Unidad Académica Multidisciplinaria Mante

Universidad Autónoma de TAMAULIPAS

Universidad Autónoma De Tamaulipas

Materia: Diseño Electrónico Basado En Sistemas Embebidos

Trabajo: Proyecto Integrador (Documentación)

Integrantes:

Daniel Turrubiates Cervantes

Leonardo Ramos Espinoza

Sebastián Rodela Castillo

Jorge Alejandro Quiroga Hoy

Axel Aram Verlage Aceves

Enríquez Hernández Galdino

Marco Antonio Rojas Olvera

Docente: López Piña Daniel

Grado Y Grupo: 8 E, J, F

Fecha: 11/03/2025

Unidad Académica Multidisciplinaria Mante

Universidad Autónoma de TAMAULIPAS

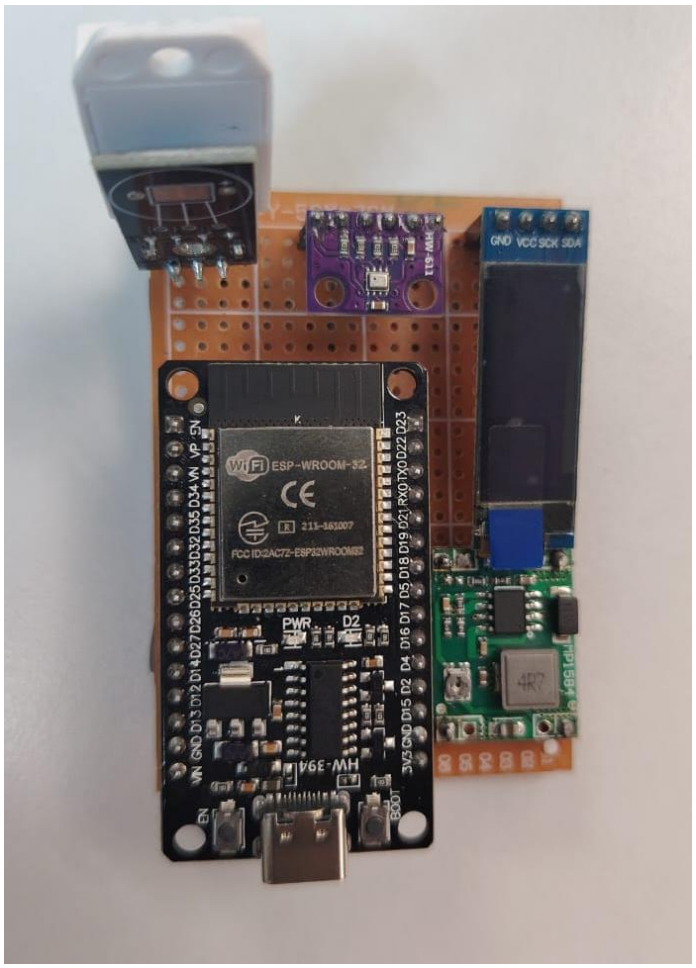
1. Introducción

El monitoreo ambiental es fundamental en diversas áreas como la agricultura, la climatología y el control de calidad del aire. Este proyecto tiene como objetivo desarrollar un sistema meteorológico basado en ESP32 para la medición de temperatura, humedad y presión atmosférica, utilizando los sensores DHT22 y BMP280. Los datos obtenidos se visualizarán en una pantalla OLED y serán enviados en tiempo real a la plataforma ThingSpeak para su análisis y almacenamiento. Este sistema proporciona una solución eficiente y económica para el monitoreo de condiciones ambientales.

Proceso

2.1 Componentes utilizados

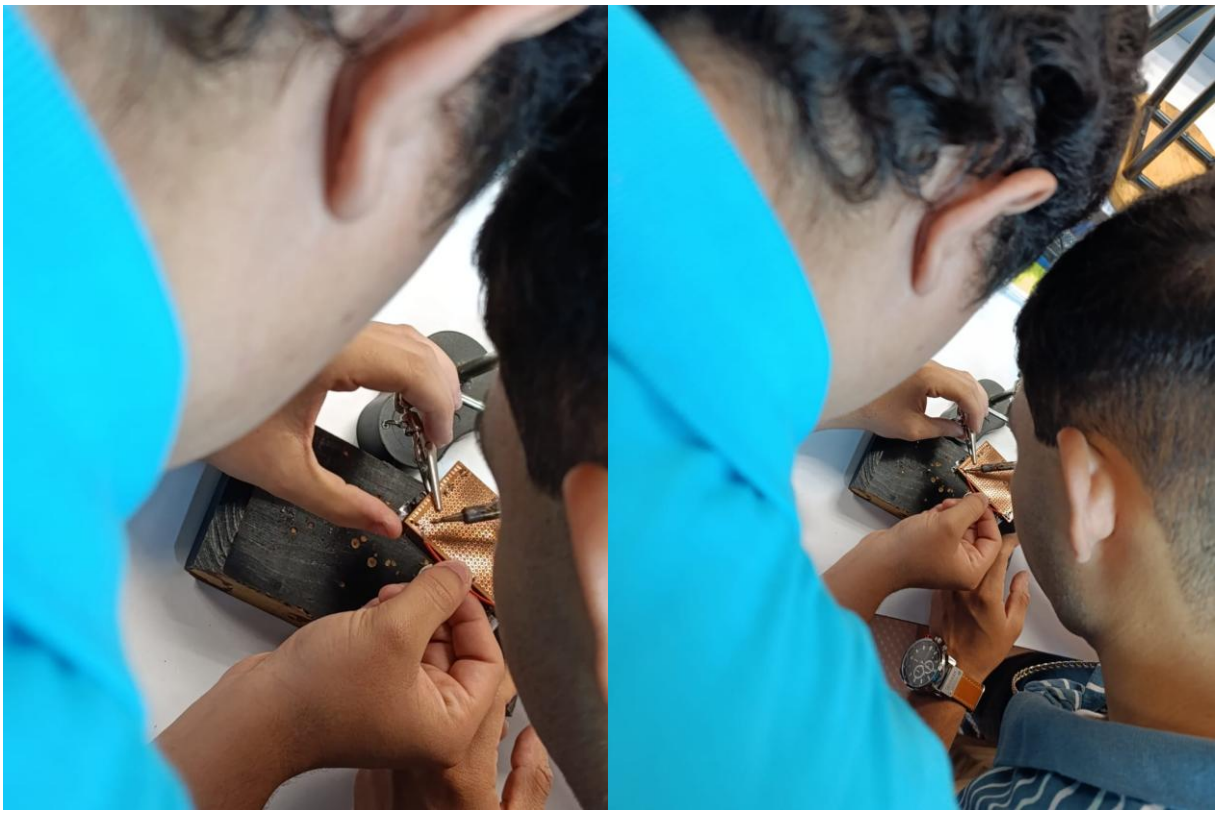
- ESP32
- Sensor de temperatura y humedad DHT22
- Sensor de presión BMP280
- Pantalla OLED
- Plataforma ThingSpeak
- Placa de circuito impreso
- Fuente de alimentación
- Regulador MP1584

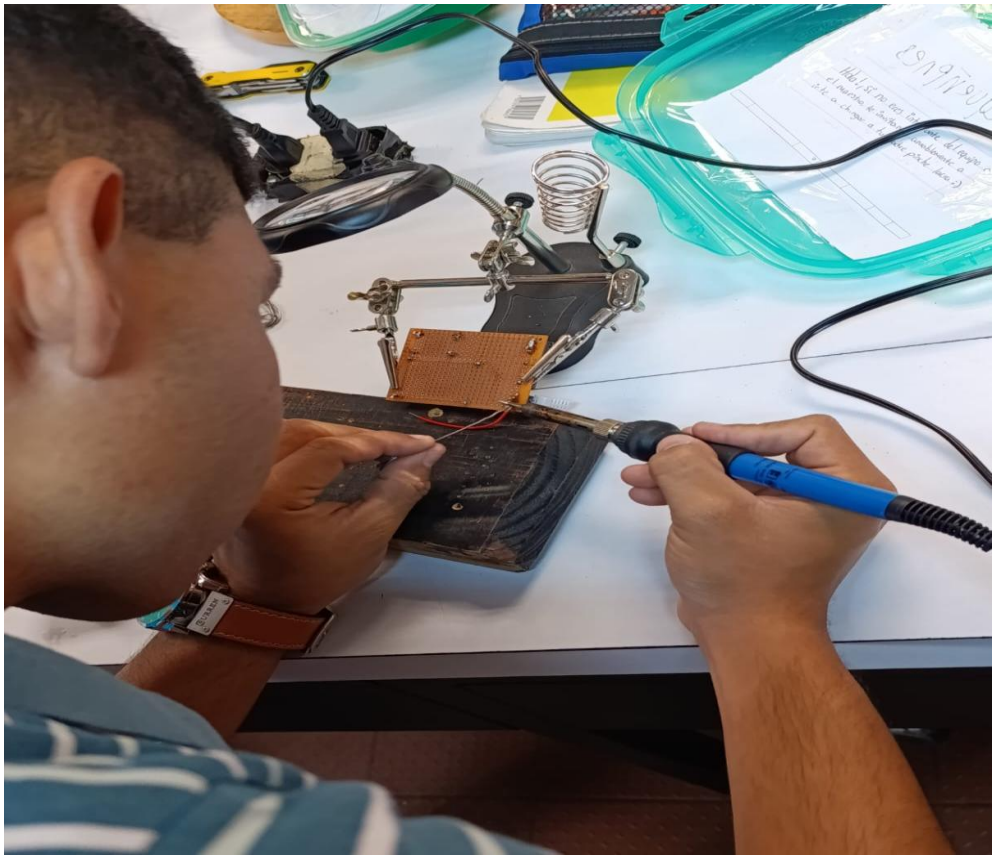


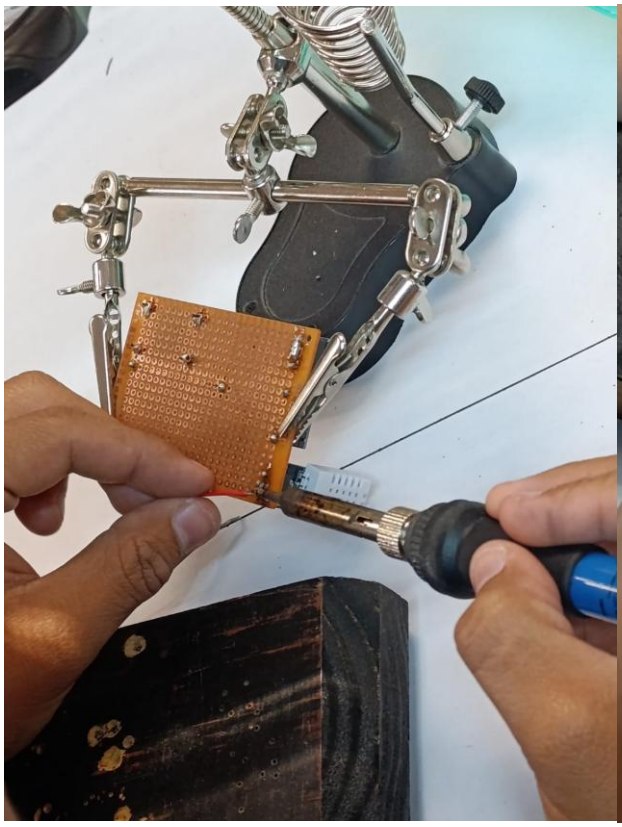
2.2 Diagrama de conexión

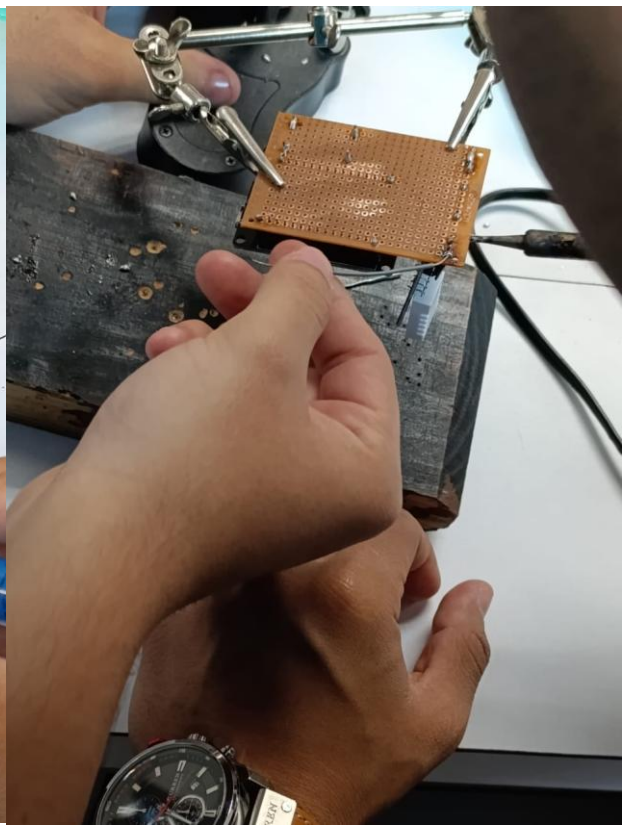
Para el ensamblaje del sistema, se realizan las siguientes conexiones:

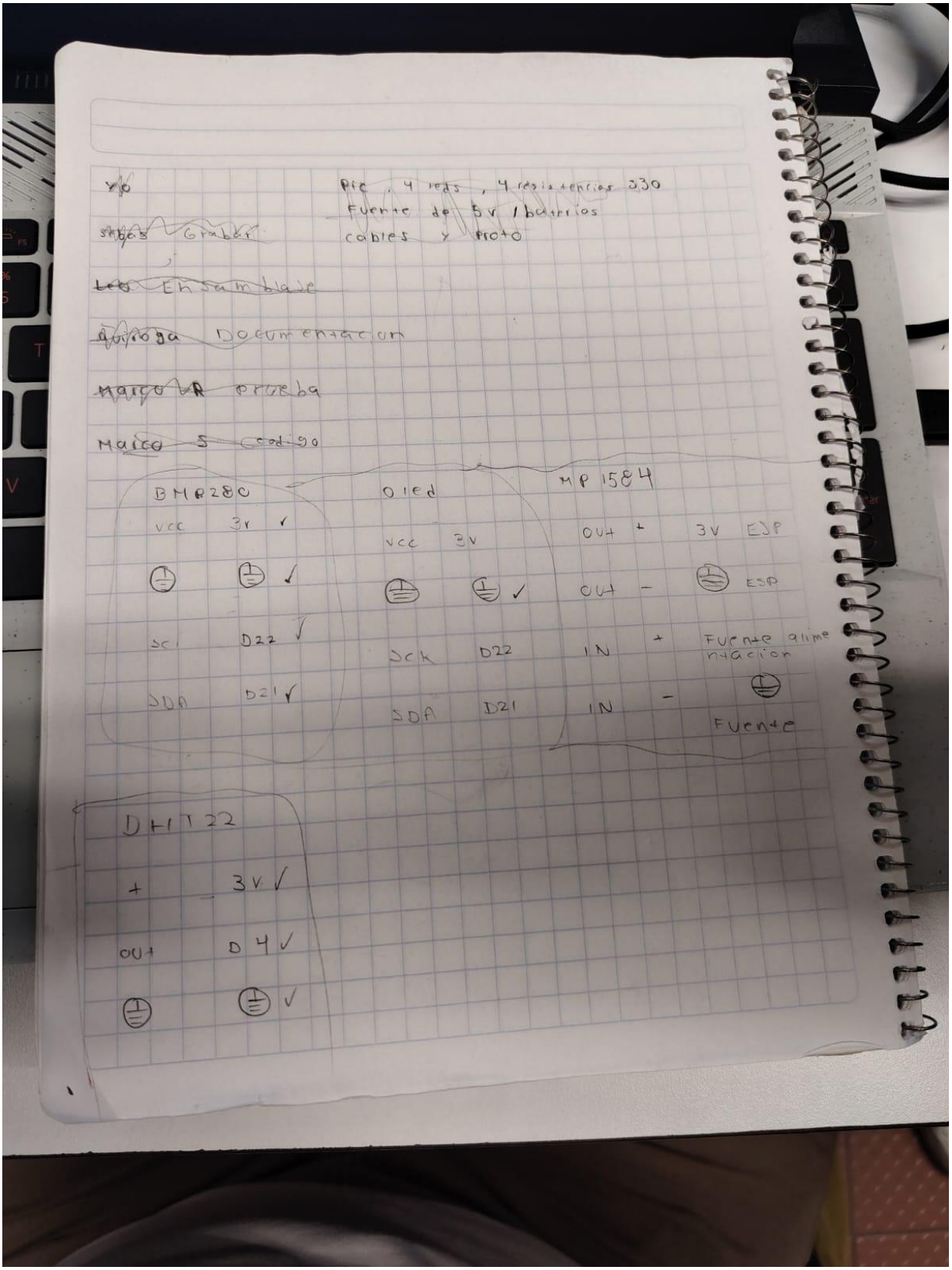
- El ESP32 se conecta a la pantalla OLED mediante el protocolo I2C.
- El sensor DHT22 se conecta al ESP32 mediante una de sus entradas digitales.
- El sensor BMP280 se conecta también mediante I2C.
- La fuente de alimentación provee energía estable al sistema.

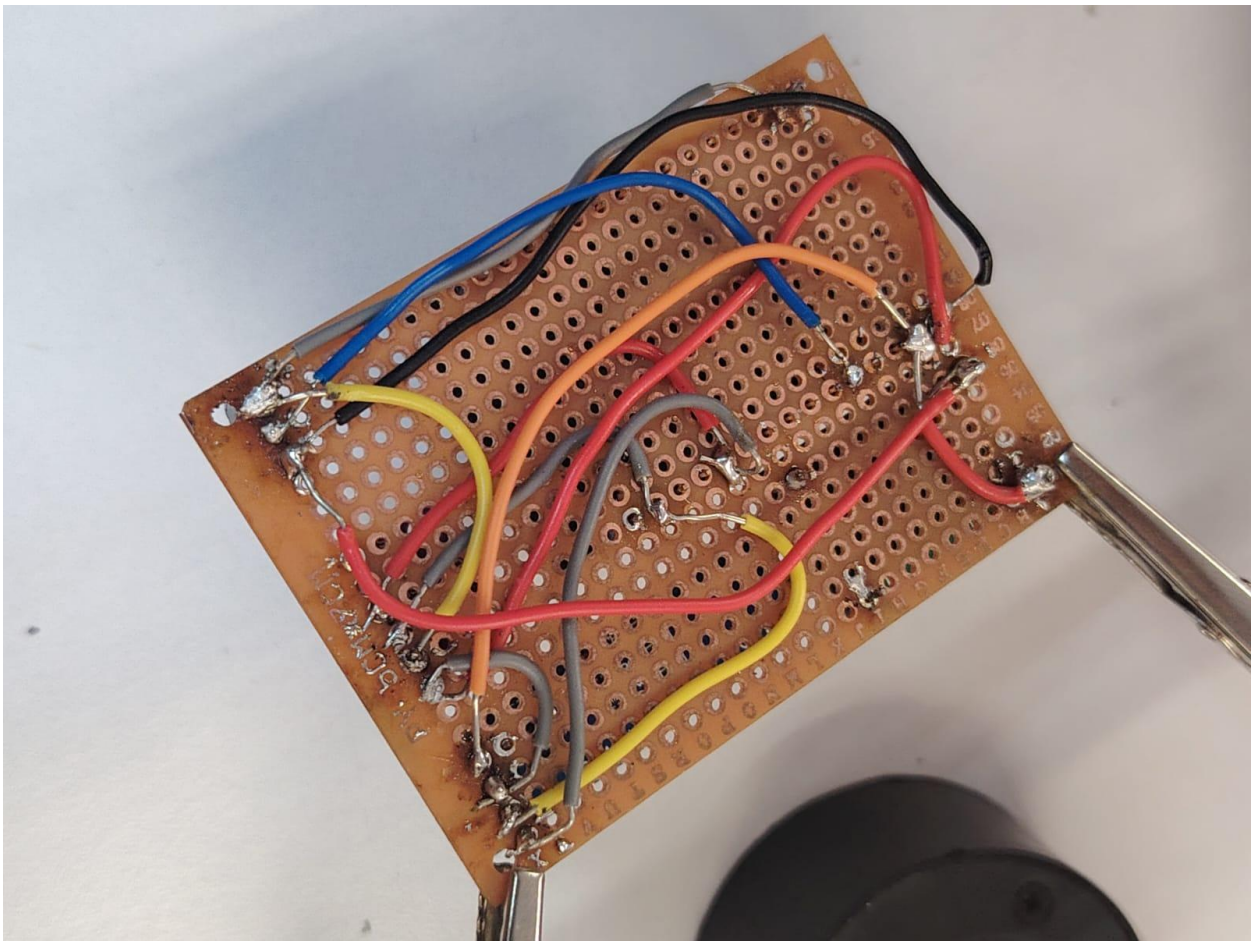






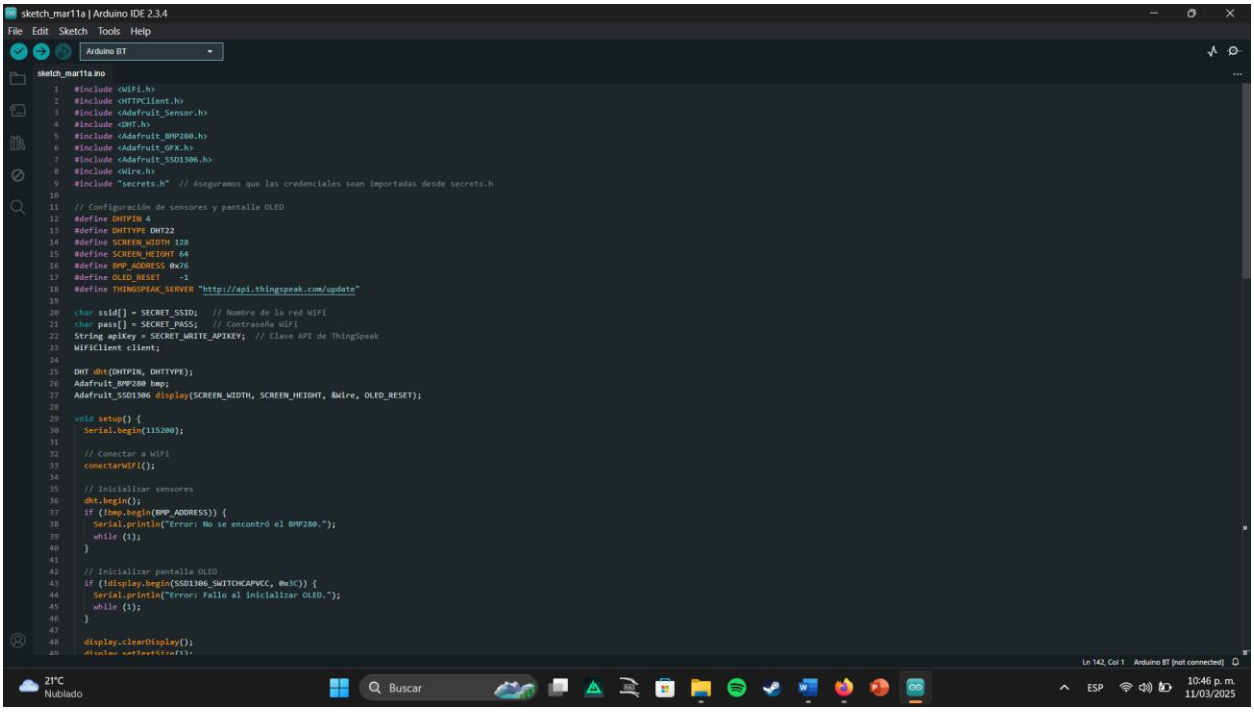


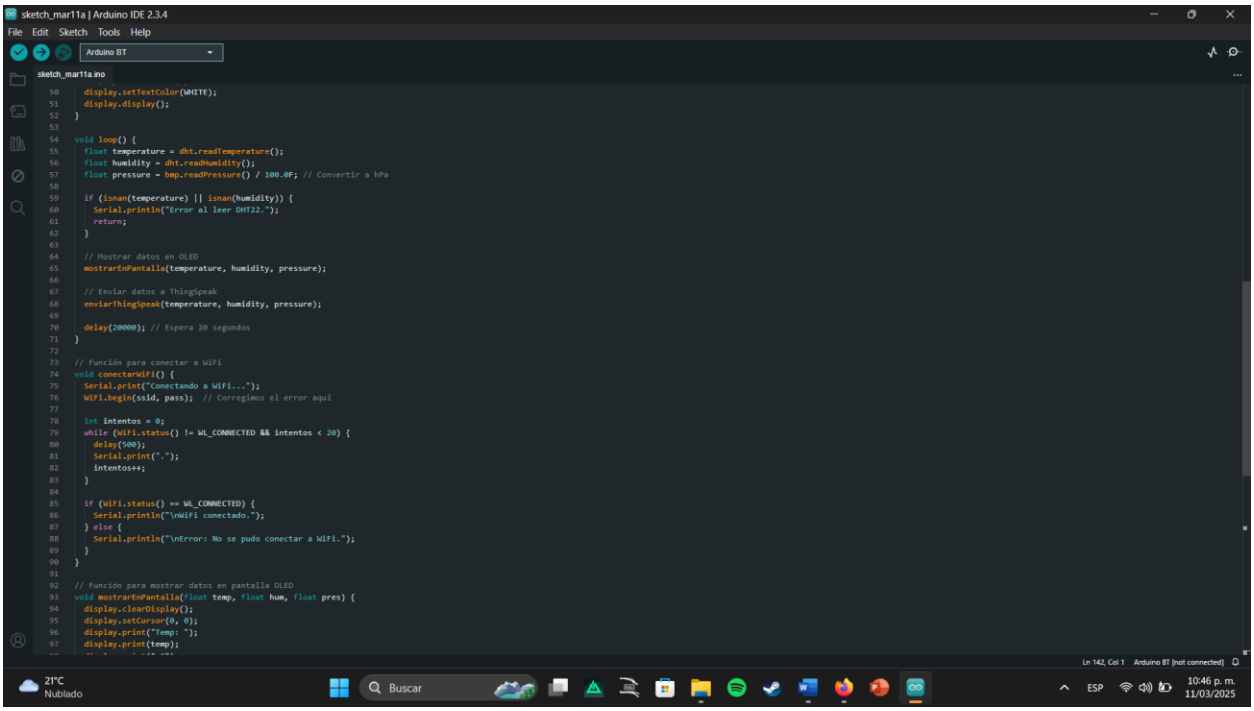

Pic , 4 leds , 4 resistencias 330
Fuente de 5v / baterias
cables y proto
Grabar
Ensamblaje
Documentacion
Prueba
Codigo
BMP280
vcc 3v ✓
SCL D22 ✓
SDA D21 ✓
Oled
vcc 3V
SCK D22
SDA D21
MP 1584
OUT + 3V ESP
OUT - ESP
IN + Fuente alimentacion
IN - Fuente
DHT22
+ 3V ✓
OUT D4 ✓

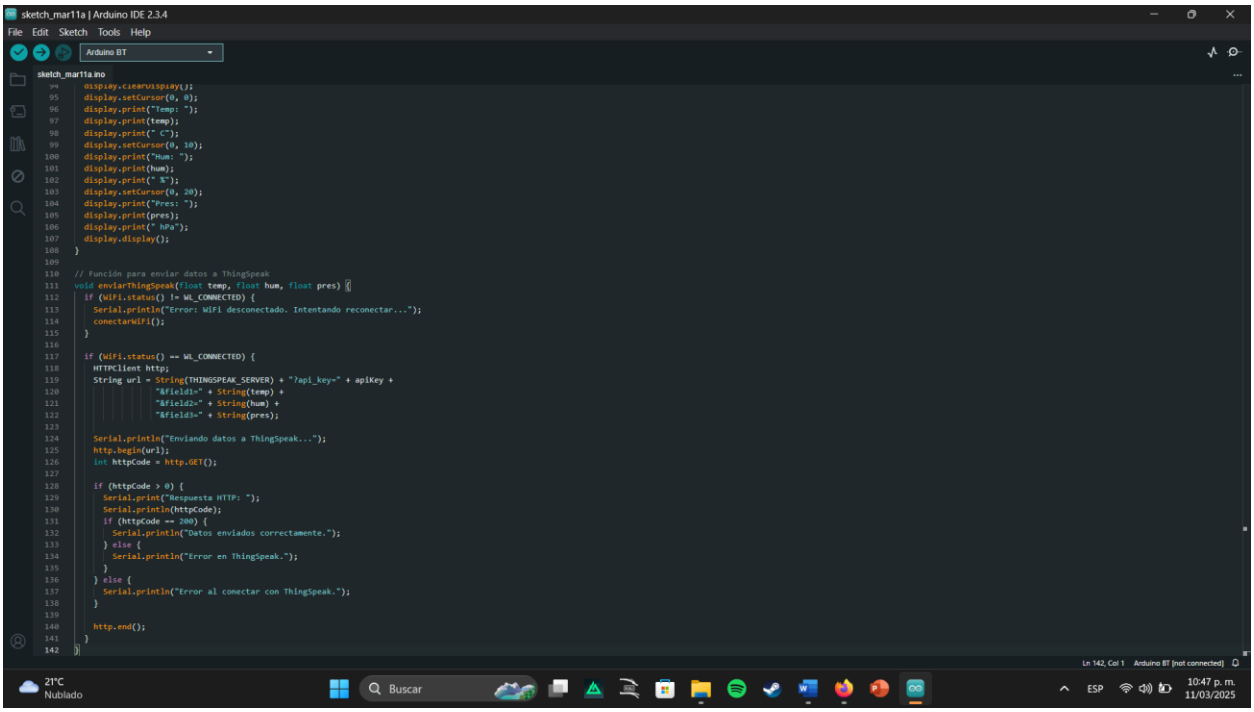


2.3 Desarrollo del código

1. **Lectura de sensores:** Se configura el ESP32 para leer los valores de temperatura y humedad del DHT22, y la presión atmosférica del BMP280.
2. **Visualización en OLED:** Se despliegan los datos en la pantalla OLED para su monitoreo local.
3. **Envío a ThingSpeak:** Se establece la conexión WiFi y se envían los datos a la plataforma en intervalos definidos.




sketch_mar11a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino BT
sketch_mar11a.ino
display.setTextColor(WHITE);
display.display();
}
void loop() {
float temperature = dht.readTemperature();
float humidity = dht.readHumidity();
float pressure = bmp.readPressure() / 100.0F; // Convertir a hPa
if (isnan(temperature) || isnan(humidity)) {
Serial.println("Error al leer DHT22.");
return;
}
// Mostrar datos en OLED
mostrarEnPantalla(temperature, humidity, pressure);
// Enviar datos a ThingSpeak
enviarThingSpeak(temperature, humidity, pressure);
delay(20000); // Espera 20 segundos
}
// Función para conectar a WiFi
void conectarWiFi() {
Serial.print("Conectando a WiFi...");
WiFi.begin(ssid, pass); // Corregimos el error aquí
int intentos = 0;
while (WiFi.status() != WL_CONNECTED && intentos < 20) {
delay(500);
Serial.print(".");
intentos++;
}
if (WiFi.status() == WL_CONNECTED) {
Serial.println("\nWiFi conectado.");
} else {
Serial.println("\nError: No se pudo conectar a WiFi.");
}
}
// Función para mostrar datos en pantalla OLED
void mostrarEnPantalla(float temp, float hum, float pres) {
display.clearDisplay();
display.setCursor(0, 0);
display.print("Temp: ");
display.print(temp);
Ln 142, Col 1 Arduino BT (not connected)
21°C Nublado
Buscar
ESP
10:46 p. m. 11/03/2025


sketch_mar11a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino BT
sketch_mar11a.ino
display.clearDisplay();
display.setCursor(0, 0);
display.print("Temp: ");
display.print(temp);
display.print(" C");
display.setCursor(0, 10);
display.print("Hum: ");
display.print(hum);
display.print(" %");
display.setCursor(0, 20);
display.print("Pres: ");
display.print(pres);
display.print(" hPa");
display.display();
}
// Función para enviar datos a ThingSpeak
void enviarThingSpeak(float temp, float hum, float pres) {
if (WiFi.status() != WL_CONNECTED) {
Serial.println("Error: WiFi desconectado. Intentando reconectar...");
conectarWiFi();
}
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
String url = String(THINGSPEAK_SERVER) + "?api_key=" + apiKey +
"&field1=" + String(temp) +
"&field2=" + String(hum) +
"&field3=" + String(pres);
Serial.println("Enviando datos a ThingSpeak...");
http.begin(url);
int httpCode = http.GET();
if (httpCode > 0) {
Serial.print("Respuesta HTTP: ");
Serial.println(httpCode);
if (httpCode == 200) {
Serial.println("Datos enviados correctamente.");
} else {
Serial.println("Error en ThingSpeak.");
}
} else {
Serial.println("Error al conectar con ThingSpeak.");
}
http.end();
}
}
Ln 142, Col 1 Arduino BT (not connected)
21°C Nublado
Buscar
ESP
10:47 p. m. 11/03/2025

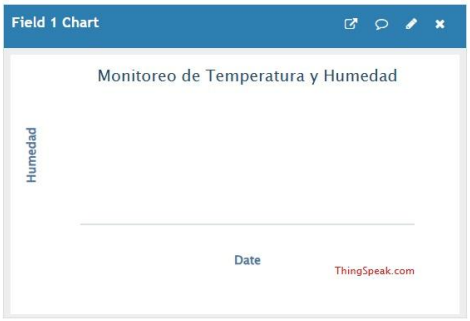
2.4 Configuración de ThingSpeak

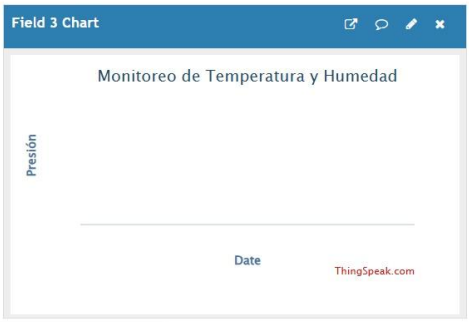
1. Crear una cuenta en ThingSpeak.
2. Configurar un canal para almacenar los datos.
3. Obtener las claves API para la transmisión de datos.
4. Configurar ThingSpeak para graficar la información.

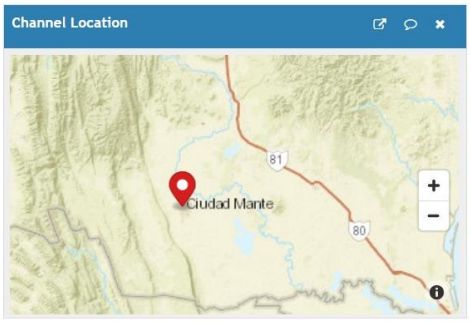
Unidad Académica Multidisciplinaria Mante

Universidad Autónoma de Tamaulipas

ThingSpeak™ Channels Apps Devices Support Commer







Tags humedad, dht22, bmp280

(Tags are comma separated)

Link to External Site http://

Link to GitHub https://github.com/

Elevation

Show Channel Location ☑

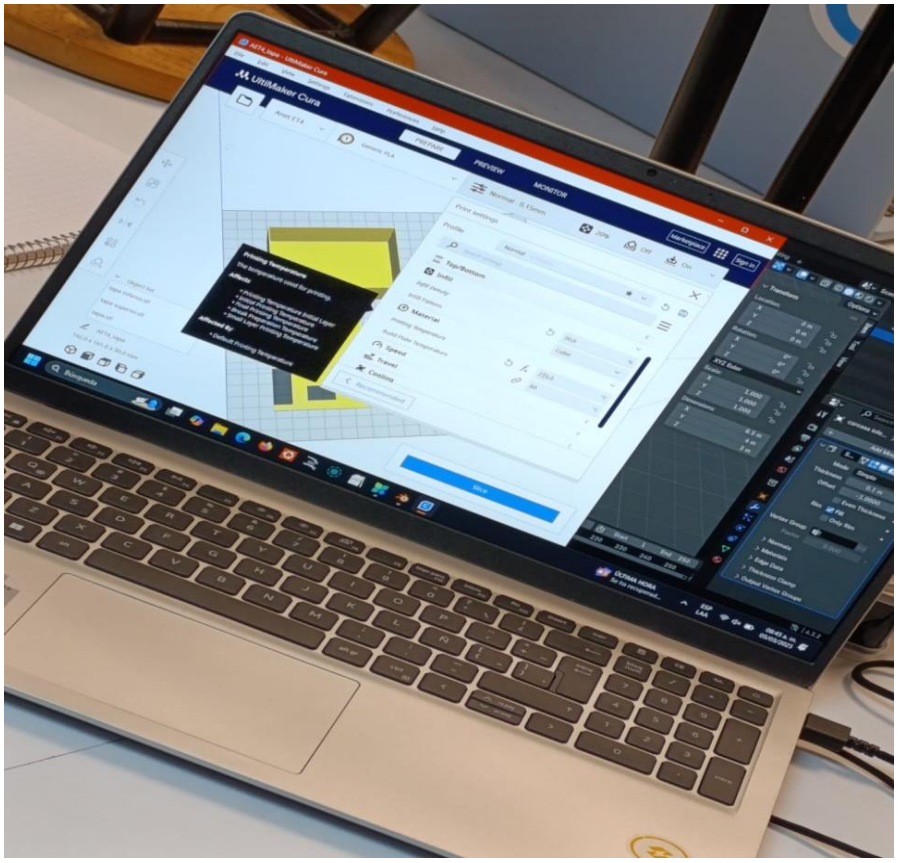
Latitude 22.7469426852

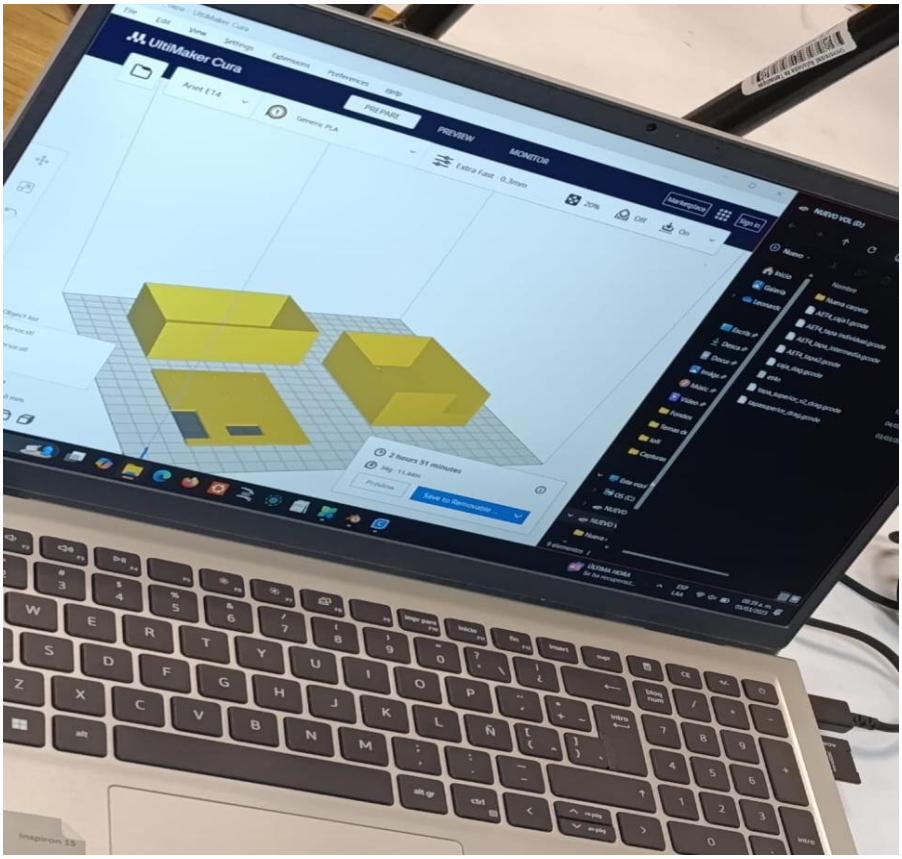
Longitude -98.983488471

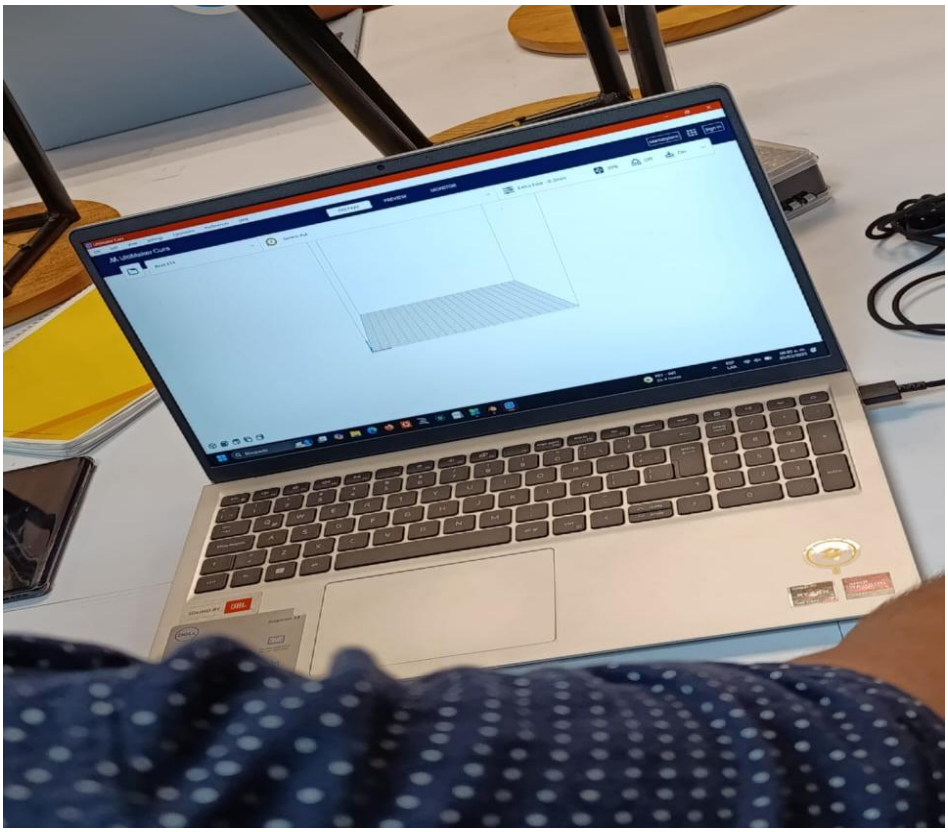
2.5 Fabricación de la carcasa impresa en 3D

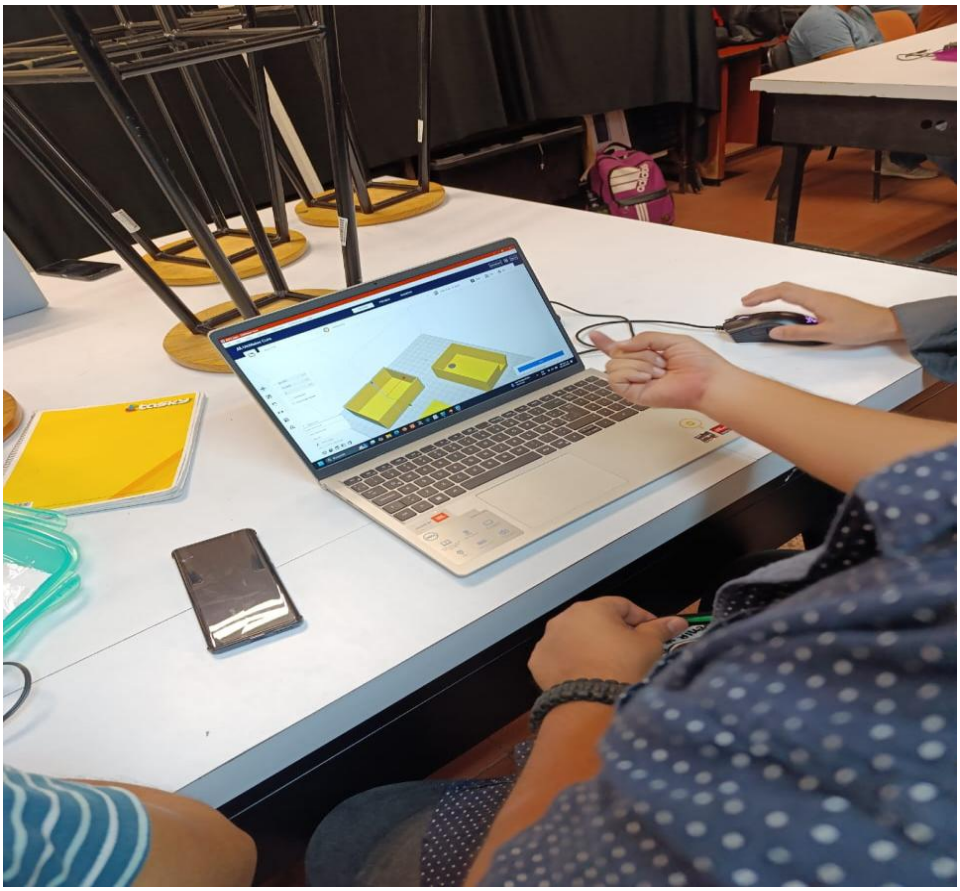
Para proteger y organizar los componentes del sistema meteorológico, se diseñó y fabricó una carcasa impresa en 3D. Esta estructura permite:

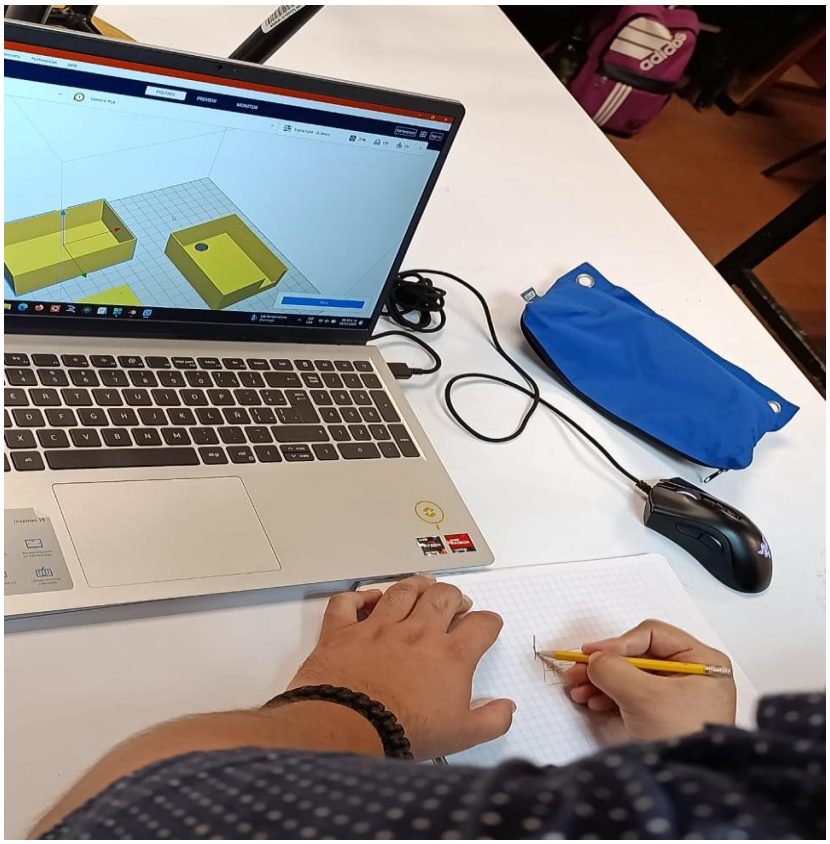
- Asegurar los sensores y la pantalla OLED en una posición estable.
- Proteger los componentes de condiciones externas como polvo y humedad.
- Facilitar el mantenimiento y acceso a los componentes internos.
- Mejorar la estética y portabilidad del sistema.

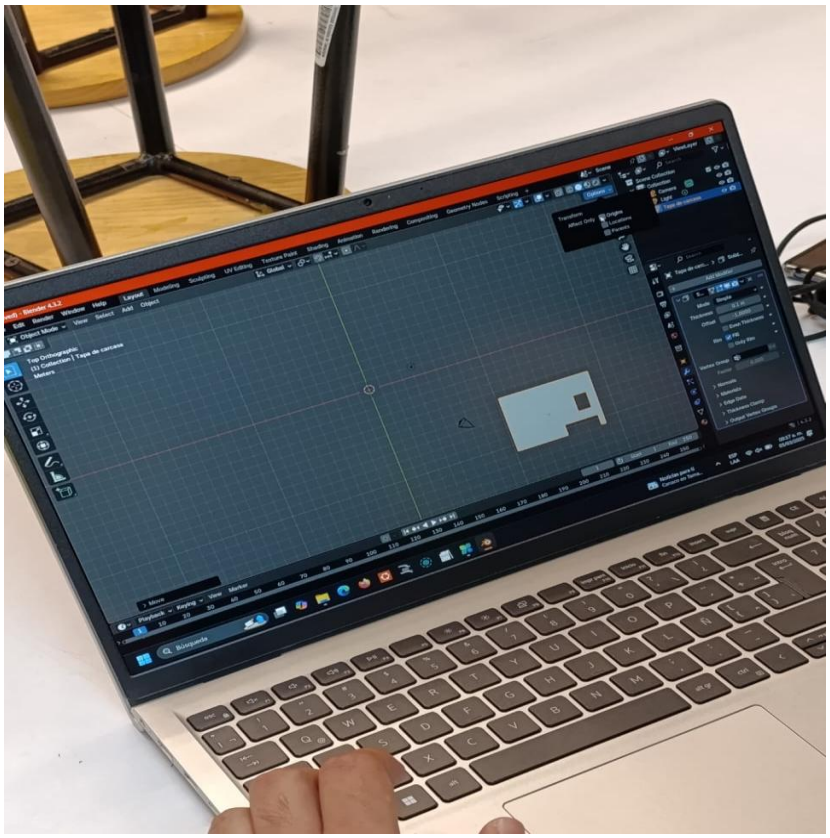

UltiMaker Cura
PREPARE
PREVIEW
MONITOR

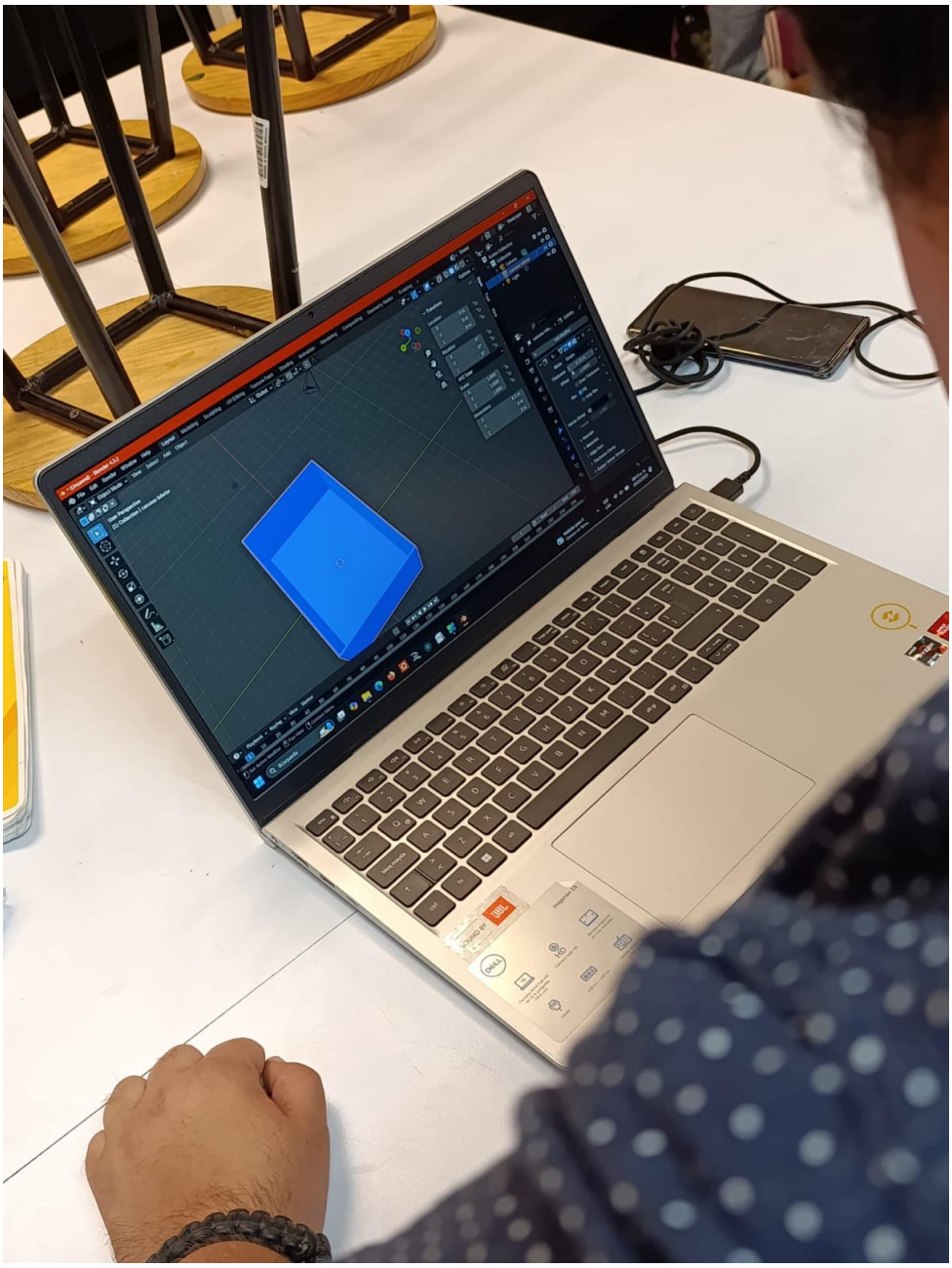

UltiMaker Cura
PREPARE
PREVIEW
MONITOR





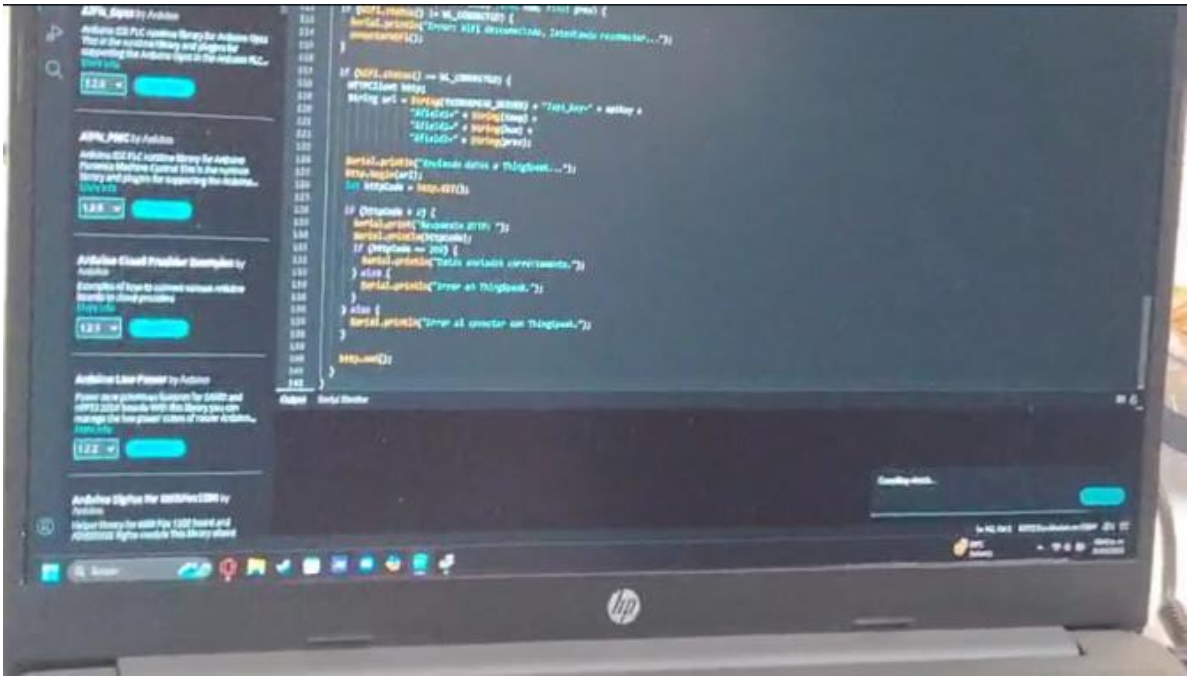


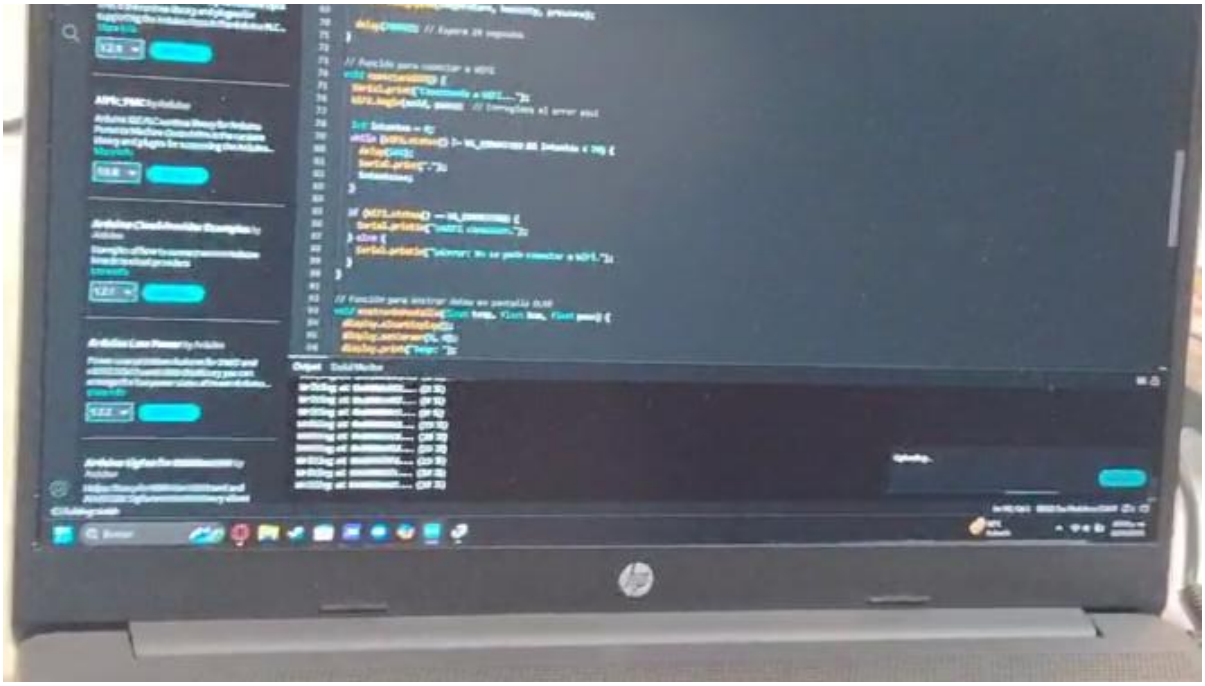


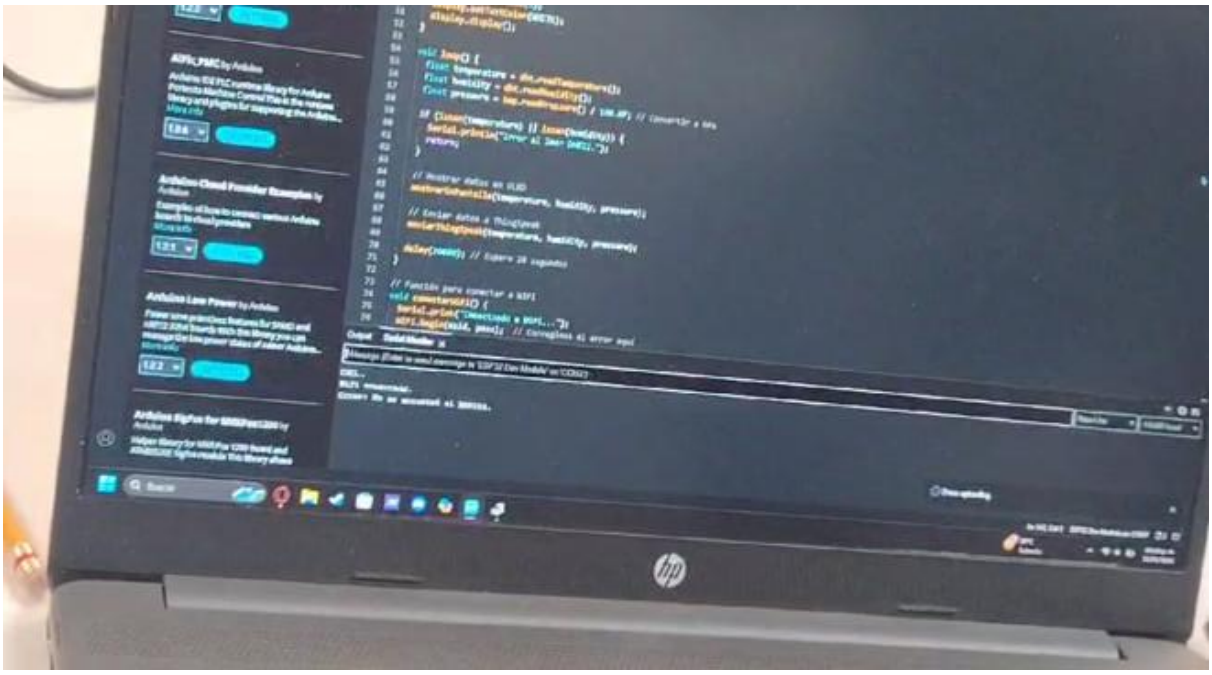


2.6 Pruebas y resultados

- Capturas de pantalla de los datos en la OLED y ThingSpeak.
- Comparación de valores con otros dispositivos de referencia para verificar la precisión.







Conclusión

Aun que tuvimos problemas con el DHT22 ya que encontramos un error y no lo solucionamos a tiempo El desarrollo de este sistema meteorológico basado en ESP32 demostró ser una solución eficiente para el monitoreo en tiempo real de temperatura, humedad y presión atmosférica. La integración con ThingSpeak permitió la visualización remota de los datos, facilitando su análisis y almacenamiento. En futuras versiones, se podrían agregar más sensores para obtener información más detallada, o mejorar la interfaz gráfica de la pantalla OLED para una mejor experiencia de usuario.

4. Bibliografía

- Datasheet del sensor DHT22: https://www.adafruit.com/datasheets/DHT22.pdf
- Datasheet del sensor BMP280: https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/
- Documentación de ESP32: https://docs.espressif.com/projects/esp-idf/en/latest/
- Documentación de ThingSpeak: https://www.mathworks.com/help/thingspeak/

Unidad Académica Multidisciplinaria Mante

Universidad Autónoma de TAMAULIPAS