



**UNIVERSIDAD POLITÉCNICA  
SALESIANA SEDE GUAYAQUIL  
CARRERA DE  
TELECOMUNICACIONES**

**DESARROLLO DE UN SISTEMA INTELIGENTE DE ALERTA  
TEMPRANA PARA MONITOREO DE VARIABLE CRÍTICAS DE  
SALUD CON NOTIFICACIÓN VÍA MENSAJERÍA EN CABINAS  
INSONORIZADAS**

Trabajo de titulación previo a la obtención del  
Título de ingeniero en telecomunicaciones

**AUTORES:** LUIS FERNANDO CUNALEMA ZAMBRANO  
CARLOS DANIEL VERA IBARRA

**TUTOR:** JOSE RENATO CORDOVA LEÓN, Msc

Guayaquil - Ecuador

2025

**II. Certificado De Responsabilidad Y Autoría Del Trabajo De Titulación**

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE  
TITULACIÓN**

Nosotros, Luis Fernando Cunalema Zambrano con documento de identificación N° 0952778272, y Carlos Daniel Vera Ibarra con documento de identificación N° 1350135362 manifestamos que:

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

Guayaquil, del mes de agosto del año 2025.

Atentamente,



---

Luis Fernando Cunalema

Zambrano

0952778272



---

Carlos Daniel Vera Ibarra

1350135362



**III. Certificado De Cesión De Derechos De Autor Del Trabajo De Titulación A  
La Universidad Politécnica Salesiana.**

Nosotros, Luis Fernando Cunalema Zambrano, con C.I. 0952778272 y Carlos Daniel Vera Ibarra, con C.I. 1350135362, expresamos nuestra voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana del Ecuador la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Científico: “DESARROLLO DE UN SISTEMA INTELIGENTE DE ALERTA TEMPRANA PARA MONITOREO DE VARIABLE CRÍTICAS DE SALUD CON NOTIFICACIÓN VÍA MENSAJERÍA EN CABINAS INSONORIZADAS”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Institución facultada para ejercer plenamente los derechos concedidos.

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


Guayaquil, del mes de agosto del año 2024.

Atentamente,

  
\_\_\_\_\_  
Luis Fernando Cunalema

Zambrano

0952778272

  
\_\_\_\_\_  
Carlos Daniel Vera Ibarra

1350135362

**IV. Certificado de Dirección del Trabajo de Titulación.**

**CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.**

Yo, Jose Renato Cordova León, con documento de identificación N° 0919123067 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: "DESARROLLO DE UN SISTEMA INTELIGENTE DE ALERTA TEMPRANA PARA MONITOREO DE VARIABLE CRÍTICAS DE SALUD CON NOTIFICACIÓN VÍA MENSAJERÍA EN CABINAS INSONORIZADAS", realizado por , Luis Fernando Cunalema Zambrano con documento de identificación N° 0952778272, y Carlos Daniel Vera Ibarra con documento de identificación N° 1350135362, obteniendo como resultado final el trabajo de titulación bajo la opción de Artículo académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, del mes de agosto del año 2025.

Atentamente,



Jose Renato Cordova León

0919123067

## V. Dedicatoria

Esta tesis la dedico con toda mi gratitud y afecto a mi familia, en particular a mis padres, quienes han sido el soporte esencial en mi vida. Su apoyo incondicional, sus palabras de motivación y sus esperanzas en mí me han motivado a vencer cada dificultad en este trayecto. Agradezco que sean mi motivación y que me enseñen que, con empeño y tenacidad, todo puede lograrse.

A mis seres queridos cercanos, cuyo cariño y entendimiento me han apoyado en los momentos más difíciles de este proceso, deseo manifestar mi más sincero agradecimiento. Deseo también honrar el recuerdo de aquellas personas que ya no están a mi lado, pero que dejaron una marca indeleble en mi vida. Su herencia y lecciones me siguen en cada avance, y estoy convencido de que estarían orgullosos de este logro.

## **VI. Agradecimiento**

Me siento profundamente agradecido con mi familia, en particular con mis padres y mi hermana, quienes con su amor, respaldo continuo y dedicación diaria han sido esencial en mi desarrollo personal y académico. Su confianza en mí ha sido mi mayor incentivo para continuar y esforzarme en cada etapa de mi vida.

A todas las personas maravillosas que han hecho este trayecto conmigo, ya sea con palabras de motivación, su tiempo o su orientación, les agradezco de forma sincera por aportar de forma positiva a mi crecimiento personal y profesional.

Expreso mi agradecimiento a mi tutor, cuya dedicación y guía han resultado fundamentales para llevar a cabo este trabajo. Aparte de lo académico, su respaldo y apoyo han sido crucial en mi aprendizaje y crecimiento personal.

## VII. Resumen

Este sistema de alerta temprana tiene como propósito la seguridad y salud en entornos controlados como las cabinas insonorizadas, mediante el monitoreo continuo de variables de los datos críticos como la temperatura, la humedad y los niveles de oxígeno. Lo cual tiene como objetivo el desarrollar un sistema electrónico de alerta temprana que integre un programa preventivo de código abierto y sensores para recopilar y enviar información a un programa que registre, analice y pronostique un evento crítico. Por lo cual se diseñó un prototipo utilizando Arduino, con sensores como el DHT22, MQ-135 y el módulo HC-05 y se diseña un código en Python con la biblioteca de TensorFlow para la predicción y el envío de alerta a telegram. Las pruebas realizadas mostraron márgenes de error bajos, en el de temperatura se obtuvo 8.98% de error, en la humedad tiene 2.27% y en los de niveles de oxígeno de 0.10% lo que confirma la fiabilidad del sistema para emitir alertas en tiempo real. La integración del hardware y la configuración de notificaciones via telegram fueron exitosas. Concluyendo que el prototipo cumple con los objetivos propuestos, dado que, se logró diseñar un modelo predictivo que anticipa eventos críticos con alta precisión.

### i. Palabras Claves

1; Python 2; TensorFlow 3; Alerta temprana 4; Thingspeak 5; Monitoreo

## VIII. Abstract

This early warning system aims to strengthen the safety and health of controlled environments, such as soundproof booths, by continuously monitoring critical data variables such as temperature, humidity, and oxygen levels in parts per million (PPM). The goal is to develop an electronic early warning system that integrates an open-source preventive program and sensors to collect and send information to a program that records, analyzes, and predicts a critical event. Therefore, a prototype was designed using Arduino, along with sensors such as the DHT22, MQ-135, and the HC-05 module. Python code was also designed with the TensorFlow library for prediction and alert sending to Telegram. Tests showed low margins of error: 8.98% for temperature, 2.27% for humidity, and 0.10% for oxygen levels, confirming the system's reliability in issuing real-time alerts. Furthermore, the hardware integration and Telegram notification setup were successful. The conclusion is that the prototype meets the proposed objectives, as a predictive model was designed that anticipates critical events with high accuracy.

### ii. Keywords

1; Python 2; TensorFlow 3; Early warning 4; A Thingspeak 5; Monitoring

## IX. Índice de Contenido

I. Portada	1
II. Certificado De Responsabilidad Y Autoría Del Trabajo De Titulación	2
III. Certificado De Cesión De Derechos De Autor Del Trabajo De Titulación A La Universidad Politécnica Salesiana.	3
IV. Certificado de Dirección del Trabajo de Titulación.	4
V. Dedicatoria	5
VI. Agradecimiento	6
VII. Resumen	7
i. Palabras Claves	7
VIII. Abstract	8
ii. Keywords	8
IX. Índice de Contenido	9
X. Introducción	11
A. Arduino uno.	12
B. Sensor DHT 22.	12
C. Sensor MQ-135.	13
D. Módulo HC-05.	13
E. Tensor Flow.	14
F. Python.	14
G. Telegram.	14
H. Thingspeak.	15
I. Ecuaciones empleadas.	15
1. Cálculo del Índice de Variabilidad de Temperatura (IVT).	15
2. Cálculo de Índice de Variabilidad de Gas (IVG).	16
3. Tiempo de Respuesta del Sistema (TRS).	16
4. Cálculo del Margen de Error.	17
XI. Artículos Relacionados	18
XII. Metodología	20
a) Diseñar del código para el prototipo físico.	21
b) Diseñar del Bot de telegram y del Dashboard en Thingspeak.	21
c) Diseñar del código en Python con biblioteca de TensorFlow.	22
d) Esquema de prototipo.	23
e) Resultado del prototipo.	25
XIII. Interpretación de resultados	31

XIV. Discusión	37
XV. Conclusiones	40
XVI. Recomendaciones	41
XVII. Bibliografía	42
XVIII. Anexos	45

## X. Introducción

El cambio climático es un problema que ha llegado en los últimos años, estos cambios también afectan a las cabinas acústicas, laboratorio de investigación en bioacústica, laboratorio de pruebas de equipos médicos de, cabinas de radio, cabinas musicales, etc. Dado que mayor temperatura halla en el exterior aun mayor costo en lo operativo, además no solo el calor sería un problema las tormentas eléctricas serían más frecuentes y la sobrecarga en las redes eléctricas puede provocar interrupciones en el suministro energético. Como se presenta en el artículo [1] se subraya que actualmente Ecuador enfrenta un aumento de temperatura, lluvias más fuertes con tormentas eléctricas y un incremento de nivel del mar.

Este sistema de alerta temprana tiene como propósito fortalecer la seguridad y salud en entornos controlados como las cabinas insonorizadas, mediante el monitoreo continuo de variables de los datos críticos como la temperatura, la humedad y los niveles de oxígeno por partículas por millón (PPM). Este sistema cuenta con capacidades de comunicación la cual permite identificar condiciones que podrían poner en riesgo el bienestar físico del personal, especialmente en espacios cerrados donde una falla en la ventilación o en los sistemas de control atmosférico podría representar un potencial peligro para la salud humana [2].

Desarrollar un sistema electrónico de alerta temprana que integre un programa preventivo de código abierto y sensores capaces de recopilar y enviar información a una plataforma que registre, analice y pronostique un evento crítico. Para ello, se propone identificar los sensores y componentes más adecuados, evaluando su compatibilidad con la plataforma de hardware seleccionada y su capacidad de recopilar datos en tiempo real; desarrollar un prototipo y un modelo autónomo que permitan la recopilación y el análisis de datos para determinar en qué situaciones se debe emitir una notificación de posible evento crítico; y

finalmente, evaluar el desempeño del prototipo en un entorno controlado con el fin de validar su eficacia y confiabilidad [3].

Este sistema mide las diferentes variables que se encuentran presentes en las cabinas insonorizadas. Estas variables serán la calidad del aire, humedad y la temperatura, se utiliza el modelo de prevención basados en que permiten anticipar eventos críticos, como cambios bruscos de temperatura o disminución en los niveles de oxígeno.

En el contexto de un sistema de alerta temprana, permite integrar datos provenientes de sensores como el MQ-135 y el DHT22 para predecir eventos críticos, los cuales lleva a cabo en TensorFlow para poder hacer esas predicciones que pueden ser los cambios en la calidad del aire o condiciones en la temperatura o humedad, con un margen de error significativamente menor.

#### **A. Arduino uno.**

Arduino permite desarrollar diferentes prototipos y proyectos como podemos ver en el siguiente artículo [4] el cual desarrollaron un robot sumo de sumo con programación de Arduino, pero hay que añadir que la creación de prototipos con Arduino no es una tarea fácil para los no programadores interesados en el campo.

En el artículo [5] se desarrolla una placa de Arduino, este proyecto desarrollo seis servomotores, la cual lograron integrar una estructura de brazo de robot, el cual logro ejecutar los movimientos que están programados en los microcontroladores, este permite poder desarrollar prótesis para gente discapacitada y ayuda a recuperar esa capacidad de movimiento [6].

#### **B. Sensor DHT 22.**

Es este artículo [7] se presenta un diseño de una mini-estación autónomo, con estudiantes de la Facultad Ciencias Físico y Matemáticas-UNSCH que tiene el sensor DHT 22, el cual es un dispositivo que opera en diferentes campos científicos, este provee al sistema

parámetros que operan en un rango de 0-100% en una resolución de  $\pm 2\%$ , este trabajo es para controlar y monitorear la temperatura y la humedad del medio ambiente, el mismo está diseñado con Arduino.

En el siguiente artículo [8] un diseño de un sistema mecatrónico para el monitoreo de temperatura y humedad en una planta empacadora de camarón, este trabajo presento un diseño con bases de datos incluyendo una interfaz para revisar los datos que se obtienen, uno de esos datos es del sensor de temperatura y humedad DHT 22, el cual estuvo listo para entrar en la fase de pruebas.

### **C. Sensor MQ-135.**

En este artículo [9] se presenta el uso del sensor de MQ-135 para diseñar una herramienta que pueda monitorear la calidad del aire en muchos lugares utilizando un concepto basado en Internet de las cosas, el sensor de gas MQ-135 y mostrarlo en una aplicación LCD y Blynk de 16x2. El usuario puede monitorear la calidad del aire de forma remota utilizando la aplicación Blynk en el teléfono inteligente [10].

En el artículo [11] se desarrolla un sistema de monitoreo de aire utilizando micro controladores como el MQ-135, la cual permite el análisis de los datos, la cual se utiliza para medir la calidad del aire de una ubicación particular. La pantalla I2C se implementa para monitorear los datos. De hecho, con el aumento en el número de vehículos, la urbanización no planificada y el rápido crecimiento de la población.

### **D. Módulo HC-05.**

El módulo HC-05 permite transmitir datos a través de bluetooth, en el artículo [12] se diseña una Aplicación Remota Para El Control De Recicladora Extrusora De Filamento Usando Modulo Hc-05, la cual su función consiste en gestionar a distancia a través de un enlace de puerto serial, este prototipo aprovecha las herramientas que ofrece el hardware con una codificación abierta, además de controlar de manera remota con una aplicación.

### **E. Tensor Flow.**

En el artículo [13] se usa el software de Tensor Flow para diseñar varias redes neuronales y algoritmos en Deep Learning, el objetivo es la investigación y el mejoramiento de tomas de decisiones sin que un tercero lo esté supervisando en la plataforma científica Ecuencia, este trabajo dio como resultado que la plataforma científica podrá ser capaz de dar una predicción más acertada de las clasificaciones de Líneas y Sublíneas de investigación.

En el siguiente artículo [14], se desarrolla un método de clasificación de imagen usando TensorFlow, este se usa en el área de negocio de la moda, esta valida las prendas, usar una base de 15 categorías, la cual uso un modelo autónomo utilizando la librería de Google TensorFlow, para elaborar un clasificador de ropa de moda. Concluyendo después de la fase de pruebas se alcanzó un 95 % de exactitud.

### **F. Python.**

En este artículo [15] presenta un trabajo usando Python el cual se desarrolla una aplicación web con el uso del patrón de desarrollo Modelo-Plantilla-Vista (MTV, en inglés). de las primeras fases del ciclo de desarrollo de software, requerimientos y diseño. Los proyectos que se presentan son sistemas web prototipo con operaciones para crear, leer, actualizar y eliminar (CRUD, en inglés) registros de la base de datos.

### **G. Telegram.**

En el siguiente artículo [16] se utiliza Telegram como herramienta como multidisciplinario para analizar los datos, para obtener información de alerta temprana sobre amenazas cibernéticas. Con la proliferación de grupos hacktivistas que utilizan Telegram para difundir información sobre futuros ataques cibernéticos El estudio evalúa la efectividad de cada método en la detección y categorización de amenazas cibernéticas, comparando su desempeño e identificando áreas de mejora. La investigación

contribuirá a impulsar acciones orientadas a reforzar las medidas de ciberseguridad en un panorama digital cada vez más interconectado. comparar su desempeño e identificar áreas de mejora [17].

#### **H. Thingspeak.**

En el artículo [18] se implementa un sistema de monitoreo de manteniendo DESERT GECKO usando la plataforma Thingspeak, el gecko es una especie de reptil que necesita un cuidado desafiante lo cual se desarrolla un prototipo que monitoree su cuidado por lo cual realizo el desarrollo en “leopardoempleando” el algoritmo C4.5 y la plataforma Thingspeak además usar el Wifi ESP8266 para la comunicación del prototipo y la plataforma.

En este artículo [19] se presenta un proyecto de un prototipo de monitoreo de inundaciones usando la plataforma de Thingspeak para almacenar los datos, los resultados de la prueba, se encuentra que la herramienta puede funcionar leyendo los niveles de agua mediante sensores HC-SR04 y se muestra en LCD y Thinkspk De esta forma, el nivel del agua en cada presa se puede controlar en cualquier momento sin tener que ir al lugar simplemente a través del sitio web Thingspeak.

#### **I. Ecuaciones empleadas.**

##### **1. Cálculo del Índice de Variabilidad de Temperatura (IVT).**

En la ecuación 1 permite conocer que tan amplias son las fluctuaciones térmicas en el entorno donde opera el sistema inteligente. En el artículo [20] se utilizaron la ecuación 1 para poder realizar un estudio para analizar y evaluar el cambio de temperatura de los últimos 30 años y con ello poder hacer una predicción de los próximos 60 años.

$$IVT = \frac{T_{max} - T_{min}}{T_{prom}} \quad (1)$$

Donde:

- IVT: Índice de Variabilidad de Temperatura.
- Tmax: Temperatura máxima registrada.
- Tmin: Temperatura mínima registrada.
- Tprom: Temperatura promedio obtenida durante las mediciones.

## 2. Cálculo de Índice de Variabilidad de Gas (IVG).

La ecuación 2 indica que un IVG elevado existen picos de contaminación intermitentes, lo cual puede representar riesgo respiratorio si no se ventila o alerta oportunamente. En el artículo [21] usar la misma idea matemática, pero con variables diferentes, la cual se centra en la vulnerabilidad en espacios para la salud ambiental, como la contaminación en el aire, este estudio se realiza en Australia.

$$IVG = \frac{G_{max} - G_{min}}{G_{prom}} \quad (2)$$

Donde:

- IVG: Índice de Variabilidad de Gas.
- Gmax = Lectura máxima de gas (ppm).
- Gmin = Lectura mínima registrada.
- Gprom = Promedio de todas las mediciones de Gas.

## 3. Tiempo de Respuesta del Sistema (TRS).

Mide el tiempo desde que se detecta una anomalía hasta que se envíe la notificación: En el artículo [22] la ecuación 3 se lleva a cabo la misma teoría matemática para la medición de telemetría de detección de anomalías en tiempo real basada en aprendizaje automático mediante procesamiento previo de datos en granjas de servidores.

$$TRS = T_{envio} - T_{deteccion} \quad (3)$$

Donde:

- TRS: Tiempo de Respuesta del Sistema.
- Tdeteccion: La hora exacta en la que el sistema identifica una condición crítica.
- Tenvio: La hora en que se envió la notificación vía mensajería.

#### 4. Cálculo del Margen de Error.

Para determinar el margen de error general del sistema, se aplica la ecuación 4 del Error Porcentual Absoluto, que nos permite expresar el nivel de precisión del sistema. Este tipo de cálculo es útil para evaluar el rendimiento del sistema de predicción, ya que considera el error relativo a cada lectura individual. En el artículo [23] hace uso la misma idea del margen error, en este caso se le aplico para predecir los posibles tráficos que se pueden generar en la red, también conocida como redes de tráfico.

$$\text{Margen de Error (\%)} = \left( \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \right) \times 100 \quad (4)$$

Donde:

- $Y_i$  = Valor real registrado por el sensor.
- $\hat{y}_i$  = Valor de predicción del modelo.
- $N$  = Número total de simulaciones o lecturas.
- El resultado final se multiplica por 100 para expresarlo en porcentaje.

## **XI. Artículos Relacionados**

En el artículo [24] se desarrolla un sistema de alerta temprana para la prevención de inundaciones en el área del río Arauco Colombia, el sistema procesa los datos a través de un modelo que obtiene variables intervinientes, como: precipitación, nivel y caudal. Este modelo de información suministra la data al modelo matemático para el cauce del río, que se obtiene a partir de tres clases de tendencias: lineal, potencia y relaciones potenciales. El modelo del cauce se compara con un observador basado en técnicas inteligentes, redes neuronales y ANFIS.

En este estudio [25] se diseña e implementar un sistema de monitoreo de alerta temprana con el fin de prevenir el riesgo de siniestros en residencias familiares, este sistema es capaz de monitorear y alertar sobre el inicio de estos problemas, el sistema se compone de un panel basado en tecnología libres de hardware y software hechos por diferentes sensores de detección de fuego, humo y temperatura para el monitoreo de las variables.

En la investigación [26] se presenta un prototipo de alerta temprana con la tecnología de Zigbee, el cual su propósito era cautelar los desastres causados por el huaico, un fenómeno natural que consiste en el deslizamiento de una masa de lodo, piedras y agua por una pendiente, generalmente causado por lluvias intensas. El mismo fue desarrollado con la tecnología de Arduino, usando un visualizador Display GLCD. El mismo prototipo dio los resultados de un más del 90% fueron positivos o correcto, el cual representa un aporte significativo para la tecnología de alerta temprana, así mismo aplica en un prototipo para el monitoreo de pavimento destacando como ventajas el poco consumo de energía en modo transmisión, velocidad de datos y rango de comunicación.

En el artículo [27] se implementa un sistema de alerta temprana inundaciones y desbordamientos, el proyecto tuvo como nombre cable LIFE BAETULO la cual mantiene como adaptación el cambio climático liderado por AQUATEC y con presupuesto de 1.2

millones de euros, la cual fue financiada por la Comisión Europea. Con el objetivo de desarrollar la implementación de un sistema de Alerta Temprana “Integral Multirriesgo” en la ciudad de Badalona para reducir la exposición y la vulnerabilidad de los ciudadanos frente a los eventos climáticos.

Este artículo [28] presenta el diseño e implementación de una herramienta tecnológica destinada a brindar respuestas ante eventos de emergencia en una empresa de producción. El sistema operativo en un manual descriptivo el cual tendrá como objetivo dar a conocer el funcionamiento del software a través de los mandos de control y generar, por ende, un impacto positivo en temas de prevención de riesgos.

En este artículo [29] se enfoca un método para monitorear el estado de los parámetros atmosféricos de la cuenca del lago Baikal, incluido el análisis en tiempo real de datos distribuidos reales y las distribuciones predictivas a largo plazo más probables de temperaturas superficiales, precipitación, presión, y geopotencial con un plazo de entrega de hasta 9–12 meses. El desarrollo del sistema de análisis de información GeoGIPSAR realizó el análisis en tiempo real de datos espaciales y puntuales mediante diversos métodos de procesamiento y obtener estimaciones pronosticas a largo plazo de la entrada de agua al lago.

## XII. Metodología

El " Desarrollo de un sistema inteligente de alerta temprana para monitoreo de variable críticas de salud con notificación vía mensajería en cabinas insonorizadas" se desarrolla empleando un método que combina técnicas de experimentación, simulación y aprendizaje automático, en la figura 1 se presenta un esquema del proceso de desarrollo del proyecto.

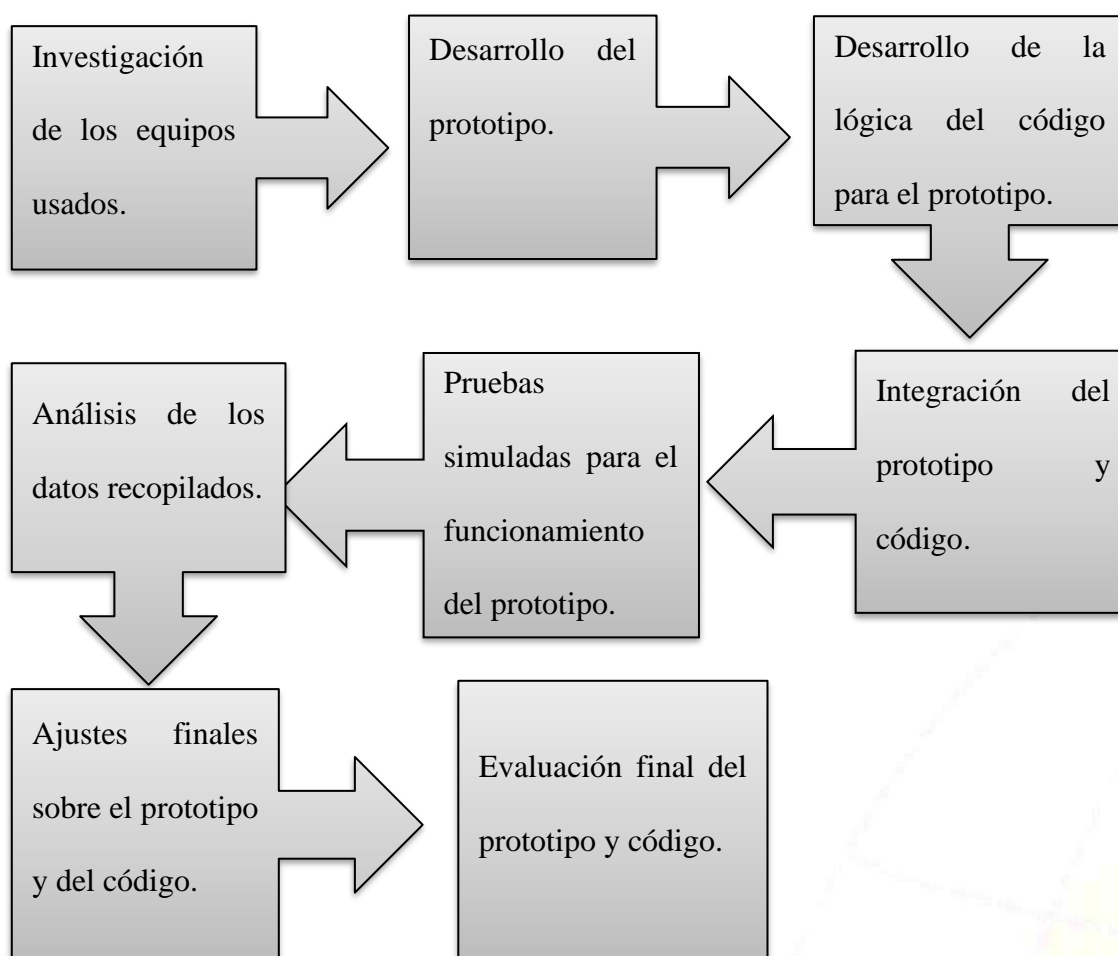


Figura 1 - Esquema del desarrollo del proyecto.

Durante la etapa inicial, se diseña un prototipo utilizando Arduino, un sensor DHT22 de temperatura y humedad, sensor MQ-135 para la calidad de aire y vinculados a un protoboard. La lógica del sistema se elabora en Python, incorporando algoritmos de que pueden examinar datos en tiempo real, y además prediga situaciones de riesgo y pueda enviar alertas a través de canales de mensajería.

El sistema se centra en registrar parámetros del entorno (temperatura, humedad y calidad de aire) y, basándose en los datos registrados, se envía automáticamente una alerta con notificación vía mensajería. Por ejemplo, mantiene un rango establecido para cada variante, para la temperatura se establece un rango de 24°C a 18°C y para los niveles de oxígeno se le indica que sea mayor a 65 ppm y a la variable de humedad de 30% a 75% [30], si unos de estas variables se salen del rango establecidos se produce una señal que enviara la alerta cuando los parámetros ya no se encuentren en los límites antes mencionado.

Se diseñó un modelo de simulación para anticipar el comportamiento del sistema en diversas circunstancias ambientales, lo que facilita la modificación de parámetros esenciales y aumenta la exactitud de las proyecciones.

**a) Diseñar del código para el prototipo físico.**

Para poder configurar el prototipo físico se hace uso de la plataforma de Arduino IDE, primero se configuro el módulo bluetooth, asignándole un nombre y una clave de acceso, Luego se configura los dos sensores indicando que pin usaran en el Arduino UNO, además de indicar de qué forma se expresan los datos y por último se configuro el puerto de salida de los mismo a 9600 baudio. En código hecho en Arduino IDE se visualiza en el anexo 1.

**b) Diseñar del Bot de telegram y del Dashboard en Thingspeak.**

Para diseñar la llegada de alertas se hace uso la plataforma telegram dado que tiene la facilidad de poder crear Bots de simple y fácil. Para configurar el Bot se inicia un chat con el creador de Bots de telegram llamado BotFather, una vez iniciada la conversación se pone el comando “/NewBot”, luego ponemos el nombre del Bot que se crea, en este caso “Sistema Autonomo” y finalmente ponemos el username el cual es AutoSistemBot, hechos todos los pasos el BotFather nos da el token para poder enviar las alertas y el link

de nuestro Bot que es “t.me/AutoSistemBot.”. La visualización de la llegada de los mensajes se encuentra en el anexo 3.

Para el Dashboard se configura en la plataforma Thingspeak de la empresa The Mathworks, esta herramienta permite diseñar de forma fácil un Dashboard, lo primero es crear un canal llamado “Monitoreo”, lo siguiente es configurar los campos, en este caso son tres: temperatura, humedad y calidad de aire ppm, hechos los canales se le añade a la página visualizaciones donde se establecen cada una de las gráficas ya configuradas por el sistema, luego se va al canal de configuración para elegir cuantos campos se visualizan en la página de las gráficas finalmente se llega a la API Key para copiar el código de acceso y colocarlo en código de Python para la llegada de los datos a la nube de la web, y el Dashboard se visualiza en el anexo 3.

#### **c) Diseñar del código en Python con biblioteca de TensorFlow.**

El diseño del código se realiza en Python con ayuda de la herramienta de Tensorflow, el sistema se divide en tres modelos para mejorar la eficiencia de la llegada de los datos en tiempo real, velocidad y precisión en las predicciones y él envió rápido de las alertes de peligro. Pero antes de estructurar los modelos, se configura la detención de usuarios que iniciaron el Bot de telegram para la confirmación de obtención de los chats ID de cada usuario, además de la configuración de la interfaz del monitor principal.

Para la estructura del primer modelo es de adquirir los datos provenientes del Arduino UNO por el módulo bluetooth este se conecta al puerto serial “COM 7”, este se encarga de leer los datos de temperatura, humedad y nivel de oxígeno ppm las cuales son almacenada de forma segura en el sistema., este puerto puede leer los datos a una velocidad de 9600 baudios. Se configura si detecta que uno de los datos no es correcto lo menciona como “nan” (Dato nulo) se convierten a float y se guardan con protección de lock para evitar conflictos y los validos se almacena y se envían a los demás modelos.

La estructura del segundo modelo es la más importante dado que esta predice un futuro acontecimiento, para ello se hace uso de la red neuronal de TensorFlow para realizar las antes mencionada, esta parte del código se encarga de predecir si los rangos de temperatura, humedad y niveles de oxígeno ppm se salen del límite, este proceso lo hace con el modelo LSTM (Long Short-Term Memory) de TensorFlow este recibe los datos almacenados y entrena modelo con los últimos veinte valores, en una línea del código se configuro que se usen los primeros diez datos para predecir el siguiente valor, el cual se transforma de vuelta a su valor original. Esto se repite para cada variable, devolviendo las tres predicciones. Y los datos de predicción se almacena en el sistema y se le envía a la línea de código del tercer modelo.

El último modelo se encarga de enviar las notificaciones de alerta y predicción al Bot de telegram, además de enviar los datos al Thingspeak y lo pueda almacenar en la nube. Para que el modelo considere enviar una notificación se le configuro rangos de envío, para el de temperatura se indica que debe ser, si está por debajo de 20°C o si está por encima de 30°C se envía la notificación al Bot, así mismo se le configura a la humedad con menor a 30% y mayor 75% se envíe la alerta si pasa esos y para el nivel de oxígeno ppm se establece, si está por debajo de 65ppm se enviara la notificación. Una de las ultimas de las configuraciones que se añade al código es el envío de un resumen de todos los datos en ese periodo, él puede configurarse como el usuario lo prefiera. Todo el código se puede visualizar en el anexo 2.

#### **d) Esquema de prototipo.**

Finalmente, se evalúa el prototipo en un ambiente controlado, comprobando su capacidad para manejar datos, producir advertencias y notificar alarmas vía mensajería de acuerdo a los valores críticos fijados. Esta metodología asegura que el sistema pueda ser escalado

y ajustado a diversas aplicaciones futuras.

En la figura 2 se observa un esquema de cómo funciona la llegada de los datos del prototipo.

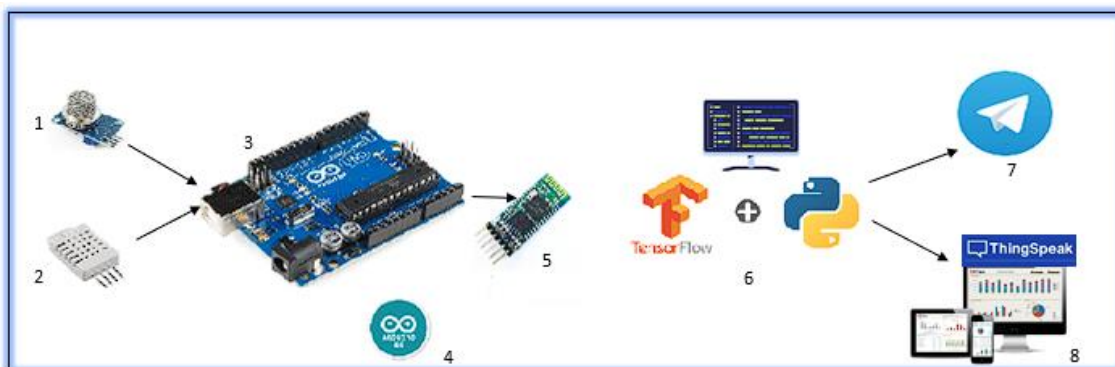


Figura 2 - Esquema de prototipo.

Componentes y aplicaciones:

1. Sensor MQ 135.
2. Sensor DHT22.
3. Arduino Uno.
4. Arduino IDE.
5. Módulo HC-05 (Bluetooth).
6. Python - Tensor Flow (Monitor principal).
7. Telegram.
8. Thingspeak.

Prototipo montado se muestra en la figura 3 con sus respectivas conexiones que se hizo al Arduino Uno, Al ejecutar el programa, el sensor DHT22 y MQ-135 captura en tiempo real los valores de temperatura, humedad y calidad de aire, los cuales son procesados por el Arduino Uno y enviados por el módulo HC-05(Bluetooth) a un código estructurada en Python con TensorFlow haciendo que el programa sea el monitor principal. En función de estos datos, el sistema analiza los parámetros y activa la alarma en caso de superar los rangos limitados dentro del modelo autónomo, los datos se envían al Thingspeak para que

el usuario pueda revisar en cualquier lugar los parámetros que llegan del modelo, y si los datos llegan al rango limite se envía una alerta al canal de Telegram.



Figura 3 - Prototipo físico.

Para realizar las pruebas se diseña un código en Python, como se menciona anteriormente, este aplica varias formas de trabajo para ver los datos de temperatura, humedad y calidad de aire en tiempo real, realizar predicciones con el modelo de TensorFlow o machine learning (aprendizaje automático) y con una interfaz gráfica para ver los datos. Para la comunicación se hace uso del puerto serial bluetooth, en nuestro caso sería el COM7 registrado en el equipo, el cual recibir los datos del HC-05 los cuales son de temperatura y de aire.

La red neuronal del TensorFlow crea y entrena de forma continua este como propósito analizar los datos enviados por el módulo bluetooth, usar diferentes funciones que son de recibir datos y enviar comandos por el mismo medio por el cual recibir los datos, además lleva una tercera función la cual es de predicción, esta compara datos y valores de temperatura, humedad y calidad de aire para que la misma pueda tomar decisiones y determinar la acción que se envía al dispositivo.

#### e) **Resultado del prototipo.**

En la tabla 5 la cual se encuentra en el anexo 4, se realiza la prueba de funcionamiento del prototipo, durante un periodo de dos horas y media a partir de la 8:30 AM hasta las 11:00 AM, el objetivo de la prueba es ver cómo funciona el prototipo en una jornada

laboral corta, la prueba se realiza en la sala musical (cabina insonorizada) esta se visualiza en el anexo 6, la cual se encuentra en la Universidad Politécnica Salesiana Sede Guayaquil en el edificio D, la misma tiene una altura de 2.50 m, un largo de 5.48 m y una ancho de 3.35 m. Está equipada con dos aires acondicionados, una salida de ventilación, además cuenta con dos puertas, una va sala de monitoreo y la otra es la salida. En la tabla 5 se añadieron 100 datos durante esta prueba.

En la figura 4 se visualiza que durante la prueba no se presentó ninguna variante peligrosa en este periodo de prueba solo una muy pequeña variable en los niveles de oxígeno ppm de 99.5 siendo la más baja, la temperatura se mantuvo variando durante la prueba teniendo un pico mínimo de 19.8°C y un pico máximo de 21.5°C y en cambio la humedad fue más variable, dado que llevo a un pico alto de 81.3% humedad y un mínimo de 69.0%.

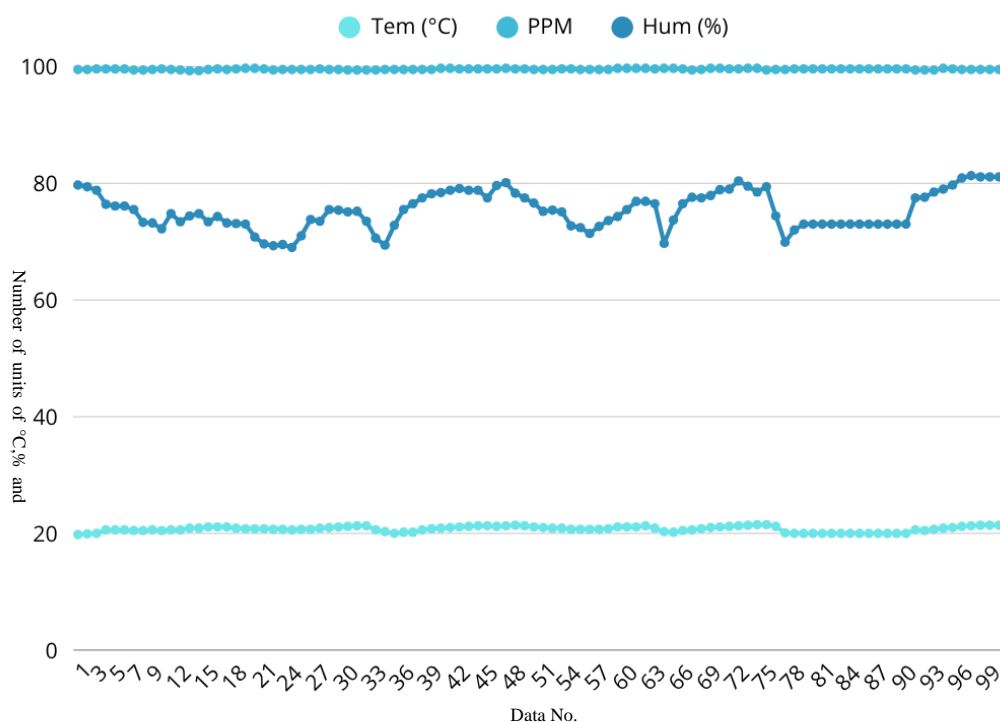


Figura 4 - Gráfica de los datos tomados de la tabla 5.

En la tabla 6 la cual se encuentra en el anexo 4, se presenta la prueba de notificación de la alta temperaturas, la misma se lleva a cabo en la sala musical, se le acerca al sensor DHT-22 una vela encendida durante un periodo de una hora, la cual duro de las 11 AM hasta las 12PM, la vela estuvo a una distancia de 3 cm, esta prueba se realiza para aumentar la temperatura y ver sus cambios, además se confirma el envío de notificación de predicción y alerta al chat de telegram. Los rangos que se escogieron son: si la temperatura es igual o mayor a 23°C.

En la figura 5 se observa cómo va aumentando la temperatura considerablemente comienza a subir de 21.5 °C teniendo un pico máximo de 29.2 °C siendo su tope final en esta prueba, además se visualiza una disminución en el porcentaje de humedad.

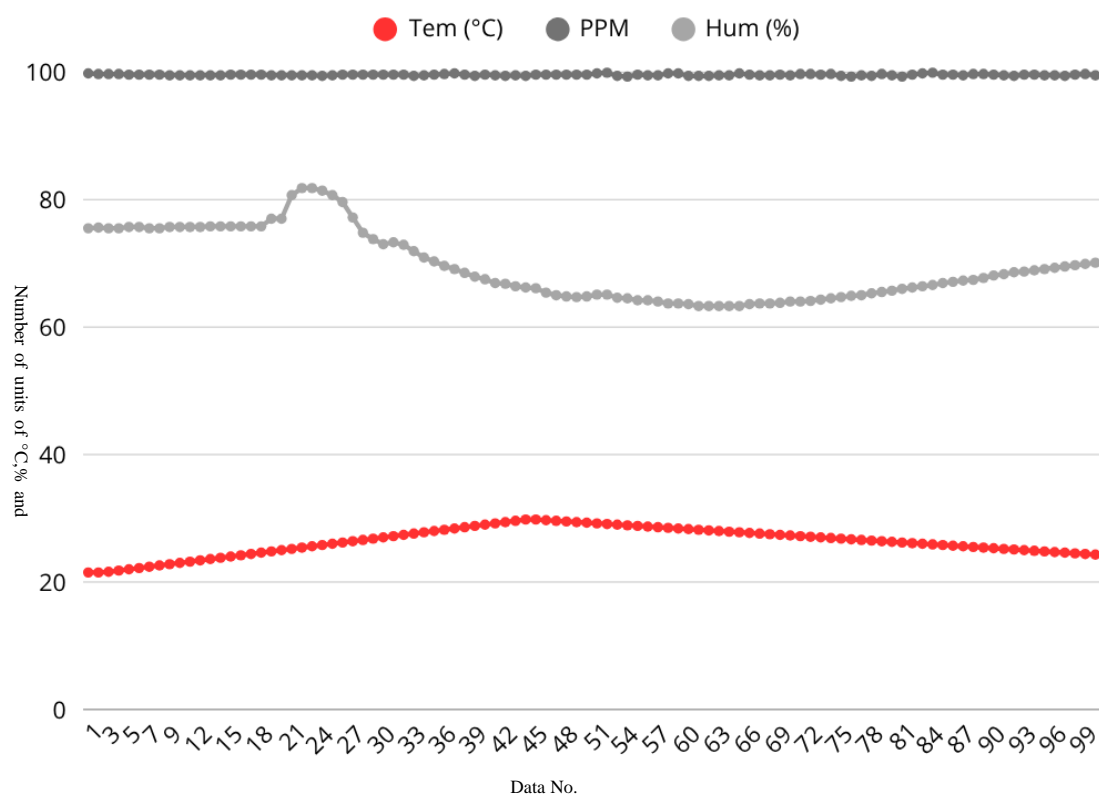


Figura 5 - Gráfica de los datos tomados de la tabla 6.

En la tabla 7 la cual se encuentra en el anexo 4, se realiza la prueba de notificación con temperaturas bajas, se lleva a cabo en la sala música, para realizar la misma se llena un recipiente plástico con agua y se coloca en el congelador, una vez congelado se le acerca al sensor DHT-22 a una distancia de 0.9cm y se visualiza la disminución de la temperatura, la prueba dure una hora desde las 9AM hasta las 10AM. Para que llegue una notificación al chat de telegram se establece el rango de igual o menor a 18°C.

En la figura 6 se visualiza como los niveles de temperatura van disminuyendo, de 21.5°C hasta llegar a un pico mínimo de 14.9°C, además de observa que los niveles de porcentaje humedad aumentaron durante esta prueba con un pico máximo de 81.1% hasta llegar a 18.6°C, llegada a esa temperatura la húmeda comienza a disminuir a un punto pico mínimo de 57%.

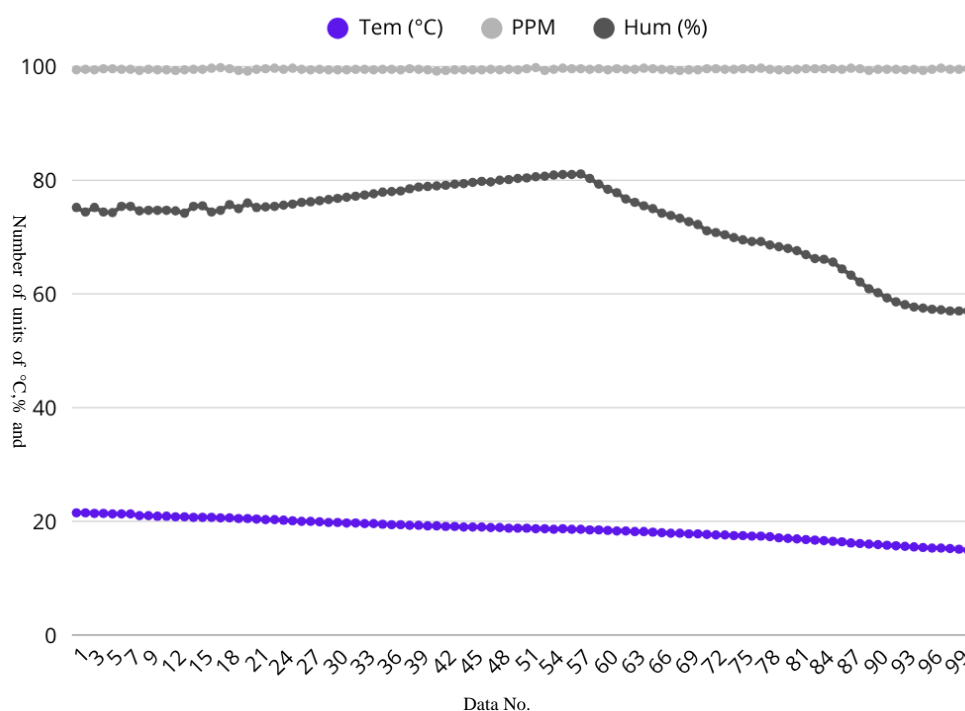


Figura 6 - Gráfica de los datos tomados de la tabla 7.

En la tabla 8 la cual se encuentra en el anexo 4, se añade los datos de la prueba de notificación de humedad, la misma se realiza en la sala música, para realizar prueba se le coloca un pañuelo húmedo cerca del sensor DHT-22 este proceso se aplica periódicamente, el mismo se acercaba cada 10 minutos aproximadamente, la misma duro un periodo de una hora desde las 12PM hasta las 13PM, la notificación de la misma llegaría si el rango sube con un porcentaje mayor a 75%.

En la figura 7 se observa cómo los niveles de humedad comienzan en 75%, este mismo suben lentamente, pero muestra un aumento considerable, la línea no directamente recta, pero sube gradualmente, indicando un incremento lento pero sostenido. Llegando a un pico máximo de 89.8%.

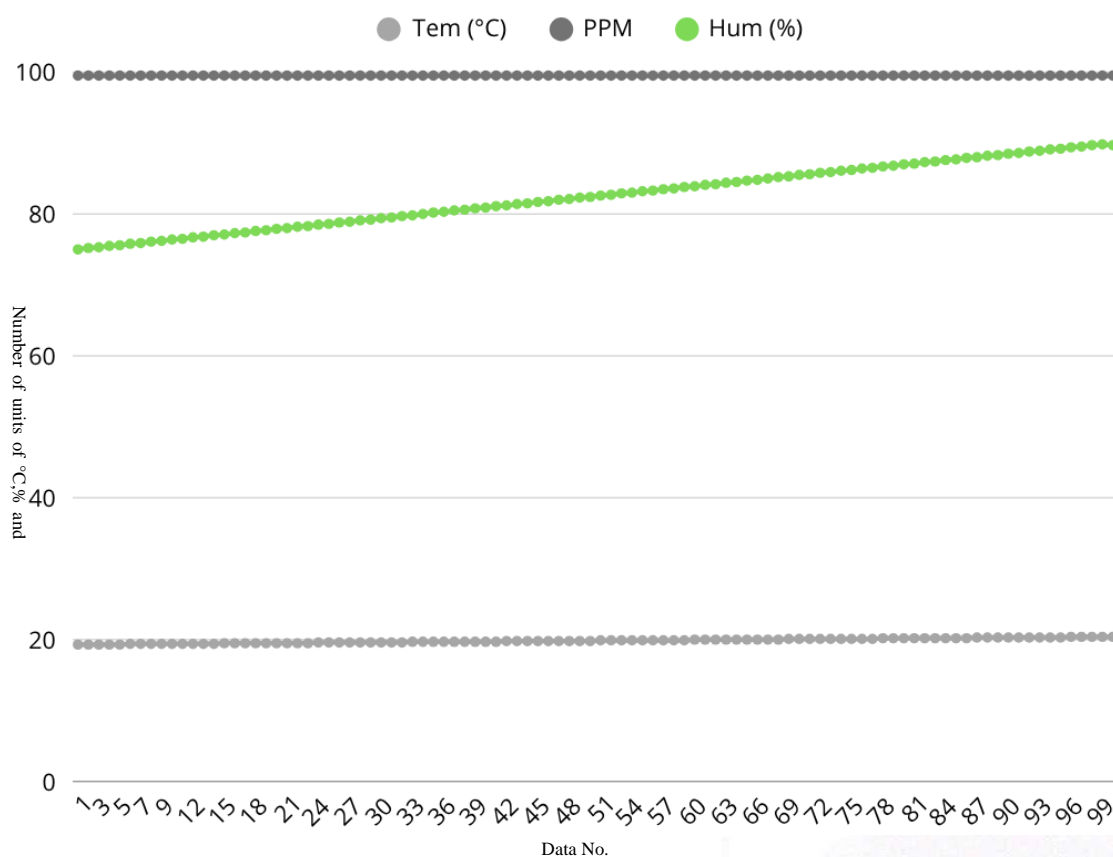


Figura 7 - Gráfica de los datos tomados de la tabla 8.

En la tabla 9 la cual se encuentra en el anexo 4, se lleva a cabo la prueba de medición de niveles de oxígeno en ppm (Partícula por millón), la cual duro cuarto minutos y se realiza acercando una fosforera que contiene gas butano, acercándolo al sensor MQ-135, el misma detecta que hay oxígeno, una vez acercado la fosforera el sensor no detectara el oxígeno, sino el butano llegando a bajar los niveles de oxígeno en ppm, lo cual llegara una notificación al chat de telegram si los niveles son menores a 65 ppm.

En la figura 8 se visualiza una curva descendente, lo cual indica que los niveles de oxígeno están disminuyendo rápidamente dado que los datos se recopilan cada 3 segundos, cada vez que se le acerca el butano los niveles bajan a un pico mínimo de 3.8ppm, ya alejado y con ambiente limpio, se ve que los niveles vuelven a subir y se estabiliza.

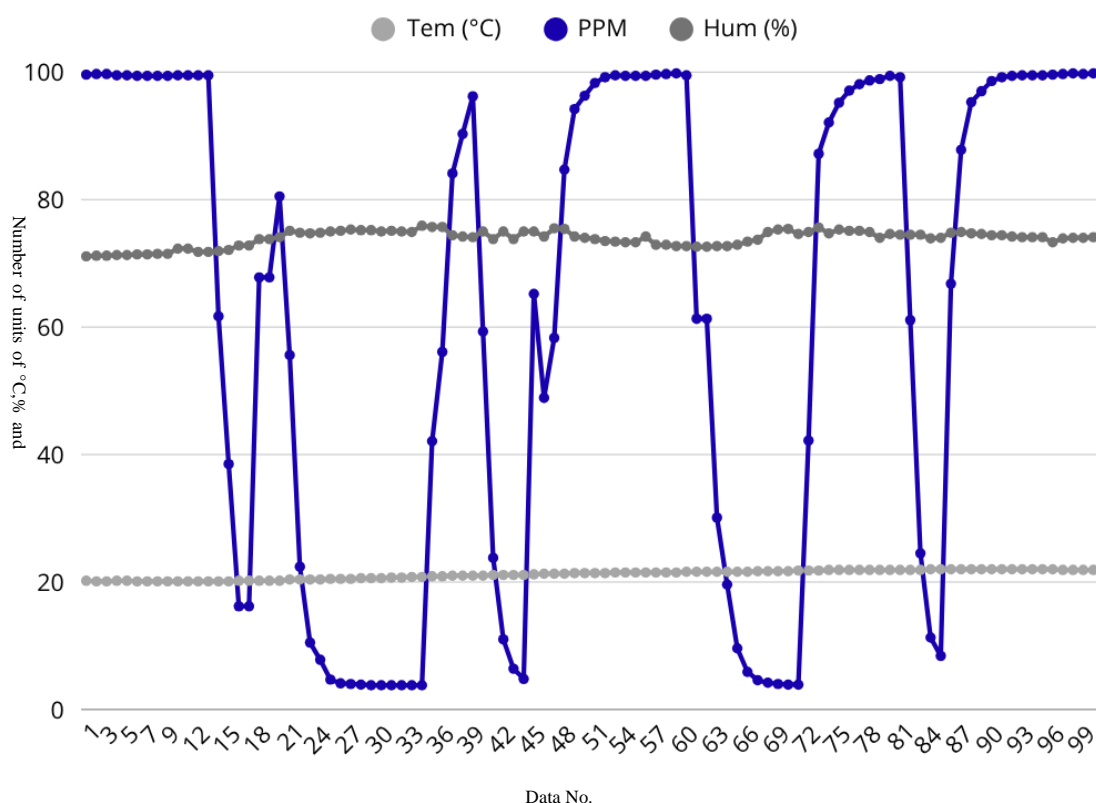


Figura 8 - Gráfica de los datos tomados de la tabla 9.

### XIII. Interpretación de resultados

- **Cálculo del Índice de Variabilidad de Temperatura (IVT).**

Se analiza la temperatura medida en 100 simulaciones realizadas por el sistema con la ecuación 1. De esas simulaciones se obtuvieron los siguientes valores:

Temperatura máxima = 29.3 °C.

Temperatura mínima = 14.9°C.

Temperatura promedio = 20.6 °C.

$$IVT = \frac{T_{max} - T_{min}}{T_{prom}} = \frac{29.3 - (14.9)}{20.6} = 0.69$$

Este resultado indica una alta variabilidad térmica en el entorno estable. Es decir, el sistema opera en un rango amplio de temperaturas, lo que requiere una calibración robusta para mantener la efectividad de la alerta en el sistema.

- **Cálculo del Índice de Variabilidad de Gas (IVG).**

En la misma serie de simulaciones, se analizaron los niveles de gases nocivos en ppm con la ecuación 2, con los siguientes resultados:

PPM máxima = 99.7 ppm.

PPM mínima = 3.8 ppm.

Promedio = 57.86 ppm.

$$IVG = \frac{G_{max} - G_{min}}{G_{prom}} = \frac{99.7 - 3.8}{57.86} = 1.66$$

Este resultado muestra una alta dispersión en los niveles de gases registrados, lo cual valida la necesidad de un sistema de notificación proactiva ante elevaciones repentinas en la concentración.

- **Cálculo del Tiempo de Respuesta del Sistema (TRS).**

El tiempo de respuesta del sistema evalúa cuantos segundos transcurren desde que el sistema detecta una condición crítica hasta que emite la notificación automática por

mensajería la cual se calcula con la ecuación 3. Esto nos permite validar la eficiencia del sistema de alerta en situaciones donde la velocidad es crucial especialmente en espacios cerrados como cabinas insonorizadas.

$$TRS = T_{envio} - T_{deteccion}$$

Durante las pruebas realizadas se registra un promedio de detección a las 10:00:05 y el envío del mensaje automático se ejecuta a las 10:00:09. Aplicando la ecuación:

$$TRS = 10:00:09 - 10:00:05 = 4 \text{ Segundos}$$

- **Cálculo del Margen de Error.**

Se toman los 100 datos de cada tabla para la medición del margen de error, para ello, se calcula la diferencia absoluta entre cada valor y el promedio correspondiente, por ejemplo, 20.60°C para temperatura, y luego se sumaron todos estos errores, después se divide esta suma entre el número total de muestras en este caso serían 100 así se obtiene el MAE para cada variable, primera es 1.85 °C para temperatura, la segunda es 1.69 % para humedad y por último 0.03 ppm para calidad de aire. Estos valores reflejan cuán alejadas están las mediciones individuales del promedio, y permiten calcular luego el margen de error general del sistema.

$$MAE_T = \frac{\sum_{i=1}^{100} |T_i - 20.60|}{100} = 1.85^\circ\text{C}$$

$$\text{Margen deError} = \frac{1.85}{20.60} * 100 = 8.98\%$$

$$MAE_H = \frac{\sum_{i=1}^{100} |T_i - 74.40|}{100} = 1.69\%$$

$$\text{Margen deError} = \frac{1.69}{74.40} * 100 = 2.27\%$$

$$MAE_A = \frac{\sum_{i=1}^{100} |T_i - 29.24|}{100} = 0.03\text{ppm}$$

$$\text{Margen deError} = \frac{0.03}{29.24} * 100 = 0.10\%$$

Para obtener un valor único representativo del desempeño general del sistema, se promedia el error porcentual de las tres variables:

$$\text{Margen de Error General} = \frac{8.98 + 2.27 + 0.10}{3} = 3.78\%$$

Sacando un margen de error del todo el sistema del 3.78 %, por lo cual es un comportamiento altamente confiable, dado que las tres variables críticas se encuentran dentro de un margen de precisión aceptable, validando así la eficacia del modelo predictivo y su aplicabilidad en entornos reales.

- **Validación de los datos del prototipo respecto a un dispositivo comercial.**

Para validar la precisión del sensor de temperatura utilizado en el sistema, se realizó una comparación con una pistola de temperatura infrarroja comercial la cual se observa en la tabla 1 y en la tabla 2. Esta comparación se hizo en los escenarios de las pruebas de temperatura alta y baja. La tabla 10 y 11 completa se encuentra en el anexo 5.

*Tabla 1 Resumen de comparación de Mediciones de Temperatura alta.*

N° Dato	Hora	Temperatura Sistema (°C)	Temperatura Pistola Infrarroja (°C)	Diferencia (°C)	Error % Aprox
1	11:11:00	21.5	20.1	1.4	6.97
10	11:16:24	23	21.6	1.4	6.48
20	11:22:24	25	23.4	1.6	6.84
30	11:28:24	27	25.2	1.8	7.14
40	11:34:24	29	27.2	1.8	6.62
50	11:40:24	29.3	27.4	1.9	6.93
60	11:46:24	28.3	26.4	1.9	7.20
70	11:52:24	27.3	25.4	1.9	7.48
80	11:58:24	26.3	24.4	1.9	7.79
90	12:04:24	25.3	23.4	1.9	8.12
100	12:10:24	24.3	22.4	1.9	8.48

Tabla 2 Resumen de comparación de Mediciones de Temperatura baja.

N° Dato	Hora	Temp Sistema (°C)	Temp IR (°C)	Pistola (°C)	Diferencia (°C)	Error Aprox %
1	9:04:33	21.5	21.6		0.1	0.47
10	9:09:21	20.9	21		0.1	0.48
20	9:14:41	20.5	21		0.5	2.44
30	9:20:01	19.8	19.9		0.1	0.51
40	9:25:21	19.2	19		-0.2	1.04
50	9:30:41	18.8	17.5		-1.3	6.91
60	9:36:01	18.4	19.7		1.3	7.07
70	9:41:21	17.8	19.6		1.8	10.11
80	9:46:41	17	17.1		0.1	0.59
90	9:52:01	15.9	16.1		0.2	1.26
100	9:57:21	14.9	16.9		2	13.42

La diferencia promedio entre el prototipo y la pistola infrarroja fue de aproximadamente 1.9°C, lo que representa un error promedio del 7.48%, lo cual está dentro del rango aceptable para aplicaciones de alerta temprana. Esto demuestra que el sistema, aunque esta implementado con sensores más económicos presenta un comportamiento muy similar al de un dispositivo comercial de lectura instantánea, lo que respalda su validez técnica.

Para validar la precisión del sensor de humedad, se realiza pruebas la cuales están en la una tabla 3, para observar la comparativa entre la humedad relativa medida por el sensor DHT22 y la medida por el HTC-2 Termohigrómetro el cual es un instrumento multifuncional utilizado para medir y monitorear las condiciones ambientales. La tabla 12 completa se visualiza en el anexo 5.

Tabla 3 Resumen de comparación de Mediciones de Humedad.

Nº Dato	Hora	Humedad DHT22 (%)	Humedad HTC-2 (%)	Diferencia (%)	Error Aprox. %
1	12:15:23	75	75.6	-0.6	0.79
10	12:20:48	76.4	77	-0.6	0.81
20	12:26:48	77.9	76.9	1	1.36
30	12:32:48	79.4	79.5	-0.1	0.08
40	12:38:48	80.9	81.7	-0.8	0.95
50	12:44:48	82.4	82.6	-0.2	0.26
60	12:50:48	83.9	83.9	0	0.05
70	12:56:48	85.5	85.2	0.3	0.30
80	13:02:48	87	87.5	-0.5	0.55
90	13:08:48	88.5	88.4	0.1	0.16
100	13:14:48	89.7	89.4	0.3	0.29

La comparación entre los sensores MQ-135 y MQ-2 se observa en la tabla 4, evidencia diferencias significativas en sus curvas de respuesta frente a la exposición al gas butano. Mientras que el MQ-2 presenta una sensibilidad mucho más elevada y reacciona con aumentos bruscos en la concentración de PPM, el MQ-135 responde de forma más gradual, detectando cambios en la calidad general del aire. La tabla 13 completa se visualiza en el anexo 5.

Tabla 4 Resumen de comparación de Mediciones MQ-135 y MQ-2.

Nº Dato	Hora	PPM MQ-135	PPM MQ-2	Diferencia (%)	Error% Aprox
1	11:04:02	99.6	98.6	1	1
9	11:04:10	99.4	96.7	2.7	2.72
17	11:04:26	16.2	14.9	1.3	8.02
25	11:05:10	4.7	4.2	0.5	10.64
34	11:05:47	3.8	3.4	0.4	10.53
44	11:06:26	4.8	4.3	0.5	10.42
66	11:07:35	5.9	5.3	0.6	10.17
67	11:07:39	4.6	4.1	0.5	10.87
70	11:07:53	3.9	3.5	0.4	10.26
90	11:08:51	98.6	96.8	1.8	1.83
100	11:09:22	99.8	98.5	1.3	1.30

Se muestra una diferencia porcentual entre ambos sensores, que se mantuvo dentro de un margen general de 10.53%. El resultado demuestra que el sistema, incluso utilizando el sensor MQ-135, presenta un comportamiento altamente comparable al del sensor MQ-2, lo que valida su utilidad y precisión para aplicaciones donde la concentración de PPM es crítica. La minimización de esta diferencia subraya la fiabilidad técnica del sistema para proporcionar datos consistentes.

#### **XIV. Discusión**

En este artículo [31] se implementa un análisis de cambios en los signos vitales son un indicador importante del deterioro fisiológico y brindan oportunidades para el reconocimiento y la intervención tempranos. La cual realizaron un sistema inteligente de alerta temprana la cual explora las tecnologías de IoT para proporcionar un sistema EWS en situaciones hospitalarias inteligentes. La solución propuesta presenta una configuración adaptativa del proceso de monitorización de los signos vitales en función de la variación del estado de salud del paciente y de las decisiones del personal médico.

En cambio, el trabajo presentado se enfoca en diseñar un modelo que presenta la integración de TensorFlow para analizar patrones históricos y estacionales para mejorar el sistema de alerta temprana, esto garantiza una mayor precisión en la predicción de eventos críticos. Además, lo hacen adaptable a una amplia variedad de contextos, por lo cual, este trabajo no solo es más autónomo, sino también más versátil y eficiente, ofreciendo soluciones preventivas proactivas y multifuncionales. Además de almacenar datos en la nube de Thingspeak.

En este artículo [32] se desarrolla un estudio para el monitoreo y alerta temprana en para pacientes que reciben atención médica domiciliaria. El sistema se implementa en un centro médico y su agencia de salud domiciliaria afiliada. De los 28 participantes, 27 completaron el estudio, mientras que uno abandona prematuramente debido a presión arterial baja y dificultad para respirar. En general, 24 de los 28 (86%) participantes informaron estar satisfechos con el sistema.

En cambio, el proyecto presente se centra en las variables críticas de salud en cabinas insonorizadas, esta se monitorea en un programa el cual envía notificación mediante mensajería y guarda la información en la web Thingspeak, las notificaciones que llegan son dos, una de predicción de una posible situación peligrosa y otra de cual ya está

ocurriendo.

En este artículo [33] se realiza un monitoreo de la calidad del aire exterior a largo plazo" monitorea la calidad del aire en tiempo real y transmite datos a través de un LoRaWAN red a una plataforma pública de IoT. monitoreo de la calidad del aire exterior a largo plazo" monitorea la calidad del aire en tiempo real y transmite datos a través de un LoRaWAN red a una plataforma pública de IoT. Los datos llegan a una red llamada CasaHabla, y se muestra en un panel web. Además, está sincronizado con el Virtuoso Aplicación para teléfono inteligente para monitoreo móvil.

En el proyecto presente realizo algo parecido, solo no en el exterior sino en el interior de una cabina insonorizadas, el cual maneja la calidad del aire a tiempo real, pero el nuestro tiene un código que monitorea y alerta de las situaciones peligrosas que pueden ocurrir además de añadir en el sistema él envió de un resumen de todos los datos de un periodo de 8 horas laborales, los mismo se almacenan en Thingspeak.

En el artículo [34] se desarrolla un monitoreo vital en varios entornos de salud el cual crea un entorno en el que el médico y el paciente pueden participar. El médico también puede estar al día del estado de salud actual del paciente. La inicialización del sensor es la primera etapa de este sistema. El mismo de conecta Raspberry Pi 3 Modelo B+, que transmite una señal a través de Internet que Este enfoque puede monitorear a varios pacientes en condiciones graves y ahorrar tiempo. Este proyecto proporciona monitoreo de bajo costo desde el hogar.

En cambio, este proyecto utiliza el Arduino UNO el cual es más accesible y permite añadir sensores conectado al mismo, el mismo no necesita una conexión a internet, dado que los datos se envían al código de Python mediante bluetooth, lo cual puede hacer que el monitoreo tenga más rapidez al hacer los envíos de datos.

En el artículo [35] se realiza un prototipo de monitoreo de la salud del paciente el cual se

desarrolla con el ESP8266 y Arduino con plataforma IoT, el propósito de este proyecto proporcionar una solución a los pacientes, especialmente a los ancianos, que se enfrentan al riesgo de muerte inesperada debido a la falta de atención médica. El sistema propuesto utiliza un sensor de latidos del corazón y un sensor de temperatura IR infrarrojo conectados a Arduino UNO y Nodemcu, respectivamente, para monitorear los signos vitales del paciente.

En el proyecto presente se desarrolla la transmisión de datos con la tecnología de Arduino uno y el módulo bluetooth HC-05 el cual proporciona un envío de dato eficientemente al sistema autónomo y que pueda realizar el monitoreo de manera rápida y a tiempo real, lo cual es necesario dado que recopilan datos de importes de salud en interiores como las cabinas insonorizadas.

## **XV. Conclusiones**

La implementación de un sistema electrónico de alerta temprana con notificación vía mensajería es efectiva en anticipar emergencias, dado que, mejorará la precisión de las notificaciones y el tiempo de respuesta.

La identificación de los componentes y del programa fue clave para el desarrollo del prototipo, puesto que, esto garantizo una recolección eficiente de datos y se llegó a obtener una base para que el modelo funcione de manera efectiva y una respuesta rápida ante condiciones anómalas.

Desarrollar del código con programa hecho en Python con la biblioteca de TensorFlow, se determinó que diseñar un sistema que monitorea continuamente variables críticas como temperatura, humedad y calidad del aire, integrando un modelo predictivo que anticipa eventos críticos con precisión resulta clave para mejorar la prevención y la respuesta ante posibles riesgos.

Las pruebas realizadas mostraron márgenes de error bajos, en el de temperatura se obtuvo 8.98% de error, en la humedad es del 2.27% y en los de niveles de oxígeno de 0.10% lo que confirma la fiabilidad del sistema para emitir alertas en tiempo real. Por lo tanto, podemos afirmar que le prototipo es viable para su aplicación en ambientes controlados, ofreciendo una herramienta eficaz para la prevención temprana de riesgos relacionados con condiciones ambientales.

## **XVI. Recomendaciones**

Se sugiere verificar constantemente las conexiones de los sensores y el Arduino, ya que durante el ensamblaje del prototipo tuvimos varios errores en las lecturas por cables flojos o falsos contactos. Esto evito que el sistema respondiera correctamente en algunas pruebas.

Se propone dedicar más tiempo a la etapa de pruebas individuales y verificación de cada componente del prototipo, ya que durante la integración se presentaron problemas con la comunicación entre sensores y módulos que dificultaron el avance. Realizar estas pruebas ayudara a identificar fallas tempranas y optimizar el desarrollo del sistema.

Se aconseja realizar un mantenimiento periódico de los sensores y del módulo de conectividad HC-05 para garantizar que se comunican de forma eficaz en el sistema para realizar predicciones en consecuencia. Esto permitirá seguir garantizando que el sistema emita alertas tempranas de alta precisión.

Se sugiere incorporar un sistema GPS y un módulo GSM para mejorar el dispositivo y adaptarlo a otros proyectos, lo cual permitirá el envío de alertas con ubicación por mensaje de texto, incluso en lugares donde no hay Wi-Fi y Bluetooth. También se sugiere añadir una tarjeta SD para guardar los datos cuando no haya conexión, lo que haría que el sistema sea útil en zonas rurales, vehículos o situaciones de emergencia fuera de las cabinas.

## XVII. Bibliografía

- [1] J. D. Pabón-Caicedo, «El cambio climático en la América ecuatorial», *Ikara. Revista de Geografías iberoamericanas*, n.º 1, may 2022, doi: 10.18239/Ikara.3019.
- [2] J. P. P. Rajakumar y J. Choi, «Helmet-Mounted Real-Time Toxic Gas Monitoring and Prevention System for Workers in Confined Places», *Sensors*, vol. 23, n.º 3, p. 1590, ene. 2023, doi: 10.3390/s23031590.
- [3] L. M. Easterline, A. A.-Z. R. Putri, P. S. Atmaja, A. L. Dewi, y A. Prasetyo, «Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32», *Procedia Computer Science*, vol. 245, pp. 815-824, ene. 2024, doi: 10.1016/j.procs.2024.10.308.
- [4] R. Moran-Borbor, V. Galvis-Roballo, J. Niño-Vega, y F. Fernández-Morales, «Desarrollo de un robot sumo como material educativo orientado a la enseñanza de programación en Arduino», *Revista Habitus: Semilleros de investigación*, vol. 1, n.º 2, pp. e12178-e12178, ago. 2021, doi: 10.19053/22158391.12178.
- [5] J. M. Saldaña-Acosta, «Desarrollo de Actitudes de Colaboración, Cooperación e Innovadoras en el Aula Basado en Proyectos», *Revista Docentes 2.0*, vol. 11, n.º 1, pp. 130-139, abr. 2021, doi: 10.37843/rtded.v11i1.201.
- [6] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, y S. Vappangi, «A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations», *Computer Science Review*, vol. 40, p. 100364, may 2021, doi: 10.1016/j.cosrev.2021.100364.
- [7] J. L. H. Villar, «Diseño y construcción de una mini-estación autónoma, con estudiantes de la Facultad Ciencias Físico y Matemáticas-UNSCH», *Revista de Investigación Científica de la UNF – Aypate*, vol. 2, n.º 2, pp. 25-34, 2023, doi: 10.57063/ricay.v2i2.42.
- [8] F. Daru, «Temperature and Humidity Monitoring Using DHT22 Sensor and Cayenne API», *Jurnal Transformatika*, sep. 2021, doi: 10.26623/TRANSFORMATIKA.V17I2.1820.
- [9] D. Sitanggang, C. S. Sitompul, J. H. Suyanto, S. Kumar, y E. Indra, «Analysis of Air Quality Measuring Device Using Internet of Things-Based MQ-135 Sensor», *Sinkron*, vol. 7, n.º 3, pp. 1078-1084, 2022, doi: 10.33395/sinkron.v7i3.11618.
- [10] M. G. Arkhan y Z. R. S. Elsi, «Air Quality Monitoring System Based Internet Of Things», *Brilliance: Research of Artificial Intelligence*, vol. 4, n.º 2, pp. 669-673, sep. 2024, doi: 10.47709/brilliance.v4i2.4924.
- [11] R. Shrestha, M. Maharjan, y M. Sharma, «Air Pollution Monitoring System Using Micro Controller Atmega 32A and MQ135 Gas Sensor at Chandragiri Municipality of Kathmandu City», *Semiconductor Science and Information Devices*, vol. 4, n.º 2, pp. 35-44, oct. 2022, doi: 10.30564/ssid.v4i2.4884.
- [12] R. M. Nava, A. R. Macías, V. C. Serna, J. A. L. Tello, y A. S. Sales, «Diseño Y Desarrollo De Aplicación Remota Para El Control De Recicladora Extrusora De Filamento Usando Modulo Hc-05», *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, n.º 1, pp. 6349-6363, mar. 2024, doi: 10.37811/cl\_rcm.v8i1.9978.
- [13] D. G. F. Punguil, «Algoritmos de Deep Learning utilizando Tensorflow para el Tratamiento de Datos de Producción Científica», *Tesla Revista Científica*, vol. 3, n.º 2, pp. e226-e226, ago. 2023, doi: 10.55204/trc.v3i2.e226.
- [14] «(PDF) Método para la clasificación de imágenes de moda con TensorFlow», *ResearchGate*, ago. 2025, doi: 10.46588/invurnus.v18i1.88.
- [15] C. L. Vidal-Silva *et al.*, «Academic experience in rapid development of web

- information systems with Python and Django», *Formación universitaria*, vol. 14, n.º 5, pp. 85-94, oct. 2021, doi: 10.4067/S0718-50062021000500085.
- [16] V. Varbanov, K. Kopanov, y T. Atanasova, «A Multidisciplinary Approach to Telegram Data Analysis», nov. 2024, pp. 3-10. doi: 10.5593/sgem2024/2.1/s07.01.
- [17] «Broadcasting Messages via Telegram: Pro-Government Social Media Control During the 2020 Protests in Belarus and 2022 Anti-War Protests in Russia: Political Communication: Vol 41 , No 4 - Get Access». Accedido: 16 de agosto de 2025. [En línea]. Disponible en: <https://www.tandfonline.com/doi/full/10.1080/10584609.2023.2233444>
- [18] A. E. Wijaya y R. Muhromi, «PROTOTYPE OTOMATIS SISTEM MONITORING PEMELIHARAAN TOKEK GURUN BERBASIS IoT PADA PLATFORM THINGSPEAK MENGGUNAKAN ALGORITMA C4.5», *Jurnal Teknologi Informasi dan Komunikasi*, vol. 15, n.º 2, pp. 91-106, 2022, doi: 10.47561/a.v15i2.234.
- [19] Y. Hendrian, G. F. Ramadhan, y P. A. Nugroho, «Prototype Sistem Monitoring Banjir Berbasis Thingspeak Menggunakan Sensor HC-SR04 Dan Node MCU Esp-8266», *INSANtek*, vol. 5, n.º 1, pp. 01-06, may 2024, doi: 10.31294/insantek.v5i1.3301.
- [20] A. Chinasho, B. Bedadi, T. Lemma, T. Tana, B. Elias, y T. Hordofa, «Analysis of past and future temperature variability and change in Southern Ethiopia», *International Journal of Climate Change Strategies and Management*, vol. 17, n.º 1, pp. 399-423, ene. 2025, doi: 10.1108/IJCCSM-04-2023-0052.
- [21] A. Price, K. Mengersen, M. Rigby, y P. Fiévez, «Creating a Spatial Vulnerability Index for Environmental Health», 22 de marzo de 2024, *arXiv*: arXiv:2403.14954. doi: 10.48550/arXiv.2403.14954.
- [22] D. L. Vajda, T. V. Do, T. Bérczes, y K. Farkas, «Machine learning-based real-time anomaly detection using data pre-processing in the telemetry of server farms», *Sci Rep*, vol. 14, n.º 1, p. 23288, oct. 2024, doi: 10.1038/s41598-024-72982-z.
- [23] M. M. Kara, H. I. Turkmen, y M. A. Guvensan, «Smart Organization of Imbalanced Traffic Datasets for Long-Term Traffic Forecasting», *Sensors (Basel)*, vol. 25, n.º 4, p. 1225, feb. 2025, doi: 10.3390/s25041225.
- [24] S. Cárdenas-Rodríguez, C. A. Vides-Herrera, y A. Pardo-García, «Sistema de alerta temprana de inundaciones para el río Arauca basado en técnicas de inteligencia artificial», *Revista de Investigación, Desarrollo e Innovación*, vol. 12, n.º 2, pp. 315-326, ago. 2022, doi: 10.19053/20278306.v12.n2.2022.15274.
- [25] E. T. Vinueza y R. L. Cañar, «Implementación de una mini-red de sensores inalámbricos para detección temprana de incendios forestales», *Revista de Investigación en Tecnologías de la Información*, vol. 10, n.º 21 Especial, pp. 88-99, ago. 2022, doi: 10.36825/RITI.10.21.008.
- [26] D. R. C. Armas, B. C. H. Atencio, R. M. A. Castañeda, O. A. U. Ayala, A. F. A. Ramos, y M. W. O. Vergara, «Prototipo de sistema de alerta temprana para la prevención de huaicos, Chosica, Perú.», *Dilemas contemporáneos: Educación, Política y Valores*, ene. 2022, doi: 10.46377/dilemas.v9i2.3163.
- [27] M. M. Puentes *et al.*, «Implementación de un sistema de alerta temprana integral contra las inundaciones urbanas y desbordamientos de sistemas de saneamiento: el Proyecto LIFE BAETULO», *Ingeniería del Agua*, vol. 27, n.º 2, pp. 93-110, abr. 2023, doi: 10.4995/ia.2023.19129.
- [28] D. P. Aranda-Rúa, M. del P. Gómez-Fernández, M. P. de la Cruz, Q. A. Peralta-Núñez, M. Ortiz-Barrios, y L. Borrero-López, «Diseño de Herramientas para la Detección y Atención Temprana de Emergencias en Empresas de Producción: Un

- Caso de Estudio», *Boletín de Innovación, Logística y Operaciones*, vol. 4, n.º 1, jun. 2022, doi: 10.17981/bilo.4.1.2022.12.
- [29] N. V. Abasov, V. M. Nikitin, T. V. Bereznykh, y E. N. Osipchuk, «Monitoring and Predictive Estimations of Atmospheric Parameters in the Catchment Area of Lake Baikal», *Atmosphere*, vol. 13, n.º 1, p. 49, ene. 2022, doi: 10.3390/atmos13010049.
- [30] E. R. Jones, J. G. Cedeño Laurent, A. S. Young, B. A. Coull, J. D. Spengler, y J. G. Allen, «Indoor humidity levels and associations with reported symptoms in office buildings», *Indoor Air*, vol. 32, n.º 1, p. e12961, ene. 2022, doi: 10.1111/ina.12961.
- [31] «Adaptative vital signs monitoring system based on the early warning score approach in smart hospital context - Ben Ida - 2021 - IET Smart Cities - Wiley Online Library». Accedido: 16 de agosto de 2025. [En línea]. Disponible en: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/smc2.12004>
- [32] C.-F. Lin *et al.*, «Evaluation of a Telemonitoring System Using Electronic National Early Warning Scores for Patients Receiving Medical Home Care: Pilot Implementation Study», *JMIR Med Inform*, vol. 12, p. e63425, dic. 2024, doi: 10.2196/63425.
- [33] «(PDF) Smart Sensors and Intelligent Analysis: A Literature Review on More Effective Early Warning Systems with IoT and Machine Learning», *ResearchGate*, ago. 2025, doi: 10.69930/jsi.v1i4.182.
- [34] S. Pradhan, «Investigation into Smart Healthcare Monitoring System in an IoT Environment», *Malaysian Journal of Science and Advanced Technology*, feb. 2025, doi: 10.56532/MJSAT.V2I2.53.
- [35] J. A. J. Alsayaydeh, M. F. bin Yusof, M. Z. B. A. Halim, M. N. S. Zainudin, y S. G. Herawan, «Patient Health Monitoring System Development using ESP8266 and Arduino with IoT Platform», *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, n.º 4, 39/29 2023, doi: 10.14569/IJACSA.2023.0140467.

## XVIII. Anexos

- Anexo 1

La figura 9 y 10 se observa el código de configuración de Arduino uno en la plataforma de Arduino IDE.

```
1  #include <DHT.h>
2
3  #define DHTPIN 8
4  #define DHTTYPE DHT22
5  DHT dht(DHTPIN, DHTTYPE);
6
7  const int pinMQ135 = A0;
8
9  unsigned long previousMillis = 0;
10 const long intervalo = 2000;
11
12 void setup() {
13     Serial.begin(9600);
14     dht.begin();
15 }
16
17 void loop() {
18     unsigned long currentMillis = millis();
19
20     if (currentMillis - previousMillis >= intervalo) {
21         previousMillis = currentMillis;
```

Figura 9 - Código Arduino IDE.

```
22
23     float temperatura = dht.readTemperature();
24     float humedad = dht.readHumidity(); // NUEVO: lectura de humedad
25     int valorCrudo = analogRead(pinMQ135);
26     float ppmGas = map(valorCrudo, 0, 1023, 0, 100); // Escala simple 0-100
27
28
29     Serial.print(temperatura, 2);
30     Serial.print(",");
31     Serial.print(humedad, 2);
32     Serial.print(",");
33     Serial.println(ppmGas, 2);
34 }
35 }
```

Figura 10 - Código Arduino IDE.

- Anexo 2

Desde la figura 11 y 20 se observar el código de programa hecho en Python.

```
1 import serial
2 import tkinter as tk
3 import threading
4 import numpy as np
5 import requests
6 import time
7 from tensorflow.keras.models import Sequential
8 from tensorflow.keras.layers import LSTM, Dense
9 from sklearn.preprocessing import MinMaxScaler
10 from datetime import datetime
11 import matplotlib.pyplot as plt
12 import os
13
14 TOKEN = '8049896337:AAGv0j755vg-KuNTiAu61M8ni7WicjCWqjw'
15 chat_ids = set()
16 historico = []
17
18
19 def obtener_chat_ids():
20     global chat_ids
21     try:
22         res = requests.get(f'https://api.telegram.org/bot{TOKEN}/getUpdates')
23         data = res.json()
24         for resultado in data['result']:
25             if 'message' in resultado:
26                 cid = resultado['message']['chat']['id']
27                 chat_ids.add(cid)
28                 print(f"✅ chat_ids detectados: {chat_ids}")
29     except Exception as e:
30         print("⚠️ No se pudo obtener los chat_ids automáticamente:", e)
```

Figura 11 - Código Python.

```

33 def enviar_mensaje(texto):
34     for cid in chat_ids:
35         url = f'https://api.telegram.org/bot{TOKEN}/sendMessage'
36         requests.post(url, data={'chat_id': cid, 'text': texto, 'parse_mode': 'Markdown'})
37
38 root = tk.Tk()
39 root.title("Monitor principal")
40 root.geometry("700x280")
41 root.configure(bg="black")
42
43 label_tabla = tk.Label(root, text="", font=("Consolas", 14), bg="black", fg="white", justify="left")
44 label_tabla.pack(pady=10)
45
46 def actualizar_tabla(t, p, h, pred_t, pred_p, pred_h):
47     tabla = (
48         f" Parámetro | Actual | Predicción | Rango Activación Alerta\n"
49         f"-----|-----|-----|-----\n"
50         f" Temperatura (°C) | {t:7.2f} | {pred_t:12.2f} | 20 - 29\n"
51         f" PPM | {p:7.2f} | {pred_p:12.2f} | >= 65\n"
52         f" Humedad (%) | {h:7.2f} | {pred_h:12.2f} | 30 - 85\n"
53     )
54     label_tabla.config(text=tabla)
55
56 class ModeloAdquisicion:
57     def __init__(self, puerto):
58         self.puerto = puerto
59         self.temperatura = None
60         self.humedad = None
61         self.ppm = None
62         self.lock = threading.Lock()

```

Figura 12 - Código Python.

```

64 def leer_datos(self):
65     try:
66         ser = serial.Serial(self.puerto, 9600, timeout=1)
67         print(f"✅ Conectado a {self.puerto}...")
68         while True:
69             if ser.in_waiting > 0:
70                 linea = ser.readline().decode().strip()
71                 if linea.count(",") == 2:
72                     temp_str, hum_str, ppm_str = linea.split(",")
73                     t = float(temp_str) if temp_str != "nan" else np.nan
74                     h = float(hum_str) if hum_str != "nan" else np.nan
75                     p = float(ppm_str) if ppm_str != "nan" else np.nan
76
77                     if not np.isnan(t) and not np.isnan(h) and not np.isnan(p):
78                         with self.lock:
79                             self.temperatura = t
80                             self.humedad = h
81                             self.ppm = p
82                     else:
83                         enviar_mensaje(
84                             f"⚠️ ALERTA: Valor inválido detectado.\nTemperatura: {temp_str} °C\nHumedad:
85                         )
86             except Exception as e:
87                 print(f"⚠️ No se pudo abrir el puerto {self.puerto}: {e}")
88
89 def obtener_datos(self):
90     with self.lock:
91         return self.temperatura, self.humedad, self.ppm

```

Figura 13 - Código Python.

```
93 class ModeloPrediccion:
94     def __init__(self):
95         self.temperaturas = []
96         self.ppms = []
97         self.humedades = []
98         self.modelo_temp = self.crear_modelo()
99         self.modelo_ppm = self.crear_modelo()
100        self.modelo_hum = self.crear_modelo()
101        self.scaler_temp = MinMaxScaler()
102        self.scaler_ppm = MinMaxScaler()
103        self.scaler_hum = MinMaxScaler()
104
105    def crear_modelo(self):
106        model = Sequential([
107            LSTM(64, return_sequences=True, input_shape=(10, 1)),
108            LSTM(32),
109            Dense(1)
110        ])
111        model.compile(optimizer='adam', loss='mse')
112        return model
113
114    def entrenar_y_predecir(self, datos, modelo, scaler):
115        if len(datos) < 20:
116            return datos[-1]
117        serie = np.array(datos[-20:]).reshape(-1, 1)
118        serie_norm = scaler.fit_transform(serie).flatten()
119        X, y = [], []
120        for i in range(len(serie_norm) - 10):
121            X.append(serie_norm[i:i+10])
122            y.append(serie_norm[i+10])
```

Figura 14 - Código Python.

```

123 X = np.array(X).reshape(-1, 10, 1)
124 y = np.array(y)
125 model.fit(X, y, epochs=20, verbose=0)
126 entrada = np.array(serie_norm[-10:]).reshape(1, 10, 1)
127 pred_norm = model.predict(entrada, verbose=0)
128 return scaler.inverse_transform(pred_norm.reshape(-1, 1))[0][0]
129
130 def procesar(self, t, h, p):
131     self.temperaturas.append(t)
132     self.ppms.append(p)
133     self.humedades.append(h)
134     pred_t = self.entrenar_y_predecir(self.temperaturas, self.modelo_temp, self.scaler_temp)
135     pred_p = self.entrenar_y_predecir(self.ppms, self.modelo_ppm, self.scaler_ppm)
136     pred_h = self.entrenar_y_predecir(self.humedades, self.modelo_hum, self.scaler_hum)
137     return pred_t, pred_p, pred_h
138
139 class ModeloNotificaciones:
140     def __init__(self):
141         self.RANGO_TEMP_MIN = 20
142         self.RANGO_TEMP_MAX = 29
143         self.RANGO_PPM_MIN = 65
144         self.RANGO_HUM_MIN = 30
145         self.RANGO_HUM_MAX = 85
146
147     def verificar_y_enviar(self, t, p, h, pred_t, pred_p, pred_h):
148         alertas_pred = []
149         if pred_t <= self.RANGO_TEMP_MIN or pred_t >= self.RANGO_TEMP_MAX:
150             alertas_pred.append("Temperatura")
151         if pred_p < self.RANGO_PPM_MIN:

```

Figura 15 - Código Python.

```

152         alertas_pred.append("PPM")
153     if pred_h < self.RANGO_HUM_MIN or pred_h > self.RANGO_HUM_MAX:
154         alertas_pred.append("Humedad")
155
156     if alertas_pred:
157         mensaje_pred = "▲ *Predicción de posible salida de rango:* \n"
158         for param in alertas_pred:
159             mensaje_pred += f"Hay posibilidad que {param} se salga de los rangos.\n"
160         enviar_mensaje(mensaje_pred)
161
162     alertas_actuales = []
163     if t <= self.RANGO_TEMP_MIN or t >= self.RANGO_TEMP_MAX:
164         alertas_actuales.append(f"Temperatura: {t:.2f} °C fuera de rango")
165     if p < self.RANGO_PPM_MIN:
166         alertas_actuales.append(f"PPM: {p:.2f} fuera de rango")
167     if h < self.RANGO_HUM_MIN or h > self.RANGO_HUM_MAX:
168         alertas_actuales.append(f"Humedad: {h:.2f}% fuera de rango")
169
170     if alertas_actuales:
171         mensaje = "▲ ALERTA DE PARÁMETROS FUERA DE RANGO:\n" + "\n".join(alertas_actuales)
172         mensaje += f"\nValores actuales:\nTemperatura: {t:.2f} °C\nPPM: {p:.2f}\nHumedad: {h:.2f}%"
173         enviar_mensaje(mensaje)
174
175     def resumen_general():
176         while True:
177             time.sleep(10800)
178             if not historico:
179                 continue
180             temps = [x[1] for x in historico]

```

Figura 16 - Código Python.

```

181 ppms_ = [x[2] for x in historico]
182 hums = [x[3] for x in historico]
183 horas = [x[0] for x in historico]
184
185 resumen = " 📄 *Resumen general (últimos 3 horas)*\n"
186 tabla = "Hora\t\tTemp (°C)\tPPM\tHumedad (%)\n"
187
188 if max(temps) - min(temps) > 3:
189     resumen += " ▲ *Variación de temperatura* (≥3°C)\n"
190 if max(ppms_) - min(ppms_) > 50:
191     resumen += " ▲ *Variación de PPM* (≥50)\n"
192 if max(hums) - min(hums) > 20:
193     resumen += " ▲ *Variación de humedad* (≥20%)\n"
194 if resumen.strip() == " 📄 *Resumen general (últimos 5 minutos)*":
195     resumen += " ✅ Sin variaciones significativas.\n"
196
197 for reg in historico:
198     tabla += f"{reg[0]}\t{reg[1]:.1f}\t\t{reg[2]:.1f}\t\t{reg[3]:.1f}\n"
199
200 txt_path = "resumen_tabla.txt"
201 with open(txt_path, "w", encoding="utf-8") as f:
202     f.write(resumen + "\n" + tabla)
203
204 for cid in chat_ids:
205     with open(txt_path, "rb") as archivo:
206         requests.post(
207             f'https://api.telegram.org/bot{TOKEN}/sendDocument',
208             data={'chat_id': cid},

```

Figura 17 - Código Python.

```

209         files={'document': archivo}
210     )
211
212 def graficar_individual(datos, color, titulo, filename):
213     plt.figure(figsize=(8,3))
214     plt.plot(datos, color=color, marker='o')
215     plt.xticks([])
216     plt.title(titulo)
217     plt.grid(True)
218     plt.tight_layout()
219     plt.savefig(filename)
220     plt.close()
221
222 graficar_individual(temps, 'red', 'Temperatura (°C)', 'graf_temp.png')
223 graficar_individual(ppms_, 'blue', 'PPM', 'graf_ppm.png')
224 graficar_individual(hums, 'green', 'Humedad (%)', 'graf_hum.png')
225
226 for cid in chat_ids:
227     for archivo_graf in ['graf_temp.png', 'graf_ppm.png', 'graf_hum.png']:
228         with open(archivo_graf, "rb") as archivo:
229             requests.post(
230                 f'https://api.telegram.org/bot{TOKEN}/sendPhoto',
231                 data={'chat_id': cid},
232                 files={'photo': archivo}
233             )
234
235 historico.clear()

```

Figura 18 - Código Python.

```

236     os.remove(txt_path)
237     for f in ['graf_temp.png', 'graf_ppm.png', 'graf_hum.png']:
238         os.remove(f)
239
240
241     ultimo_envio = 0
242     def enviar_a_thingspeak(t, h, p):
243         global ultimo_envio
244         if time.time() - ultimo_envio >= 15:
245             try:
246                 params = {
247                     'api_key': 'EXATGSW863LZQAY1',
248                     'field1': t,
249                     'field2': h,
250                     'field3': p
251                 }
252                 requests.get("https://api.thingspeak.com/update", params=params)
253                 ultimo_envio = time.time()
254                 print("✅ Enviado a ThingSpeak")
255             except:
256                 print("❌ Error al enviar a ThingSpeak")
257
258     def ciclo_principal():
259         adquisicion = ModeloAdquisicion("COM7")
260         prediccion = ModeloPrediccion()
261         notificaciones = ModeloNotificaciones()
262
263         threading.Thread(target=adquisicion.leer_datos, daemon=True).start()
264         threading.Thread(target=resumen_general, daemon=True).start()

```

Figura 19 - Código Python.

```

265
266     def loop():
267         while True:
268             t, h, p = adquisicion.obtener_datos()
269             if t is not None and h is not None and p is not None:
270                 pred_t, pred_p, pred_h = prediccion.procesar(t, h, p)
271                 notificaciones.verificar_y_enviar(t, p, h, pred_t, pred_p, pred_h)
272                 root.after(0, actualizar_tabla, t, p, h, pred_t, pred_p, pred_h)
273                 enviar_a_thingspeak(t, h, p)
274                 historico.append((datetime.now().strftime('%H:%M:%S'), t, p, h))
275                 time.sleep(1)
276
277         threading.Thread(target=loop, daemon=True).start()
278
279     obtener_chat_ids()
280     ciclo_principal()
281     root.mainloop()
282

```

Figura 20 - Código Python.

- **Anexo 3**

Desde la figura 21 y 24 se presenta las notificaciones en Telegram, los resúmenes, el monitor principal y los datos en Thingspeak:

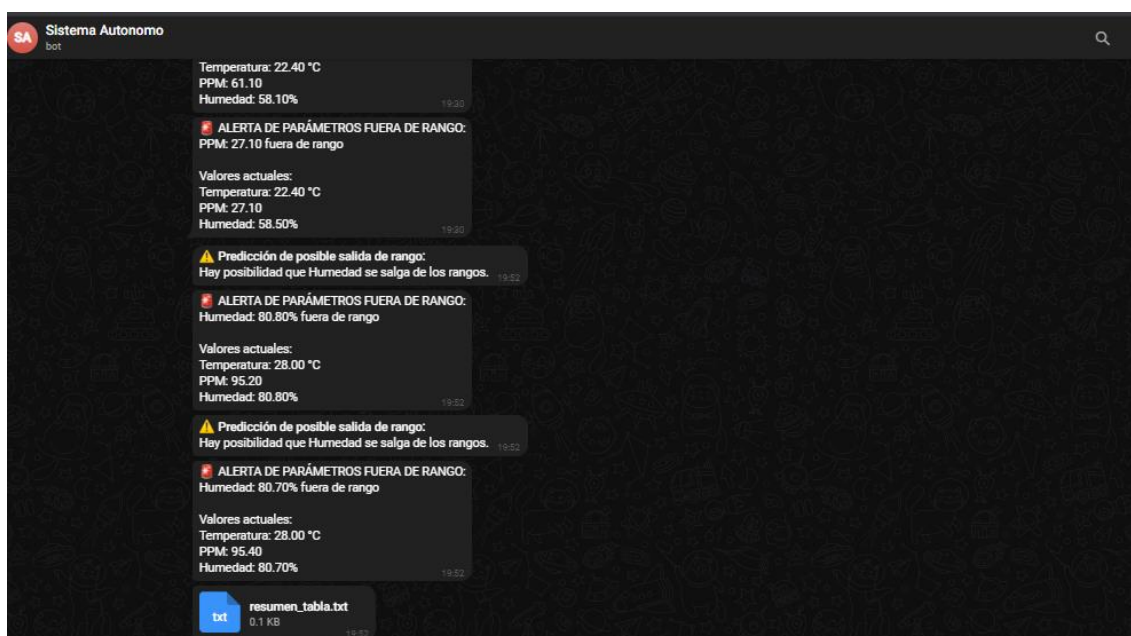


Figura 21 - Notificación de predicción y alerta en Telegram.

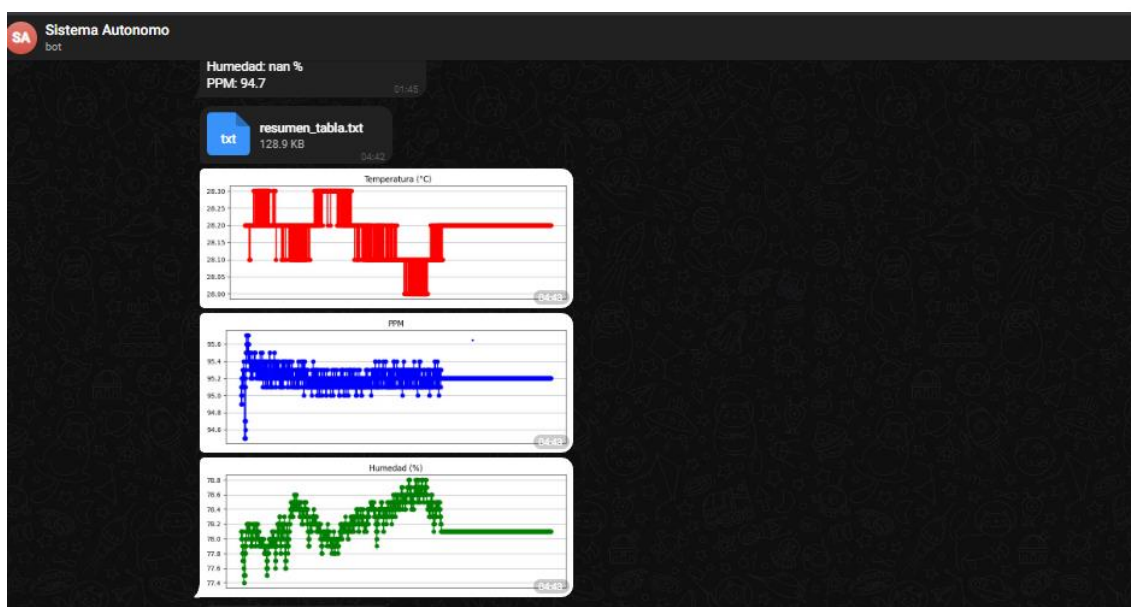


Figura 22 - Notificación de resumen en Telegram.

Channel Stats

Created: 20 days ago  
Last entry: 4 days ago  
Entries: 4232

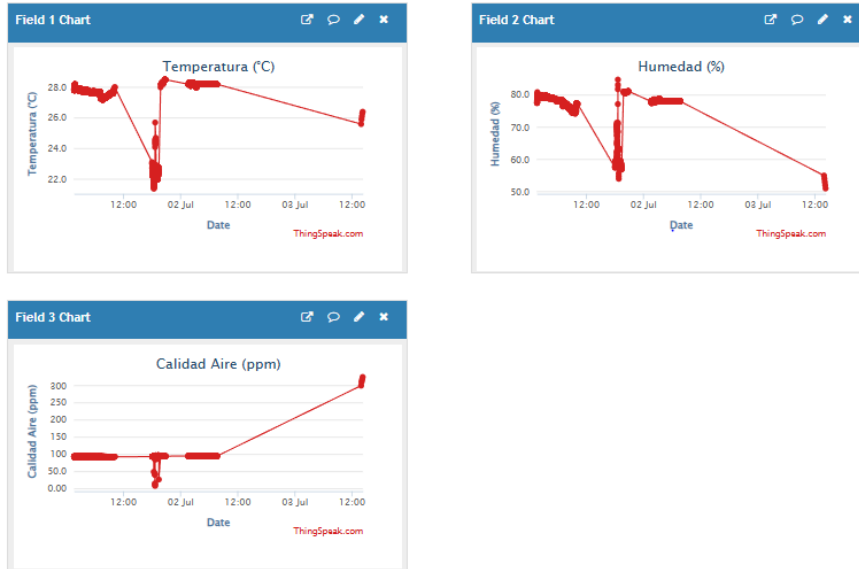


Figura 23 - Dashboard en Thingspeak.

Monitor principal

Parámetro	Actual	Predicción	Rango Activación Alerta
Temperatura (°C)	15.40	15.39	17 - 22
PPM	99.40	99.43	<= 65
Humedad (%)	57.60	57.08	30 - 82

Figura 24 - Monitor principal.

- **Anexo 4**

Tabla 5- Prueba del funcionamiento del prototipo.

Datos:

N° Dato	Hora	Tem (°C)	PPM	Hum (%)	N° Dato	Hora	Tem (°C)	PPM	Hum (%)
1	8:40:41	19,8	99,5	79,7	51	9:48:19	21,0	99,5	75,2
2	8:42:00	19,9	99,5	79,4	52	9:49:31	20,9	99,5	75,4
3	8:44:00	20,0	99,6	78,8	53	9:50:54	20,9	99,6	75,1
4	8:45:15	20,6	99,6	76,4	54	9:52:04	20,7	99,6	72,7
5	8:46:28	20,6	99,6	76,1	55	9:53:08	20,7	99,5	72,4
6	8:48:29	20,6	99,6	76,1	56	9:54:36	20,7	99,5	71,4
7	8:49:50	20,5	99,4	75,5	57	9:56:16	20,7	99,5	72,6
8	8:50:59	20,5	99,4	73,3	58	9:57:09	20,8	99,5	73,6
9	8:52:19	20,6	99,5	73,2	59	9:57:59	21,1	99,7	74,3
10	8:53:34	20,5	99,6	72,2	60	9:59:58	21,1	99,7	75,5
11	8:55:04	20,6	99,5	74,8	61	10:01:22	21,1	99,7	76,9
12	8:56:14	20,6	99,4	73,4	62	10:03:00	21,3	99,7	76,9
13	8:57:08	20,9	99,3	74,4	63	10:04:36	20,9	99,6	76,5
14	8:58:52	20,9	99,3	74,8	64	10:06:19	20,3	99,7	69,7
15	9:00:11	21,1	99,5	73,4	65	10:07:48	20,2	99,7	73,7
16	9:01:28	21,1	99,6	74,3	66	10:09:14	20,5	99,6	76,5
17	9:02:44	21,1	99,5	73,2	67	10:10:41	20,6	99,4	77,6
18	9:04:14	20,9	99,6	73,1	68	10:12:08	20,8	99,5	77,5
19	9:05:46	20,8	99,7	73,0	69	10:13:35	21,0	99,7	77,9
20	9:07:02	20,8	99,7	70,8	70	10:15:10	21,1	99,7	78,9
21	9:08:14	20,8	99,6	69,6	71	10:16:42	21,2	99,6	79,0
22	9:09:46	20,7	99,4	69,3	72	10:18:14	21,3	99,6	80,4
23	9:12:35	20,7	99,5	69,5	73	10:19:41	21,4	99,7	79,5
24	9:14:17	20,6	99,5	69,0	74	10:21:05	21,5	99,7	78,5
25	9:15:30	20,7	99,5	71,0	75	10:22:23	21,5	99,4	79,4
26	9:16:37	20,7	99,5	73,8	76	10:23:45	21,2	99,5	74,4
27	9:16:23	20,9	99,6	73,5	77	10:25:17	20,1	99,5	69,9
28	9:17:40	21,0	99,5	75,5	78	10:26:47	20,0	99,6	72,0
29	9:19:00	21,1	99,5	75,4	79	10:28:17	20,0	99,6	73,0
30	9:20:13	21,2	99,4	75,1	80	10:29:43	20,0	99,6	73,0
31	9:21:40	21,3	99,4	75,2	81	10:30:45	20,0	99,6	73,0
32	9:23:01	21,3	99,4	73,5	82	10:31:54	20,0	99,6	73,0
33	9:24:21	20,6	99,4	70,6	83	10:33:14	20,0	99,6	73,0
34	9:25:40	20,3	99,5	69,4	84	10:34:28	20,0	99,6	73,0
35	9:27:02	20,0	99,5	72,8	85	10:35:43	20,0	99,6	73,0
36	9:27:56	20,2	99,5	75,5	86	10:36:57	20,0	99,6	73,0

37	9:29:52	20,2	99,5	76,5	87	10:38:22	20,0	99,6	73,0
38	9:31:22	20,6	99,5	77,5	88	10:39:27	20,0	99,6	73,0
39	9:32:44	20,8	99,5	78,2	89	10:40:41	20,0	99,6	73,0
40	9:34:00	20,9	99,7	78,4	90	10:42:00	20,0	99,6	73,0
41	9:35:25	21,0	99,7	78,8	91	10:47:19	20,6	99,4	77,5
42	9:37:43	21,1	99,6	79,1	92	10:49:43	20,5	99,4	77,6
43	9:39:17	21,2	99,6	78,8	93	10:51:00	20,7	99,4	78,5
44	9:39:13	21,3	99,6	78,8	94	10:52:50	20,9	99,7	79,0
45	9:40:24	21,3	99,6	77,5	95	10:55:17	21,0	99,6	79,7
46	9:41:49	21,2	99,6	79,6	96	10:56:09	21,2	99,5	80,9
47	9:43:18	21,3	99,7	80,1	97	10:57:08	21,3	99,5	81,3
48	9:44:25	21,4	99,6	78,3	98	10:59:13	21,4	99,5	81,1
49	9:45:51	21,3	99,6	77,5	99	11:00:27	21,4	99,5	81,1
50	9:46:59	21,1	99,5	76,6	100	11:00:27	21,4	99,5	81,1

Tabla 6 - Prueba de notificación de la alta temperaturas.

Datos:

N° Dato	Hora	Tem (°C)	PPM	Hum (%)	N° Dato	Hora	Tem (°C)	PPM	Hum (%)
1	11:11:00	21,5	99,8	75,5	51	11:41:00	29,2	99,8	65,1
2	11:11:36	21,5	99,7	75,6	52	11:41:36	29,1	99,9	65,1
3	11:12:12	21,6	99,7	75,5	53	11:42:12	29,0	99,4	64,6
4	11:12:48	21,8	99,7	75,5	54	11:42:48	28,9	99,3	64,5
5	11:13:24	22,0	99,6	75,7	55	11:43:24	28,8	99,6	64,2
6	11:14:00	22,2	99,6	75,7	56	11:44:00	28,7	99,5	64,2
7	11:14:36	22,4	99,6	75,5	57	11:44:36	28,6	99,5	64,0
8	11:15:12	22,6	99,6	75,5	58	11:45:12	28,5	99,8	63,7
9	11:15:48	22,8	99,5	75,7	59	11:45:48	28,4	99,8	63,7
10	11:16:24	23,0	99,5	75,7	60	11:46:24	28,3	99,4	63,6
11	11:17:00	23,2	99,5	75,7	61	11:47:00	28,2	99,4	63,3
12	11:17:36	23,4	99,5	75,7	62	11:47:36	28,1	99,4	63,3
13	11:18:12	23,6	99,5	75,8	63	11:48:12	28,0	99,5	63,3
14	11:18:48	23,8	99,5	75,8	64	11:48:48	27,9	99,5	63,3
15	11:19:24	24,0	99,6	75,8	65	11:49:24	27,8	99,8	63,3
16	11:20:00	24,2	99,6	75,8	66	11:50:00	27,7	99,6	63,6
17	11:20:36	24,4	99,6	75,8	67	11:50:36	27,6	99,5	63,7
18	11:21:12	24,6	99,6	75,8	68	11:51:12	27,5	99,5	63,7
19	11:21:48	24,8	99,5	77,0	69	11:51:48	27,4	99,6	63,8
20	11:22:24	25,0	99,5	77,0	70	11:52:24	27,3	99,5	64,0
21	11:23:00	25,2	99,5	80,7	71	11:53:00	27,2	99,7	64,0
22	11:23:36	25,4	99,5	81,8	72	11:53:36	27,1	99,7	64,1

23	11:24:12	25,6	99,5	81,8	73	11:54:12	27,0	99,6	64,3
24	11:24:48	25,8	99,4	81,4	74	11:54:48	26,9	99,7	64,5
25	11:25:24	26,0	99,5	80,7	75	11:55:24	26,8	99,4	64,7
26	11:26:00	26,2	99,6	79,6	76	11:56:00	26,7	99,3	64,9
27	11:26:36	26,4	99,6	77,2	77	11:56:36	26,6	99,5	65,0
28	11:27:12	26,6	99,6	74,8	78	11:57:12	26,5	99,4	65,3
29	11:27:48	26,8	99,6	73,8	79	11:57:48	26,4	99,7	65,5
30	11:28:24	27,0	99,6	73,0	80	11:58:24	26,3	99,5	65,7
31	11:29:00	27,2	99,6	73,3	81	11:59:00	26,2	99,3	66,0
32	11:29:36	27,4	99,6	72,9	82	11:59:36	26,1	99,6	66,2
33	11:30:12	27,6	99,4	71,9	83	12:00:12	26,0	99,8	66,4
34	11:30:48	27,8	99,5	70,9	84	12:00:48	25,9	99,9	66,6
35	11:31:24	28,0	99,6	70,3	85	12:01:24	25,8	99,6	66,9
36	11:32:00	28,2	99,7	69,6	86	12:02:00	25,7	99,6	67,1
37	11:32:36	28,4	99,8	69,1	87	12:02:36	25,6	99,5	67,3
38	11:33:12	28,6	99,6	68,5	88	12:03:12	25,5	99,7	67,4
39	11:33:48	28,8	99,4	67,9	89	12:03:48	25,4	99,7	67,7
40	11:34:24	29,0	99,6	67,5	90	12:04:24	25,3	99,6	68,1
41	11:35:00	29,2	99,5	66,9	91	12:05:00	25,2	99,5	68,3
42	11:35:36	29,4	99,4	66,8	92	12:05:36	25,1	99,4	68,6
43	11:36:12	29,6	99,5	66,4	93	12:06:12	25,0	99,6	68,7
44	11:36:48	29,8	99,4	66,2	94	12:06:48	24,9	99,6	68,9
45	11:37:24	29,8	99,6	66,1	95	12:07:24	24,8	99,5	69,1
46	11:38:00	29,7	99,6	65,4	96	12:08:00	24,7	99,5	69,3
47	11:38:36	29,6	99,6	65,0	97	12:08:36	24,6	99,4	69,5
48	11:39:12	29,5	99,6	64,8	98	12:09:12	24,5	99,6	69,7
49	11:39:48	29,4	99,6	64,7	99	12:09:48	24,4	99,7	69,9
50	11:40:24	29,3	99,6	64,8	100	12:10:24	24,3	99,5	70,1

Tabla 7- Prueba de notificación con temperaturas bajas.

Datos:

N° Dato	Hora	Tem (°C)	PPM	Hum (%)	N° Dato	Hora	Tem (°C)	PPM	Hum (%)
1	9:04:33	21.5	99.4	75.2	51	9:31:13	18.8	99.6	80.4
2	9:05:05	21.5	99.5	74.4	52	9:31:45	18.7	99.8	80.6
3	9:05:37	21.4	99.4	75.2	53	9:32:17	18.7	99.3	80.7
4	9:06:09	21.4	99.6	74.4	54	9:32:49	18.6	99.5	80.9
5	9:06:41	21.3	99.6	74.3	55	9:33:21	18.7	99.7	81
6	9:07:13	21.3	99.5	75.4	56	9:33:53	18.6	99.6	81
7	9:07:45	21.3	99.5	75.4	57	9:34:25	18.6	99.6	81.1

8	9:08:17	21	99.3	74.6	58	9:34:57	18.5	99.5	80.3
9	9:08:49	21	99.5	74.7	59	9:35:29	18.5	99.6	79.3
10	9:09:21	20.9	99.4	74.7	60	9:36:01	18.4	99.4	78.4
11	9:09:53	20.9	99.4	74.7	61	9:36:33	18.3	99.6	77.8
12	9:10:25	20.8	99.3	74.6	62	9:37:05	18.3	99.5	76.7
13	9:10:57	20.8	99.4	74.2	63	9:37:37	18.2	99.5	76.1
14	9:11:29	20.7	99.5	75.4	64	9:38:09	18.2	99.7	75.5
15	9:12:01	20.7	99.5	75.5	65	9:38:41	18.1	99.6	75
16	9:12:33	20.7	99.7	74.4	66	9:39:13	18	99.5	74.2
17	9:13:05	20.6	99.8	74.7	67	9:39:45	17.9	99.4	73.8
18	9:13:37	20.6	99.6	75.7	68	9:40:17	17.9	99.3	73.3
19	9:14:09	20.5	99.3	75	69	9:40:49	17.8	99.4	72.7
20	9:14:41	20.5	99.2	76	70	9:41:21	17.8	99.4	72.2
21	9:15:13	20.4	99.5	75.2	71	9:41:53	17.7	99.6	71.1
22	9:15:45	20.3	99.6	75.3	72	9:42:25	17.6	99.6	70.8
23	9:16:17	20.3	99.7	75.4	73	9:42:57	17.6	99.5	70.4
24	9:16:49	20.2	99.5	75.6	74	9:43:29	17.5	99.5	69.9
25	9:17:21	20.1	99.7	75.8	75	9:44:01	17.5	99.6	69.5
26	9:17:53	20	99.5	76.1	76	9:44:33	17.4	99.6	69.2
27	9:18:25	20	99.4	76.2	77	9:45:05	17.4	99.7	69.2
28	9:18:57	19.9	99.5	76.4	78	9:45:37	17.3	99.5	68.6
29	9:19:29	19.8	99.4	76.6	79	9:46:09	17.1	99.4	68.3
30	9:20:01	19.8	99.4	76.8	80	9:46:41	17	99.4	68
31	9:20:33	19.7	99.4	77	81	9:47:13	16.9	99.5	67.6
32	9:21:05	19.7	99.5	77.2	82	9:47:45	16.8	99.6	66.9
33	9:21:37	19.6	99.5	77.4	83	9:48:17	16.7	99.6	66.2
34	9:22:09	19.6	99.4	77.6	84	9:48:49	16.6	99.6	66.1
35	9:22:41	19.5	99.5	77.9	85	9:49:21	16.5	99.6	65.6
36	9:23:13	19.4	99.5	78	86	9:49:53	16.4	99.5	64.4
37	9:23:45	19.4	99.4	78.1	87	9:50:25	16.2	99.7	63.3
38	9:24:17	19.3	99.6	78.5	88	9:50:57	16.1	99.6	62.1
39	9:24:49	19.3	99.5	78.8	89	9:51:29	16	99.3	60.9
40	9:25:21	19.2	99.4	78.9	90	9:52:01	15.9	99.5	60.2
41	9:25:53	19.2	99.2	79	91	9:52:33	15.8	99.5	59.3
42	9:26:25	19.1	99.3	79.1	92	9:53:05	15.7	99.5	58.6
43	9:26:57	19.1	99.4	79.3	93	9:53:37	15.6	99.4	58.1
44	9:27:29	19	99.4	79.4	94	9:54:09	15.5	99.5	57.7
45	9:28:01	19	99.4	79.6	95	9:54:41	15.4	99.3	57.5
46	9:28:33	19	99.4	79.8	96	9:55:13	15.3	99.5	57.3
47	9:29:05	18.9	99.5	79.7	97	9:55:45	15.3	99.7	57.2
48	9:29:37	18.9	99.4	80	98	9:56:17	15.2	99.5	57
49	9:30:09	18.8	99.5	80.1	99	9:56:49	15.1	99.5	57
50	9:30:41	18.8	99.4	80.3	100	9:57:21	14.9	99.6	57

Tabla 8- Prueba de notificación de humedad.

Datos:

N° Da to	Hora	Tem (°C)	PPM	Hum (%)	N° Dat o	Hora	Tem (°C)	PPM	Hum (%)
1	12:15:23	19.3	99.5	75	51	12:45:24	19.9	99.5	82.6
2	12:16:00	19.3	99.5	75.2	52	12:46:00	19.9	99.5	82.7
3	12:16:36	19.3	99.5	75.3	53	12:46:36	19.9	99.5	82.9
4	12:17:12	19.3	99.5	75.5	54	12:47:12	19.9	99.5	83
5	12:17:48	19.3	99.5	75.6	55	12:47:48	19.9	99.5	83.2
6	12:18:24	19.4	99.5	75.8	56	12:48:24	19.9	99.5	83.3
7	12:19:00	19.4	99.5	75.9	57	12:49:00	19.9	99.5	83.5
8	12:19:36	19.4	99.5	76.1	58	12:49:36	19.9	99.5	83.6
9	12:20:12	19.4	99.5	76.2	59	12:50:12	19.9	99.5	83.8
10	12:20:48	19.4	99.5	76.4	60	12:50:48	20	99.5	83.9
11	12:21:24	19.4	99.5	76.5	61	12:51:24	20	99.5	84.1
12	12:22:00	19.4	99.5	76.7	62	12:52:00	20	99.5	84.2
13	12:22:36	19.4	99.5	76.8	63	12:52:36	20	99.5	84.4
14	12:23:12	19.4	99.5	77	64	12:53:12	20	99.5	84.5
15	12:23:48	19.5	99.5	77.1	65	12:53:48	20	99.5	84.7
16	12:24:24	19.5	99.5	77.3	66	12:54:24	20	99.5	84.8
17	12:25:00	19.5	99.5	77.4	67	12:55:00	20	99.5	85
18	12:25:36	19.5	99.5	77.6	68	12:55:36	20	99.5	85.2
19	12:26:12	19.5	99.5	77.7	69	12:56:12	20.1	99.5	85.3
20	12:26:48	19.5	99.5	77.9	70	12:56:48	20.1	99.5	85.5
21	12:27:24	19.5	99.5	78	71	12:57:24	20.1	99.5	85.6
22	12:28:00	19.5	99.5	78.2	72	12:58:00	20.1	99.5	85.8
23	12:28:36	19.5	99.5	78.3	73	12:58:36	20.1	99.5	85.9
24	12:29:12	19.6	99.5	78.5	74	12:59:12	20.1	99.5	86.1
25	12:29:48	19.6	99.5	78.6	75	12:59:48	20.1	99.5	86.2
26	12:30:24	19.6	99.5	78.8	76	13:00:24	20.1	99.5	86.4
27	12:31:00	19.6	99.5	78.9	77	13:01:00	20.1	99.5	86.5
28	12:31:36	19.6	99.5	79.1	78	13:01:36	20.2	99.5	86.7
29	12:32:12	19.6	99.5	79.2	79	13:02:12	20.2	99.5	86.8
30	12:32:48	19.6	99.5	79.4	80	13:02:48	20.2	99.5	87
31	12:33:24	19.6	99.5	79.5	81	13:03:24	20.2	99.5	87.1
32	12:34:00	19.6	99.5	79.7	82	13:04:00	20.2	99.5	87.3
33	12:34:36	19.7	99.5	79.8	83	13:04:36	20.2	99.5	87.4
34	12:35:12	19.7	99.5	80	84	13:05:12	20.2	99.5	87.6
35	12:35:48	19.7	99.5	80.2	85	13:05:48	20.2	99.5	87.7
36	12:36:24	19.7	99.5	80.3	86	13:06:24	20.2	99.5	87.9
37	12:37:00	19.7	99.5	80.5	87	13:07:00	20.3	99.5	88

38	12:37:36	19.7	99.5	80.6	88	13:07:36	20.3	99.5	88.2
39	12:38:12	19.7	99.5	80.8	89	13:08:12	20.3	99.5	88.3
40	12:38:48	19.7	99.5	80.9	90	13:08:48	20.3	99.5	88.5
41	12:39:24	19.7	99.5	81.1	91	13:09:24	20.3	99.5	88.6
42	12:40:00	19.8	99.5	81.2	92	13:10:00	20.3	99.5	88.8
43	12:40:36	19.8	99.5	81.4	93	13:10:36	20.3	99.5	88.9
44	12:41:12	19.8	99.5	81.5	94	13:11:12	20.3	99.5	89.1
45	12:41:48	19.8	99.5	81.7	95	13:11:48	20.3	99.5	89.2
46	12:42:24	19.8	99.5	81.8	96	13:12:24	20.4	99.5	89.4
47	12:43:00	19.8	99.5	82	97	13:13:00	20.4	99.5	89.5
48	12:43:36	19.8	99.5	82.1	98	13:13:36	20.4	99.5	89.7
49	12:44:12	19.8	99.5	82.3	99	13:14:12	20.4	99.5	89.8
50	12:44:48	19.8	99.5	82.4	100	13:14:48	20.4	99.5	89.7

Tabla 9- Prueba de medición de niveles de oxígeno en ppm.

Datos:

N° Dato	Hora	Tem (°C)	PPM	Hum (%)	N° Dato	Hora	Tem (°C)	PPM	Hum (%)
1	11:04:02	20,2	99,6	71,1	51	11:06:52	21,4	98,3	73,8
2	11:04:03	20,1	99,7	71,2	52	11:06:55	21,4	99,2	73,5
3	11:04:04	20,1	99,7	71,2	53	11:06:57	21,5	99,5	73,4
4	11:04:05	20,2	99,5	71,3	54	11:07:00	21,5	99,4	73,3
5	11:04:06	20,2	99,5	71,3	55	11:07:02	21,5	99,4	73,3
6	11:04:07	20,1	99,4	71,4	56	11:07:05	21,5	99,4	74,2
7	11:04:08	20,1	99,4	71,4	57	11:07:07	21,5	99,6	72,9
8	11:04:09	20,1	99,4	71,5	58	11:07:11	21,5	99,7	72,9
9	11:04:10	20,1	99,4	71,5	59	11:07:13	21,5	99,8	72,7
10	11:04:11	20,1	99,5	72,3	60	11:07:15	21,6	99,5	72,7
11	11:04:12	20,1	99,5	72,3	61	11:07:17	21,6	61,3	72,6
12	11:04:13	20,1	99,5	71,8	62	11:07:20	21,6	61,3	72,6
13	11:04:14	20,1	99,5	71,8	63	11:07:23	21,6	30,1	72,7
14	11:04:17	20,1	61,7	71,9	64	11:07:27	21,6	19,6	72,7
15	11:04:20	20,1	38,5	72,1	65	11:07:31	21,6	9,6	72,9
16	11:04:22	20,2	16,2	72,8	66	11:07:35	21,6	5,9	73,4
17	11:04:26	20,2	16,2	72,8	67	11:07:39	21,7	4,6	73,7
18	11:04:27	20,2	67,8	73,8	68	11:07:44	21,7	4,2	74,9
19	11:04:28	20,2	67,8	73,8	69	11:07:49	21,7	4,0	75,3
20	11:04:50	20,2	80,5	74,1	70	11:07:53	21,7	3,9	75,4
21	11:04:54	20,4	55,6	75,1	71	11:07:56	21,8	3,9	74,6
22	11:04:58	20,4	22,4	74,8	72	11:08:00	21,8	42,2	74,9
23	11:05:02	20,4	10,5	74,7	73	11:08:02	21,8	87,2	75,6

24	11:05:06	20,4	7,8	74,8	74	11:08:04	21,9	92,1	74,7
25	11:05:10	20,5	4,7	75,0	75	11:08:06	21,9	95,2	75,3
26	11:05:14	20,5	4,1	75,1	76	11:08:08	21,9	97,1	75,1
27	11:05:17	20,5	4,0	75,3	77	11:08:11	21,9	98,1	75,1
28	11:05:22	20,6	3,9	75,2	78	11:08:13	21,9	98,7	74,9
29	11:05:25	20,6	3,8	75,2	79	11:08:15	21,9	98,9	74,0
30	11:05:29	20,6	3,8	75,0	80	11:08:17	21,9	99,4	74,6
31	11:05:32	20,7	3,8	75,1	81	11:08:19	21,9	99,2	74,5
32	11:05:36	20,7	3,8	75,0	82	11:08:23	21,9	61,1	74,5
33	11:05:40	20,8	3,8	74,9	83	11:08:27	21,9	24,5	74,5
34	11:05:47	20,8	3,8	75,9	84	11:08:30	22,0	11,3	73,9
35	11:05:50	20,9	42,1	75,7	85	11:08:34	22,0	8,4	74,0
36	11:05:54	20,9	56,1	75,7	86	11:08:37	22,0	66,8	74,8
37	11:05:58	21,0	84,1	74,4	87	11:08:41	22,0	87,8	74,9
38	11:06:00	21,0	90,3	74,2	88	11:08:44	22,0	95,3	74,7
39	11:06:02	21,0	96,2	74,1	89	11:08:47	22,0	97,0	74,6
40	11:06:06	21,0	59,3	75,0	90	11:08:51	22,0	98,6	74,4
41	11:06:11	21,1	23,8	73,8	91	11:08:55	22,0	99,2	74,4
42	11:06:15	21,1	11,0	75,0	92	11:08:59	22,0	99,4	74,2
43	11:06:19	21,1	6,4	73,8	93	11:09:02	22,0	99,5	74,1
44	11:06:26	21,1	4,8	75,0	94	11:09:06	22,0	99,5	74,1
45	11:06:30	21,2	65,2	75,0	95	11:09:10	22,0	99,5	74,1
46	11:06:36	21,3	48,9	74,2	96	11:09:13	22,0	99,6	73,3
47	11:06:39	21,3	58,3	75,5	97	11:09:15	21,9	99,7	73,9
48	11:06:42	21,3	84,7	75,4	98	11:09:17	21,9	99,8	74,0
49	11:06:45	21,4	94,2	74,2	99	11:09:20	21,9	99,7	74,0
50	11:06:48	21,4	96,3	74,0	100	11:09:22	21,9	99,8	74,1

- **Anexo 5**

En las siguientes tablas 10 y 11 se realiza la verificación de los datos obtenidos por el prototipo y se analiza la comparación con una pistola de temperatura infrarroja comercial.

Tabla 10- Comparación de la prueba de alta temperatura.

N° Dat o	Hora	Protot ipo (°C)	Pist ola Infr arro jo (°C)	Dife renc ia (°C)	Error % Apro x	N° Dat o	Hora	Pro toti po (°C )	Pisto la Infra rojo (°C)	Dif ere ncia (°C )	Error % Apro x
1	11:11: 00	21.5	20.1	1.4	6.97	51	11:41: 00	29. 2	27.3	1.9	6.96
2	11:11: 36	21.5	20.2	1.3	6.44	52	11:41: 36	29. 1	27.2	1.9	6.99
3	11:12: 12	21.6	20.6	1	4.85	53	11:42: 12	29	27.1	1.9	7.01
4	11:12: 48	21.8	20.6	1.2	5.83	54	11:42: 48	28. 9	27	1.9	7.04
5	11:13: 24	22	20.2	1.8	8.91	55	11:43: 24	28. 8	26.9	1.9	7.06
6	11:14: 00	22.2	20.8	1.4	6.73	56	11:44: 00	28. 7	26.8	1.9	7.09
7	11:14: 36	22.4	21	1.4	6.67	57	11:44: 36	28. 6	26.7	1.9	7.12
8	11:15: 12	22.6	21.2	1.4	6.6	58	11:45: 12	28. 5	26.6	1.9	7.14
9	11:15: 48	22.8	21.5	1.3	6.05	59	11:45: 48	28. 4	26.5	1.9	7.17
10	11:16: 24	23	21.6	1.4	6.48	60	11:46: 24	28. 3	26.4	1.9	7.2
11	11:17: 00	23.2	21.9	1.3	5.94	61	11:47: 00	28. 2	26.3	1.9	7.22
12	11:17: 36	23.4	22.1	1.3	5.88	62	11:47: 36	28. 1	26.2	1.9	7.25
13	11:18: 12	23.6	22.3	1.3	5.83	63	11:48: 12	28	26.1	1.9	7.28
14	11:18: 48	23.8	22.4	1.4	6.25	64	11:48: 48	27. 9	26	1.9	7.31
15	11:19: 24	24	22.7	1.3	5.73	65	11:49: 24	27. 8	25.9	1.9	7.34
16	11:20: 00	24.2	22.8	1.4	6.14	66	11:50: 00	27. 7	25.8	1.9	7.36
17	11:20: 36	24.4	23	1.4	6.09	67	11:50: 36	27. 6	25.7	1.9	7.39

18	11:21: 12	24.6	23.1	1.5	6.49	68	11:51: 12	27. 5	25.6	1.9	7.42
19	11:21: 48	24.8	23.2	1.6	6.9	69	11:51: 48	27. 4	25.5	1.9	7.45
20	11:22: 24	25	23.4	1.6	6.84	70	11:52: 24	27. 3	25.4	1.9	7.48
21	11:23: 00	25.2	23.6	1.6	6.78	71	11:53: 00	27. 2	25.3	1.9	7.51
22	11:23: 36	25.4	23.7	1.7	7.17	72	11:53: 36	27. 1	25.2	1.9	7.54
23	11:24: 12	25.6	23.9	1.7	7.11	73	11:54: 12	27	25.1	1.9	7.57
24	11:24: 48	25.8	24.2	1.6	6.61	74	11:54: 48	26. 9	25	1.9	7.6
25	11:25: 24	26	24.3	1.7	7	75	11:55: 24	26. 8	24.9	1.9	7.63
26	11:26: 00	26.2	24.5	1.7	6.94	76	11:56: 00	26. 7	24.8	1.9	7.66
27	11:26: 36	26.4	24.7	1.7	6.88	77	11:56: 36	26. 6	24.7	1.9	7.69
28	11:27: 12	26.6	24.8	1.8	7.26	78	11:57: 12	26. 5	24.6	1.9	7.72
29	11:27: 48	26.8	25	1.8	7.2	79	11:57: 48	26. 4	24.5	1.9	7.76
30	11:28: 24	27	25.2	1.8	7.14	80	11:58: 24	26. 3	24.4	1.9	7.79
31	11:29: 00	27.2	25.4	1.8	7.09	81	11:59: 00	26. 2	24.3	1.9	7.82
32	11:29: 36	27.4	25.6	1.8	7.03	82	11:59: 36	26. 1	24.2	1.9	7.85
33	11:30: 12	27.6	25.8	1.8	6.98	83	12:00: 12	26	24.1	1.9	7.88
34	11:30: 48	27.8	26	1.8	6.92	84	12:00: 48	25. 9	24	1.9	7.92
35	11:31: 24	28	26.2	1.8	6.87	85	12:01: 24	25. 8	23.9	1.9	7.95
36	11:32: 00	28.2	26.4	1.8	6.82	86	12:02: 00	25. 7	23.8	1.9	7.98
37	11:32: 36	28.4	26.6	1.8	6.77	87	12:02: 36	25. 6	23.7	1.9	8.02
38	11:33: 12	28.6	26.8	1.8	6.72	88	12:03: 12	25. 5	23.6	1.9	8.05
39	11:33: 48	28.8	27	1.8	6.67	89	12:03: 48	25. 4	23.5	1.9	8.09
40	11:34: 24	29	27.2	1.8	6.62	90	12:04: 24	25. 3	23.4	1.9	8.12
41	11:35: 00	29.2	27.4	1.8	6.57	91	12:05: 00	25. 2	23.3	1.9	8.15
42	11:35: 36	29.4	27.5	1.9	6.91	92	12:05: 36	25. 1	23.2	1.9	8.19

43	11:36: 12	29.6	27.7	1.9	6.86	93	12:06: 12	25	23.1	1.9	8.23
44	11:36: 48	29.8	27.9	1.9	6.81	94	12:06: 48	24. 9	23	1.9	8.26
45	11:37: 24	29.8	27.9	1.9	6.81	95	12:07: 24	24. 8	22.9	1.9	8.3
46	11:38: 00	29.7	27.8	1.9	6.83	96	12:08: 00	24. 7	22.8	1.9	8.33
47	11:38: 36	29.6	27.7	1.9	6.86	97	12:08: 36	24. 6	22.7	1.9	8.37
48	11:39: 12	29.5	27.6	1.9	6.88	98	12:09: 12	24. 5	22.6	1.9	8.41
49	11:39: 48	29.4	27.5	1.9	6.91	99	12:09: 48	24. 4	22.5	1.9	8.44
50	11:40: 24	29.3	27.4	1.9	6.93	100	12:10: 24	24. 3	22.4	1.9	8.48

Tabla 11- Comparación de la prueba de baja temperatura.

N° Dat o	Hora	Protot ipo (°C)	Pist ola Infr arro jo (°C)	Dife renc ia (°C)	Error % Apro x	N° Dat o	Hora	Pro toti po (°C )	Pisto la Infra rojo (°C)	Dif ere ncia (°C )	Error % Apro x
1	9:04:3 3	21.5	21.6	0.1	0.47	51	9:31:1 3	18. 8	18.3	-0.5	2.66
2	9:05:0 5	21.5	20.6	-0.9	4.19	52	9:31:4 5	18. 7	16.6	-2.1	11.23
3	9:05:3 7	21.4	23.3	1.9	8.88	53	9:32:1 7	18. 7	19.7	1	5.35
4	9:06:0 9	21.4	19.7	-1.7	7.94	54	9:32:4 9	18. 6	19.3	0.7	3.76
5	9:06:4 1	21.3	22.8	1.5	7.04	55	9:33:2 1	18. 7	17	-1.7	9.09
6	9:07:1 3	21.3	22.8	1.5	7.04	56	9:33:5 3	18. 6	17.1	-1.5	8.06
7	9:07:4 5	21.3	20.7	-0.6	2.82	57	9:34:2 5	18. 6	18.6	0	0
8	9:08:1 7	21	20.8	-0.2	0.95	58	9:34:5 7	18. 5	18.5	0	0
9	9:08:4 9	21	21.9	0.9	4.29	59	9:35:2 9	18. 5	19.2	0.7	3.78
10	9:09:2 1	20.9	21	0.1	0.48	60	9:36:0 1	18. 4	19.7	1.3	7.07
11	9:09:5 3	20.9	19.1	-1.8	8.61	61	9:36:3 3	18. 3	16.6	-1.7	9.29

12	9:10:2 5	20.8	19.6	-1.2	5.77	62	9:37:0 5	18. 3	17.8	-0.5	2.73
13	9:10:5 7	20.8	23.2	2.4	11.54	63	9:37:3 7	18. 2	19.9	1.7	9.34
14	9:11:2 9	20.7	21.9	1.2	5.8	64	9:38:0 9	18. 2	20.3	2.1	11.54
15	9:12:0 1	20.7	20.8	0.1	0.48	65	9:38:4 1	18. 1	19.4	1.3	7.18
16	9:12:3 3	20.7	19.1	-1.6	7.73	66	9:39:1 3	18	20.2	2.2	12.22
17	9:13:0 5	20.6	20.8	0.2	0.97	67	9:39:4 5	17. 9	19.4	1.5	8.38
18	9:13:3 7	20.6	21	0.4	1.94	68	9:40:1 7	17. 9	19.7	1.8	10.06
19	9:14:0 9	20.5	20.1	-0.4	1.95	69	9:40:4 9	17. 8	20.3	2.5	14.04
20	9:14:4 1	20.5	21	0.5	2.44	70	9:41:2 1	17. 8	19.6	1.8	10.11
21	9:15:1 3	20.4	19.6	-0.8	3.92	71	9:41:5 3	17. 7	19.5	1.8	10.17
22	9:15:4 5	20.3	19.7	-0.6	2.96	72	9:42:2 5	17. 6	19	1.4	7.95
23	9:16:1 7	20.3	20.9	0.6	2.96	73	9:42:5 7	17. 6	20.3	2.7	15.34
24	9:16:4 9	20.2	21.9	1.7	8.42	74	9:43:2 9	17. 5	18.3	0.8	4.57
25	9:17:2 1	20.1	18.5	-1.6	7.96	75	9:44:0 1	17. 5	17.2	-0.3	1.71
26	9:17:5 3	20	19.1	-0.9	4.5	76	9:44:3 3	17. 4	19.6	2.2	12.64
27	9:18:2 5	20	20.4	0.4	2	77	9:45:0 5	17. 4	17.4	0	0
28	9:18:5 7	19.9	22	2.1	10.55	78	9:45:3 7	17. 3	20.1	2.8	16.18
29	9:19:2 9	19.8	22	2.2	11.11	79	9:46:0 9	17. 1	18.3	1.2	7.02
30	9:20:0 1	19.8	19.9	0.1	0.51	80	9:46:4 1	17	17.1	0.1	0.59
31	9:20:3 3	19.7	18.8	-0.9	4.57	81	9:47:1 3	16. 9	19.7	2.8	16.57
32	9:21:0 5	19.7	20.5	0.8	4.06	82	9:47:4 5	16. 8	17.5	0.7	4.17
33	9:21:3 7	19.6	20	0.4	2.04	83	9:48:1 7	16. 7	17.8	1.1	6.59
34	9:22:0 9	19.6	18.9	-0.7	3.57	84	9:48:4 9	16. 6	16.4	-0.2	1.2
35	9:22:4 1	19.5	20.1	0.6	3.08	85	9:49:2 1	16. 5	18.5	2	12.12
36	9:23:1 3	19.4	17.3	-2.1	10.82	86	9:49:5 3	16. 4	17.6	1.2	7.32

37	9:23:45	19.4	17.8	-1.6	8.25	87	9:50:25	16.2	17.8	1.6	9.88
38	9:24:17	19.3	17.8	-1.5	7.77	88	9:50:57	16.1	16.9	0.8	4.97
39	9:24:49	19.3	18	-1.3	6.74	89	9:51:29	16	15.5	-0.5	3.13
40	9:25:21	19.2	19	-0.2	1.04	90	9:52:01	15.9	16.1	0.2	1.26
41	9:25:53	19.2	17.5	-1.7	8.85	91	9:52:33	15.8	16.8	1	6.33
42	9:26:25	19.1	19.3	0.2	1.05	92	9:53:05	15.7	17.7	2	12.74
43	9:26:57	19.1	18.7	-0.4	2.09	93	9:53:37	15.6	16	0.4	2.56
44	9:27:29	19	17.3	-1.7	8.95	94	9:54:09	15.5	16.7	1.2	7.74
45	9:28:01	19	20.7	1.7	8.95	95	9:54:41	15.4	17.4	2	12.99
46	9:28:33	19	18.5	-0.5	2.63	96	9:55:13	15.3	17.4	2.1	13.73
47	9:29:05	18.9	18.8	-0.1	0.53	97	9:55:45	15.3	16.4	1.1	7.19
48	9:29:37	18.9	19.2	0.3	1.59	98	9:56:17	15.2	15.2	0	0
49	9:30:09	18.8	17.9	-0.9	4.79	99	9:56:49	15.1	17.1	2	13.25
50	9:30:41	18.8	17.5	-1.3	6.91	100	9:57:21	14.9	16.9	2	13.42

En la tabla 12 se realiza la comparación entre la humedad relativa medida por el sensor DHT22 y la medida por el HTC-2 Termohigrómetro.

Tabla 12- Comparación de la prueba de humedad.

N° Dato	Hora	Prototipo (%)	HT C-2 (%)	Diferencia (%)	Error % Aprox	N° Dato	Hora	Prototipo (%)	HTC-2 (%)	Diferencia (%)	Error % Aprox
1	12:15:23	75	75.6	-0.6	0.79	51	12:45:24	82.6	82.3	0.3	0.39
2	12:16:00	75.2	75.2	0	0.02	52	12:46:00	82.7	83.1	-0.4	0.44
3	12:16:36	75.3	74.9	0.4	0.55	53	12:46:36	82.9	82	0.9	1.13
4	12:17:12	75.5	75.1	0.4	0.47	54	12:47:12	83	83.4	-0.4	0.49

5	12:17: 48	75.6	75.2	0.4	0.57	55	12:47: 48	83. 2	83.4	-0.2	0.23
6	12:18: 24	75.8	75.9	-0.1	0.11	56	12:48: 24	83. 3	82.6	0.7	0.86
7	12:19: 00	75.9	75.5	0.4	0.49	57	12:49: 00	83. 5	84.4	-0.9	1.04
8	12:19: 36	76.1	75.3	0.8	1.08	58	12:49: 36	83. 6	84.2	-0.6	0.73
9	12:20: 12	76.2	77	-0.8	1.10	59	12:50: 12	83. 8	83.2	0.6	0.70
10	12:20: 48	76.4	77	-0.6	0.81	60	12:50: 48	83. 9	83.9	0	0.05
11	12:21: 24	76.5	76.3	0.2	0.26	61	12:51: 24	84. 1	83.1	1	1.16
12	12:22: 00	76.7	77.6	-0.9	1.13	62	12:52: 00	84. 2	84.7	-0.5	0.65
13	12:22: 36	76.8	76.3	0.5	0.68	63	12:52: 36	84. 4	84.9	-0.5	0.64
14	12:23: 12	77	76.9	0.1	0.18	64	12:53: 12	84. 5	84.6	-0.1	0.11
15	12:23: 48	77.1	77.2	-0.1	0.14	65	12:53: 48	84. 7	85.4	-0.7	0.84
16	12:24: 24	77.3	77.7	-0.4	0.54	66	12:54: 24	84. 8	84.1	0.7	0.78
17	12:25: 00	77.4	78.4	-1	1.22	67	12:55: 00	85	84.4	0.6	0.66
18	12:25: 36	77.6	78.1	-0.5	0.65	68	12:55: 36	85. 2	85.3	-0.1	0.12
19	12:26: 12	77.7	77	0.7	0.92	69	12:56: 12	85. 3	85.3	0	0.04
20	12:26: 48	77.9	76.9	1	1.36	70	12:56: 48	85. 5	85.2	0.3	0.30
21	12:27: 24	78	77.3	0.7	0.93	71	12:57: 24	85. 6	85.5	0.1	0.06
22	12:28: 00	78.2	78.1	0.1	0.13	72	12:58: 00	85. 8	85.8	0	0.00
23	12:28: 36	78.3	79.3	-1	1.28	73	12:58: 36	85. 9	85.6	0.3	0.37
24	12:29: 12	78.5	79.3	-0.8	1.02	74	12:59: 12	86. 1	85.6	0.5	0.55
25	12:29: 48	78.6	79.3	-0.7	0.94	75	12:59: 48	86. 2	85.6	0.6	0.71
26	12:30: 24	78.8	78	0.8	1.08	76	13:00: 24	86. 4	87.5	-1.1	1.31
27	12:31: 00	78.9	79.2	-0.3	0.36	77	13:01: 00	86. 5	87	-0.5	0.63
28	12:31: 36	79.1	78.4	0.7	0.89	78	13:01: 36	86. 7	86.3	0.4	0.41
29	12:32: 12	79.2	78.6	0.6	0.76	79	13:02: 12	86. 8	86.9	-0.1	0.07

30	12:32: 48	79.4	79.5	-0.1	0.08	80	13:02: 48	87	87.5	-0.5	0.55
31	12:33: 24	79.5	79.8	-0.3	0.37	81	13:03: 24	87. 1	86.1	1	1.11
32	12:34: 00	79.7	80.1	-0.4	0.53	82	13:04: 00	87. 3	87.1	0.2	0.23
33	12:34: 36	79.8	80.4	-0.6	0.71	83	13:04: 36	87. 4	88.3	-0.9	1.02
34	12:35: 12	80	79.2	0.8	1.07	84	13:05: 12	87. 6	87.2	0.4	0.43
35	12:35: 48	80.2	79.8	0.4	0.47	85	13:05: 48	87. 7	87.2	0.5	0.52
36	12:36: 24	80.3	79.6	0.7	0.83	86	13:06: 24	87. 9	88.5	-0.6	0.71
37	12:37: 00	80.5	80.3	0.2	0.20	87	13:07: 00	88	88.7	-0.7	0.83
38	12:37: 36	80.6	81.4	-0.8	0.95	88	13:07: 36	88. 2	87.7	0.5	0.57
39	12:38: 12	80.8	80.5	0.3	0.33	89	13:08: 12	88. 3	88	0.3	0.39
40	12:38: 48	80.9	81.7	-0.8	0.95	90	13:08: 48	88. 5	88.4	0.1	0.16
41	12:39: 24	81.1	80.5	0.6	0.71	91	13:09: 24	88. 6	89.3	-0.7	0.80
42	12:40: 00	81.2	81.1	0.1	0.08	92	13:10: 00	88. 8	87.9	0.9	1.02
43	12:40: 36	81.4	82.4	-1	1.17	93	13:10: 36	88. 9	88.7	0.2	0.26
44	12:41: 12	81.5	80.9	0.6	0.70	94	13:11: 12	89. 1	88.2	0.9	1.07
45	12:41: 48	81.7	80.9	0.8	0.94	95	13:11: 48	89. 2	88.9	0.3	0.29
46	12:42: 24	81.8	81	0.8	0.99	96	13:12: 24	89. 4	89.4	0	0.00
47	12:43: 00	82	82.5	-0.5	0.55	97	13:13: 00	89. 5	90.6	-1.1	1.20
48	12:43: 36	82.1	83.1	-1	1.26	98	13:13: 36	89. 7	90.1	-0.4	0.39
49	12:44: 12	82.3	83.2	-0.9	1.12	99	13:14: 12	89. 8	90.5	-0.7	0.75
50	12:44: 48	82.4	82.6	-0.2	0.26	100	13:14: 48	89. 7	89.4	0.3	0.29

En la tabla 13 se realiza la comparación entre los sensores MQ-135 y MQ-2.

Tabla 13 - Comparación de la prueba de nivel de oxígeno.

Nº Dato	Hora	Prototipo MQ-135	MQ-2	Diferencia (%)	Error % Aprox	Nº Dato	Hora	Prototipo MQ-135	MQ-2	Diferencia (%)	Error % Aprox
1	11:04:02	99.6	98.6	1	1.00	51	11:06:52	98.3	96.4	1.9	1.93
2	11:04:03	99.7	97.2	2.5	2.51	52	11:06:55	99.2	97.1	2.1	2.12
3	11:04:04	99.7	98.8	0.9	0.90	53	11:06:57	99.5	97.8	1.7	1.71
4	11:04:05	99.5	97.5	2	2.01	54	11:07:00	99.4	97.2	2.2	2.21
5	11:04:06	99.5	96.9	2.6	2.61	55	11:07:02	99.4	96.9	2.5	2.52
6	11:04:07	99.4	98.1	1.3	1.31	56	11:07:05	99.4	97.5	1.9	1.91
7	11:04:08	99.4	97.8	1.6	1.61	57	11:07:07	99.6	98.1	1.5	1.51
8	11:04:09	99.4	98.5	0.9	0.90	58	11:07:11	99.7	97.6	2.1	2.11
9	11:04:10	99.4	96.7	2.7	2.72	59	11:07:13	99.8	98.3	1.5	1.50
10	11:04:11	99.5	98.9	0.6	0.60	60	11:07:15	99.5	97.8	1.7	1.71
11	11:04:12	99.5	97.3	2.2	2.21	61	11:07:17	61.3	58.7	2.6	4.24
12	11:04:13	99.5	98.2	1.3	1.31	62	11:07:20	61.3	59.1	2.2	3.59
13	11:04:14	99.5	97.6	1.9	1.91	63	11:07:23	30.1	28.4	1.7	5.65
14	11:04:17	61.7	59.4	2.3	3.73	64	11:07:27	19.6	18.1	1.5	7.65
15	11:04:20	38.5	36.8	1.7	4.42	65	11:07:31	9.6	8.7	0.9	9.38
16	11:04:22	16.2	15.1	1.1	6.79	66	11:07:35	5.9	5.3	0.6	10.17
17	11:04:26	16.2	14.9	1.3	8.02	67	11:07:39	4.6	4.1	0.5	10.87
18	11:04:27	67.8	65.2	2.6	3.83	68	11:07:44	4.2	3.8	0.4	9.52
19	11:04:28	67.8	64.8	3	4.42	69	11:07:49	4	3.6	0.4	10.00
20	11:04:50	80.5	78.1	2.4	2.98	70	11:07:53	3.9	3.5	0.4	10.26

21	11:04: 54	55.6	53.2	2.4	4.32	71	11:07: 56	3.9	3.5	0.4	10.26
22	11:04: 58	22.4	20.8	1.6	7.14	72	11:08: 00	42.2	40. 3	1.9	4.50
23	11:05: 02	10.5	9.7	0.8	7.62	73	11:08: 02	87.2	84. 6	2.6	2.98
24	11:05: 06	7.8	7.1	0.7	8.97	74	11:08: 04	92.1	89. 8	2.3	2.50
25	11:05: 10	4.7	4.2	0.5	10.64	75	11:08: 06	95.2	93. 1	2.1	2.21
26	11:05: 14	4.1	3.7	0.4	9.76	76	11:08: 08	97.1	95. 2	1.9	1.96
27	11:05: 17	4	3.6	0.4	10.00	77	11:08: 11	98.1	96. 4	1.7	1.73
28	11:05: 22	3.9	3.5	0.4	10.26	78	11:08: 13	98.7	96. 8	1.9	1.93
29	11:05: 25	3.8	3.4	0.4	10.53	79	11:08: 15	98.9	97. 1	1.8	1.82
30	11:05: 29	3.8	3.4	0.4	10.53	80	11:08: 17	99.4	97. 6	1.8	1.81
31	11:05: 32	3.8	3.5	0.3	7.89	81	11:08: 19	99.2	97. 3	1.9	1.91
32	11:05: 36	3.8	3.4	0.4	10.53	82	11:08: 23	61.1	58. 6	2.5	4.09
33	11:05: 40	3.8	3.5	0.3	7.89	83	11:08: 27	24.5	22. 8	1.7	6.94
34	11:05: 47	3.8	3.4	0.4	10.53	84	11:08: 30	11.3	10. 4	0.9	7.96
35	11:05: 50	42.1	40.1	2	4.75	85	11:08: 34	8.4	7.6	0.8	9.52
36	11:05: 54	56.1	53.8	2.3	4.10	86	11:08: 37	66.8	64. 2	2.6	3.89
37	11:05: 58	84.1	81.2	2.9	3.45	87	11:08: 41	87.8	85. 4	2.4	2.73
38	11:06: 00	90.3	87.8	2.5	2.77	88	11:08: 44	95.3	93. 2	2.1	2.20
39	11:06: 02	96.2	94.1	2.1	2.18	89	11:08: 47	97	95. 1	1.9	1.96
40	11:06: 06	59.3	56.8	2.5	4.22	90	11:08: 51	98.6	96. 8	1.8	1.83
41	11:06: 11	23.8	22.1	1.7	7.14	91	11:08: 55	99.2	97. 4	1.8	1.81
42	11:06: 15	11	10.1	0.9	8.18	92	11:08: 59	99.4	97. 6	1.8	1.81
43	11:06: 19	6.4	5.8	0.6	9.38	93	11:09: 02	99.5	97. 8	1.7	1.71
44	11:06: 26	4.8	4.3	0.5	10.42	94	11:09: 06	99.5	97. 7	1.8	1.81
45	11:06: 30	65.2	62.7	2.5	3.83	95	11:09: 10	99.5	97. 9	1.6	1.61

46	11:06: 36	48.9	46.8	2.1	4.29	96	11:09: 13	99.6	98. 1	1.5	1.51
47	11:06: 39	58.3	55.9	2.4	4.12	97	11:09: 15	99.7	98. 2	1.5	1.50
48	11:06: 42	84.7	82.1	2.6	3.07	98	11:09: 17	99.8	98. 4	1.4	1.40
49	11:06: 45	94.2	91.8	2.4	2.55	99	11:09: 20	99.7	98. 3	1.4	1.40
50	11:06: 48	96.3	94.2	2.1	2.18	100	11:09: 22	99.8	98. 5	1.3	1.30

- **Anexo 6**

En la figura 25, 26 y 27 se visualiza la sala musical (cabina insonorizada) se encuentra en Universidad Politécnica Salesiana Sede Guayaquil en el edificio D.



*Figura 25 - Sala musical de la Universidad Politécnica Salesiana Sede Guayaquil.*



*Figura 26 - Sala musical de la Universidad Politécnica Salesiana Sede Guayaquil.*



*Figura 27 - Sala musical de la Universidad Politécnica Salesiana Sede Guayaquil.*