**MONITOREO EN TIEMPO REAL DE TEMPERATURA Y HUMEDAD CON ARDUINO MEGA, SENSOR DHT11, MONGODB Y FLASK**

**MARIO ALEJANDRO PEÑA ARENAS**

**OMAR FERNANDO PEÑA GARCÍA**

**YEISSON CAMILO VILLAMIL BLANDON**

**IVAN CAMILO GARCÍA URREGO**

**UNIVERSITARIA DE COLOMBIA**

[**APLICACIONES OPEN SOURCE**](https://ww1.aulavirtualuniversitariadecolombia.co/course/view.php?id=2575)

**LAURA ARTEAGA**

**26/06/2025**

**Título del Proyecto**

Monitoreo en Tiempo Real de Temperatura y Humedad con Arduino Mega, Sensor DHT11, MongoDB y Flask

**Objetivo General**

Diseñar e implementar un sistema de monitoreo en tiempo real de temperatura y humedad usando Arduino Mega y el sensor DHT11, con visualización de datos mediante una aplicación web desarrollada con Flask y almacenamiento en MongoDB Atlas.

**Objetivos Específicos**

1. Integrar el sensor DHT11 con Arduino Mega para la captura periódica de datos de temperatura y humedad.
2. Almacenar los datos capturados en una base de datos MongoDB en la nube utilizando Python.
3. Visualizar los datos en tiempo real mediante gráficos interactivos en una aplicación web con Flask y Chart.js.

**Planteamiento del Problema**

En entornos como talleres, laboratorios o invernaderos, el control de condiciones ambientales es esencial. Sin embargo, muchos espacios pequeños carecen de sistemas automatizados que monitoreen estos valores en tiempo real. Este proyecto aborda la necesidad de una solución accesible y de código abierto que permita recolectar, almacenar y visualizar información ambiental, facilitando la toma de decisiones basada en datos.

**Pregunta Problema**

¿Cómo desarrollar un sistema de monitoreo en tiempo real de temperatura y humedad utilizando herramientas de bajo costo y tecnologías open-source como Arduino, MongoDB y Flask?

**Estado del Arte**

1. “Sistema de Monitoreo IoT con NodeMCU y DHT11” (ResearchGate): Usa WiFi para subir datos a ThingSpeak, sin almacenamiento permanente. Este proyecto agrega almacenamiento en MongoDB.
2. “Aplicación Web para Visualización de Sensores con Flask” (Google Académico): Se enfoca en mostrar datos de sensores industriales, pero no incluye la recolección física. Nuestro sistema combina captura física y visualización.
3. “Monitoreo Ambiental con Raspberry Pi y MongoDB” (Revistas Académicas): Similar en backend, pero más costoso. Este proyecto opta por Arduino Mega para mayor accesibilidad.

**Fases Metodológicas**

1. Planeación: Selección de componentes y herramientas (Arduino, DHT11, MongoDB, Flask).
2. Implementación del Sensor: Lectura de datos desde el DHT11 conectado a Arduino Mega.
3. Conexión Serial y Lectura en Python: Captura de datos desde el puerto COM8 con pyserial.
4. Almacenamiento en MongoDB Atlas: Inserción de datos con marca de tiempo en la colección arduino\_uno.
5. Diseño del Dashboard en Flask: Uso de Chart.js para graficar temperatura y humedad.
6. Pruebas y Validación: Verificación en tiempo real del correcto envío, almacenamiento y visualización de datos.

**Captura de Datos**

El sensor DHT11, conectado a Arduino Mega, envía cada segundo lecturas de temperatura y humedad al puerto serial. Un script Python en Flask lee estas líneas, las transforma en JSON, y las guarda en MongoDB con una marca de tiempo ajustada a la zona horaria de Bogotá.

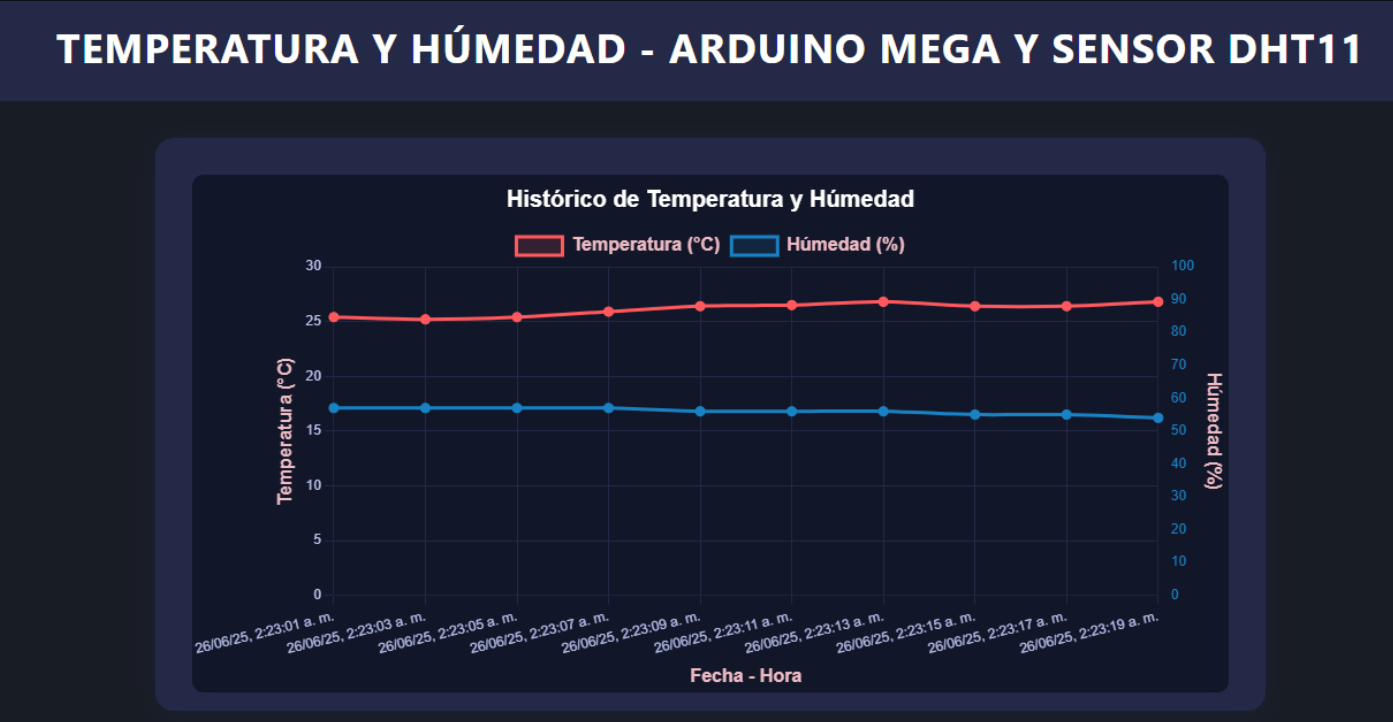
**Procesamiento de Datos y Búsqueda con MongoDB**

Los datos se almacenan en la base de datos MongoDB Atlas, colección arduino\_uno. Se realiza una consulta con .find().sort("fecha\_hora", -1).limit(10) para obtener los 10 registros más recientes. Estos datos se formatean para ser visualizados por el cliente web.

**Visualización de Datos con Flask**

Se usa Flask para levantar un servidor web, donde se genera un gráfico dinámico en JavaScript usando Chart.js. Cada 3 segundos, el cliente realiza una solicitud a /datos, y actualiza la gráfica con los datos más recientes de temperatura (°C) y humedad (%).

**Captura del Dashboard:**



**Resultados**

* Se logró una visualización clara de los datos en tiempo real.
* El gráfico muestra dos líneas: temperatura y humedad, con escalas independientes.
* Se almacena la información en MongoDB, garantizando su persistencia.

**Evidencia de inserciones en consola:**

📤 Subido -> Temp: 26.1 °C | Humedad: 55.0 %  
📤 Subido -> Temp: 26.2 °C | Humedad: 54.8 %

**Enlace de GitHub del repositorio:**

https://github.com/Marioalejop/Proyecto\_Final

**Conclusiones**

* Se integraron con éxito tecnologías open-source para crear un sistema de monitoreo ambiental.
* El uso de MongoDB Atlas permitió escalabilidad y acceso remoto a los datos.
* Flask y Chart.js facilitaron una visualización moderna y responsiva.
* Es un sistema ideal para aplicaciones pequeñas y medianas que requieran control de ambiente.

**Bibliografía**

* Arduino.cc. (n.d.). \*DHT11 Sensor\*. Recuperado de <https://www.arduino.cc>
* MongoDB Inc. (2024). \*MongoDB Atlas Documentation\*. <https://www.mongodb.com/cloud/atlas>
* Flask Documentation. (n.d.). \*Flask Web Development\*. <https://flask.palletsprojects.com/>
* Chart.js (n.d.). \*JavaScript Charting Library\*. <https://www.chartjs.org>
* ResearchGate, Google Scholar (2024). Revisión de proyectos relacionados con IoT y monitoreo ambiental.