

UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



UCSC

Informe Habilitación Profesional para optar al Título de Ingeniero Civil Eléctrico.

Diseño de un sistema de trazabilidad de contagios de COVID-19 usando internet de las cosas

Pablo Andrés Díaz Paredes

Profesor Guía:

Dra. Silvia Restrepo Medina

Comisión

Dr. Ricardo Bustos Placencia – Dr. Jaime Cariñe Catrileo

Concepción, octubre de 2022

Agradecimientos

En primer lugar, me gustaría expresar mis agradecimientos a mi profesora guía y tutora, la Dra Silvia Restrepo Medina, por su apoyo, tiempo y dedicación durante el desarrollo de este trabajo, por ayudarme en la búsqueda de soluciones ante las diversas eventualidades que se presentaron durante el desarrollo del proyecto, por atender mis ideas y aportar sugerencias, compartir su conocimiento y experiencia para la realización de las mismas.

En segundo lugar, quiero dar mis agradecimientos a la Dirección de Investigación de la Universidad Católica de la Santísima Concepción quien me brindó un apoyo económico, gracias a la adjudicación del Fondo de Apoyo a la Ejecución de Proyectos de Investigación para Estudiantes de Pregrado Año 2021, el cual fue de mucha ayuda para el desarrollo del proyecto, muchas gracias.

En tercer lugar, quiero agradecer a mi familia, a mi pareja afectiva y su grupo familiar y especialmente a mi madre, por acompañarme en esta etapa y apoyarme en todo lo que estuvo a su alcance, especialmente en lo afectivo.

En cuarto lugar, quiero dar gracias a mis amigos, compañeros de carrera y en general al ambiente universitario por sus compañías y apoyos que me brindaron en todo momento, Cómo no reconocer ese estímulo que me entregaron cuando pasaba un mal momento y las fuerzas se quebraban, siempre hubo una palabra de aliento en los momentos difíciles.

La suma de los aportes de todas estas personas, a quienes estaré por siempre agradecidos, les dedico mi trabajo y mis logros en esta etapa de mi vida.

Resumen

La trazabilidad consiste de una serie de procedimientos que permiten registrar e identificar la ubicación, trayectoria y comportamiento de objetos o personas, permitiendo un seguimiento de éstos. Los sistemas de trazabilidad generaron un impacto positivo en el control de la pandemia del COVID-19, aportando con su seguimiento de casos a la efectiva vigilancia del mal. El objetivo de este trabajo es diseñar un sistema de trazabilidad capaz de identificar posibles casos de COVID-19 por medio de la toma de la temperatura corporal; ésta, al ser uno de los síntomas con mayor incidencia en casos positivos de COVID 19, sumado a que es una variable de estudio rápida y no invasiva para las personas, llevó a considerarla como factor interesante para este trabajo.

El sistema propuesto está basado en el Internet de las Cosas (IoT) y se divide en tres partes, un subsistema de temperatura, otro de aforo y uno de identificación, para reconocer posibles casos de esta enfermedad. Para el subsistema de temperatura se utilizan sensores térmicos, como el MLX90614 y el arreglo de sensores AMG 8833. Del primer sensor se obtiene una temperatura de referencia de la persona. Mientras que del segundo sensor se obtiene una matriz de 64 mediciones de temperatura, cuyos valores son promediados considerando únicamente aquellos que superen la temperatura de referencia obtenida con el otro sensor y posteriormente se realiza un seguimiento por medio de plataformas digitales compatibles con arduino. Para el subsistema de aforo se emplea un sensor de ultrasonido para identificar entradas y salidas de un recinto. Por último, para el subsistema de identificación se utiliza la tecnología RFID y una base de datos alojada en la plataforma firebase para hacer un reconocimiento de usuarios.

De los resultados obtenidos de esta investigación se comprueba que el sistema diseñado distingue diferentes casos dependiendo de la temperatura obtenida por medio del subsistema de temperatura, pudiendo identificar si la persona tiene una temperatura normal (entre 35,5°C y 37,5°C), si tiene fiebre (entre 37,51°C y 42,9°C), si tiene baja temperatura (entre 32,0°C y 35,49°C) o si se encuentra fuera de los rangos de operación (menor a 32,0°C o superior a 42,9°C). Por medio del subsistema de identificación es posible reconocer a los usuarios por sus tarjetas RFID, su nombre y si éstos se encuentran autorizados o no para el ingreso a una determinada dependencia, lugar al que desea ingresar, etc. Por otra parte, el sistema de aforo es capaz de diferenciar si una persona entra o sale del lugar y avisar la capacidad total y restante del lugar.

Tabla de Contenidos

CAPÍTULO 1: INTRODUCCIÓN	6
1.1 INTRODUCCIÓN GENERAL	6
1.2 MOTIVACIÓN	8
1.3 TRABAJOS PREVIOS	9
1.4 ANÁLISIS TRABAJOS PREVIOS	11
1.5 HIPÓTESIS DE TRABAJO	11
1.6 OBJETIVOS	11
1.6.1 <i>Objetivo General</i>	11
1.6.2 <i>Objetivos Específicos</i>	12
1.7 ALCANCES Y LIMITACIONES	12
1.8 METODOLOGÍA	12
CAPÍTULO 2: TEORÍA GENERAL	14
2.1 INTERNET DE LAS COSAS (IoT)	14
2.2 SENSORES	15
2.3 ACTUADORES Y PERIFÉRICOS	16
2.3.1 <i>Actuadores</i>	16
2.3.2 <i>Periféricos</i>	17
2.4 PLATAFORMAS	18
2.4.1 <i>Plataformas Digitales</i>	18
2.4.2 <i>Plataformas de control</i>	19
• Arduino Uno	19
• Raspberry	20
2.5 PROTOCOLOS DE COMUNICACIÓN	20
• Wifi	20
• Bluetooth	21
• Zigbee	21
2.6 EL INTERNET DE LAS COSAS MÉDICAS	22
2.7 SISTEMA DE TRAZABILIDAD	24
2.7.1 <i>Trazabilidad en el contexto del COVID-19</i>	24
2.7.2 <i>Riesgos a la privacidad</i>	26
2.8 SITUACIÓN MUNDIAL DE LA PANDEMIA	27
2.8.1 <i>Situación mundial entre los años 2020-2021</i>	29
2.8.2 <i>Situación de la pandemia hasta julio del 2022</i>	30
CAPÍTULO 3: DISEÑO	33
3.1 PLATAFORMA PROGRAMABLE	34
3.2 SENSORES	35
3.2.1 <i>Sensor de ultrasonido</i>	35
3.2.2 <i>Sensor de temperatura infrarrojo MLX90614</i>	37
3.2.3 <i>Cámara térmica infrarroja AMG8833</i>	37
3.2.4 <i>Módulo RFID</i>	39
3.3 PLATAFORMAS DIGITALES	40
3.4 PERIFÉRICO: DISPLAY OLED	44
3.5 CALIBRACIÓN Y CARACTERIZACIÓN DE SENSORES TÉRMICOS	45
3.6 SISTEMA DE TRAZABILIDAD	50
3.6.1 <i>Subsistema de identificación</i>	51
3.6.2 <i>Subsistema de aforo</i>	54
3.6.3 <i>Subsistema de temperatura</i>	57
CAPÍTULO 4: RESULTADOS	62
4.1 RESULTADOS MONITOR SERIE IDE ARDUINO	62
4.2 RESULTADOS DISPOSITIVO MÓVIL CON BLYNK	66
4.3 RESULTADOS DEL HISTORIAL VIRTUAL ADAFRUIT.IO	68
4.4 RESULTADOS FIREBASE	70
4.5 RESULTADOS DISPLAY OLED	72

CAPITULO 5: CONCLUSIONES.	74
5.1 CONCLUSIONES.....	74
5.2 TRABAJO FUTURO.....	77
REFERENCIAS BIBLIOGRÁFICAS	78
ANEXO A. VERIFICACION DE CALIBRACION.	82
ANEXO B. CÓDIGOS	85

Capítulo 1: Introducción

1.1 Introducción General

Las enfermedades contagiosas han tenido un especial impacto y protagonismo en el curso de la historia. Es así que cuando una enfermedad se extendía y afectaba a varias regiones del planeta pasaba a ser una pandemia [1], provocando grandes cambios en la sociedad e influyendo fuertemente en el curso de la era humana. La primera pandemia de la que se tiene conocimiento es la peste de Justiniano que afectó al Imperio Bizantino (tierras que hoy corresponden a Italia, Suiza, Austria, Eslovenia, Croacia, Hungría) entre los años 541 y 549, matando cerca de unas 40 millones de personas [2]. Otra que afectó a la humanidad de forma agresiva fue la peste negra, que asoló al continente europeo en el siglo XIV entre los años 1346 y 1353, dejando más de 50 millones de personas fallecidas [3]. Más reciente en la era humana, siglo pasado, se registraron la gripe española (1918), la gripe asiática (1957) y la gripe de Hong Kong (1968) como pandemias reconocidas a nivel mundial [1]. Ahora, lamentablemente, nos golpea la pandemia del COVID-19 que es una enfermedad infecciosa causada por el virus SARS-CoV-2 [4]. Comenzó en diciembre de 2019 en Wuhan (China). En enero de 2020 la Organización Mundial de la Salud (OMS) evaluó el riesgo de epidemia, recomendando a los países tomar medidas preventivas para la gripe, pero sin limitar los viajes internacionales. Ante el aumento de casos en el mundo, el 11 de marzo de 2020 se decreta estado de pandemia [5].

Uno de los puntos más relevantes que diferencia a las antiguas pandemias con la del COVID-19 es la importancia de la tecnología en nuestra vida cotidiana. La modernización de los servicios se han vuelto claves para el control y gestión de la pandemia, incentivando el uso de tecnologías para evitar la propagación de este virus, tales como la digitalización, la creación de aplicaciones y el uso de la inteligencia artificial [6]. Una de las principales herramientas para combatir esta pandemia es el uso del Internet de las Cosas (IoT). Al contar con un monitoreo remoto se pueden evitar o minimizar las interacciones públicas en el área de la salud, impulsando que en esta área avance con mayor rapidez la integración de las tecnologías IoT [7].

Este documento de tesis está organizado de la siguiente forma. En el primer capítulo, se describe la motivación detrás del proyecto y se abarca la importancia y relevancia que posee el internet de las cosas en nuestra actual sociedad. Se plantea la problemática, los objetivos generales y el objetivo específico del presente estudio, junto a los alcances y limitaciones del mismo. El segundo capítulo muestra el marco teórico en ámbito general, qué es el internet de las cosas, las diversas áreas en que tiene aplicación, haciendo especial mención a las aplicaciones médicas y la comparación de la condición de contagios y muertes a nivel mundial y continental relacionados a la pandemia. En el tercer capítulo se presenta el diseño del proyecto, donde se describen y explican los sensores a utilizar y cómo se usarán para cada uno de los tres subsistemas que componen el proyecto final, definidos como subsistema de temperatura, subsistema de aforo y subsistema de identificación. Además, se menciona la plataforma programable ESP32 que usa la IDE de arduino para el desarrollo del código del proyecto, los actuadores seleccionados, donde en su mayoría son plataformas digitales, el cómo están constituidos los subsistemas y como operan. En el cuarto capítulo, se presentan los resultados preliminares y los resultados finales obtenidos de los diferentes subsistemas del proyecto abordado en esta tesis, usando las diferentes interfaces digitales y el monitor serie propio de la IDE de arduino. En el quinto capítulo y último del trabajo, se presentan las conclusiones, los trabajos a futuro, las recomendaciones personales sobre el sistema y cómo éste puede ser mejorado considerando diversos casos aplicables según lo realizado en el proyecto.

1.2 Motivación

Nuestro país no ha sido la excepción en cuanto a los efectos de la pandemia, generando estados de alerta y acciones tendientes a disminuir los casos positivos. Durante la contingencia sanitaria, los países del orbe, proporcionalmente a la disposición de recursos económicos, médicos y tecnológicos, se han visto enfrentados a desafíos que persiguen minimizar el daño que provoca en la sociedad una pandemia. Para nuestro país, ha sido complejo implementar medidas, en parte por las consecuencias colaterales que ha significado la pandemia, no obstante, se han efectuado estrategias que tienden a obtener un eficiente seguimiento de casos confirmados y para ello el manejo de recursos e información [8]. Las autoridades buscan que las personas interactúen a distancia evitando el contacto que propague el virus y si ello es inevitable, se piensa en acciones de control que permitan chequear el estado de la persona en relación a síntomas que presentaren y que pudieran poner en peligro a otras. En tal sentido, resulta valioso e interesante la implementación de un sistema capaz de monitorear el seguimiento y comportamiento de casos de alerta COVID (trazabilidad) y así poder identificar a las personas con las cuales interactuó en un periodo determinado. Esta medida podría permitir implementar a tiempo acciones de cuidado que minimicen el riesgo de contagio y a su vez llegar rápidamente a las personas con las que tuvo contacto estrecho y a las cuales eventualmente pudiera haber contagiado. La cantidad de sistemas de trazabilidad relacionados a la detección de casos de COVID -19 no es tan extensa, pero si han sido relevantes para contener y aislar a los casos positivos, ejemplos como los de China, Corea del Sur y Singapur son de los más relevantes y con un impacto positivo luego de ser implementados [9]. En Chile el método utilizado fue por medio de las autoridades municipales que integran la Atención Primaria de Salud (APS) y la SEREMI de Salud por medio de un call center, método de relativa o cuestionada efectividad ya que dependen de la asistencia de personal calificado para confirmar los casos y antes de ser notificados podían pasar días de espera [10].

La propuesta del sistema de trazabilidad presente en este informe mejora en gran manera el sistema que se utilizó para realizar un seguimiento de casos, al implementar el IoT. Junto a su principal ventaja de automatizar procesos, se acelera, además, la recolección de información y su envío a las autoridades pertinentes.

En la actualidad, la situación nacional ha mejorado considerablemente con la implementación de un sistema de vacunación, pero no es la realidad de otros países, en que la pandemia ha golpeado fuerte, con altas tasas de mortalidad y sin poder determinar al corto plazo, cuando podrán retomar la normalidad.

1.3 Trabajos Previos

Para la elaboración de este trabajo de investigación se pone en contexto la crisis sanitaria que afecta a la humanidad en estos días debido al COVID-19, la situación pasada y su impacto a nivel mundial. A esta investigación se suma el estudio de las medidas tomadas por los primeros países que se vieron afectados por este virus. Junto a esto se considera el análisis de los medios de muestreo y toma de datos sobre el virus, como el uso de cámaras térmicas en aeropuertos para poder verificar las temperaturas de personas, el cómo han ayudado para prevenir y controlar el flujo de gente portadora del virus, y qué tan vulnerable puede llegar a ser nuestro criterio al realizar mediciones.

A su vez, se hace seguimiento y estudio de la evolución de la pandemia a lo largo de los meses y como esta crisis ha impulsado el uso de tecnologías móviles de uso médico para realizar seguimiento y chequeos de salud a personas con otras afecciones de base, como las enfermedades crónicas, o incluso el seguimiento de los afectados por el virus.

- En [9], se hace mención al impacto que hubo al implementar sistemas de control para identificar posibles casos de COVID-19 por medio de aplicaciones móviles en diversos países como China, India, Alemania, Reino Unido y Estados Unidos. Se muestra cómo estos sistemas operan y las cualidades tecnológicas como herramientas para frenar la propagación del virus. Además, se compara la relevancia del cambio en la recolección de información del rastreo de casos y la importancia del cambio tecnológico en esta área.
 - En [11], se ha realizado un resumen sobre la situación actual a nivel mundial, reuniendo información sobre métodos usados, seguimiento, situaciones clínicas, análisis estadísticos sobre el virus y como ha ido avanzando y evolucionando, con la finalidad de ser un paper recopilatorio con información útil sobre la situación actual, medidas tomadas, y esperanzas a futuro.
- Entre estos está el uso de distintos dispositivos como drones, aplicaciones médicas enfocadas al control de la pandemia, como control de casos, búsqueda de casos aislados en el hogar y notificando de forma remota la condición.

- En [12], se presentan las medidas que fueron tomadas por parte de Corea del Sur y como controlaron la pandemia en sus primeras etapas, debido a su previa experiencia con el brote del Síndrome Respiratorio del Medio Oriente (MERS) durante el año 2015. Cabe destacar que en Corea del Sur la privacidad de datos no es un problema como ocurre con otros países del mundo como lo son los europeos y americanos, donde se ven afectados por el “Reglamento General de Protección de Datos”, definido en el año 2018 y que busca reforzar los derechos a la privacidad de datos de las personas. Corea del Sur al tener acceso de los datos de sus habitantes, hizo uso del GPS de los dispositivos móviles, y de esta forma notificar de forma arbitraria, quienes habían estado cerca de una persona diagnosticada con COVID-19 y en qué ciudades o lugares se habían registrado alzas de casos positivos.
- En [13], se presentan los beneficios que ha aportado la adopción del Internet de las cosas Médicas (IoMT) en la cultura actual, junto con la inteligencia artificial y el Big Data para afrontar las consecuencias de la pandemia del COVID-19. A su vez se menciona la importancia para el personal médico el monitoreo de las condiciones de salud de diversos pacientes de forma remota. El impacto producido por otras enfermedades como el SARS-CoV en Asia, el MERS-CoV en el oriente medio y el virus del dengue en Singapur, demuestra la importancia de la IoMT para contrarrestar las propagaciones de virus en el futuro.
- En [14], se expone una aplicación de monitoreo llamada Coronapp para seguimiento de mutaciones de la enfermedad del SARS-CoV-2. La información es proporcionada por los mismos usuarios de la aplicación. Su finalidad es poder generar un registro y monitoreo de datos que apoye y beneficie a los laboratorios médicos en primera línea, pudiendo definir con mayor rapidez las mutaciones del virus y en qué partes del globo se expande.
- En [15] se exhibe el impacto que tendrían las diversas tecnologías de trazabilidad empleadas en otras partes del mundo si se aplicaran en territorio chileno. Se evalúa la rentabilidad, los aspectos legales respecto a la privacidad de los datos y los peligros que conllevan los sistemas muy intrusivos respecto a la información personal del usuario.

- En [16] se expone un estudio respecto a la viabilidad de los sistemas de trazabilidad en la actualidad para detener las propagaciones de los virus, apelando a la necesidad de una intervención política en la recolección de información y que esto se integre en los sistemas de salud. También se realiza mención a la aplicación **NHS COVID-19** de Reino Unido y como esta genera una mayor efectividad en la notificación a los usuarios que han estado en contacto con personas contagiadas que un sistema manual tradicional.

1.4 Análisis trabajos previos.

Los sistemas de trazabilidad son fundamentales para poder realizar un monitoreo constante de las personas que pueden ser casos positivos de la enfermedad del COVID-19. Esto se logra gracias a la implementación del IoT y a su cadena de valor, haciendo uso de diversos sensores, actuadores o plataformas digitales dependiendo de cada sistema, pudiendo controlar los niveles de contagios al identificar con mayor rapidez los contactos estrechos de los contagiados.

1.5 Hipótesis de trabajo.

Utilizando IoT basado en el uso médico y en sus características de mejorar y automatizar procesos, se puede realizar un sistema de trazabilidad capaz de detectar posibles casos de COVID-19 por medio de la temperatura corporal. Así, se puede disminuir y moderar la propagación del virus.

1.6 Objetivos

1.6.1 Objetivo General

Diseñar un sistema de trazabilidad no invasivo para monitorear posibles casos de COVID-19 en un ambiente controlado, haciendo uso de sensores, actuadores, aplicaciones digitales y dispositivos electrónicos.

1.6.2 Objetivos Específicos

- Revisar bibliografía relacionada con sensores de temperatura infrarrojos y de plataformas asociadas al internet de las cosas (IoT), y el uso de tecnologías de la información y la comunicación (TICs) en el monitoreo de COVID-19.
- Diseñar un sistema capaz de identificar y monitorear los posibles casos de COVID-19.
- Generar una base de datos con información de personas ficticias para agilizar la detección de contactos estrechos de posibles casos de COVID-19.
- Implementar un prototipo basado en el sistema de monitoreo de COVID-19 propuesto.
- Realizar pruebas de carácter físico del sistema de monitoreo de COVID-19.

1.7 Alcances y Limitaciones

El alcance de este proyecto será el diseño y desarrollo de un sistema de trazabilidad en un ambiente controlado, para realizar pruebas de carácter físico y verificar la implementación de un sistema eficiente por medio del uso del IoT. Para el desarrollo de las pruebas se tomará en consideración la temperatura corporal de un adulto sano y el uso de plataformas digitales para la obtención de datos. Una limitación en este trabajo fue la situación sanitaria en la Región del Bio Bío que restringió la disponibilidad de poder realizar pruebas en laboratorio.

El sistema de trazabilidad diseñado en este proyecto puede ser derivado a otras funciones, tales como: el seguimiento de insumos médicos, monitoreo y seguimiento de paquetes por correspondencia y seguimiento de las mantenciones realizadas a equipos o máquinas de una empresa.

1.8 Metodología

1. Revisión bibliográfica acorde a la contingencia mundial debido a la aparición y propagación del virus SARS-CoV-2 teniendo un enfoque en cómo se puede usar la tecnología en pos de frenar el impacto de la pandemia.
2. Revisión bibliográfica sobre programación de sensores térmicos para realizar mediciones de la temperatura de las personas como principal variable para identificar casos de COVID-19.
3. Revisión bibliográfica sobre programación de sensores de movimiento y de proximidad para el desarrollo del sistema de aforo.

4. Revisión bibliográfica sobre programación de módulo RFID para el desarrollo e inclusión del sistema de identificación y trazabilidad de usuarios.
5. Revisión bibliográfica de plataformas digitales como Blynk, Adafruit.IO y Firebase con la finalidad de usarlas como registros, bases de datos y bitácoras digitales de diferentes datos de interés.
6. Realización de experimentos con sensor térmico MLX90614 para la detección de temperatura, tomando como referencia los valores de un termómetro infrarrojo y calibrando el sensor.
7. Realización de experimentos con cámara térmica AMG8833 con la finalidad de obtener una medición más precisa, haciendo uso de los valores referenciales del sensor térmico MLX90614 y el termómetro infrarrojo, limitando ciertos valores para una correcta medición.
8. Realización de experimentos con sensor PIR y sensor de ultrasonido para diseño y programación del sistema de control de aforo mediante conteo de ingresos y egresos de un determinado lugar
9. Realización de experimentos con módulo RFID para el diseño y programación del sistema de identificación y trazabilidad de personas, pudiendo identificar y diferenciar usuarios con acceso autorizado o denegado.
10. Realización de experimentos relacionados con el registro de la temperatura, capacidad, ingresos autorizados, lugar de ingreso, contactos estrechos por medio de las diferentes plataformas digitales.
11. Análisis de resultados de sistemas implementados a nivel individual y en conjunto, verificando la correcta recopilación de información y aplicación de la base de datos del proyecto.

2.2 Sensores.

Los sensores son dispositivos capaces de magnificar diferentes parámetros físicos de interés. A su vez, son capaces de convertir esta medición a una señal que pueda ser interpretada, tanto por las personas como por distintos dispositivos o sistemas compatibles. Entre los diferentes tipos de sensores están [20]:

a) **Sensores de Temperatura:** Son capaces de monitorear variaciones en la temperatura de forma remota (ambiente) o por contacto (superficial) y la transforman en una señal eléctrica. Entre estos sensores se destacan los termistores por su funcionamiento y los sensores infrarrojos sin contacto para realizar mediciones de alto rendimiento a objetos inaccesibles o en movimiento. Entre sus aplicaciones se encuentra el uso en plantas de procesos químicos, motores, computadores, etc.

b) **Sensores de movimiento:** Son capaces de identificar cualquier tipo de movimiento dentro de un área determinada, mediante el uso de emisores y receptores de señales, según las características de cada transductor se emitirá una señal en particular. Entre estos dispositivos se destaca el sensor infrarrojo pasivo (PIR). Estos sensores operan por medio de lectores infrarrojos que detectan el movimiento debido a las variaciones bruscas de temperatura [21].

Entre sus aplicaciones se encuentran los sistemas de alarma, iluminación, activación de procesos, etc.

c) **Sensores de color:** Obtienen el color de un objeto analizando y calculando coordenadas cromáticas de referencia debido a la radiación reflejada. Estos sensores en particular trabajan con el modelo RGB (Red, Green y Blue).

Las aplicaciones que poseen estos tipos de sensores son la detección de colores, detección de marcas, clasificación, entre otras [22].

Para la elección de un determinado sensor, se deben identificar 3 simples factores de análisis [23]:

1. Tipo de medición que se requiere.
2. Tipo de salida que requiere el sensor.
3. Selección de posibles sensores que cumplan con lo requerido, de los cuales se consideran parámetros tales como el intervalo, la exactitud, linealidad, velocidad de respuesta, costo, disponibilidad, etc.

2.3 Actuadores y periféricos.

2.3.1 Actuadores.

Dentro del mundo de IoT, uno de los componentes que conforman este sistema son los actuadores, estos son dispositivos que cuentan con la capacidad de transformar energía para activar dispositivos o sistemas, generando un efecto sobre un proceso automatizado [24].

En resumen, los actuadores son dispositivos que permiten realizar una acción luego de que un estímulo es captado por los sensores y que es procesado por la plataforma programable. Tomando como ejemplo la temperatura ambiente, su valor es enviado a la plataforma programable para compararlo con referencias definidas por el usuario y en consecuencia ejecutar algún tipo de acción. Por ejemplo, encender un aire acondicionado si la temperatura ambiente es superior a cierta referencia o en caso contrario apagarlo si la temperatura es inferior a la referencia.

Cabe destacar, que cualquier dispositivo que pueda conectarse a la “red” y pueda ejecutar algún tipo de acción por ejemplo encender una luz, que un interruptor se active o desactive, entre otros puede ser considerado como un actuador.

Existen tres distintos tipos de actuadores que son utilizados en IoT, los cuales son:

- a) **Hidráulicos:** Hacen uso de la presión de líquidos para realizar algún movimiento mecánico.
- b) **Neumáticos:** Hacen uso de aire comprimido para permitir el funcionamiento mecánico.
- c) **Eléctricos:** Son actuadores que convierten la energía eléctrica en alguna función mecánica.

El tipo de actuador es completamente independiente de cómo se realiza la acción, ya que su función principal es recibir una señal, y en base a esta señal, convertirla en una acción. Pero, estos dispositivos por lo general no tienen la función de procesar los datos recibidos, sólo cumplen lo establecido por algún tipo de controlador, teniendo como esencia una entrada (sensores), y salidas (actuadores).

2.3.2 Periféricos.

Se le denomina periférico a un dispositivo auxiliar e independiente conectado a la plataforma de control. Estos dispositivos tienen como cualidad la de poder comunicar al exterior ciertos comandos que se indiquen por medio de la plataforma. A diferencia de los actuadores que proceden a realizar una acción ya definida a nivel de programación, los periféricos pueden presentar información respecto a lo detectado por algún sensor, alguna condición ya definida o el funcionamiento de un actuador. También permite que el usuario pueda interactuar con la plataforma programable. Y por último pueden cumplir como dispositivos que pueden almacenar información, siendo una memoria auxiliar para el sistema [25]. Algunos ejemplos de periféricos son:

- A. Pantallas LCD u OLED.
- B. Teclados.
- C. Memorias externas.
- D. Cámaras.
- E. Pantalla táctil.
- F. Display numéricos.

2.4 Plataformas.

2.4.1 Plataformas Digitales

El desarrollo de distintos tipos de plataformas ha permitido el desarrollo virtual y/o digital, tanto en procesos como en almacenamiento de información. Si a esto le sumamos la incorporación de los sensores y la plataforma programable se obtiene el sistema IoT.

Para este caso una plataforma de IoT, no es algo tangible o físico, ya que considera distintos tipos de software, que funcionan como un tipo de puente entre los sensores y los actuadores. Estos tipos de plataformas tienen diversos tipos de niveles de complejidad y utilidad. Algunas tienen un muy bajo nivel de programación, y sólo sirven como vínculo, mientras que otras son complejas y trabajan con un lenguaje de programación propio o proveniente de algún otro.

Debido a que estas plataformas se desarrollan y crecen a la par que avanza el internet, será complicado elegir sólo una como referencia. Para lograr una adecuada toma de decisión, se pueden considerar tres puntos: **i)** presupuesto, habilidades de desarrollo y capacidades internas, **ii)** modelo de negocios y sus necesidades específicas y **iii)** tiempo o dinero mencionadas por Link Labs [19].

Entre estas plataformas digitales, se pueden encontrar diversas opciones como Zapier, IFTTT, Adafruit IO, Ubidots, destinadas a la simplificación de comandos, siendo en casos, meros puentes para conectar las respuestas entregadas por sensores, y los actuadores. Mientras que otras empresas se centran en desarrollar sus propias plataformas como servicios, conocidas como Backends, que son infraestructuras como servicio que proporcionan alojamiento de potencia de procesamiento para aplicaciones y servicios [26].

Hace unos años atrás estos Backends eran usados y optimizados para funciones de escritorio y para móviles, pero ahora con la masificación de IoT, la venta de estos dispositivos en el mercado ha ido al alza en los últimos años. Por consiguiente, las empresas desarrolladoras les gusta referirse a sus Backends como plataformas IoT.

A su vez los Backends poseen su contraparte, que serían los frontend, mayormente conocidos como el **Lado del Cliente**. En resumidas palabras es toda la interacción que tendrá el usuario con cualquier plataforma que él interactúe, es la “cara” visible de las diversas plataformas, los cuales pueden ser colores, tipos de letras, imágenes, efectos del ratón, y las interacciones que tiene el usuario con el servicio, entre otras particularidades. La principal función de los front end es la de transmitir una buena experiencia y ser cómoda a los usuarios.

Un ejemplo de la relación entre ambos conceptos es Alexa, que es un asistente virtual que presta un servicio de ayuda y simplificación a los usuarios.

2.4.2 Plataformas de control

Las plataformas de control a diferencia de las plataformas digitales donde reside el funcionamiento y la operación de IoT son físicas y se encarga del control de los sensores y de los actuadores, siendo la columna vertebral de los sistemas basados en la automatización. Entre ellas, se encuentran las plataformas Arduino y Raspberry, siendo las más comunes y sencillas de programar y operar.

- **Arduino Uno:**

Arduino es una plataforma de hardware libre y software de código abierto diseñada para facilitar el uso de electrónica en proyectos multidisciplinarios. Está basada en una sencilla placa con entradas digitales y analógicas, donde su diseño y distribución le permiten llegar a ser utilizada para el desarrollo de cualquier proyecto sin necesidad de licencias externas (ver Fig 2). Este dispositivo puede ser usado en diferentes aplicaciones y campos de uso, cuenta con variados modelos de placa, que se diferencian en su tamaño, conectividad, cantidad de entradas y salidas (tanto analógicas como digitales) y especificaciones más técnicas como el tamaño de su memoria interna.



Fig. 2. Placa Arduino Uno. Fuente: [27]

● Raspberry:

Se puede considerar a esta plataforma como una computadora de tamaño compacto, la cual puede operar de la misma forma que lo haría una computadora de escritorio. Según especificaciones de la página oficial [26], esta placa posee integrados varios puertos y entradas, dos de tipo USB, uno de ethernet y salida HDMI, permitiendo de esta manera conectar otros dispositivos como el teclado, ratón y pantalla (ver Fig 3). A diferencia del Arduino, éste tiene un comportamiento más complejo debido a su funcionalidad similar a una computadora, donde se debe tener un conocimiento previo de programación o computación para un trabajo óptimo.



Fig. 3 Placa Raspberry. Fuente: [26]

2.5 Protocolos de comunicación.

Dentro del mundo de la automatización existen diversas formas de realizar una comunicación entre la plataforma y los actuadores. Al referirse al esquema de trabajo IoT, estos protocolos de comunicación deben de ser inalámbricos, entre los cuales resaltan 3 tipos: Wi-fi, Bluetooth, y Zigbee, siendo la comunicación inalámbrica la que prescinde de cables o medios físicos de programación para hacer una conexión entre el punto emisor y el receptor.

● Wi-Fi.

Esta es una tecnología de comunicación inalámbrica entre dispositivos electrónicos que pueden ser computadores, tablet, celulares y dispositivos de programación como las plataformas mencionadas anteriormente (ver Fig 2 y Fig 3). Mediante el uso de radiofrecuencia para comunicarse con los dispositivos receptores, el servicio de Wi-Fi necesita un equipo conectado a internet y de una antena que distribuye la señal dentro de un radio determinado. Este sistema de comunicación permite la conexión de múltiples dispositivos en la misma red tal como se presenta en la Fig 4, cuyo principal problema es la distancia que hay entre el equipo que distribuye la señal (router) y el elemento que

dispone de esta conexión inalámbrica. Esto debido a que a menor distancia entre estos elementos, mejor será la comunicación y mejor será la respuesta deseada [28].



Fig. 4. Ejemplificación Comunicación Wi-fi. Fuente: [28]

● **Bluetooth.**

Esta es una tecnología de comunicación inalámbrica de corto alcance entre dispositivos, como celulares, computadores, televisores, equipos de música, entre otros, con un radio de operación y de comunicación de 10 metros aproximadamente.

De manera habitual la comunicación Bluetooth se produce entre 2 dispositivos únicamente, pero existen otros dispositivos capaces de conectarse de manera simultánea a más dispositivos y de manera eficiente.

● **Zigbee.**

Es el nombre usado para referirse al conjunto de protocolos de alto nivel de comunicación inalámbrica. Esta tecnología tiene un mayor impacto en el área de domótica y se usa para la radiodifusión digital de datos para ahorrar lo máximo posible en energía, basándose en los estándares de la IEEE. Una de sus principales características es su sencillez y bajo costo que representa su elaboración, comparándolo con otros medios de comunicación inalámbrica, siendo el más económico y el que reduce el coste total de la energía utilizada.

La tecnología cuenta con nodos que se conectan entre sí cumpliendo diversas acciones. Existen tres tipos de nodos, los tipos controlador, enrutador y dispositivo final. El nodo controlador es el más importante de la red. Su función es organizar las rutas y solo existe un en cada tipo de red. El nodo enrutador es parecido al controlador, aunque su función es poder conectar dos o más dispositivos que no están conectados entre sí. Por último, el dispositivo final son los actuadores y sensores de la red. Esta tecnología puede separarse en distintas topologías, como estrella, árbol o malla, siendo este último el más utilizado e interesante. En una red tipo malla los nodos de tipo enrutador están comunicados entre sí, de esta forma se crean diversas rutas para tener una comunicación estable, en caso de que alguno de estos nodos falle la red es reorganizada por el controlador. [29]

2.6 El internet de las cosas médicas.

El internet de las cosas Médicas o IoMT (Internet of Medical Things), es una de las tantas aplicaciones que se le puede dar a IoT, específicamente en el área de la salud. Dentro de esta área se encuentran los dispositivos como las aplicaciones médicas que se conectan a plataformas de nube como lo es Amazon Web Services, donde la información es almacenada y luego analizada para la toma de decisiones. IoMT tiene diversas aplicaciones como la telemedicina, los dispositivos médicos para medir datos vitales o Wearables, drones, realidad aumentada, inteligencia artificial y Big Data. Todas estas aplicaciones tienen como fin apoyar asistencialmente a un usuario, tal como se presenta en la Fig. 5.

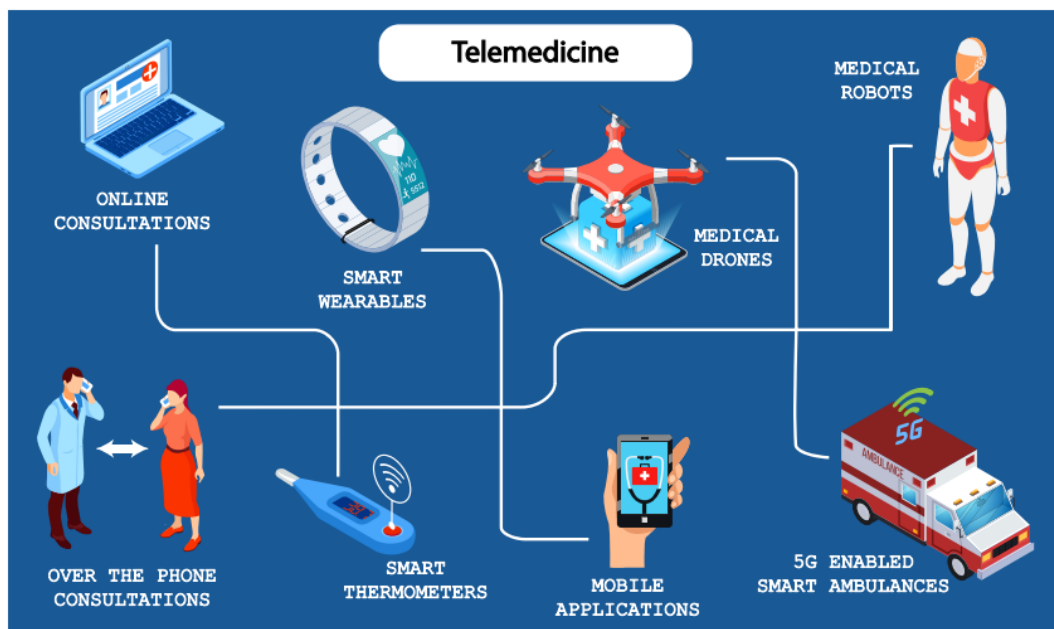


Fig. 5. Tipos de tecnologías aplicadas en IoMT. Fuente [11]

Las ventajas del IoMT que se pueden identificar son 4, cuyo objetivo final es mejorar el tratamiento de los pacientes y su calidad de vida. Junto a esto, se aumenta la eficiencia de los sistemas sanitarios y se reducen costos. Al combinar el espacio físico con el digital, se **transforma la relación médico-paciente** [30].

- 3.1 **Mejorar Diagnósticos:** En sentido práctico, es poder hacer test o pruebas médicas a distancia gracias al uso de dispositivos Wi-fi. De esta forma se agilizan las tomas de muestras y la pronta etapa de análisis, en caso como el COVID-19, reduciendo el contacto y evitando nuevos casos.
- 3.2 **Monitorizar las enfermedades:** Principalmente es poder hacer un seguimiento al paciente para enfermedades temporales o crónicas, donde su estado debe ser constantemente monitoreado.
- 3.3 **Hacer un seguimiento de los datos:** Los diversos equipos o dispositivos usados se pueden usar para poder hacer un levantamiento de datos más asertivo y en tiempo real. Esto ayudaría a personalizar los distintos tipos de tratamientos, ayudando así a la toma de decisiones más precisas ante el historial de datos.
- 3.4 **Registrar la información:** Es la recopilación de datos en masa o big data. En casos especiales como la actual pandemia mundial, se puede recoger una gran cantidad de información que puede ser ocupada para hacer un seguimiento o rastreo de la situación, viendo cómo avanza y en consecuencia tomar medidas preventivas cuando sea posible.

Entre los principales dispositivos de mayor utilidad en el IoMT, están las aplicaciones médicas tipo apps móviles. Por ejemplo, la aplicación Mediclic, que tiene como iniciativa tener un doctor en la palma de tu mano y la aplicación e-Alert realizan un monitoreo del funcionamiento de los equipos de resonancia magnética. El uso de Wearables y también el uso de drones para realizar entregas de insumos y productos médicos, como lo serían medicamentos y muestras de todo tipo. En definitiva, el uso de IoMT es crucial para el desarrollo médico hoy en día, considerando que estas tecnologías han tenido un alza increíble debido a la pandemia, siendo herramientas vitales y necesarias en la actualidad y situaciones futuras [11].

2.7 Sistema de trazabilidad.

El concepto de trazabilidad es un término que apareció en 1996 en respuesta a una fuerte crisis sanitaria europea y al descubrimiento de la enfermedad de las Vacas Locas. A lo largo del tiempo se le han dado diferentes definiciones a la trazabilidad, considerando la situación sanitaria debido al COVID-19 la trazabilidad se puede definir como: “Proceso que permite identificar de manera continua a las personas que tuvieron contacto con un caso contagiante” [31].

2.7.1 Trazabilidad en el contexto del COVID-19.

Considerando la situación mundial debido a la pandemia del COVID-19, es que ha sido una necesidad fundamental el poder desarrollar e implementar distintos sistemas de trazabilidad aplicados al control y detección de casos de COVID-19, para identificar de manera continua y activa a toda persona que haya tenido contacto con un caso positivo y así poder evitar brotes como puntos de reunión.

Durante el tiempo que ha permanecido la pandemia se han intentado aplicar diversas tecnologías para poder controlar la propagación del virus (ver Fig. 6). Por ejemplo, aplicaciones móviles como WeChat y Alipay, ambas usadas en China y con la cualidad de utilizar el sistema de códigos QR. Así, los usuarios reciben un código estilo semáforo que les indica en base a los colores verde, si la persona está habilitada para viajar o rojo si es necesario que deba estar en cuarentena durante el tiempo correspondiente.

Junto a estas dos aplicaciones también se debe de destacar las aplicaciones de Trace Together de Singapur, Aarogya Setu de la India, NHS COVID-19 de Reino Unido o CovidSafe creada por la Universidad de Washington, cuya función es rastrear posibles casos de COVID-19 de manera voluntaria [9].

	App Name	Developer or Country	Purpose		Technologies Used			Data Storage		Participation		Government access
			Proximity-based exposure notification	Digital contact tracing (DCT)	Bluetooth LE	GPS	SMS	Centralized	Decentralized	Mandatory (actually or functionally)	Voluntary/opt-in	
Intervention Type	Max	WeChat / Alipay	China	✓	✓	✓		✓		✓		Data comes from government sources; location data sent to police
	Middle Ground	Trace Together	Singapore		✓	✓			✓		✓	Mandatory government access if positive
		NHSX/Oxford	Oxford		✓	✓		✓			✓	Government maintains data
		NextTrace	Fred Hutchinson Cancer Research Center		✓		✓				✓	Government maintains data, but no storage
		COVID SafePaths	MIT		✓	✓			✓		✓	Voluntary upload by users who test positive
		Aarogya Setu	India		✓	✓		✓	✓		✓	Anonymized, aggregate
		Care19	North Dakota	✓	✓	✓		✓	✓		✓	In aggregate; optional if positive
	Minimal	CovidSafe	Univ of Washington	✓	✓	✓			✓		✓	None
		CovidWatch	Univ of Stanford & Univ of Waterloo			✓			✓		✓	To validate test results
		CoEpi	CoEpi	✓		✓			✓		✓	Opt-in to share BT and symptom log with CoEpi server
		ito	Germany	✓		✓			✓		✓	None; positive results to ito server

Fig. 6. Tipos de aplicaciones y sus principales funciones. Fuente [9]

Aunque estas aplicaciones resultan muy útiles para lograr controlar la propagación del virus, muchas de estas son voluntarias, dejando completamente a elección del usuario el descargar y usarlas. A diferencia de países como Corea del Sur, Polonia, Israel y China que hizo obligatorio el uso de este tipo de sistemas, esta obligatoriedad en el uso de aplicaciones no es replicable en todo el resto del mundo debido a los derechos de privacidad de datos y al recelo existente por la manipulación de información privada como sus direcciones, cuentas, nombres, direcciones entre otras, o de información menos relevante como la ubicación vía GPS de la persona.

2.7.2 Riesgos a la privacidad.

Los riesgos ante la privacidad de los datos de una persona pueden ser de carácter directo o indirecto. Todo riesgo a la privacidad de la persona es catalogado como directo, mientras que los riesgos indirectos se relacionan con los bienes que protegen la privacidad, como, por ejemplo: la autonomía de la persona y su poder de elección. Por esto cuando se plantea el desarrollo de alguna aplicación o sistema de trazabilidad, se debe de tener en consideración el impacto que tiene ante la privacidad de los usuarios [15].

Uno de los riesgos directos más relevantes a considerar es el término de la reidentificación del usuario. La identificación de usuario se define como “una entidad lógica utilizada para reconocer a un usuario en un software, sistema, sitio web o dentro de cualquier entorno de TI (Tecnología de la Información) genérico.” [32]. Esto permite que un usuario sea reconocido por medio de algún método de autenticación o por medio de nombres de usuarios y contraseña. De esta forma el riesgo de la reidentificación será el hurto de datos para hacer ingreso a diversas plataformas que utilice el usuario original de los datos, siendo el mayor riesgo dentro de la trazabilidad centralizada.

Por otra parte, los dos mayores riesgos indirectos a la privacidad serían la igualdad de oportunidad y la dictadura de datos. El primero es un riesgo que atenta contra la autonomía de las personas, pudiendo obtenerse beneficios de la salud, algún puesto de trabajo y créditos bancarios al copiar los datos de otros. Y el segundo es de carácter radical, debido a que las autoridades gubernamentales de algún país pueden tomar decisiones a su conveniencia al hacer uso de los datos de los habitantes de ese país.

2.8 Situación Mundial de la pandemia.

A nivel mundial, la pandemia del COVID-19 ha provocado una profunda crisis económica, afectando todos los ámbitos de la vida, el sector productivo y el sector de servicios en todos los países del mundo. La pandemia ha provocado cambios en los comportamientos de la sociedad. Las relaciones interpersonales se vieron perjudicadas, la asistencia a recintos académicos y laborales se vió limitada o anulada en el peor de los casos

Bajo la incertidumbre de este nuevo virus y quienes serían los más afectados por él, la **universidad estadounidense Johns Hopkins** publicó en octubre del 2019 el índice de Seguridad Sanitaria Global, posicionando 195 países en una tabla para medir su eficiencia ante posibles pandemias. El estudio arrojó que ningún país cumple con la nota máxima de preparación (100 puntos) y se aclara que actualmente *“los riesgos son magníficos por la globalización, la inestabilidad política, la urbanización, el cambio climático y los avances tecnológicos que hacen más fácil, barato y rápido crear patógenos”* [33].

Esta tabla de posición mide los siguientes grandes parámetros.

- a) **Prevención:** Prevención en la aparición o liberación de patógenos.
- b) **Detección e informe:** Detección temprana de contagios e informes sobre potencial epidemia y preocupación internacional.
- c) **Respuesta rápida:** Mitigación de la propagación de una pandemia.
- d) **Sistema de Salud:** Suficiente y robusto para tratar a los enfermos y proteger a los trabajadores de la salud.
- e) **Conforme a las normas internacionales:** Compromiso de mejorar la capacidad nacional, planes de financiamiento para abordar las brechas y adherirse a las normas mundiales.
- f) **Entorno de riesgo general:** Vulnerabilidades del país ante las amenazas biológicas.

En la Tabla 1 se muestran los 13 países que se encuentran “mejor” preparados ante una pandemia, mientras que 109 países están “más preparados” y 73 países están “menos preparados” (aquellos países con menos recursos y un sistema de salud precario o inexistente).

TABLA 1. Ranking de los 13 países mejor posicionados ante una posible pandemia a nivel mundial: Fuente: [33].

Ranking	País	Puntaje	Ingresos
1	Estados Unidos	83.5	Alto
2	Reino Unido	77.9	Alto
3	Países Bajos	75.6	Alto
4	Australia	75.5	Alto
5	Canadá	75.3	Alto
6	Tailandia	73.2	Medio alto
7	Suecia	72.1	Alto
8	Dinamarca	70.4	Alto
9	Corea del Sur	70.2	Alto
10	Finlandia	68.7	Alto
11	Francia	68.2	Alto
12	Eslovenia	67.2	Alto
13	Suiza	67	Alto

TABLA 2. Comparación Chile y Colombia en el ranking de países mejor preparados ante una pandemia, comparando con otros países latinoamericanos y China a nivel mundial: Fuente: [33].

Ranking	País	Puntaje	Ingresos
22	Brasil	59.7	Medio alto
23	Irlanda	59	Alto
24	Singapur	58.7	Alto
25	Argentina	58.6	Alto
26	Austria	58.5	Alto
27	Chile	58.3	Alto
28	México	57.6	Medio alto
49	Perú	49.2	Medio alto
50	Vietnam	49.1	Medio bajo
51	China	48.2	Medio alto
63	Rusia	44.3	Medio alto
63	Uganda	44.3	Bajo
65	Colombia	44.2	Medio alto

Tal como se presenta en la Tabla 2, las posiciones de los países latinoamericanos como Chile y Colombia están ubicados entre los que están más preparados ante una situación de pandemia. Chile es el tercer país de Latinoamérica en estar más preparados, sólo superado por Brasil y Argentina respectivamente. Colombia se encuentra en el octavo puesto entre los países latinoamericanos, superando a Panamá y Uruguay respectivamente.

Éste es un ranking bajo ciertas condiciones específicas, pero es preocupante, que en su mayoría ningún país cumpla con los 100 puntos, y aún más, que aproximadamente el 77% de los países en el ranking no cumplan con un mínimo de 50 puntos. Estos puntajes dan a entender o justificar de cierta forma la situación a nivel mundial al tener estos datos.

2.8.1 Situación mundial entre los años 2020-2021

A continuación, se presentan los datos relacionados a los casos confirmados a finales de 2020 y 2021 a nivel mundial y a nivel latinoamericano.

TABLA 3. Datos de contagios y muertes a nivel mundial y Latinoamérica en los años 2020 y 2021. Fuente: [34]

Año	Contagios nivel mundial.	Contagios nivel Latinoamérica	Muertes nivel mundial.	Muertes nivel Latinoamérica
<i>2020</i>	78.725.345	15.032.206	1.731.115	493.414
<i>2021</i>	257.579.393	46.868.051	5.151.560	1.540.788

En la Tabla 3 se presentan los datos referentes a la situación mundial y latinoamericana en los años 2020 y 2021. Al revisar los datos se observa un aumento considerable entre un año y el otro. Al analizar la cantidad de muertes entre un periodo y otro se puede ver cómo empeoró la condición, donde a finales del 2020 no se alcanzaba la cifra de 2 millones y ahora al compararse con los datos a finales del año 2021, la cantidad de muertes es de un poco más de 5 millones de personas asociadas al virus que representa un aumento de casi un 198%. Los contagios tampoco han bajado considerando el gran número de vacunas distribuidas a nivel mundial, llegando a ser aproximadamente 260 millones de personas contagiadas en el periodo 2021, lo que en porcentaje sería un incremento de aproximadamente un 227% en relación al periodo 2020.

2.8.2 Situación de la pandemia hasta julio del 2022.

En la Fig. 7 se presenta el mapa y los datos sobre casos confirmados y muertes de COVID-19 a nivel mundial y en la Fig. 8 se observa el mapa y los datos de casos confirmados y muertes de COVID-19 a nivel latinoamericano durante el primer semestre del año 2022. En relación a la información del año 2021 se puede observar un incremento significativo de casos tanto a nivel mundial como a nivel latinoamericano, aún si se consideran todas las medidas propuestas para contener el avance del virus presentadas por los distintos países. Se debe de considerar que durante el periodo del 13 de diciembre hasta el 14 de abril aproximadamente se alcanzaron los niveles de contagios más grandes a nivel mundial debido a la variante ómicron (ver Fig. 9). Al evaluar el incremento de casos confirmados entre los años 2021 y 2022 en porcentajes se obtiene que hubo un incremento de aproximadamente un 114% en relación al año 2021 a nivel mundial y de aproximadamente un 56% a nivel latinoamericano.

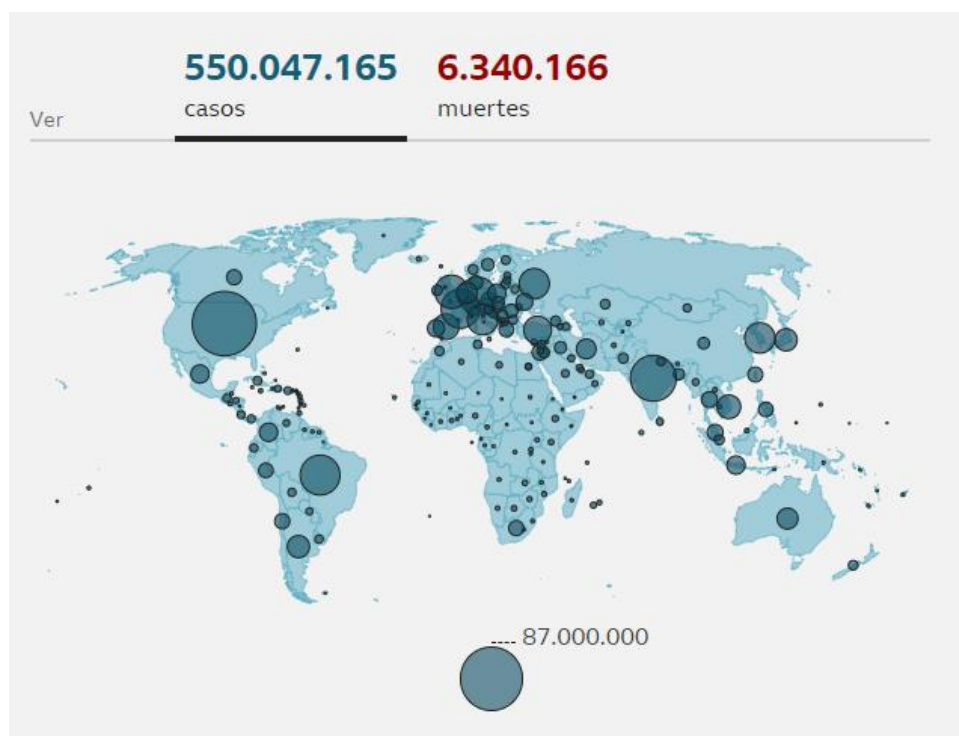


Fig. 7. Situación de casos confirmados de COVID-19, hasta el 5 de julio del 2022. Fuente: [34]



Fig. 8. Situación de casos confirmados de COVID-19 en América Latina, hasta el 5 de julio del 2022. Fuente: [34]



Fig. 9. Curva de casos confirmados entre el 13 de diciembre de 2021 y el 14 de abril 2022. Fuente [35]

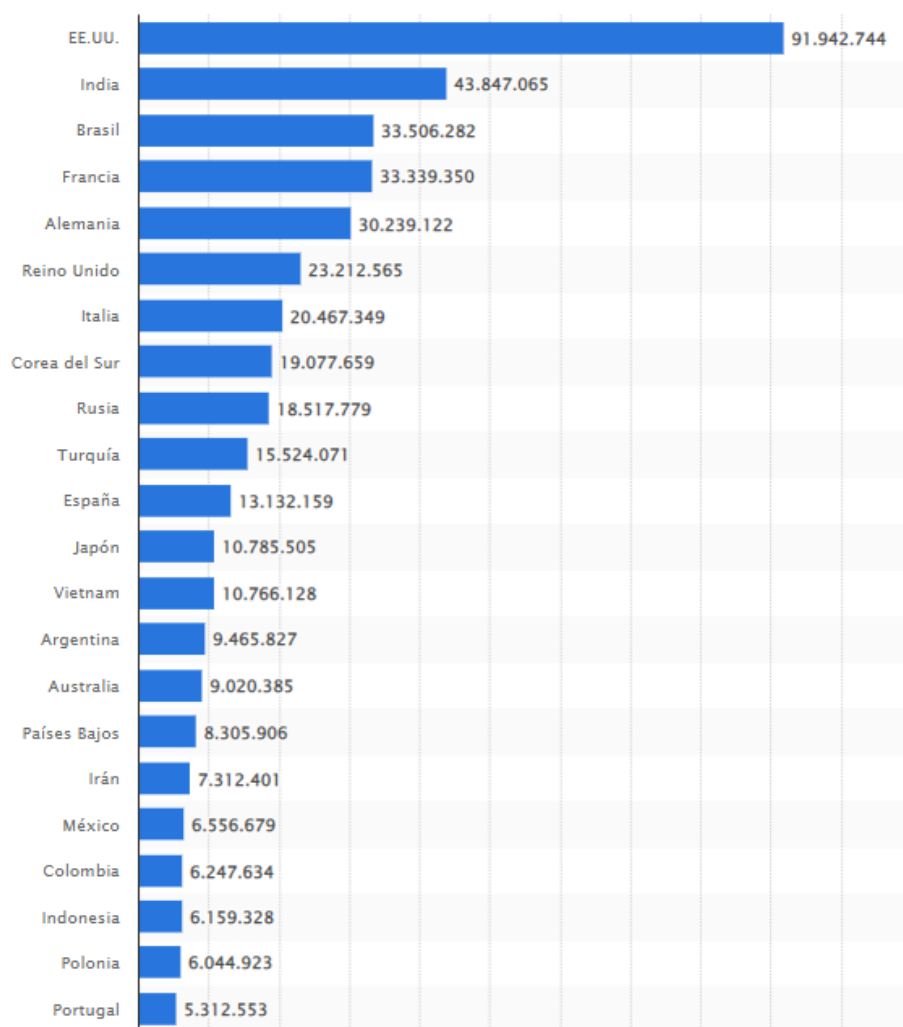


Fig. 10. Estadísticas de contagiados a nivel mundial, hasta el 22 de julio de 2022. Fuente [36]

En relación a la Fig. 10, se observa que tanto Estados Unidos, India y Brasil encabezan la lista de países con mayor número de contagios, siendo Estados Unidos el país con mayor número de casos acumulados por amplia diferencia.

Capítulo 3: Diseño

Teniendo en cuenta los conceptos y situación actual presentada en el capítulo 2, se describe a continuación de qué trata y qué se abordará durante el desarrollo del estudio.

El proyecto desarrollado consiste en realizar un sistema de trazabilidad que por medio de la temperatura de las personas sea posible identificar casos de COVID-19 e identificar la red de personas con las que se relaciona dentro de un espacio cerrado, como un edificio u oficina. Esta decisión fue tomada considerando los diversos síntomas posibles que tiene una persona al contraer el virus, los que se pueden ver en la Fig. 11, entre los cuales se identifican tres síntomas destacables: el primero es la fiebre, un síntoma capaz de ser medido por sensores térmicos, el segundo las complicaciones respiratorias, las que se podrían presentar como una simple tos, o en el peor de los casos una complicación respiratoria que afecte la inhalación y exhalación normal y tercero, un síntoma menos frecuente, pero presente en contagios COVID que es un malestar corporal generalizado que afecta de manera directa a los músculos.

Si se consideran estos tres síntomas, el más simple de evaluar es la fiebre, por ser un síntoma temprano y con un alto porcentaje de padecimiento en las personas sintomáticas. Al revisar diversas fuentes de literatura, se define la temperatura corporal promedio de un adulto sano como 37.0 °C. Según MINSAL, la temperatura que se considera como fiebre y puede asociarse con COVID es sobre los 37.8 °C [37].

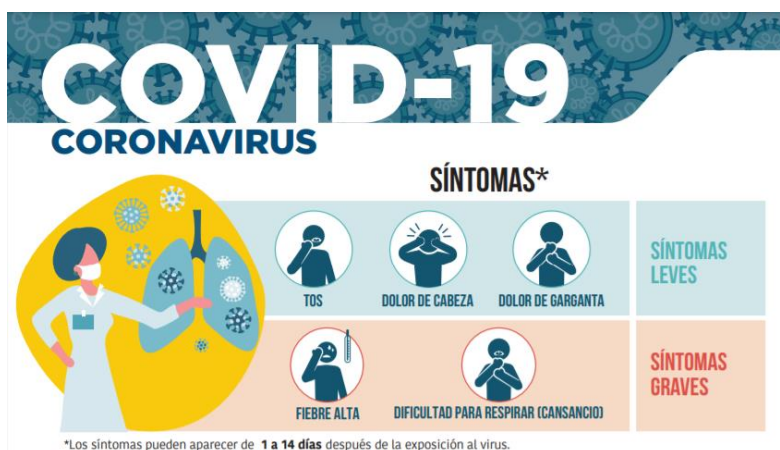


Fig. 11. Síntomas leves y graves COVID-19. Fuente [38]

A continuación, se presentará la estructura, tanto física como digital del sistema y cómo éste opera de forma secuencial, por medio de los diversos sensores y plataformas digitales, siguiendo así la cadena IoT.

3.1 Plataforma programable.

Tal como se indica en el capítulo 2, la cadena de valor IoT cuenta con 3 segmentos principales. La parte central es la plataforma de control, que es el medio programable y definirá el lenguaje de programación a ocupar. En este proyecto se usó la versión mejorada del NodeMCU ESP8266, que sería el ESP32 devKit presentado en la Fig. 12. Aunque tiene las mismas funciones principales que la plataforma antes mencionada, ésta arregla varios problemas propios de la ESP8266, entre los que se encontraban problemas de conexión, compatibilidad de librerías, cantidad de puertos programables y soporte.

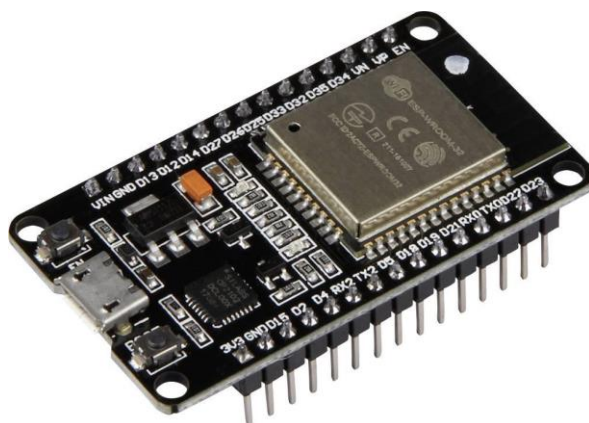


Fig. 12. NodeMCU ESP32. Fuente [39]

Se definió el ESP32 como la plataforma de control de este proyecto por ser más potente, tener más puertos, una velocidad de codificación más rápida, y al ser igualmente de fácil de programar que su antecesor.

3.2 Sensores.

Siendo el dispositivo sensible de la cadena IoT, éstos se encargarán de la interacción con el medio y de enviar una respuesta de forma digital a la plataforma de control.

El sistema contará con 4 distintos tipos de sensores y con opción de agregar un quinto en caso de realizar un cambio en la estructura del sistema. El grupo de sensores está compuesto por un sensor de ultrasonido, un sensor de temperatura infrarrojo, una cámara térmica y un lector de tarjetas RFID.

3.2.1 Sensor de ultrasonido.

Este dispositivo tiene un sistema emisor y receptor, donde la parte emisora emite un sonido que no es perceptible por el oído humano. Una vez que esta onda impacta contra un objeto, esta regresa y es recibida por la parte receptora. De esta forma se obtiene el tiempo que se demoró la onda entre ser emitida y ser recibida. Dicho tiempo es controlado por el pin “Echo”, que se activa para empezar el conteo y se desactiva una vez que el rebote de la onda haya sido captado por la parte receptora (ver Fig 13). Como ambas partes están alineadas y la velocidad de la onda no cambia, sólo deberá considerar la mitad del tiempo para obtener la distancia entre el objeto y el sensor. Por último, la magnitud en que trabaja el sensor está en la escala de los microsegundos (μs). La onda sónica se desplaza a una velocidad de 343 m/s , que al hacer la conversión a $\text{cm}/\mu s$, se obtiene que:

$$\frac{343\text{m}}{1\text{s}} = 0,0343 \frac{\text{cm}}{\mu s}, \quad (3.1)$$

Una vez obtenida la velocidad de la onda sónica y la relación del tiempo para calcular la distancia, se obtiene la siguiente ecuación:

$$\text{Distancia} = \frac{\text{tiempo}}{2} * \frac{1\text{cm}}{29,2\mu s}. \quad (3.2)$$

Donde

Distancia: es la distancia entre la parte emisora y el objeto.

Tiempo: el tiempo en μs desde que se emitió el sonido, hasta que se recibió.

Por medio de esta ecuación se obtiene la distancia en centímetros (cm). Pero como todo sensor, hay un límite y una tolerancia para una correcta medición; en el caso de los rangos de distancia eficientes del sensor, la precisión y su ángulo de medición se presentan en la Tabla 4.

TABLA 4. Tabla de valores del HC-SR04. Fuente: [40]

Variables	Parámetros
<i>Rango de detección</i>	2 a 400 cm
<i>Resolución</i>	3 mm
<i>Ángulo de visión</i>	15°

La finalidad del sensor de ultrasonido, es servir como base para un sistema de aforo para entrada y salida, y poder medir la distancia entre la persona y el sensor de temperatura.

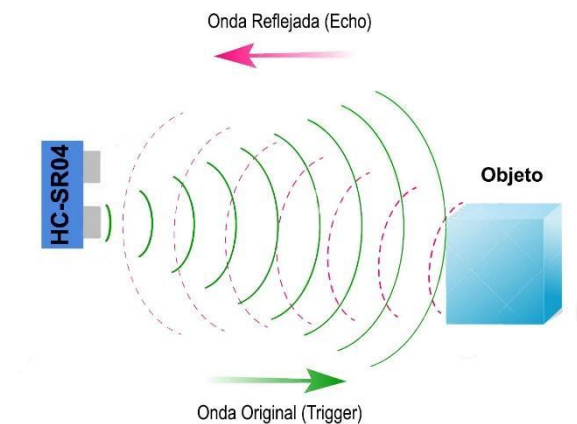


Fig. 13. Funcionamiento del sensor de ultrasonido. Fuente: [40]

3.2.2 Sensor de temperatura infrarrojo MLX90614

Este dispositivo es un sensor infrarrojo de tipo termopila. Recibe su nombre de termo por termocupla, y pila porque se genera un arreglo de estas termocuplas. Utiliza el protocolo de comunicación I2C. En la Fig. 14 se observa el modelo usado, el MLX90614.



Fig. 14. Sensor de temperatura infrarrojo MLX90614. Fuente: [41]

Su objetivo es realizar comparaciones de temperatura de la superficie objetivo y compararlo con una temperatura de referencia que puede ser la temperatura ambiente, agua fría o caliente, etc. Este sensor posee un ángulo de visión de gran tamaño (80°), dificultando que las mediciones sean del todo precisas. Existen versiones de este sensor aplicadas específicamente al área de la salud, donde el ángulo de visión es aproximadamente de 5° y con una calibración de fábrica que es bastante precisa entre el rango de temperatura de 35 a 38°C , con una precisión de $0,1^\circ\text{C}$.

El modelo usado para este proyecto cuenta con dos rangos de operación, uno para la temperatura ambiente con un rango de temperatura de entre los -40 a 85°C , y otro para la temperatura del objeto entre -70 a 382°C . Ambos calibrados de fábrica y con una precisión de $0,5^\circ\text{C}$.

Este sensor será utilizado para medir la temperatura mediante la lectura de la mano de la persona que desee ingresar al establecimiento.

3.2.3 Cámara térmica infrarroja AMG8833

Este módulo consiste de un arreglo de sensores de temperatura de tipo termopila formando una matriz que poseen el mismo principio del MLX90614. Su dimensión es de 8×8 , teniendo un total de 64 mediciones térmicas. Cada medición es independiente permitiendo captar un espectro más amplio. Igualmente, este sensor también funciona con el protocolo I2C, compartiendo puertos de entrada y salida con el MLX90614.

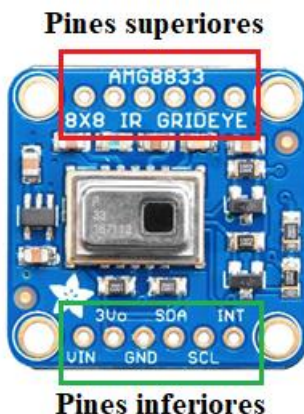


Fig. 15. Cámara térmica AMG8833. Fuente: [42]

En la Fig. 15 se puede notar que el sensor posee 12 pines. Los seis de la zona superior no tienen ningún tipo de conexión y sólo existen para dar estabilidad al dispositivo cuando se conecta a una protoboard. En cambio los otros seis pines inferiores se conectan a los distintos puertos de la plataforma, pudiendo operar con voltajes de 5V o de 3.3V. Los pines SDA y SCL son exclusivos de las conexiones I2C. El pin INT es opcional y funciona como un interruptor del sensor.

Los rangos de operación son bastante más acotados que el del MLX90614. Las condiciones óptimas de operación, se pueden ver en la Tabla 5. Aunque la distancia de percepción del sensor sea bastante amplia se recomienda realizar las mediciones cerca del lente para evitar tener mediciones poco significativas por la diferencia con la temperatura ambiente.

La finalidad que tiene este dispositivo es realizar la medición final de la persona luego de comparar con la medición del MLX90614.

TABLA 5 Tabla de valores de operación AMG8833. Fuente: [42].

Variables	Parámetros
Rango de operación	0 a 80 °C
Precisión	±2,5 °C
Distancia máxima	7 m
Ángulo de visión	60°

3.2.4 Módulo RFID.

La tecnología RFID (identificación por radiofrecuencia) ha ganado terreno que se refleja en un crecimiento bastante importante. Una de las principales razones es por su versatilidad para distintos usos, como control de abrir y cerrar puertas, controles de seguridad, identificaciones, para hacer seguimiento de productos, entre otras posibilidades. Los sistemas RFID se separan en dos partes: La primera corresponde al transpondedor o la etiqueta RFID, y la segunda parte es el transceptor o el lector, tal y como se observa en el kit básico RFID de la Fig 16. Viene con una tarjeta y un llavero que cumplen el papel del transpondedor y el lector RC522 del transceptor.

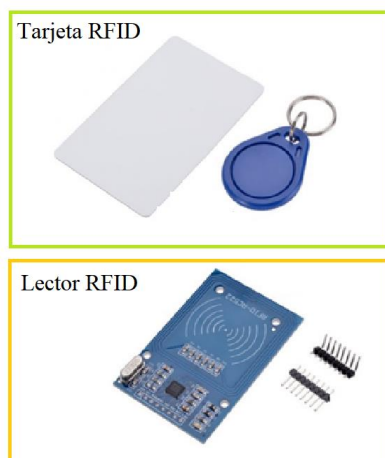


Fig 16. Kit RFID MIFARE RC522. Fuente: [43]

La diferencia más notoria entre etiquetas RFID es que una es de lectura y la otra puede ser de escritura y lectura. Las primeras sólo tienen información que consigue ser leída y que no puede ser alterada. En cambio, el segundo grupo, pueden cambiar su información base y realizar cierta edición en la etiqueta. El funcionamiento de las etiquetas es por medio del principio básico de la inducción, siendo consideradas pasivas sin necesidad de una batería ni ser previamente conectadas a la red para operar, a diferencia del lector. Otra de las diferencias que poseen estas etiquetas es que son separadas en grupos y funciones distintas. Estos grupos dependen de la frecuencia de la señal con la cual se comunican. Existen etiquetas de baja, alta y ultra alta frecuencia. En este caso de estudio se usarán altas frecuencias, siendo sólo compatibles con etiquetas RFID de 13.56 MHz. y aunque según el distribuidor, el rango de detección sería alrededor de 5-7 cm, se recomienda que se haga contacto con el lector para evitar problemas de lectura.

La utilidad dentro del sistema de trazabilidad es la de verificar los datos de los usuarios por medio de la UID “única” que tendría cada una de las tarjetas RFID. Habrá una base de datos donde se almacenará antecedentes personales y de esta forma tener un registro sobre dónde, a qué hora fue y con cuántas personas tuvo posible contacto.

3.3 Plataformas digitales.

Para este proyecto se consideró sólo el uso de plataformas digitales para recabar información obtenida de los sensores. La principal razón de esto es porque las plataformas digitales son las más fáciles de trabajar, debido a que sólo se necesita que tengan soportes con la plataforma y que permitan conectarse por el protocolo de comunicación designado.

Se usaron cuatro plataformas digitales que sirviendo como interfaz, base de datos y registro final.

Las plataformas digitales consideradas en este trabajo son:

a) Adafruit IO

Es una plataforma digital basada en el internet de las cosas, usando plataformas de control ya conocidas como Raspberry Pi, Arduino, entre otros. La interfaz de programación de aplicaciones (API) de comunicación está basada en el protocolo MQTT (cola de mensajes de telemetría y transporte) con servidores de la empresa Adafruit.IO. Es intuitiva para el usuario, permitiéndole crear en cosa de minutos un tablero con las funciones o monitores para cumplir una necesidad, como el ejemplo visualizado en la Fig. 17.



Fig. 17. Ejemplo de tablero en Adafruit.IO. Fuente: [44]

La particularidad de esta plataforma es que se pueden crear unas variables llamadas feeds que contienen los valores obtenidos por parte de los sensores. Estos cuentan con una cantidad máxima de 10, que pueden ser desplegados en los distintos tableros de la plataforma.

La utilidad de esta plataforma es crear un respaldo de la información obtenida del sistema de trazabilidad desarrollado, además de notificar en pantalla, las veces que se cumpla una condición definida por el usuario; para este caso en específico se plantea usar un medidor circular donde se definirá el porcentaje de aforo, una gráfica donde se verá el comportamiento de la temperatura a lo largo del día con un registro diario y un cuadro de texto que notifica las temperaturas y a la hora de la medición (ver Fig. 18).

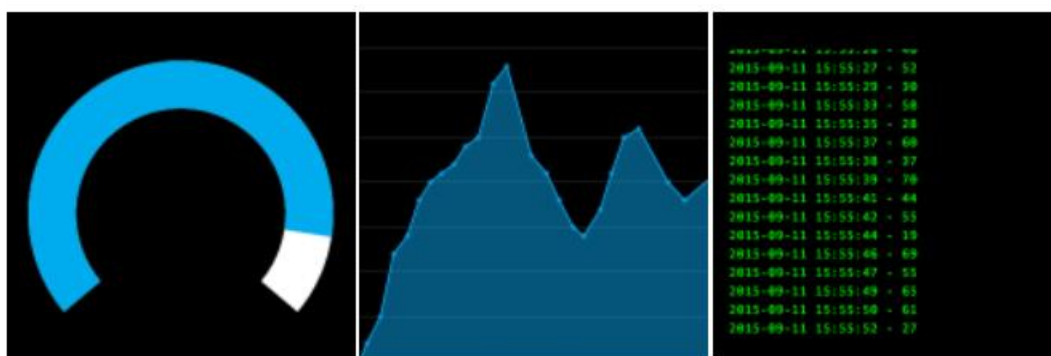


Fig. 18. Bloques de Adafruit.IO de medidor circular, gráfica temporal y cuadro de texto respectivamente. Fuente: [44]

b) Blynk.

Blynk es una aplicación móvil que tiene la ventaja de poder vincularse con diversas plataformas de programación, entre las que se destaca la ESP32 que se usará para el proyecto. Esta aplicación funciona como una bitácora física ya que puede ser revisada desde un celular o una tablet. Las limitaciones de la aplicación es que los dispositivos móviles deben ingresar con la misma cuenta para visualizar los mismos datos en tiempo real.

Una de las grandes ventajas que tiene esta aplicación es que el programador puede crear su propia interfaz que tiene diversos widgets presentando algunos en la Fig 19, que son componentes para realizar los proyectos, estos pueden ser botones, relojes, leds virtuales, cuadros de texto, etc. Este sistema de widgets posee dos grandes problemáticas: la primera es que para la versión gratuita (versión usada para este proyecto) existen restricciones en su uso siendo limitados por una moneda virtual finita y la segunda problemática, es que la energía es compartida entre proyectos, lo que vuelve difícil mantener dos interfaces de control si sólo se cuenta con la versión gratuita.

Para este proyecto sólo se hará uso de una interfaz de control siendo suficiente la moneda virtual habilitada para la versión gratuita.

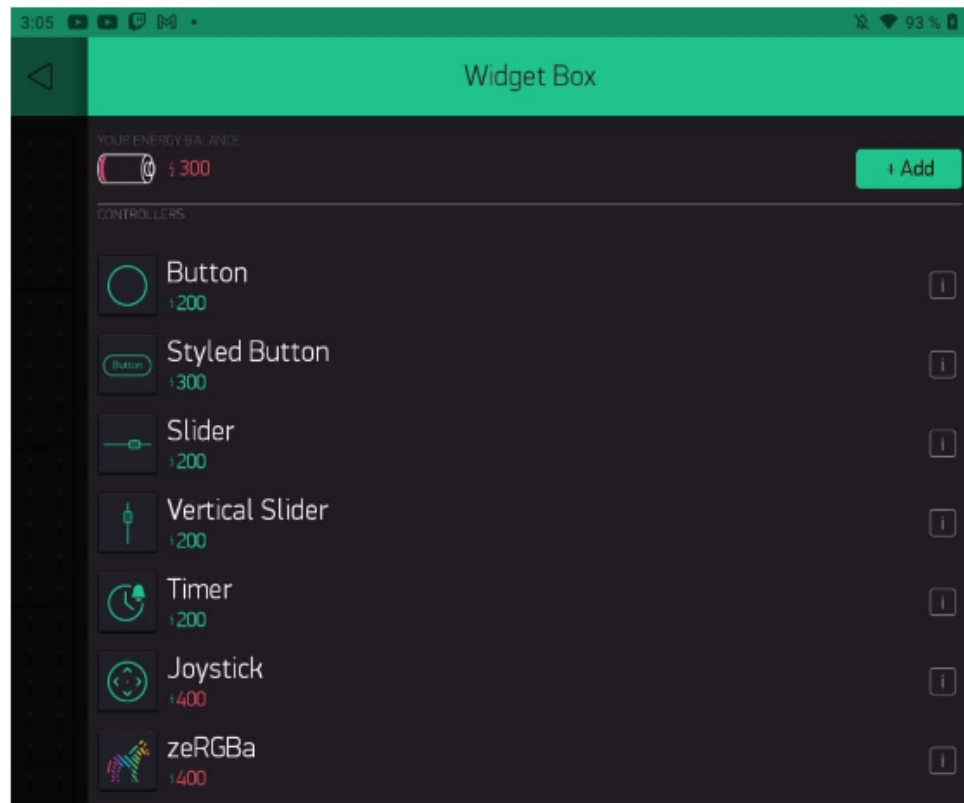


Fig. 19. Tablero de widgets y energía de Blynk. Fuente: [45]

Cabe destacar que la empresa creadora de Blynk (Blynk Inc.) diseñó una revisión de la plataforma ocupada en este proyecto. Esta aplicación recibe el nombre de Blynk IoT, que según su creador es una revisión con mejoras en la edición de plantillas y widgets. Sin mencionar correcciones de errores y optimizaciones. Esta revisión de Blynk puede ser revisada y monitoreada desde un dispositivo móvil o de un computador de acuerdo a las necesidades del usuario de la plataforma.

c) **Firebase.**

Esta es una plataforma digital diseñada y alojada por Google, con la función principal de facilitar el desarrollo de aplicaciones web. Una de las virtudes con mayor relevancia al momento de trabajarla es su base de datos en tiempo real, teniendo varias opciones más que puede usar el usuario (ver Fig. 20).

Es una aplicación bastante útil que en la práctica es cómoda de usar, aunque no muy intuitiva. Requiere comprender bien la página y sus aplicaciones, pero luego de superar esta etapa las acciones se vuelven repetitivas al crear nuevos proyectos.

Firebase se separa en 3 principales grupos de trabajo: desarrollo, crecimiento y la monetización en conjunto con el análisis [46].

Las ventajas de la aplicación son:

- Un buen recurso para enviar y revisar datos en tiempo real.
- El envío de notificaciones o información a la plataforma es sencillo y su gestión no requiere un trabajo arduo.
- Se ofrece un sistema seguro, gracias a los certificados SSL, ayudando a que no se vuelvan públicos los datos de carácter privado.
- Aunque el servicio puede ser de pago, las opciones de la versión gratuita son suficientes para adaptarse a cualquier tipo de proyecto.

La gran desventaja que posee esta aplicación es la necesidad de la versión pagada, debido a sus limitaciones de almacenamiento. La versión pagada es la mejor opción para trabajos más profesionales y que necesiten una mayor transferencia de datos o almacenamiento.



Fig. 20. Interfaz de Firebase, Fuente [47]

3.4 Periférico: Display OLED

Como único tipo de periférico utilizado para este proyecto se dispone de un display OLED, dispositivo presentado en la Fig. 21. Cumple la función de notificar información relevante al usuario: aforo y/o temperatura. Al igual que los dos sensores térmicos, este periférico funciona por medio del protocolo I2C. La resolución de la pantalla es de 128 x 64 píxeles. Las ventajas que posee este dispositivo son tres:

1. El tamaño de la letra es ajustable.
2. Se pueden tener varias líneas de texto o numéricas con diversos tamaños en la misma pantalla a la vez.
3. Se puede ajustar la posición del texto en cualquier posición dentro de la pantalla.

Este último punto es más de ensayo y error que de conocer el dispositivo debido a que dependiendo de las letras, números y sus tamaños, utilizará cierta cantidad de píxeles.

La principal desventaja del dispositivo es que es demasiado pequeño para algunos proyectos específicos donde se necesita presentar una mayor cantidad de información en pantalla.

Para este caso en particular el tamaño y la utilidad del display es más que suficiente. Entrega la información necesaria acorde a lo que se necesita usando 2 pantallas, una para la temperatura y otra para el aforo.

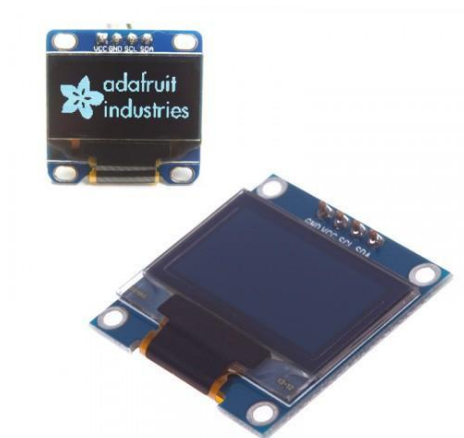


Fig 21. Display Oled 128x64 I2C. Fuente: [48]

3.5 Calibración y caracterización de sensores térmicos.

Para el desarrollo de la calibración se dispone de un termómetro infrarrojo (IRT) que se utilizará como base para las medidas de referencia, el cual es mostrado en la Fig. 22. Este termómetro está diseñado para monitorear la temperatura de forma fácil, rápida y sin contacto. Posee dos modos posibles para realizar la medición de temperatura: la corporal o del ambiente.



Fig. 22. Termómetro infrarrojo Hynaut AXD-518. Fuente: [49]

El termómetro cuenta con una precisión de $0,5^{\circ}\text{C}$ con unos rangos de operación entre 32 y $42,9^{\circ}\text{C}$. Cuando se obtiene una temperatura inferior a 32°C en pantalla del dispositivo aparecerán las siglas Lo (low), caso contrario cuando se obtenga una medición superior a $42,9^{\circ}\text{C}$ aparecerá en pantalla las siglas Hi (high). Para que la medición sea óptima, el termómetro debe encontrarse a una distancia entre 5 a 15 cm de lo que se desea medir, proporcionando una resolución de $0,1^{\circ}\text{C}$.

Para caracterizar el sensor térmico MLX90614 se compara un total de 10 mediciones con la referencia obtenida del termómetro infrarrojo, obteniendo la temperatura de la palma de la mano. Este proceso se realiza a distintas distancias un total de 6 veces, siendo estas distancias 5 , 10 , 15 , 20 , 25 y 30 cm. Para mantener una consistencia en las pruebas, la toma de datos se realizó cuando la temperatura de la palma de la mano estaba a $36,5^{\circ}\text{C}$. La distancia sólo es relevante para tomar los datos del MLX90614, debido a que el termómetro tiene una toma de temperatura óptima entre el rango de 5 y 15 cm.

En la Fig. 23 se presentan los errores de temperatura del MLX90614 asociados a las 6 distancias antes mencionadas.

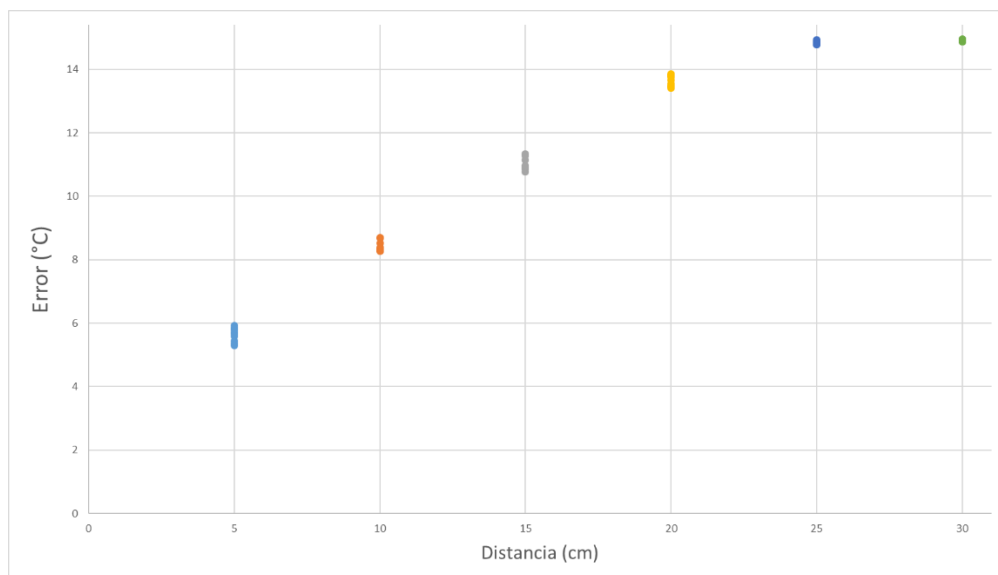


Fig. 23. Representación gráfica de la variación del error de la temperatura del MLX90614.

Se debe tener en consideración que existen diferentes factores que alteran las mediciones, entre estos se encuentran los dispositivos electrónicos cerca del sensor y la relevancia del ángulo de visión del sensor al aumentar la distancia entre la lente y la palma de la mano.

Al observar el comportamiento de los errores presentes en la Fig. 23, se puede observar que la dispersión de los valores para una misma distancia no es considerable, por lo tanto, se puede calcular la media aritmética de todos los errores para obtener un valor promedio en cada distancia, que se observan en la Fig. 24.

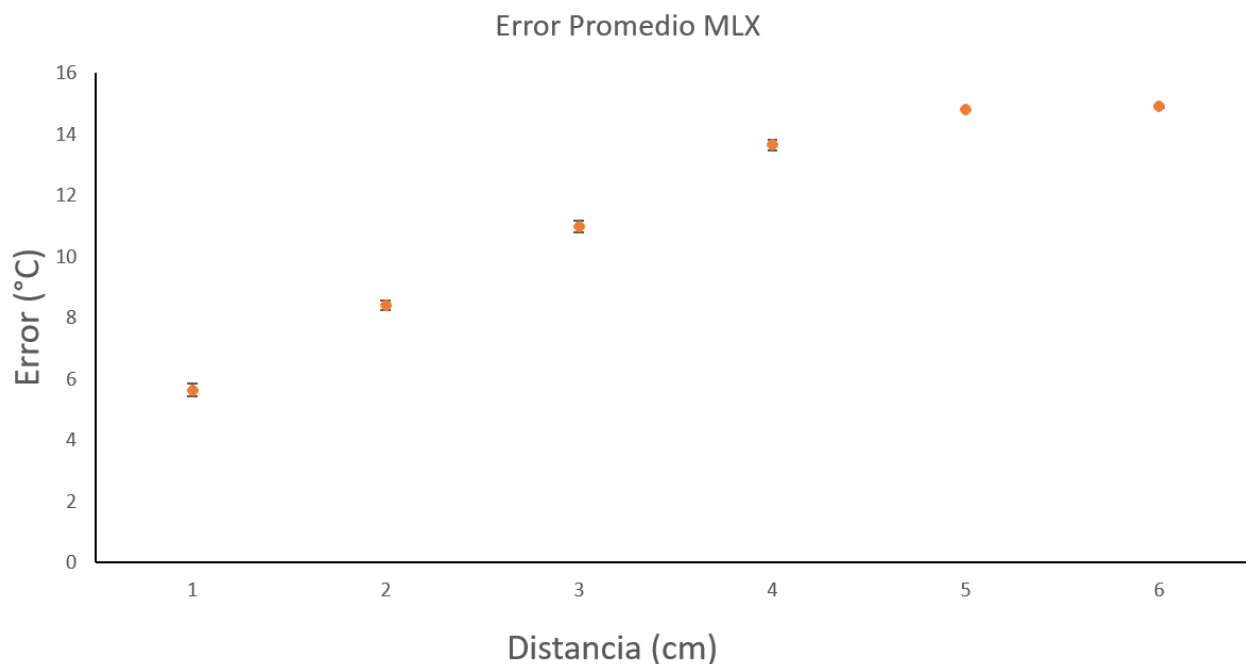


Fig. 24. Representación gráfica del error medio por cada distancia de la temperatura del MLX90614.

De la Fig. 23 y Fig. 24 se interpreta que la dispersión de los datos es mínima, la mayor variación de datos se encuentra a los 5 cm, con una desviación de 0,22 aproximadamente y a medida que la distancia aumenta la variación de los datos disminuye. Estos datos se presentan a continuación en la Tabla 6.

TABLA 6. Desviación estándar MLX90614 a distintas distancias.

Distancia (cm)	Desviación estándar
5	0,2205
10	0,1648
15	0,1875
20	0,1657
25	0,0506
30	0,0333

Para la Calibración de la AMG 8833 se utiliza el mismo principio que al caracterizar el MLX90614, la única variación será la cantidad de mediciones realizadas a ciertas distancias reduciéndolas a 6 mediciones. Las mediciones se seguirán realizando a la palma de la mano a las 6 distancias antes presentadas y de igual forma se mantiene la consistencia en las pruebas al solo realizar la toma de datos cuando la temperatura obtenida del termómetro infrarrojo sea igual a 36,5°C.

A diferencia del MLX90614 la AMG 8833 es un arreglo de 64 píxeles, razón por la que se deben de calibrar las 64 lecturas. Por conveniencia se define a una posición fija la palma de la mano en relación al lente del sensor (centrada). Para realizar la calibración se tomará en consideración la recomendación del fabricante del arreglo del sensor infrarrojo [50], quien indica que se debe considerar una temperatura como referencia que deben de poseer todos los pixeles (para este caso de estudio 36,5°C), luego de hacer la medición se debe sumar o restar un offset a cada píxel en relación a la diferencia entre el valor de referencia y el valor obtenido. Este proceso se repetirá para cada una de las 6 distancias. En la Fig. 25 se presenta el cómo se realiza la calibración según fabricante.

p1	p2	p3	p4	p5	p6	p7	p8		e1	e2	e3	e4	e5	e6	e7	e8
p9	p10	p11	p12	p13	p14	p15	p16		e9	e10	e11	e12	e13	e14	e15	e16
p17	p18	p19	p20	p21	p22	p23	p24		e17	e18	e19	e20	e21	e22	e23	e24
p25	p26	p27	p28	p29	p30	p31	p32		e25	e26	e27	e28	e29	e30	e31	e32
p33	p34	p35	p36	p37	p38	p39	p40	+	e33	e34	e35	e36	e37	e38	e39	e40
p41	p42	p43	p44	p45	p46	p47	p48		e41	e42	e43	e44	e45	e46	e47	e48
p49	p50	p51	p52	p53	p54	p55	p56		e49	e50	e51	e52	e53	e54	e55	e56
p57	p58	p59	p60	p61	p62	p63	p64		e57	e58	e59	e60	e61	e62	e63	e64

Matriz de temperaturas
AMG 8833

Matriz de errores

Fig. 25. Método de calibración recomendado por el fabricante al sumar la matriz de píxeles y la matriz de errores.

Para verificar la calibración del módulo AMG 8833, se presenta la matriz de errores u offset a los 5 cm en la Fig. 26 y en la Tabla 7 se presenta la comparación entre el valor de referencia (36,5°C) y la temperatura obtenida luego de la calibración al realizar 10 pruebas a los 5 cm.

5,44	5,11	5,03	4,98	4,94	5,28	5,32	5,53
5,11	5,23	5,28	5,61	5,28	5,48	5,61	5,36
5,28	5,36	5,19	5,53	5,61	5,57	5,23	5,32
5,19	5,53	5,61	5,57	5,23	5,32	5,19	5,15
5,15	5,32	5,65	5,82	5,11	5,23	5,32	5,61
5,53	5,61	5,86	5,11	4,98	5,03	5,28	5,82
6,03	5,86	6,11	5,15	5,11	5,48	5,48	5,57
5,90	5,36	5,69	5,36	5,40	5,78	5,94	6,03

Matriz de errores

Fig. 26. Matriz de errores a los 5cm de distancia del sensor AMG 8833.

TABLA 7. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 5cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
36,95	36,5	0,45
37,1	36,5	0,6
37,19	36,5	0,69
36,9	36,5	0,4
36,87	36,5	0,37
36,82	36,5	0,32
36,66	36,5	0,16
36,81	36,5	0,31
36,78	36,5	0,28
36,83	36,5	0,33

Las tablas de verificación del resto de distancias se encuentran en el apartado **Anexo A**.

3.6 Sistema de trazabilidad.

Una vez presentadas las partes que componen la cadena de valor IoT y mencionados todos los dispositivos y aplicaciones que se utilizarán para el proyecto presentados en la Fig. 27, se describe a continuación el funcionamiento del mismo y de cada subsistema.

El sistema de trazabilidad consiste de tres subsistemas que funcionan de forma secuencial. Se centrará en el sistema basado en el uso del sensor de ultrasonido, debido a su comodidad y no ser tan restrictivo en su operación. Este sistema funciona en óptimas condiciones si el lugar físico donde se implementará cuenta con sólo una puerta que funciona como entrada y salida, con un ancho amplio. Los 3 subsistemas que componen el proyecto son:

- a) **Subsistema de identificación:** Para este subsistema se hará uso de la tecnología RFID por medio de una tarjeta entregada a cada posible usuario. El sistema identificará si el usuario está autorizado para ingresar al establecimiento o no. Esta información de acceso se encuentra almacenada dentro de la base de datos de Firebase junto a otros datos de interés de cada persona, como el nombre completo, domicilio, RUT, teléfono, contactos estrechos (familia por lo general).
- b) **Subsistema de Aforo:** Para este sistema se utilizará el sensor de ultrasonido para identificar las entradas y salidas de las personas. Esta información será presentada por medio de la pantalla Oled y la aplicación Blynk y los valores registrados se almacenarán en las páginas de Firebase y Adafruit.IO.
- c) **Subsistema de Temperatura:** En este caso se usarán ambos sensores de temperatura, el MLX90614 y la cámara térmica AMG 8833 para obtener la temperatura de la persona. Una vez obtenida la temperatura se considera la distancia entre el sensor y la persona por medio de un segundo sensor de ultrasonido para aplicar la calibración respecto a la temperatura promedio de 36,5 °C. Luego este valor se visualiza en la pantalla Oled y en la aplicación Blynk, y se registra en la Firebase y en Adafruit.IO.

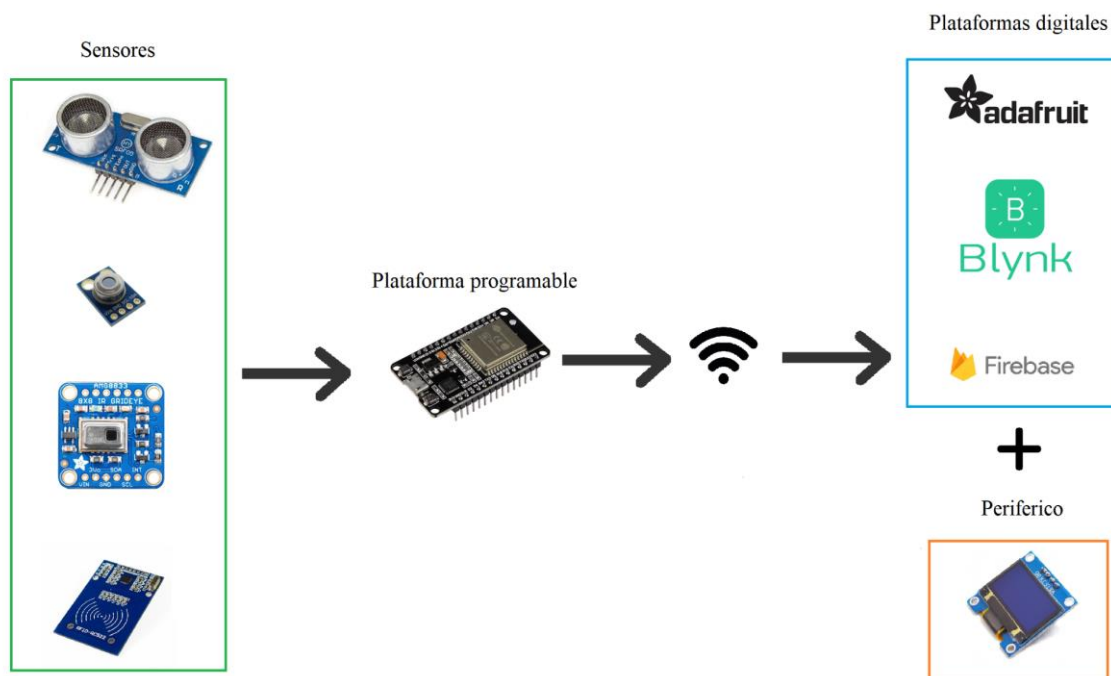


Fig 27. Cadena de valor IoT del Proyecto.

Cada uno de los subsistemas se complementa con los otros, pasando primero por la identificación, el conteo de aforo y la medición de temperatura. La idea de este planteamiento es poder identificar primero a la persona y si está autorizada, luego verificar si es posible su ingreso al lugar debido a la capacidad máxima y disponible y por último cotejar la temperatura del usuario.

3.6.1 Subsistema de identificación.

El subsistema de identificación presentado en la Fig. 28 cumple la función de identificar a las personas a la entrada del establecimiento. Por medio del sistema de RFID y una base de datos alojada en el servidor de Firebase, se crean previamente cierta cantidad de carpetas de acuerdo a la cantidad de usuarios. Estas carpetas están definidas con el prefijo “ID” acompañado de un número que contiene la información personal de cada usuario. Así se agiliza la forma de obtener la información sin necesidad de que se revise cada una de las carpetas. Se dispone de un lector RFID que se encuentra a la entrada para identificar al usuario y no existe un segundo lector a la salida.

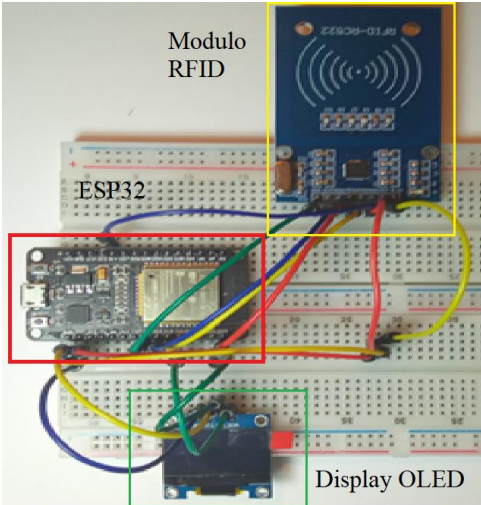


Fig. 28. Montaje subsistema de identificación.

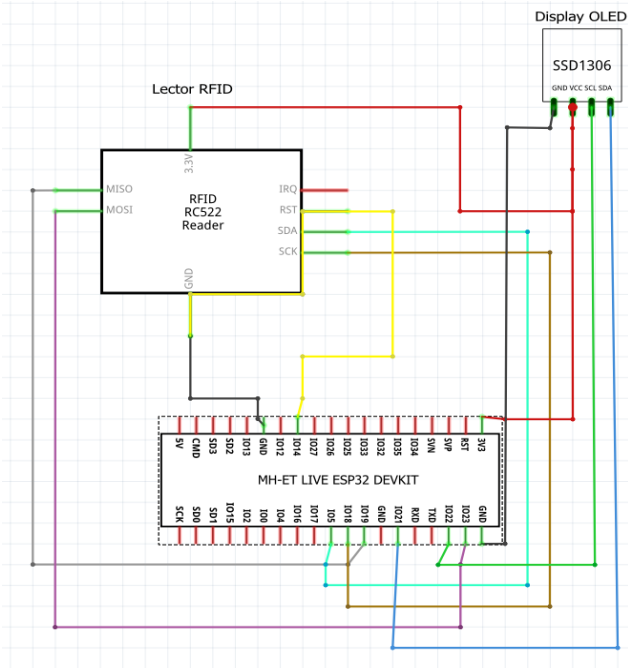


Fig. 29. Esquemático sistema de identificación.

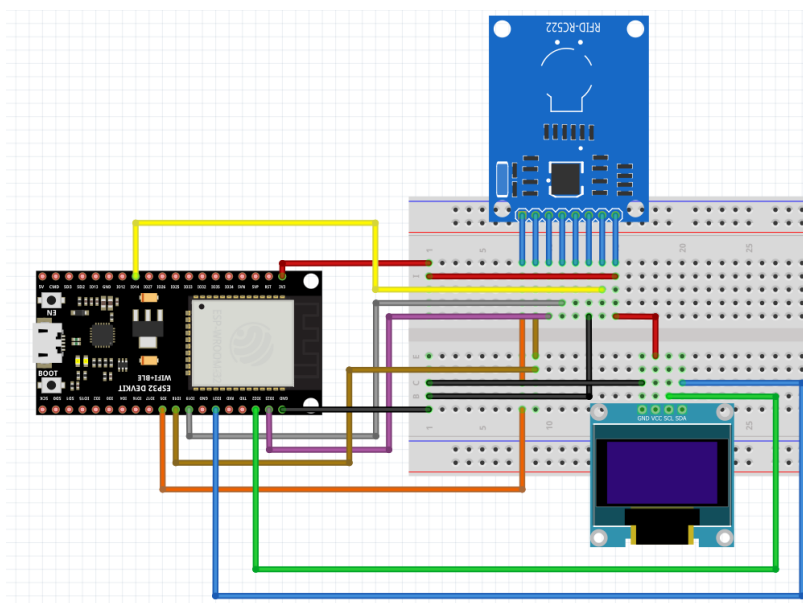


Fig. 30. Subsistema de identificación desarrollado en Fritzing.

En la Fig. 30 se expone el subsistema de identificación por medio de la aplicación de Fritzing y sus respectivas conexiones y en la Fig. 29 se observa el esquemático del mismo subsistema obtenido por medio de Fritzing.

En primer lugar, la persona debe de estar autorizada para ingresar al edificio, cuya autorización quedará registrada en la base de datos de Google, por medio del uso de la ID de la tarjeta RFID que puede tomar valores decimales o hexadecimales. Para este caso sólo se aplicará la lectura y escritura de las variables decimales vinculadas a cada tarjeta. Al pasar la persona su tarjeta RFID, el subsistema de identificación confirma si se encuentra autorizado. Consultando la base de datos se verifican 2 variables: el número de usuario dentro de las carpetas de las identificaciones de usuarios (IDN°) y la serie de número decimales de la tarjeta RFID. En caso de que no detecte la ID del usuario, inmediatamente se notificará que el usuario no tiene acceso. En caso contrario si la persona se encuentra autorizada, el sistema mandará un mensaje por el display notificando que dicha persona está autorizada junto con el nombre de la persona.

En la base de datos de Firebase se almacena la información para el monitoreo, que puede variar dependiendo de los requerimientos de la institución. Pero bajo criterio de diseño se han contemplado diversas variables de cada usuario, como las variables personales, variables de interés de estudio y las variables de identificación.

Las variables de interés de estudio corresponden a aquellas que aportan a la detección de casos directamente, como la temperatura de la persona, el lugar físico donde se solicitó ingreso y el número de lectura de la tarjeta RFID. Esta última variable existe para identificar cuántas personas previas han usado su tarjeta RFID para ingresar al lugar.

Además, se agregan 4 variables extras a la base de datos que son: ingresos totales, salidas totales, lecturas totales y aforo actual. Los ingresos y salidas totales son los registros del sensor de ultrasonido, las lecturas totales son la cantidad de lecturas que se han realizado de distintas tarjetas RFID a lo largo del día y por último la variable de aforo actual representa la ocupación en tiempo real del edificio. Esta variable existe específicamente como un respaldo en caso de un apagón centralizado o general donde una vez se reinicie el sistema consultará para verificar el último registro de aforo.

Entre las variables personales se seleccionó el nombre de la persona, su número de cédula de identidad, número de teléfono fijo y/o teléfono celular y la dirección donde reside. De esta forma es fácil localizar a la persona en caso de alguna detección de casos positivos. A este conjunto de datos personales se agregan los contactos estrechos de la persona. En situación normal corresponden al grupo familiar de la persona, quienes deben ser notificados por el mismo usuario. Opcionalmente se pueden agregar los datos como número de teléfono celular de los contactos, pero eso queda a criterio de la institución solicitarlos o no.

3.6.2 Subsistema de aforo.

El sistema de aforo funciona como un contador de entradas y de salidas de personas por medio del sensor de ultrasonido. Como la distancia máxima es de 4 metros, se puede adecuar hasta estas condiciones. Según datos antropométricos, se puede determinar que el ancho del cuerpo humano tiende a ser cercano a dos veces el ancho de la cabeza. Esto sería un caso genérico que excluye a gente con hombros grandes o con dimensiones irregulares del cuerpo. Tomando estas dimensiones corporales se decidió una media de 70 cm de ancho entre cada posible medición que corresponden a: entre 0 a 70 cm se consideran que la persona viene ingresando al recinto, caso contrario cuando la distancia está entre los 70 y 140 cm, el sistema considera que la persona se retiró del establecimiento.

Para las pruebas se consideró que el sensor se encuentra ubicado a la derecha del ingreso del lugar, según lo indica la Fig. 31. También se consideró la posibilidad de ubicar alguna separación física entre las dos regiones. Esta separación existiría para evitar que alguna persona que desee ingresar pase por la zona de salida y viceversa.

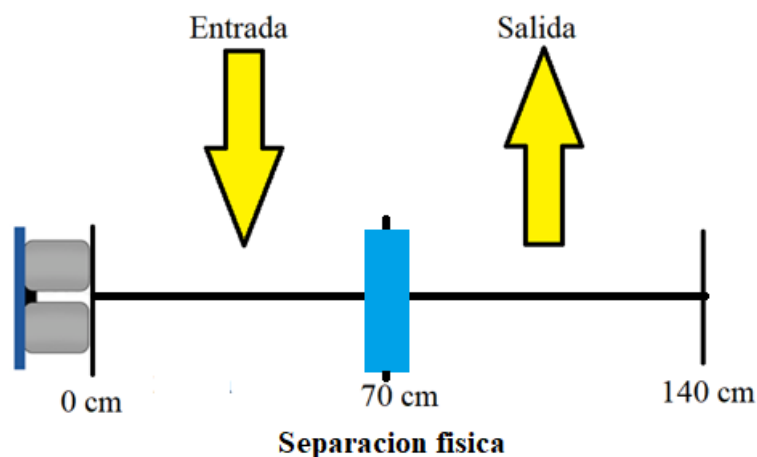


Fig. 31. Funcionamiento subsistema de aforo.

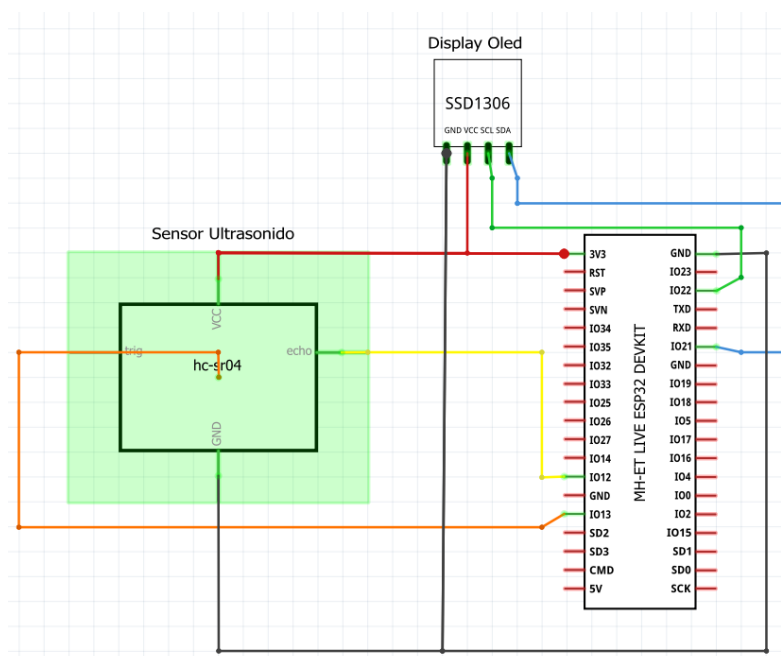


Fig. 32. Esquemático subsistema de aforo.

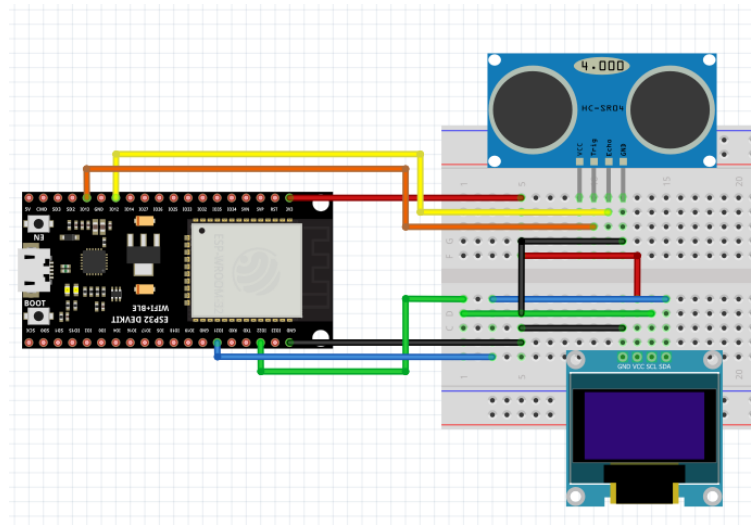


Fig. 33. Subsistema de aforo desarrollado en Fritzing.

En la Fig. 33 se presenta el subsistema de aforo elaborado en Fritzing, donde se muestra que éste opera con el sensor de ultrasonido antes mencionado y el display OLED para exponer datos relevantes asociados al aforo (cantidad máxima permitida y disponibilidad del lugar) y en la Fig. 32 se presenta el esquemático del sistema antes mencionado.

El sistema posee dos estados: cuando el aforo es menor a la cantidad máxima permitida, y cuando el aforo supera este valor máximo. El valor de aforo máximo dependerá del edificio, de los metros cuadrados del sector y de la cantidad de personas que pueda albergar. En el primer estado, se acumulan las entradas (positivamente) y salidas (negativamente) a las personas del aforo total. Esta información se envía por Wifi a la aplicación de Blynk que contiene un registro del aforo total, la cantidad de personas presentes y la cantidad de personas restantes antes de que se llegue a la capacidad máxima. También envía esta misma información a la bitácora virtual Adafruit.IO, la que funciona con un historial donde se guardará la hora de activación del sensor y el tipo de detección. Por último, esta información se registra en la base de datos de Firebase, teniendo dos valores de registro: número de veces en que se detectó una entrada y número de veces en que se detectó una salida, de tal forma que se tiene una bitácora total de entradas y salidas en caso de algún problema de medición externo al código. En el segundo estado el sistema mandará un mensaje al display, anunciando que se alcanzó la capacidad máxima del edificio, el cual se repetirá constantemente hasta que alguien abandone el lugar.

3.6.3 Subsistema de temperatura.

El sistema de temperatura se compone de tres sensores: dos térmicos y uno de ultrasonido. Este constantemente realiza las mediciones de temperatura y distancia al momento de detectar una entrada al lugar. Los pasos que sigue este sistema son los siguientes:

- 1) Primero se realiza una medición con el MLX90614 para obtener la temperatura de referencia.
- 2) Luego se realiza una segunda medición con el AMG 8833 y se obtiene la matriz con 64 valores de temperatura.
- 3) Según el valor de temperatura de referencia obtenido del otro sensor, se filtran las temperaturas, excluyendo así los valores del AMG 8833 que sean inferiores a dicho umbral.
- 4) Después con el sensor de ultrasonido se obtiene la distancia a la que se encuentra la mano de la persona.
- 5) Dependiendo de la distancia obtenida se sumará un offset a las mediciones del AMG 8833 que sean iguales o superiores al valor de referencia.
- 6) Luego se realiza media aritmética a las nuevas temperaturas.
- 7) Finalmente se compara este promedio con límites establecidos (ver Tabla 8) y se define si la persona tiene una temperatura baja, normal, alta, muy alta o temperatura fuera de rango.

TABLA 8. Límites de temperatura.

Temperatura	Rangos de temperatura
<i>Fuera de Rango</i>	$42,9^{\circ}\text{C} < X < 32,0^{\circ}\text{C}$
<i>Baja</i>	$32,0^{\circ}\text{C} < X < 35,5^{\circ}\text{C}$
<i>Normal</i>	$35,5^{\circ}\text{C} \leq X \leq 37,5^{\circ}\text{C}$
<i>Alta</i>	$37,5^{\circ}\text{C} < X \leq 39,0^{\circ}\text{C}$
<i>Muy Alta</i>	$39,0^{\circ}\text{C} < X \leq 42,9^{\circ}\text{C}$

Los parámetros presentes en la Tabla 8 fueron definidos en base a revisión bibliográfica [51] y a los rangos de detección del termómetro utilizado para la realización de pruebas.

Al igual que en el caso anterior, toda esta información se envía a las aplicaciones Blynk y Adafruit.IO. Se usa Blynk como interfaz física para visualizarla desde cualquier dispositivo móvil que tenga la aplicación y la cuenta vinculada al sistema. A la interfaz de Blynk se le agregan dos leds virtuales que se iluminan dependiendo del rango de temperatura de la persona. Si la persona tiene una temperatura normal, se iluminará el led verde. En caso contrario se iluminará el led rojo, dando aviso a quien monitoree la interfaz.

En la Fig. 34 se presenta como funciona el sistema de temperatura con los 3 sensores antes mencionados, donde el MLX genera la primera medición, la cámara térmica la segunda medición y luego se verifica la distancia de la persona para realizar la calibración correspondiente, teniendo como máximo los 30 cm propuestos. En la Fig. 30 se observa el montaje de ambos subsistemas, el de aforo y el de temperatura, donde se identifican los distintos sensores.

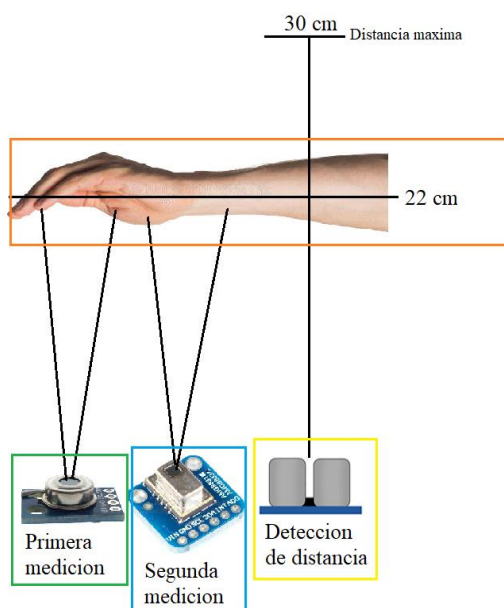


Fig. 34. Funcionamiento subsistema de temperatura.

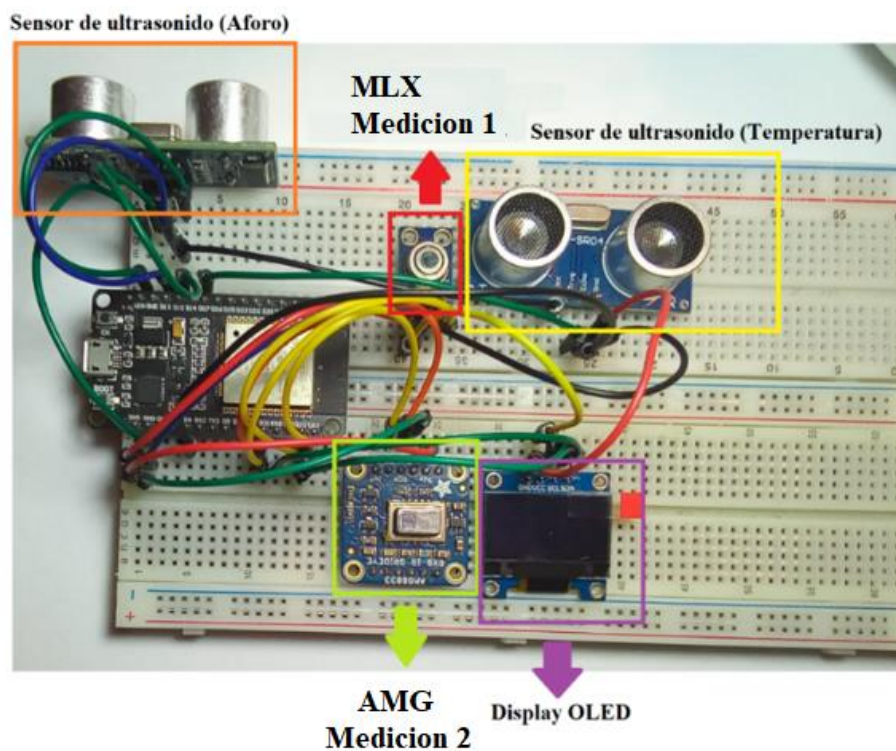


Fig. 35. Montaje subsistemas de Aforo/Temperatura.

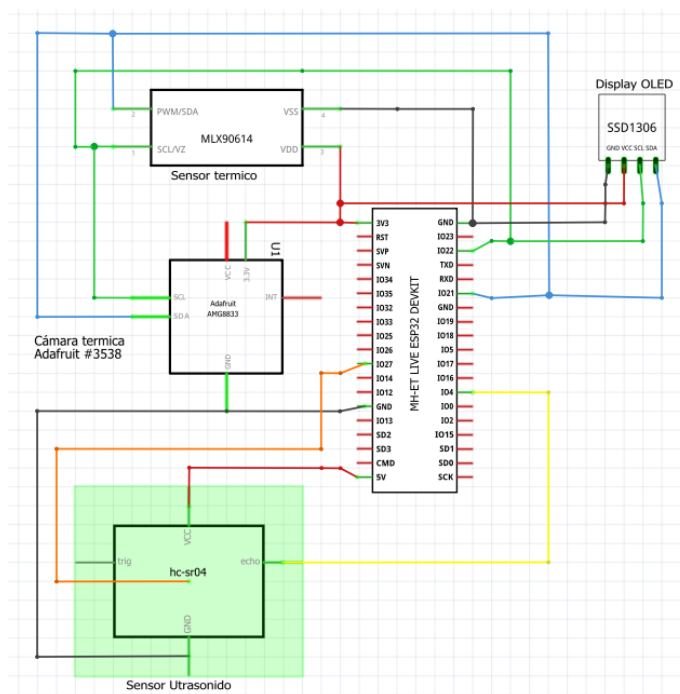


Fig. 36. Esquemático subsistema de temperatura.

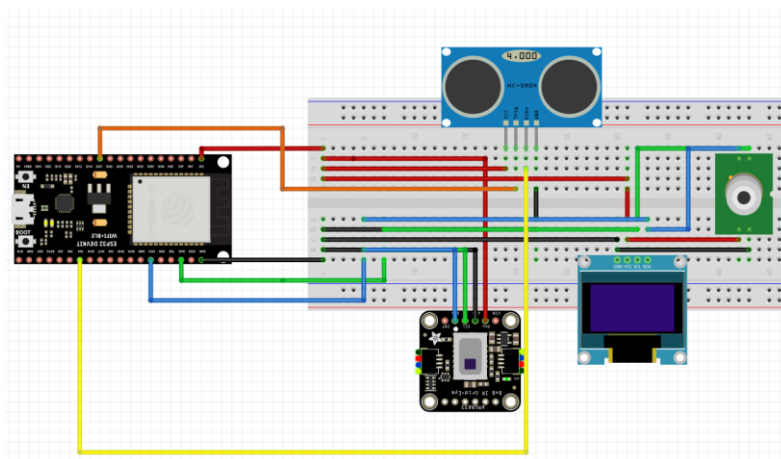


Fig. 37. Subsistema de temperatura desarrollado en Fritzing.

En la Fig. 37 se observa el subsistema de temperatura elaborado en Fritzing, donde se muestra que éste opera con el segundo sensor de ultrasonido, el sensor térmico MLX90614, la cámara térmica AMG 8833 y el display OLED para presentar datos relevantes asociados a la temperatura (Temperatura ambiente, temperatura de la persona y notificar que se está fuera de rango en caso de tener una temperatura superior a $42,9^{\circ}\text{C}$ o inferior a $32,0^{\circ}\text{C}$). En la Fig. 36 se presenta el esquemático del sistema antes mencionado.

En la Fig. 38 se muestra un esquema del sistema de trazabilidad desarrollado y en la Fig. 39 el diagrama de flujo que describe el procedimiento que lleva a cabo el sistema al momento de ingresar una persona, pasando primero por la identificación, luego el conteo de aforo y por último la medición de temperatura.

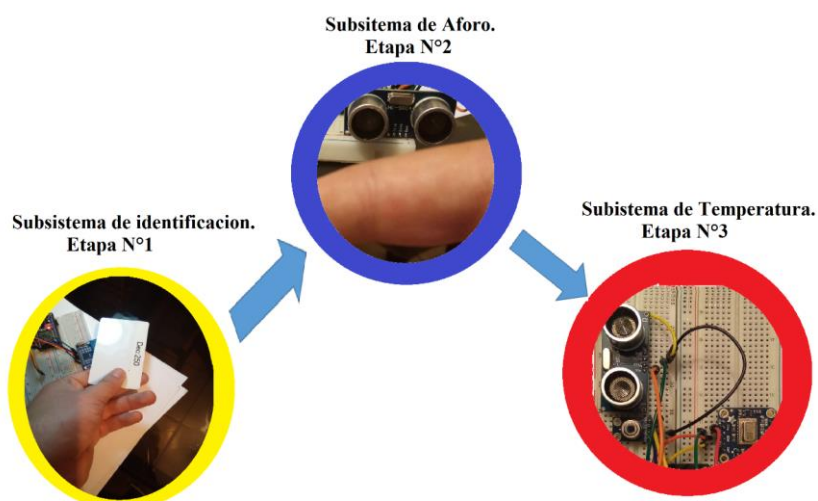


Fig. 38. Etapas del proyecto.

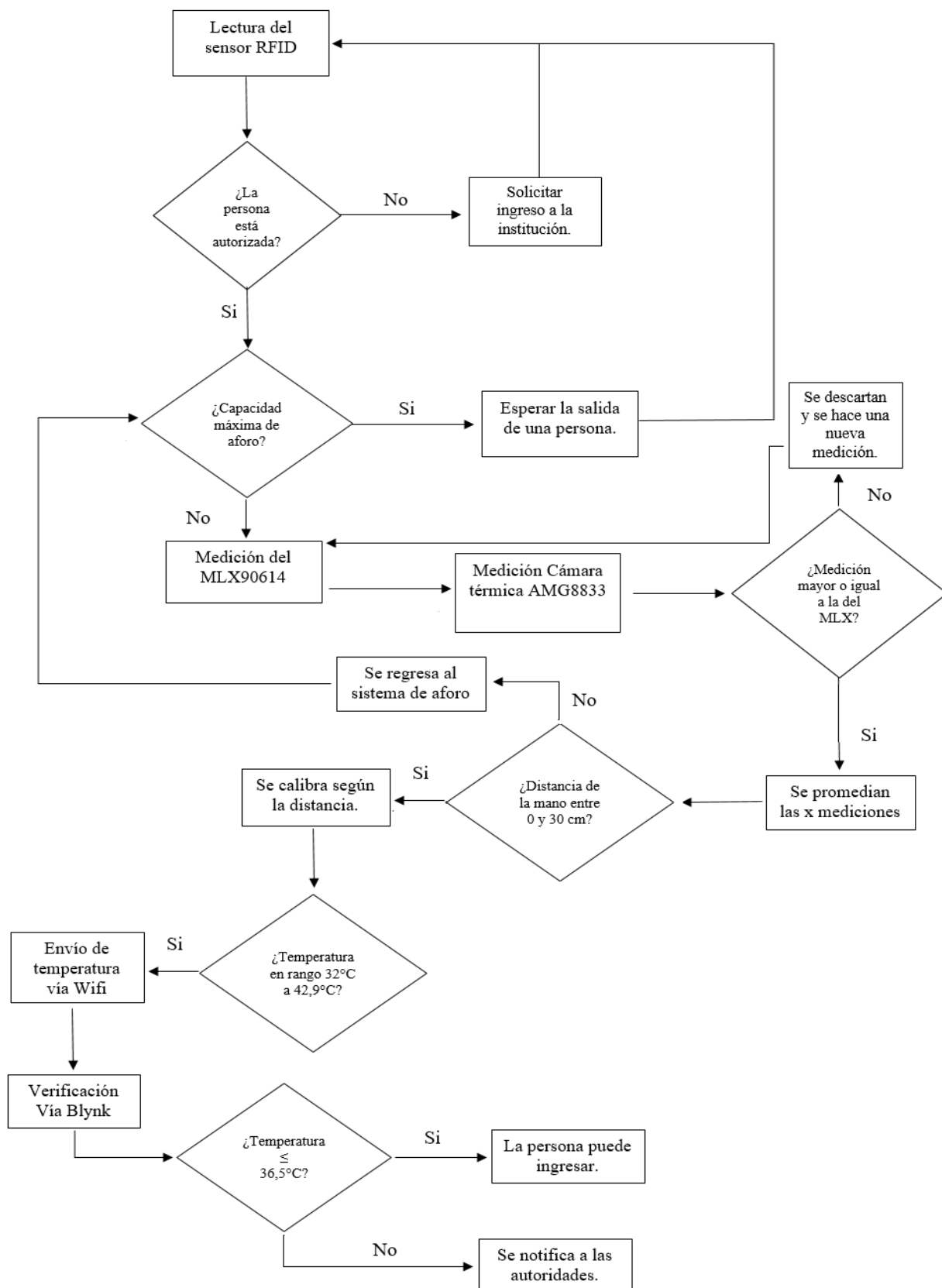


Fig. 39 Diagrama de flujo.

Capítulo 4: Resultados

En este capítulo se analizarán los resultados obtenidos del proyecto, los que están relacionados con la integración de las distintas plataformas digitales con aplicación en IoT y del monitor serial de la plataforma arduino, donde se visualiza la conexión a la red Wi-fi, las mediciones y el envío de datos. Los resultados se separan en distintos subcapítulos separados en la visualización del monitor serial, Blynk, Adafruit, Firebase y del display Oled.

4.1 Resultados monitor serie IDE arduino.

El sistema tiene una secuencia lógica de funcionamiento, siendo primero la identificación de acceso del usuario, el aforo y la temperatura de la persona. De la misma forma se presentarán los resultados obtenidos por medio del monitor serie, herramienta de arduino para visualizar la escritura de código y su funcionamiento.

En la Fig 40 se presenta la lectura de una tarjeta RFID autorizada, donde primero se identifica la secuencia decimal de la tarjeta. Luego para poder identificar de que usuario es la tarjeta vinculada se compara el primer valor decimal de la tarjeta RFID. Luego se verifica a que usuario corresponde dicha tarjeta, en dicho ejemplo corresponde al usuario vinculado a la ID1 con los números de identificación (128, 88, 75, 89, 202). A continuación, se comparan los cinco valores decimales de la tarjeta con los de la base de datos y si no coinciden entonces se notifica que la persona tiene el acceso “Denegado”. Para este caso, los 5 valores decimales de la tarjeta ID1 fueron iguales a los de la base de datos, por lo que se notifica el nombre del usuario y su estado de acceso que sería “Autorizado”.

```

Te has conectado a la red WiFi.....
Te has conectado con la IP: 192.168.43.175

Firebase Client v3.11.12

Token info: type = id token, status = on request
Token info: type = id token, status = ready

Tarjeta leida
IDE de la tarjeta:
Dec: 128, 88, 75, 89, 202

```

Identificación RFID de números decimales.

```

El primer valor decimal es:
128

```

Se considera el primer número decimal de la tarjeta.

```

La ID del usuario es la N°1
La posición es de la variable es:1

```

Se identifica el número asociado a la identificación, en este caso sería el usuario de la ID1

```

128
88
75
89
202

```

Se revisan los valores decimales almacenados en la base de datos de Firebase asociado al usuario N°1

```

Nombre: Juan Hernandez
Acceso: Autorizado

```

Nombre del usuario y condición de acceso.

Fig. 40. Detección de tarjeta asociada a la ID1 dentro de la Firebase.

En la Fig. 41 se presenta como opera el subsistema de temperatura una vez el subsistema de aforo identifica una entrada. Primero notifica la ocupación en el edificio y posteriormente se espera un tiempo antes de imprimir los valores de temperatura de sensor térmico MLX. Luego se imprimen en pantalla las 64 mediciones provenientes de la cámara térmica y se verifica la distancia que esté dentro de los rangos de 0 a 30 cm. Finalmente, se presenta en pantalla la medición de temperatura.

```

La ocupacion del edificio es de: 4
Ambient MLX = 19.23 *C
Object MLX = 27.99 *C

```

Temperatura Sensor MLX

```

Matriz de datos N°1
[30.75, 31.00, 30.75, 31.00, 31.00, 30.25, 30.50, 31.25,
30.50, 30.25, 30.75, 30.25, 30.25, 30.25, 30.50, 30.25,
29.75, 30.25, 30.50, 30.00, 30.25, 30.00, 30.00, 30.75,
30.75, 30.25, 30.50, 30.50, 30.25, 30.25, 30.00, 30.00,
30.50, 30.00, 30.25, 30.25, 30.25, 30.25, 30.00, 29.75, 30.00,
30.25, 30.25, 30.50, 30.25, 29.50, 29.75, 30.25, 29.75,
29.75, 30.00, 29.75, 29.75, 30.00, 30.00, 29.75, 29.50,
29.75, 29.75, 29.75, 30.00, 29.00, 29.25, 29.50, 29.75,

```

Matriz Cámara térmica AMG 8833

```

La distancia de deteccion fue de: 5cm
La medicion de temperatura es: 35.62 *C
Su temperatura esta dentro de lo normal

```

Medición final luego de la calibración.

Fig. 41. Respuesta en pantalla y operación del subsistema de temperatura al identificar una entrada y la temperatura está dentro del rango normal.

En la Fig. 42 se presenta como opera el subsistema de temperatura cuando la temperatura de la persona está dentro del rango de temperatura baja.

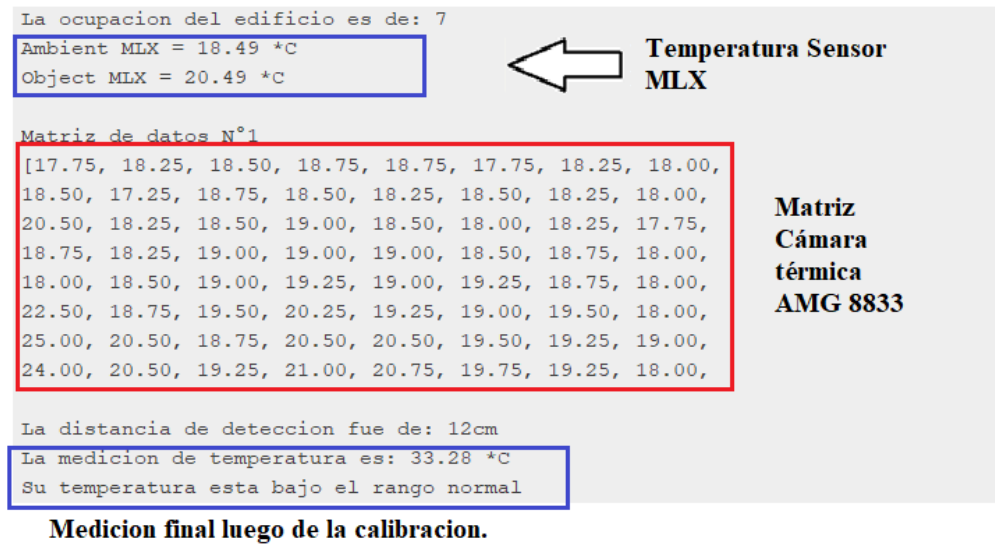


Fig. 42. Respuesta en pantalla y operación del subsistema de temperatura cuando la temperatura está dentro del rango bajo.

En la Fig. 43 se presenta como opera el subsistema de temperatura cuando la temperatura de la persona está dentro del rango de temperatura alta, se notifica que tiene fiebre y que se les notificara a las autoridades su condición.

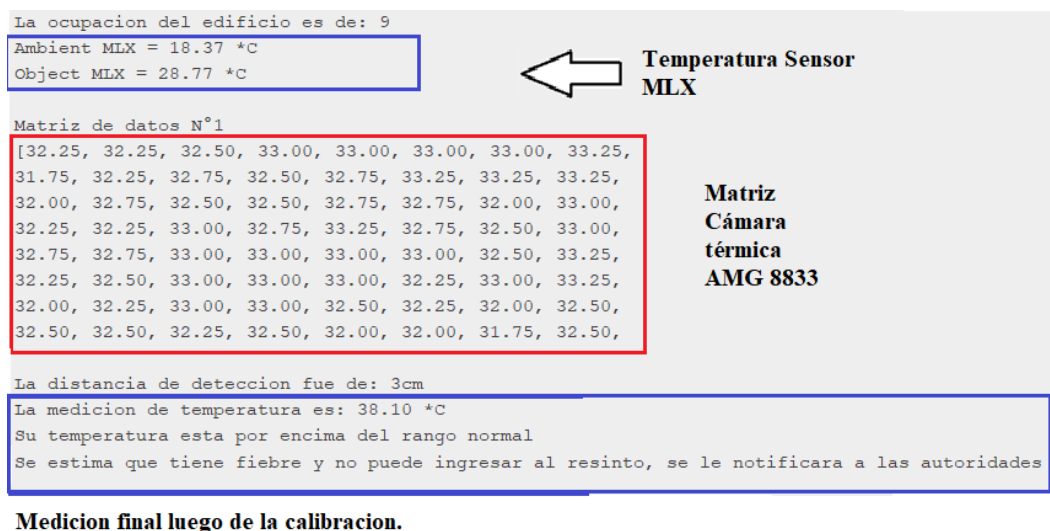
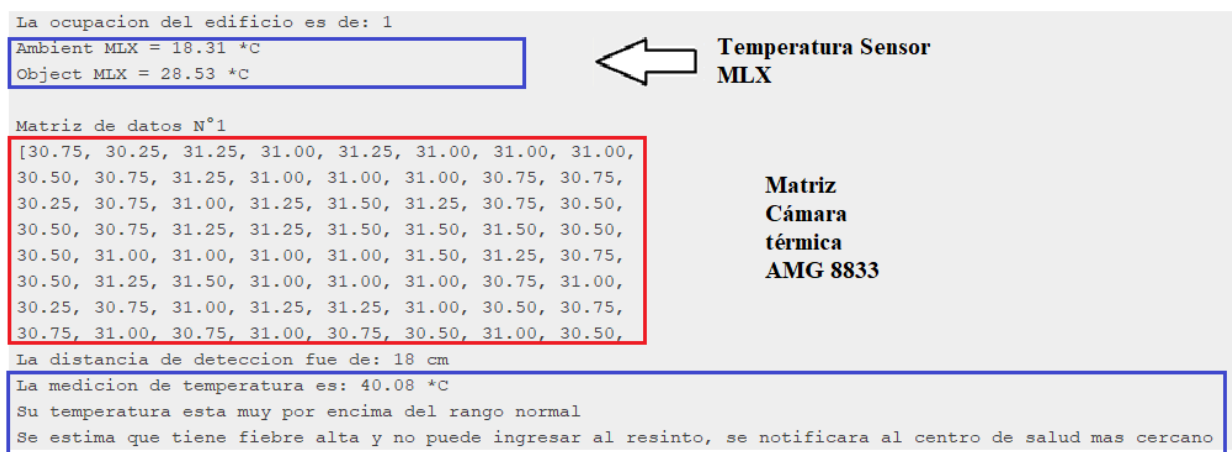


Fig. 43. Respuesta en pantalla y operación del subsistema de temperatura cuando la temperatura está dentro del rango alto.

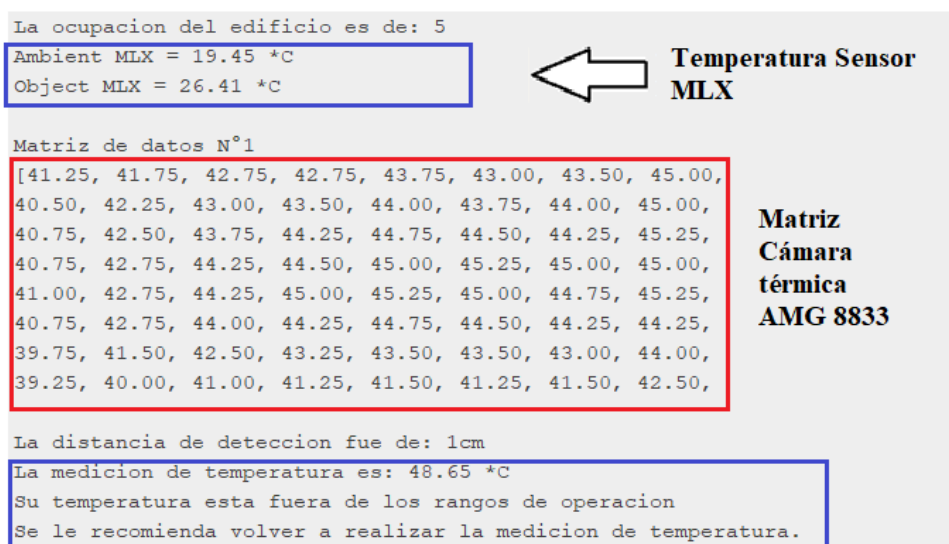
En la Fig. 44 se presenta como opera el subsistema de temperatura cuando la temperatura de la persona está dentro del rango de temperatura muy alta, se notifica que tiene fiebre alta y que se le notificara al centro de salud más cercana por la situación médica.



Medicion final luego de la calibracion.

Fig. 44. Respuesta en pantalla y operación del subsistema de temperatura cuando la temperatura está dentro del rango muy alto.

En la Fig. 45 se presenta como opera el subsistema de temperatura cuando la temperatura de la persona está fuera de los rangos de operación, teniendo una temperatura inferior a 32,0°C o superior a 42,9°C.



Medicion final luego de la calibracion.

Fig. 45. Respuesta en pantalla y operación del subsistema de temperatura cuando la temperatura está fuera de los rangos de operación.

En el caso de la Fig. 46, la distancia entre la persona y el sensor de ultrasonido del subsistema de temperatura es mayor a los 30 cm definidos como el valor máximo para las mediciones. Al ocurrir esto se notifica que la persona no se encuentra dentro de las distancias recomendadas para la medición y en este caso no se realiza la calibración en la medición final y sólo se imprime el promedio de los valores de la cámara térmica, notificando que debe volver a notificar su entrada y medirse la temperatura.

```

La ocupacion del edificio es de: 7
Ambient MLX = 23.09 *C
Object MLX = 24.15 *C

[21.25, 21.25, 21.50, 21.50, 22.25, 21.75, 21.75, 22.00,
21.25, 21.25, 22.25, 23.25, 25.50, 24.75, 22.25, 22.00,
21.50, 22.25, 23.50, 27.50, 28.00, 26.75, 22.75, 22.75,
22.50, 23.00, 26.75, 28.25, 28.75, 26.00, 23.00, 22.25,
23.00, 25.75, 27.50, 27.50, 27.25, 24.50, 22.75, 22.50,
26.25, 26.50, 27.50, 25.00, 23.25, 22.50, 23.00, 21.50,
26.00, 25.75, 23.50, 22.75, 22.00, 22.00, 21.25, 21.75,
26.50, 26.75, 22.50, 22.25, 21.50, 21.50, 21.00, 21.50,
La distancia del objeto a la camara termica es: 114cms
no se detecta objeto entre las distancias recomendadas, volver a intentar
La medicion de temperatura es de : 26.35C°
La ocupacion del edificio es de: 7

```

Fig. 46. Caso de distancia superior al rango máximo de 30cm. Fuente: Elaboración propia

4.2 Resultados Dispositivo móvil con Blynk.

En las Fig. 47 y Fig. 48, se presenta la dashboard de Blynk usada como bitácora móvil del proyecto, en la que se visualiza lo siguiente: la distancia de detección del sensor de ultrasonido asociado al subsistema de temperatura, la capacidad máxima de aforo, el aforo restante y el aforo actual en el lugar, las temperaturas ambiente y del objeto/persona obtenidas del sensor térmico MLX90614, la temperatura captada por la cámara térmica luego de la calibración dependiendo de la distancia de detección, un indicador circular con un mínimo y un máximo relacionado a la temperatura de la cámara térmica y por último, dos leds virtuales de color verde y rojo, que indican si se respetan los márgenes descritos en el código o no, respectivamente.



Fig. 47. Bitácora móvil usada para monitoreo para un caso menor a 37.5°C. Fuente: [45]

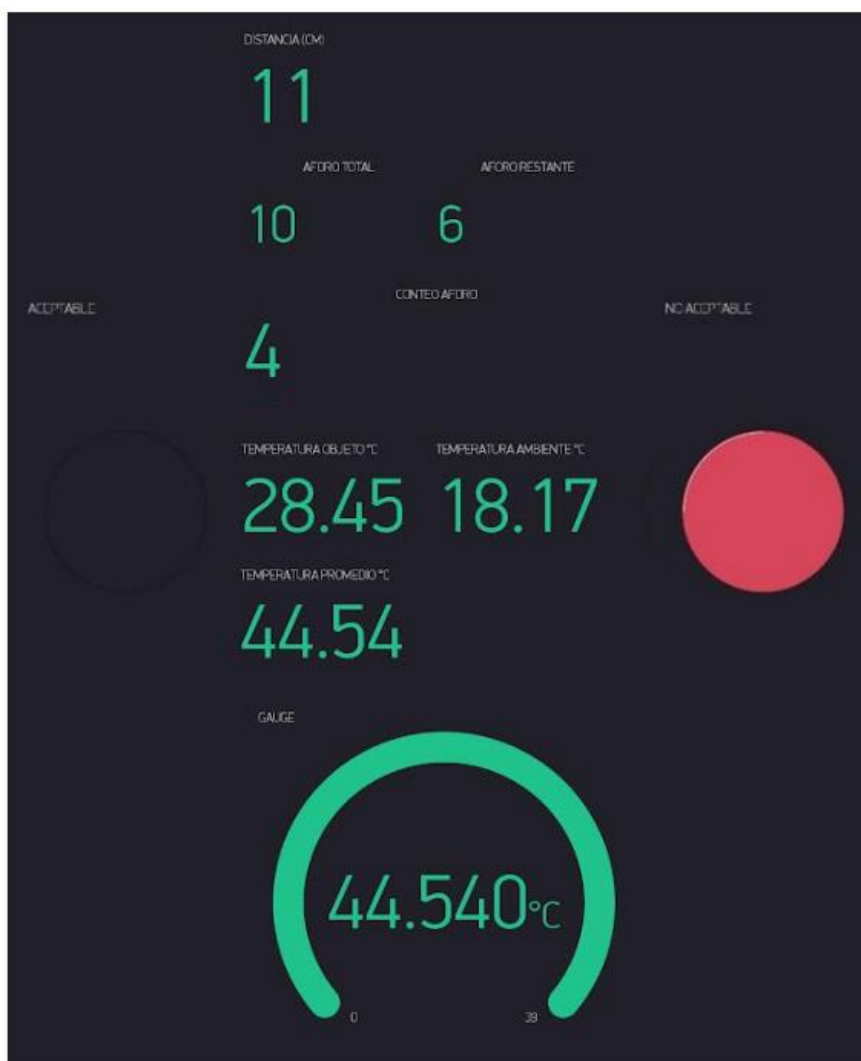


Fig. 48. Bitácora móvil usada para monitoreo para un caso mayor a 37.5°C. Fuente: [45]

4.3 Resultados del historial virtual Adafruit.IO

En la Fig. 49 se aprecian los tres bloques que se emplearon para este proyecto, antes presentados en la Fig. 22, siendo los más útiles y aplicables a distintos proyectos. En el caso de la temperatura, se utiliza una gráfica para facilitar su visualización y un cuadro de texto con indicaciones horarias con cada uno de los valores de temperatura medidos. Se usa un indicador circular para medir el porcentaje de ocupación del lugar, notificando una advertencia cuando se llegue al 90% de ocupación. En caso de ser necesario, se puede incluir una mayor cantidad de bloques para una recopilación de datos más precisa.

Las temperaturas presentadas en la plataforma de Adafruit.IO son distintas a las presentadas en la bitácora virtual de Blynk debido a que fueron evaluadas en distintos momentos.

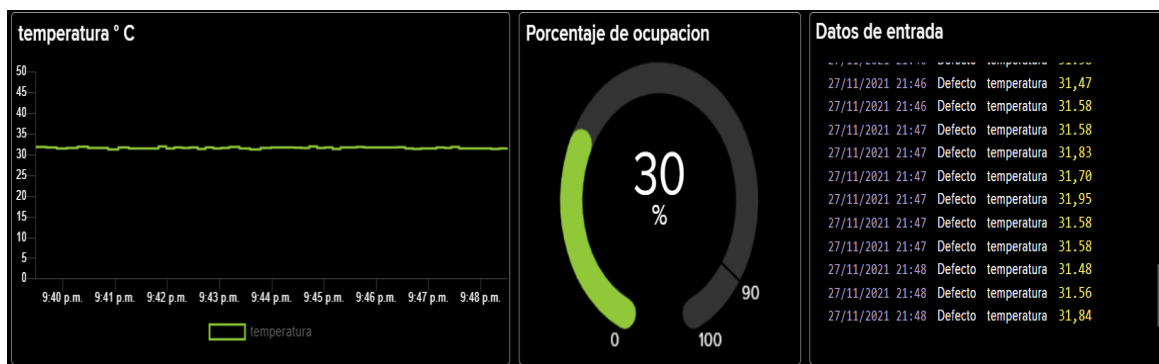


Fig 49. Dashboard Adafruit.IO. Fuente: [52]

Además, Adafruit. IO permite registrar el historial de la temperatura como se muestra en la Fig. 50, la cual muestra los datos obtenidos durante 3 horas de mediciones cuyo tiempo entre mediciones es de 3 segundos. La gráfica de Adafruit.IO respeta valores mínimos y máximos representados al momento de editar los bloques. Por temas de visualización, se estimó dejar como valor máximo 50°C debido a que es poco probable que se registre una temperatura mayor a menos que se produzca algún tipo de problema mientras se realiza la medición.

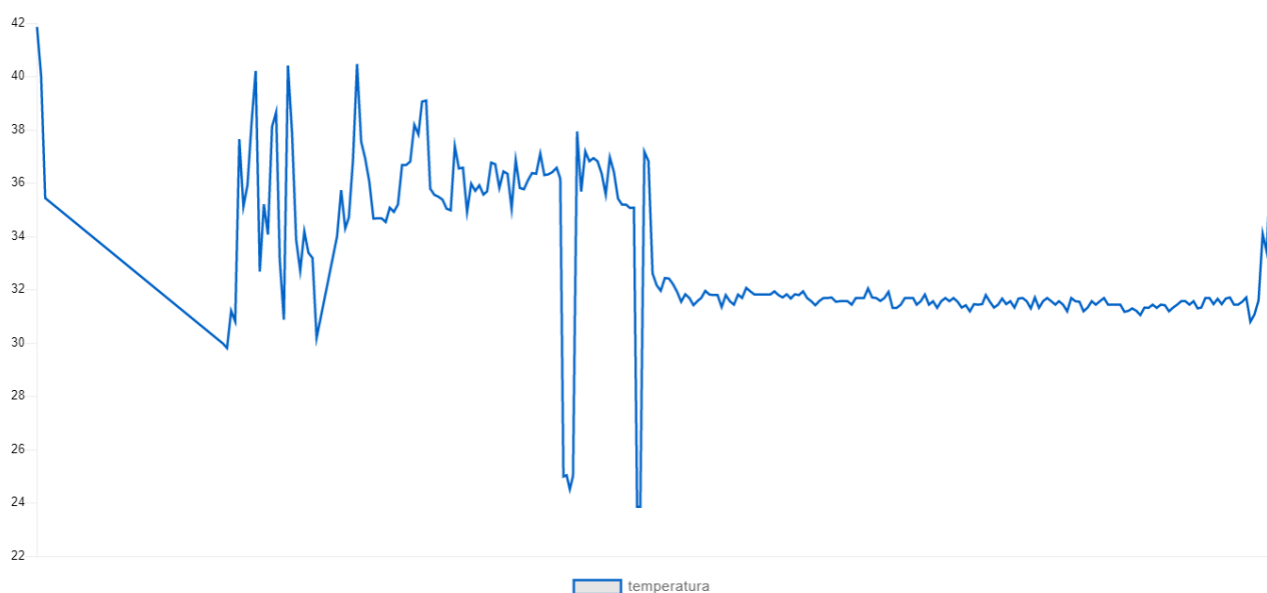


Fig. 50. Registro histórico de temperatura de Adafruit.IO. Fuente: [52]

4.4 Resultados Firebase

Firebase presenta muchas formas de ingreso para poder controlar las bases de datos por medio de plataformas de control. En este caso se usa la identificación por correo para representar a un usuario, y se crea el proyecto ESP32 (ver Fig. 51), en el cual se almacena la base de datos de los usuarios.

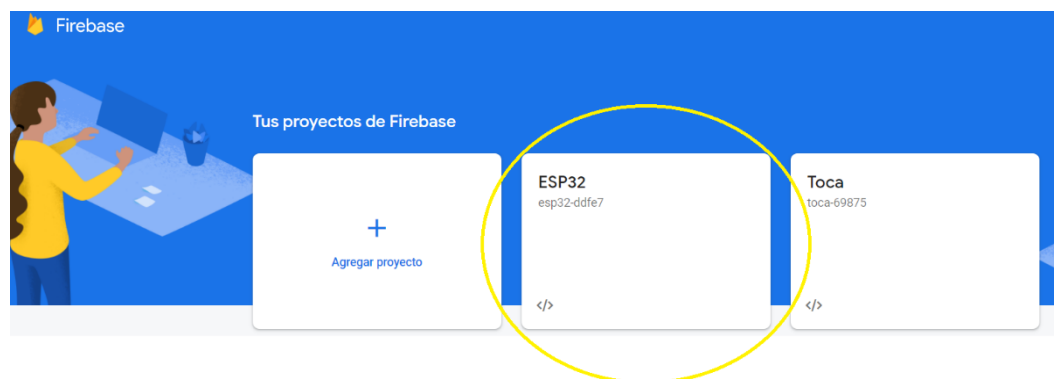


Fig. 51 Consola de proyectos personal. Fuente: [47]

Firebase ofrece varias opciones para manipular los datos, tales como se presentan en la Fig 52. Se eligió la opción de la base de datos en tiempo real debido a que entre sus ventajas se permite al código leer y reescribir información en tiempo real, ayudando a que el código sea autónomo al momento de solicitar información de diversas carpetas o reescribir información de interés. Dentro de las capacidades de la base de datos en tiempo real de Firebase se encuentra su cantidad máxima de carpetas anidadas, pudiendo ser hasta un total de 32. Aunque esta cantidad no es necesaria para este proyecto, se menciona para explicar que se puede tener un nivel de detalle más amplio a la vez que se agregan carpetas específicas.

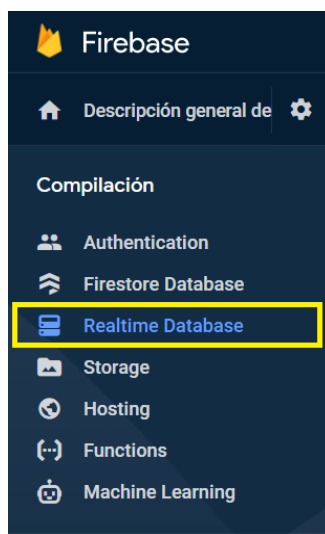


Fig. 52. Opciones para manipular datos dentro de la Firebase. Fuente: [47]

En la base de datos se registran 2 carpetas iniciales, una para guardar los datos de los usuarios y otra para guardar los registros generales con las entradas totales, salidas totales, aforo actual y lecturas totales en el recinto, como se muestra en la Fig. 53.

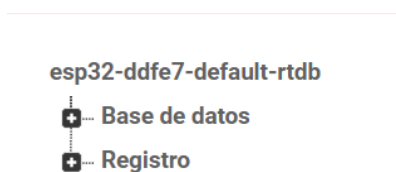


Fig. 53. Base de datos en tiempo real del proyecto ESP32. Fuente: [47]

En la Fig. 54 se muestran las variables almacenadas en la capeta “Registro” y las 6 identificaciones de usuarios definidas para el proyecto alojadas dentro de la capeta “Base de datos”. Cada una de estas sub-carpetas IDN° contienen la información asociada a las tarjetas RFID. En la Fig 55 se exponen las variables de interés que previamente fueron definidas, entre las que resaltan los 4 datos personales del usuario (Dirección, Nombre, Rut y Teléfono de contacto). También se hace mención a otros datos relevantes para la institución como el lugar al que desea ingresar, su temperatura, el número de lectura de su tarjeta RFID. En este caso al ser la N°9, significa que antes de identificarse para ingresar, otras 8 personas ya habían ingresado al lugar y su estado de acceso (autorizado y denegado). Por último, están los datos propios de la Tarjeta RFID, los cuales serían los números decimales propios de la tarjeta RFID en cuestión.

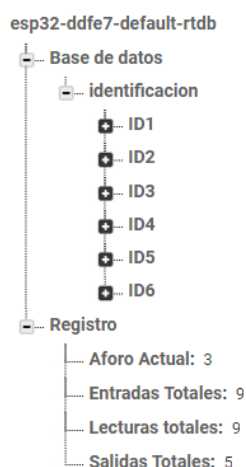


Fig. 54. Base de datos en tiempo real del proyecto ESP32, carpetas de registro y base de datos. Fuente: [47]

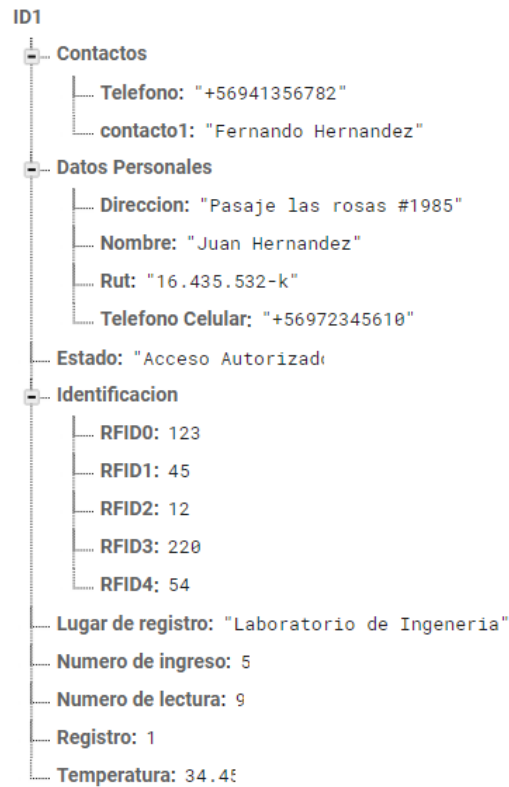


Fig. 55. Base de datos en tiempo real de la ID1 completa. Fuente: [47]

4.5 Resultados Display Oled.

A continuación, se presentan los resultados del display Oled asociados al subsistema de identificación (Fig. 56(a)). El sistema identifica el número de ID de una persona, cuando es reconocido (Fig. 56(b)) y se visualiza el nombre y su estado de acceso (Fig. 56(c)). En caso de no identificar el número de la ID, se notifica que el usuario tendría el acceso denegado al establecimiento (Fig. 56(d)).

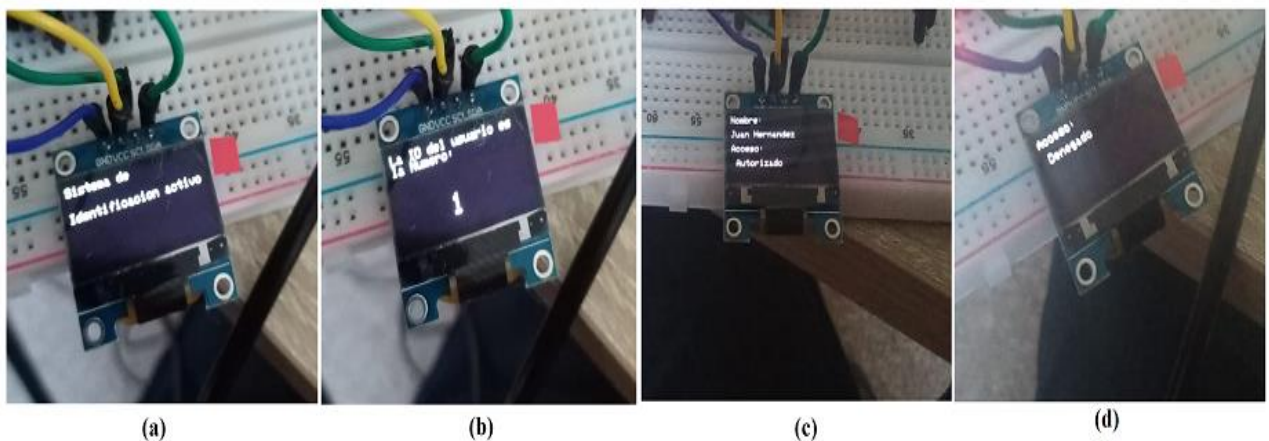


Fig. 56. Display Oled subsistema de identificación, y sus 4 casos.

Por otra parte, se presentan los resultados del display Oled asociados a los subsistemas de temperatura y aforo con un mayor énfasis en la temperatura (Fig. 57(a)), el registro de temperatura y del aforo en el lugar (Fig. 57(b)). En caso que la temperatura esté fuera de los rangos de operación, ya sea inferior a 32°C (Fig. 57(c)) o superior a 42,9°C (Fig. 57(d)), se notifica por medio del display el mensaje por un tiempo de 10 segundos. Luego de este tiempo volverá a la presentación del sistema operativo.

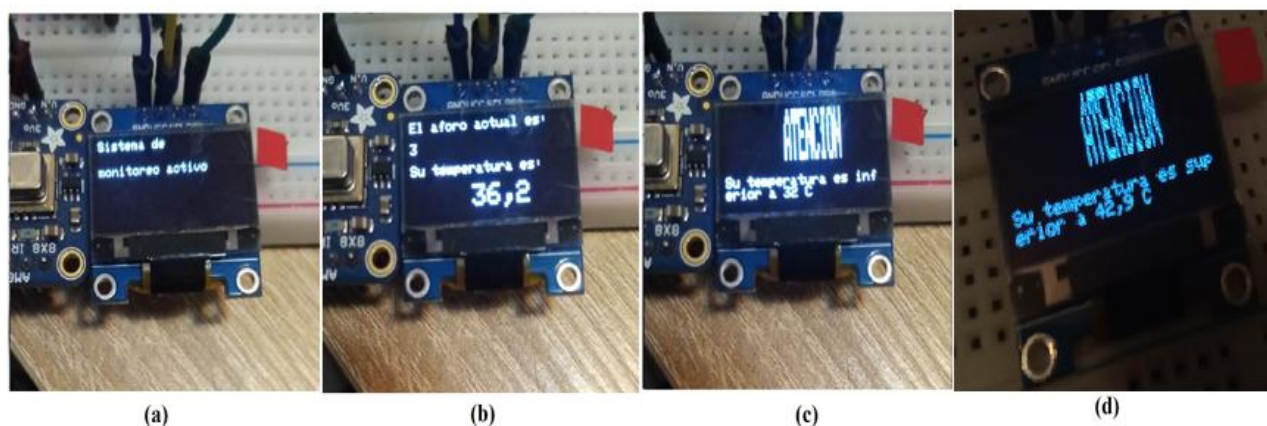


Fig. 57. Display Oled subsistema de monitoreo, y sus casos asociados a la temperatura.

Finalmente se muestran los mensajes entregados por el display Oled asociados a los subsistemas de temperatura y aforo con mayor énfasis en el aforo. Cuando el aforo es igual a 0 y está en espera de detectar una entrada (Fig. 58(a)), los parámetros de “aforo actual”, “aforo total” y “aforo restante” (Fig. 58(b)) y cuando el aforo ha llegado a la máxima capacidad de ocupación del lugar, por lo que debe de esperar a que alguien se retire (Fig. 58(c)).

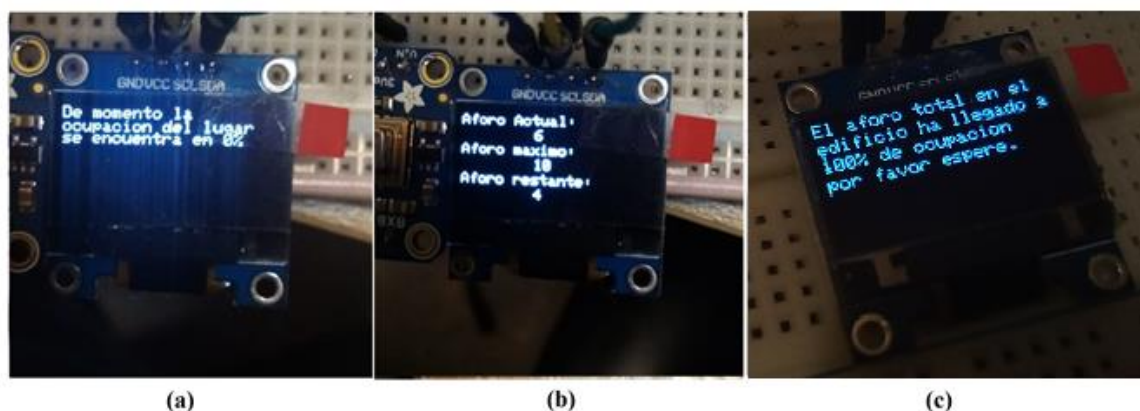


Fig. 58. Display Oled subsistema de monitoreo, y sus casos asociado al aforo.

CAPITULO 5: Conclusiones.

5.1 Conclusiones.

Según la revisión bibliográfica relacionada con el impacto de la trazabilidad en el mundo afectado por la pandemia del COVID-19, se observa la importancia y valoración que tiene la implementación de estos sistemas para aportar al control de esta enfermedad y con ello reducir significativamente la expansión del virus. En países como China y Corea del Sur, donde se implementaron estos modelos, pudieron reconocer que era un aporte a las estrategias que implementaban las autoridades de estas naciones, destacando la rapidez y eficiencia en la obtención de datos que permitían tomar decisiones que apuntaban a minimizar y controlar la propagación del virus por sus territorios. Caso contrario al expuesto de los países asiáticos, un número importante de naciones de la Unión Europea (UE), latinoamericanos y norteamericanos, que no incorporaron en sus prioridades estos sistemas, al menos en las primeras etapas de propagación del virus, no pudieron mostrar estadísticas sanitarias favorables en el combate de este mal y que al no tomar medidas inmediatas para implementar diversos sistemas de trazabilidad, el virus avanzó rápidamente entre la población y no sólo eso sino que facilitó que éste mutara con el tiempo.

Así los hechos, donde se consideró, la integración de los sistemas de trazabilidad se vio enormemente beneficiado con el IoT, ya que pudieron identificar factores de riesgos en personas y por tanto, separarlas o aislarlas de las demás y en definitiva, controlar los contagios en lugares donde la aglomeración de personas era inevitable (espacios de trabajo y locomoción colectiva, entre otras), pudiendo con estas acciones ir retomando la normalidad en actividades estratégicas de los países.

Con el tiempo se fue incorporando este sistema masivamente porque sin duda ha sido un aporte y una contribución a la vuelta de la normalidad. En efecto, el avance en IoT creció enormemente en estos últimos años, aunque este crecimiento fue más notorio en el área de la salud o IoMT. El aumento de nuevos sistemas con base en IoMT avanzó con bastante rapidez en el desarrollo de aplicaciones, sistemas de monitorio, consultas virtuales e inteligencia artificial.

Bajo los objetivos planteados al inicio del proyecto se puede concluir que es posible diseñar un sistema de trazabilidad enfocado en la prevención e identificación de casos de COVID-19 por medio de la implementación del IoT, considerando la identificación de usuarios, el control del aforo en espacios cerrados y el control de casos por medio de la variable de temperatura. Dado que las pruebas realizadas en un ambiente controlado presentaron resultados favorables, es posible de replicar a un escenario real debido a como está diseñado, ajustado y calibrado el sistema.

Al implementar IoT en conjunto a la trazabilidad se logra un sistema capaz de recopilar información de las personas gracias a los diversos sensores utilizados, el recibir esta información en la ESP32 y que por medio del WiFi esta placa se comuniquen con diversas plataformas digitales que cumplen los roles de bitácora, historial y de base de datos, permite que el sistema sea parcialmente autónomo y que la intervención humana sea mínima y sólo en casos específicos. Este sistema se apega al área del IoMT, al tener como principal objetivo el cuidado de la salud de las personas y por tener una mayor paridad con sistemas propuestos por otros países e instituciones en pos de la salud de la ciudadanía.

Para las pruebas del subsistema de temperatura se tomó en consideración los rangos de temperatura corporal de un adulto saludable. Para definir los límites de operación del subsistema se utilizó el rango de medición del termómetro infrarrojo usado para las pruebas de calibración. Los resultados obtenidos muestran como el subsistema es capaz de diferenciar a que distancia la persona posiciona su mano, puede interpretar si la temperatura de la persona es baja, normal, alta, muy alta o la medición esta fuera de rango de acuerdo a los rangos ya establecidos y también puede almacenar estas mediciones en la base de datos de firebase.

Respecto al subsistema de aforo, el monitoreo por medio de distintas plataformas digitales permite tener un registro más detallado del movimiento de individuos, limitado únicamente a una sobrecarga de personas, pudiendo no detectar la cantidad exacta de quienes ingresan o se retiran. De los resultados obtenidos en conjunto al subsistema de temperatura, se obtiene que el subsistema es capaz de identificar correctamente si la persona está ingresando o saliendo de acuerdo a los rangos de distancia que se definieron en el capítulo 3. Al existir un registro en tiempo real en la base de datos de firebase definido como “Aforo actual”, el subsistema puede solicitar esta variable en caso de un apagón y retomar el conteo de aforo una vez se restablezca la energía.

En el subsistema de identificación, la integración de la base de datos en tiempo real de la plataforma Firebase es de gran utilidad para revisar, verificar, extraer y reemplazar información de los usuarios. Resulta relevante la no interacción humana para comprobar la identidad y autorización de la persona, siendo completamente autónomo en la verificación de registros. Los resultados obtenidos muestran que el subsistema distingue a un usuario con ingreso autorizado o no, de acuerdo a la serie decimal de su tarjeta RFID y a la información personal almacenada dentro de la base de datos.

Así las cosas, se puede concluir que este sistema sería útil como ayuda en la detección de posibles casos de COVID-19 en diversos espacios cerrados, ya que permite instalar un sistema de control, mediante un seguimiento de personas recurrentes en un recinto determinado, especialmente en aquellos de alta concurrencia y vitales para el desarrollo y progreso de una nación como lo es un colegio, una universidad, una fábrica, un recinto de salud, una oficina, un departamento, entre muchas instalaciones en que se convoca un importante número de personas. Además, se plantea como una propuesta económica y eficaz para monitorear los posibles contagios dentro de estos establecimientos.

5.2 Trabajo Futuro

En base al sistema desarrollado, a la experiencia adquirida y a los resultados obtenidos se pueden proyectar otras investigaciones y por tanto, se proponen los siguientes trabajos futuros:

- Realizar una base de datos con una mayor capacidad de usuarios, aproximadamente entre unos 1000 a 5000 usuarios en la base de datos en tiempo real de Firebase, dependiendo de las dimensiones del lugar físico donde se instale el sistema.
- Migrar el control y monitoreo de datos a la aplicación Blynk IoT, una aplicación más completa y sofisticada capaz de reemplazar las funciones de la aplicación móvil de Blynk y de Adafruit.IO.
- Optimizar la programación de cada subsistema para reducir los tiempos de respuesta del sistema global.
- Integración de un módulo tipo reloj de tiempo real (RTC) para evitar la intervención humana para realizar cambios en la base de datos, como lo serian el día de ingreso al lugar y que se reestablezca el aforo del lugar cuando cronológicamente sean las 00:00 horas de cada día.
- Realizar un estudio que pueda evaluar el rendimiento del sistema una vez implementado en un lugar físico.

Referencias Bibliográficas

- [1] Guiomar Huguet Pané (2021, septiembre 02) Grandes Pandemias de la historia. https://historia.nationalgeographic.com.es/a/grandes-pandemias-historia_15178
- [2] Dr. Antoni Esteve Fundacion. El secreto de la peste que tumbo al Imperio Romano. https://www.esteve.org/otras-sugerencias/materia-pestes/?doing_wp_cron=1662655376.7431609630584716796875#:~:text=La%20llamada%20Plaga%20de%20Justiniano,devastaci%C3%B3n%20ha%20sido%20un%20misterio.
- [3] Antoni Virgili (2021, marzo 23) La peste negra, la epidemia mas mortífera. https://historia.nationalgeographic.com.es/a/peste-negra-epidemia-mas-mortifera_6280
- [4] Organización Mundial de la Salud. Coronavirus. https://www.who.int/es/health-topics/coronavirus#tab=tab_1
- [5] Joan Miguel Tejedor (2021, junio 13) Impactos sociales y económicos de la pandemia de la COVID-19 en Latinoamérica. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S2248-60462020000200329
- [6] Begoña Fuente Rey (2020, septiembre 10) El impacto de las tecnologías en salud en tiempos de pandemia. <https://www.minsait.com/es/actualidad/insights/el-impacto-de-las-tecnologias-en-salud-en-tiempos-de-pandemia#>
- [7] Gigi Onag (2020, mayo 8) Analysts say COVID-19 pandemic will spur IoT adoption. <https://futureiot.tech/analysts-say-covid-19-pandemic-will-spur-iot-adoption/>
- [8] Felipe Elorrieta y Camilo Becerra Rodríguez (2021, abril 06) Trazabilidad: el factor clave que fue olvidado. <https://www.ciperchile.cl/2021/04/06/trazabilidad-el-factor-clave-que-fue-olvidado/>
- [9] Khan, J.P., Ali, J., Barnihill, A. Cicero, A., Esmode, K., Hood, A., Hutler, B., Regenberg, A., Watson, C. y Watson, M., (2020) Digital Contact Tracing for Pandemic Response. https://www-cdn.law.stanford.edu/wp-content/uploads/2020/05/project_muse_75831-full.pdf
- [10] Cristóbal Ríos, Lucas Cifuentes y Nicolás Sepúlveda (2021, mayo 07) El punto débil del combate al Covid: hay tres sistemas para trazar los casos. <https://www.ciperchile.cl/2021/05/07/el-punto-debil-del-combate-al-covid-hay-tres-sistemas-para-trazar-los-casos/>
- [11] V. Chamola, V. Hassija, V. Gupta and M. Guizani, "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact," in IEEE Access, vol. 8, pp. 90225-90265, 2020, doi: 10.1109/ACCESS.2020.2992341. <https://ieeexplore.ieee.org/document/9086010>

- [12] Pablo Moralea (2020, abril 21) Preparacion de los sistemas sanitarios y aprendizajes para futuras pandemias: el caso de Corea del Sur (MERS). https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio/10221/28578/1/Preparacion_de_los_sistemas_sanitarios_y_los_aprendizajes_para_futuras_pandemias_el_caso_de_Corea_del_Sur_MERS.pdf
- [13] Azana, H.M., Wan, H.H., Shilan, S., Zainab, S.A., Mojtaba, A., Liza, A.L., (2021) IoMT amid COVID-19 pandemic: Application, architecture, technology, and security. www.elsevier.com/locate/jnca
- [14] Daniele Mercatelli, Luca Triboli, Eleonora Fornasari, Forest Ray, Federico M. Giorgi. (2020, noviembre 18) Coronapp: A web application to annotate and monitor SARS-CoV-2 mutations. <https://onlinelibrary.wiley.com/doi/10.1002/jmv.26678>
- [15] Fernando Arancibia-Collao, Ernesto San Martin. (2021, julio) El valor de la privacidad en el combate al COVID-19 en Chile: análisis de las tecnologías de trazabilidad. <https://repositorio.uc.cl/xmlui/bitstream/handle/11534/61663/paper143-25junio.pdf>
- [16] Dyani Lewis (2021, febrero 22) Contact-tracing apps help reduce COVID infections, data suggest. <https://www.nature.com/articles/d41586-021-00451-y>
- [17] Red Hat (2019, enero 08) ¿Que es el Internet de las Cosas (IoT)? <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>
- [18] Oracle; ¿Qué es Internet of Things (IoT)? <https://www.oracle.com/cl/internet-of-things/what-is-iot.html>
- [19] Christopher Holloway (2018, noviembre 01) Plataformas IoT: Que son y cómo elegir la mejor para el negocio. <https://itmastersmag.com/noticias-analisis/plataformas-iot-que-son-y-como-elegir-la-mejor-para-el-negocio/>
- [20] MecatronicaLATAM <https://www.mecatronicalatam.com/es/tutoriales/sensores/>
- [21] Mena, Como Funciona. <https://como-funciona.co/sensor-de-movimiento/>
- [22] Juan Carlos G (2021, septiembre 17) ¿Qué son los sensores de colores? Y ¿Cómo funcionan? <https://sensormanía.org/sensor-de-color#Aplicaciones-del-sensor-de-color>
- [23] MecatronicaLATAM (2021, mayo 04) Sensores <https://www.mecatronicalatam.com/es/tutoriales/sensores/>
- [24] Tuatara Tech (2015) Sensores (Sensors) vs Actuadores (Actuators) https://www.tuataratech.com/2015/06/sensores-sensors-vs-actuadores-actuators_8.html
- [25] Aprendiendo Arduino. Hardware Arduino para la educación. <https://aprendiendoarduino.wordpress.com/2018/05/06/hardware-arduino-para-la-educacion/>
- [26] BricoGeek, <https://tienda.bricogeek.com/placas-raspberry-pi/1089-raspberry-pi-3-b-plus.html>
- [27] Reichelt elektronik. <https://www.reichelt.com/de/en/arduino-uno-rev-3-atmega328-usb-arduino-uno-p119045.html>

- [28] Alfonso Esparza (2018, septiembre 18) Zigbee Wifi: domótica sustentable para hoteles. <https://insisfenix.com/2018/09/18/zigbee-wifi/>
- [29] Manuel J. Gutiérrez (2015, agosto 10) Todo sobre Zigbee, la tecnología ultra barata para comunicación inalámbrica. <https://elandroidelibre.lespanol.com/2015/08/todo-sobre-zigbee-la-tecnologia-ultrabarata-para-comunicacion-inalambrica.html>
- [30] Tania Alonso Cascallana (2010, marzo 26) IoMT: La importancia del internet de las cosas médicas en contextos como el actual. <https://hablemosdeempresas.com/empresa/iomt/>
- [31] Subsecretaria de Salud Publica (2020, julio 02) Protocolo de Coordinación para acciones de vigilancia epidemiológica durante la pandemia COVID 19 en Chile, Estrategia de Testeo, Trazabilidad y Aislamiento. <https://www.minsal.cl/wp-content/uploads/2020/07/Estrategia-Testeo-Trazabilidad-y-Aislamiento.pdf>
- [32] theastrology (2022) ¿Qué es la identificación de usuario (ID de usuario)? – definición de techopedia - Seguridad – 2022. <https://es.theastrologypage.com/user-identification>.
- [33] 24horas.cl Tvn (2020, febrero 28) Ranking: Los países mejor y peor preparados para enfrentar una epidemia y la ubicación de Chile. <https://www.24horas.cl/data/ranking-los-paises-mejor-y-peor-preparados-para-enfrentar-una-epidemia-y-la-ubicacion-de-chile-3975885>
- [34] BBC News Mundo (2020, septiembre 30) Coronavirus: el mapa que muestra el número de infectados y muertos en el mundo por covid-19. <https://www.bbc.com/mundo/noticias-51705060>
- [35] Google (2022) Datos de estadística sobre el coronavirus (COVID-19). https://support.google.com/websearch/answer/9814707?hl=es&visit_id=637884125605467431-69459243&p=cvd19_statistics&rd=1
- [36] Statista (2020, octubre 23) Número de casos Confirmados de coronavirus en el mundo a fecha 23 de octubre de 2020, por país. <https://es.statista.com/estadisticas/1091192/paises-afectados-por-el-coronavirus-de-wuhan-segun-los-casos-confirmados/>
- [37] Minsal, Plan de acción coronavirus COVID-19. <https://www.minsal.cl/nuevo-coronavirus-2019-ncov/>
- [38] Organización Panamericana de la Salud (OPS) (2020, agosto 28) Afiche-Síntomas COVID-19. <https://www.paho.org/es/documentos/afiche-sintomas-covid-19>.
- [39] Conrad, Placa de desarrollo Joy-it Node MCU ESP32 modul. <https://www.conrad.com/p/joy-it-development-board-node-mcu-esp32-modul-1656367>
- [40] Raúl Diosdado, Sensor de ultrasonidos HC-SR04. <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>
- [41] Mechatronic store, Sensor de temperatura GY-906 BBA infrarrojo. <https://www.mechatronicstore.cl/sensor-de-temperatura-gy-906-mlx90614esf/>
- [42] Pcfactory <https://www.pcfactory.cl/producto/30436-adafruit-camara-termica-ir-amg8833>

- [43] Iberobotics, Modulo RFID MIFARE RC522 MFRC522 lector/grabador + tarjeta + llavero. <https://www.iberobotics.com/producto/modulo-rfid-mifare-rc522-mfrc522-lector-grabador-kit-tarjeta-llavero/>
- [44] Jeremy Morgan (2015, diciembre 26) Getting Started with Adafruit.IO <https://www.jeremymorgan.com/tutorials/raspberry-pi/how-to-adafruit-io>.
- [45] Aplicación móvil de Blynk. Disponible en: <https://play.google.com/store/apps>
- [46] Administrador Digital55 (2020, mayo 17) Firebase: qué es, para qué sirve, funcionalidades y ventajas. <https://digital55.com/que-es-firebase-funcionalidades-ventajas-conclusiones/> .
- [47] Pagina de Firebase. Disponible en: <https://firebase.google.com/>
- [48] Pagina de Altronics. Disponible en: <https://altronics.cl/display-oled-128x64-i2c>
- [49] Pagina de HYNAUT. Disponible en: <https://www.hynautgroup.cn/product/infrared-thermometer-axd-518/>
- [50] Mubeen Abbas (2015, noviembre 06) Panasonic: Infrared Grideye Sensor.
- [51] Pau Seguí (Pablo)/OVACEN. Cámaras termográficas: Como detectar fiebre ante Coronavirus COVID 19. <https://ovacen.com/camaras-termograficas-fiebre/>
- [52] Pagina de Adafruit. Disponible en: <https://io.adafruit.com/>

Anexo A. Verificación de Calibración.

En la tabla A.1 se presenta la verificación de la calibración a la AMG 8833 a los 10 cm.

TABLA A.1. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 10 cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
35,22	36,5	1,28
35,88	36,5	0,62
36,03	36,5	0,47
35,79	36,5	0,71
35,71	36,5	0,79
35,87	36,5	0,63
35,89	36,5	0,61
35,52	36,5	0,98
35,38	36,5	1,12
35,27	36,5	1,23

En la tabla A.2 se presenta la verificación de la calibración a la AMG 8833 a los 15 cm.

TABLA A.2. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 15 cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
36,4	36,5	0,1
36,43	36,5	0,07
36,58	36,5	0,08
37,08	36,5	0,58
36,82	36,5	0,32
36,8	36,5	0,3
36,64	36,5	0,14
37,01	36,5	0,51
36,88	36,5	0,38
36,5	36,5	0

En la tabla A.3 se presenta la verificación de la calibración a la AMG 8833 a los 20 cm.

TABLA A.3. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 20 cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
35,07	36,5	1,43
35,33	36,5	1,17
35,62	36,5	0,88
35,44	36,5	1,09
35,48	36,5	1,02
35,68	36,5	0,82
35,79	36,5	0,71
35,62	36,5	0,88
35,42	36,5	1,08
35,41	36,5	1,09

En la tabla A.4 se presenta la verificación de la calibración a la AMG 8833 a los 25 cm.

TABLA A.4. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 25 cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
36,95	36,5	0,45
36,88	36,5	0,38
36,84	36,5	0,34
36,44	36,5	0,06
36,59	36,5	0,09
36,47	36,5	0,03
36,65	36,5	0,15
36,34	36,5	0,16
36,42	36,5	0,08
36,34	36,5	0,16

En la tabla A.5 se presenta la verificación de la calibración a la AMG 8833 a los 30 cm.

TABLA A.5. Comparación Temperatura AMG 8833 luego de la calibración y la temperatura del termómetro infrarrojo a 30 cm.

Temperatura AMG 8833	Temperatura Termómetro	Diferencia
37,1	36,5	0,6
36,75	36,5	0,25
36,57	36,5	0,07
36,54	36,5	0,04
36,44	36,5	0,06
36,78	36,5	0,28
36,73	36,5	0,23
36,33	36,5	0,17
36,35	36,5	0,15
36,3	36,5	0,2

Anexo B. Códigos

```

1 #include <WiFi.h>
2 #include <FirebaseESP32.h>
3 #include <addons/TokenHelper.h>
4 #include <addons/RTDBHelper.h>
5 #include <SPI.h>
6 #include <RFID.h>
7 #include <Adafruit_GFX.h>
8 #include <Adafruit_SSD1306.h>
9 #include <Wire.h>
10
11 String path = "";
12 String pathh = "";
13
14
15 #define SS_PIN 5
16 #define RST_PIN 14
17
18 RFID rfid(SS_PIN, RST_PIN);
19 RFID RC522(SS_PIN, RST_PIN);
20
21 ////////////////////////////////////////////////// definiciones de la red WiFi ///////////////////////////////////
22 #define WIFI_SSID ""
23 #define WIFI_PASSWORD ""
24
25
26 ////////////////////////////////////////////////// definiciones de la Firebase ///////////////////////////////////
27 #define API_KEY ""
28 #define USER_EMAIL ""
29 #define USER_PASSWORD ""
30 #define DATABASE_URL ""
31 #define DATABASE_SECRET ""
32 FirebaseData fbdo;
33 FirebaseAuth auth;
34 FirebaseConfig config;
35 unsigned long dataMillis = 0;
36 int contador = 0;
37 int contadores = 0;
38 int PIREntrada = 4;
39 int valorentrada;
40 int respuesta_correcta=0;
41 int readData;
42
43     int val0;
44     int val1;
45     int val2;
46     int val3;
47     int val4;
48
49     int dec128=1;
50     int dec224=4;
51     int dec112=3;
52     int dec16=5;
53     int dec0=6;
54
55     int a;
56     int posicion;
57 FirebaseData firebaseData;
58
59
60
61 unsigned long sendDataPrevMillis = 0;
62
63 unsigned long count = 0;
64
65 WiFiClient client; // Create an ESP8266 WiFiClient class to connect to the MQTT server.
66
67 ////////////////////////////////////////////////// definicion Setup//////////////////////////////////////
68
69 #define Ancho_pantalla 128
70 #define Largo_pantalla 64
71 #define OLED_RESET -1
72 Adafruit_SSD1306 display(Ancho_pantalla, Largo_pantalla, &Wire, OLED_RESET);
73
74
75 void setup()
76 {
77
78     Serial.begin(115200);
79
80     pinMode(PIREntrada, INPUT);
81
82     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
83     Serial.print("Te has conectado a la red WiFi");
84     while (WiFi.status() != WL_CONNECTED)
85     {
86         Serial.print(".");
87         delay(300);
88     }
89     Serial.println();
90     Serial.print("Te has conectado con la IP: ");
91     Serial.println(WiFi.localIP());
92     Serial.println();
93
94     Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
95
96     /* Assign the api key (required) */
97     config.api_key = API_KEY;
98
99     /* Assign the user sign in credentials */
100     auth.user.email = USER_EMAIL;
101     auth.user.password = USER_PASSWORD;
102
103     /* Assign the RTDB URL (required) */
104     config.database_url = DATABASE_URL;

```

```

106  /* Assign the callback function for the long running token generation task */
107  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
108
109  Firebase.begin(&config, &auth);
110
111  //Comment or pass false value when WiFi reconnection will control by your code or third party library
112  Firebase.reconnectWiFi(true);
113
114  Firebase.setDoubleDigits(5);
115
116  SPI.begin();
117  rfid.init();
118  RC522.init();
119
120  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
121  display.setTextColor(SSD1306_WHITE);
122  display.clearDisplay();
123  display.setCursor(0,0);
124  display.setTextSize(1);
125  display.println("Sistema de");
126  display.setTextSize(1);
127  display.setCursor(0,20);
128  display.print("Identificacion activo");
129  display.display();
130  delay(2000);
131
132 }
133
134 void loop()
135 {
136   posicion=0;
137   if (rfid.isCard()) {
138     if (rfid.readCardSerial()) {
139       if (rfid.serNum[0] != val0
140         && rfid.serNum[1] != val1
141         && rfid.serNum[2] != val2
142         && rfid.serNum[3] != val3
143         && rfid.serNum[4] != val4
144       ) {
145         /* With a new cardnumber, show it. */
146         Serial.println(" ");
147         Serial.println("Tarjeta leida");
148         val0 = rfid.serNum[0];
149         val1 = rfid.serNum[1];
150         val2 = rfid.serNum[2];
151         val3 = rfid.serNum[3];
152         val4 = rfid.serNum[4];
153
154         //Serial.println(" ");
155         Serial.println("IDE de la tarjeta:");
156         Serial.print("Dec: ");
157         Serial.print(rfid.serNum[0],DEC);
158         Serial.print(", ");
159         Serial.print(rfid.serNum[1],DEC);
160         Serial.print(", ");
161         Serial.print(rfid.serNum[2],DEC);
162         Serial.print(", ");
163         Serial.print(rfid.serNum[3],DEC);
164         Serial.print(", ");
165         Serial.print(rfid.serNum[4],DEC);
166         Serial.println(" ");
167
168
169         Serial.println("El primer valor decimal es:");
170         Serial.print(val0);
171         Serial.println();
172
173         if (val0 == 128){
174           Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"128") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
175           posicion= Firebase.getInt(fbdo, pathh + "/"128");
176           Serial.println("La posicion de la variable es:");
177           Serial.print(posicion);
178           Serial.print("La posicion es de la variable es:");
179           Serial.print(posicion);
180           display.clearDisplay();
181           display.setTextSize(1);
182           display.setCursor(0,0);
183           display.print("La ID del usuario es la Numero:");
184           display.setTextSize(2);
185           display.setCursor(50,40);
186           display.print(Firebase.getInt(fbdo, pathh + "/"128") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
187           display.display();
188           delay(1500);
189
190           Serial.println();
191
192           display.clearDisplay();
193           display.setTextSize(1);
194           display.setCursor(0,0);
195           display.print("El aforo actual es:");
196           display.setTextSize(1);
197           display.setCursor(0,15);
198           display.print(Aforo);
199           display.setTextSize(1);
200           display.setCursor(0,30);
201           display.print("Su temperatura es:");
202           display.setTextSize(2);
203           display.setCursor(50,45);
204           display.print(Temp);
205           display.display();
206           delay(1000);

```

```

207     }
208     if (val0 == 2){
209         Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"2") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
210         posicion=Firebase.getInt(fbdo, pathh + "/"2");
211         Serial.print("La posicion de la variable es:");
212         Serial.print(posicion);
213         Serial.println();
214         display.clearDisplay();
215         display.setTextSize(1);
216         display.setCursor(0,0);
217         display.print("La ID del usuario es la Numero:");
218         display.setTextSize(2);
219         display.setCursor(50,40);
220         display.print(Firebase.getInt(fbdo, pathh + "/"2") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
221         display.display();
222         delay(1500);
223
224         display.clearDisplay();
225         display.setTextSize(1);
226         display.setCursor(0,0);
227         display.print("El aforo actual es:");
228         display.setTextSize(1);
229         display.setCursor(0,15);
230         display.print(Aforo);
231         display.setTextSize(1);
232         display.setCursor(0,30);
233         display.print("Su temperatura es:");
234         display.setTextSize(2);
235         display.setCursor(50,45);
236         display.print(Temp);
237         display.display();
238         delay(1000);
239
240     }
241     if (val0 == 16){
242         Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"16") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
243         posicion=Firebase.getInt(fbdo, pathh + "/"16");
244         Serial.print("La posicion de la variable es:");
245         Serial.print(posicion);
246         Serial.println();
247         display.clearDisplay();
248         display.setTextSize(1);
249         display.setCursor(0,0);
250         display.print("La ID del usuario es la Numero:");
251         display.setTextSize(2);
252         display.setCursor(50,40);
253         display.print(Firebase.getInt(fbdo, pathh + "/"16") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
254         display.display();
255         delay(1500);
256
257         display.clearDisplay();
258         display.setTextSize(1);
259         display.setCursor(0,0);
260         display.print("El aforo actual es:");
261         display.setTextSize(1);
262         display.setCursor(0,15);
263         display.print(Aforo);
264         display.setTextSize(1);
265         display.setCursor(0,30);
266         display.print("Su temperatura es:");
267         display.setTextSize(2);
268         display.setCursor(50,45);
269         display.print(Temp);
270         display.display();
271         delay(1000);
272
273     }
274     if (val0 == 250){
275         Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"224") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
276         posicion=Firebase.getInt(fbdo, pathh + "/"224");
277         Serial.print("La posicion de la variable es:");
278         Serial.print(posicion);
279         Serial.println();
280         display.clearDisplay();
281         display.setTextSize(1);
282         display.setCursor(0,0);
283         display.print("La ID del usuario es la Numero:");
284         display.setTextSize(2);
285         display.setCursor(50,40);
286         display.print(Firebase.getInt(fbdo, pathh + "/"224") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
287         display.display();
288         delay(1500);
289
290         display.clearDisplay();
291         display.setTextSize(1);
292         display.setCursor(0,0);
293         display.print("El aforo actual es:");
294         display.setTextSize(1);
295         display.setCursor(0,15);
296         display.print(Aforo);
297         display.setTextSize(1);
298         display.setCursor(0,30);
299         display.print("Su temperatura es:");
300         display.setTextSize(2);
301         display.setCursor(50,45);
302         display.print(Temp);
303         display.display();
304         delay(1000);
305     }

```

```

306 if (val0 == 0) {
307     Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"0") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
308     posicion= Firebase.getInt(fbdo, pathh + "/"0");
309     Serial.println("La posicion de la variable es:");
310     Serial.print(posicion);
311     Serial.println();
312     display.clearDisplay();
313     display.setTextSize(1);
314     display.setCursor(0,0);
315     display.print("La ID del usuario es la Numero:");
316     display.setTextSize(2);
317     display.setCursor(50,40);
318     display.print(Firebase.getInt(fbdo, pathh + "/"0") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
319     display.display();
320     delay(1500);
321
322     display.clearDisplay();
323     display.setTextSize(1);
324     display.setCursor(0,0);
325     display.print("El aforo actual es:");
326     display.setTextSize(1);
327     display.setCursor(0,15);
328     display.print(Aforo);
329     display.setTextSize(1);
330     display.setCursor(0,30);
331     display.print("Su temperatura es:");
332     display.setTextSize(2);
333     display.setCursor(50,45);
334     display.print(Temp);
335     display.display();
336     delay(1000);
337
338 }
339
340 if (val0 == 112) {
341     Serial.printf("La ID del usuario es la N°%s\n",Firebase.getInt(fbdo, pathh + "/"112") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
342     posicion= Firebase.getInt(fbdo, pathh + "/"112");
343     Serial.println("La posicion de la variable es:");
344     Serial.print(posicion);
345     Serial.println();
346     display.clearDisplay();
347     display.setTextSize(1);
348     display.setCursor(0,0);
349     display.print("La ID del usuario es la Numero:");
350     display.setTextSize(2);
351     display.setCursor(50,40);
352     display.print(Firebase.getInt(fbdo, pathh + "/"112") ? String(fbdo.to<int>()).c_str() : fbdo.errorReason().c_str());
353     display.display();
354     delay(1500);
355
356     display.clearDisplay();
357     display.setTextSize(1);
358     display.setCursor(0,0);
359     display.print("El aforo actual es:");
360     display.setTextSize(1);
361     display.setCursor(0,15);
362     display.print(Aforo);
363     display.setTextSize(1);
364     display.setCursor(0,30);
365     display.print("Su temperatura es:");
366     display.setTextSize(2);
367     display.setCursor(50,45);
368     display.print(Temp);
369     display.display();
370     delay(1000);
371
372 } else {
373     /* If we have the same ID, just write a dot. */
374     Serial.print(".");
375 }
376
377 }
378
379 if(checkAuthorization(rfid.serNum))
380 {
381     display.clearDisplay();
382     display.setTextSize(1);
383     display.setCursor(0,0);
384     display.print("Nombre: ");
385     display.setTextSize(1);
386     display.setCursor(0,15);
387     display.print(Firebase.getString(fbdo, path + "/"ID"+a + "/Datos Personales/Nombre") ? fbdo.to<const char *>() : fbdo.errorReason().c_str());
388     display.setTextSize(1);
389     display.setCursor(0,30);
390     display.print("Acceso: ");
391     display.setTextSize(1);
392     display.setCursor(0,45);
393     display.print(" Autorizado ");
394     display.display();
395     delay(1500);
396     Serial.println("Acceso: Autorizado");
397     delay(5000);
398
399     display.clearDisplay();
400     display.setCursor(0,0);
401     display.setTextSize(1);
402     display.println("Sistema de");
403     display.setTextSize(1);

```

```

406 display.setCursor(0,20);
407 display.print("Identificacion activo");
408 display.display();
409 delay(200);
410 }
411 else
412 {
413     Serial.println("Acceso: Denegado");
414
415     display.clearDisplay();
416     display.setTextSize(1);
417     display.setCursor(0,0);
418     display.print("Acceso: ");
419     display.setTextSize(1);
420     display.setCursor(0,15);
421     display.print(" Denegado ");
422     display.display();
423     delay(5000);
424
425     display.clearDisplay();
426     display.setCursor(0,0);
427     display.setTextSize(1);
428     display.println("Sistema de");
429     display.setTextSize(1);
430     display.setCursor(0,20);
431     display.print("Identificacion activo");
432     display.display();
433     delay(2000);
434 }
435 Serial.println();
436 Serial.println();
437 }
438
439
440 rfid.halt();
441
442
443 }
444
445 bool checkAuthorization(byte* data) //Verify Access
446 {
447     a=posicion;
448     // Serial.println(a);
449     // Serial.println(posicion);
450     Serial.println();
451
452     if(a==0) return false;
453
454     for(int i=0;i<5;i++)
455     {
456         Firebase.getInt(fbdo, path + "/"ID"+posicion + "/Identificacion" + "/"RFID"+i);
457         Serial.println(fbdo.intData());
458         if (fbdo.intData() != data[i])
459             return false;
460         delay(10);
461     }
462
463     Serial.printf("Nombre: %s\n", Firebase.getString(fbdo, path + "/"ID"+a + "/Datos Personales/Nombre") ? fbdo.to<const char *>() : fbdo.errorReason().c_str());
464     return true;
465
466 }

```

```

1 #include <Wire.h>
2 #include <Adafruit_GFX.h>
3 #include <Adafruit_SSD1306.h>
4 #include <Adafruit_AMG88xx.h>
5 #include <Adafruit_MLX90614.h>
6 #include "Adafruit_MQTT.h"
7 #include "Adafruit_MQTT_Client.h"
8 #define BLYNK_PRINT Serial
9 #include <WiFi.h>
10 #include <WiFiClient.h>
11 #include <BlynkSimpleEsp32.h>
12
13 char auth[] = ;
14 char ssid[] = ;
15 char pass[] = ;
16 #define WLAN_SSID
17 #define WLAN_PASS
18 #define AIO_SERVERPORT
19 #define AIO_USERNAME
20 #define AIO_KEY
21
22 const char MQTT_SERVER[] = AIO_SERVER;
23 const char MQTT_USERNAME[] = AIO_USERNAME;
24 const char MQTT_PASSWORD[] = AIO_KEY;
25
26
27 WiFiClient client;
28
29 // Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
30 Adafruit_MQTT_Client mqtt(&client, MQTT_SERVER, AIO_SERVERPORT, MQTT_USERNAME, MQTT_PASSWORD);
31
32 //***** Feeds *****/
33
34 Adafruit_MQTT_Publish temperatura = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME"/feeds/temperatura");
35 Adafruit_MQTT_Publish aforo = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME"/feeds/aforo");
36
37 /*****
38
39
40 Adafruit_AMG88xx amg;
41 Adafruit_MLX90614 mlx = Adafruit_MLX90614();
42 float pixels[AMG88xx_PIXEL_ARRAY_SIZE];
43 int contador;
44 float sumatoria;
45 float promedio;
46 int a;
47 float temperatura_promedio=37.00;
48 float temp_objeto;
49 float temp_ambiente;
50
51 int trig =13;
52 int trigger=27;
53 int echo = 12;
54 int echoo = 4;
55 //float temperatura;
56 int Aforo =0;
57 int valorentrada=0;
58 int valorsalida=0;
59 int aforo_total=10;
60 int aforo_restante=0;
61
62 void setup()
63 {
64   Serial.begin (115200);
65
66   Serial.println(F("AMG88xx pixels"));
67   bool status;
68
69   // default settings
70   status = amg.begin();
71   if (!status) {
72     Serial.println("Could not find a valid AMG88xx sensor, check wiring!");
73     while (1);
74   }
75
76   Serial.println("-- Pixels Test --");
77   Serial.println();
78   delay(100);
79
80   mlx.begin();
81   delay (2000);
82
83   pinMode(trig,OUTPUT);
84   pinMode(echo,INPUT);
85   pinMode(trigger,OUTPUT);
86   pinMode(echoo,INPUT);
87   Serial.println("Inicializacion del sistema de Aforo");
88
89   Blynk.begin(auth, ssid, pass);
90 }
91
92 void loop()
93 {
94   MQTT_connect();
95
96   long duracion;
97   long distancia;
98   long duracionT;
99   long distanciaT;
100
101 digitalWrite(trig,LOW);

```

```

102 delayMicroseconds(4);
103 digitalWrite(trig,HIGH);
104 delayMicroseconds(10);
105 digitalWrite(trig,LOW);
106
107 duracion = pulseIn(echo,HIGH);
108 duracion = duracion/2;
109
110 distancia =duracion/29;
111 if (distancia > 0 && distancia <=70){
112
113     Aforo = Aforo +1;
114
115     if (Aforo <= aforo_total){
116         Aforo= Aforo;
117         String StrUno= "La ocupacion del edificio es de: ";
118         String StrDos= StrUno + Aforo;
119         Serial.println(StrDos);
120     }
121     if (Aforo > aforo_total){
122         Aforo= aforo_total;
123         Serial.println("Se a llegado a la ocupacion maxima del edificio");
124     }
125
126     Serial.print("Ambient MLX = "); Serial.print(mlx.readAmbientTemp()); Serial.println(" °C");
127     Serial.print("Object MLX = "); Serial.print(mlx.readObjectTempC()); Serial.println(" °C");
128     Serial.println();
129
130     sumatoria=0;
131     contador=0;
132     promedio=0;
133
134     amg.readPixels(pixels);
135
136     // Serial.print("[");
137     for(int i=1; i<=AMG88xx_PIXEL_ARRAY_SIZE; i++){
138         // Serial.print(pixels[i-1]);
139         // Serial.print(", ");
140         if( i%8 == 0 ) //Serial.println();
141         a=pixels[i-1];
142         if (a >=mlx.readObjectTempC()){
143             sumatoria=a+sumatoria;
144             contador=1+contador;
145         }
146     }
147     promedio= sumatoria/contador;
148
149     digitalWrite(trigger,LOW);
150     delayMicroseconds(4);
151     digitalWrite(trigger,HIGH);
152     delayMicroseconds(10);
153     digitalWrite(trigger,LOW);
154
155     duracionT = pulseIn(echo,HIGH);
156     duracionT = duracionT/2;
157
158     distanciaT =duracionT/29;
159     delay (500);
160     String StrdistT= "La distancia del objeto a la camara termica es: ";
161     String Strdiast= "cms";
162     String Strdist= StrdistT + distanciaT + Strdiast;
163     Serial.println(Strdist);
164
165     if(distanciaT>30){
166         Serial.println("no se detecta objeto entre las distancias recomendadas, volver a intentar");
167     }
168     Aforo = Aforo-1;
169 }
170
171 if(distanciaT>=0 && distanciaT <=30){
172     if(distanciaT <= 5){
173         promedio = promedio+8.94;
174     }
175     if (distanciaT >5 && distanciaT<= 10){
176         promedio = promedio+9.82;
177     }
178     if (distanciaT >10 && distanciaT<= 15){
179         promedio = promedio+11.45;
180     }
181     if (distanciaT >15 && distanciaT<= 20){
182         promedio = promedio+10.89;
183     }
184     if (distanciaT >20 && distanciaT<= 25){
185         promedio = promedio+12.2;
186     }
187     if (distanciaT >25 && distanciaT<= 30){
188         promedio = promedio+12.84;
189     }
190     String StrdistanciaU= "La distancia de deteccion fue: ";
191     String StrdistanciaD= "cms";
192     String StrdistanciaT= StrdistanciaU + distanciaT + StrdistanciaD;
193     Serial.println(StrdistanciaT);
194     delay (2000);
195 }
196
197 String StrtemperaturaU= "La medicion de temperatura es de : ";
198 String StrtemperaturaD= "°C";
199 String StrtemperaturaT= StrtemperaturaU + promedio + StrtemperaturaD;
200 Serial.println(StrtemperaturaT);

```

```

201     delay(2000);
202 }
203
204 temp_objeto= mlx.readObjectTempC();
205 temp_ambiente= mlx.readAmbientTempC();
206
207 //delay a second
208 delay(2000);
209
210 aforo_restante=aforo_total-Aforo;
211
212 Blynk.run();
213 Blynk.virtualWrite(V3, promedio );
214 Blynk.virtualWrite(V4, temp_objeto);
215 Blynk.virtualWrite(V5, temp_ambiente);
216 Blynk.virtualWrite(V6, distancia);
217 Blynk.virtualWrite(V0, Aforo);
218 Blynk.virtualWrite(V1, aforo_total);
219 Blynk.virtualWrite(V2, aforo_restante);
220
221 if (!temperatura.publish(promedio))           // Publish to Adafruit the PIR sensor value '1'
222 {
223     Serial.println(F("Failed"));              // If it failed to publish, print Failed
224 } else
225 {
226     Serial.println(F("Data Sent!"));          // If data successfully published
227 }
228     delay (1000);
229
230 if (!aforo.publish(Aforo))                   // Publish to Adafruit the PIR sensor value '1'
231 {
232     Serial.println(F("Failed"));              // If it failed to publish, print Failed
233 } else
234 {
235     Serial.println(F("Data Sent!"));          // If data successfully published
236 }
237     delay(1000);
238
239 if(distancia > 70 && distancia <=140){
240
241     Aforo = Aforo-1;
242     if(Aforo<=0){
243         Aforo =0;
244         Serial.println("No se encuentra nadie en el edificio");
245         String StrUno= "La ocupacion del edificio es de: ";
246         String StrDos= StrUno + Aforo;
247         Serial.println(StrDos);
248         delay(1000);
249     }
250
251     if(Aforo<=10 && Aforo >0){
252         Serial.println("Gracias por su visita");
253         String StrUno= "La ocupacion del edificio es de: ";
254         String StrDos= StrUno + Aforo;
255         Serial.println(StrDos);
256         delay(1000);
257     }
258     aforo_restante=aforo_total-Aforo;
259
260 Blynk.run();
261 Blynk.virtualWrite(V0, Aforo);
262 Blynk.virtualWrite(V1, aforo_total);
263 Blynk.virtualWrite(V2, aforo_restante);
264
265 if (!aforo.publish(Aforo))                   // Publish to Adafruit the PIR sensor value '1'
266 {
267     Serial.println(F("Failed"));              // If it failed to publish, print Failed
268 } else
269 {
270     Serial.println(F("Data Sent!"));          // If data successfully published
271 }
272     delay(1000);
273 }
274
275
276 if(distancia >=140){
277     Aforo;
278     if(Aforo<=0 && Aforo <10){
279         Serial.println("No se ah detectado entrada ni salida en este momento, pero aun no se ah llegado a la maxima capacidad del edificio");
280         delay(1000);
281     }
282     if(Aforo>=10){
283         Serial.println("No se ah detectado entrada ni salida en este momento, pero se ah llegado a la maxima capacidad del edificio, porfavor espere");
284         delay(1000);
285     }
286     aforo_restante=aforo_total-Aforo;
287
288 Blynk.run();
289 Blynk.virtualWrite(V0, Aforo);
290 Blynk.virtualWrite(V1, aforo_total);
291 Blynk.virtualWrite(V2, aforo_restante);
292
293 if (!aforo.publish(Aforo))                   // Publish to Adafruit the PIR sensor value '1'
294 {
295     Serial.println(F("Failed"));              // If it failed to publish, print Failed
296 } else
297 {
298     Serial.println(F("Data Sent!"));          // If data successfully published
299 }
300     delay(1000);
301
302 }
303 }
304
305 void MQTT_connect()
306 {
307     int8_t ret;
308     if (mqtt.connected()) { return; }          // Stop if already connected to Adafruit
309     Serial.println("Connecting to MQTT... ");
310
311     uint8_t retries = 3;
312     while ((ret = mqtt.connect()) != 0) {      // Connect to Adafruit, will return 0 if connected
313         Serial.println(mqtt.connectErrorString(ret));
314         Serial.println("Retrying MQTT...");
315         mqtt.disconnect();
316         delay(100);                            // wait 5 seconds
317         retries--;
318         if (retries == 0) {                    // basically die and wait for WatchDogTimer to reset me
319             while (1);
320         }
321     }
322     Serial.println("MQTT Connected!");
323 }

```