

Sistema de Monitorización Ambiental Inteligente y Datalogger Utilizando ESP32 y Firebase.

Andrade Gómez, Leidy Jhoana y Catachunga Lugo, Camilo Alejandro

Institución Universitaria Antonio José Camacho

ljandrade@estudiante.uniajc.edu.co, ccatachunga@estudiante.uniajc.edu.co

Resumen— En este trabajo se presenta un sistema de monitoreo ambiental inteligente y Datalogger utilizando la plataforma ESP32 y una interfaz web conectada con la base de datos en tiempo real de Firebase (Real Time Database). El sistema permite la recolección de datos ambientales como temperatura ambiente, humedad relativa, calidad del aire, entre otros, de manera remota y en tiempo real. Además, se propone una arquitectura que permite la visualización de estos datos de manera sencilla y accesible para el usuario a través de una interfaz web, además del análisis e interpretación de la data almacenada haciendo uso del lenguaje Python.

Abstract-- This work presents an intelligent environmental monitoring system and Datalogger using the ESP32 platform and a web interface connected to the Firebase real-time database (Real Time Database). The system allows the collection of environmental data such as ambient temperature, relative humidity, air quality, among others, remotely and in real time. Furthermore, an architecture is proposed that allows the visualization of this data in a simple and accessible way for the user through a web interface, in addition to the analysis and interpretation of the stored data using the Python language.

Palabras clave: ESP32, Datalogger, RTDB, Firebase, Desarrollo Web.

I. INTRODUCCIÓN

En la actualidad, la importancia de obtener y manejar variables ambientales ha tomado mayor relevancia en la sociedad. Por ello, es crucial contar con sistemas de monitoreo ambiental que permitan recolectar y almacenar información de manera precisa y oportuna. En este sentido, la integración de tecnologías como el ESP32 y Firebase ofrece una solución eficiente y accesible para la creación de un sistema de monitoreo ambiental inteligente.

El ESP32 es un microcontrolador de bajo costo y alto rendimiento que cuenta con capacidades de conectividad Wi-Fi y Bluetooth, lo que lo hace ideal para aplicaciones IoT. Por otro lado, Firebase es una plataforma en la nube que permite el almacenamiento y la sincronización de datos en tiempo real. La combinación de estas dos tecnologías permite la creación de un sistema de monitoreo ambiental inteligente que puede ser controlado y visualizado a través de una interfaz web ya sea en un PC o un dispositivo móvil desde cualquier parte del mundo.

II. METODOLOGÍA

A. EQUIPOS Y MATERIALES

- **Microcontrolador ESP32 Wroom32:** es un módulo de desarrollo basado en el microcontrolador ESP32 de Espressif Systems. Este módulo incluye conectividad Wi-Fi 802.11 b/g/n y Bluetooth 4.2 (clásico y BLE). Con una arquitectura de doble núcleo Tensilica LX6, opera a una velocidad de hasta 240 MHz y cuenta con 520 KB de SRAM. Proporciona una amplia gama de interfaces, incluyendo UART, SPI, I2C, PWM, ADC (12 bits, hasta 18 canales), y DAC (8 bits). Este módulo es ideal para aplicaciones IoT debido a su bajo consumo de energía y alta capacidad de procesamiento.

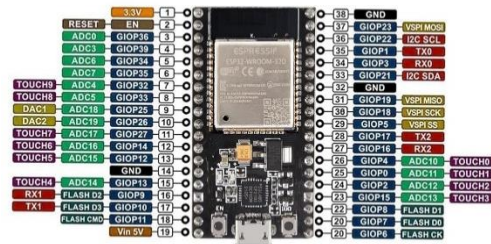


Fig. 1 PINOUT ESP32 Wroom 32.

- **Sensores**

DHT22: es un sensor digital de temperatura y humedad de alta precisión. Utiliza un termistor y un sensor capacitivo para medir la humedad relativa y la temperatura. Ofrece una precisión de ± 0.5 °C para la temperatura y $\pm 2-5\%$ para la humedad. El rango de medición de temperatura es de -40 a 80 °C y para la humedad de 0 a 100%. La interfaz de comunicación es de un solo cable (single-wire) y la tasa de muestreo es de 0.5 Hz (una lectura cada 2 segundos).

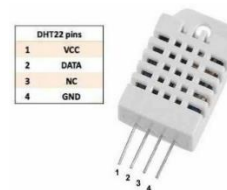


Fig. 2 Sensor DHT22.

BMP280: es un sensor de presión barométrica de alta precisión que también puede medir la temperatura. Fabricado por Bosch Sensortec, ofrece una precisión de ± 1 hPa para la presión y ± 1 °C para la temperatura. El rango de medición de presión es de 300 a 1100 hPa, lo que lo hace adecuado para altitudes desde -500 hasta 9000 metros. Se comunica a través de interfaces I2C o SPI, lo que proporciona flexibilidad en su integración con diferentes microcontroladores.

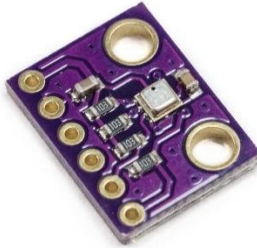


Fig. 3 Sensor BMP280.

MQ-135: es un sensor de gas diseñado para detectar una amplia gama de gases, incluyendo amoníaco, alcohol, benceno, humo y dióxido de carbono. Funciona basado en la conductividad de un material semiconductor que cambia en presencia de gases. Este sensor proporciona una salida analógica que puede ser leída por un ADC para determinar la concentración de los gases. La sensibilidad y el tiempo de respuesta pueden ajustarse según la aplicación específica.



Fig. 4 Sensor MQ-135.

- **Memoria**

Módulo lector de Memoria MicroSD: cuenta con interfaz SPI permite a los microcontroladores almacenar y leer grandes cantidades de datos en tarjetas Micro SD. Este módulo incluye un regulador de voltaje y un traductor de nivel para permitir la comunicación con tarjetas SD que funcionan a 3.3V, mientras que el microcontrolador puede operar a 5V, aunque en los ESP32 no hay inconvenientes ya que este trabaja a 3.3V para la lógica de sus pines. La interfaz SPI proporciona una forma rápida y eficiente de transferir datos, lo que es ideal para aplicaciones de registro de datos como lo es en este caso un Datalogger.

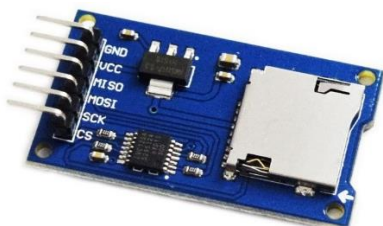


Fig. 5 Módulo lector de MicroSD.

- **Conexión Wi-Fi:** La conexión Wi-Fi es un componente crucial del sistema, permitiendo la transmisión de datos en tiempo real a la base de datos de Firebase.
- **Alimentación:** El sistema se alimenta desde un adaptador de pared de 5V el cual proporciona el voltaje y la corriente necesaria para que funcione correctamente el dispositivo, además se cuenta con una batería de respaldo de LiPO (Litio Polímero) de 7.4V con 1100 mAh, lo cual permite una autonomía del sistema de hasta 5 horas en caso de no contar con la corriente proporcionada por el adaptador.
- **Software**

IDE de Arduino: Toda la programación de los sensores, el módulo lector de microSD, la conexión y comunicación Wi-Fi para el enlace con Firebase se desarrolló en el IDE de Arduino, puesto que en este entorno se cuenta con una librería que facilita la comunicación del ESP32 con la base de datos en tiempo real *Real Time Database* de Firebase.

VSCode (Visual Studio Code): En este editor de código fuente se desarrolló la interfaz web haciendo uso de HTML, CSS y JavaScript, desde este entorno se realizó el “deploy” de la página o interfaz al hosting de Firebase.

Bibliotecas: algunas librerías o bibliotecas utilizadas han sido: *Firebase_ESP_Client*, *Adafruit_Sensor*, *DHT*, *SPI*, *SD*.

Servicios en la nube: Firebase es una plataforma de desarrollo de aplicaciones móviles y web respaldada por Google. Proporciona una variedad de herramientas y servicios diseñados para ayudar a los desarrolladores a crear aplicaciones de alta calidad de manera eficiente, unas de las herramientas utilizadas para el desarrollo del proyecto fueron el servicio de *Hosting* el cual nos permite el alojamiento de la plataforma web con certificado SSL, la otra herramienta fue RTDB o Real Time Database, la cual es una base de datos NoSQL alojada en la nube que permite almacenar y sincronizar datos en tiempo real entre los clientes y la nube.

B. DISEÑO DEL ESTUDIO

El sistema de monitoreo ambiental está diseñado para recopilar y reportar datos en tiempo real sobre las condiciones ambientales, incluyendo temperatura ambiente, humedad relativa, calidad del aire (PPM de CO₂), presión atmosférica y altitud. Los datos se envían a una base de datos en tiempo real (Firebase) cada 5 segundos y se almacenan localmente en una tarjeta microSD cada 5 minutos.

A continuación se proporciona un diagrama de flujo general del proceso anteriormente descrito (véase Fig.6):

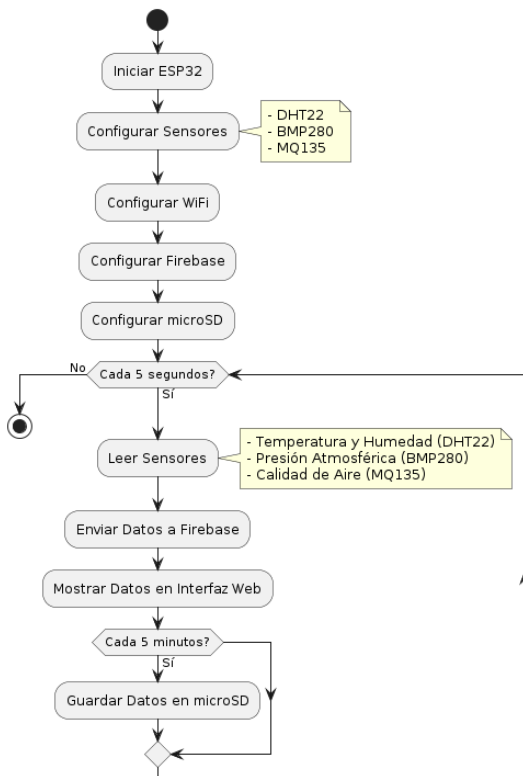


Fig. 6 Diagrama de flujo general del proceso.

C. PROCEDIMIENTOS

- **Interfaz Web** (véase Fig. 8, 9, 10): La Interfaz web desarrollada en VsCode se conecta a Firebase mediante unas credenciales y de esta manera arrastra la data que llega mediante Real Time Database, el ESP32 funciona como un sistema de adquisición de datos siendo este el API Backend del proyecto.

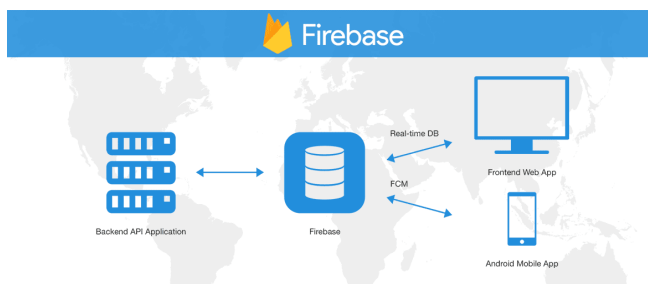


Fig. 7 Esquema de conexión Backend y Frontend.



Fig. 8 Interfaz de Inicio de sesión.



Fig. 9 Interfaz Web "Monitor de Variables Ambientales"



Fig. 10 Gráficas en tiempo real de Temperatura Vs. Tiempo y Humedad Vs. Tiempo.

- **Inicialización del sistema para la adquisición y reporte de los datos:**
 - Conexión del ESP32 a una red WI-FI.
 - Configuración de los sensores y la tarjeta MicroSD.
 - Sincronización de la hora con el servidor NTP: Una de las características esenciales del ESP32 es la incorporación de un RTC interno, que permite realizar tareas basadas en el tiempo. Sin embargo, para mantener una alta precisión temporal, es común sincronizar este reloj interno con un servidor NTP (Network Time Protocol). La sincronización con un servidor NTP garantiza que el reloj del ESP32 esté alineado con el tiempo universal coordinado (UTC), corrigiendo automáticamente cualquier desviación temporal y asegurando que las operaciones dependientes del tiempo se ejecuten con precisión. Este proceso se realiza una sola vez al encender el dispositivo mediante la conexión a internet, donde el ESP32 solicita la hora exacta al servidor y ajusta su reloj interno en consecuencia.
- **Recopilación de datos:**
 - Lectura de datos de temperatura y humedad del sensor DHT22.
 - Lectura de datos de presión y altitud del sensor BMP280.
 - Lectura de la calidad del aire (PPM de CO2) del sensor MQ135.
 - Lectura del voltaje de la batería del ESP32.

- Almacenamiento y reporte:

- Envío de los datos a Firebase cada 5 segundos.
- Almacenamiento de los datos en la tarjeta microSD cada 5 minutos.

III. RESULTADOS

A. DATOS RECOPIADOS

Los datos recopilados por el sistema se almacenan en una memoria MicroSD con el siguiente formato “AAAAMMDD.CSV” esto con el fin de ser identificados los archivos fecha a fecha y poderse analizar posteriormente ya sea en Excel o Python, para así presentar la data por medio de tablas o gráficos que ilustren las variaciones ambientales a lo largo del tiempo.

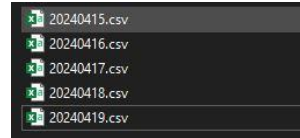


Fig. 12 Ejemplo data almacenada en memoria.

	A	B	C
1	hora,	temperatura,	humedad_relativa
2	2024-04-15 00:00:00,	29.0,	56.9
3	2024-04-15 01:00:00,	0.0,	0.0
4	2024-04-15 02:00:00,	29.0,	59.4
5	2024-04-15 03:00:00,	29.0,	60.0
6	2024-04-15 04:00:00,	28.0,	61.5
7	2024-04-15 05:00:00,	27.0,	62.6

Fig. 13 Ejemplo estructura de la data en el archivo .CSV.

Para analizar la data recopilada se han creado algunos algoritmos en Python, los cuales nos permiten crear graficas de la data histórica y así analizar tendencias y comportamientos, a continuación, se muestran las gráficas creadas en Python:

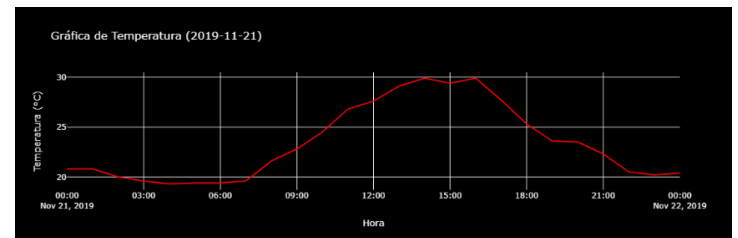


Fig. 14 Gráfica de Temperatura Vs. Tiempo generada con data histórica.

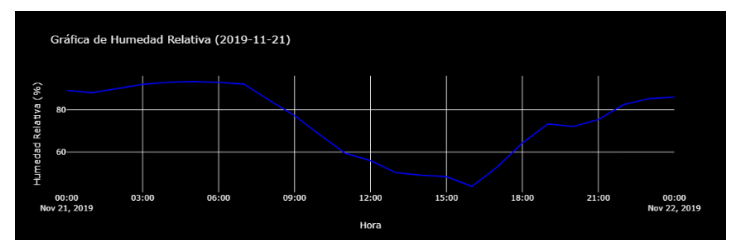


Fig. 15 Gráfica de Humedad Relativa Vs. Tiempo generada con data histórica.

Igualmente, en Python podremos extraer información valiosa de toda la data, podremos identificar las temperaturas y humedades mínimas y máximas para un rango de tiempo. Cabe resaltar que para efectos de pruebas de estos algoritmos se han analizado solamente las variables de temperatura y humedad relativa.

```

Temperatura máxima: 29.9 °C, Hora: 2019-11-21 14:00:00
Temperatura mínima: 19.3 °C, Hora: 2019-11-21 04:00:00
Humedad relativa máxima: 93.4 %, Hora: 2019-11-21 05:00:00
Humedad relativa mínima: 43.7 %, Hora: 2019-11-21 16:00:00

```

Fig. 16 Información extraída de la data graficada.

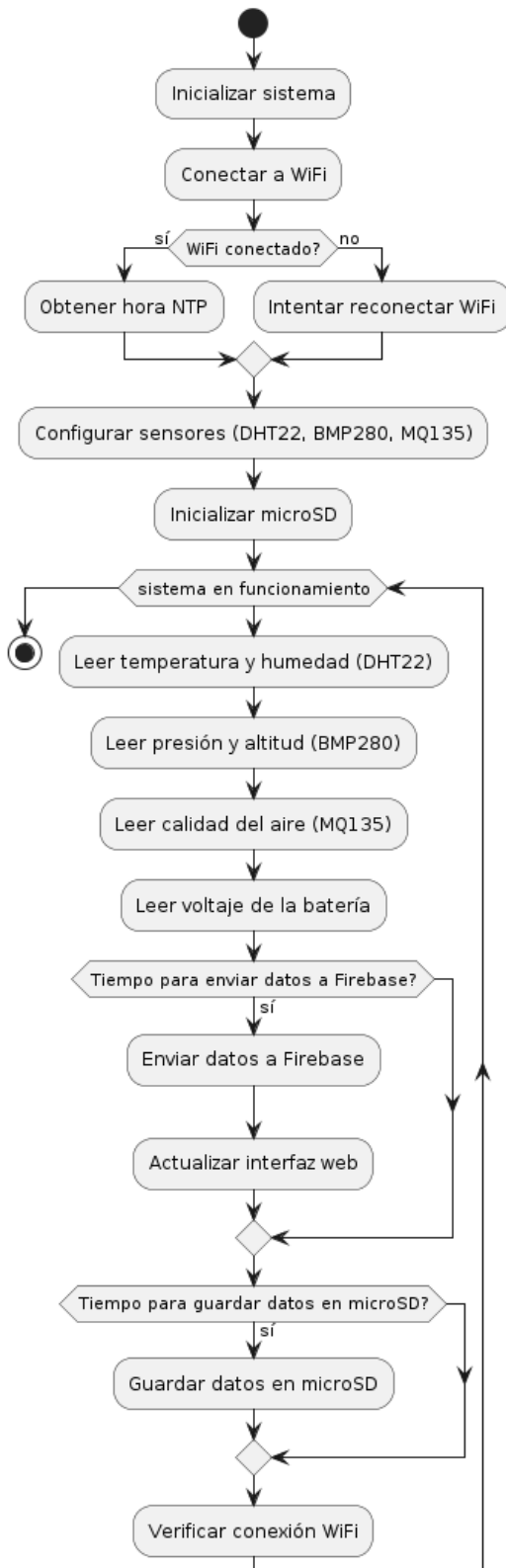


Fig. 11 Diagrama de flujo de todo el procedimiento.

IV. DISCUSIÓN

A. INTERPRETACIÓN DE LOS RESULTADOS

Los resultados indican que la temperatura y la humedad varían ligeramente a lo largo del día, alcanzando un pico de temperatura alrededor del mediodía y temperaturas mínimas en horas de la madrugada. Esta variación diaria sugiere una estacionalidad en los datos, lo cual es crucial para el análisis de esta información. Se concluye que, para proyectos futuros, es necesario analizar estas series temporales con mayor detalle utilizando algoritmos de predicción y modelos estadísticos para identificar patrones y realizar predicciones, algunos modelos comúnmente utilizados son los SARIMA (Seasonal AutoRegressive Moving Average Integrated Models Additional) “Modelos Autorregresivos Integrados de Media Móvil con Componentes Estacionales”, modelos de regresión exponencial, suavizado exponencial, técnicas de Machine Learning, entre otros.

En Python, la biblioteca *statsmodels* proporciona una implementación robusta de los modelos SARIMA, facilitando a los analistas y científicos de datos la creación, ajuste y evaluación de estos modelos de manera eficiente. Con *statsmodels*, se pueden realizar análisis complejos de series temporales y obtener pronósticos precisos.

Como ejemplo de implementación, se realizó la prueba y ajuste de un modelo SARIMA para la data real graficada anteriormente (véase Fig. 14 y Fig.15), mediante un algoritmo en Python haciendo uso de *Statsmodels* se realizó la predicción para el mismo periodo de tiempo, los resultados obtenidos fueron bastante acertados y se acercan al comportamiento de los datos reales.

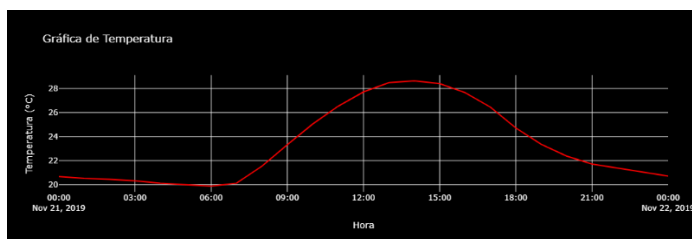


Fig. 17 Predicción de Temperatura Vs. Tiempo.

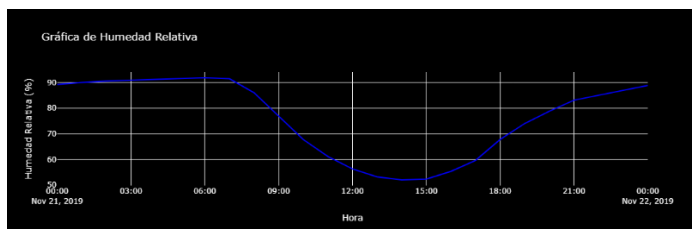


Fig. 18 Predicción de Humedad Vs. Tiempo.

```

Temperatura máxima: 28.64 °C, Hora: 2019-11-21 14:00:00
Temperatura mínima: 19.88 °C, Hora: 2019-11-21 06:00:00
Humedad relativa máxima: 91.85 %, Hora: 2019-11-21 06:00:00
Humedad relativa mínima: 51.99 %, Hora: 2019-11-21 14:00:00

```

Fig. 19 Información extraída de la data predicha.

Adicionalmente, podemos decir que los niveles de CO2 también muestran fluctuaciones que podrían estar correlacionadas con la actividad humana en el área monitoreada.

Las variables de presión atmosférica y altitud son estáticas y varían según la posición geográfica en donde se realicen las mediciones, sin embargo, al igual que la temperatura y humedad, se podrían analizar en conjunto con las demás variables y encontrar alguna correlación o patrones que determinen estados de cambio climático.

B. COMPARACIÓN CON ESTUDIOS PREVIOS

Nuestros hallazgos son consistentes con estudios previos que indican que la calidad del aire en espacios cerrados puede fluctuar significativamente en función de la ventilación y la densidad de ocupación, además de la estacionalidad generada por las variables de temperatura y humedad.

C. RELEVANCIA DE LOS RESULTADOS

Estos datos son cruciales para comprender las condiciones ambientales en tiempo real, lo que puede ayudar en la gestión eficiente de la calidad del aire y el confort ambiental en espacios interiores o incluso exteriores, para así llevar a cabo una mejor toma de decisiones basado en datos verdaderos.

D. LIMITACIONES DEL ESTUDIO

Cobertura de la Red Wi-Fi: Las desconexiones intermitentes o prolongadas pueden causar lagunas en los datos enviados a Firebase, sin embargo, la data quedará guardada de manera local en la memoria MicroSD.

Precisión de los Sensores: La precisión de los sensores puede verse afectada por factores ambientales como la temperatura y la humedad extremas.

Almacenamiento Local: La capacidad de la tarjeta microSD limita la cantidad de datos que se pueden almacenar localmente, aunque el tamaño de los archivos es bastante pequeño, se deberá revisar a lo largo del tiempo o calcular cuanta data podrá ser almacenada según el tamaño de memoria instalado.

V. CONCLUSIONES

El sistema desarrollado ha demostrado ser eficaz en la monitorización en tiempo real de las condiciones ambientales, proporcionando datos precisos sobre la temperatura, humedad, calidad del aire y presión atmosférica.

La implementación de este sistema puede ayudar a mejorar la calidad del aire en espacios interiores, proporcionando alertas tempranas sobre condiciones

adversas y permitiendo una gestión proactiva de las mismas.

Los datos recopilados por el sensor DHT22, que mide temperatura y humedad, han sido cruciales para entender las condiciones ambientales del entorno monitoreado.

La monitorización continua de temperatura y humedad contribuye al bienestar humano. Niveles adecuados de humedad, por ejemplo, pueden reducir la incidencia de problemas respiratorios y alergias.

A lo largo del desarrollo del proyecto se identificó que sería ideal la incorporación de sensores adicionales, ya que añadir sensores de otros contaminantes del aire nos ayudaría a obtener un panorama más completo de la calidad ambiental.

Una de las mayores ventajas de recopilar datos de temperatura y humedad en tiempo real es la capacidad de analizar estos datos como series temporales. El análisis de series temporales permite no solo entender las tendencias y patrones históricos, sino también realizar predicciones futuras, lo cual es fundamental para la planificación y la toma de decisiones informadas.

La creación de una interfaz web propia para la comunicación con el ESP32 ofrece numerosas ventajas en términos de personalización, control, seguridad, independencia y eficiencia. Mientras que plataformas como ThingSpeak, Blynk, Ubidots STEM, pueden proporcionar soluciones rápidas y fáciles de usar para proyectos simples o temporales, una interfaz personalizada se adapta mejor a proyectos complejos o a largo plazo que requieren una alta flexibilidad, seguridad y control sobre los datos y el funcionamiento del sistema. Esta capacidad de personalización y control puede ser crucial para el éxito y la sostenibilidad del proyecto a largo plazo.

VI. REFERENCIAS

- [1] J. Smith, *The Internet of Things: Connecting Objects*, New York: McGraw-Hill, 2015.
- [2] M. K. G. a. P. G. Bajpai, *Design and Implementation of IoT-based Environmental Monitoring System*, *Journal of Environmental Science and Engineering*, 2021.
- [3] E. Systems, «ESP32 Series Datasheet,» [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [4] Firebase, «Firebase Real Time Database,» [En línea]. Available: <https://firebase.google.com/docs/database..>
- [5] Adafruit, «DHT22 Temperature-Humidity Sensor,» [En línea]. Available: <https://learn.adafruit.com/dht>.
- [6] Arduino, «Getting Started with the Arduino IDE,»

[En línea]. Available: <https://www.arduino.cc/en/Guide>.

- [7] P. S. Foundation, «Python Language Reference, version 3.8,» [En línea]. Available: <https://www.python.org>.
- [8] S. Developers, «Statsmodels: Statistical models, hypothesis tests, and data exploration,» [En línea]. Available: <https://www.statsmodels.org>.
- [9] J. R. y. L. S. F. López, «Desarrollo de un sistema de monitoreo ambiental con ESP32 y Firebase,» de *Actas del Congreso Latinoamericano de Tecnología y Sociedad (LATSOC)*, Buenos Aires, 2021.
- [10] C. G. y. M. Torres, «Sistemas de monitoreo ambiental basados en IoT: Un enfoque práctico,» *Revista Iberoamericana de Automática e Informática Industrial*, 2020.

Enlace GitHub del proyecto:

<https://github.com/catachunga9564/iot-proycomunicaciones>

