campo de la mecatrónica fueron una fuente constante de inspiración y aprendizaje.

Este logro es el fruto del esfuerzo conjunto de numerosas personas excepcionales, y les agradezco de corazón por ser parte fundamental de mi vida y por hacer posible esta realización

FICHA DEL TRABAJO FIN DE TITULO	
Titulación	Máster universitario en Ingeniería Mecatrónica
Modalidad	Proyecto de Ingeniería
Especialidad	
Idioma	Castellano
Tipo	
TFT en equipo	No
Fecha de Asignación	16/01/2023
Descripción corta	Los ambientes inteligentes son entornos donde parte de sus objetos u ocupantes se encuentran envueltos o asignados por una red de sensores y actuadores con el objetivo de buscar el bienestar de los ocupantes y conseguir una relación más amigable, sostenible y segura del ocupante en dicho entorno. El presente trabajo fin de máster está centrado en el análisis, diseño y desarrollo de un sistema de control de aforo en espacios cerrados. Para ello, se utilizarán dispositivos bajo coste para orquestar el sistema junto con sensores de visión o de escaneado inalámbrico para conocer el número de personas que se encuentra en el espacio. Los datos procesados de número de personas del aforo serán enviados y procesados a una plataforma IoT de terceros para su visualización. Finalmente, se llevarán a cabo diferentes escenarios para evaluar el sistema desarrollado

NRMAS APLICADAS EN ESTE DOCUMENTO	
LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

Índice

Índice de Ilustraciones

Índice de tablas

1. INTRODUCCIÓN

En la actualidad, el control de aforo y la evaluación de la calidad del aire en espacios cerrados se han convertido en aspectos cruciales para garantizar la seguridad y el bienestar de las personas. La creciente preocupación por la propagación de enfermedades y la importancia de mantener ambientes interiores saludables ha llevado al desarrollo de nuevas tecnologías para abordar estos desafíos.

El presente proyecto de fin de máster propone la implementación de un prototipo para el control de aforo en tiempo real y la medición de la pureza del aire en espacios cerrados, mediante el uso de una cámara térmica y sensores de gas. El objetivo principal del proyecto es proporcionar una herramienta eficiente, precisa y de bajo coste que permita conocer el número de personas presentes en un lugar y, a su vez, establecer una relación directa entre la pureza del aire y la ocupación del ambiente.

El proyecto se basa en la integración y manejo de tecnología de bajo coste, utilizando el sensor térmico modelo AMG8833, capaz de generar imágenes a partir de la radiación infrarroja emitida. Asimismo, se ha incorporado un sensor de gas de la familia MQ para evaluar la calidad del aire. Este componente posee la capacidad de detectar diversos gases nocivos presentes en el ambiente, como dióxido de carbono, amoníaco y otros compuestos, que pueden afectar la calidad del aire interior.

Con el fin de brindar una representación gráfica y fácil para interpretar de los datos obtenidos, se ha creado una dashboard personalizada. Esta herramienta permite a los usuarios visualizar en tiempo real tanto el número de personas presentes como los niveles de pureza del aire en el ambiente.

La combinación de estos elementos ha permitido desarrollar un prototipo funcional que contribuye a la mejora de la gestión de espacios interiores y la promoción de ambientes más seguros y saludables. Además, el enfoque de este proyecto radica en la integración de la tecnología de visión térmica y los sensores de gas, lo que proporciona una evaluación completa y precisa del ambiente en tiempo real.

En resumen, este proyecto de fin de máster presenta una solución eficiente y tecnológicamente avanzada para el control de aforo y la evaluación de la pureza del aire en espacios cerrados, brindando una herramienta útil tanto en el ámbito público como privado, con potencial para contribuir al bienestar de las personas en distintos escenarios de la vida cotidiana.

1.1. Marco Teórico

La pandemia de COVID-19 ha sido un evento sin precedentes que ha resaltado la urgente necesidad de implementar medidas efectivas para mitigar la propagación de enfermedades infecciosas en espacios cerrados, donde la concentración de personas aumenta considerablemente el riesgo de contagio. El desarrollo de un prototipo para controlar el aforo en estos lugares se presenta como una solución imprescindible para proteger la salud y bienestar de las personas.

La utilización del sensor AMG8833, como cámara térmica, resulta ser una herramienta valiosa y de bajo coste que permite detectar rápidamente la presencia de cualquier individuo en un espacio cerrado. A su vez, el sensor MQ-135, con su capacidad para medir la pureza del aire y detectar gases nocivos como el dióxido de carbono (CO_2) y los compuestos orgánicos volátiles (COVs), se convierte en un aliado esencial. La mala ventilación en estos ambientes puede favorecer la propagación de enfermedades, y el monitoreo constante del aire contribuye a mantener un ambiente seguro y saludable.

Es importante destacar que, aunque la pandemia de COVID-19 haya sido el punto de partida, este prototipo posee una ventaja significativa al ser una solución versátil y adaptable. En caso de futuros brotes de enfermedades contagiosas o la aparición de nuevos virus, el sistema de control de aforo con tecnologías de detección podría ajustarse rápidamente para responder a la situación.

A pesar de que en la mayoría de países ya no son obligatorias las medidas de bioseguridad en espacios cerrados debido a la menor letalidad del virus. Se cuestiona cómo el número de personas influye en la pureza del aire en dichos lugares y la importancia de mantener la calidad del aire para proteger a las personas más

vulnerables de nuestros hogares, no solo del COVID-19, sino también de cualquier otra toxina presente.

Por lo que, es importante seguir manteniendo protocolos y medidas de seguridad en espacios cerrados, así como buscar soluciones para garantizar la calidad del aire mediante ventilación adecuada, purificadores y monitoreo constante. La idea central es la importancia de cuidar la salud y el bienestar de las personas, considerando la calidad del aire en entornos cerrados.

Aunque aún existen incertidumbres sobre este virus altamente contagioso, Castro, R. L. (2020), en su artículo ***'Coronavirus, una historia en desarrollo'*** publicado en la Revista Médica de Chile, concluye que el distanciamiento físico puede marcar la diferencia, reduciendo la transmisión, dado que el COVID-19 se transmite de persona a persona vía gotas de origen respiratorio cuando una persona infectada tose o estornuda. (Castro, 2020)

En muchos países, el control de aforo y la implementación de medidas preventivas, especialmente para personas de alto riesgo, son requisitos obligatorios en lugares como residencias de mayores, oficinas, restaurantes e incluso en nuestros propios hogares, con el objetivo de mantener condiciones de seguridad. La decisión de incorporar tecnologías de bajo coste para controlar el aforo y monitorear la calidad del aire refleja un compromiso por parte de los responsables de los espacios cerrados hacia la seguridad y salud de sus visitantes y empleados. Esto puede generar confianza y tranquilidad en la población, fomentando la participación en actividades sociales y económicas.

El desarrollo de este prototipo de sistema de control de aforo para espacios cerrados, a través de cámaras térmicas y sensores de aire, es una medida necesaria y justificada, ya que tiene como objetivo proteger la salud pública, prevenir contagios, garantizar un ambiente seguro y saludable, cumplir con las regulaciones y adaptarse a situaciones futuras. Esta tecnología representa un valioso recurso para enfrentar situaciones similares en el futuro y promover la confianza en la sociedad. Para ello, es fundamental conocer la base teórica que rodea y permite la implementación del prototipo, orientado principalmente en sistemas de visión por computadora, así como los sensores y protocolos de comunicación.

1.1.1. Visión por computador

La visión por computadora, surgida en el siglo XX, fusiona informática, inteligencia artificial y percepción visual con el fin de permitir a las máquinas interpretar el mundo visual como los humanos. Daniel Jara, director y Fundador de INNOVATIVE-NET en su artículo ***"Computer Vision, ¿Qué es? ¿Qué valor genera? ¿En qué industria se aplica?*** *¨ (2021),* nos cuenta un poco la historia del surgimiento de la visión por computador, donde sus primeros pasos se centraron en representar imágenes digitalmente y detectar bordes mediante el operador de Sobel en la década de 1950. Sin embargo, no fue hasta que los años 60 que se establecen sus orígenes cuando los científicos intentaron replicar la vista humana utilizando herramientas computacionales junto con el "Perceptrón" de Rosenblatt para el reconocimiento de patrones, aunque sus limitaciones dieron paso a enfoques más avanzados en las siguientes décadas. La década de 1990 vio la incorporación de métodos basados en aprendizaje automático, como máquinas de soporte vectorial y clasificadores basados en árboles de decisión, mientras que la explosión definitiva ocurrió en la década de 2020, las redes neuronales convolucionales son la base estándar, permitiendo aplicaciones precisas en dispositivos económicos. La IA en el borde, con hardware especializado, posibilita un procesamiento eficiente para la visión por computadora (Boesch,2023).

Desde un punto de vista ingenieril, un sistema de visión artificial es un sistema autónomo que hace las veces en algunas de las tareas del sistema de visión humano. El conjunto de tareas o información que un sistema de visión artificial puede llegar a realizar o extraer van desde una detección de objetos en una imagen hasta la interpretación tridimensional de escenas complicadas (Muñoz Manzo, 2014).

Un sistema de visión por computadora se fundamenta en examinar imágenes o secuencias visuales con el fin de obtener datos y llevar a cabo funciones específicas. Este sistema se compone de diversos elementos esenciales, tales como la captura de imágenes, el preprocesamiento, la identificación y extracción de atributos, así como la interpretación y procesamiento de la información obtenida.

1.1.2. Detección de Objetos en Imágenes

Uno de los aspectos más fundamentales de un sistema de visión artificial es la capacidad de detectar y localizar objetos en imágenes. Utilizando algoritmos de detección de objetos, el sistema puede identificar objetos de interés y marcar su posición en la imagen. (Howese, 2013)

1.1.3. Extracción de Características y Atributos

Para realizar tareas más avanzadas, el sistema debe extraer características relevantes de la imagen. Esto implica identificar patrones, texturas y formas específicas que ayuden a comprender la naturaleza de los objetos presentes en la escena. (Howese, 2013)

1.1.4. Procesamiento e Interpretación de la información obtenida

El procesamiento de imágenes comprende una variedad de técnicas y algoritmos que posibilitan la modificación y análisis de imágenes para alcanzar distintos propósitos. En otras palabras, se aplican reglas a las características de las imágenes para que el algoritmo se adapte a nuestras necesidades. (Howese, 2013)

1.1.5. Sensor de imagen

Un sensor de imagen se define como cualquier dispositivo capaz de percibir la energía empleada, en nuestro contexto la luz constituye la forma de energía predominante. En otras palabras, emplea un componente fotosensible, como un semiconductor o una película fotosensible, que reacciona ante la frecuencia magnética de la luz, transformándola en señales eléctricas. Estas señales, en su mayoría, adoptan la forma de variaciones en el voltaje (Domingo , 2004).

El proceso fundamental detrás de la operación de un sensor de imagen implica la conversión de la energía luminosa en una señal eléctrica, la cual puede ser procesada posteriormente para crear una representación visual de la escena capturada. En el caso de sensores de imagen digitales, como los utilizados en cámaras y dispositivos fotográficos, estas señales eléctricas se convierten en valores digitales que representan píxeles en una imagen.

La calidad de un sensor de imagen se evalúa en función de varios parámetros, como su sensibilidad a la luz, la resolución espacial que puede capturar, el rango dinámico para diferenciar luces y sombras, así como la capacidad para manejar condiciones de iluminación diversas. A lo largo de los años, la tecnología de sensores de imagen ha avanzado significativamente, lo que ha permitido la creación de imágenes más nítidas, detalladas y realistas en una variedad de aplicaciones, desde la fotografía cotidiana hasta la ciencia y la tecnología médica.

1.1.6. OpenCV

OpenCV (Open Source Computer Vision Library) es una poderosa biblioteca de código abierto diseñada para la visión por computadora. Esta biblioteca incluye una amplia gama de algoritmos que permiten a los desarrolladores implementar soluciones avanzadas de visión artificial en sus proyectos (OpenCV, 2023).

Constituyendo una herramienta fundamental en el campo de la visión por computadora, OpenCV ofrece un conjunto de capacidades que abarcan desde el procesamiento de imágenes hasta la detección y seguimiento de objetos, la reconstrucción en 3D, la identificación de patrones y mucho más. Su amplia variedad de algoritmos ha permitido a científicos, ingenieros y desarrolladores en todo el mundo crear aplicaciones innovadoras en diversas áreas, incluyendo la robótica, la automatización industrial, la realidad aumentada, la medicina y la seguridad.

Es importante destacar que el documento describe específicamente la API de OpenCV 2.x, que ha evolucionado para convertirse en una API de C++, por lo que nos facilita poder integrar de forma sencilla con una plataforma como Python (OpenCV, 2023).

1.1.7. Python

Python es un lenguaje de programación creado en la década de 1990 por Guido van Rossum. Fue concebido como el sucesor de un lenguaje llamado ABC. Guido es el principal autor de Python, pero ha recibido numerosas contribuciones de otros desarrolladores.

En 2001, se estableció la Python Software Foundation (PSF), una organización sin fines de lucro destinada a gestionar la propiedad intelectual relacionada con

Python. Todas las versiones de Python son de código abierto, lo que significa que el código fuente está disponible públicamente para que cualquiera pueda acceder y modificar (Perez, Díaz, & Becerra Garcia , 2014).

Python se ha convertido en un lenguaje de programación muy popular y versátil, utilizado en una amplia gama de aplicaciones, desde desarrollo web hasta inteligencia artificial y análisis de datos.

1.1.8. Protocolo MQTT (HiveMQ, 2015)

El protocolo MQTT (Message Queue Telemetry Transport) es un protocolo de mensajería ligero y de código abierto diseñado para facilitar la comunicación entre dispositivos con ancho de banda limitado y conexiones inestables. Fue desarrollado por Andy Stanford-Clark de IBM y Arlen Nipper de Cirrus Link Solutions en 1999 y es ampliamente utilizado en el ámbito del Internet de las cosas (IoT) y la telemetría

El funcionamiento del protocolo MQTT se basa en el modelo de publicación/suscripción, lo que significa que los dispositivos se comunican a través de un intermediario central llamado "broker".

Broker: Es el servidor central que recibe, enruta y distribuye los mensajes entre los clientes. Los clientes se conectan al broker y pueden publicar mensajes en ciertos temas (topics) o suscribirse a temas para recibir mensajes. (referencia)

Cliente: Son los dispositivos o aplicaciones que se comunican a través del protocolo MQTT. Hay dos tipos de clientes:

Publicador (Publisher): Envía mensajes a temas específicos.

Suscriptor (Subscriber): Se suscribe a uno o varios temas para recibir los mensajes publicados en esos temas.

Tema (Topic): Es una cadena que actúa como etiqueta o identificador de los mensajes. Los mensajes se publican en temas específicos y los suscriptores se conectan a esos temas para comunicarse.

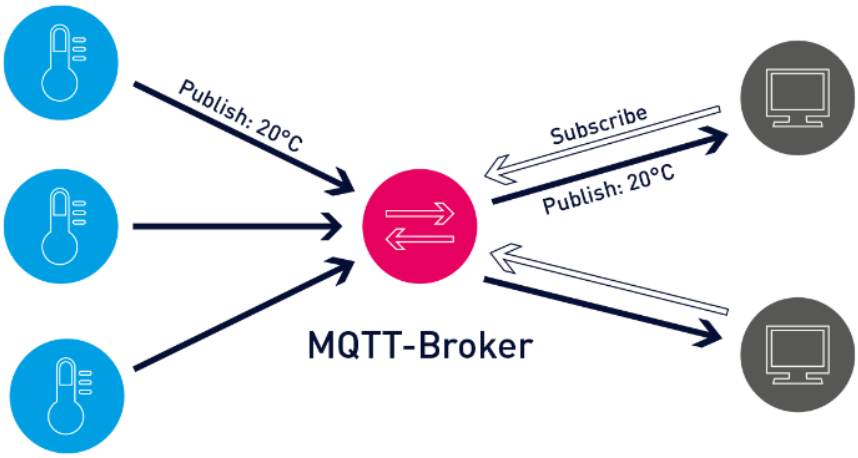

Ilustración 1 Protocolo MQTT 1.1.6.1 obtenido de: https://www.paessler.com/es/it-explained/mqtt

1.2. Propósitos generales y objetivos

En esta sección, se presentan los objetivos que trazaron el camino hacia la creación e implementación del sistema prototipo de control de aforo en espacios cerrados, empleando una cámara térmica y un sensor de gas.

1.2.1. Objetivo general

El objetivo general de este trabajo es diseñar e implementar un sistema prototipo de control de aforo en espacios cerrados utilizando una cámara térmica y un sensor de gas MQ-135. El sistema permitirá la recopilación, persistencia y representación gráfica de datos en tiempo real, mediante una plataforma IoT. Con este sistema, se podrá monitorear el flujo de personas en el espacio y evaluar la relación entre el número de personas presentes y la pureza del aire.

1.2.2. Objetivos específicos

Para alcanzar el objetivo general, se plantean los siguientes objetivos específicos:

- Revisar la literatura asociada a las tecnologías implicadas en el trabajo:

Se llevará a cabo una revisión exhaustiva de la literatura científica, técnicas y avances relacionados con las cámaras térmicas, sensores de gas de la familia MQ, sistemas de control de aforo y plataformas IoT. Esta revisión permite comprender el

estado actual de estas tecnologías, sus capacidades y limitaciones, y proporcionan una base sólida para el diseño del sistema.

- Analizar la relación existente del número de personas presentes dentro de un espacio cerrado con la pureza del aire mediante la utilización de un sensor de gas de la familia MQ y elección del sensor MQ ideal:

Se llevarán a cabo una serie de experimentos y mediciones en diferentes espacios cerrados para evaluar la concentración de gases contaminantes en el aire en función del número de personas presentes. Esto nos permitirá comprender cómo el aforo afecta la calidad del aire y si existe alguna relación significativa entre ambos factores.

- Implementar una arquitectura de almacenamiento de datos que permita de manera efectiva y eficiente la recolección, almacenamiento y persistencia de datos, garantizando así la disponibilidad y la integridad de la información a lo largo del tiempo.

Se realizará un breve análisis de las tecnologías que mejor se adaptan al sistema.

- Diseñar una dashboard para representar gráficamente los resultados obtenidos y tener un control de aforo controlado en tiempo real:

Se desarrollará una interfaz de usuario intuitiva y funcional que permita visualizar los datos recopilados de la cámara térmica y el sensor de gas en tiempo real. La dashboard mostrará gráficos y estadísticas

1.3. Estado del arte

Esta sección proporciona información sobre las últimas tecnologías y enfoques en el campo de los sistemas de control de aforo de bajo coste que han surgido en respuesta a la pandemia del COVID-19. Esta información permite tomar decisiones informadas en el desarrollo e implementación del sistema prototipo.

1.3.1. Control de Aforo

En diciembre de 2019, un gran número de casos de neumonía de origen desconocido se presentaron en Wuhan-China. El 7 de enero de 2020, las autoridades anunciaron que se trataba de un virus perteneciente a la familia de los Coronaviridae. Debido a la rápida propagación del virus a nivel global, el 12 de marzo de 2020, la *Organización Mundial de la Salud (OMS)* declaró pandemia al virus actualmente conocido como COVID-19, abreviatura en inglés de Coronavirus Disease 2019. Posteriormente, el virus recibió el nombre de SARS-CoV-2 (Worlddimeter, 2020).

El 21 de junio del 2021, se registraron 556.766 contagios de COVID-19 en todo el mundo, causando 16,951 muertes, siendo este el registro más alto de muertes registrado por día. [1]

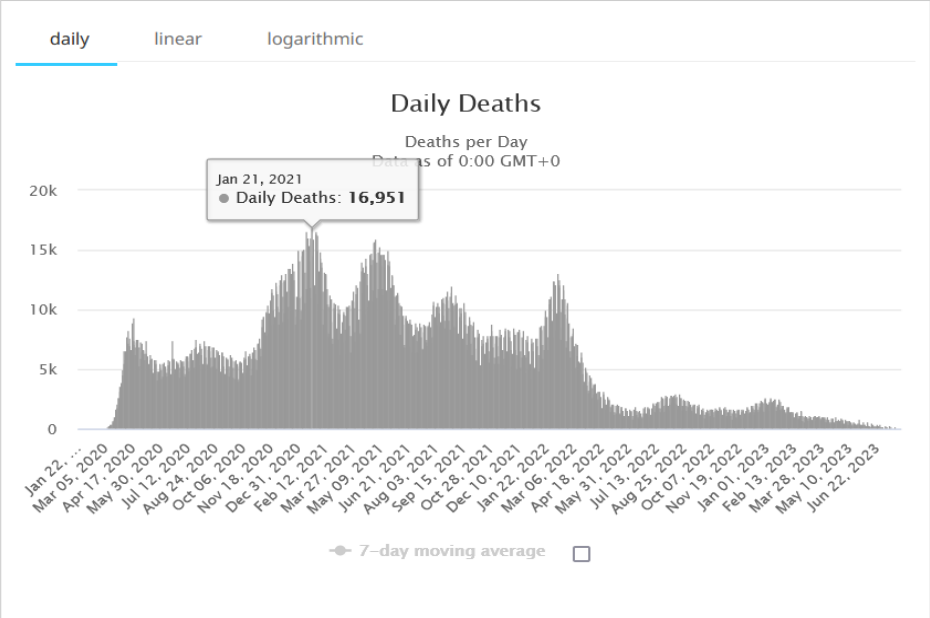

Ilustración 2 Registros Casos de COVID 1.3.1.1 (WORLDOMETER, 2020)

La propagación pandémica a nivel mundial ha incitado a los países a implementar medidas específicas de bioseguridad con el objetivo de contener el contagio. Entre estas medidas, se destaca la regulación del aforo en entornos cerrados como una estrategia esencial para prevenir la diseminación de patógenos infecciosos. La reducción del flujo de individuos en ubicaciones concretas desempeña un papel crítico en el mantenimiento del distanciamiento social y la minimización del riesgo de transmisión.

[1] WORLDOMETER. CORONAVIRUS CASES. 2020. DISPONIBLE EN: HTTPS://WWW.WORLDOMETERS.INFO/CORONAVIRUS/

En relación a esta problemática, se ha identificado un artículo de investigación titulado ***"Bidirectional People Counting System in Video Surveillance,"*** elaborado por Satish D. Pore y B.F. Momin (2017). Este estudio presenta un sistema computarizado que emplea técnicas avanzadas para clasificar y contabilizar personas que atraviesan pasillos o puertas.

El método de detección de personas en este sistema se basa en la aplicación del descriptor HOG (Histogram of Oriented Gradients), el cual permite identificar personas en imágenes de vigilancia mediante el análisis de los gradientes de orientación. Además, se utiliza el canal de Kalman para generar trayectorias de personas, lo que facilita el seguimiento preciso de su movimiento a lo largo del pasillo o puerta.

Este enfoque tecnológico se revela como una herramienta altamente efectiva para monitorear y controlar el flujo de personas en espacios públicos, particularmente en situaciones de salud pública críticas, donde el distanciamiento social y el control de aforo adquieren una relevancia excepcional para mitigar la propagación de agentes patógenos y salvaguardar la salud y bienestar de la población.

Es importante mencionar que, el desarrollo de sistemas de control de aforo por medio de cámaras térmicas resulta muy costoso implementarla, por lo que, para aplicaciones domésticas esta solución resultaría inalcanzable para los usuarios. Sin embargo, en la actualidad existen sensores de bajo coste como la AMG9933, que nos permiten obtener imágenes a partir de este sensor. Así es como Supria, & Nasir, M. en su trabajo de investigación ***“MONITORING OF BODY TEMPERATURE NON CONTACT USING AMG8833 THERMAL CAMERA AND FACE DETECTION”*** (2020), proponen un Sistema de Monitoreo de Temperatura Corporal Sin Contacto utilizando la Cámara Térmica AMG8833 y la Detección Facial. La cámara térmica AMG8833 se utiliza para medir la temperatura corporal, mientras que la detección facial utilizando el Clasificador Haarcascade se emplea para identificar rostros.

El sistema propuesto mediante la utilización del sensor AMG8833 que fue comparado con un sensor de termómetro conocido como modelo DN-997. El propósito era detectar la temperatura corporal y los rostros de cinco individuos. Los resultados

demostraron que el sistema propuesto tuvo un promedio de error de 0.10 °C, equivalente al 0.28% de error en las mediciones (Supria, 2020).

1.3.2. Sensores de Gas

Hoy en día, existen varios sistemas de detección y alerta de gases tóxicos basados en microcontroladores. ***Internet of Things (IOT) Based Gas Leakage Monitoring and Alerting System with MQ-2 Sensor*** es un sistema que detecta los gases peligrosos como el gas licuado de petróleo (LPG) y el propano y estos son mostrados en una pantalla LCD, notificando cada segundo. Si estos gases exceden el nivel normal, se genera una alarma de inmediato y también se envía un mensaje de alerta por correo electrónico a la persona autorizada a través de internet, utilizando una placa de desarrollo ARM. La ventaja de este sistema de detección y alerta automatizado sobre el método manual es que ofrece un tiempo de respuesta rápido y una detección precisa de una emergencia, lo que conduce a una difusión más rápida de la situación crítica (Chandra, Verma, & Kumar, 2017).

1.4. ANÁLISIS

En este apartado, se presenta el análisis de los requisitos funcionales y no funcionales del sistema prototipo de control de aforo en espacios cerrados, así como la planificación adecuada de las actividades correspondientes.

1.4.1. Requisitos

En esta sección se describen las funciones específicas que debe tener el sistema, además de las restricciones bajo las cuales el sistema debe funcionar.

1.4.1.1. Requisitos funcionales:

- Detección de personas: La cámara térmica debe ser capaz de detectar personas en tiempo real en el espacio objetivo y contar la cantidad de personas presentes.
- Medición de la pureza del aire: El sensor de gas debe proporcionar mediciones precisas de la calidad del aire en términos de concentración de gases contaminantes.
- Relación entre la pureza del aire y la ocupación: El sistema debe establecer una relación directa entre la pureza del aire y el número de

personas presentes, proporcionando datos que muestran cómo la ocupación afecta la calidad del aire.

- Comunicación MQTT: Los dispositivos Raspberry Pi 3 modelo b y ESP8266 deben comunicarse mediante el protocolo MQTT con Mosquito como broker para transmitir datos entre ellos.
- Almacenamiento en base de datos NoSQL: Los datos de aforo y calidad del aire deben almacenarse en una base de datos NoSQL para su posterior análisis y presentación.
- Dashboard: Se utilizará una dashboard interactiva que permita visualizar los datos en tiempo real, mostrando el número de personas presentes y la calidad del aire.

1.4.1.2. Requisitos no funcionales:

- Tiempo real: El sistema debe ser capaz de proporcionar información en tiempo real para una toma de decisiones efectiva y oportuna.
- Rango de detección: El prototipo debe posicionarse a una distancia que cumpla el margen de detección según las características específicas del sensor AMG8833.
- Usabilidad: La interfaz de usuario de la dashboard debe ser intuitiva y fácil de usar, para que los usuarios puedan interpretar los datos de manera clara y sencilla.
- Eficiencia energética: Se debe optimizar el consumo de energía de los dispositivos, especialmente de la Raspberry Pi y el ESP8266, para prolongar la vida útil de las baterías o reducir el consumo eléctrico

1.4.2. Planificación de tareas

En este apartado se presenta la planificación de las tareas que permitieron el desarrollo del trabajo final de máster.

Actividad	Asignado	Fecha
Revisión Bibliográfica	Pablo Calero	22/12/2022
Elección de la metodología	Pablo Calero	13/01/2023
Definir Requisitos Funcionales y No funcionales del sistema	Pablo Calero	17/01/2023
Fin Fundamentación Teórica	Pablo Calero	18/01/2023
Diseño Hardware	Pablo Calero	23/01/2023
Diseño del Circuito	Pablo Calero	25/01/2023
Diseño Software	Pablo Calero	21/02/2023
Instalación SI (Rasbian)	Pablo Calero	25/02/2023
Fin Etapa de diseño	Pablo Calero	27/02/2023
Instalación Librería Cámara Térmica (Adafruit)	Pablo Calero	03/03/2023
Implementación Circuito de Gas MQ-135	Pablo Calero	05/03/2023
Implementación circuito cámara térmica AMG8833	Pablo Calero	05/03/2023
Prueba de la librería cámara Térmica	Pablo Calero	13/03/2023
Implementación Programa Control de Aforo	Pablo Calero	06/05/2023
Instalación Librerías Complementarias	Pablo Calero	11/05/2023
Implementación Protocolo de Comunicación (MQTT)	Pablo Calero	17/05/2023
Prueba Programa de Sensor MQ-135	Pablo Calero	07/06/2023
Pruebas de Programa con Cámara AMG8833	Pablo Calero	21/06/2023
Prueba Envío de paquete de datos (envío de mensaje)	Pablo Calero	26/06/2023
Diseño de Dashboard	Pablo Calero	30/06/2023
Implementación Dashboard con Home Assistant	Pablo Calero	03/07/2023
Integración Base de Datos MongoDB	Pablo Calero	06/07/2023
Experimentación	Pablo Calero	07/29/2023

Tabla 1 Planificación de tareas 1.4.2.1

1.4.2.1. Diagrama de Gantt.

El diagrama de Gantt, es una representación visual simple de las actividades y duraciones del proyecto Es una herramienta clave que mejoran la eficiencia y la toma de decisiones en la planificación y ejecución de proyectos por su capacidad para descomponer tareas complejas en elementos más manejables y su capacidad para mostrar visualmente las dependencias entre ellas (Geraldi & Lechler, 2012).

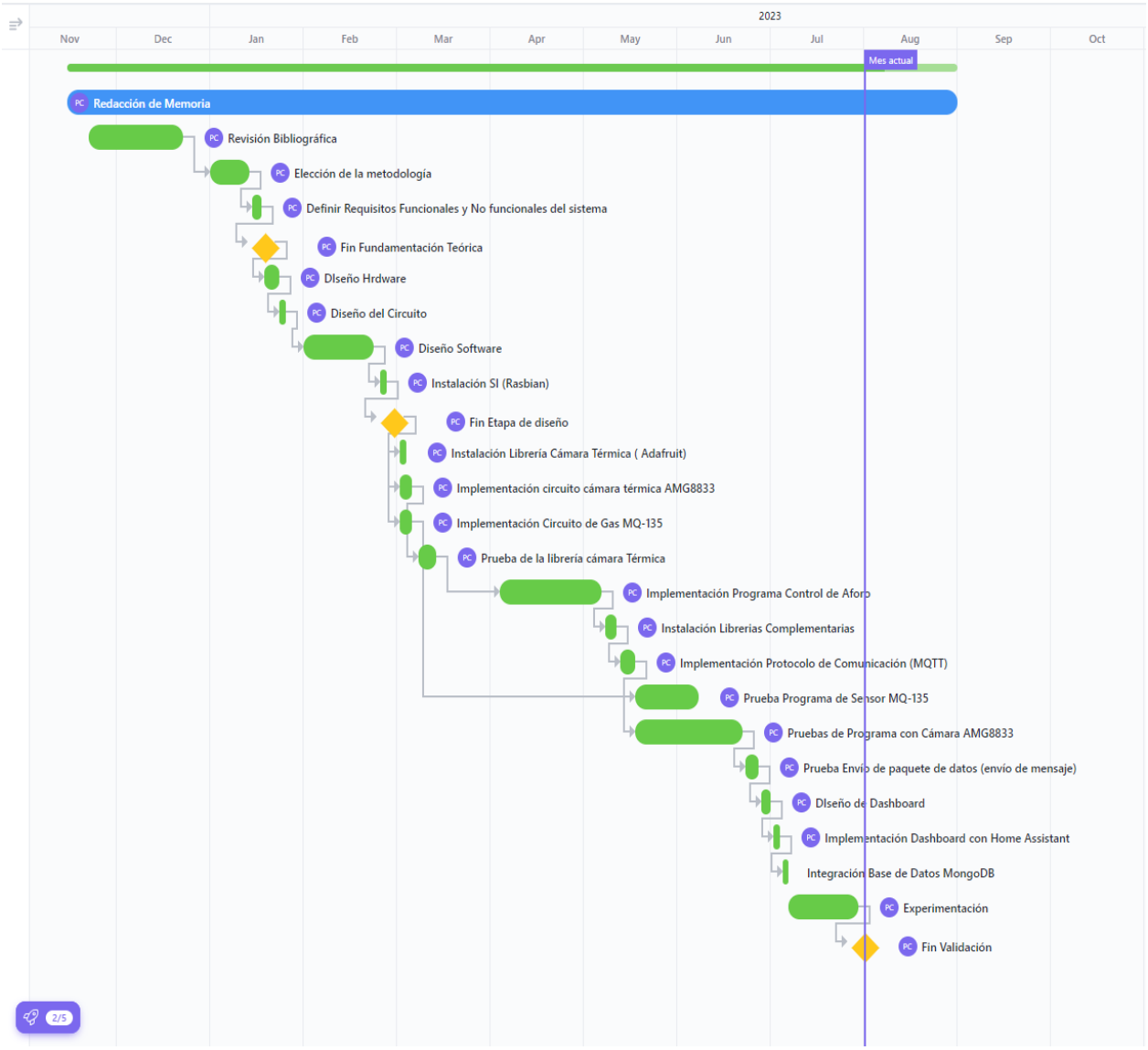

Ilustración 3 Diagrama de Gantt 1.4.2.1.1

1.5. Estimación de los costes

En esta sección se realiza una breve estimación de los costes totales que implica el desarrollo del proyecto.

1.5.1. Costes de recursos humanos

La tabla 2 proporciona información detallada sobre los salarios y las contribuciones a la seguridad social de los 255 días hábiles excluyendo los fines de semana, vacaciones y días festivos de tres perfiles que requiere el desarrollo del proyecto: jefe de proyecto, analista programador y programador.

Concepto	Jefe de Proyecto	Analista Programador	Programador
Salario bruto mensual	2454.0 €	1998.0 €	1345.0 €
Base de cotización	2863.0 €	2331.0 €	86,30€
Contingencias comunes: 23.60%	672,81€	550,12€	23,54€
Desempleo: 5.5%	157,47€	128,21€	86,30€
Contingencias profesionales: 1,5%	42,95€	34,97€	23,54€
Formación profesional: 0,6%	17,18€	13,99€	9,42€
Fogasa: 0,2%	5,73€	4,66€	3,14€
Total Seguridad Social	896,14€	731,95€	492,72€
Total mensual	3.350,14€	3.062,95€	1.837,72€
Sueldo anual	34356.0 €	27972.0 €	18830.0 €
Seguridad Social anual	10.753,68€	8.783,4€	5.912,64€
Total anual	39673.0 €	32301.0 €	20175.0 €

Tabla 2 Seguridad social 1.5.1.1

Las contribuciones a la Seguridad Social se dividen en varios componentes que incluyen:

- **Contingencias Comunes (23.60%):** Esta contribución tiene como objetivo financiar contingencias como enfermedad, maternidad, accidentes no laborales, entre otras.
- **Desempleo (5.5%):** Se destina a financiar prestaciones por desempleo.
- **Contingencias Profesionales (1.5%):** Cubre riesgos profesionales y accidentes de trabajo.

- **Formación Profesional (0.6%):** Financia programas de formación para los trabajadores.
- **Fogasa (0.2%):** Es una contribución al Fondo de Garantía Salarial, utilizado para cubrir ciertas deudas laborales en casos de insolvencia.

El total de Seguridad Social es la suma de todas las contribuciones a la seguridad social y representa el costo total que tanto la empresa como el empleado deben pagar para financiar los beneficios y las prestaciones sociales.

El total mensual es el salario bruto mensual menos las contribuciones a la seguridad social, lo que refleja el salario neto que el empleado recibe después de las deducciones.

El sueldo anual se calcula multiplicando el salario bruto mensual por 12 meses, mostrando el salario anual antes de las deducciones.

La Seguridad Social anual es la suma de las contribuciones a la seguridad social multiplicadas por 12 meses, reflejando el costo anual de las contribuciones a la seguridad social.

La tabla 3 proporciona una visión general de los costos anuales de tres perfiles del proyecto, considerando tanto el costo anual como el tiempo en el que se desempeñan a tiempo parcial, y muestra el costo total para cada uno.

Puesto de trabajo	Coste/anual	Tiempo	Coste total
Jefe de proyecto	39673.0 €	8 meses/ tiempo parcial	13224.33 €
Analista programador	32301.0 €	4 meses/ tiempo parcial	5383.50 €
Programador	20175.0 €	8 meses/ tiempo parcial	6725.00 €
		Total	25332.83 €

Tabla 3 Costes humanos 1.5.1.2

1.5.2. Costes de hardware y software

En la tabla 4, se puede observar los costes de los componentes de hardware utilizados en el proyecto.

Hardware	Cantidad	Precio unitario
Raspberry Pi 3 Modelo B	1	45.95 €
ESP8266	1	8.47 €
Protoboard	1	4.49 €
Cables	20	0.1 €
Resistencias	3	0.1 €
AMG8833	1	24.0 €
MQ-135	1	4.99 €
Total		90.2 €

Tabla 4 Costes hardware 1.5.1.3

El uso de software gratuito, como se muestra en la tabla 5, ofrece varias ventajas. En primer lugar, representa un ahorro significativo de costos al eliminar la necesidad de adquirir licencias. Además, este tipo de software suele ser de código abierto, permitiendo la personalización y flexibilidad. Las comunidades activas de usuarios brindan soporte y actualizaciones regulares, garantizando un uso seguro. La variedad de opciones y la compatibilidad con diferentes sistemas operativos proporcionan versatilidad. Aunque es gratuito, no tiene limitaciones de tiempo y puede ayudar en el aprendizaje y desarrollo de habilidades. Además, al usar software gratuito, uno contribuye al mantenimiento de proyectos de código abierto y potencialmente mejora la seguridad.

Software	Cantidad	Precio unitario
Rasbian	1	0 €
IDE Arduino	1	0 €
Prython 3.9.2	1	0 €
Cables	1	0 €
MongosDB	1	0 €
Home Assistant	1	0 €
Total		0 €

Tabla 5 Costes Software 1.5.1.4

1.5.3. Costes indirectos

La tabla 6 muestra los costes indirectos en el desarrollo del proyecto. Estos costes son esenciales para planificar y gestionar los gastos desde una perspectiva integral.

Servicios	Coste/mes	Tiempo de uso	Coste total
Internet	20 €	8 meses	160 €
Gastos de contigencias (luz, agua..)	100 €	8 meses	800 €
Total			960 €

Tabla 6 Costes adicionales 1.5.3.1

1.5.4. Costes totales

La tabla 7 muestra la estimación total de los costes.

Tipo de Coste	Coste Total
Coste recursos humanos	25332.83 €
Costes hardware	90.2 €
Costes software	0 €
Costes indirectos	960 €
Total de costes	26383.39 €
Beneficio (6%)	1582.62 €
Total	27966.01 €

Tabla 7 Costes totales 1.5.4.1

1.6. Metodología

Para el desarrollo del trabajo de fin de máster, se utilizará la metodología de cascada, ya que resulta altamente apropiada para la creación de un prototipo que permita controlar el aforo en tiempo real y medir la concentración del aire en espacios cerrados. Esta estructura secuencial ofrece diversas ventajas fundamentales para el desarrollo exitoso de este proyecto.

Ilustración 4 Metodología de Castada 1.6.1 (Niñas, 2011)

La metodología de cascada se basa en identificar de manera temprana y detallada los requisitos del sistema. Dado que el objetivo principal es desarrollar una herramienta eficiente y precisa para determinar el aforo de un lugar y la calidad del aire, esta metodología asegura capturar todos los aspectos esenciales de los requisitos del sistema.

La metodología de cascada permite una planificación adecuada y secuencial para implementar y probar cada uno de estos componentes, asegurando su correcto funcionamiento e integración de componentes específicos, como la cámara térmica AMG8833, el sensor de gas MQ-135, la en el sistema global.

Es importante enfatizar la realización de pruebas y verificaciones exhaustivas después de completar cada etapa. Esto permitirá detectar y corregir errores o problemas oportunamente, evitando realizar mayor cantidad de trabajos en etapas posteriores del desarrollo. Por lo tanto, la metodología de cascada facilita una gestión efectiva de riesgos, al permitir una evaluación adecuada e implementación de medidas de mitigación en cada etapa.

2. DISEÑO

En esta sección, se presenta el diseño de los tres ejes principales de desarrollo: hardware, software y diseño del programa. Además, se expone la arquitectura del proyecto, la cual demuestra la interconexión entre estos elementos

2.1. Diseño Hardware

El diseño del hardware establece el marco esencial para la comprensión de los elementos clave del sistema, permitiendo su reproducción para futuras investigaciones. Detallar las características y especificaciones técnicas de las placas de desarrollo, como la Raspberry Pi, la ESP8266, la cámara térmica AMG8833 y del sensor de gas MQ-135, facilita la comprensión de la funcionalidad y diseño del prototipo, fomentando la colaboración y permitiendo mejoras.

2.2. Placas de desarrollo

2.2.1. Raspberry Pi 3 Modelo B

Esta placa constituye una de las primeras iteraciones de la célebre serie de computadoras de placa única Raspberry Pi. Diseñada para un propósito específico, esta versión despliega un papel fundamental al operar como la unidad central encargada tanto de capturar y procesar imágenes provenientes de la cámara térmica AMG8833 como de gestionar la transmisión y recepción de paquetes de datos. Su presencia resulta esencial para el funcionamiento fluido y eficiente de este prototipo.

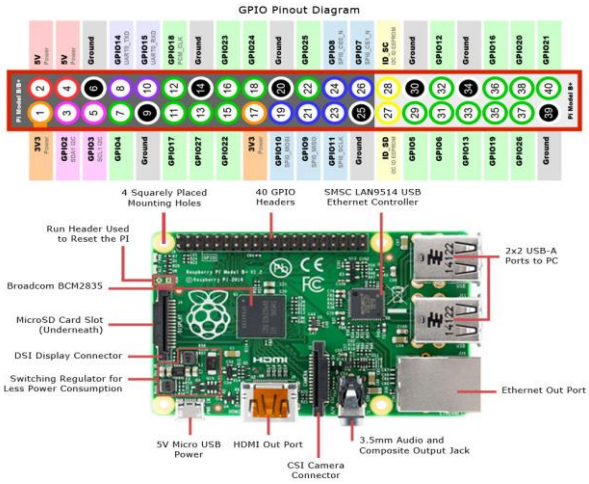

Ilustración 5 RaspberryPi Modelo B 2.2.1.1 (Electrónica y Ciencia,2016)

Las principales características de la Raspberry Pi Modelo B son las siguientes:

Característica	Descripción
Procesador	Broadcom BCM2837B0, Cortex-A53 (64 bits)
Núcleos de CPU	4 (Quad-Core)
Velocidad de CPU	1.2 GHz
Memoria RAM	1 GB LPDDR2 SDRAM
Almacenamiento	Ranura para tarjeta microSD (hasta 64 GB)
Conectividad de red	Ethernet (RJ45) y Wi-Fi 802.11n
Bluetooth	Bluetooth 4.2 BLE
Puertos USB	4 puertos USB 2.0
Salida de vídeo	HDMI, DSI (Interfaz de pantalla)
Salida de audio	Conector de 3.5 mm para auriculares
GPIO	40 pines GPIO
Puertos de cámara	Conector CSI para cámara Raspberry Pi
Puertos de pantalla táctil	Conector DSI para pantalla táctil Raspberry Pi
Alimentación	Micro USB (5V, 2.5A)
Sistema operativo soportado	Raspbian (basado en Linux) y otros sistemas operativos

Tabla 8 Especificaciones Raspberrypi Modelo B 2.2.1.1

2.2.2. Node MCU ESP8266

La ESP8266 es una familia de microcontroladores de bajo costo y bajo consumo de energía que están diseñados para permitir la conectividad Wi-Fi en dispositivos electrónicos. La ESP8266 es la encargada de capturar y transmitir los valores de pureza del aire, actuando como un puente inalámbrico entre la Raspberry Pi. Esto facilita la transmisión y recepción de datos mediante el protocolo MQTT, conocido por su eficiencia en la comunicación de mensajes entre dispositivos, lo que resulta esencial para lograr una interacción fluida y confiable para el desarrollo del prototipo.

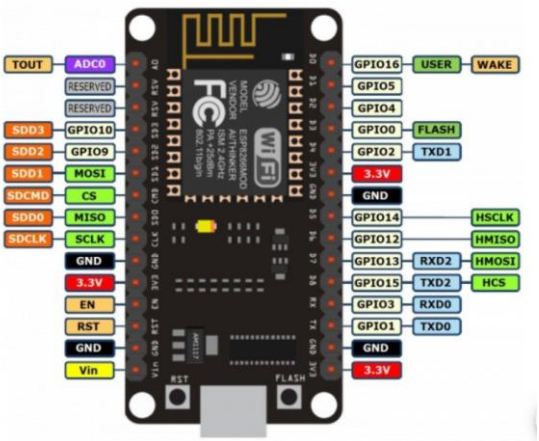

Ilustración 6 ESP8266 2.2.2.1 (Abellán,2019).

Sus principales características son:

Característica	Descripción
Microcontrolador	Tensilica L106 32-bit RISC CPU
Frecuencia de reloj	Hasta 160 MHz
Memoria Flash	512 KB a 16 MB
Memoria RAM	32 KB a 160 KB
Conectividad	WiFi 802.11 b/g/n
Modos WiFi	Estación, Punto de Acceso, Estación+AP
GPIO	Hasta 17 pines de entrada/salida digital
Interfaces	UART, SPI, I2C, ADC, PWM
Consumo energético	Modo activo: 80 mA - 170 mA Modo de bajo consumo: ~15 µA
Sistema Operativo	Ninguno (se programa directamente)
Lenguajes de Programación	C, C++, Python, LUA
Entorno de desarrollo	Arduino IDE, PlatformIO, Espressif SDK, etc.
Antenas	Interna o externa (dependiendo del modelo)
Voltaje de operación	3.0V - 3.6V
Tamaño	Variante de módulo, generalmente pequeño

Tabla 9 Especificaciones ESP8266 2.2.2.1

2.3. Sensores

2.3.1. AMG8833

La base teórica se enfoca principalmente en el análisis y comprensión de los conceptos que nos permiten explotar al máximo las funciones del sensor AMG8833, el cual es fundamental para el funcionamiento del sistema, ya que todas sus características y condiciones dependen de este dispositivo. Desarrollado por la compañía japonesa Panasonic, el AMG8833 es un sensor de imagen térmica que tiene la capacidad de captar la radiación infrarroja emitida por objetos, generando una imagen térmica en una matriz de 8x8 píxeles (Torres).

Este sensor se ha convertido en una herramienta ampliamente utilizada en diversas aplicaciones gracias a su bajo coste. Su asequibilidad lo hace especialmente atractivo para una amplia gama de campos, como la robótica, la domótica, donde el coste es esencial.

La versatilidad de este sensor radica en su capacidad para detectar temperaturas en un rango aproximado de -20 °C a 80 °C, lo que lo hace adecuado para una variedad de escenarios. Además, su interfaz de comunicación I2C facilita la integración con otros dispositivos electrónicos, como microcontroladores y computadoras, sin incurrir

en costosos gastos adicionales. Esto lo convierte en una solución rentable para una amplia gama de aplicaciones

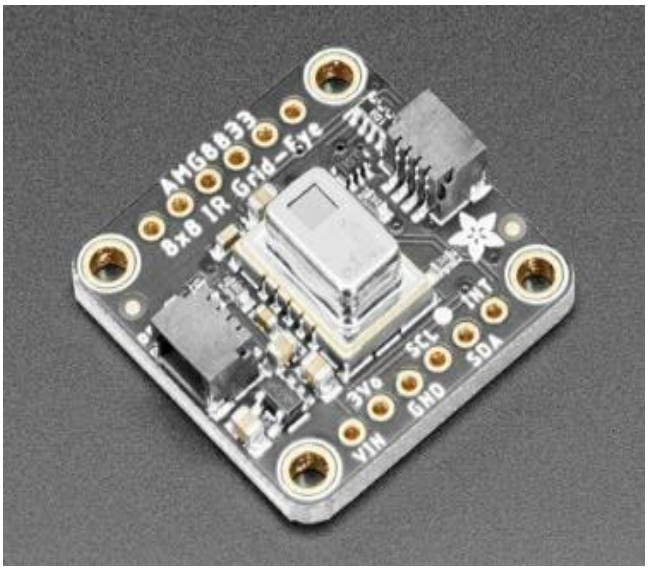

Ilustración 7 AMG8833 2.3.1.1 (Torres,s.f)

A continuación, se presentan algunas de las especificaciones técnicas clave del sensor AMG8833:

Característica	Valor
Matriz de píxeles	8x8
Rango de temperatura	-20°C a 100°C (-4°F a 212°F)
Resolución térmica	Aprox. 0.25°C
Campo de visión (FOV)	60° H x 60° V
Comunicación	Interfaz I2C
Frecuencia de actualización	1 Hz a 10 Hz
Alimentación	3.3V
Formato de salida	Matriz de datos térmicos
Características adicionales	Compensación de temperatura ambiente, detección de movimiento, opciones de filtrado

Tabla 10 Especificaciones AMG8833 7.3.1.1

2.3.2. MQ-135

El sensor MQ-135 representa un componente crucial en la detección de gases, ampliamente empleado para evaluar la presencia de diversas sustancias gaseosas en el ambiente. Principalmente, su utilidad radica en la identificación de gases perjudiciales como el dióxido de carbono (CO_2), monóxido de carbono (CO), amoníaco (NH_3) y otros compuestos orgánicos volátiles. El fundamento de este sensor se basa en la tecnología semiconductor de óxido metálico (MOS), cuya resistencia eléctrica se modifica en reacción a la existencia y concentración de diferentes gases.

Su funcionamiento se sustenta en un principio de variabilidad de resistencia, donde la concentración de gases influye directamente en la resistencia eléctrica del sensor. Montoro Ladines, A (2022), en su estudio de Fin de Máster ¨***Diseño y Prototipado de un Concentrador de Gases***¨**,** destaca que el sensor MQ-135 incorpora una capa sensible que altera su resistencia eléctrica en consonancia con los gases presentes en el ambiente. Esta capa, compuesta mayormente de óxido de estaño (SnO_2), es sometida a calentamiento eléctrico para posibilitar su operatividad. Adicionalmente, se resalta la importancia de un precalentamiento de 24 horas al utilizar el sensor por primera vez, con el fin de asegurar mediciones precisas.

Mediante el análisis de la curva de sensibilidad extraída del datasheet del sensor MQ-135, es posible aproximar el valor en partes por millón (ppm) en función de la relación de resistencias Rs/Ro. No obstante, para los propósitos de este trabajo, se abordará la concentración de CO_2, con el objetivo de establecer la correlación entre la calidad general del aire y el número de personas presentes en una sala.

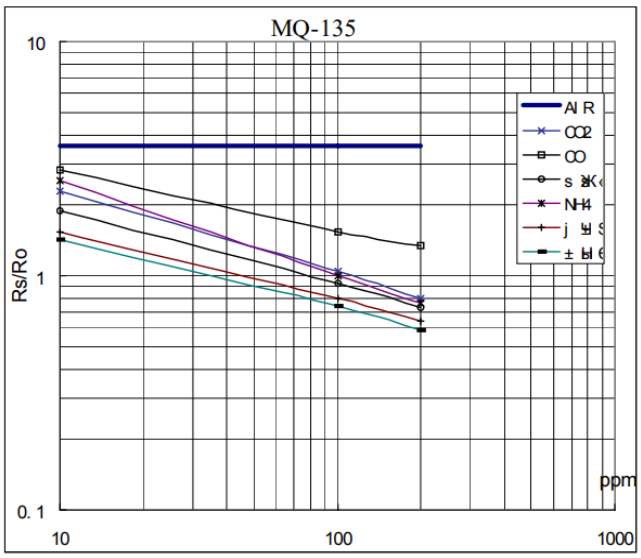

Ilustración 8 Curva de sensibilidad MQ-135 2.3.2.1

2.4. Diseño del circuito electrónico

En esta sección, se presentan los dos circuitos que forman parte del estudio del prototipo de sistema de control de aforo en espacios cerrados. Por un lado, se encuentra el circuito de la cámara térmica, responsable de cuantificar el flujo de individuos. Por otro lado, está el circuito del sensor de gas, el cual permite evaluar la calidad del aire presente dentro del salón de estudio.

2.4.1. Circuito cámara térmica

La ilustración 9 presenta el diseño del circuito principal, el cual este compuesto por el sensor térmico AMG8833 y la Raspberry Pi 3 Modelo B. Este circuito establece una comunicación efectiva entre ambos dispositivos. Usualmente, este proceso se lleva a cabo siguiendo las siguientes asignaciones de pines:

- El pin VCC de la cámara se conecta al pin Vcc de la Raspberry Pi, con el propósito de suministrar la energía necesaria para el funcionamiento de la cámara.
- El pin GND de la cámara se enlaza a uno de los pines GND de la Raspberry Pi, completando así el circuito de alimentación.
- El pin SDA de la cámara se vincula al pin GPIO2 de la Raspberry Pi, destinado a la transmisión de datos en el protocolo I2C.
- El pin SCL de la cámara se conecta al pin GPIO3 de la Raspberry Pi, que cumple la función de señalización de reloj en el protocolo I2C.

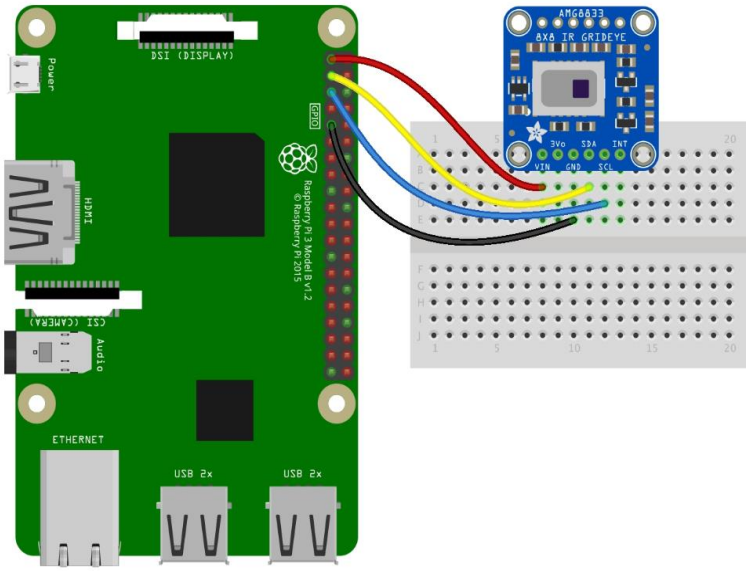

Ilustración 9 Diseño Circuito Principal- AMG8833 2.4.1.1

Una vez completadas las conexiones físicas, es necesario habilitar el protocolo I2C en la Raspberry Pi e instalar las bibliotecas pertinentes.

2.4.2. Circuito sensor de gas

Con el propósito de llevar a cabo una recolección de datos efectiva, es imperativo considerar que, en el contexto de la placa de desarrollo ESP8266, el pin ADC

únicamente es capaz de medir un rango de voltaje que oscila entre 0 y 3.3 voltios. No obstante, es esencial señalar que la señal analógica emitida por el pin del sensor MQ-135 abarca un rango de voltaje que varía entre 0 y 5 voltios. Por lo tanto, se hace necesario adaptar esta señal antes de conectarla al pin analógico de la placa. Para abordar este inconveniente de incompatibilidad de rangos de voltaje, se ha implementado un divisor de tensión con el propósito de reducir la tensión de entrada de 5 voltios a 3.3 voltios.

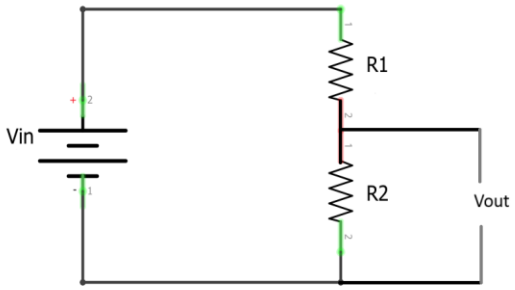

Ilustración 10 Divisor de Tensión 2.4.2.1

A continuación, se exponen las directrices que se han seguido para adecuar las muestras desde el intervalo de 0-5 voltios al intervalo de 0-3.3 voltios:

Utilizando un divisor de voltaje y aplicando la Ley de Ohm en el circuito ilustrado en la ilustración 11, es necesario calcular los valores de las resistencias. Este cálculo se realiza aplicando los principios teóricos a través del método de mallas de Kirchhoff, siguiendo los siguientes pasos:

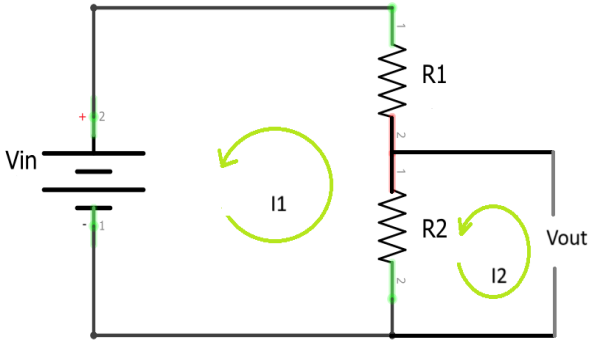

Ilustración 11 Divisor de Tensión - Mallas 2.4.2.2

Igualación de la intensidad en las dos mallas del circuito.

Fijación del valor de la resistencia R1 en 10 ohmios y resolvemos el valor de la segunda resistencia R2= 20 ohmios.

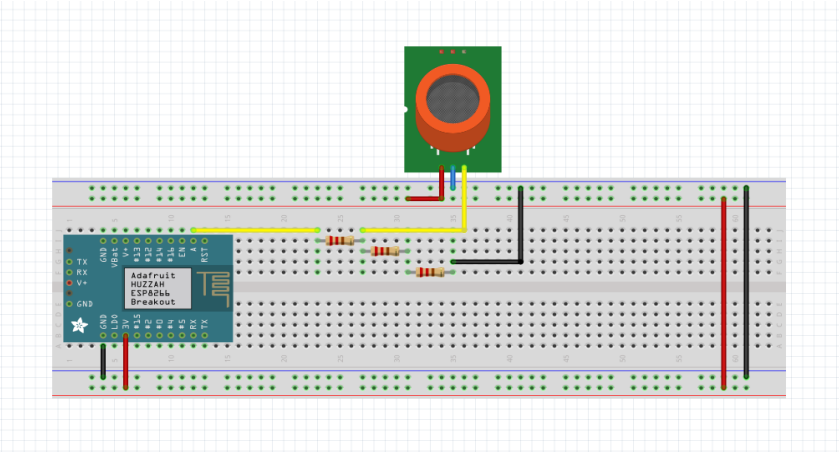

Ilustración 12 Circuito secundario MQ-135 2.4.2.3

La Ilustración 12 muestra el diseño definitivo del circuito encargado de evaluar la concentración de CO2 presente en una habitación.

2.5. Diseño Software

2.5.1. Diagrama de Flujo

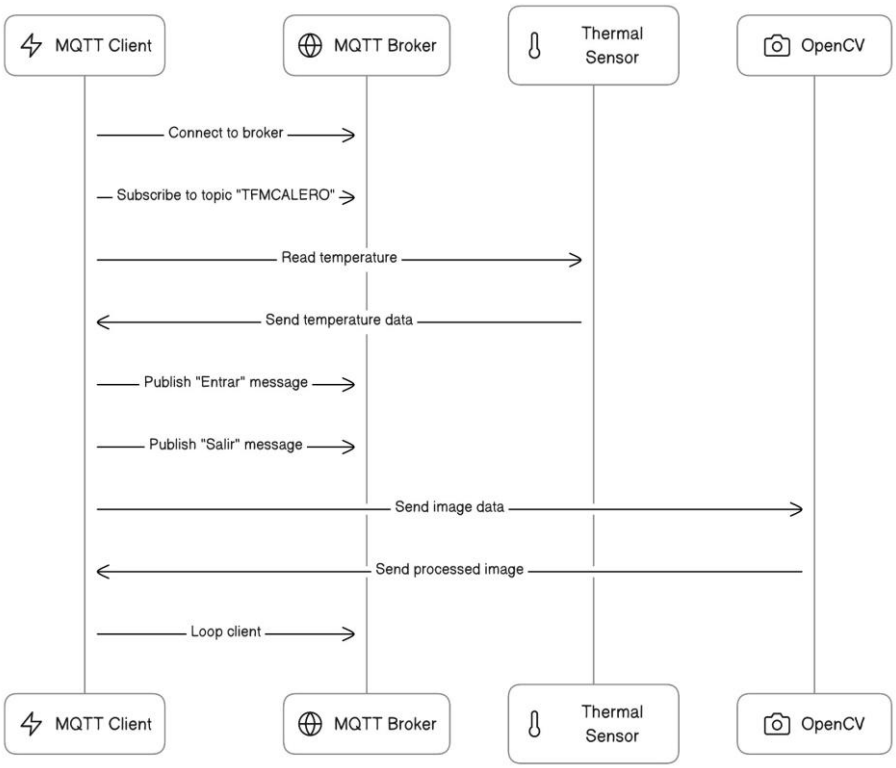

Ilustración 13 Diagrama de Flujo lógica del Programa 2.5.1.1

El diagrama de flujo representado en la Ilustración 11 ofrece una representación gráfica del algoritmo subyacente en el sistema de control de aforo mediante una cámara térmica. Este sistema se basa en la detección de objetos a través del sensor AMG8833 y la comunicación de datos a través del protocolo MQTT. Cuando el sensor detecta objetos que entran o salen de una región específica de interés, se establece una comunicación con un broker MQTT para transmitir mensajes relacionados con estas detecciones. Además, el sistema también recibe datos del sensor de gas mediante el mismo protocolo MQTT, lo que contribuye a un monitoreo más completo de la región de interés. Este diagrama de flujo proporciona una visión clara y sistemática de cómo se lleva a cabo el proceso de control de aforo, facilitando la comprensión y la mejora continua del sistema.

2.5.2. Arquitectura del Sistema IoT

En esta sección se presenta un compendio exhaustivo de los elementos constituyentes del sistema (IoT), acompañado de una exposición detallada acerca de sus interrelaciones, con el propósito de instaurar un entorno integrado de manera coherente.

La concepción del sistema se cristaliza mediante la coexistencia de los siguientes elementos:

Raspberry Pi 3 modelo B: Desempeñando un papel central en el sistema, este dispositivo alberga el código fuente que rige el control de aforo de personas. Funciona como el núcleo para el procesamiento e interpretación de los datos adquiridos a través del sensor térmico AMG8833, presentándose en una forma visualmente comprensible. Adicionalmente, cumple con la tarea de recolectar los datos concernientes a la calidad del aire, los cuales son proporcionados por el sensor MQ-135. Estos datos se transmiten posteriormente a una base de datos en la nube para su almacenamiento y persistencia, además, de ser visualizados a través de una plataforma de IoT.

Sensor térmico AMG8833: Actuando en calidad de sensor primordial, este dispositivo es responsable de la detección de individuos que transitan hacia el salón de estudio, basándose en las lecturas térmicas.

NodeMcu (ESP8266): Cumpliendo el rol de sensor, el NodeMcu monitorea la concentración del aire en el interior del espacio designado para el estudio.

Base de datos (MongoDB): Este entorno constituye la plataforma designada para el almacenamiento de los datos acumulados durante las diversas instancias de estudio.

DashBoard (HomeAssistant): Operando como una interfaz gráfica, esta plataforma IoT se encarga de presentar de manera visual los datos de mayor relevancia, facilitando su análisis y permitiendo un nivel de control más efectivo.

En la ilustración 14 se puede apreciar una representación panorámica de la arquitectura general del sistema, detallando la interconexión y la integración sinérgica de los diferentes componentes involucrados.

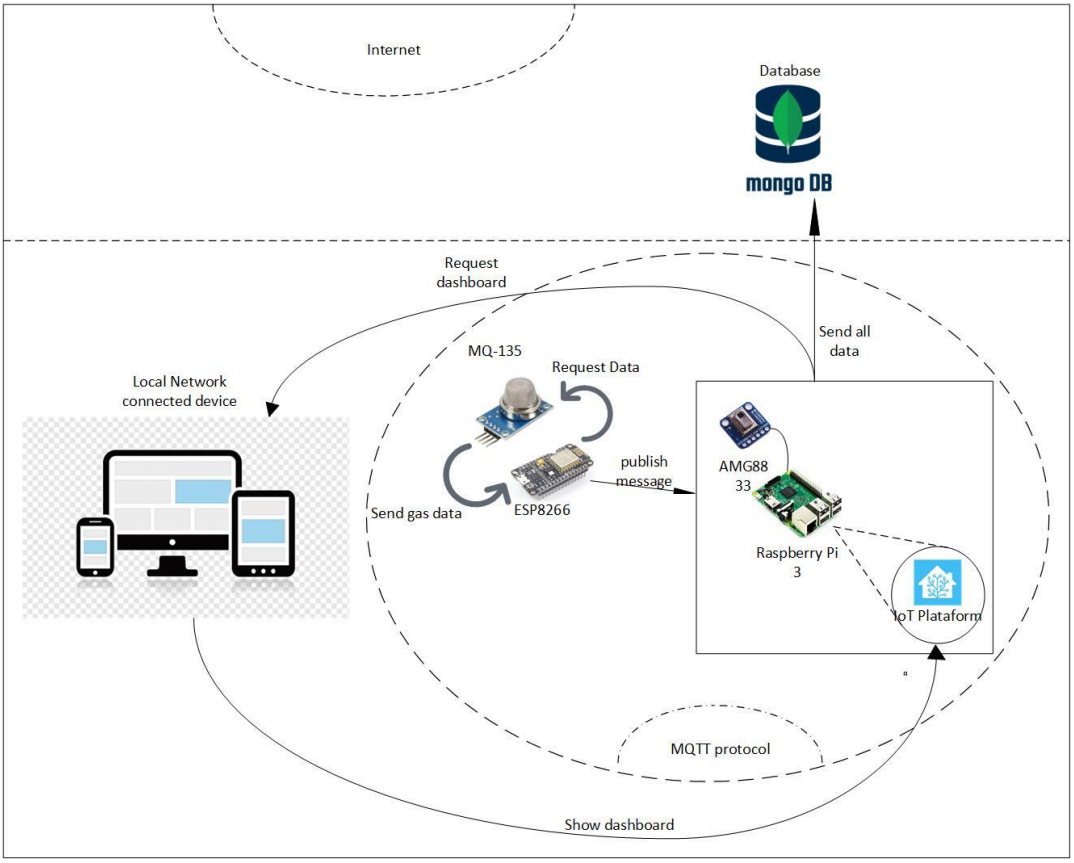

Ilustración 14 Arquitectura del Software 2.5.2.1

3. IMPLEMENTACIÓN

3.1. Entorno de Desarrollo

Como se ha mencionado con anterioridad, la Raspberry Pi se considera la pieza central en el desarrollo completo del prototipo del sistema de control de Aforo debido a su amplia gama de capacidades. Por esta razón, se busca aprovechar al máximo las herramientas proporcionadas por el software Rasbian. En este contexto, el entorno de desarrollo principal seleccionado es Thonny, el cual está especialmente diseñado para trabajar de manera eficaz con la plataforma Raspberry Pi. Thonny se destaca por su simplicidad y facilidad de uso, y está principalmente orientado al lenguaje de programación Python.

Este IDE incluye herramientas esenciales para la depuración y gestión de paquetes, facilitando la identificación y corrección de errores en el código, así como la instalación y actualización de bibliotecas y módulos externos, una funcionalidad crucial para la interacción entre los diversos dispositivos involucrados en la adquisición y representación de datos, así como para su comunicación. Además, Thonny simplifica la manipulación de los pines GPIO de la Raspberry Pi y proporciona recursos educativos, como un shell interactivo de Python y un visor de variables, lo que permite mostrar en tiempo real la información del sensor térmico.

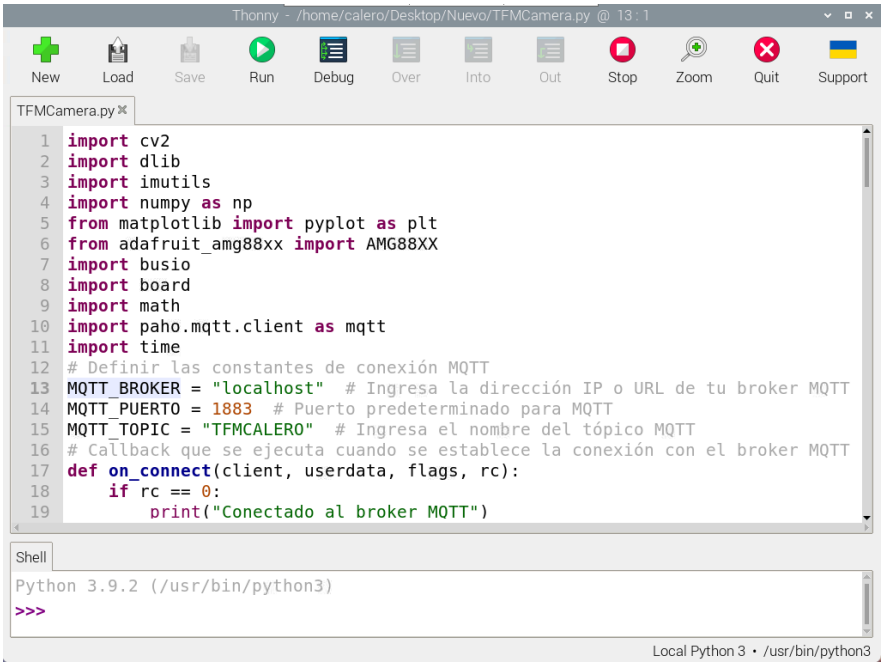

Ilustración 15 Thony 3.1.1

La IDE de Arduino es un entorno de desarrollo de código abierto altamente accesible y versátil, diseñado para programar microcontroladores de manera intuitiva. Su popularidad radica en su interfaz amigable, amplia comunidad de usuarios y extensa biblioteca de soporte, que simplifican la programación y depuración de dispositivos electrónicos como la placa ESP8266. Se utiliza la IDE de Arduino para programar una placa ESP8266 y poder obtener los valores numéricos del sensor MQ-135 y a su vez comunicarlo por medio del protocolo de comunicación MQTT.

La elección lógica se plantea debido a la facilidad de integración de bibliotecas, su flexibilidad para manejar múltiples periféricos e implementación de protocolos de comunicación, así como su amplio soporte para la conexión a Internet, lo que permite una rápida implementación de proyectos IoT como este.

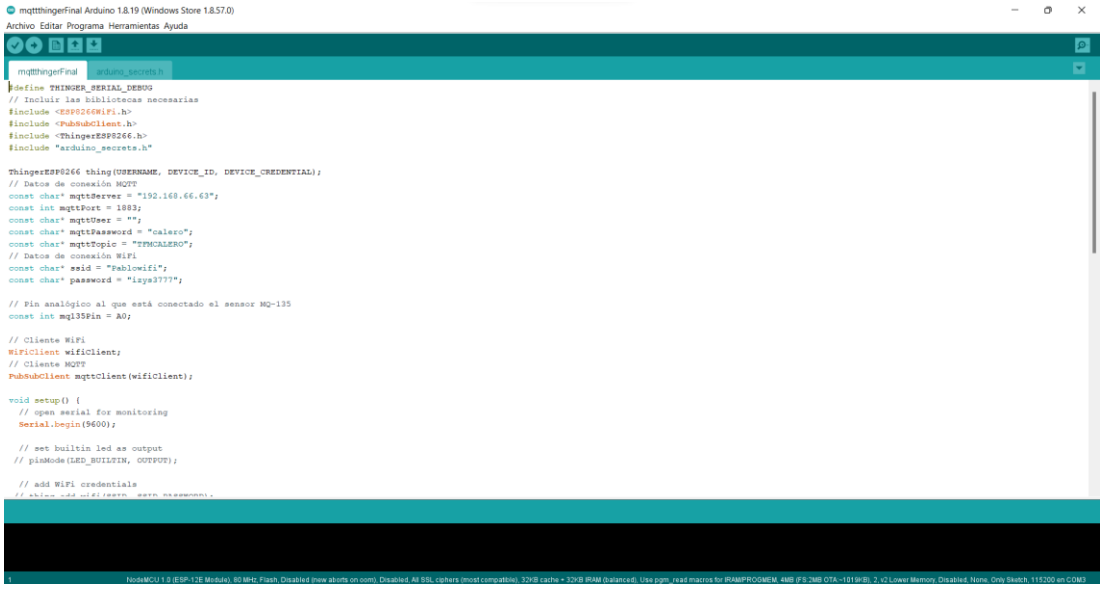

Ilustración 16 IDE Arduino 3.2.2

3.2. librería AMG88XX

La librería adafruit_amg88xx, desarrollada por Adafruit Industries, emerge como una herramienta de considerable relevancia en la adquisición y procesamiento de datos termográficos mediante el empleo del sensor de matriz de infrarrojos AMG88xx en la plataforma Raspberry Pi. A pesar de que el AMG88xx no tiene la capacidad de capturar imágenes visuales en el sentido tradicional, su capacidad para detectar y generar matrices de temperatura mediante la medición de la radiación infrarroja emitida por objetos, es lo que ha permitido a este sensor utilizarlo como una solución

de vanguardia en el ámbito de la termografía. La secuencia de pasos que comprende la implementación de esta librería implica: la instalación de la misma en el entorno de desarrollo de la Raspberry Pi, la correcta conexión física del sensor al dispositivo, la elaboración de un script personalizado para la recopilación de datos de temperatura, y finalmente, el análisis y visualización de dichos datos a través de bibliotecas de procesamiento de datos en Python, como OpenCV. De esta manera, se pretende aprovechar las capacidades del AMG88xx en aplicaciones de detección y monitoreo de temperatura para el prototipo de sistema de aforo (Torres).

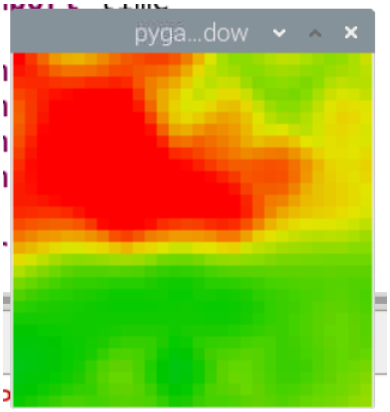

Ilustración 17 librería Amg88xx de Adafruit 3.2.1

3.3. Control de Aforo

En el desarrollo central del código fuente que impulsa el prototipo del sistema de control de aforo mediante una cámara térmica, se ha empleado la biblioteca adafruit_amg88xx, previamente abordada en la sección anterior, en conjunción con la biblioteca OpenCV y otros módulos y bibliotecas de Python. Este enfoque posibilita la manipulación y procesamiento de las imágenes térmicas provenientes del sensor AMG8833, aprovechando los pilares fundamentales de la visión por computadora y el procesamiento de imágenes. Este enfoque permite la detección de objetos que ingresan o salen de una región específica de interés. Además, el programa ha sido configurado para establecer comunicación con un broker MQTT (Message Queuing Telemetry Transport), facilitando así la transmisión y recepción de mensajes relacionados con las detecciones de entrada y salida.

3.3.1. Importación de bibliotecas:

- cv2: Biblioteca OpenCV para procesamiento de imágenes.
- dlib: Biblioteca para realizar tareas de visión por computadora y aprendizaje profundo.
- imutils: Proporciona funciones útiles para simplificar las operaciones de OpenCV.
- numpy y math: Bibliotecas para operaciones numéricas y matemáticas.
- matplotlib.pyplot: Utilizado para mostrar gráficos y visualizaciones.
- adafruit_amg88xx: Biblioteca para comunicarse con la cámara térmica AMG88XX.
- busio y board: Bibliotecas para comunicación I2C con el sensor.
- paho.mqtt.client: Biblioteca para implementar la funcionalidad de MQTT.
- pymongo: Biblioteca para enviar los datos a la base de datos MongoDB

3.3.2. Configuración de conexión MQTT:

- Se definen las constantes MQTT_BROKER, MQTT_PUERTO y MQTT_TOPIC para configurar la conexión con el broker MQTT.
- Se definen funciones de callback (on_connect y on_message) que se ejecutan cuando se establece la conexión y se recibe un mensaje MQTT.
- Se crea un cliente MQTT y se establecen los callbacks.
- El cliente MQTT se conecta al broker utilizando la dirección IP, puerto y tiempo de espera especificados.

3.3.3. Definición de funciones de envío de mensajes:

send_entrada(): Publica un mensaje MQTT para indicar una entrada detectada.

send_salida(): Publica un mensaje MQTT para indicar una salida detectada.

3.3.4. Configuración de la cámara térmica:

Se crea un objeto I2C para comunicarse con la cámara AMG88XX.

Se configura una ventana de visualización de OpenCV llamada "Thermal Camera", la cual nos permite utilizar las herramientas de OpenCV para realizar un correcto procesado de imágenes.

3.3.5. Main:

En un bucle infinito (while True), donde se obtienen lecturas de temperatura del sensor. Las lecturas se convierten en una imagen de escala de grises y se redimensionan para una visualización amigable.

Se aplica una máscara en el rango de temperaturas para resaltar objetos en función de la temperatura deseada como se puede apreciar en la ilustración 18.

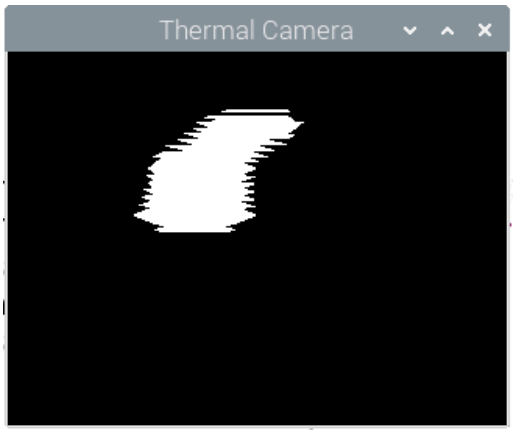

Ilustración 18 Resaltar Umbral 3.3.5.1

Se calcula la mitad de la altura y el ancho de la imagen para referencia, esto nos ayudará a delimitar las zonas de entrada y salida de personas, además de obtener regiones de interés para procesar.

Como se puede observar en la ilustración 19, se dibuja una línea roja en la mitad de la imagen para marcar la separación entre las zonas de entrada y salida. De esa forma, se puede apreciar la dirección de flujo.

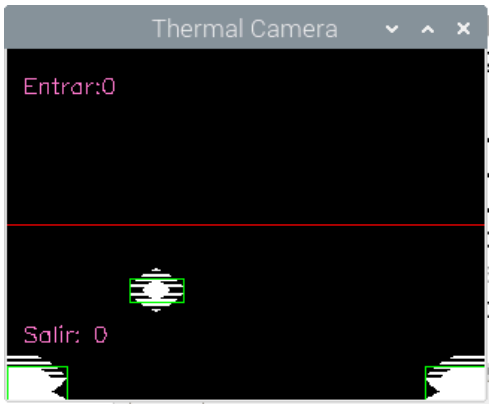

Ilustración 19 Separación región de interés 3.3.5.1

3.3.6. Detección de objetos y publicación MQTT:

Se definen regiones de interés (ROI) para la zona de entrada y salida, como se muestra en las ilustraciones 20 y 21 respectivamente.

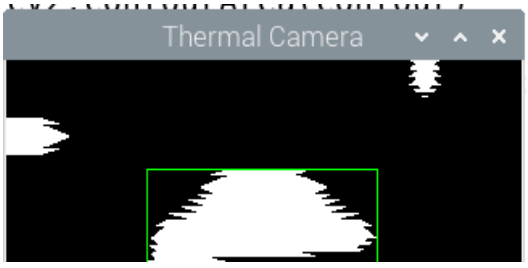

Ilustración 20 Región de interés de Entrada 3.3.5.2

Ilustración 21 Región de interés salida 3.3.5.3

Cuando un objeto cruza una posición específica en la zona de entrada o salida, se actualizan los contadores auxiliares y se envían mensajes MQTT correspondientes, como se puede apreciar en la ilustración 22.

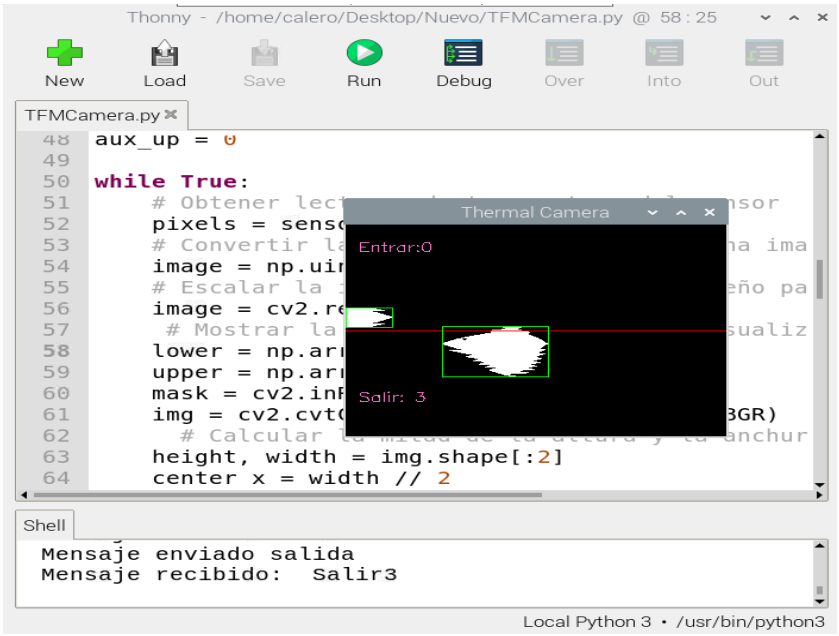

Ilustración 22 Visualización del contador 3.3.5.4

En el sistema, se incorporan mensajes en la imagen para mostrar el número de entradas y salidas detectadas, y la imagen resultante se presenta en la ventana de visualización. Posteriormente, se ejecuta el bucle de manejo de eventos de MQTT para recibir y procesar mensajes

Por otro lado, el sistema incorpora un sensor de gas MQ-135 para detectar la concentración de gas, el cual está conectado a una placa ESP8266 y se comunica con la Raspberry Pi, que se encuentra integrada con la cámara térmica mediante MQTT.

3.4. Concentración de Aire

El programa está diseñado para un dispositivo ESP8266 que emplea un sensor de gas MQ135 para medir los niveles de dióxido de carbono ambientes cerrados. Los datos se transmiten a través del protocolo de comunicación MQTT.

Configurado con credenciales de red WiFi y detalles del servidor MQTT, el programa se encarga de recopilar mediciones del sensor, como la resistencia y la concentración de CO_2 en partes por millón (PPM), ajustando estos valores según la temperatura y la humedad.

Luego, estos datos son publicados en el servidor MQTT bajo un topic específico. El ciclo de lectura y publicación se repite cada 10 segundos para monitorear continuamente los niveles de CO_2.

Ilustración 23 Programa para concentración de aire 3.4.1

En la Ilustración 23, se puede apreciar la implementación del código que permite obtener las mediciones de concentración de CO_2 en el entorno. A continuación, se detallan los elementos principales que lo componen.

- **Inclusión de bibliotecas**: El programa comienza incluyendo las bibliotecas necesarias para el funcionamiento del ESP8266, el sensor MQ135, la comunicación MQTT y la comunicación serial.
- **Configuración de WiFi y MQTT:** Se establecen las credenciales de la red WiFi a la que se conectará el dispositivo y los detalles del servidor MQTT al que se conectará. En este caso, no se utiliza un usuario MQTT, pero se especifica una contraseña.
- **Configuración del sensor:** Se define el pin análogo al que está conectado el sensor MQ135 (A_0) y se crea una instancia del sensor MQ135. También se establecen valores iniciales para la temperatura y la humedad, que se utilizan en el cálculo de la concentración de CO_2.
- **Configuración del cliente MQTT:** Se crea un cliente MQTT utilizando la biblioteca PubSubClient y se configura para utilizar la conexión WiFi.
- **Función setup():** Se inicializa la comunicación serial y se conecta el ESP8266 a la red WiFi y al servidor MQTT. Esta parte del código asegura que el dispositivo esté conectado a la red y al servidor antes de continuar.
- **Función loop():** Se ejecuta en un bucle continuo. Se obtienen lecturas del sensor MQ135, incluyendo el valor de resistencia, la concentración de CO_2 en partes por millón (PPM) y valores corregidos para la temperatura y la humedad ór medio de la librería MQ135. Se imprimen estos valores en la consola serial para fines de depuración y se publica la concentración corregida de CO_2 en el servidor MQTT en el topic especificado (TFMCALERO), después de convertir el valor en una cadena de texto.
- El programa espera 10 segundos antes de realizar otra lectura y publicación. Esto se hace con delay(10000).

3.5. Control de aforo y concentración de CO2

Esta combinación de componentes permite no solo la detección de objetos que ingresan o salen de una región específica de interés, sino también la monitorización de la concentración de gas en el entorno. Además, el programa ha sido configurado para establecer comunicación con un broker MQTT, lo que facilita la transmisión y recepción de mensajes relacionados con las detecciones de entrada, salida y las mediciones de CO_2 en el ambiente. Este enfoque integral proporciona una solución completa para el control de aforo y la monitorización del entorno en tiempo real.

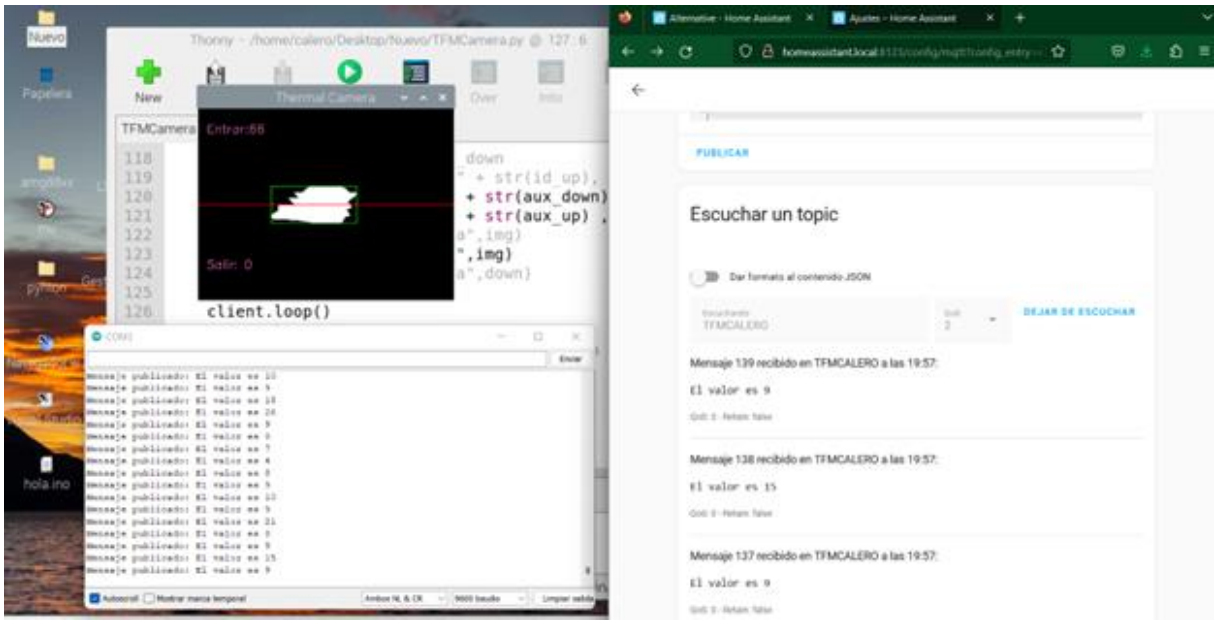

Ilustración 24 Integración y Comunicación de Programas 3.5.1

3.6. Base de datos

El registro de los datos obtenidos en el sistema de control de aforo preserva un historial detallado de las detecciones de entrada y salida de personas, así como la concentración de aire presente. Esta funcionalidad, habilitada mediante el almacenamiento en una base de datos, facilita análisis retrospectivos esenciales para la comprensión de patrones y tendencias en el flujo de personas. Además, la base de datos posibilita la creación de informes y visualizaciones personalizados para comunicar los resultados de cada caso de experimentación. En última instancia, este enfoque no solo asegura la disponibilidad y la integridad de los registros, sino que también potencia la eficacia y la eficiencia operativa del sistema, aportando un marco sólido para la gestión y el análisis de datos en tiempo real y retrospectivo.

3.6.1. Elección de la base de datos

Para este proyecto se ha seleccionado una base de datos de tipo NoSQL (MongoDB) por varias razones fundamentales. En primer lugar, MongoDB es altamente flexible y puede gestionar datos no estructurados o semiestructurados, lo que se alinea perfectamente con la naturaleza de los datos termográficos generados por las cámaras térmicas, que a menudo no tienen una estructura rígida. Además, MongoDB ofrece una escalabilidad horizontal excepcional, lo que permite manejar grandes volúmenes de datos en constante crecimiento, una característica crucial en aplicaciones de monitoreo de aforo donde se espera un aumento de registros con el tiempo. Asimismo, su capacidad para realizar búsquedas y consultas rápidas y flexibles es esencial para analizar datos en tiempo real y obtener información instantánea sobre el flujo de personas. Por último, MongoDB garantiza la alta disponibilidad y la redundancia, lo que asegura la integridad y la continuidad de los datos, un aspecto crítico en sistemas de control de aforo en tiempo real que deben mantenerse operativos sin interrupciones. En conjunto, estas características convierten a MongoDB en una elección estratégica para respaldar la infraestructura de datos de un sistema de control de aforo basado en cámaras térmicas, permitiendo la eficiente gestión, análisis y almacenamiento de datos heterogéneos y en constante evolución.

MongosDb presenta una estructuración de datos en formato JSON lo que la convierte en un componente fundamental para la representación y el intercambio de información relevante. Este enfoque de organización de datos permite una codificación eficiente de los registros térmicos asociados a las detecciones de entrada y salida de personas en un entorno específico. En este sentido, un objeto JSON se utiliza para representar los datos de una detección individual.

La utilización de JSON como formato de estructuración de datos proporciona una base sólida y versátil para la gestión, el análisis y la interoperabilidad de los datos generados por el sistema de control de aforo con cámara térmica.

Colección	Descripción	Tipo de Variable
-id	Identificador único	**ObjectId**
Name	Nombre del evento Identificador - entra o sale un participante	**String**
Número de Personas	Contador de Aforo	**Int32**
Date	Fecha y hora	**ISODate(UTC)**
Aire	Valro obtenido del sensor MQTT concentración de aire	**Int32**

Tabla 11 Colección datos JSON 3.6.1.1

En la ilustración 25, se puede apreciar que cada documento representa un registro de aforo en una fecha y hora determinada ("fecha"). Dentro de cada registro de aforo, hay un array llamado "detecciones", que contiene objetos JSON representando cada detección individual. Cada objeto de detección incluye información de si es una entrada o una salida, y la marca de tiempo de la detección. MongoDB asigna automáticamente un campo "_id" único a cada documento si no se proporciona uno. Esto se utiliza para identificar de manera única cada registro en la colección. La "fecha" se almacena en el formato de fecha y hora ISO para facilitar el análisis de tiempo.

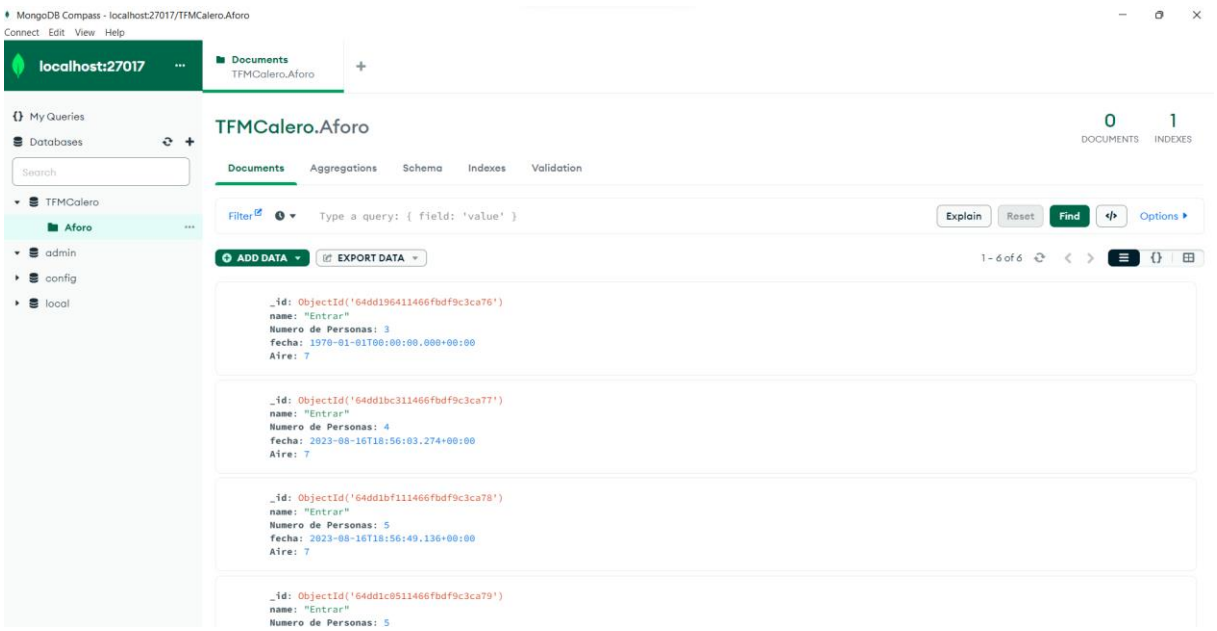

Ilustración 25 Base de Datos MongoDB 3.6.1.2

3.7. Plataforma IoT

La plataforma IoT(internet of things) ha revolucionado nuestra forma de interactuar con el mundo que nos rodea y ha transformado fundamentalmente la

manera en que las organizaciones operan en diversos sectores. Esta tecnología se basa en la idea de conectar una amplia gama de dispositivos y sensores a través de internet, permitiéndoles comunicarse y compartir datos de manera eficiente y en tiempo real.

Estos datos pueden representarse visualmente a través de gráficos, tablas de mando y otros tipos de representaciones visuales. Estas representaciones permiten una comprensión más profunda de los datos, una toma de decisiones más informada y una mejora significativa en la eficiencia y la experiencia del usuario en una variedad de aplicaciones, desde la industria hasta la vida cotidiana.

3.7.1. Elección de la Plataforma

Existen un gran número de plataformas que permiten conectar tus dispositivos e interactuar con sus servicios. Sin embargo, la elección de la plataforma puede convertirse en un desafío significativo. Al tomar una decisión, es importante considerar factores como la facilidad de uso, la disponibilidad de aplicaciones, y, lo más importante, cómo se alinea con tus objetivos y preferencias personales.

En este proceso de exploración, inicialmente se optó por Thinger.Io debido a su destacada facilidad de conexión. Sin embargo, pasado el período de prueba en las cuentas gratuitas, comenzaron las limitaciones relacionadas con los protocolos de comunicación MQTT.

En busca de una solución más versátil, se decidió utilizar la plataforma Home Assistant. Esta elección se basó en su capacidad para proporcionar una amplia gama de servicios para la conexión de dispositivos IoT. Lo que realmente destaca de Home Assistant es su flexibilidad, que permite a usuarios con diferentes niveles de experiencia, desde aquellos con conocimientos limitados hasta aquellos con más experiencia en programación, sacar el máximo provecho de la plataforma.

3.7.2. Desarrollo de la dashboard en Home Assistant:

Node-RED, parte integral de la plataforma Home Assistant, se destaca como un servicio versátil. Esta plataforma de programación visual de código abierto simplifica la creación de aplicaciones IoT al permitir la interconexión de nodos en un flujo de trabajo visual, donde los datos se desplazan de un nodo a otro.

El sistema de control de aforo se integra con Node-RED a través de MQTT. Como se observa en la ilustración 26, los nodos MQTT capturan los datos de las entradas y salidas, mientras que los nodos de cadena procesan esta información. Los resultados se presentan en tiempo real mediante nodos de gráficos y texto.

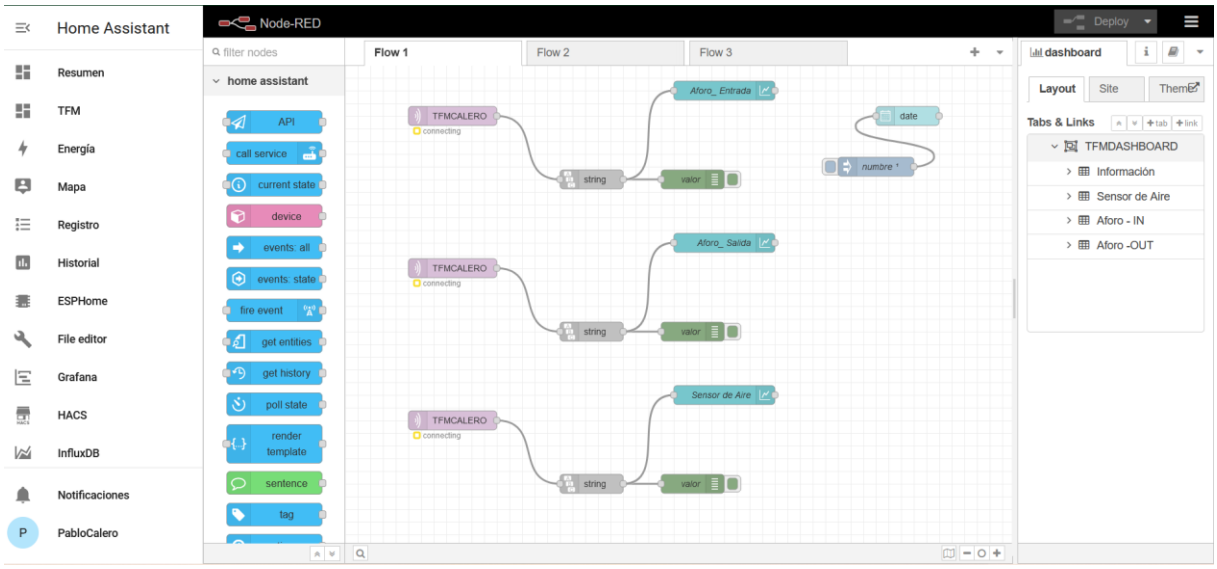

Ilustración 26 NodeRED 3.7.2.1

La facilidad de conexión entre estos nodos en un flujo de trabajo visual y la capacidad de configurar notificaciones y acciones adicionales hacen que Node-RED sea una solución eficaz y flexible para la implementación y supervisión de datos en el sistema de control de aforo. El diseño final se puede visualizar en la ilustración 27.

Ilustración 27 Diseño Final Dashboard 3.7.2.2

El diseño de esta plataforma se basa en la idea de proporcionar información de manera clara y concisa, aprovechando al máximo la disposición de cuatro cuadrantes principales.

Primer cuadrante: Información del flujo de personas

En el primer cuadrante, se encuentra la información esencial sobre el flujo de personas. Este cuadrante actúa como el centro de control, mostrando datos puntuales sobre la cantidad de personas presentes en un área específica. Para garantizar una comunicación efectiva, se utiliza el protocolo MQTT, que confirma la entrega exitosa de mensajes. Esto asegura que la información crítica llegue de manera oportuna y precisa a quienes la necesitan.

Segundo cuadrante: Medida del sensor MQ-135

El segundo cuadrante se dedica a mostrar la medida obtenida por el sensor MQ-135. Este valor, que varía de 350 a 1500, representa la concentración de CO_2, expresada en partes por millón (ppm). Proporcionar esta información es crucial, ya que permite a los usuarios evaluar la calidad del aire en tiempo real.

Tercer y cuarto cuadrante: Representación gráfica del flujo de personas

Los cuadrantes 3 y 4 están dedicados a representar gráficamente el flujo de personas que ingresan y salen de un espacio. Estos gráficos muestran visualmente la evolución del flujo de personas a lo largo del tiempo. Además, estos cuadrantes indican la hora exacta en la que ocurrió cada entrada y salida. Esta información es valiosa para el análisis de patrones de comportamiento y puede ser utilizada para tomar decisiones informadas sobre la gestión de espacios y recursos.

4. VALIDACIÓN Y EVALUACIÓN

En la presente sección, se focaliza la atención en la exhaustiva evaluación integral del sistema en cuestión, así como de sus funcionalidades correlacionadas. El propósito primordial consiste en someter a un análisis riguroso la operatividad y eficacia inherentes al sistema en su conjunto.

En una primera instancia, se procedió a llevar a cabo una evaluación referente al objetivo central del sistema: el registro preciso del flujo de individuos que acceden a un espacio predeterminado. A través de esta evaluación, se aspira a poner a prueba el prototipo primario en función de su capacidad para identificar el flujo de individuos que ingresan y salen de un entorno cerrado bajo circunstancias controladas. Esta evaluación, no únicamente satisface la necesidad de validar la funcionalidad inherente al sistema, sino que también sienta las bases sólidas para su iteración y mejoras continuas.

Consecuentemente, se procedió a llevar a cabo pruebas meticulosamente planificadas que incorporan la comparación entre el número efectivo de individuos que acceden a un salón de estudio específico. Dicha comparativa se desarrolló paralelamente a la supervisión de la concentración de aire presente en dicho entorno, estableciendo una relación directa con el número de individuos presentes. A través de este conjunto de pruebas controladas, se persigue la comprensión de la correlación existente entre el número de personas presentes en un entorno cerrado y la calidad del aire interior.

Esta fase de evaluación representa un pilar fundamental en el desarrollo del sistema, ya que proporciona una comprensión más profunda de su operatividad en escenarios de aplicación real. Las pruebas comparativas no solo brindan la capacidad de identificar posibles limitaciones en el sistema, sino que también revelan áreas susceptibles de optimización. Asimismo, al considerar la interdependencia entre el flujo de individuos y la calidad del aire, se aborda un aspecto crítico relacionado con la seguridad y el bienestar de los ocupantes del espacio.

4.1. Resultados Obtenidos

El experimento realizado tenía como objetivo evaluar la efectividad de un sistema de detección ubicado sobre una puerta en un entorno controlado. Para ello, se llevaron a cabo dos fases distintas, cada una con sus propias consideraciones y resultados.

En la primera fase, se colocó el circuito de detección sobre la puerta a una distancia de 450 mm, asegurando que estuviera fuera del alcance y no pudiera ser manipulado para evitar daños. Se destacó que las dimensiones de la puerta eran de 2000 mm de alto y 1500 mm de ancho. Bajo estas condiciones iniciales, se observó que el sistema era capaz de detectar con éxito a personas cuya altura era de 1800 mm, pero presentaba dificultades en la detección cuando se trataba de personas más bajas que esta altura de referencia. Este resultado indicaba que el sistema tenía limitaciones en su capacidad para detectar objetos de menor tamaño.

Ilustración 28 Primer Emplazamiento cámara 4.1.1

Ante esta observación, se tomó la decisión de mejorar el sistema. En la segunda fase del experimento, el circuito de detección fue reubicado en el marco de la puerta, es decir, a cota cero, teniendo en cuenta el valor de altura de la propia puerta. Esta nueva ubicación permitía al sistema tener una perspectiva más adecuada para detectar objetos de diferentes alturas. Como resultado de esta reubicación, se logró mejorar significativamente la capacidad del sistema para detectar objetos de menor tamaño. Ahora, el sistema puede detectar a cualquier individuo de altura superior a los 800 mm, lo que indicaba una mejora sustancial en su rango de detección.

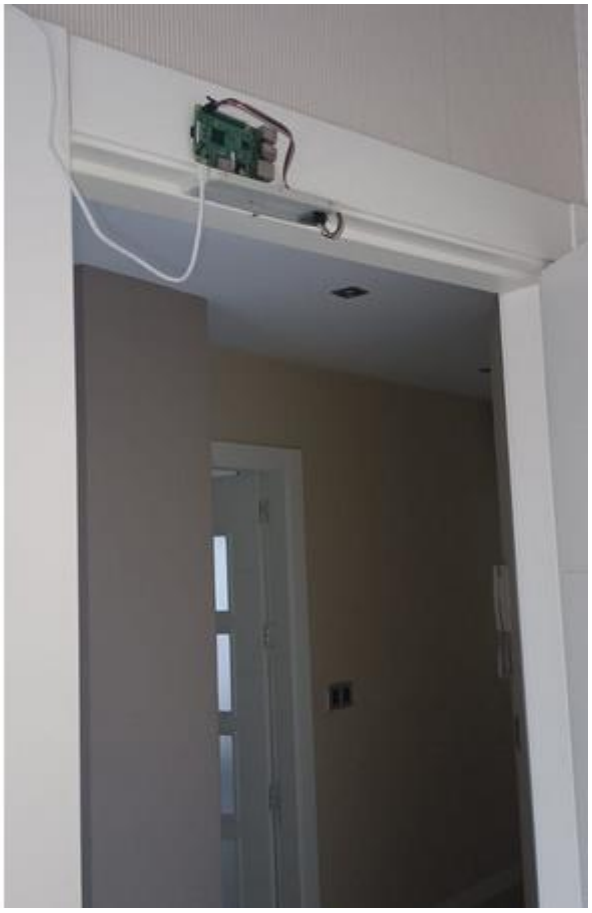

Ilustración 29 Emplazamiento Final cámara 4.2.2

Una vez que el sistema estaba posicionado en condiciones óptimas para la detección de objetos, se procedió a la siguiente fase del experimento: evaluar el flujo de individuos a través de la puerta. Esto implicaba contabilizar tanto el ingreso como la salida de personas a través de la puerta. El sistema de detección tenía la tarea de registrar con precisión cada vez que una persona pasaba por la puerta, ya fuera entrando o saliendo. Esto proporcionaba información valiosa sobre la afluencia de personas y su comportamiento de movimiento en ese entorno específico.

4.2. Caso de estudio 1 - Control de flujo

Con la finalidad de validar el objetivo inicial del prototipo, la primera etapa de evaluación se centra en poner a prueba la precisión del sistema en el registro del flujo de personas que ingresan y salen del espacio cerrado. Este proceso implica la implementación de una entrada y salida controlada de personas, seguida de una meticulosa comparación entre los datos capturados por el sistema y los eventos observados en la realidad

4.2.1. Control de entrada

Se llevó a cabo una evaluación exhaustiva del prototipo, utilizando pruebas de flujo que se centraron específicamente en el ingreso de personas en un entorno interior o salón cerrado. Los resultados de esta evaluación se presentan a través de dos ilustraciones clave, la ilustración 21 y la ilustración 22, que ofrecen una visión clara de los logros y el funcionamiento del sistema.

La ilustración 30 se presenta como una imagen representativa de los casos de éxito que ha experimentado nuestro sistema durante las pruebas. Esta imagen destaca cómo el sistema ha logrado gestionar y facilitar eficazmente el ingreso de personas en el entorno de un salón cerrado. Los resultados exitosos en esta etapa de prueba son un indicativo sólido de la capacidad del prototipo para abordar este escenario específico de manera efectiva.

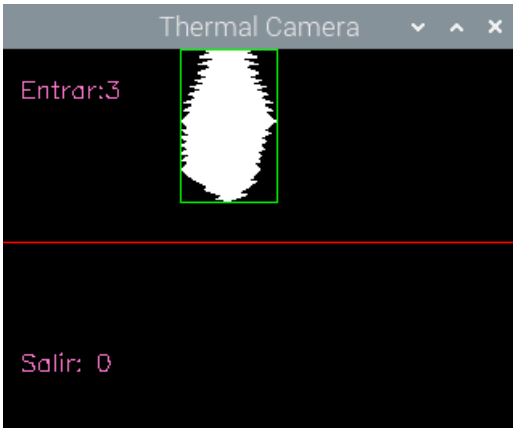

Ilustración 30 Registro de entrada participantes 4.2.1.1

Por otro lado, la ilustración 31 aporta una valiosa perspectiva sobre la interacción del sistema con la plataforma IoT. Esta imagen demuestra claramente que el prototipo está en perfecta armonía con los datos que se generan y visualizan a través de la

plataforma IoT. Esto es fundamental ya que la sincronización efectiva entre el sistema y la plataforma IoT es esencial para garantizar un funcionamiento suave y una recopilación de datos precisa.

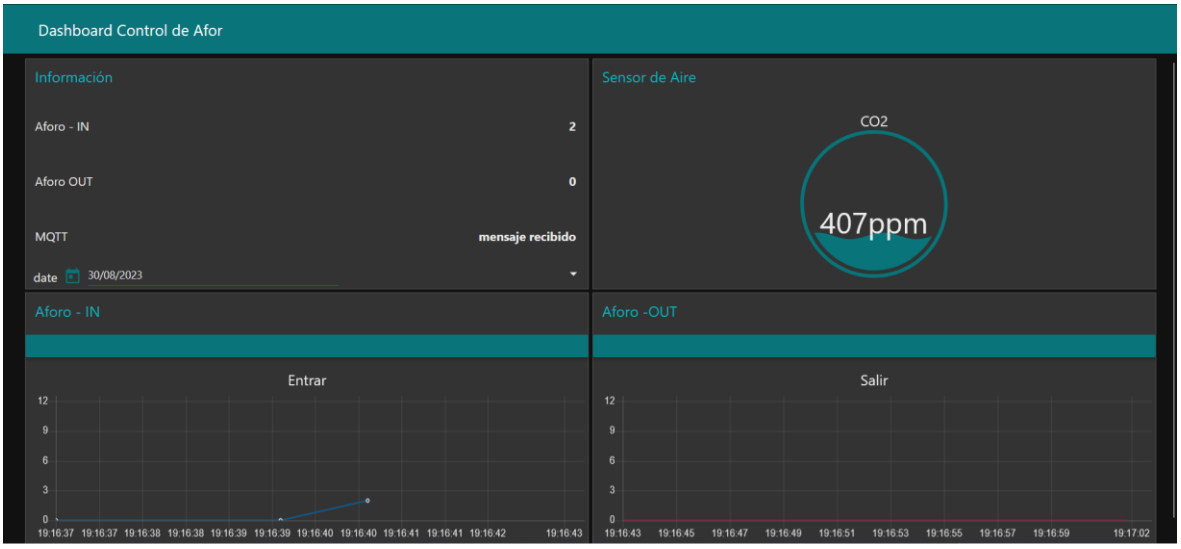

Ilustración 31 Control ingreso Plataforma IoT 4.2.1.2

4.2.2. Control de Salida

De manera análoga, las ilustraciones 32 y 33 capturan visualmente los momentos en los que nuestro sistema ha logrado gestionar con éxito la salida de personas de un entorno particular. Son representativas de los resultados positivos obtenidos en este aspecto crítico. Al igual que en el caso anterior, estas imágenes brindan una visión gráfica de los logros de nuestro sistema en el contexto del flujo de salida de personas.

Ilustración 32 Control de Salida por plataforma IoT 4.2.2.1

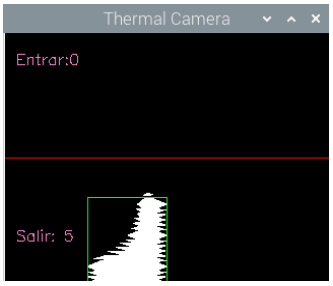

Ilustración 33 Registro Salida Participantes 4.2.3.2

4.2.3. Resultados obtenidos

Los resultados obtenidos, como se detallan en la Tabla 6, ofrecen una evaluación de la eficacia del prototipo en comparación con la detección manual en relación con el seguimiento de la entrada y salida de participantes en diversos ensayos. Cada entrada en la tabla corresponde a un ensayo específico y presenta el número de participantes que ingresaron y salieron, así como el número de estas transacciones que fueron correctamente identificadas por el sistema.

Ensayos	Número de participantes que ingresaron	Ingresos detectados por el sistema	Número de participantes que salieron	Salidas detectadas por el sistema
1	1	1	1	1
2	3	3	2	2
3	5	4	4	4
4	1	1	5	5
5	2	2	9	9
6	8	8	5	7
7	5	5	2	2
8	2	2	1	0
9	3	3	6	6
10	4	4	8	8
11	5	5	4	6
12	9	8	6	6
13	4	4	3	3
14	5	5	9	9
15	7	5	6	6
16	6	6	3	3
17	3	3	2	2
18	3	3	5	5
19	9	9	5	5
20	11	10	3	3

Tabla 12 Evaluación Aforo 4.2.3.1

Para una síntesis concisa de estos resultados, la Tabla 7 y la Ilustración 34 proporcionan un resumen global de la efectividad del prototipo. Según esta recopilación, el prototipo logró una tasa de detección del 80.50%, indicando que la mayoría de detecciones se realizaron con precisión. No obstante, se evidenció un margen de error del 12.50% en la detección.

Total ensayos	100%	40
Total de aciertos	87.50%	35
Total de fallos	12.50%	5

Tabla 13 Resumen de Resultados Aforo 4.2.3.2

En términos específicos, en un conjunto de 40 ensayos, el sistema registró un total de 35 aciertos y 5 fallos. Este conjunto de resultados sugiere un alto grado de eficacia en la detección de movimientos en condiciones controladas y óptimas.

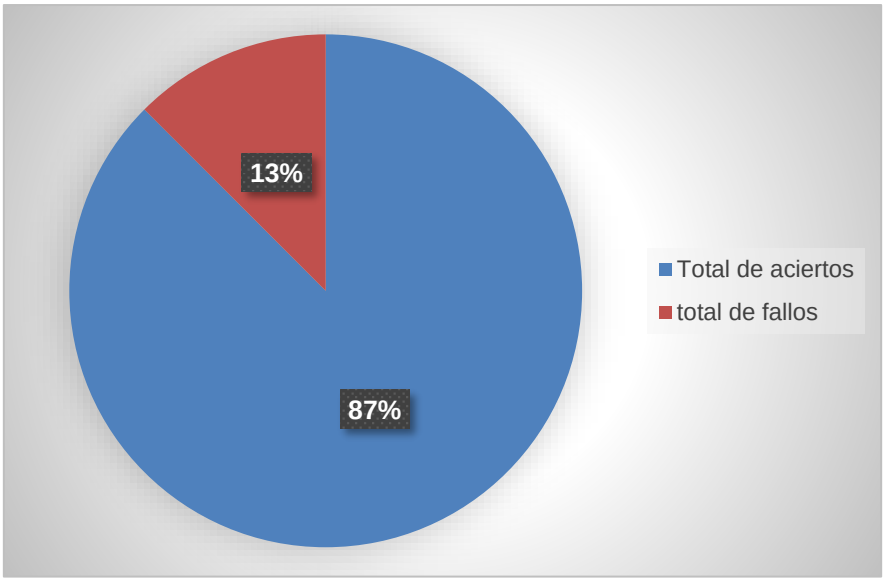

Ilustración 34 Porcentaje de Acierto 4.2.3.2

4.3. Caso de estudio 2 - Relación entre el Número de Individuos y la Concentración de Aire en Espacios Cerrados

En este caso de experimentación, se evalúa la interacción que existe entre la concentración de CO_2 y el número de individuos que ocupan un espacio cerrado. El experimento se llevó a cabo en una habitación de dimensiones reducidas, con el propósito de emular condiciones que comúnmente se encuentran en situaciones cotidianas. Se estableció un límite máximo de 10 personas en el interior del espacio de estudio.

Es importante mencionar que la concentración de CO_2 en una habitación generalmente mantiene una alta calidad del aire cuando los niveles no superan los 350 ppm. Entre 350 y 500 ppm, la calidad del aire en el interior se considera buena, mientras que entre 500 y 700 ppm, la calidad del aire se considera moderada. Cuando los niveles de CO_2 oscilan entre 800 y 1200 ppm, la calidad del aire se vuelve baja, por lo que no se recomienda mantener una exposición continua a niveles superiores a 700 ppm durante más de 24 horas. Por encima de los 1200 ppm, la calidad del aire se considera mala. (Universidad Complutense Madrid, 2020)

4.3.1. Aforo con 2 personas

En la ilustración 35, se puede observar que, con la presencia de solamente dos individuos, la concentración de aire mantiene un rango de medición constante en el sensor de aire, que se encuentra establecido entre los 405 y 420.95 ppm. Esto implica que, en esta configuración inicial, la presencia de dos personas en el espacio no ocasiona alteraciones significativas en la concentración del aire.

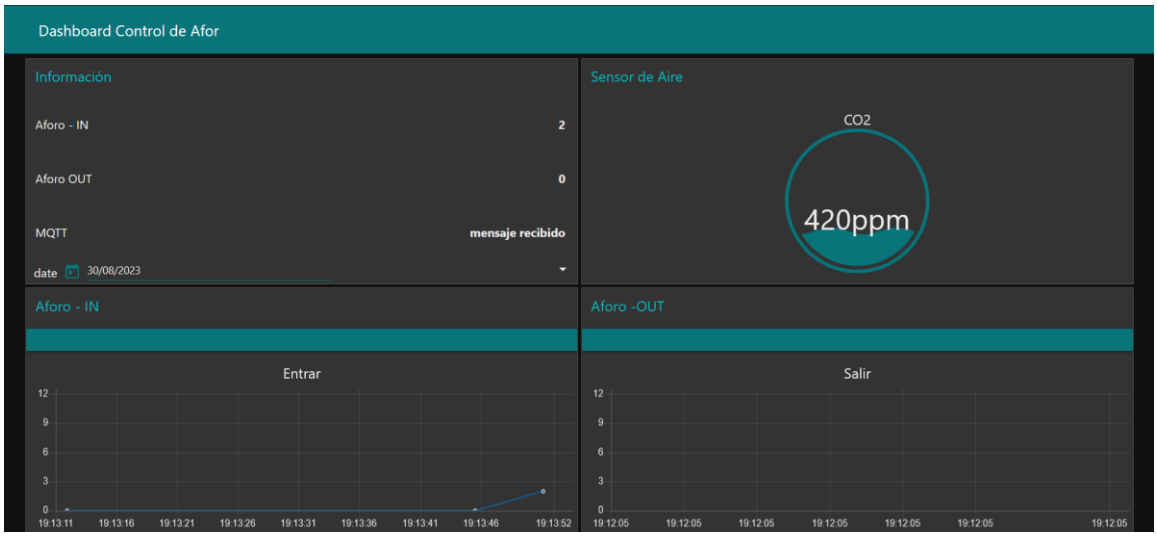

Ilustración 35 Concentración de Aire - 2 participantes 4.3.1.1

4.3.2. Aforo con 5 personas

A medida que se aumenta la cantidad de individuos dentro de la habitación a un total de cinco, como se observa en la ilustración 36, se evidencia un aumento en la concentración de CO_2. La concentración aumenta a 705.12 ppm. Este incremento de CO_2 es indicativo que la presencia de más personas en el espacio cerrado puede tener implicaciones significativas para la calidad del aire y, por ende, para la salud y comodidad de los ocupantes.

Ilustración 36 Concentración de Aire - 5 participantes 4.3.2.1

4.3.3. Aforo con 10 personas

Finalmente, en el escenario en el cual el espacio se encuentra ocupado por la máxima capacidad permitida de 10 personas, como se refleja en la ilustración 37, los resultados obtenidos en las evaluaciones revelan un aumento sustancial, se observa que el valor máximo de concentración de aire alcanzado es de 812.23 ppm. Con una mayor densidad de ocupantes, la calidad de aire disminuye considerablemente debido al metabolismo y la respiración de las personas, Por lo que se recomienda ventilar y limitar el número de ocupantes.

Ilustración 37 Concentración de Aire - 10 participantes 4.3.3.1

4.3.4. Resultados obtenidos

Se realizaron varias evaluaciones intentando emular situaciones cotidianas de control de aforo con objetivo de observar la relación entre la concentración de aire y el número de participantes en un espacio cerrado. Se realizaron múltiples evaluaciones utilizando un sensor de gas para medir la concentración de aire. Los resultados revelaron variaciones en los valores del sensor, lo que condujo a un análisis estadístico para comprender la distribución normal de los datos. Este análisis permitió determinar las posibles concentraciones de aire en función de 2, 5 y 10 participantes.

4.3.4.1. Aforo con 2 personas.

Se tomaron 10 muestras. La media de los datos muestreados fue de 410.66 ppm, con una desviación estándar de 5.74. Esto indica que la concentración de aire más probable con un aforo de 2 personas oscila entre el 404.92 y 416.4 ppm em 68% de los casos.

Aforo con 2 personas		En rango de mediciones		
Mediciones	CO_2 (ppm)	medidas	Distribución	frecuencia
1	418	400	0.01217093	1
2	411	401	0.01663569	1
3	419	402	0.02205	4
4	413	403	0.02834178	4
5	408	404	0.03532617	6
6	410	405	0.0426989	6
7	412	406	0.05004811	7
8	402	407	0.05688652	8
9	414	408	0.06270207	7
10	410	409	0.06702012	7
11	420	410	0.06946712	7
12	415	411	0.06982391	10
13	407	412	0.0680581	10
14	411	413	0.06432892	9
15	402	414	0.05896353	9
16	414	415	0.05240968	8
17	400	416	0.04517418	7
18	417	417	0.03775894	7
19	404	418	0.03060555	6
20	406	419	0.02405644	5
21	411	420	0.01833637	5
Media	410.666667			
Desv. Est	5.7037999			
Max	420			
Min	400			

Tabla 14 Resultados concentración de aire - 2 participantes 4.3.4.1.1

Esto se puede evidenciar claramente en la ilustración 38 que muestra una distribución normal de los datos de los valores que obtenidos.

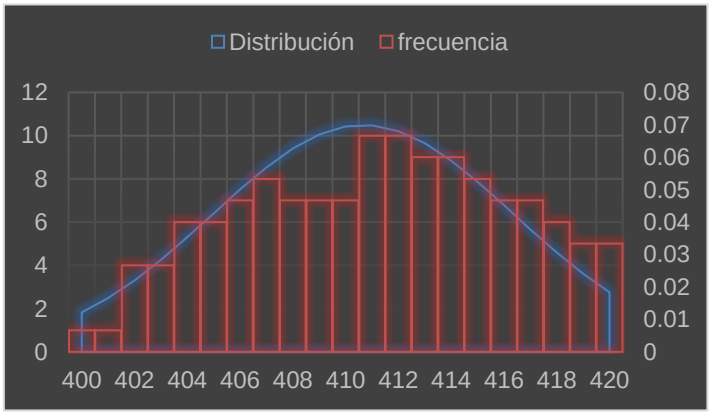

Ilustración 38 Distribución normal - 2 participantes 4.3.4.1.1

4.3.4.2. Aforo con 5 personas

Con un adoro de 5 personas, los resultados mostraron una media de 580.85 ppm y una desviación estándar de 75.20. Esto implica que en un 68% de los casos, la concentración de aire se encuentra en el rango de valores típicos de 54.8 a 655.20 ppm.

Aforo con 5 personas		**En rango de mediciones**		
Mediciones	CO_2 (ppm)	Rango	Distribución	frecuencia
1	613	400	0.00029439	0
2	512	415	0.00046621	0
3	623	430	0.00070952	1
4	674	445	0.00103769	1
5	626	460	0.00145848	3
6	510	475	0.00196994	3
7	679	490	0.002557	5
8	639	505	0.00318958	6
9	684	520	0.0038235	6
10	452	535	0.00440467	7
11	668	550	0.00487628	8
12	658	565	0.00518788	9
13	539	580	0.00530413	9
14	492	595	0.00521151	10
15	633	610	0.00492082	9
16	562	625	0.00446515	9
17	478	640	0.00389367	8
18	524	655	0.00326292	7
19	490	670	0.00262771	6
20	548	685	0.00203364	6
21	594	700	0.00151249	5
Media	580.857143			
Desv. Est	75.2085671			
Max	684			
Min	452			

Tabla 15 Resultados concentración de aire - 5 participantes 4.3.4.2.1

La ilustración 39 ilustra esta distribución de valores obtenidos.

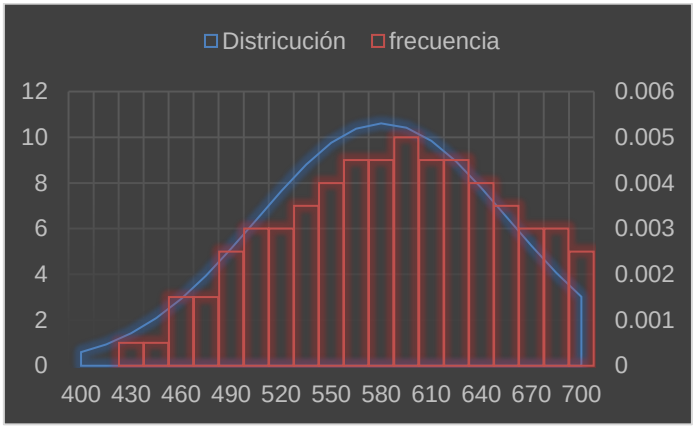

Ilustración 39 Distribución normal - 5 participantes 4.3.4.2.1

4.3.4.3. Aforo con 10 personas

Para 10 participantes, la media de los datos muestreados fue de 712.81 ppm, con una desviación estándar del 62.19. Esto sugiere que la concentración de aire típica con 10 participantes se encuentra en el intervalo de 650.62 a 774.19 ppm.

Aforo con 10 personas			**En rango de mediciones**	
Mediciones	C02 (ppm)	medidas	Distribución	Frecuencia
1	768	610	0.00163584	0
2	681	620	0.00210656	2
3	741	630	0.00264348	5
4	773	640	0.00323259	5
5	621	650	0.00385207	6
6	727	660	0.00447311	6
7	692	670	0.00506169	7
8	736	680	0.00558152	7
9	755	690	0.00599764	7
10	697	700	0.00628028	8
11	775	710	0.00640838	8
12	627	720	0.00637219	9
13	662	730	0.00617447	8
14	613	740	0.00583018	7
15	655	750	0.00536456	6
16	708	760	0.00481014	5
17	612	770	0.00420293	6
18	770	780	0.00357864	5
19	805	790	0.0029693	5
20	763	800	0.00240083	5
21	788	810	0.00189164	5
Media	712.809524			
Desv. Est	62.1897251			
Max	805			
Min	612			

Tabla 16 Resultados concentración de aire - 2 participantes 4.3.4.3.1

La ilustración40 visualiza claramente esta distribución de datos obtenidos.

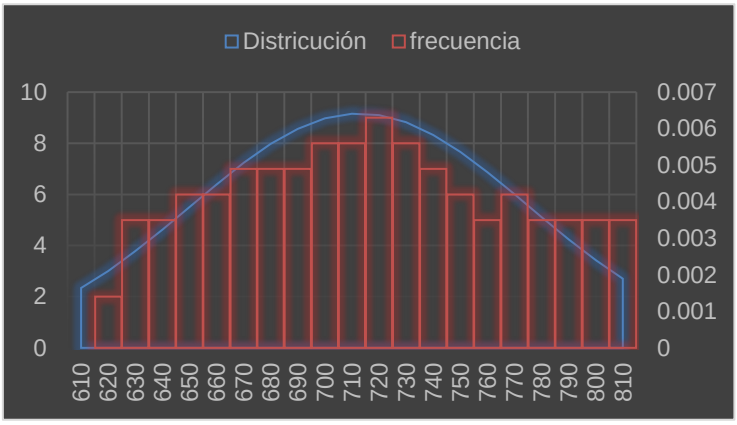

Ilustración 40 Distribución normal - 10 participantes 4.3.4.3.1

En resumen, los datos evaluados indican que la calidad de aire en un espacio cerrado se ve significativamente afectada por el número de participantes. Para un total de 5 participantes, la concentración de CO_2 se mantiene en niveles aceptables. Sin embargo, cuando se alcanzan los 10 participantes, la concentración de aire se acerca al límite permitido sin representar un riesgo alto; no obstante, en estas condiciones es crucial mantener una ventilación adecuada.

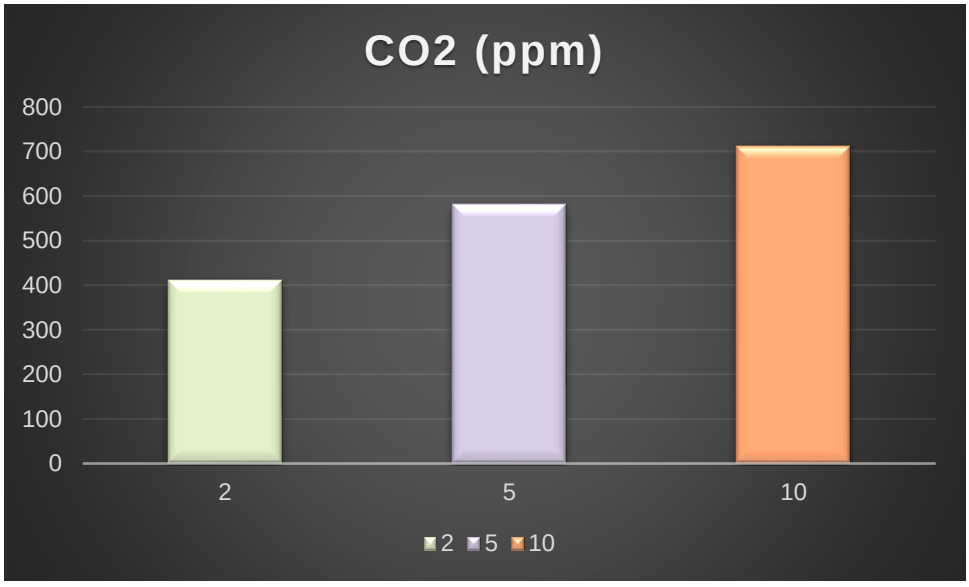

Ilustración 41 CO2-Aforo 4.3.4.3.2

La Ilustración 41 presenta un resumen de los valores típicos en cada uno de los casos. Los valores para 2 participantes se han resaltado en verde, indicando una buena calidad de aire. A medida que el aforo aumenta a 5 participantes, se resaltan en amarillo debido a una alerta importante, y cuando llega a 10 participantes, se

marcan en naranja. Aunque no se trata de una calidad de aire crítica, no se encuentra en niveles adecuados para ambientes cerrados.

Estos hallazgos proporcionan una comprensión significativa de la relación entre la concentración de aire y la ocupación en espacios cerrados, lo cual puede tener relevancia para la calidad del aire en entornos similares.

5. CONCLUSIONES

La observación de la relación entre la ubicación del circuito de detección y su capacidad para identificar objetos de diferentes alturas ha subrayado la importancia crítica de la precisión espacial en sistemas de este tipo.

Posicionar el sistema en el marco de la puerta a cota cero, puede tener un impacto sustancial en su rendimiento. Estos hallazgos son consistentes con la literatura existente sobre la importancia de la ubicación en la eficacia de sistemas de detección y seguimiento. Sin embargo, es importante reconocer que, a pesar de las mejoras logradas, el diseño y la implementación de sistemas de detección siguen siendo un campo en evolución, con consideraciones adicionales que deben abordarse en situaciones más complejas y dinámicas.

Desde una perspectiva investigativa, este trabajo destaca la necesidad de una comprensión más profunda de las interacciones entre la tecnología de detección y el entorno en el que se implementa. Los resultados sugieren oportunidades para futuras investigaciones en términos de optimización adicional de sistemas de detección y la adaptación de estos sistemas a diversos contextos. Además, la metodología empleada en este experimento podría servir como punto de partida para investigaciones similares en otras áreas de detección y seguimiento, impulsando así el avance científico y tecnológico en este campo.

En base a los resultados del caso de estudio 1, se puede concluir que el sistema cumple con éxito su objetivo principal, que es el de diseñar un sistema de control de aforo. El alto nivel de detección, con un 87.50% de aciertos, sugiere que el prototipo es eficaz en su tarea principal de contabilizar la entrada y salida de participantes en situaciones controladas. No obstante, es importante recordar que el 12.50% restante,

podrían deberse a varias razones, como condiciones ambientales no ideales, velocidad de procesamiento de datos o limitaciones del prototipo, velocidad en situaciones más complejas.

Con los resultados obtenidos en el caso de estudio 2, se puede concluir que la calida de aire en espacios cerrados está directamente relacionada con la cantidad de personas presentes. Cuando hay solo 2 participantes, la calidad de CO_2 en el ambiennte se mantiene en niveles considerados buenos. Sin embargo, a medida que aumenta la ocupación a 5 participantes, se observa una alerta importante, indicando una disminución en la calidad del aire. Cuando el número de participantes llega a 10, la calidad del aire se sitúa en niveles altos, lo que significa que no es crítica pero tampoco adecuada para ambientes cerrados.

Estos resultados subrayan la importancia de la ventilación adecuada en espacios cerrados, especialmente cuando hay una mayor afluencia de personas. Mantener una buena calidad del aire en tales entornos es esencial para garantizar un ambiente saludable y seguro.

6. APÉNDICES

6.1. Instalación Rabian

Raspbian está diseñado específicamente para la Raspberry Pi, lo que significa que está optimizado para aprovechar al máximo los recursos y capacidades del hardware de la Pi. Esto permite un rendimiento más eficiente y una mejor experiencia general del usuario. Además, su interfaz se basa en el escritorio LXDE, siendo muy intuitiva al momento de usar y configurar librerías y herramientas de visualización.

Para su correcta instalación se debe seguir una serie de pasos que se detallan a continuación:

6.1.1. Descargar Raspberry Pi OS:

Descargar del sitio web oficial de Raspberry Pi:

https://www.raspberrypi.org/downloads/raspberry-pi-os/

Descargar la última versión del sistema operativo para Raspberry Pi 3 (generalmente, la versión "Raspberry Pi OS with Desktop" es la más adecuada para la mayoría de los casos).

6.1.2. Formatear la tarjeta microSD:

Utilizar una aplicación como "SD Card Formatter" para formatear completamente la tarjeta microSD. Se puede descargar desde:

https://www.sdcard.org/downloads/formatter/

6.1.3. Grabar la imagen en la tarjeta microSD:

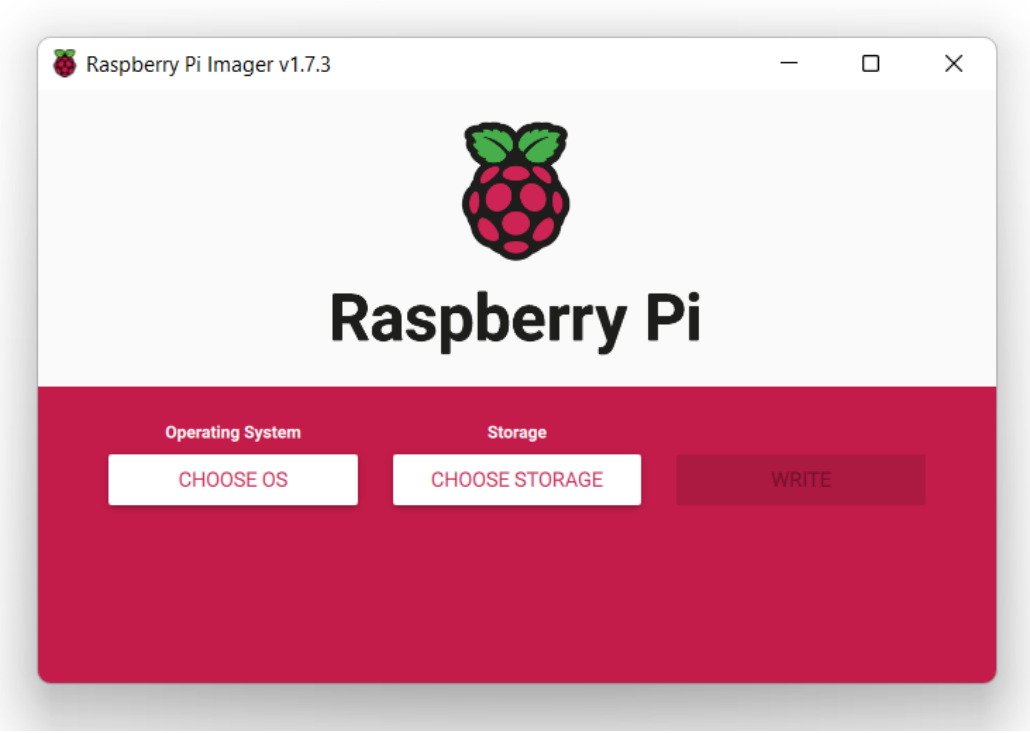

Ilustración 42 Imagen Rabian 6.1.3.1

6.1.4. Configurar el acceso de red Raspberry Pi:

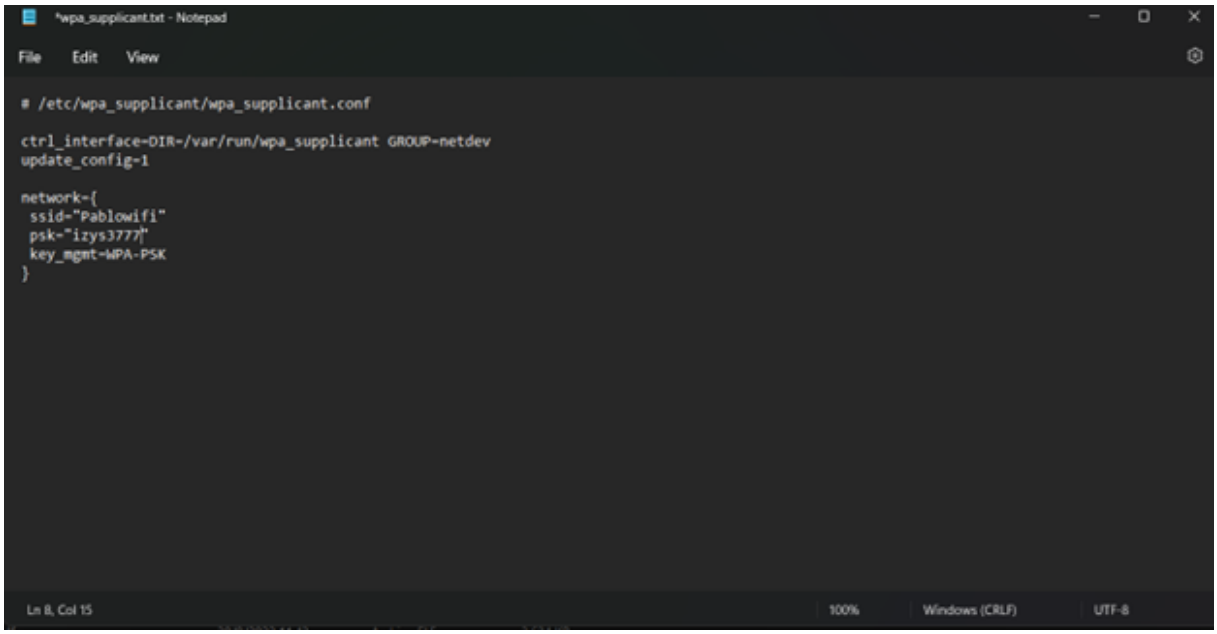

Ilustración 43 Configuración de red RasberryPi 6.1.4.1

Buscar al dispositivo por medio de Scanner

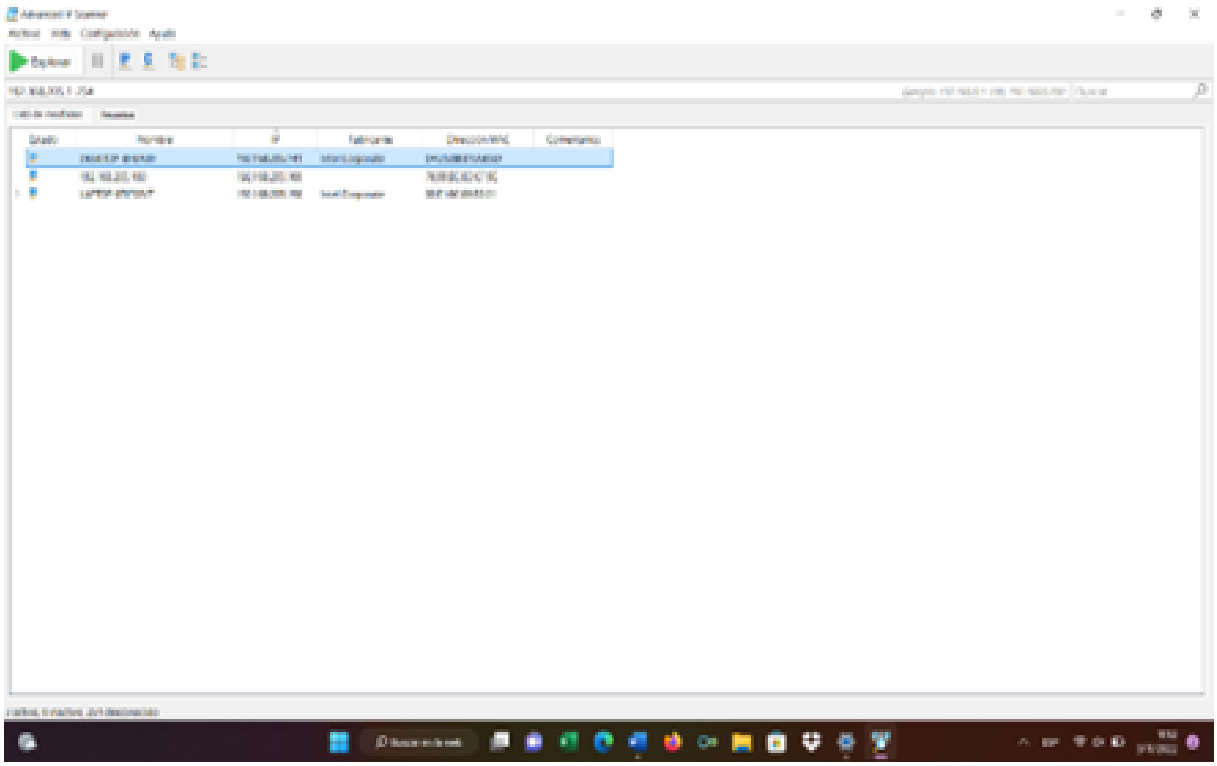

Ilustración 44 PuTTY scanner 6.1.4.2

Abrir un terminal:

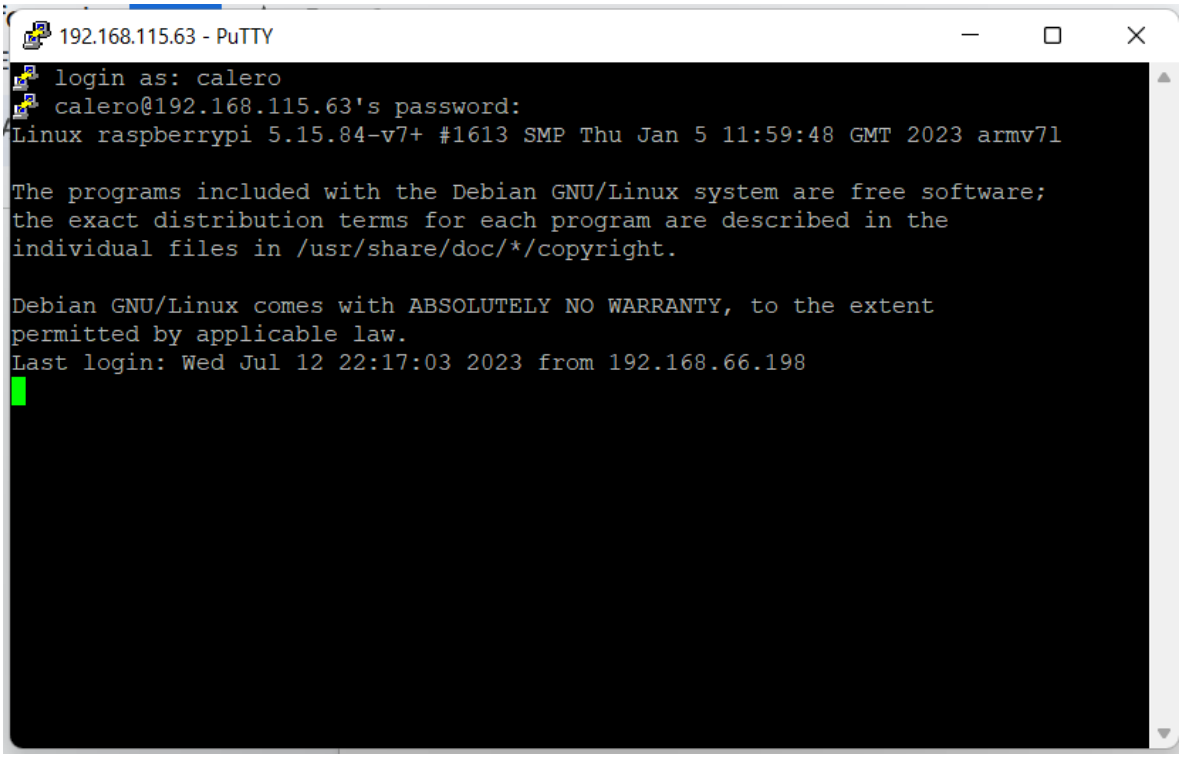

Ilustración 45 terminal iniciado 6.1.4.3

Para poder acceder al escritorio de la raspberry de una forma visual, se debe ingresar al servicio de VNC viewer, ejecutando el siguiente comando en la terminal.

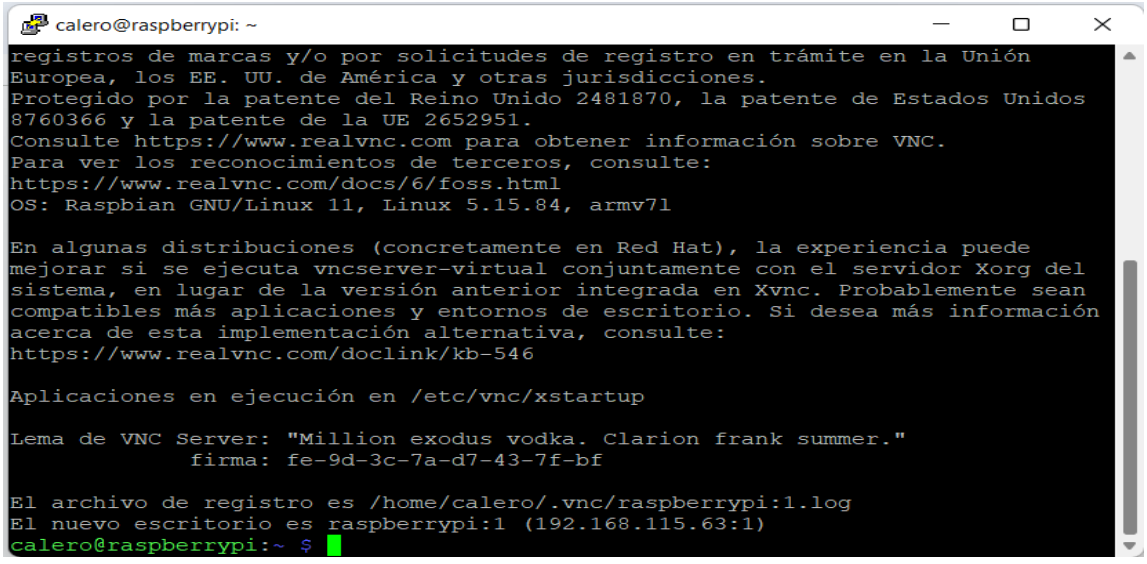

Ilustración 46 iniciar vncserver 6.1.4.4

Abrir el escritorio por medio de la aplicación VNC Viewer

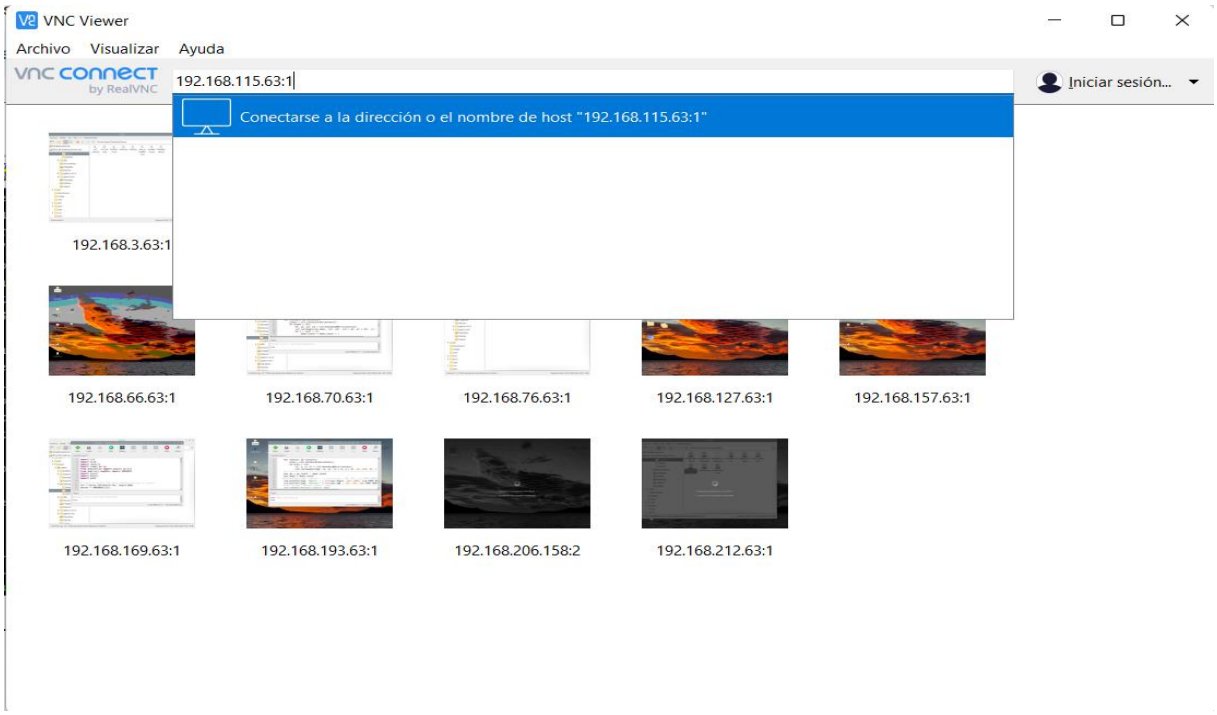

Ilustración 47 VNC viewer 6.1.4.5

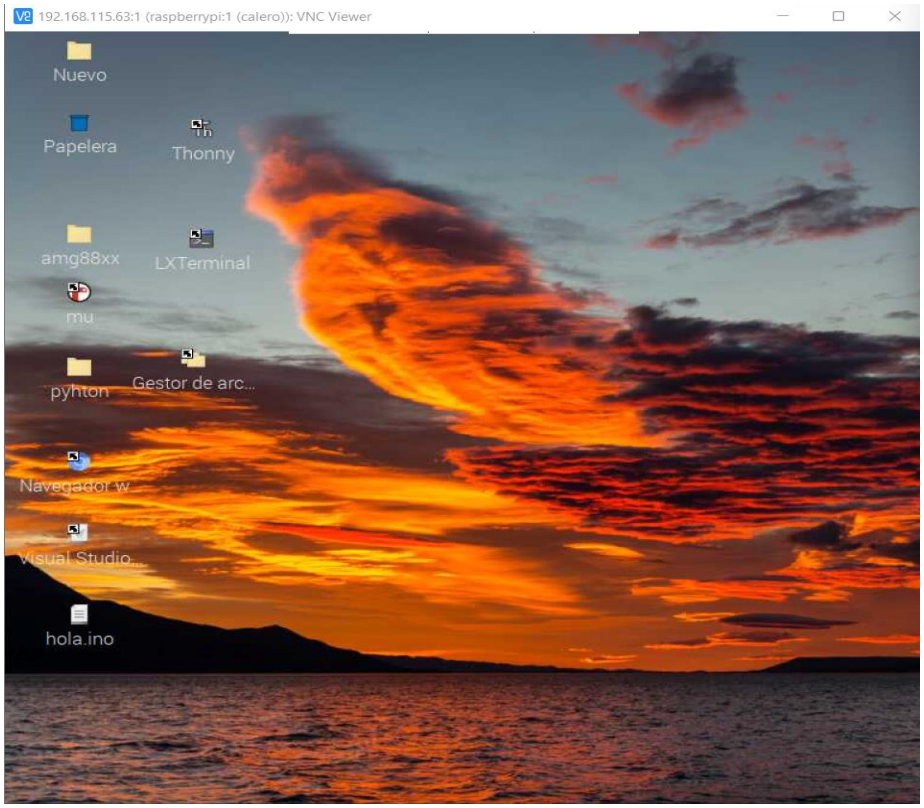

Ilustración 48 Escritorio Raspberry 6.1.4.6

Abrir una terminal y ejecutar los siguientes comandos para asegurar de que el sistema esté actualizado

- sudo apt update
- sudo apt upgrade
- i2c detect

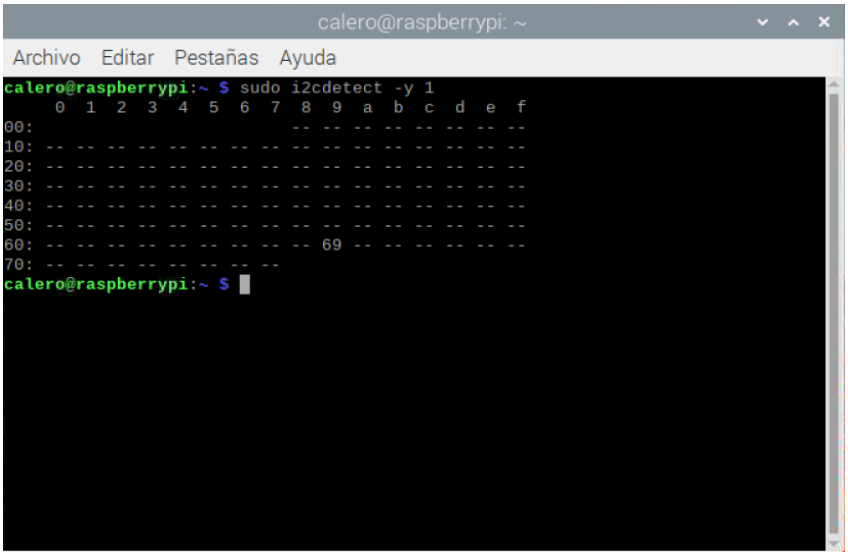

Ilustración 49 lectura I2c 11.1.4.7

6.2. Instalación base de datos NoSQL MongosDB

6.2.1. instalación de la librería en la raspberry

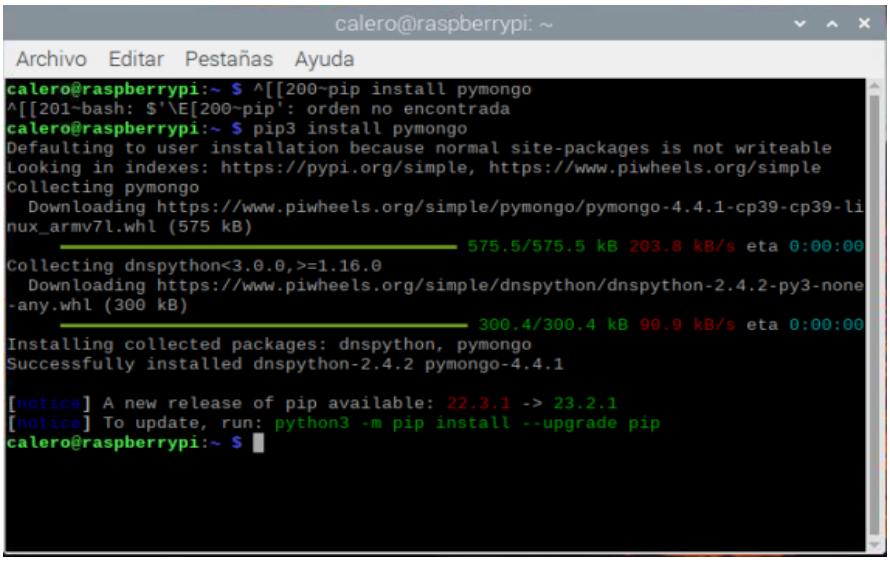

Ilustración 50 Instalación Mongos en Raspberry 6.2.1.1

Descargar Mongo en nuestro Ordenador

Ejecutar por medio de la consola los siguientes programas utilizando los siguientes comandos:

- mongo
- mongosh

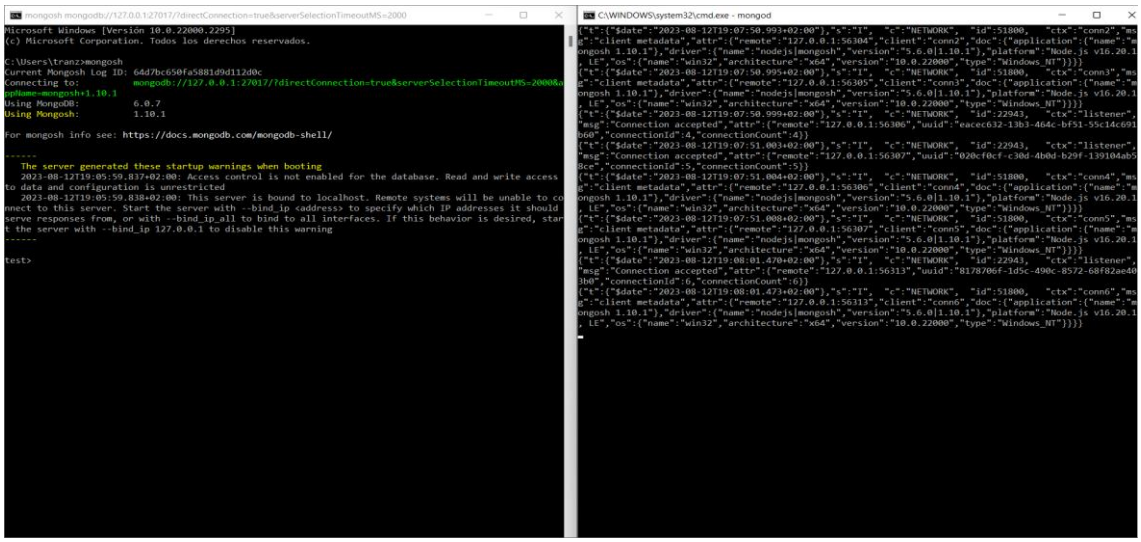

Ilustración 51 inicialización app de escritorio 6.2.2.1

Se puede observar que la base de datos se ha iniciado correctamente y ahora ya se puede acceder a la base de datos.

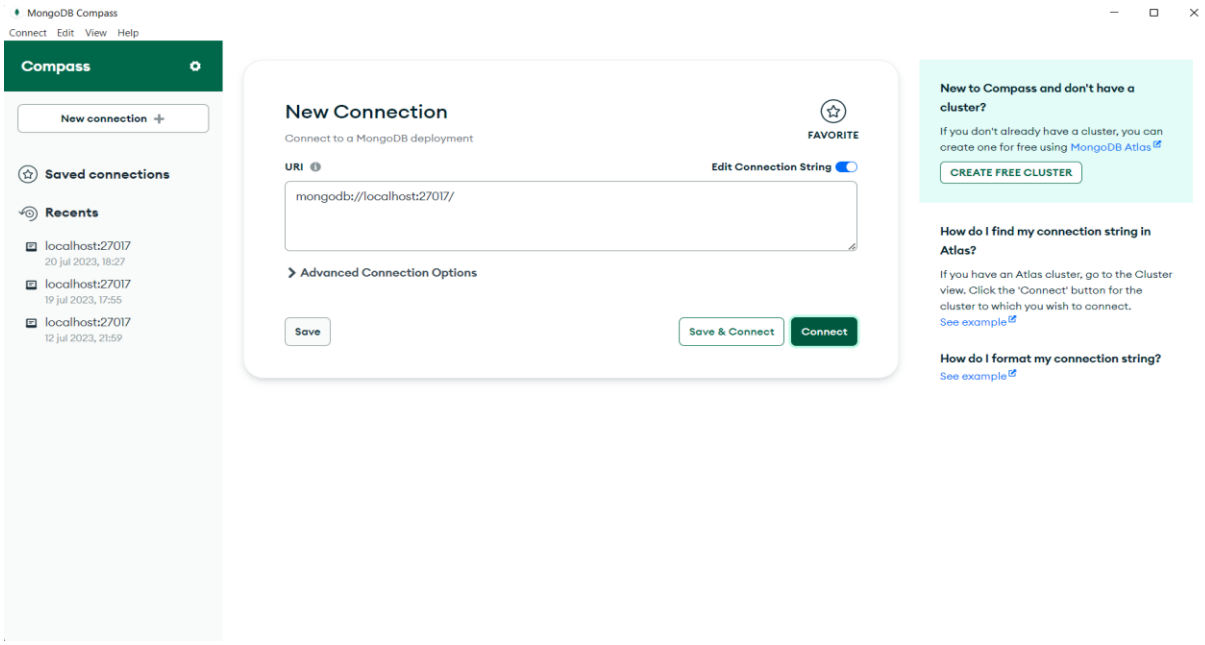

Ilustración 52 Aplicación iniciada 6.2.2.2

Seleccionar connect. Una vez seleccionado, se podrá acceder.

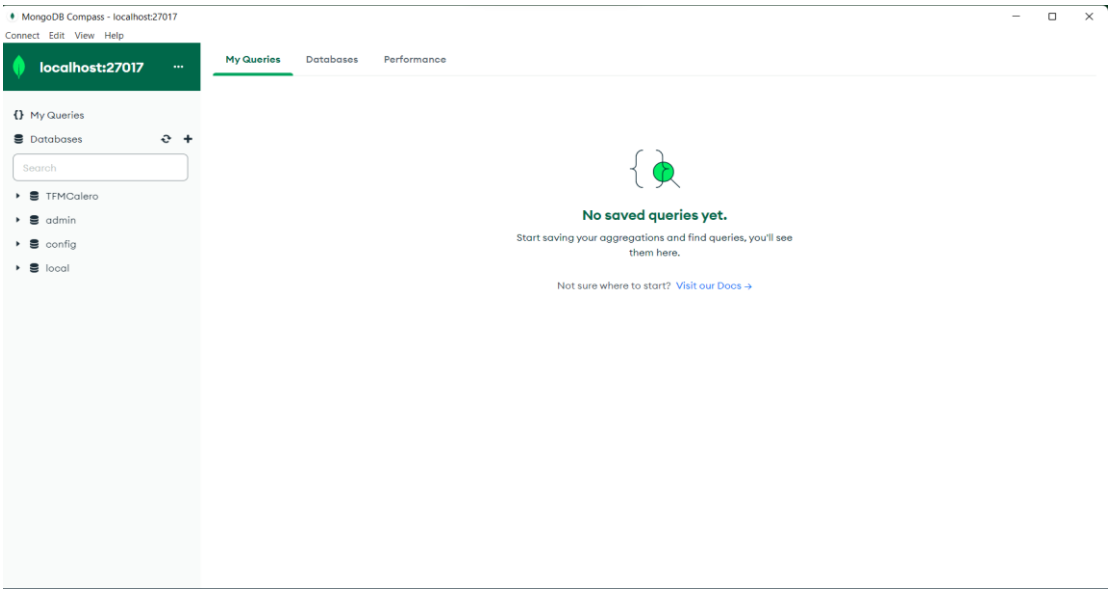

Ilustración 53 Ilustración 51 Conexión Host 11.2.2.3

Se crea la base de datos de interés.

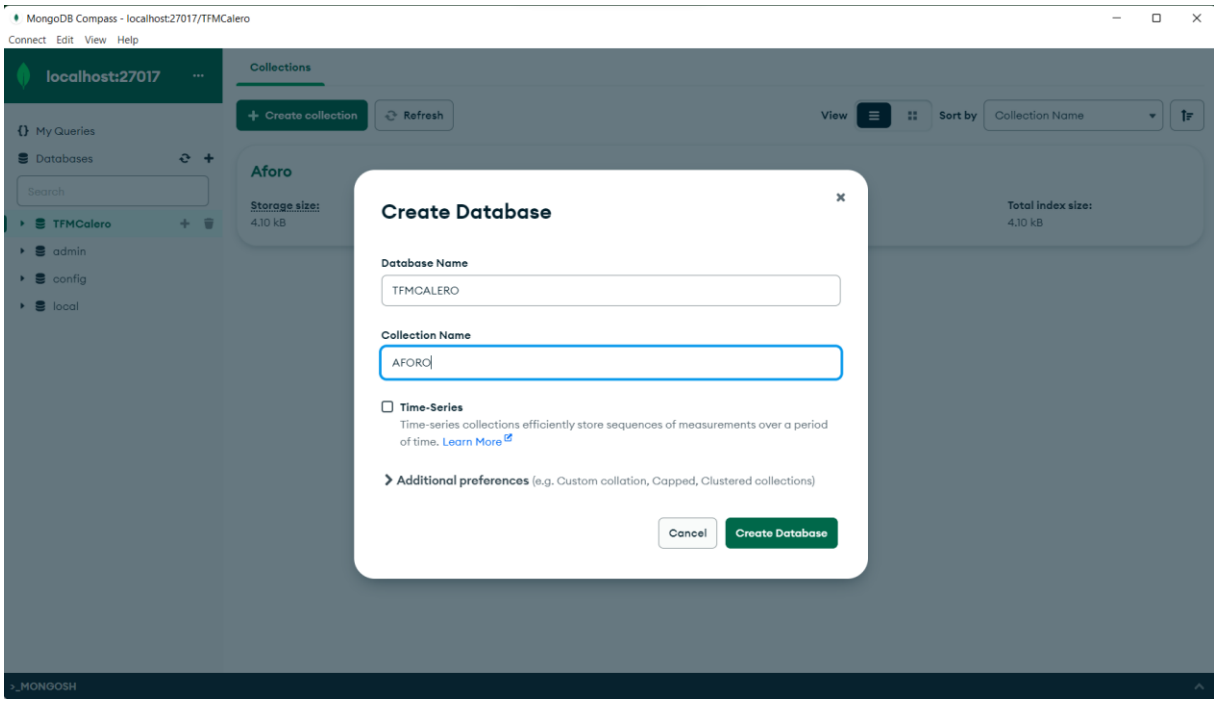

Ilustración 54 Creación base de datos 6.2.2.4

6.3. Proyecto

6.3.1. Proyecto creado en Thony – Rarspberry Pi

```
import cv2
import dlib
import imutils
import numpy as np
from matplotlib import pyplot as plt
from adafruit_amg88xx import AMG88XX
import busio
import board
import math
import paho.mqtt.client as mqtt
import pymongo
import time

# Definir las constantes de conexión MQTT

MQTT_BROKER = "localhost"  # Ingresa la dirección IP o URL de tu broker MQTT
MQTT_PUERTO = 1883  # Puerto predeterminado para MQTT
MQTT_TOPIC = "TFMCALERO"  # Ingresa el nombre del tópico MQTT
# Callback que se ejecuta cuando se establece la conexión con el broker MQTT
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Conectado al broker MQTT")
        client.subscribe("TFMCALERO")  # Suscribirse a un tema para recibir mensajes
    else:
        print("Error de conexión. Código de resultado: " + str(rc))
# Callback que se ejecuta cuando se recibe un mensaje MQTT
def on_message(client, userdata, msg):
    mensaje = msg.payload.decode("utf-8")
    print("Mensaje recibido: ", mensaje)

# Configurar el cliente MQTT

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
# Conectar al broker MQTT
client.connect(MQTT_BROKER, MQTT_PUERTO, 60)
def send_entrada():
    client.publish("TFMCALERO", "Entrar" + str(aux_up))
    print("mensaje enviado entrada")
def send_salida():
    client.publish("TFMCALERO", "Salir" + str(aux_down))
    print("Mensaje enviado salida")
# Crear un objeto I2C para comunicarse con el sensor
i2c = busio.I2C(board.SCL, board.SDA)
sensor = AMG88XX(i2c)

# Configurar una ventana de visualización con OpenCV

cv2.namedWindow("Thermal Camera", cv2.WINDOW_NORMAL)
```

```
aux_down = 5
aux_up = 0
# Configura la conexión a la base de datos
client = pymongo.MongoClient("mongodb://127.0.0.1:27017/")  # Cambia la URL según sea necesario
db = client["TFMCalero"]
# Selecciona la colección en la que deseas insertar los datos
collection = db["Aforo"]
while True:
    # Obtener lecturas de temperatura del sensor
    pixels = sensor.pixels
    # Convertir la matriz de temperatura a una imagen de escala de grises
    image = np.uint8(pixels)
    # Escalar la imagen a un tamaño más pequeño para una mejor visualización
    image = cv2.resize(image, (320, 240))
     # Mostrar la imagen en la ventana de visualización
    lower = np.array([32])
    upper = np.array([66])
    mask = cv2.inRange(image, lower, upper)
    img = cv2.cvtColor(mask, cv2.COLOR_GRAY2BGR)
      # Calcular la mitad de la altura y la anchura de la imagen
    height, width = img.shape[:2]
    center_x = width // 2
    center_y = height // 2
    # Dibujar una línea roja desde la esquina superior izquierda hasta la esquina inferior derecha
    cv2.line(img, (0, center_y), (width, center_y), (0, 0, 255), 1)
    # zona de interes
    up = image[0:120, 0:width]
    down = image[120:240, 0:320]
    mask_up = cv2.inRange(up, lower, upper)
    mask_down = cv2.inRange(down, lower, upper)
    im_up = cv2.cvtColor(mask_up, cv2.COLOR_GRAY2BGR)
    im_down = cv2.cvtColor(mask_down, cv2.COLOR_GRAY2BGR)

    # Encontrar contornos en la máscara

    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    contours1, _ = cv2.findContours(mask_up, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    contours2, _ = cv2.findContours(mask_down, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # contadores

    up_count = 0
    down_count = 0
    positions_up = set()
    positions_down = set()
   # Dibujar un rectángulo
    for contour1 in contours1:
        area1 = cv2.contourArea(contour1)
        if area1 > 2000:
            x1, y1, w1, h1 = cv2.boundingRect(contour1)
```

```
            cv2.rectangle(im_up, (x1, y1), (x1 + w1, y1 + h1), (0, 255, 0), 1)
            if 109 < (y1 + h1) < 113:
                if (x1,y1) not in positions_up:
                    positions_up.add((x1,y1))
                    up_count += 1
                    aux_up += 1+up_count-down_count
                    send_entrada()
                    result = collection.insert_one(aforo_entrar)

                cv2.line(img, (0, 120), (width, 120), (0, 255, 255), 2)
    for contour2 in contours2:
        area2 = cv2.contourArea(contour2)
        if area2 > 2000:
            x2, y2, w2, h2 = cv2.boundingRect(contour2)
            cv2.rectangle(im_down, (x2, y2), (x2 + w2, y2 + h2), (0, 255, 0), 1)
            if 0 < (y2) < 2:
                if (x2,y2) not in positions_down:
                    positions_down.add((x2,y2))
                    down_count += 1
                    aux_down += 1
                    send_salida()
                    result = collection.insert_one(aforo_entrar)
                cv2.line(img, (0, 120), (width, 120), (0, 255, 255), 2)
    for contour in contours:
        area = cv2.contourArea(contour)
        if area > 200:
            (x, y, w, h) = cv2.boundingRect(contour)
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 1)

    #****************************************************************

    #aux_up = (up_count+1)-aux_down
   #cv2.putText(img, "entrar: " + str(id_up), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
(0, 0, 255), 1)
    cv2.putText(img, "Salir: " + str(aux_down), (10, 200), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (205, 120, 255), 1)
    cv2.putText(img, "Entrar:" + str(aux_up) , (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
(205, 120, 255), 1)
    #cv2.imshow("Thermal Camera",img)
    cv2.imshow("Thermal Camera",img)
    #cv2.imshow("Thermal Camera",down)

    client.loop()

    #time.sleep(5)  # Esperar 5 segundos
    # Salir del bucle si se presiona la tecla 'q'
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Datos que deseas insertar colleccion de datos
aforo_entrar = {

"name": "Entrar",
    "Personas": aux_up,
    "Date": ISODate()
```

```
    "sensor de aire": mensaje,
}

# Datos que deseas insertar colleccion de datos
aforo_salir = {

    "name": "Salir",
    "Personas": aux_down,
    "Date": ISODate()
    "sensor de aire": mensaje,
}
# Liberar la ventana de visualización
cv2.destroyAllWindows()
```

6.3.2. Proyecto creado IDE Arduino -ESP8266

```
#include <MQ135.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

/************************ WiFi Configuration ******************************/
const char* ssid = "Pablowifi";
const char* password = "izys3777";

/************************ MQTT Configuration ******************************/
const char* mqttServer = "192.168.82.63";
const int mqttPort = 1883;
const char* mqttUser = "";
const char* mqttPassword = "calero";
const char* mqttTopic = "TFMCALERO";

/************************ Sensor Configuration ****************************/
#define PIN_MQ135 A0

MQ135 mq135_sensor(PIN_MQ135);

float temperature = 28.0; // Assume current temperature. Recommended to measure with
DHT22
float humidity = 25.0; // Assume current humidity. Recommended to measure with DHT22

/************************ MQTT Client *************************************/
// Cliente WiFi
WiFiClient wifiClient;
// Cliente MQTT
PubSubClient mqttClient(wifiClient);

void setup() {
  // open serial for monitoring
  Serial.begin(9600);

  // more details at http://docs.thinger.io/arduino/
// Inicializar la comunicación serial
  Serial.begin(9600);
```

```
  // Conectar a WiFi
 WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando a WiFi...");
  }
  //Serial.println("Conexión WiFi establecida.");

  // Conectar al broker MQTT
   mqttClient.setServer(mqttServer, mqttPort);
  while (!mqttClient.connected()) {
    if (mqttClient.connect("ArduinoClient", mqttUser, mqttPassword)) {
      Serial.println("Conexión MQTT establecida.");
    } else {
      Serial.print("Error de conexión MQTT, estado: ");
      Serial.println(mqttClient.state());
      delay(2000);
    }
  }

}

void loop() {

float rzero = mq135_sensor.getRZero();
  float correctedRZero = mq135_sensor.getCorrectedRZero(temperature, humidity);
  float resistance = mq135_sensor.getResistance();
  float co2ppm = mq135_sensor.getPPM()+400;
  float correctedCo2PPM = mq135_sensor.getCorrectedPPM(temperature, humidity)+400;

  Serial.print("MQ135 RZero: ");
  Serial.print(rzero);
  Serial.print("\t Corrected RZero: ");
  Serial.print(correctedRZero);
  Serial.print("\t Resistance: ");
  Serial.print(resistance);
  Serial.print("\t co2 PPM: ");
  Serial.print(co2ppm);
  Serial.print("\t Corrected PPM: ");
  Serial.print(correctedCo2PPM);
  Serial.println("ppm");

   // Publicar el mensaje en el topic MQTT

  Serial.print("Mensaje publicadoc02: ");
  Serial.println(correctedCo2PPM);

  String message = String(correctedCo2PPM);

  // Publicar el mensaje en el topic MQTT
  mqttClient.publish(mqttTopic, message.c_str());

  delay(10000);  // Wait for 10 seconds before taking another reading
}
```

Bibliografía

Abellán , Á. (09 de julio de 2019). Obtenido de Introducción a la placa de desarrollo NodeMCU ESP8266: https://www.programoergosum.es/tutoriales/introduccion-a-nodemcu-esp8266/

Castro, R. (febrero de 2020). Coronavirus, una historia en desarrollo. Santiago de Chile, Chile: Revista Mérdica de Chile.

Chandra, R., Verma, M., & Kumar, S. (2017). *INTERNET OF THINGS (IOT) BASED GAS LEAKAGE MONITORING AND ALERTING SYSTEM WITH MQ-2 SENSOR.*

Comité Español de Automáica. (2016). *Conceptos y Métodos en Visión por Computador*. Espaá. Obtenido de http://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf

Domingo , M. (17 de agosto de 2004). *Visión por Computador*. Santiago de Chile, Chile. Obtenido de http://dmery.sitios.ing.uc.cl/Prints/Books/2004-ApuntesVision.pdf

Electronica y Ciencia. (20 de noviembre de 2016). *Conexión GPIO de Raspberry Pi 3*. Obtenido de https://www.electronicayciencia.com/2016/11/conexion-gpio-de-raspberry-pi-3.html

Geraldi, J., & Lechler, T. (septiembre de 2012). Gantt Chart and the Scientific Management in. International Journal of Managing Projects in Business. Obtenido de https://www.researchgate.net/publication/263571905_Gantt_charts_revisited_A_critical_analysis_of_its_roots_and_implications_to_the_management_of_projects_today/link/5ff77a2145851553a02aeb4d/download

HiveMQ. (12 de enero de 2015). *Introducing the MQTT Protocol - MQTT Essentials*. Obtenido de https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/

Howese, J. (2013). *OpenCV computer Vision whith Python.* Birmingham: Packt Publishing ltd.

Jara, D. (2021). *"Computer Vision", Que es? que valor Genera? en que industria se aplica?*

Ministerio de Sanidad. (2020). *Información por coronavirus, COVID-19.* Obtenido de Información cientifico tecnico-tecnica: https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/ITCoronavirus.pdf

Montoro Ledinez, A. (2022). *Diseño y Prototipo de un concentrador de Gas.* Jaén: Universidad de Jaén.

Muñoz Manzo, R. (Julio de 2014). *Sistema de vision artificial para la detección y lectura de matriculas*. Valladolid, Espeña: Universidad de Valladolid.

Perez, I., Díaz, R., & Becerra Garcia , R. (2014). *EL lenguaje de programación Python.* Ciencias Holguin.

Phthon Software Foundation . (2023 de agosto de 8). Obtenido de Phthon Software Foundation : https://www.python.org/psf-landing/

Pore, S. D., & Momin, B. F. (2017). *Bidirectional people counting system in video surveillance.* Bangalere.

Raoul, S. ,., & Estela Saquete, B. (Julio de 2021). Aplicación multiplatafora para el control de aforo. Alicante, España: Universidad de Alicante. Obtenido de https://rua.ua.es/dspace/bitstream/10045/118070/1/Aplicacion_multiplataforma_para_control_de_aforo_Rhazili_Raoui_Soufyan.pdf

Raspberry Pi. (2023). Obtenido de https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-3-model-b

Supria, &. M. (05 de 11 de 2020). MONITORING OF BODY TEMPERATURE NON CONTACT USING AMG8833 THERMAL CAMERA AND FACE DETECTION. *Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV)*. Obtenido de https://proceeding.isas.or.id/index.php/sentrinov/article/view/379

Torres , A. (s.f.). Adadrui AMG8833 thermal camera sensor. Obtenido de https://learn-adafruit-com.translate.goog/adafruit-amg8833-8x8-thermal-camera-sensor/overview?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es-419&_x_tr_pto=sc

Universidad Complutense Madrid. (2020). *RECOMENDACIONES PARA LA UTILIZACIÓN DE MEDIDORES CO2 PARA EVALUAR LA EFICIENCIA EN EL ESCENARIO COVID DE LA PAUTA DE VENTILACIÓN DE LOS ESPACIOS DE LA UCM CON OCUP*.

Worlddimeter. (2020). *Coronavirus Cases*. Obtenido de https://www.worldometers.info/coronavirus/