

[cloud](#), [nube](#), [temperatura](#), [esp8266](#)

2 Temperatura y Humedad en la nube

Una vez que hemos visto como montar nuestro propio servidor web, vamos a ver ahora como hacerlo directamente desde un servidor en la nube.

En primer lugar vamos a usar la plataforma <http://dweet.io>, en la cual no hay que registrarse para usarla, basta con crear un nombre de dispositivo único para llamar a la HAPI

```
// wiki.intrusos.info

// Librerias
#include "ESP8266WiFi.h"
#include "DHT.h"

// Parámetros de Conexión a la WiFi
const char* ssid = "miredwifi";
const char* password = "miclave";

// Pin del ESP8266 al que está conectado.
// El GPIO 4 corresponde al D2 del ESP8266-12E NodeMCU v3
#define DHTPIN 4

// tipo de sensor DHT
#define DHTTYPE DHT11 // DHT 11

// Inicializa el sensor
DHT dht(DHTPIN, DHTTYPE);

// Host al que nos vamos a conectar
const char* host = "dweet.io";

void setup() {
  Serial.begin(115200);
  delay(100);

  // Inicializamos el sensor
  dht.begin();

  // Conectamos a la red WIFI
  Serial.println();
  Serial.println();
  Serial.print("Conectando con la red WIFI ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
}
Serial.println("");
Serial.println("Conectado a la WiFi");
Serial.println("Dirección IP: ");
Serial.println(WiFi.localIP());
}

void loop() {
    // Usa la clase WiFiClient para crear una conexión TCP
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("Error de Conexión");
        return;
    }

    // Obtiene la Humedad
    float h = dht.readHumidity();

    // Obtiene la Temperatura en Celsius
    float t = dht.readTemperature();

    delay(3000); //retardo para que no de fallo de lectura del sensor
    // Control de errores, valida que se obtuvieron valores para los datos
    medidos
    if (isnan(h) || isnan(t)) {
        Serial.println("Falla al leer el sensor DHT!");
        client.print("Falla al leer el sensor DHT");
        return;
    }

    // Enviamos los datos al servidor.

    client.print(String("GET
/dweet/for/tunombrededispositivounico?temperatura=") + String(t) +
"&humedad=" + String(h) + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "Connection: close\r\n\r\n");

    // Leemos la respuesta del servidor y la sacamos por el puerto serie
    while (client.available()) {
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }

    Serial.println();
    Serial.println("Cerramos la conexión");

    // Repetir cada 10s
```

```
delay(10000);  
  
}
```

Si toda ha ido correctamente deberías de ver por el puerto serial algo similar a

```
{"this":"succeeded","by":"dweeting","the":"dweet","with":{"thing":"el esp8266 delc","created":"2016-08-31T22:04:33.970Z","content":{"temperatura":27,"humedad":48},"transaction":"dd2d251d-e6ab-4006-9cd3-168fae589158"}}
```

Cerramos la conexión

HTTP/1.1 200 OK

Access-Control-Allow-Origin: *

Content-Type: application/json

Content-Length: 215

Date: Wed, 31 Aug 2016 22:04:49 GMT

Connection: close

Visualizar gráficamente nuestros datos

Como última parte de esta práctica vamos a visualizar gráficamente las medidas de humedad y temperatura usando la plataforma <https://www.freeboard.io>

Referencias

- <https://www.openhomeautomation.net/cloud-temperature-logger-esp8266/>

From:

<http://wiki.intrusos.info/> - LCWIKI

Permanent link:

<http://wiki.intrusos.info/doku.php?id=electronica:esp8266:nube&rev=1472683489>

Last update: **2023/01/18 14:14**

