

UNIVERSIDAD DE COSTA RICA
ESCUELA DE INGENIERÍA ELÉCTRICA
IE0624 - LABORATORIO DE
MICROCONTROLADORES

INFORME DEL PROYECTO FINAL
ESTACIÓN METEOROLÓGICA CON
PREDICCIÓN DE CLIMA

PROFESOR
MARCO VILLALTA FALLAS

JOSE DANIEL BLANCO SOLIS, B71137

III 2024

03 DE MARZO DE 2025

Índice

1. Introducción	3
2. Objetivo General	4
3. Objetivo Específicos	4
4. Nota teórica	5
4.1. Internet of things	5
4.2. Machine Learning	5
4.3. Arduino Nano BLE 33	5
4.4. Sensor DHT22	6
5. Desarrollo/Análisis	7
5.1. Configuración del hardware y conexión de componentes	7
5.1.1. Conexión del Circuito	7
5.1.2. Pruebas iniciales del hardware	7
5.2. Desarrollo del firmware y envío de datos a la nube	8
5.2.1. Configuración	8
5.2.2. Envío datos	8
5.3. Entrenamiento del modelo	10
5.4. Visualización en dashboard	10
6. Conclusiones y Recomendaciones	11

Resumen

Este informe muestra el desarrollo de una estación meteorológica basada en internet de las cosas con capacidad de predicción climática utilizando machine learning, se compone de un microcontrolador Arduino Nano BLE 33, un sensor DHT22 para la medición de temperatura y humedad y un módulo de conectividad ESP8266 que permite el envío de datos a la plataforma en la nube ThingsBoard, en este caso la integración de un modelo de Machine Learning desarrollado con Edge Impulse permite analizar datos históricos y predecir tendencias de temperatura y humedad a futuro.

1. Introducción

El presente proyecto tiene como objetivo diseñar y desarrollar una estación meteorológica basada en el microcontrolador Arduino Nano BLE 33, equipada para la medición de temperatura y humedad y conectada a internet para enviar los datos capturados por el sensor a la plataforma en la nube donde se almacenan y visualizan mediante un dashboard interactivo, se implementa un modelo de machine learning que analiza los datos históricos y permite realizar predicciones sobre futuras condiciones ambientales.

A lo largo de este informe se describen detalladamente cada una de las fases del desarrollo del proyecto, comenzando por la configuración del hardware donde se explican los componentes utilizados y su interconexión, se aborda la implementación del firmware en el microcontrolador lo que implica la escritura de código en el entorno Arduino IDE para la adquisición de datos del sensor DHT22 y la posterior transmisión de estos datos a la plataforma ThingsBoard mediante el módulo ESP8266, se describe la configuración de la infraestructura en la nube explicando el proceso de integración con ThingsBoard, la creación de dashboards y la gestión de dispositivos, finalmente se desarrolla el modelo de machine learning utilizando Edge Impulse, permitiendo la predicción de tendencias climáticas basadas en los datos recolectados.

2. Objetivo General

Desarrollar una estación meteorológica que permita la medición y análisis de datos ambientales, así como la predicción de tendencias climáticas mediante técnicas de machine learning.

3. Objetivo Específicos

- Diseñar y ensamblar un sistema basado en el microcontrolador Arduino Nano BLE 33 y el sensor DHT22, garantizando una adquisición precisa de datos de temperatura y humedad.
- Implementar un mecanismo de comunicación eficiente mediante el módulo WiFi ESP8266 para la transmisión de datos a la plataforma ThingsBoard.
- Configurar la infraestructura en la nube para la recepción, almacenamiento y visualización de los datos recopilados en un dashboard interactivo.
- Desarrollar y entrenar un modelo de machine learning en Edge Impulse que a partir de los datos recolectados, sea capaz de predecir tendencias de temperatura y humedad.

4. Nota teórica

4.1. Internet of things

Es un paradigma tecnológico que permite la interconexión de dispositivos físicos a través de internet, permitiendo la recolección, transmisión y análisis de datos en tiempo real, se define como una red global de objetos interconectados que pueden comunicarse entre sí y con los usuarios mediante el uso de sensores, redes de comunicación y plataformas en la nube [1].

4.2. Machine Learning

Es una rama de la inteligencia artificial que permite a las computadoras aprender patrones a partir de datos y hacer predicciones basadas en la experiencia previa, el aprendizaje automático se basa en algoritmos que analizan datos históricos para identificar tendencias y hacer inferencias sobre eventos futuros [6].

4.3. Arduino Nano BLE 33

Es una de las placas más avanzadas de la familia Arduino, diseñada específicamente para aplicaciones IoT y machine Learning en el borde, está basado en el microcontrolador ARM Cortex-M4 de 32 bits y cuenta con conectividad Bluetooth Low Energy (BLE) permitiendo la comunicación con otros dispositivos de manera inalámbrica y con bajo consumo energético, una de las principales ventajas es su capacidad para ejecutar modelos de machine Learning directamente en el hardware, lo que reduce la necesidad de depender de servidores externos para el procesamiento de datos, esta capacidad es crucial para aplicaciones IoT que requieren procesamiento en tiempo real y baja latencia, como el monitoreo de variables ambientales y la automatización de sistemas [2].

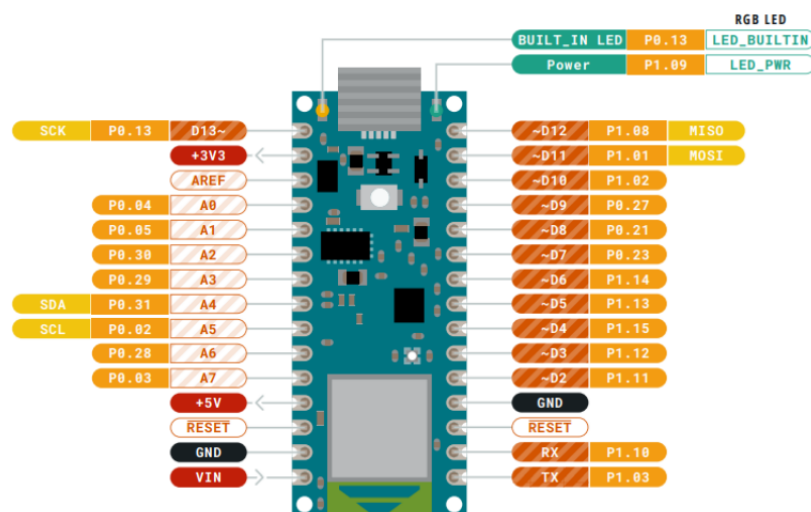


Figura 1: Diagrama del arduino Nano 33 BLE.

4.4. Sensor DHT22

Es un sensor digital que permite la medición de temperatura y humedad relativa con alta precisión, utiliza un termistor para la medición de temperatura y un sensor capacitivo para la detección de la humedad, proporcionando valores en formato digital a través de un solo pin de datos, tiene un rango de medición de temperatura de -40°C a 80°C con una precisión de $\pm 0,5^{\circ}\text{C}$ y un rango de medición de humedad de 0 % a 100 % con una precisión de $\pm 2 - 5\%$ [3].

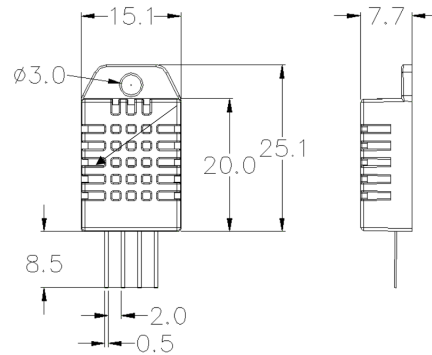


Figura 2: Diagrama del DHT22.

5. Desarrollo/Análisis

5.1. Configuración del hardware y conexión de componentes

5.1.1. Conexión del Circuito

Fue diseñado para garantizar una comunicación estable entre los componentes, asegurando que los niveles de voltaje fueran compatibles y que la transmisión de datos se realizara sin interferencias, a continuación, se detallan las conexiones realizadas:

- Sensor DHT22:
 - VCC \rightarrow 5 V del Arduino Nano BLE 33
 - GND \rightarrow GND del Arduino
 - Data \rightarrow Pin digital D2 del Arduino (con una resistencia pull-up de 10 $k\Omega$ para garantizar estabilidad en la lectura)

5.1.2. Pruebas iniciales del hardware

Antes de proceder con el desarrollo se realizaron pruebas individuales en cada componente para garantizar su correcto funcionamiento, se escribió un código simple en Arduino IDE para leer los valores del DHT22 y mostrarlos en el monitor serie.

```
1 #include <Adafruit_Sensor.h>
2 #include <DHT.h>
3 #include <DHT_U.h>
4
5 #define DHTPIN 2          // Pin donde est  conectado el DHT22
6 #define DHTTYPE DHT22    // Tipo de sensor
7
8 DHT dht(DHTPIN, DHTTYPE);
9
10 void setup() {
11     Serial.begin(115200); // Configurar comunicaci n serie
12     dht.begin();          // Iniciar el sensor DHT22
13 }
14
15 void loop() {
16     float temp = dht.readTemperature(); // Leer temperatura
17     float hum = dht.readHumidity();     // Leer humedad
18
19     if (!isnan(temp) && !isnan(hum)) { // Verificar que los valores
        sean v lidos
20         Serial.print("TEMP:");
21         Serial.print(temp);
22         Serial.print(", HUM:");
23         Serial.println(hum);
```

```

24 } else {
25     Serial.println("Error al leer el sensor");
26 }
27
28 delay(2000); // Esperar 2 segundos antes de la pr xima lectura
29 }

```

Los resultados de estas pruebas confirmaron que el sensor estaba operativo y proporcionaba mediciones precisas de temperatura y humedad.

5.2. Desarrollo del firmware y envío de datos a la nube

5.2.1. Configuración

Para almacenar y visualizar los datos se generó una clave de autenticación Access Token que se utilizó para autenticar el envío de datos desde el Arduino.

5.2.2. Envío datos

Se implementó el siguiente código para obtener datos del DHT22 y enviarlos periódicamente a la plataforma ThingsBoard mediante MQTT:

```

1 import paho.mqtt.client as mqtt
2 import json
3 import time
4 import serial
5 import sys
6
7 # Puerto Serial (ajustar seg n el sistema operativo)
8 PORT = "/dev/ttyACM0" # Linux/Mac (en Windows usa "COMX")
9
10 # Configuraci n del broker MQTT (ThingsBoard)
11 BROKER = "iot.eie.ucr.ac.cr" # Servidor MQTT de ThingsBoard
12 PORT_MQTT = 1883
13 TOPIC_TELEMETRY = "v1/devices/me/telemetry"
14 USERNAME = "mJfQiM1Tqf7fAXMfULNE" # Access Token de ThingsBoard
15
16 # Funci n al conectarse al broker
17 def on_connect(client, userdata, flags, rc):
18     print(f"Conectado con c digo {rc}")
19     client.subscribe(TOPIC_TELEMETRY)
20
21 # Configurar MQTT
22 def setup_mqtt_client():
23     client = mqtt.Client()
24     client.username_pw_set(USERNAME)
25     client.on_connect = on_connect
26     client.connect(BROKER, PORT_MQTT, 60)

```



```
27     client.loop_start()
28     return client
29
30 # Configurar conexión Serial con Arduino
31 def setup_serial_connection(port):
32     try:
33         return serial.Serial(port, 115200, timeout=1)
34     except serial.SerialException as e:
35         print(f"Error al abrir el puerto serial: {e}")
36         sys.exit(1)
37
38 # Leer y procesar datos del Arduino
39 def main():
40     client = setup_mqtt_client()
41     ser = setup_serial_connection(PORT)
42
43     try:
44         while True:
45             if ser.in_waiting > 0:
46                 raw_data = ser.readline().decode('utf-8', errors='
ignore').strip()
47                 print(f"Datos recibidos: {raw_data}")
48
49                 # Separar los datos de temperatura y humedad
50                 try:
51                     temp, hum = map(float, raw_data.split())
52                     payload = {'temperature': temp, 'humidity': hum}
53
54                     # Enviar datos a ThingsBoard
55                     client.publish(TOPIC_TELEMETRY, json.dumps(
payload))
56                     print(f"Datos enviados: {payload}")
57
58                 except ValueError:
59                     print("Error: Datos inválidos recibidos")
60
61                 time.sleep(2) # Esperar 2 segundos entre lecturas
62
63     except KeyboardInterrupt:
64         print("\nSaliendo del programa.")
65
66     finally:
67         ser.close()
68         client.loop_stop()
69         client.disconnect()
```

```
70         print("Conexi n serial cerrada y desconectado del broker  
          MQTT.")  
71  
72 if __name__ == "__main__":  
73     main()
```

5.3. Entrenamiento del modelo

El proceso de integración se llevó a cabo utilizando Edge Impulse, el primer paso fue la recolección de datos, que se realizó directamente desde la plataforma ThingsBoard donde los valores de temperatura y humedad enviados por el ESP8266 eran almacenados en tiempo real, se descargaron los registros en formato CSV asegurando que el conjunto de datos fuera lo suficientemente representativo, con los datos se procedió al entrenamiento del modelo de predicción utilizando un enfoque basado en regresión lineal y redes neuronales ligeras, permitiendo modelar la relación entre el tiempo transcurrido y la temperatura con el objetivo de prever tendencias futuras en las condiciones ambientales, una vez validado el modelo fue exportado en formato TensorFlow Lite para Microcontroladores y cargado directamente en el Arduino. Se integró el modelo en el código principal y se desarrolló un procedimiento que tomaba los valores más recientes de temperatura y humedad los introducía en el modelo y generaba una predicción de las condiciones futuras.

5.4. Visualización en dashboard

La integración de la visualización en tiempo real representa un ecosistema completo de IoT con capacidades avanzadas de análisis y predicción climática lo que convierte a esta estación meteorológica en una solución robusta para la toma de decisiones basadas en datos, en general, la estación meteorológica IoT proporcionó datos confiables y su integración con Machine Learning permitió obtener predicciones climáticas útiles demostrando la viabilidad del sistema en aplicaciones reales.

6. Conclusiones y Recomendaciones

El desarrollo de esta estación meteorológica IoT con predicción climática permitió comprobar la integración de sensores de medición ambiental, comunicación inalámbrica y procesamiento en la nube para la obtención y análisis de datos en tiempo real, la combinación del Arduino Nano BLE 33, el sensor DHT22 y la plataforma ThingsBoard permitió establecer un sistema confiable de monitoreo, mientras que el uso de Edge Impulse para el entrenamiento del modelo de Machine Learning proporcionó predicciones meteorológicas con un nivel aceptable de precisión. No obstante, se identificaron oportunidades de mejora, como la optimización del consumo energético, ampliar la base de datos y mejorar los métodos de comunicación.

Referencias

- [1] Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- [2] Banzi, M., & Shiloh, M. (2020). *Getting Started with Arduino: The Open Source Electronics Prototyping Platform* (3rd ed.). Maker Media, Inc.
- [3] Aosong Electronics. (2021). DHT22 (AM2302) Digital Temperature and Humidity Sensor Datasheet. Disponible en: <https://www.aosong.com>
- [4] Espressif Systems. (2020). ESP8266EX Datasheet. Disponible en: <https://www.espressif.com>
- [5] ThingsBoard. (2023). ThingsBoard Documentation. Disponible en: <https://thingsboard.io/docs/>
- [6] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.