UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

PROYECTO FINAL

MONITOREO AMBIENTAL REMOTO CON SERVIDOR WEB

CARRERA:	INGENIERÍA EN REDES Y TELECOMUNICACIONES
INTEGRANTES:	DAVID CHILO
MATERIA:	APLICACIONES CON MICROPROCESADORES
FECHA:	29/06/2023
SEMESTRE:	I - 2023
DOCENTE:	ING. MARCIO CARVAJAL CORDERO

PROYECTO FINAL

MONITOREO AMBIENTAL REMOTO CON SERVIDOR WEB

Objetivo:

Desarrollo de un sistema de monitoreo ambiental utilizando sensores BMP180, DHT11, DS18B20 y YL69, integrados con un servidor montado en un ESP32. Además, se implementará una funcionalidad para consultar los datos a través de Telegram y se visualizarán en la plataforma Blynk. El proyecto también incluirá un control de un foco de luz mediante un relay.

Caso de uso:

El proceso se describe a continuación:

1. Lectura de datos de los sensores: El sistema realizará mediciones periódicas de los sensores BMP180 (presión atmosférica), DHT11 (temperatura y humedad ambiente), DS18B20 (temperatura) y YL69 (humedad del suelo).
2. Almacenamiento y procesamiento de datos: Los datos medidos se guardarán en una base de datos local.
3. Visualización de datos: Los datos medidos y el estado del riego se mostrarán en una interfaz web accesible a través de un servidor montado en el ESP32. Los usuarios podrán ver los valores actuales y recibir alertas si los niveles están fuera de los umbrales predefinidos.
4. Consulta remota a través de Telegram: Se implementará una funcionalidad para que los usuarios puedan consultar los valores de los sensores a través de un bot de Telegram. Los comandos específicos se definirán para obtener la información deseada.
5. Integración con Blynk: Los datos y el control del riego también se visualizarán en la plataforma Blynk, permitiendo a los usuarios monitorear y controlar el sistema a través de una aplicación móvil.

Con este proyecto, se logrará un monitoreo continuo del entorno.

Marco teórico:

1. Sensores utilizados:

- BMP180: El sensor BMP180 es un sensor de presión atmosférica y temperatura. Utiliza un principio de medición basado en el sensor de presión piezoeléctrico y un termistor

para medir la temperatura ambiente. Proporciona datos precisos de presión y temperatura, que pueden utilizarse para estimar la altitud.

- DHT11: El sensor DHT11 es un sensor de temperatura y humedad relativa. Utiliza un termistor para medir la temperatura ambiente y un sensor capacitivo para medir la humedad relativa. Proporciona mediciones básicas de temperatura y humedad con una precisión limitada.
- DS18B20: El sensor DS18B20 es un sensor de temperatura digital de alta precisión. Utiliza una sonda de acero inoxidable resistente al agua para medir la temperatura ambiente. Proporciona mediciones de temperatura con una alta precisión y una resolución ajustable.
- YL69: El sensor YL69 es un sensor de humedad del suelo. Utiliza una configuración de resistencia variable para medir la humedad del suelo. Proporciona mediciones de humedad relativa del suelo, lo que permite determinar el nivel de humedad en el suelo.

2. ESP32 y servidor:

- ESP32: El ESP32 es un microcontrolador de bajo costo y bajo consumo de energía que ofrece conectividad Wi-Fi y Bluetooth. Es ampliamente utilizado en proyectos de IoT debido a su capacidad de procesamiento y su soporte para diversas bibliotecas y protocolos de comunicación.
- Servidor: El servidor se monta en el ESP32 y es responsable de recibir y procesar los datos de los sensores. Utiliza una combinación de software y hardware para proporcionar una interfaz web a través de la cual los usuarios pueden acceder a los datos y controlar el sistema.

3. Telegram y Blynk:

- Telegram: Telegram es una plataforma de mensajería que permite la comunicación a través de mensajes cifrados y proporciona una API que permite la integración de bots. Los usuarios podrán consultar los datos de los sensores y controlar el sistema a través de comandos enviados a un bot de Telegram.
- Blynk: Blynk es una plataforma IoT que permite crear interfaces personalizadas para monitorear y controlar dispositivos conectados. Proporciona una aplicación móvil que se puede utilizar para visualizar los datos de los sensores y controlar el sistema de riego.

4. Control del relay y foco de luz:

- Relay: El relay es un dispositivo electromecánico que se utiliza para controlar la conexión o desconexión de circuitos eléctricos. En este proyecto, se utilizará un relay para controlar la alimentación del foco de luz. Al activarse, el relay permitirá el paso de

corriente hacia el foco, encendiéndolo, y al desactivarse, interrumpirá la corriente, apagando el foco.

Lista de materiales:

Hardware:

- ESP32: Microcontrolador con conectividad Wi-Fi y Bluetooth.
- Sensor BMP180: Sensor de presión atmosférica y temperatura.
- Sensor DHT11: Sensor de temperatura y humedad relativa.
- Sensor DS18B20: Sensor de temperatura digital de alta precisión.
- Sensor YL69: Sensor de humedad del suelo.
- Relay: Dispositivo electromecánico para controlar el encendido y apagado del foco de luz.
- Foco de luz: Foco o bombilla que se controlará mediante el relay.
- Resistencias y cables de conexión: Para realizar las conexiones eléctricas entre los componentes.
- Protoboard o placa de pruebas: Para montar y conectar los componentes de manera provisional.
- Fuente de alimentación: Para proporcionar energía al ESP32 y a los sensores.

Software:

- IDE de desarrollo: Como Arduino IDE para programar el ESP32.
- Librerías: Librerías específicas para cada sensor (por ejemplo, Adafruit BMP180, DHT, OneWire, etc.).
- Bibliotecas de comunicación: Bibliotecas necesarias para la comunicación con Telegram y Blynk.

Imágenes:

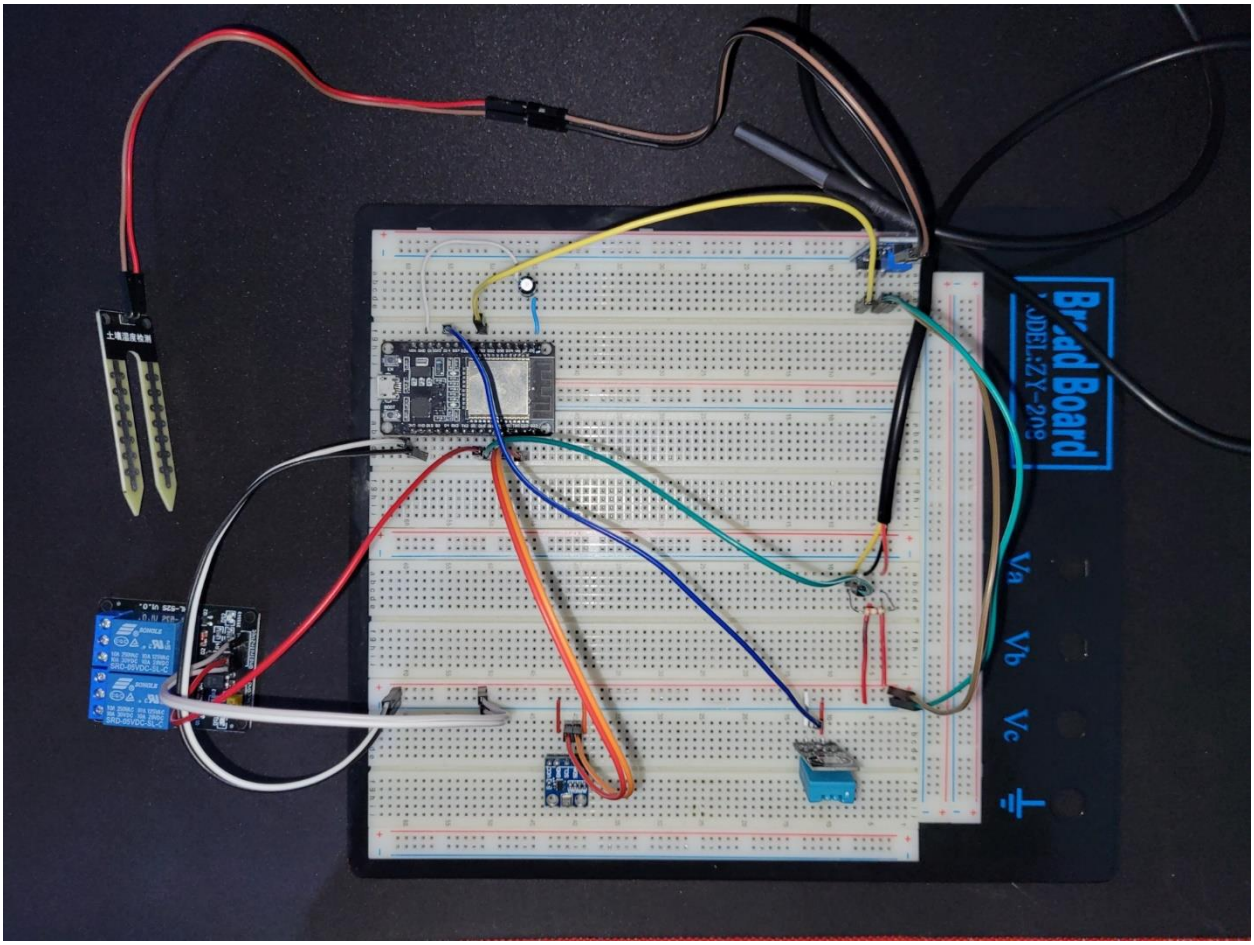
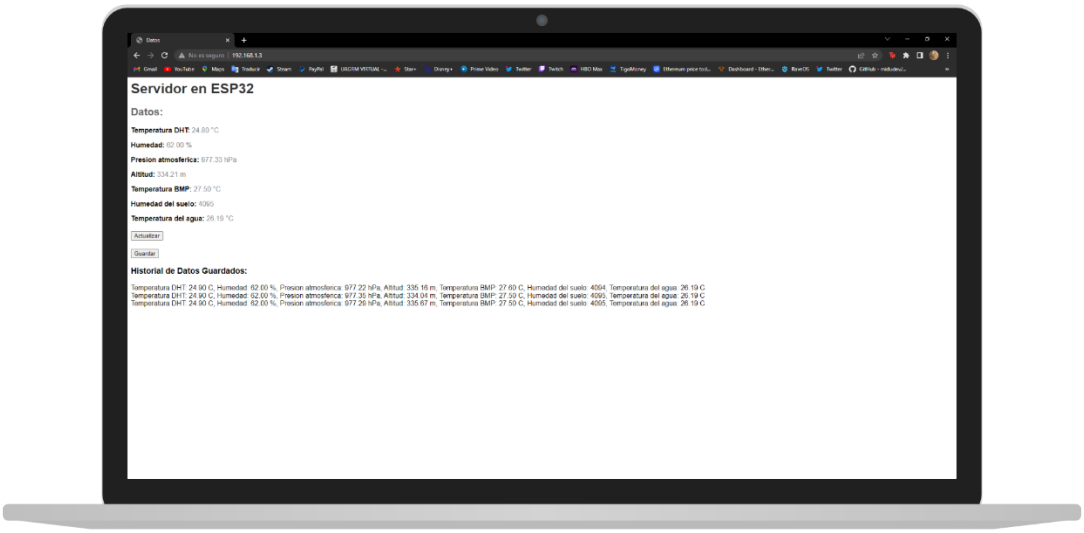
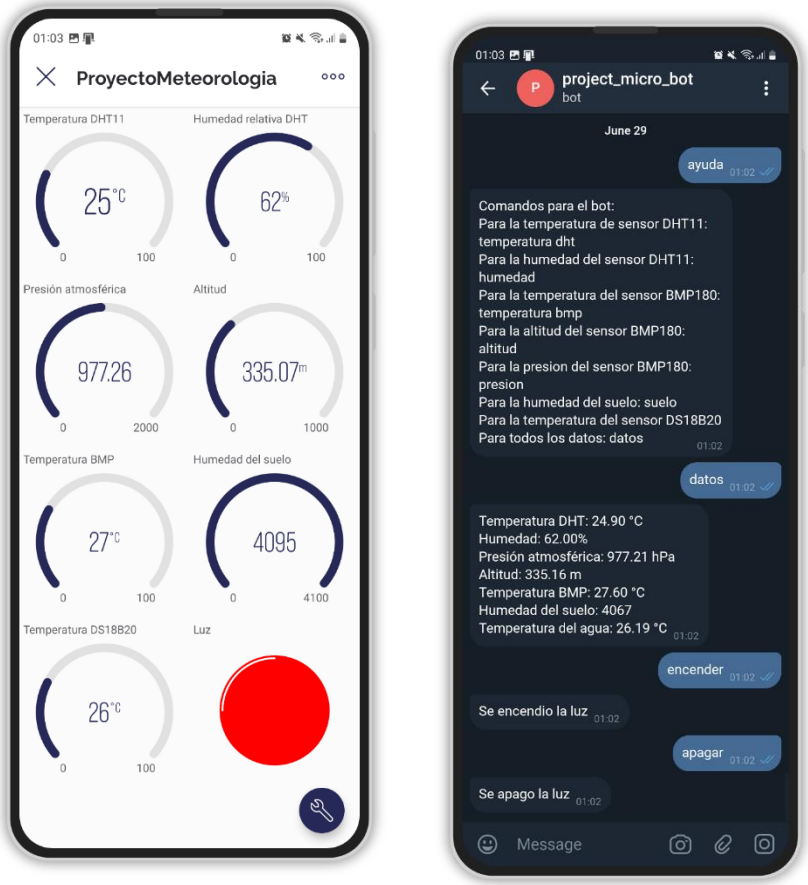


Foto del circuito





Código de programación:

```
#define BLYNK_TEMPLATE_ID "TMPL2i-3Sau1H"
#define BLYNK_TEMPLATE_NAME "ProyectoMeteorologia"
#define BLYNK_AUTH_TOKEN "U_o_AkwCDFz4WDP_SElx9BxdEXhGrZQN"

#include <CTBot.h>
#include <WiFi.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <Adafruit_BMP085.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WebServer.h>

#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp32.h>
```

```
char auth[] = BLYNK_AUTH_TOKEN;

template<class T> inline Print &operator <<(Print &obj, T arg) {
  obj.print(arg);
  return obj;
}

WebServer server(80);

// Configuracion de la red WiFi
char* ssid = "dch";
char* password = "xd123456";

// Configuracion del Bot de Telegram, cambiar a datos propios.
String token = "6207934903:AAFG7LUHSymR2YDVgTYVQa2FNZ2tY94VfZc";
const int64_t telegramId = 123456;

// Configuraciones de los pines de los sensores
#define DHTPIN 14
#define BMP_SDA 21
#define BMP_SCL 22
#define ONEWIRE_PIN 19

// Inicializando de los sensores
DHT dht(DHTPIN, DHT11);
Adafruit_BMP085 bmp;
OneWire oneWire(ONEWIRE_PIN);
DallasTemperature ds18b20(&oneWire);

int suelo = 33;

CTBot mybot;

float humidity, temperature, pressure, temperatureDs18b20, temperatureBMP,
altitude;
int sueloLectura;
String Data, savedData, lastSavedData, ayuda;
float liveh, livet, livep, livetds, livebmp, livea;
int lives;
bool isMeasuring;

BlynkTimer timer;

int luz = 18;
```

```
bool luzEncendida = false;
int luzBlynk;

float tiempo = 0;


void sendSensor(){
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V2, pressure);

  Blynk.virtualWrite(V3, altitude);

  Blynk.virtualWrite(V4, temperatureBMP);
  Blynk.virtualWrite(V5, sueloLectura);
  Blynk.virtualWrite(V6, temperatureDs18b20);

  Blynk.virtualWrite(V7, luzBlynk);
}



void handleSave() {
  lastSavedData = "Temperatura DHT: " + String(temperature) + "  C, " +"Humedad:
" + String(humidity) + " %, " + "Presion atmosferica: " + String(pressure) + "
hPa, " + "Altitud: " + String(altitude) + " m, " + "Temperatura BMP: " +
String(temperatureBMP) + " C, " + "Humedad del suelo: " + String(sueloLectura) +
", " + "Temperatura del agua: " + String(temperatureDs18b20) + " C";
  savedData += lastSavedData + "<br>";
  server.sendHeader("Location", "/");
  server.send(303);
}

void handleView() {
  isMeasuring = true;
  server.sendHeader("Location", "/");
  server.send(303);
}

void handleRoot() {
  String html = "<html><head>";
  html += "<style>";
  html += "body { font-family: Arial, sans-serif; }";
  html += "h1 { color: #333; }";
  html += "h2 { color: #666; }";
```

```
  html += "p { margin-bottom: 10px; }";
  html += "b { font-weight: bold; }";
  html += "span { color: #888; }";
  html += "form { margin-top: 20px; }";
  html += "</style>";
  html += "<title>Datos</title>";

  // Mostrar los datos en una tabla
  html += "<body><h1>Servidor en ESP32</h2>";
  html += "<h2>Datos:</h2>";

  html += "<p><b>Temperatura DHT:</b> <span id='temperature'>" + String(livet) +
" &deg;C</span></p>";
  html += "<p><b>Humedad:</b> <span id='humedad'>" + String(liveh) + "
%</span></p>";
  html += "<p><b>Presion atmosferica:</b> <span id='presion'>" + String(livep) +
" hPa</span></p>";
  html += "<p><b>Altitud:</b> <span id='altitud'>" + String(livea) + "
m</span></p>";
  html += "<p><b>Temperatura BMP:</b> <span id='temperaturebmp'>" +
String(livebmp) + " &deg;C</span></p>";
  html += "<p><b>Humedad del suelo:</b> <span id='humedadsuelo'>" + String(lives)
+ "</span></p>";
  html += "<p><b>Temperatura del agua:</b> <span id='temperature'>" +
String(livetds) + " &deg;C</span></p>";

  html += "<form action=\"/start\">";
  html += "<input type=\"submit\" value=\"Actualizar\">";
  html += "</form>";
  html += "<form action=\"/save\">";
  html += "<input type=\"submit\" value=\"Guardar\">";
  html += "</form>";

  html += "<h3>Historial de Datos Guardados:</h3>";
  html += "<p>" + savedData + "</p>";

  html += "</body></html>";

  server.send(200, "text/html", html);
}

void setup() {
  Serial.begin(115200);
  Serial.println("Iniciando Bot...");
```

```
pinMode(luz, OUTPUT);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Conectando a la red Wi-Fi...");
}

Serial.println("Conexión Wi-Fi establecida");
Serial.print("Dirección IP del servidor: ");
Serial.println(WiFi.localIP());

server.on("/", handleRoot);
server.on("/save", handleSave);
server.on("/start", handleView);
server.begin();
Serial.println("Servidor iniciado");


mybot.wifiConnect(ssid, password);

mybot.setTelegramToken(token);

dht.begin();

if (!bmp.begin()) {
  Serial.println("Error en el sensor");
  while (1); //bucle infinito
}

if (mybot.testConnection()) {
  Serial.println("\n Conectado");
} else {
  Serial.println("\n No conectado");
}

ayuda = "Comandos para el bot:\n";
ayuda += "Para la temperatura de sensor DHT11: temperatura dht\n";
ayuda += "Para la humedad del sensor DHT11: humedad\n";
ayuda += "Para la temperatura del sensor BMP180: temperatura bmp\n";
ayuda += "Para la altitud del sensor BMP180: altitud\n";
ayuda += "Para la presion del sensor BMP180: presion\n";
ayuda += "Para la humedad del suelo: suelo\n";
ayuda += "Para la temperatura del sensor DS18B20\n";
ayuda += "Para todos los datos: datos";
```

```
  Blynk.begin(auth, ssid, password);
  timer.setInterval(1000L, sendSensor);

}

void loop() {
  TBMessage msg;

  // Lectura de los datos de los sensores
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();

  temperatureBMP = bmp.readTemperature();
  altitude = bmp.readAltitude(101700);
  pressure = bmp.readPressure() / 100.0;

  sueloLectura = analogRead(suelo);

  ds18b20.requestTemperatures();
  temperatureDs18b20 = ds18b20.getTempCByIndex(0);

  if (isMeasuring) {
    liveh = humidity;
    livet = temperature;
    livep = pressure;
    livetds = temperatureDs18b20;
    livebmp = temperatureBMP;
    livea = altitude;
    lives = sueloLectura;
  }

  // Control de la luz
  if (luzEncendida == true) {
    digitalWrite(luz, HIGH);
    luzBlynk = 255;
  } else {
    digitalWrite(luz, LOW);
    luzBlynk = 0;
  }

  // Datos
  Data = "Temperatura DHT: " + String(temperature) + " °C\n";
  Data += "Humedad: " + String(humidity) + "%\n";
  Data += "Presión atmosférica: " + String(pressure) + " hPa\n";
```

```
Data += "Altitud: " + String(altitude) + " m\n";
Data += "Temperatura BMP: " + String(temperatureBMP) + " °C\n";
Data += "Humedad del suelo: " + String(sueloLectura) + "\n";
Data += "Temperatura del agua: " + String(temperatureDs18b20) + " °C\n";


// Consultas por Telegram
if (CTBotMessageText == mybot.getNewMessage(msg)) {
  Serial << "Mensaje: " << msg.sender.firstName << " - " << msg.text << "\n";

  if (msg.text.equalsIgnoreCase("temperatura dht")) {
    Serial.print("Temperatura DHT");
    mybot.sendMessage(msg.sender.id, String(temperature) + "°C");
  }
  if (msg.text.equalsIgnoreCase("humedad")) {
    Serial.print("humedad DHT");
    mybot.sendMessage(msg.sender.id, String(humidity) + "%");
  }
  if (msg.text.equalsIgnoreCase("temperatura bmp")) {
    Serial.print("Temperatura BMP");
    mybot.sendMessage(msg.sender.id, String(temperatureBMP) + "°C");
  }
  if (msg.text.equalsIgnoreCase("altitud")) {
    Serial.print("Altitud");
    mybot.sendMessage(msg.sender.id, String(altitude) + "m");
  }
  if (msg.text.equalsIgnoreCase("presion")) {
    Serial.print("Presion");
    mybot.sendMessage(msg.sender.id, String(pressure) + " hPa");
  }
  if (msg.text.equalsIgnoreCase("suelo")) {
    Serial.print("Suelo");
    mybot.sendMessage(msg.sender.id, String(sueloLectura));
  }
  if (msg.text.equalsIgnoreCase("temperatura agua")) {
    Serial.print("Temperatura DS18B20");
    mybot.sendMessage(msg.sender.id, String(temperatureDs18b20) + "°C");
  }
  if (msg.text.equalsIgnoreCase("datos")) {
    Serial.print("Todos los datos");
    Serial.print(msg.sender.id);
    mybot.sendMessage(msg.sender.id, String(Data));
  }
  if (msg.text.equalsIgnoreCase("guardar")) {
    Serial.print("Guardado");
```

```
      mybot.sendMessage(msg.sender.id, "Datos guardados");
      handleSave();
    }
    if (msg.text.equalsIgnoreCase("ayuda")) {
      Serial.print("Comandos");
      mybot.sendMessage(msg.sender.id, ayuda);
      handleSave();
    }
    if (msg.text.equalsIgnoreCase("encender")) {
      Serial.print("Luz");
      if (luzEncendida == true){
        mybot.sendMessage(msg.sender.id, "La luz ya esta encendida");
      } else {
        luzEncendida = true;
        mybot.sendMessage(msg.sender.id, "Se encendio la luz");
      }
    }
    if (msg.text.equalsIgnoreCase("apagar")) {
      Serial.print("Luz");
      if (luzEncendida == false){
        mybot.sendMessage(msg.sender.id, "La luz ya esta apagada");
      } else {
        luzEncendida = false;
        mybot.sendMessage(msg.sender.id, "Se apago la luz");
      }
    }
  }

  // ALERTAS POR TELEGRAM
  int test1 = 41;
  //Alertas de temperatura
  if (millis() - tiempo > 5000) {
    if (temperature > 40 || temperatureBMP > 40) {
      mybot.sendMessage(telegramId, "Temperatura elevada");
    }
    if (sueloLectura < 2500) {
      mybot.sendMessage(telegramId, "El suelo esta mojado");
    }
    if (temperatureDs18b20 > 40) {
      mybot.sendMessage(telegramId, "Temperatura del agua elevada");
    }
    if (humidity > 90) {
      mybot.sendMessage(telegramId, "Humedad relativa elevada");
    }
    if (test1 > 40){
```

```
      mybot.sendMessage(telegramId, "Test Realizadoo!!!");
    }
    tiempo = millis();
  }


  server.handleClient();
  delay(500);

  Blynk.run();
  timer.run();
}
```

Conclusión:

En conclusión, el proyecto propuesto tiene como objetivo desarrollar un sistema de monitoreo ambiental utilizando varios sensores, como el BMP180, DHT11, DS18B20 y YL69, junto con un microcontrolador ESP32. El sistema recopila datos de temperatura, humedad, presión atmosférica y humedad del suelo, y los muestra en un servidor montado en el ESP32. Además, se implementa la posibilidad de consultar los datos a través de Telegram y se visualizan en la plataforma Blynk. También se incluye el control de un foco de luz mediante un relay.

Este proyecto brinda la oportunidad de obtener información detallada sobre el entorno, lo que puede ser especialmente útil en aplicaciones como el monitoreo de condiciones climáticas en tiempo real, el control de riego de un jardín o la automatización de tareas relacionadas con el ambiente.

La integración de Telegram y Blynk agrega funcionalidades adicionales al proyecto, permitiendo la consulta remota de datos y el control del sistema desde cualquier ubicación. Esto facilita la supervisión y el control a distancia, lo que resulta conveniente para los usuarios.

En resumen, este proyecto combina la adquisición de datos de sensores ambientales, su visualización en una interfaz web, la consulta remota a través de Telegram y el control de un dispositivo externo, ofreciendo un sistema integral para monitorear y controlar el entorno de manera eficiente y conveniente.