Taller 5. Protocolo REST

Diego Iván Perea Montealegre (2185751) [diego.perea@uao.edu.co](mailto:diego.perea@uao.edu.co)

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

Para realizar los protocolos de REST se creó una carpeta para realizar todos los procesos de realización de obtención y postulo de datos , y así dar todas las configuraciones e instalaciones de json.

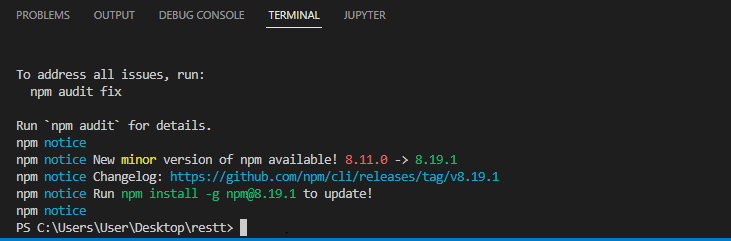
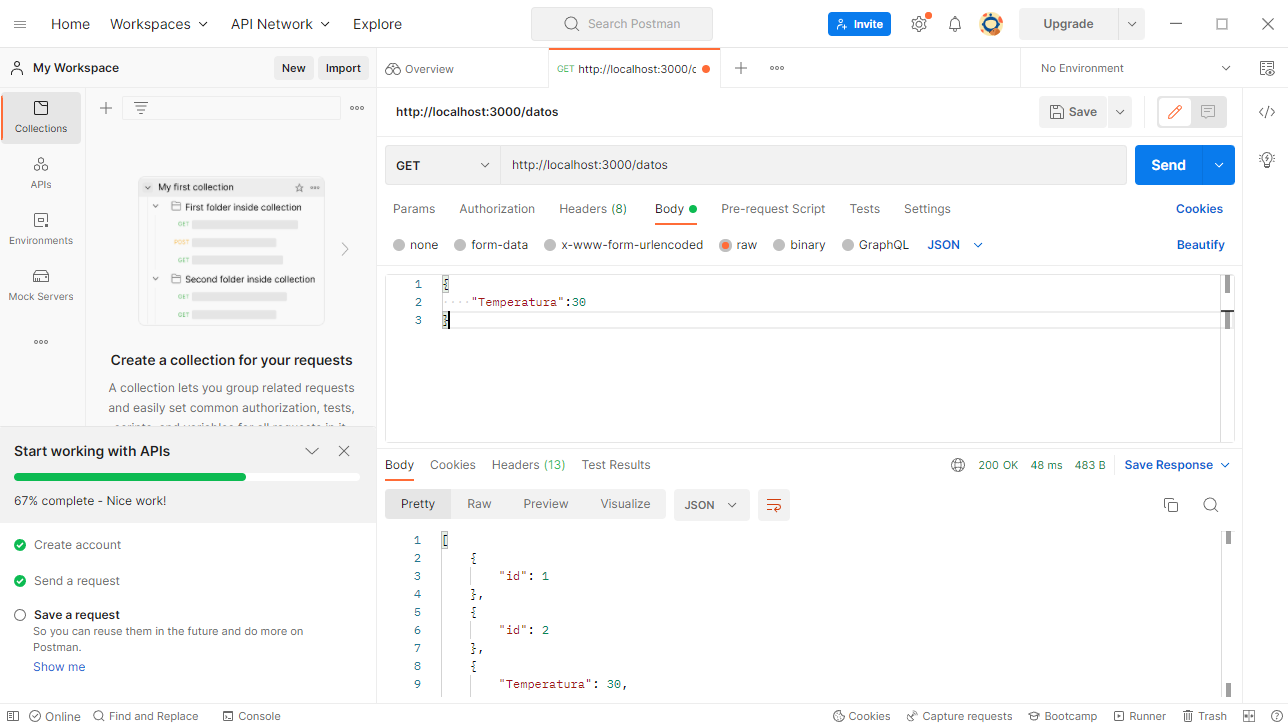


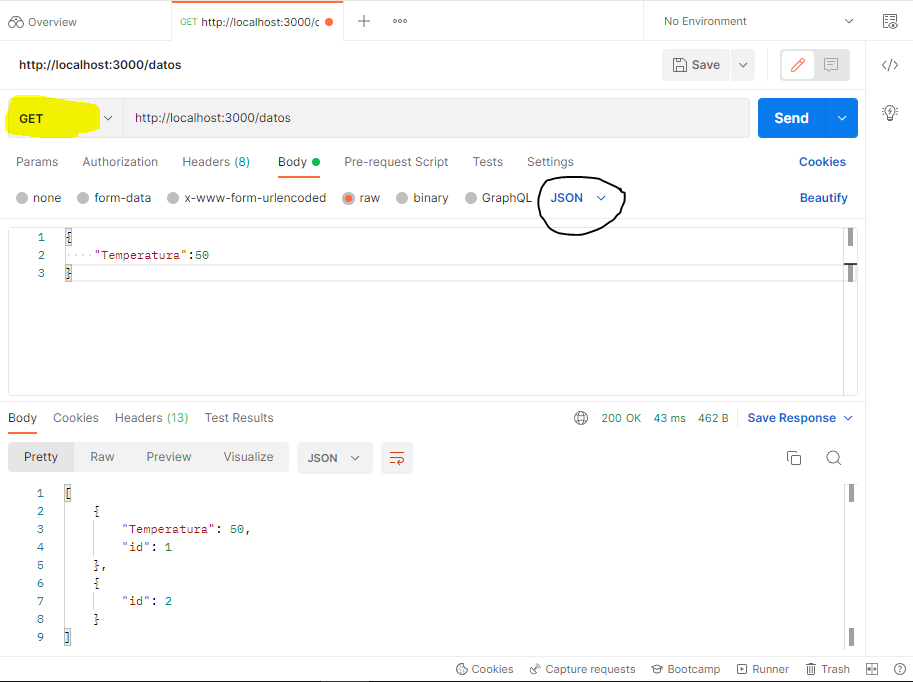
Figura 1. instalación de json server en la carpeta

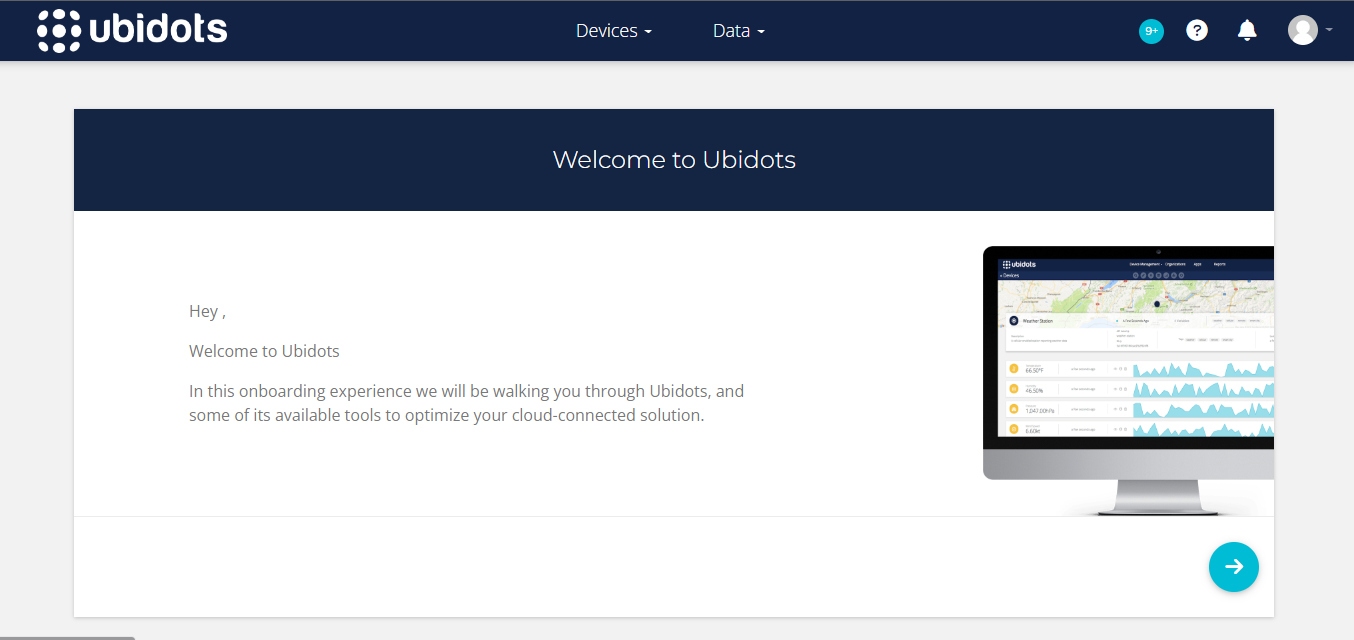
Se realiza el siguiente código

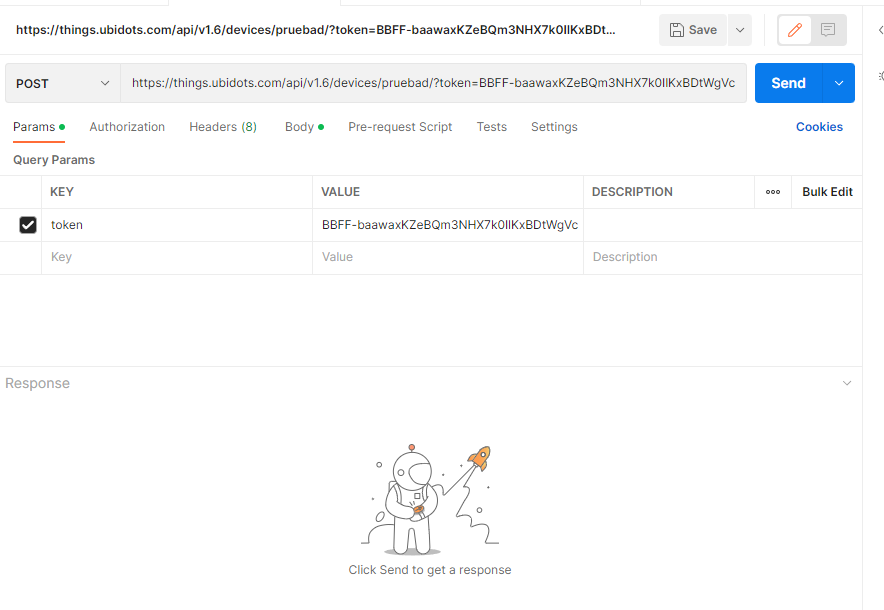


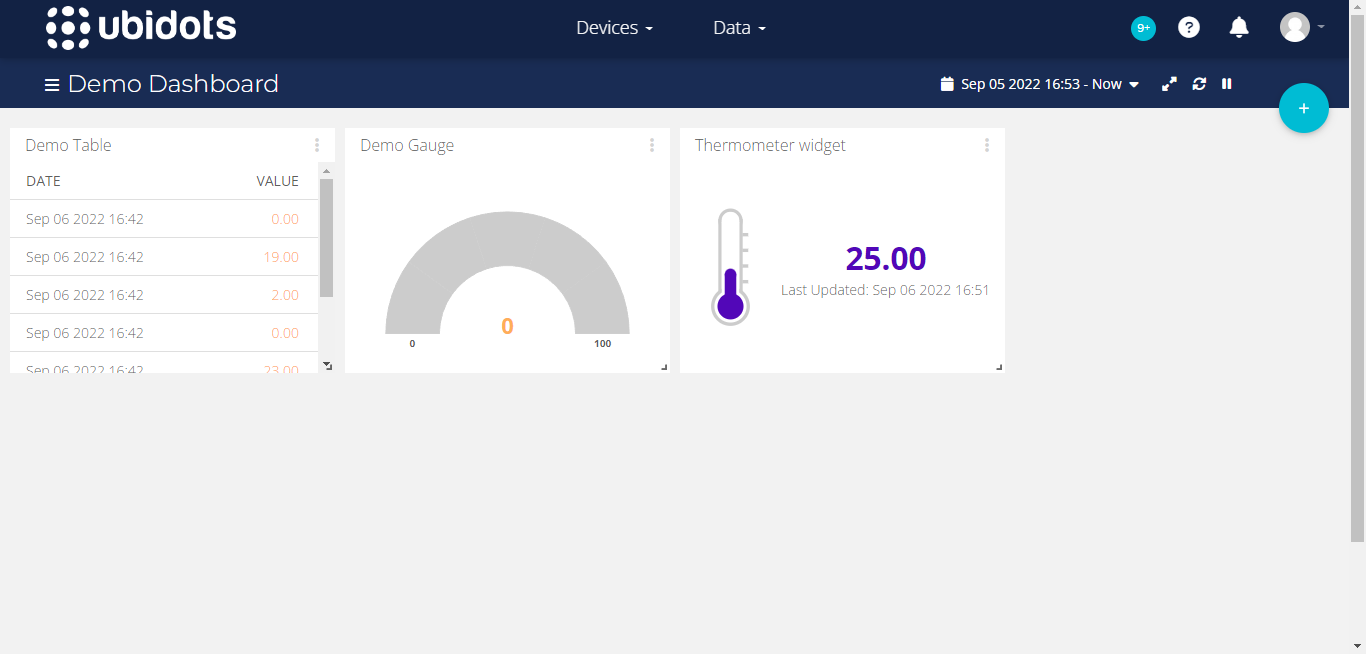


Se realiza con get (obtiene el dato), post(se postea el dato) , delete (eliminar el dato) y put (modificar el dato )









Ahora con el ESP32

#include <Arduino.h>

#include <HTTPClient.h>

#include <WiFi.h>

#include <ArduinoJson.h>

//LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)

#include <Adafruit\_Sensor.h>

#include <DHT.h>

//DEFINICION DE PINES DHT11

#define DHTPIN 4   // 4 = PIN D4

#define DHTTYPE    DHT11

DHT dht(DHTPIN, DHTTYPE);

//potenciometro ph

const int portPin=34;

int valor=0;

const char\* ssid = "\*\*\*"; //El SSID de la red wifi a la que se conectará

const char\* password = "\*\*\*\*"; //El password para conectarse a la red inalambrica

void setup\_wifi() {

  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

  delay(500);

  Serial.print(".");

  }

  Serial.println("");

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

}

void setup() {

  Serial.begin(9600); //Serial connection

  setup\_wifi(); //WiFi connection

  delay(1500);

}

void loop() {

  //CODIGO----TEMPERATURA Y HUMEDAD---------------

  float h= dht.readHumidity();

  float t =dht.readTemperature();

//potenciometro ph

  valor=analogRead(portPin)/292.5;

  //----------------------

  String variable;

  DynamicJsonDocument doc(1024); //creacion del json

  doc["temperatura"] = t;

  doc["humedad"] = h;

  doc["Ph"] = valor;

  serializeJson(doc, variable);

  Serial.println("dato a enviar: "+ variable);

  HTTPClient http; //Declare object of class HTTPClient

  WiFiClient client;

  //Specify request destination

  http.begin(client, "http://192.168.\*\*\*:3000/datos/");

  http.addHeader("Content-Type", "application/json"); //Specify contenttype header

  int httpCode = http.POST(variable); //Send the request

  String payload = http.getString(); //Get the response payload

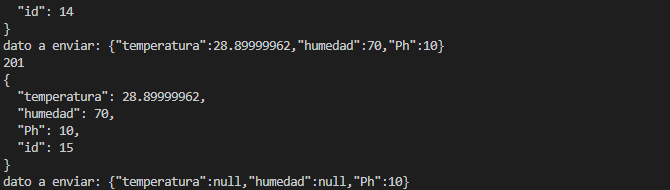
  Serial.println(httpCode); //Print HTTP return code

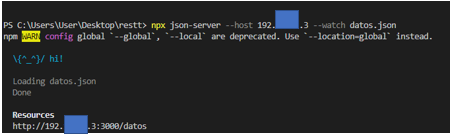
  Serial.println(payload); //Print request response payload

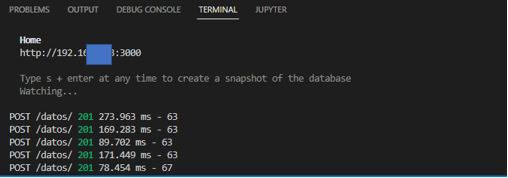
  http.end(); //Close connection

  delay(5000); //Send a request every 5 seconds

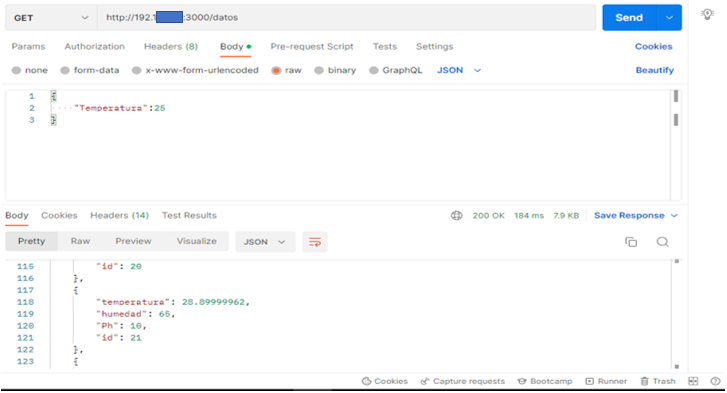
}







Post de envio de datos con cmd



Visualizacion de datos con método get con postman

Procedimiento – Plataforma Iot Con Soporte Rest con valores sensados

Código :

#include <Arduino.h>

#include <HTTPClient.h>

#include <WiFi.h>

#include <ArduinoJson.h>

//LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)

#include <Adafruit\_Sensor.h>

#include <DHT.h>

//DEFINICION DE PINES DHT11

#define DHTPIN 4   // 4 = PIN D4

#define DHTTYPE    DHT11

DHT dht(DHTPIN, DHTTYPE);

//potenciometro ph

const int portPin=34;

int valor=0;

const char\* ssid = "\*\*\*name\*\*wifi"; //El SSID de la red wifi a la que se conectará

const char\* password = "\*\*\*"; //El password para conectarse a la red inalambrica

void setup\_wifi() {

  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

  delay(500);

  Serial.print(".");

  }

  Serial.println("");

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

}

void setup() {

  Serial.begin(9600); //Serial connection

  setup\_wifi(); //WiFi connection

  delay(1500);

}

void loop() {

  //CODIGO----TEMPERATURA Y HUMEDAD---------------

  float h= dht.readHumidity();

  float t =dht.readTemperature();

//potenciometro ph

  valor=analogRead(portPin)/292.5;

  //----------------------

  String variable;

  DynamicJsonDocument doc(1024); //creacion del json

  doc["temperatura"] = t;

  doc["humedad"] = h;

  doc["Ph"] = valor;

  serializeJson(doc, variable);

  Serial.println("dato a enviar: "+ variable);

  HTTPClient http; //Declare object of class HTTPClient

  WiFiClient client;

  //Specify request destination

  //http.begin(client, "URL A INGRESAR");

  //http.begin(client, "http://192.168.\*.\*:3000/datos/"); LOCAL URL

  http.begin(client, "http://things.ubidots.com/api/v1.6/devices/esp32/?token=BBFF-baawaxKZeBQm3NHX7k0IlKxBDtWgV\*");//put your token , pon tu token

  http.addHeader("Content-Type", "application/json"); //Specify contentype header

  int httpCode = http.POST(variable); //Send the request

  String payload = http.getString(); //Get the response payload

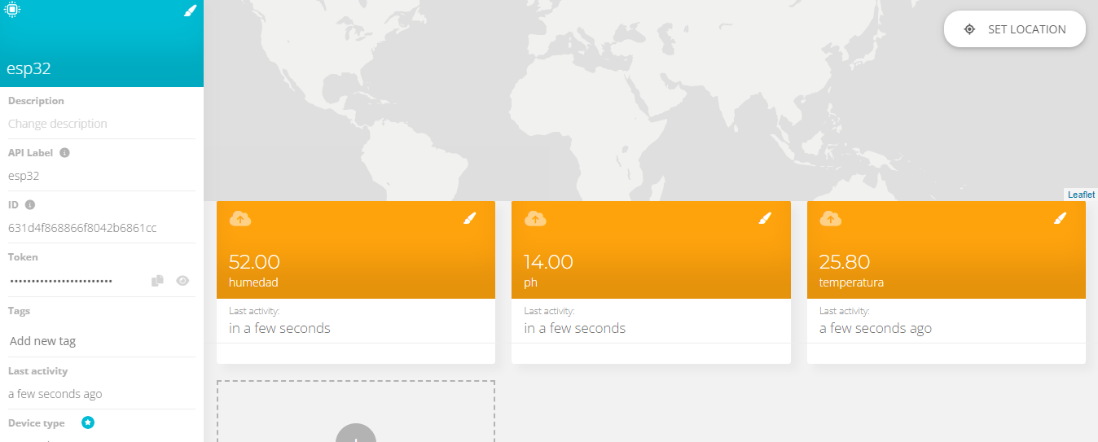
  Serial.println(httpCode); //Print HTTP return code

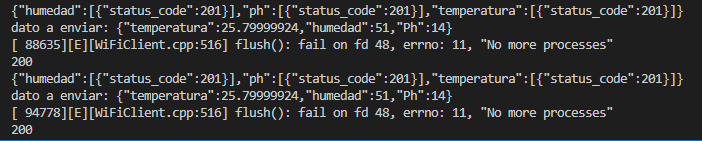
  Serial.println(payload); //Print request response payload

  http.end(); //Close connection

  delay(5000); //Send a request every 5 seconds

}





Referencias

[1]2022. [Online]. Available: https://ubidots.com/community/t/esp32-and-ubidots-subscribing-multiple-variables/3829. [Accessed: 16- Sep- 2022]

[2]2022. [Online]. Available: https://ubidots.com/community/t/esp32-and-ubidots-subscribing-multiple-variables/3829/2. [Accessed: 16- Sep- 2022]