Taller 5. Protocolo REST

Diego Iván Perea Montealegre (2185751) diego.perea@uao.edu.co

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

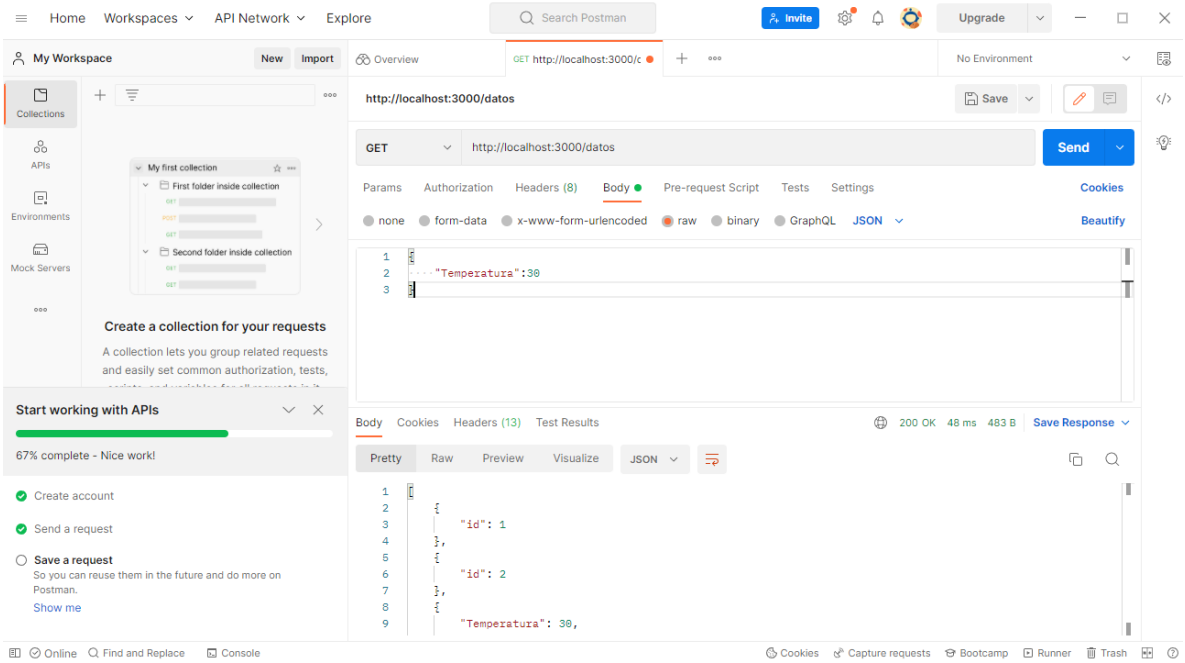
Para realizar los protocolos de REST se creó una carpeta para realizar todos los procesos de realización de obtención y postulo de datos , y así dar todas las configuraciones e instalaciones de json.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER


To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.11.0 -> 8.19.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.19.1
npm notice Run npm install -g npm@8.19.1 to update!
npm notice
PS C:\Users\User\Desktop\restt>
```

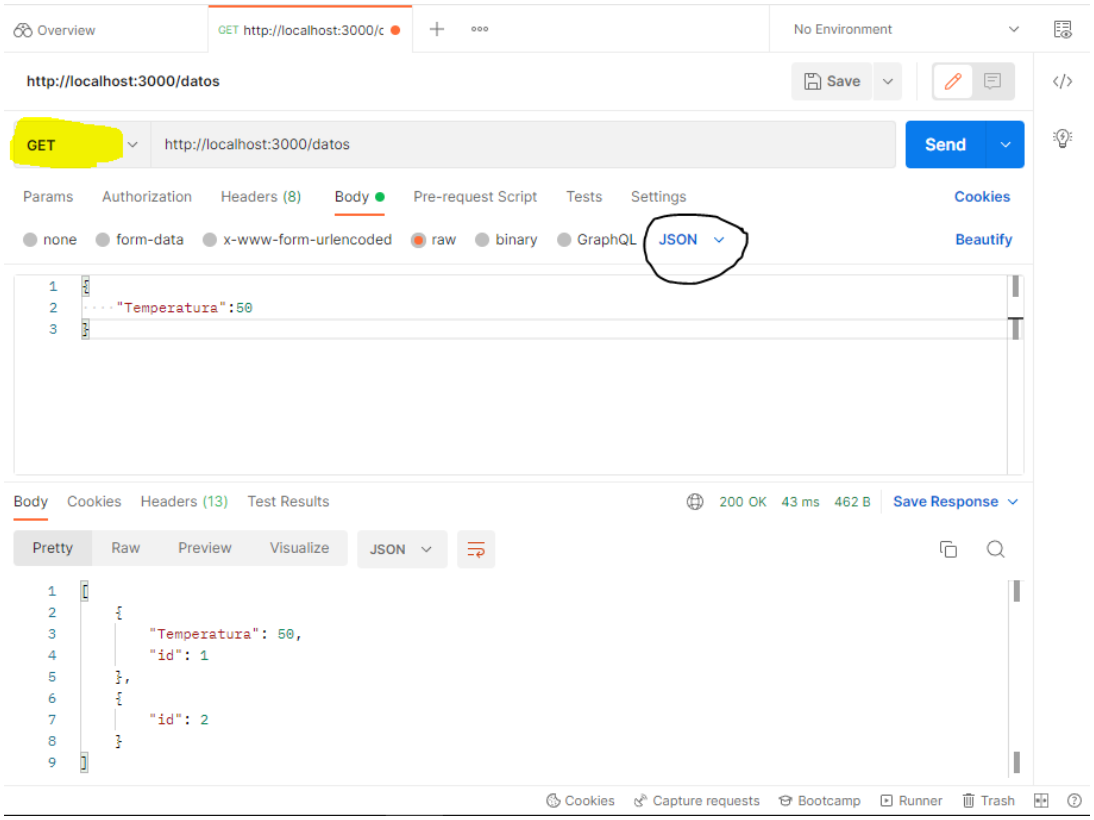
Figura 1. instalación de json server en la carpeta

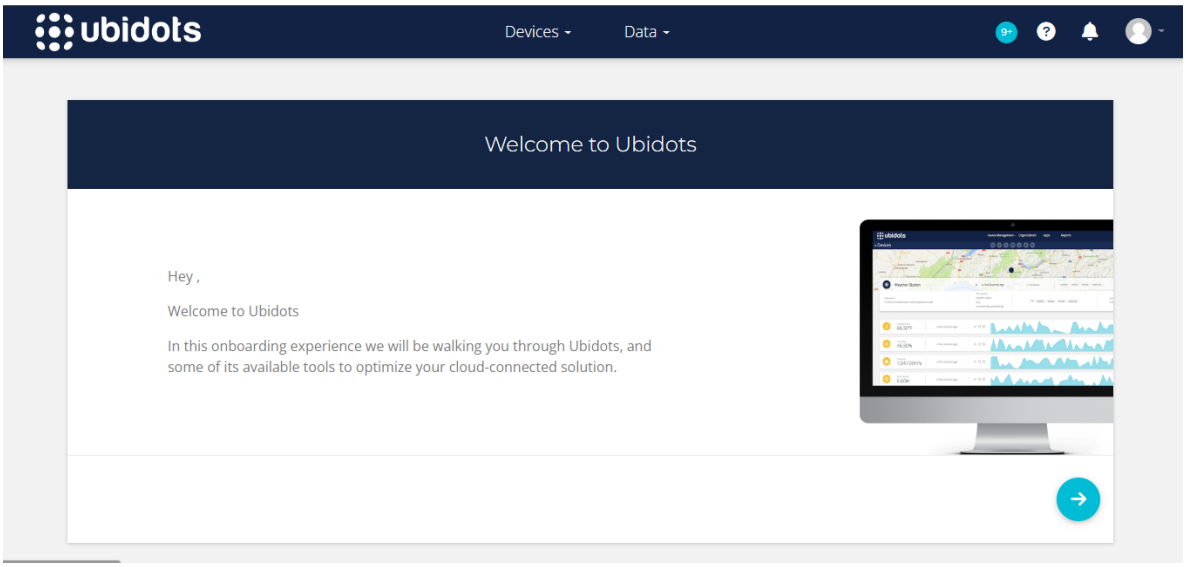
Se realiza el siguiente código

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\User\Desktop\restt> npx  json-server --watch datos.json
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead
.

  \{^_^}/ hi!

  Loading datos.json
  Done
```



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




ubidots
Devices
Data
9+
Welcome to Ubidots
Hey ,
Welcome to Ubidots
In this onboarding experience we will be walking you through Ubidots, and some of its available tools to optimize your cloud-connected solution.

https://things.ubidots.com/api/v1.6/devices/pruebad/?token=BBFF-baawaxKZeBQm3NHX7k0IlKxBDt... Save

POST https://things.ubidots.com/api/v1.6/devices/pruebad/?token=BBFF-baawaxKZeBQm3NHX7k0IlKxBDtWgVc Send

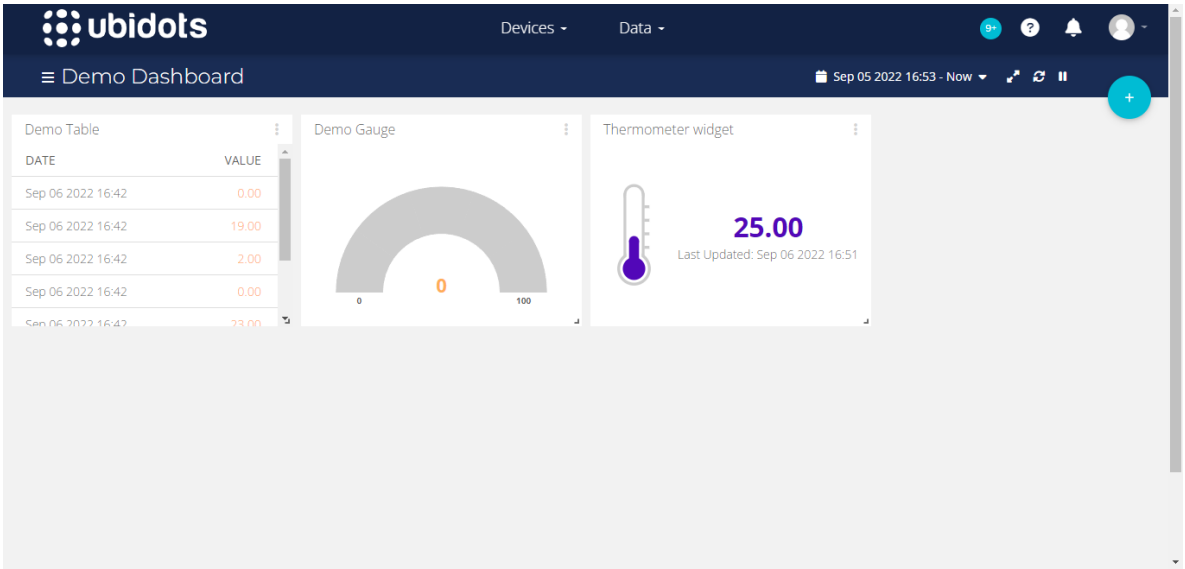
Params • Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	ooo	Bulk Edit
☑	token	BBFF-baawaxKZeBQm3NHX7k0IlKxBDtWgVc			
	Key	Value	Description		

Response

Click Send to get a response



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
}
```

```
  "id": 14
}
dato a enviar: {"temperatura":28.89999962,"humedad":70,"Ph":10}
201
{
  "temperatura": 28.89999962,
  "humedad": 70,
  "Ph": 10,
  "id": 15
}
dato a enviar: {"temperatura":null,"humedad":null,"Ph":10}
```

```
PS C:\Users\User\Desktop\restt> npx json-server --host 192.[redacted].3 --watch datos.json
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

  \{^_^}/ hi!

  Loading datos.json
  Done

  Resources
  http://192.[redacted].3:3000/datos
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

  Home
  http://192.1[redacted]3:3000

  Type s + enter at any time to create a snapshot of the database
  Watching...

POST /datos/ 201 273.963 ms - 63
POST /datos/ 201 169.283 ms - 63
POST /datos/ 201 89.702 ms - 63
POST /datos/ 201 171.449 ms - 63
POST /datos/ 201 78.454 ms - 67
```

Post de envio de datos con cmd

```
GET    http://192.[redacted]:3000/datos    Send

Params   Authorization   Headers (8)   Body •   Pre-request Script   Tests   Settings   Cookies
none   form-data   x-www-form-urlencoded   raw   binary   GraphQL   JSON   Beautify

1  {
2    "Temperatura":26
3  }

Body   Cookies   Headers (14)   Test Results   200 OK   184 ms   7.9 KB   Save Response
Pretty   Raw   Preview   Visualize   JSON

115        "id": 20
116    },
117    {
118        "temperatura": 28.89999962,
119        "humedad": 65,
120        "Ph": 10,
121        "id": 21
122    },
123    {

Cookies   Capture requests   Bootcamp   Runner   Trash
```

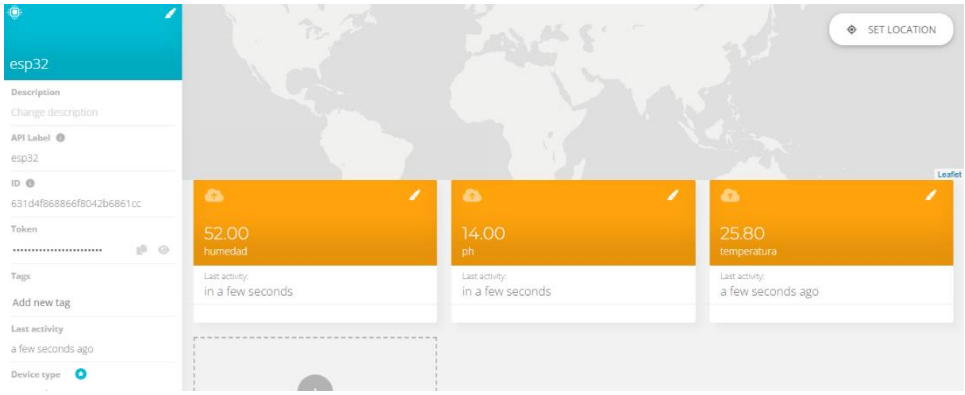
Visualizacion de datos con método get con postman

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
}
```



```
{"humedad":[{"status_code":201}],"ph":[{"status_code":201}],"temperatura":[{"status_code":201}]}
dato a enviar: {"temperatura":25.79999924,"humedad":51,"Ph":14}
[ 88635][E][WiFiClient.cpp:516] flush(): fail on fd 48, errno: 11, "No more processes"
200
{"humedad":[{"status_code":201}],"ph":[{"status_code":201}],"temperatura":[{"status_code":201}]}
dato a enviar: {"temperatura":25.79999924,"humedad":51,"Ph":14}
[ 94778][E][WiFiClient.cpp:516] flush(): fail on fd 48, errno: 11, "No more processes"
200
```

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]