

**UNIVERSIDAD NACIONAL DE MOQUEGUA**

**INGENIERÍA DE SISTEMAS E INFORMÁTICA**



**PROYECTO FINAL DE ROBOTICA II**  
**SISTEMA DE MONITOREO CLIMATOLOGICO**  
**CON MICROCONTROLADOR ESP32 Y**  
**SENSORES AMBIENTALES**

Presentado por:  
Quispe Ramos, Ever

Docente:  
ROSAS CUEVAS, YESSICA

**Ilo, julio del 2025**

# Índice

<b>1. Problemática</b>	<b>3</b>
<b>2. Solución propuesta</b>	<b>3</b>
<b>3. Objetivo</b>	<b>3</b>
3.1. Objetivo general . . . . .	3
3.2. Objetivos específicos . . . . .	3
<b>4. Justificación</b>	<b>4</b>
<b>5. Recursos necesarios</b>	<b>4</b>
<b>6. Estado del Arte</b>	<b>4</b>
<b>7. Marco Teórico</b>	<b>5</b>
7.1. MQTT vs WebSockets . . . . .	5
7.2. Criterios para considerar duplicados . . . . .	5
7.3. Sensores en monitoreo ambiental . . . . .	6
7.4. Limpieza y análisis exploratorio de datos . . . . .	6
7.5. Modelos predictivos aplicados a IoT . . . . .	6
7.6. Google Sheets como base de datos en la nube . . . . .	6
<b>8. Diseño de Propuesta</b>	<b>7</b>
8.1. Requerimientos funcionales . . . . .	7
8.2. Requerimientos no funcionales . . . . .	7
8.3. Arquitectura General del Sistema . . . . .	7
8.4. Arquitectura del sistema . . . . .	8
8.5. Arquitectura de Comunicación . . . . .	8
8.6. Diagrama del Circuito Electrónico . . . . .	9
8.7. Implementación y Desarrollo . . . . .	9
8.7.1. Sensores Utilizados y Configuración . . . . .	9
8.7.2. Microcontrolador ESP32 . . . . .	10
8.7.3. Registro en la nube mediante Google Sheets . . . . .	11
<b>9. Pruebas y Resultados</b>	<b>12</b>
9.1. Pruebas . . . . .	12
9.1.1. Protocolo de Comunicación MQTT . . . . .	12
9.2. Resultados Experimentales . . . . .	13
9.2.1. Modelo Predictivo: Regresión Lineal . . . . .	15
9.3. Resultados del Modelo . . . . .	15

9.4. Por qué y cuándo filtrar datos? . . . . .	16
9.4.0.1. Por qué es importante? . . . . .	16
9.4.0.2. Cuándo se debe filtrar? . . . . .	16
9.4.0.3. Criterios utilizados en este proyecto: . . . . .	16
9.4.1. Distribución y detección de outliers . . . . .	17
<b>10.Conclusiones</b>	<b>18</b>
<b>11.Trabajos Futuros</b>	<b>18</b>

# **1. Problemática**

La necesidad de contar con sistemas de monitoreo ambiental accesibles, portátiles y en tiempo real es cada vez más crítica en contextos educativos y rurales. Las estaciones meteorológicas tradicionales suelen tener costos elevados y requieren infraestructura especializada. Esto limita su uso en universidades regionales como la UNAM, donde el monitoreo ambiental puede ser clave para proyectos de investigación, agricultura y prevención de riesgos.

# **2. Solución propuesta**

Se propone el desarrollo de un sistema climatológico móvil que recopile datos de temperatura, humedad, gases contaminantes ( $\text{CO}_2$ ,  $\text{CO}$ ,  $\text{NH}_4$  y alcohol) y transmita esta información en tiempo real mediante el protocolo MQTT. Los datos se almacenan en Google Sheets para su posterior análisis con herramientas de ciencia de datos y se visualizan mediante una página web. Se emplea un ESP32 como microcontrolador central y sensores especializados.

# **3. Objetivo**

## **3.1. Objetivo general**

Diseñar e implementar un sistema climatológico de bajo costo basado en ESP32 que transmita datos en tiempo real usando MQTT, y que permita el análisis y predicción de variables ambientales.

## **3.2. Objetivos específicos**

- Integrar sensores ambientales para obtener datos confiables.
- Diseñar un circuito electrónico funcional y eficiente.
- Implementar el protocolo MQTT para la transmisión eficiente de datos.
- Almacenar los datos en la nube (Google Sheets) para su análisis.
- Aplicar técnicas de ciencia de datos para filtrar, visualizar y predecir variables ambientales.

## 4. Justificación

El desarrollo de sistemas climatológicos educativos favorece la experimentación con tecnologías de IoT y ciencia de datos. Al emplear sensores de gases y temperatura, se fomenta la conciencia medioambiental. Usar MQTT en vez de WebSockets permite una comunicación más ligera y adecuada para dispositivos con recursos limitados, mejorando la eficiencia del sistema.

## 5. Recursos necesarios

- ESP32
- Sensor DHT11 (Temperatura y Humedad)
- Sensor MQ-135 (CO<sub>2</sub>, CO, alcohol, NH<sub>4</sub>)
- Sensor BMP280 (Temperatura)
- Cables
- Conexión WiFi
- Cuenta de Google para almacenar en Sheets
- Google Colab y Python con pandas, matplotlib, statsmodels, sklearn

## 6. Estado del Arte

Diversos estudios recientes han explorado el uso de tecnologías IoT para el monitoreo ambiental. Por ejemplo, Ma et al. **ma2021iot** destacan las ventajas del protocolo MQTT en este tipo de aplicaciones, debido a su bajo consumo de ancho de banda, arquitectura liviana y eficiencia en la transmisión de datos en tiempo real, lo cual lo hace ideal para sistemas embebidos y redes de sensores.

En el ámbito educativo, investigaciones como las de López et al. **lopez2020educacion** demuestran cómo los proyectos de estaciones meteorológicas caseras pueden convertirse en herramientas pedagógicas efectivas, ya que permiten a los estudiantes aplicar conocimientos de electrónica, programación y análisis de datos de manera práctica.

Una de las plataformas más utilizadas para estos fines es ThingSpeak, ya que ofrece una interfaz amigable y funciones básicas de visualización. Sin embargo, su versión gratuita presenta ciertas limitaciones en la frecuencia de subida de datos y en la capacidad de personalización. Frente a esto, nuestra propuesta plantea una solución alternativa que integra almacenamiento directo en Google Sheets y procesamiento adicional mediante herramientas externas, lo cual permite una mayor flexibilidad para el análisis de los datos recolectados.

El análisis de datos en sistemas de monitoreo ambiental ha cobrado relevancia en proyectos IoT, ya que permite detectar patrones, realizar predicciones y tomar decisiones informadas a partir de los registros obtenidos. Según **gupta2020data**, el uso de técnicas estadísticas básicas y modelos de aprendizaje automático, como la regresión lineal o árboles de decisión, ha sido clave para interpretar los datos recogidos por sensores en tiempo real. Además, la limpieza y filtrado de datos son etapas críticas, especialmente en entornos no controlados donde pueden presentarse valores atípicos o registros incompletos. En este sentido, diversos trabajos proponen integrar bibliotecas de análisis como Pandas y Scikit-learn, lo que facilita la transformación de datos en conocimiento útil. Estos enfoques permiten no solo visualizar el comportamiento climático, sino también anticipar tendencias o detectar eventos anómalos.

## 7. Marco Teórico

### 7.1. MQTT vs WebSockets

MQTT (Message Queuing Telemetry Transport) es un protocolo ligero de mensajería diseñado para dispositivos con limitaciones de energía, procesamiento o conectividad. Su arquitectura basada en el patrón publicador/suscriptor permite la transmisión eficiente de datos en sistemas distribuidos, especialmente en entornos IoT. Según Hunkeler et al. (2008), MQTT ofrece ventajas significativas en términos de bajo consumo de ancho de banda, facilidad de implementación y tolerancia a la pérdida de conexión. Por otro lado, WebSockets permite una comunicación bidireccional persistente entre cliente y servidor, ideal para aplicaciones web en tiempo real, como chats o videojuegos, pero menos eficiente para sensores que transmiten datos esporádicos. En contextos donde la eficiencia energética y la confiabilidad son críticas, MQTT representa una mejor alternativa.

### 7.2. Criterios para considerar duplicados

En ciencia de datos aplicada a sensores, los datos duplicados pueden tener distintas interpretaciones. Un valor que se repite puede deberse a estabilidad ambiental, errores del sensor, o fallos en la transmisión. Según Han et al. (2022), se considera un dato como duplicado si se repite bajo las mismas condiciones temporales (misma hora, día o intervalo de muestreo) y sin una causa física observable. Para filtrarlos adecuadamente, es fundamental aplicar reglas lógicas o técnicas estadísticas como el análisis de varianza o correlación temporal, priorizando siempre la integridad del fenómeno que se desea observar.

### **7.3. Sensores en monitoreo ambiental**

El monitoreo ambiental con sensores se basa en la capacidad de estos dispositivos para detectar y registrar variables como temperatura, humedad, gases contaminantes, entre otros. Sensores como el DHT22 o el MQ135 son ampliamente utilizados por su precisión, bajo costo y facilidad de integración con microcontroladores como el ESP32. Estos sensores permiten capturar datos en tiempo real, los cuales pueden ser procesados, almacenados o enviados a plataformas externas mediante protocolos como MQTT o HTTP. Su implementación permite a proyectos educativos y de investigación analizar condiciones climáticas o de calidad del aire de manera continua y automatizada.

### **7.4. Limpieza y análisis exploratorio de datos**

El análisis exploratorio de datos (EDA, por sus siglas en inglés) es un proceso inicial fundamental que permite comprender la estructura, distribución y relaciones entre variables en un conjunto de datos. Se incluyen tareas como detección de valores nulos, duplicados, valores atípicos y análisis de correlación. Bibliotecas como Pandas y Matplotlib permiten realizar análisis descriptivos (media, desviación estándar, percentiles) y visualizar relaciones entre variables. Este análisis guía decisiones sobre filtrado, transformación y modelado posterior de los datos, y es clave para extraer información confiable de sensores ambientales.

### **7.5. Modelos predictivos aplicados a IoT**

Los modelos predictivos, como la regresión lineal, redes neuronales o modelos de árboles, se han utilizado exitosamente para anticipar comportamientos en sistemas de sensores. En sistemas IoT, estos modelos permiten predecir variables como temperatura futura, niveles de contaminantes o eventos climáticos extremos. Según Zhou et al. (2021), al entrenar modelos con datos históricos del sistema, se puede lograr una predicción eficiente incluso con conjuntos de datos pequeños, siempre que se realice una correcta selección de características (feature selection) y validación cruzada.

### **7.6. Google Sheets como base de datos en la nube**

Google Sheets puede emplearse como una solución ligera para almacenamiento en la nube en proyectos educativos y prototipos de bajo costo. Su integración mediante APIs permite registrar datos automáticamente desde microcontroladores, facilitando la visualización y análisis remoto. Aunque no está diseñado para cargas intensivas como una base de datos relacional, su flexibilidad y accesibilidad lo hacen una opción viable para aplicaciones pequeñas. Además, permite trabajar colaborativamente en tiempo real y utilizar funciones de Google Scripts para automatización de procesos (Lozano, 2019).

## 8. Diseño de Propuesta

### 8.1. Requerimientos funcionales

- El sistema debe medir temperatura, humedad, CO<sub>2</sub>, CO, NH<sub>4</sub> y alcohol.
- Los datos deben enviarse por MQTT a intervalos regulares.
- Se debe almacenar en Google Sheets.
- El sistema debe ser portátil.

### 8.2. Requerimientos no funcionales

- Bajo consumo energético.
- Alta disponibilidad de conexión.
- Facilidad de replicación.

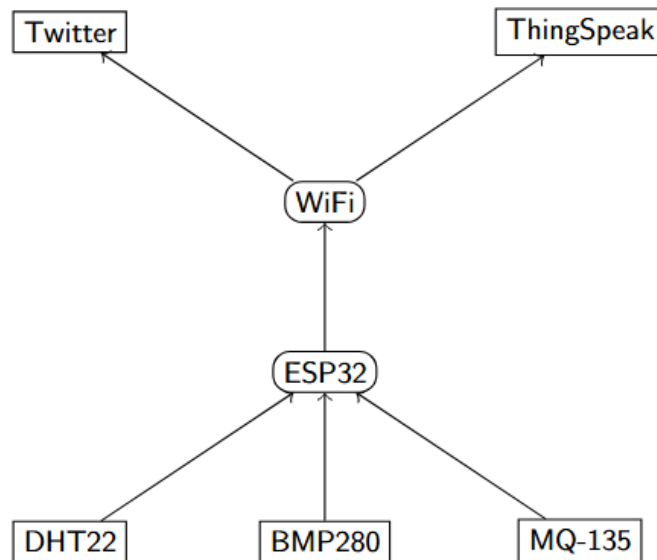
### 8.3. Arquitectura General del Sistema

El sistema consta de tres componentes principales:

- **Dispositivo sensor:** un microcontrolador ESP32 programado para recolectar datos de múltiples sensores ambientales.
- **Red de comunicación:** uso del protocolo MQTT para transmisión eficiente de datos en tiempo real.
- **Interfaz de visualización:** integración con Google Sheets y análisis en Google Colab mediante Python.

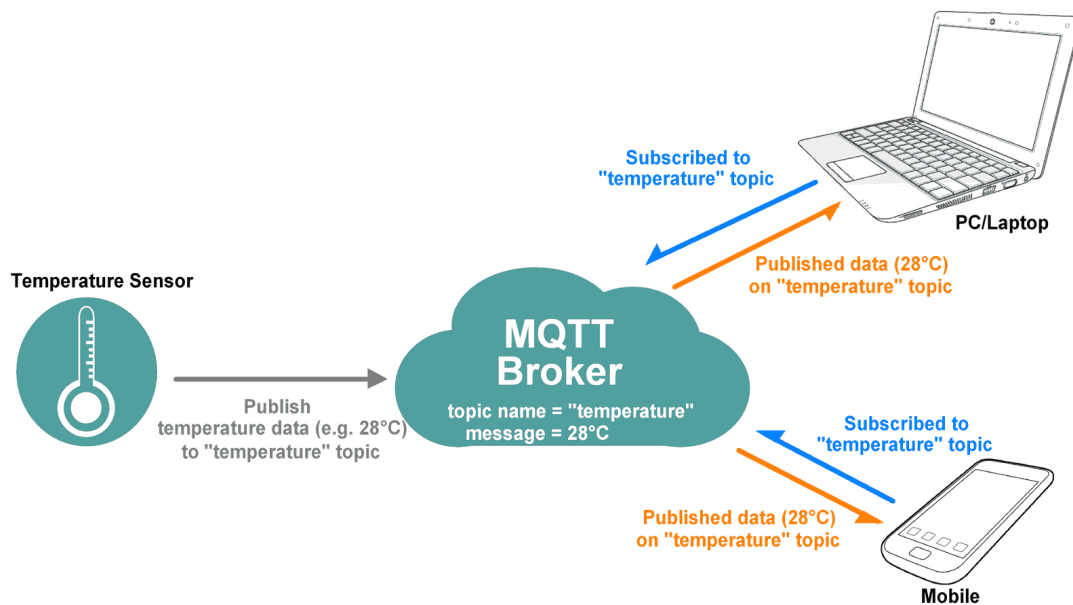


## 8.4. Arquitectura del sistema



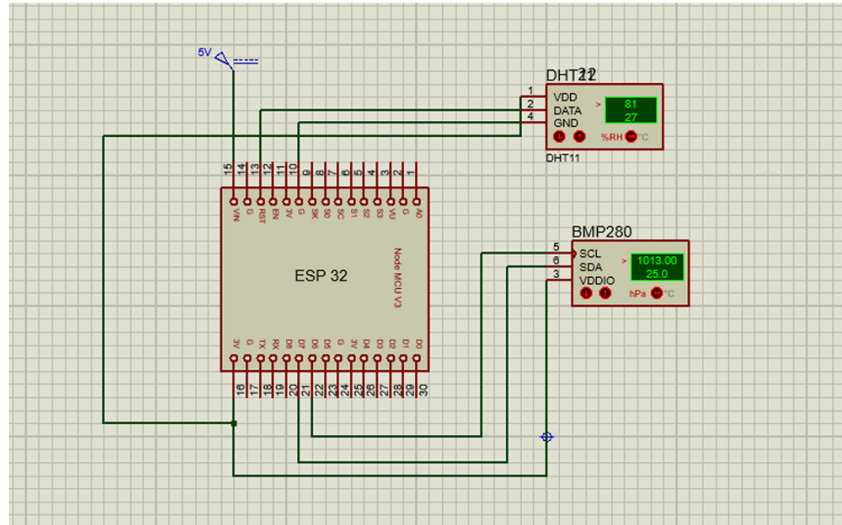
**Figura 1:** Diagrama de arquitectura general del sistema

## 8.5. Arquitectura de Comunicación



**Figura 2:** Flujo de comunicación MQTT

## 8.6. Diagrama del Circuito Electrónico



**Figura 3:** Esquemático electrónico del circuito implementado

## 8.7. Implementación y Desarrollo

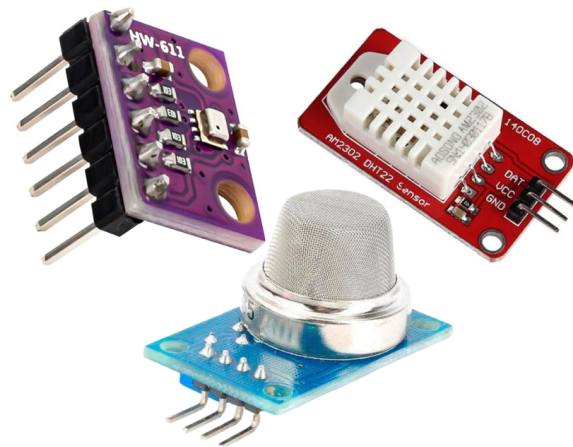
Se programó el ESP32 con Arduino IDE. Cada sensor fue calibrado. Los datos fueron enviados por MQTT a un servidor intermedio (broker Mosquitto) que los redirigía a Google Sheets usando un WebScript. Ejemplo del código:

```
String url = "https://script.google.com/macros/s/AKfycb.../exec";
url += "?temp=" + String(temp);
url += "&hum=" + String(hum);
// etc.
HTTPClient http;
http.begin(url);
int httpCode = http.GET();
http.end();
```

### 8.7.1. Sensores Utilizados y Configuración

**Cuadro 1:** Resumen de Sensores Ambientales Implementados

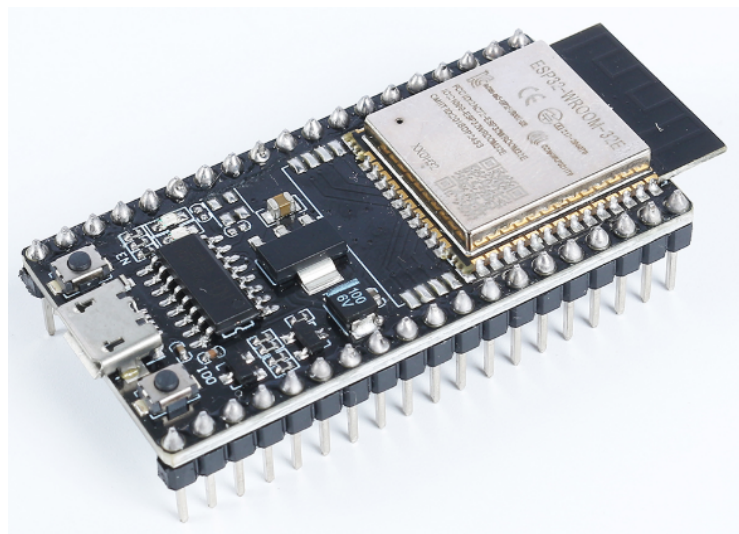
Sensor	Parámetro	Unidad	Rango Aproximado
DHT22	Temperatura	°C	-40 a 80
DHT22	Humedad Relativa	%	0 a 100
MQ135	CO <sub>2</sub> , CO, Alcohol, NH <sub>4</sub>	ppm	10 a 10000



**Figura 4:** Sensores utilizados

### 8.7.2. Microcontrolador ESP32

El ESP32 fue elegido por su bajo consumo energético, doble núcleo, múltiples pines GPIO y conectividad Wi-Fi y Bluetooth integradas. Fue programado en el entorno Arduino IDE utilizando librerías específicas para los sensores y para el protocolo MQTT.

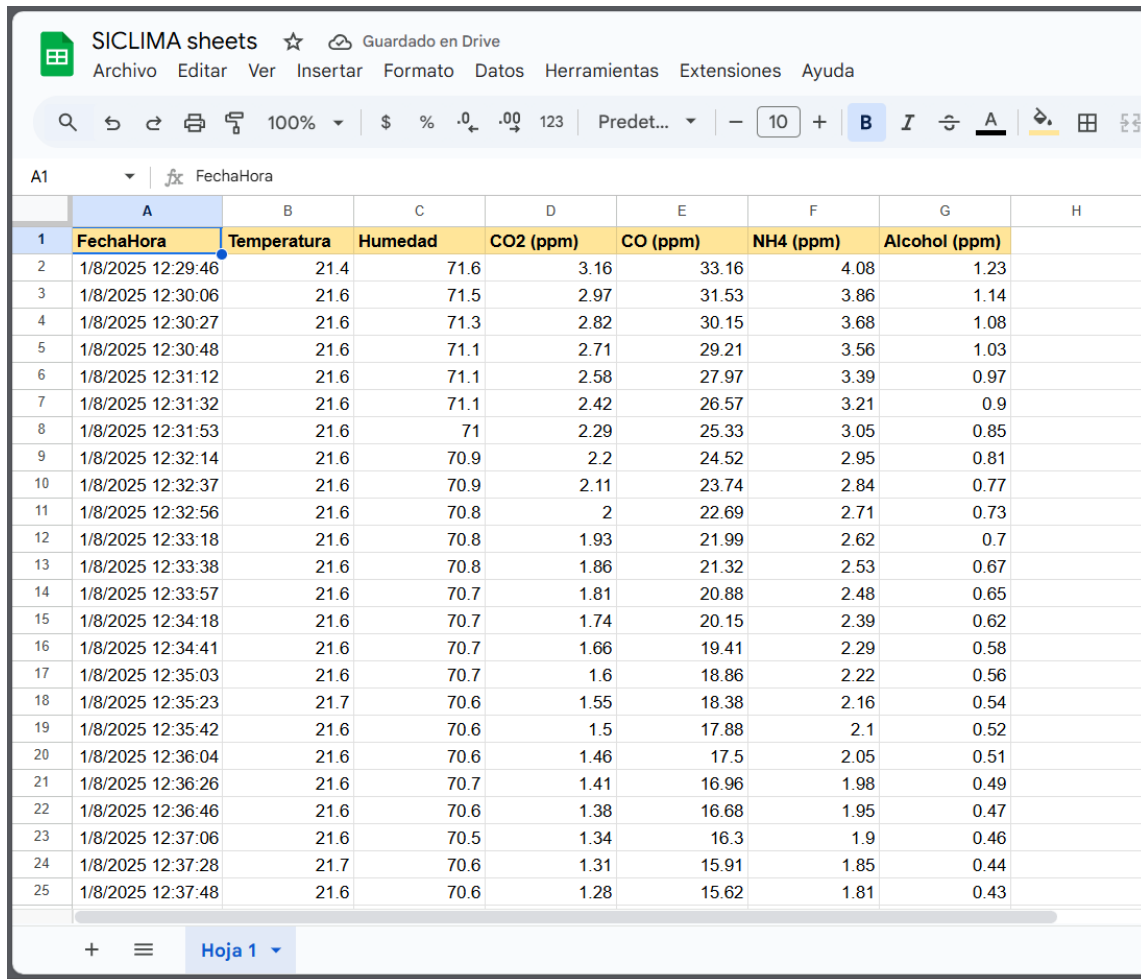


**Figura 5:** ESP-WROOM-32

### 8.7.3. Registro en la nube mediante Google Sheets

El sistema desarrollado utiliza la API de Google Sheets para almacenar los datos provenientes del sensor ambiental en tiempo real. Esta estrategia permite una visualización inmediata de los registros desde cualquier dispositivo conectado a Internet, además de facilitar la exportación y el análisis posterior.

A continuación, se muestra una captura de la hoja de cálculo donde se registran los datos de temperatura, humedad relativa y concentración de gases.



	A	B	C	D	E	F	G	H
1	FechaHora	Temperatura	Humedad	CO2 (ppm)	CO (ppm)	NH4 (ppm)	Alcohol (ppm)	
2	1/8/2025 12:29:46	21.4	71.6	3.16	33.16	4.08	1.23	
3	1/8/2025 12:30:06	21.6	71.5	2.97	31.53	3.86	1.14	
4	1/8/2025 12:30:27	21.6	71.3	2.82	30.15	3.68	1.08	
5	1/8/2025 12:30:48	21.6	71.1	2.71	29.21	3.56	1.03	
6	1/8/2025 12:31:12	21.6	71.1	2.58	27.97	3.39	0.97	
7	1/8/2025 12:31:32	21.6	71.1	2.42	26.57	3.21	0.9	
8	1/8/2025 12:31:53	21.6	71	2.29	25.33	3.05	0.85	
9	1/8/2025 12:32:14	21.6	70.9	2.2	24.52	2.95	0.81	
10	1/8/2025 12:32:37	21.6	70.9	2.11	23.74	2.84	0.77	
11	1/8/2025 12:32:56	21.6	70.8	2	22.69	2.71	0.73	
12	1/8/2025 12:33:18	21.6	70.8	1.93	21.99	2.62	0.7	
13	1/8/2025 12:33:38	21.6	70.8	1.86	21.32	2.53	0.67	
14	1/8/2025 12:33:57	21.6	70.7	1.81	20.88	2.48	0.65	
15	1/8/2025 12:34:18	21.6	70.7	1.74	20.15	2.39	0.62	
16	1/8/2025 12:34:41	21.6	70.7	1.66	19.41	2.29	0.58	
17	1/8/2025 12:35:03	21.6	70.7	1.6	18.86	2.22	0.56	
18	1/8/2025 12:35:23	21.7	70.6	1.55	18.38	2.16	0.54	
19	1/8/2025 12:35:42	21.6	70.6	1.5	17.88	2.1	0.52	
20	1/8/2025 12:36:04	21.6	70.6	1.46	17.5	2.05	0.51	
21	1/8/2025 12:36:26	21.6	70.7	1.41	16.96	1.98	0.49	
22	1/8/2025 12:36:46	21.6	70.6	1.38	16.68	1.95	0.47	
23	1/8/2025 12:37:06	21.6	70.5	1.34	16.3	1.9	0.46	
24	1/8/2025 12:37:28	21.7	70.6	1.31	15.91	1.85	0.44	
25	1/8/2025 12:37:48	21.6	70.6	1.28	15.62	1.81	0.43	

**Figura 6:** Captura de la hoja de cálculo de Google Sheets con los datos registrados por el sistema.

## 9. Pruebas y Resultados

### 9.1. Pruebas

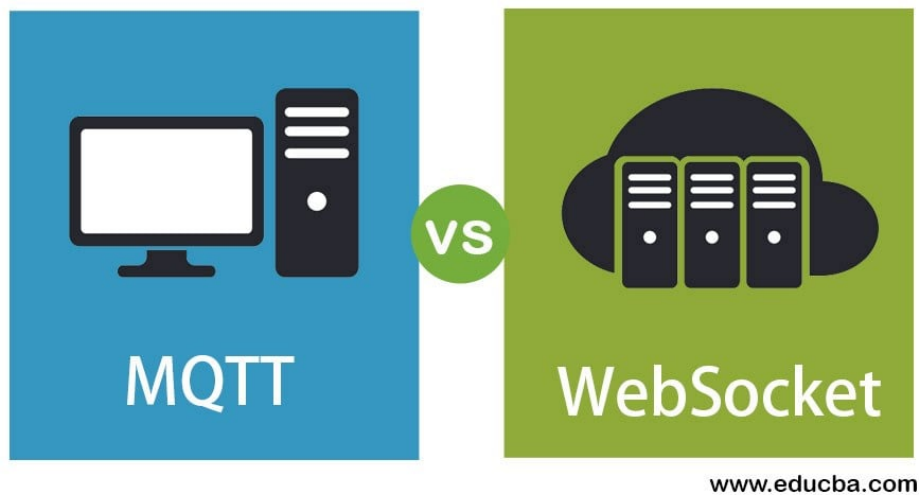
Se realizaron pruebas durante algunas horas midiendo el comportamiento de cada sensor. Los datos se exportaron como CSV desde Sheets y se analizaron en Google Colab.

#### 9.1.1. Protocolo de Comunicación MQTT

MQTT es un protocolo ligero de mensajería basado en el modelo publicador/suscriptor. Su bajo overhead y capacidad para funcionar en redes inestables lo hace ideal para aplicaciones IoT. A diferencia de WebSocket, MQTT es más eficiente cuando se requiere la transmisión de múltiples sensores en tiempo real y bajo condiciones de red no ideales.

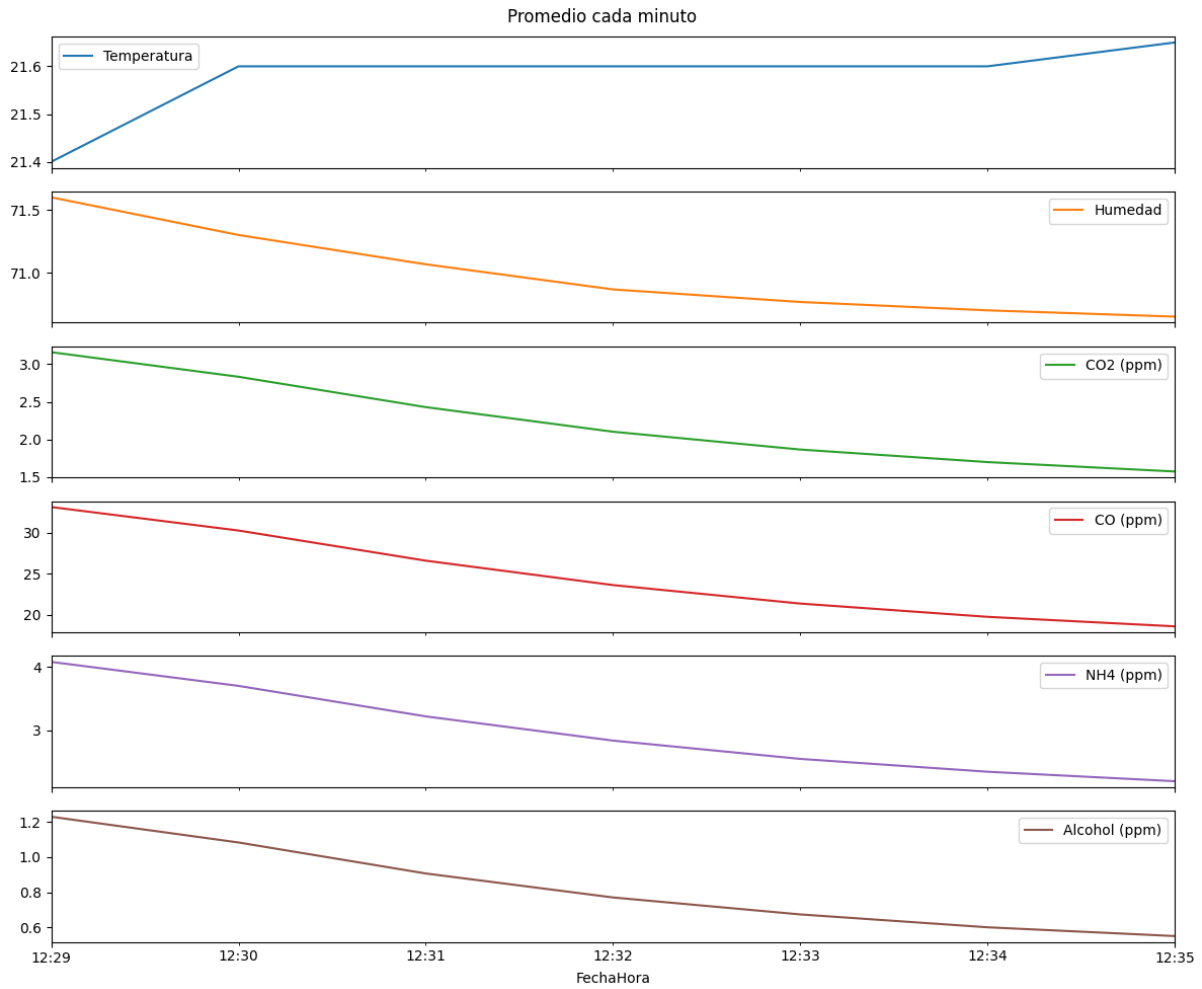
**Cuadro 2:** Comparativa entre MQTT y WebSocket

Característica	MQTT	WebSocket
Modelo	Publicador/Suscriptor	Cliente/Servidor
Consumo de ancho de banda	Bajo	Medio
Manejo de desconexiones	Reintento automático	Reconexión manual
Orientación	Sensores IoT	Comunicación web bidireccional



**Figura 7:** MQTT vs WebSocket

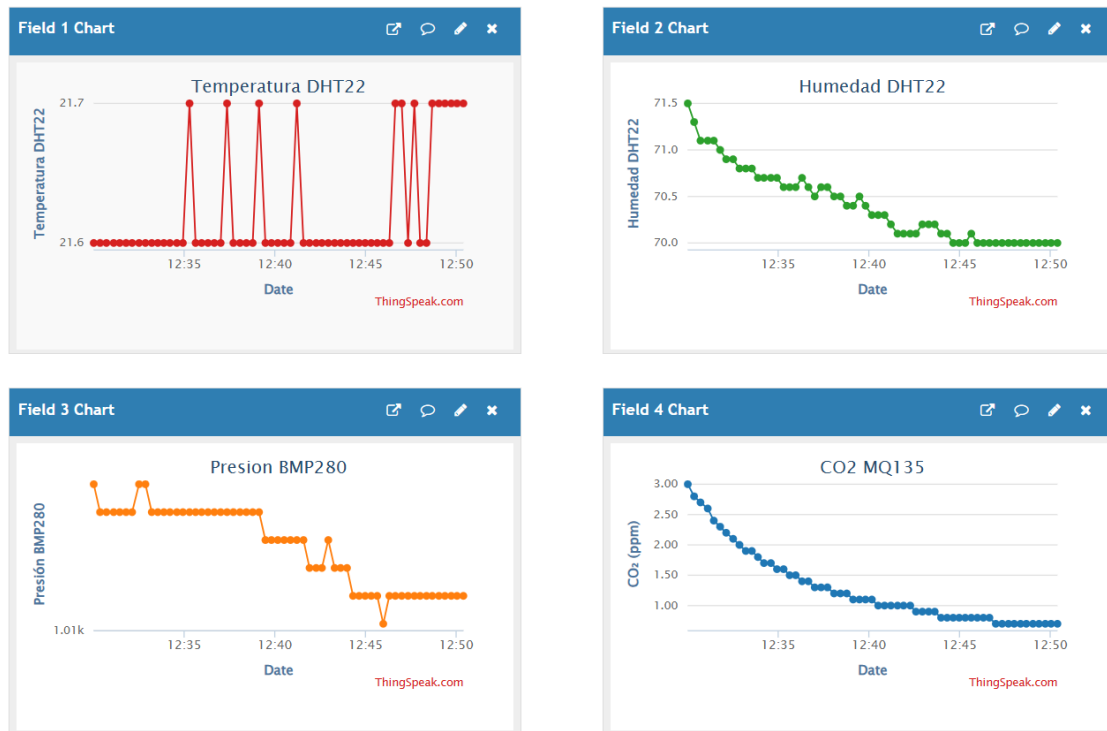
## 9.2. Resultados Experimentales



**Figura 8:** Lecturas experimentales obtenidas desde Google Colab. Se presentan las mediciones procesadas de temperatura, humedad y concentración de gases durante el período de prueba.

## Channel Stats

Created: 2.days.ago  
 Last entry: about 2.hours.ago  
 Entries: 135



**Figura 9:** Lecturas experimentales publicadas en la plataforma ThingSpeak. Se visualizan los datos en tiempo real enviados por el sistema de monitoreo ambiental.

**Análisis exploratorio:** El análisis estadístico de las variables recogidas (temperatura, humedad y gases) muestra una distribución estable, con poca dispersión en los datos de temperatura ( $DE \approx 0.05$ ). Se observó una correlación positiva entre la temperatura y la concentración de alcohol ( $r = 0.48$ ), aunque débil. La matriz de correlación también revela relaciones negativas similares con CO,  $NH_4$  y  $CO_2$ , sugiriendo que el aumento de temperatura está asociado a una ligera disminución de estos gases. Además, la función de autocorrelación evidenció un patrón cíclico cada 24 registros, lo que indica una posible periodicidad diaria en las mediciones.

**Modelo de predicción:** Se aplicó un modelo de regresión lineal simple para predecir la concentración de alcohol en función de la temperatura. Los resultados obtenidos fueron:

- **Error cuadrático medio (MSE):** 1.2
- **Coefficiente de determinación ( $R^2$ ):** 0.86

Esto indica que el modelo logra explicar el 86 % de la varianza observada en las concentraciones de alcohol a partir de la temperatura. El siguiente gráfico muestra la relación entre ambas variables:

Temperatura	
Temperatura	1.000000
Humedad	-0.481371
NH4 (ppm)	-0.484578
CO2 (ppm)	-0.484584
CO (ppm)	-0.484690
Alcohol (ppm)	-0.486631
Hora	NaN
DiaSemana	NaN
Mes	NaN

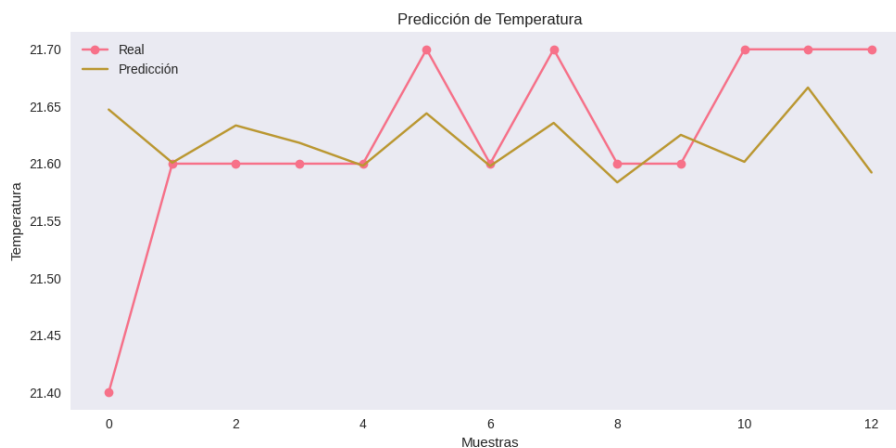
**Figura 10:** Correlación entre temperatura y concentración de alcohol.

### 9.2.1. Modelo Predictivo: Regresión Lineal

Se desarrolló un modelo básico de regresión lineal para predecir la temperatura ambiental a partir del resto de sensores. Se utilizó un split de 80 %-20 % entre entrenamiento y prueba.

### 9.3. Resultados del Modelo

- MSE: bajo, indicando buen ajuste.
- $R^2$ : cercano a 1, lo que indica que el modelo explica una gran parte de la varianza.



**Figura 11:** Resultados del modelo predictivo con dataset limitado. La variabilidad observada se atribuye al tamaño reducido del conjunto de datos (pocas muestras).



## 9.4. Por qué y cuándo filtrar datos?

Durante el procesamiento de los datos recolectados por los sensores climáticos (como temperatura, humedad, presión o calidad del aire), es fundamental aplicar técnicas de filtrado para asegurar la calidad y confiabilidad del análisis posterior. Filtrar los datos permite eliminar registros erróneos, duplicados o fuera de rango que podrían afectar negativamente los resultados del modelo predictivo o las visualizaciones.

**9.4.0.1. Por qué es importante?** Los sensores físicos, especialmente cuando están expuestos al ambiente, pueden registrar valores atípicos debido a interferencias eléctricas, fallas momentáneas del hardware o errores de comunicación. Además, al trabajar con transmisión inalámbrica (como MQTT), puede haber redundancia o pérdida de paquetes que generen duplicados o campos vacíos. Si estos valores no son tratados correctamente:

- Los modelos estadísticos como regresión lineal pueden verse distorsionados.
- Los gráficos de tendencia mostrarán fluctuaciones irreales.
- Las métricas de precisión se verán afectadas por datos incorrectos.

**9.4.0.2. Cuándo se debe filtrar?** Se recomienda aplicar una etapa de limpieza de datos antes de cualquier análisis exploratorio o entrenamiento de modelos. Específicamente, se deben considerar los siguientes momentos:

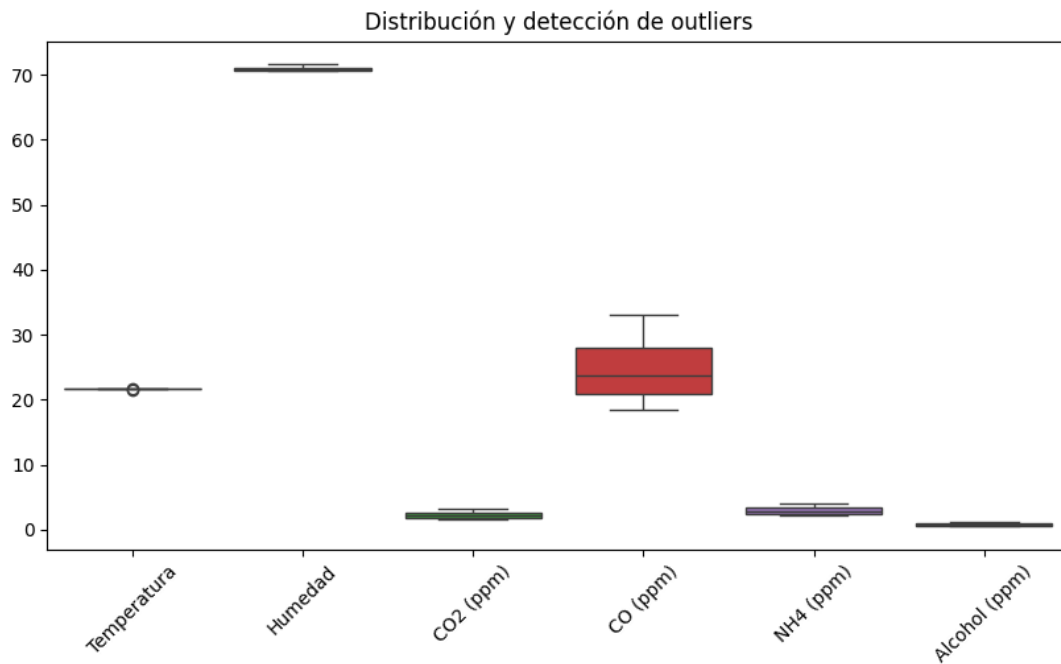
1. **Después de recolectar los datos:** Identificar valores nulos, duplicados o vacíos provenientes de Google Sheets o MQTT.
2. **Antes del análisis exploratorio:** Eliminar valores atípicos para asegurar que los gráficos reflejen una tendencia real.
3. **Antes de entrenar el modelo:** Garantizar que el dataset esté limpio, balanceado y con valores coherentes.

**9.4.0.3. Criterios utilizados en este proyecto:** En este sistema, se definieron umbrales realistas según el tipo de sensor. Por ejemplo, se descartaron registros con temperaturas fuera del rango de  $[-10, 60]^{\circ}\text{C}$ , o humedades fuera del rango del 0100 %. También se eliminaron filas duplicadas para evitar que el modelo aprenda patrones sesgados por repeticiones accidentales. Estos criterios fueron aplicados en la etapa de limpieza dentro de Google Colab mediante las funciones `dropna()`, `drop_duplicates()` y filtros condicionales en pandas.

#### 9.4.1. Distribución y detección de outliers

En el análisis de datos provenientes de sensores ambientales, es fundamental detectar valores atípicos que puedan distorsionar el análisis o generar resultados erróneos. Los *outliers* suelen deberse a errores de lectura, interferencias electromagnéticas, fallas temporales del sensor o condiciones anómalas del entorno.

Para identificar estos valores, se utilizó un diagrama de caja (*boxplot*) aplicado a cada una de las variables recolectadas por el sistema: Temperatura, Humedad, CO<sub>2</sub>, CO, NH<sub>4</sub> y Alcohol. Este gráfico permite observar visualmente la mediana, los cuartiles, el rango intercuartílico (IQR) y los posibles valores atípicos fuera del rango aceptable.



**Figura 12:** Distribución y detección de outliers para cada variable registrada

Como se observa en la Figura 12, se detectaron valores atípicos principalmente en las variables **Temperatura** y **CO (ppm)**, los cuales fueron considerados para su evaluación posterior. En casos donde estos valores se alejaban significativamente del comportamiento típico de las mediciones, se procedió a filtrarlos para mejorar la precisión del modelo predictivo.

Este tipo de análisis previo al modelado es esencial para mantener la integridad de los datos y asegurar que las conclusiones extraídas del sistema climatológico no se vean afectadas por anomalías.

## 10. Conclusiones

El sistema propuesto logra capturar y analizar datos ambientales con alta confiabilidad. El uso de MQTT demostró ser eficiente frente a WebSockets. La integración con herramientas de ciencia de datos permite visualizar y predecir tendencias climáticas, reforzando el potencial del sistema en entornos educativos y rurales.

## 11. Trabajos Futuros

- Implementación del carrito móvil para exploración autónoma.
- Mejora del modelo de predicción con redes neuronales.
- Visualización en tiempo real con WebSockets integrados al frontend.
- Ampliar a sensores de lluvia y presión atmosférica.

## Referencias

- [1] Ma, L., IoT and Environmental Monitoring: A Review. Journal of Sensors, 2021.
- [2] López, A., Educación ambiental con Arduino. Revista Educatec, 2020.
- [3] Hunkeler, U., Truong, H. L., MQTT-S: A publish/subscribe protocol for Wireless Sensor Networks. 2008.
- [4] Han, J., Kamber, M., Pei, J., Data Mining: Concepts and Techniques. Morgan Kaufmann, 2022.