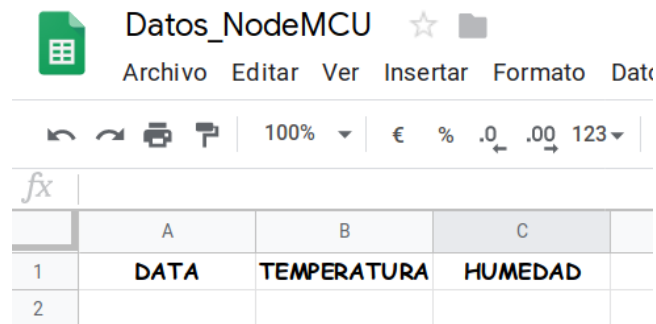


NodeMCU: mando datos de un sensor a Google Hoja de cálculo (IV)

agrportfolioeducativo.blogspot.com/2019/07/nodemcu-mando-datos-de-un-sensor-google.html

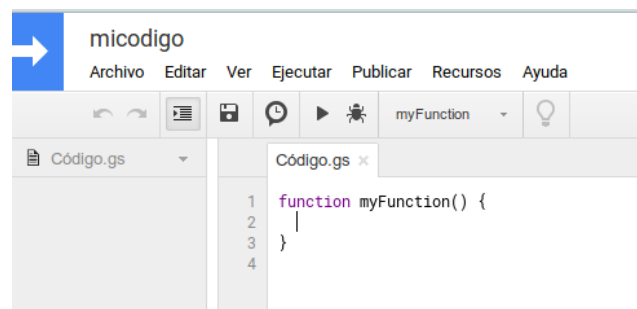


The screenshot shows a Google Sheet interface. At the top, the title 'Datos_NodeMCU' is displayed with a star icon and a folder icon. Below the title is a menu bar with options: Archivo, Editar, Ver, Insertar, Formato, and Datos. A toolbar contains icons for undo, redo, print, and a zoom dropdown set to 100%. There are also currency and decimal formatting icons. The spreadsheet grid has columns labeled A, B, and C. Row 1 contains the headers 'DATA', 'TEMPERATURA', and 'HUMEDAD' respectively. Row 2 is currently empty.

	A	B	C
1	DATA	TEMPERATURA	HUMEDAD
2			

1.- Creo una Hoja de cálculo en mi cuenta de Google.

2.- Una vez creada, iremos a **herramientas >> editor de secuencias de comandos** y crearemos un nuevo código. Lo llamaré **micodigo.gs**



3.- Copiamos el siguiente código...

```
function doGet(e) {  
  Logger.log( JSON.stringify(e) ); // Visualiza parametros  
  
  var result = 'Ok'; // Sucesso  
  if (e.parameter == 'undefined') {  
    result = 'Sem Parâmetros';  
  }  
  else {  
    var sheet_id = ""; // Insira o ID da sua planilha  
    var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); // obtêm a planilha ativa através do ID  
    var newRow = sheet.getLastRow()+1;  
    var rowData = [];  
    rowData[0] = new Date(); // TimeStamp na coluna A  
    for (var param in e.parameter) {  
      Logger.log('In for loop, param=' param);  
      var value = stripQuotes(e.parameter[param]);  
      Logger.log(param + ':'  
  
e.parameter[param]);  
  
// -- Condição para inserir parâmetros na planilha
```

```

switch(param){
case'TEMPERATURA'://Parametro deberá ser o título da columna
    rowData[1]= value;//Inclui valor na columna B
    result ='Dado inserido na columna B';
break;
case'UMIDADE'://Parameter
    rowData[2]= value;//Value in column C
    result +=',Dado inserido na columna C';
break;
default:
    result ="Parâmetro inexistente/não encontrado!";
}
}
    Logger.log(JSON.stringify(rowData));

```

// Escreve nova linha na planilha

```

    var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
    newRange.setValues([rowData]);
}
// Return result of operation
return ContentService.createTextOutput(result);
}
// -- Remove aspas simples e duplas --

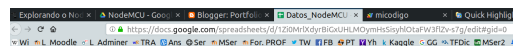
    function stripQuotes( value ) {
    return value.replace(/^["]|[""]$/g, "");
}

```

4.- Buscamos en la hoja de cálculo la ID de la misma, para ponerla en el lugar adecuado del código.

En mi caso es

1ZioMrIXdyrBiGxUHLMOymHsSisyhlOtaFW3fIZv-s7g



5.- Localiza la instrucción del código **var sheet_id = ''**; e introduce la ID anterior.

var sheet_id = '1ZioMrIXdyrBiGxUHLMOymHsSisyhlOtaFW3fIZv-s7g';

6.- Ir a **publicar >> implementar como aplicación web**, y escojo los siguientes parámetros (NOTA: mejor escoger **cualquiera, incluso de forma anónima**)

Pulsamos en implementar, y sale la siguiente información:

Implementar como aplicación web

URL actual de la aplicación web:

[Inhabilitar aplicación web](#)

https://script.google.com/macros/s/AKfycbzxJZ06IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY/exec

Verifica la aplicación web para el último código.

Versión del proyecto:

1

Ejecutar la aplicación como:

Yo (aurelio@seritium.es)

Debes autorizar la secuencia de comandos antes de distribuir la URL.

Quién tiene acceso a la aplicación:

Cualquiera

Actualizar

Cancelar

Ayuda

Implementar como aplicación web

Este proyecto se ha implementado ahora como una aplicación web.

URL actual de la aplicación web:

https://script.google.com/macros/s/AKfycbzxJZ06IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY/exec

Verifica la aplicación web para el último código.

Aceptar

Con la URL actual de la aplicación web:

https://script.google.com/macros/s/AKfycbzxJZ06IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY/exec

7.-Probamos: en la barra de direcciones del navegador escribe

https://script.google.com/macros/s/AKfycbzxJZ06IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY/exec?TEMPERATURA=20&HUMEDAD=60

Y ejecuta. Verás como en la hoja de cálculo aparece una nueva entrada de datos.

== == == ==

La parte de NodeMCU

Ahora toca programar nuestro NodeMCU. Simplemente vamos a enviar datos, de forma fija, cada vez que se ejecute. En otro post haremos las modificaciones para que estos datos sean los de un sensor DHT11 o similar.

8.- Defino bibliotecas

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
```

9.- Defino las constantes de conexión a la Wifi y a Goggle Sheet

```
// -- Variables y constantes --
const char* ssid = "miSSID"; // Rellena con el nombre de tu red WiFi
const char* password = "miCONTRASEÑA"; // Rellena con la contraseña de tu red WiFi

const char* host = "script.google.com"; // Este es el host de los scripts de google.
```

```
const int httpsPort = 443;
```

```
// Huella digital del script de Google:
```

```
const char* fingerprint = "D4 9E 40 F4 53 7A 04 93 38 F7 6B 4B DC 70 02 A9 03 98 C2 DE";  
// const char* fingerprint = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6";
```

```
String googleSheetsID = "AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY"; // El que me  
da al implementar una aplicación web en el script.  
// https://script.google.com/macros/s/AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-  
gUBY/exec?TEMPERATURA=20&HUMEDAD=60
```

Conectamos con el host de los scripts de Google. La huella que a mí me funcionó es la del script una vez lo ejecuto, aunque en las páginas que he visto viene la que está comentada. Funciona con las dos.

La identificación de la hoja de cálculo de Google es la clave que aparece cuando implementamos la aplicación web, la que aparece en la dirección del script.

10.- Defino objetos y otras variables.

```
WiFiClientSecure cliente;
```

```
long previusMillis;  
const long interval = 5000; //cada 5 segundos  
int temperature, humidity;  
float lastH, lastT;
```

Se crea el objeto cliente. Y varias variables.

11.- Función que conecta a la Wifi

```
void connectToWiFi() {  
  Serial.println("Conectando a rede: ");  
  Serial.println(ssid);  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  WiFi.mode(WIFI_STA);  
  Serial.println("");  
  Serial.println("Conectado!");  
  Serial.print("IP: ");  
  Serial.println(WiFi.localIP());  
  delay(1000);  
}
```

12.- Función que lee el sensor

```
void readSensor() {  
  sendDataToGoogleSheets(35,47);  
}
```

Esta función está puenteada. Simplemente manda datos fijos a la función que conecta con Google Sheet. En otro post, esta función leerá el sensor DHT11

13.- Función que envía un dato a Google Sheet

```

void sendDataToGoogleSheets(int temp, int hum) {

    Serial.print("Conectando à: ");
    Serial.println(host);

    cliente.setInsecure();

    if (!cliente.connect(host, httpsPort)) {
        Serial.println("Falla la conexión a Google Sheets -->" String(host) ": "
String(httpsPort));
        return;
    }

    if(cliente.verify(fingerprint, host)){
        Serial.println("Certificado OK");
    }
    else{
        Serial.println("Comprobar certificado");
    }

    String stringTemp = String(temp, DEC);
    String stringHum = String(hum, DEC);
    String url = "/macros/s/" googleSheetsID "/exec?TEMPERATURA=" stringTemp "&HUMEDAD="
stringHum;

    Serial.print("Requisitando URL ");
    Serial.println(url);

    cliente.print(String("GET ") url " HTTP/1.1\r\n"
        "Host: " host "\r\n"
        "User-Agent: BuildFailureDetectorESP8266\r\n"

    "Connection: close\r\n\r\n");

    Serial.println("Request enviada");
    while(cliente.connected()){
        String line = cliente.readStringUntil('\n');
        if(line == "\r"){
            Serial.println("Cabeceras Recibidas");
            Serial.println(line);
        }
        break;
    }
    }

    String line = cliente.readStringUntil('\n');
    if(line.startsWith("{\n\"state\":\n\"success\"}")){
        Serial.println("Éxito");
    }
}

```

```

}else{
  Serial.println("El envío falló!");
}

  Serial.println("Respuesta:");
  Serial.println("=====");
  Serial.println(line);
  Serial.println("=====");
  Serial.println("¡Cerrando la conexión!");
}

```

Esta función tiene varias partes. Por orden...

A.- cliente.setInsecure(); --> La instrucción es importante para poder conectar de forma anónima con la hoja de datos. Si no la escribimos las siguientes líneas darán error.

B.- El siguiente **if** intenta la conexión al servidor... Si tiene éxito.

C.- Con el siguiente **if** se verifica la conexión.

D.- Construye la URL en función de los valores que se pasan a la función de temperatura y humedad.

E.- Se genera una petición tipo GET.

F.- Se reciben las cabeceras de la petición.

G.- Se recibe el contenido (da error porque lo que se recibe es la etiqueta <HTML> y no empieza por {\"state\": \"success\". Sin embargo, el dato se escribe en la hoja de cálculo.

H.- Presenta información en el Monitor serie.

14.- SETUP

```

void setup() {

  Serial.begin(115200);
  connectToWiFi();
}

```

15.- LOOP

```

void loop() {
  if (millis() - previusMillis >= interval) {
    readSensor();
    previusMillis = millis();
  }
}

```

=====

El programa es una modificación de **Yhan Christian Souza Silva**.

```

/*
Envío de datos para Google Sheets
Hardware Utilizado: Nodemcu v1.0, DHT11
Autor: Yhan Christian Souza Silva - data: 27/07/2018
Referências: https://youtu.be/fSoGeaOkNRw
Modificación: Aurelio Gallardo. 06/07/2019
*/

```

```

// -- Bibliotecas auxiliares --

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
// #include "DHT.h"

// -- Hardware --

// #define DHTPIN D2
// #define DHTTYPE DHT11

// DHT dht(DHTPIN, DHTTYPE);

// -- Variables y constantes --
constchar* ssid ="miSSID";// Rellena con el nombre de tu red WiFi
constchar* password ="miCONTRASEÑA";// Rellena con la contraseña de tu red WiFi

constchar* host ="script.google.com";// Este es el host de los scripts de google.
constint httpsPort =443;

// Huella digital del script de Google:
// D4:9E:40:F4:53:7A:04:93:38:F7:6B:4B:DC:70:02:A9:03:98:C2:DE
constchar* fingerprint ="D4 9E 40 F4 53 7A 04 93 38 F7 6B 4B DC 70 02 A9 03 98 C2 DE";

// const char* fingerprint = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6";

String googleSheetsID ="AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-gUBY";// El que me da
al implementar una aplicación web en el script.
// https://script.google.com/macros/s/AKfycbzxJZo6IyF1XR8qjloDEoFE7_OBLpRdDJaSngisNotRSZS-
gUBY/exec?TEMPERATURA=20&HUMEDAD=60

WiFiClientSecure cliente;

long previusMillis;
constlong interval =5000;//cada 5 segundos
int temperature, humidity;
float lastH, lastT;

// -- Setup --

void setup(){
// dht.begin();
Serial.begin(115200);
connectToWiFi();
}

// -- LOOP: lectura del sensor y envío de datos según el intervalo--

void loop(){
if(millis()- previusMillis >= interval){
readSensor();
previusMillis = millis();
}
}

// -- Funciones auxiliares --

// -- Conectando a la red Wifi. Muestra la IP recibida --

```

```

void connectToWiFi(){
  Serial.println("Conectando a rede: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  WiFi.mode(WIFI_STA);
  Serial.println("");
  Serial.println("Conectado!");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());
  delay(1000);
}

/* Lectura del sensor de temperatura y humedad
*/

/* void readSensor() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.print("Temperatura: ");
    Serial.print(lastH);
    Serial.print(" Umidade: ");
    Serial.println(lastT);
    temperature = (int) lastT;
    humidity = (int) lastH;
    sendDataToGoogleSheets(lastT, lastH);
  } else {
    Serial.print("Temperatura: ");
    Serial.print(t);
    Serial.print(" Umidade: ");
    Serial.println(h);
    temperature = (int) t;
    humidity = (int) h;
    sendDataToGoogleSheets(t, h);
    lastH = h;
    lastT = t;
  }
} */

void readSensor(){
  sendDataToGoogleSheets(35,47);
}

/* Función de conexión. Importante la instrucción cliente.setInsecure(); para conectar de forma anónima
*/

void sendDataToGoogleSheets(int temp, int hum){

  Serial.print("Conectando a: ");
  Serial.println(host);

```



```

cliente.setInsecure();

if (!cliente.connect(host, httpsPort)) {
    Serial.println("Falla la conexión a Google Sheets --> " String(host) ": "
String(httpsPort));
return;
}

if(cliente.verify(fingerprint, host)){
    Serial.println("Certificado OK");
}
else{
    Serial.println("Comprobar certificado");
}

    String stringTemp = String(temp, DEC);
    String stringHum = String(hum, DEC);
    String url = "/macros/s/" googleSheetsID "/exec?TEMPERATURA=" stringTemp "&HUMEDAD="
stringHum;

Serial.print("Petición URL ");
Serial.println(url);

cliente.print(String("GET ") url " HTTP/1.1\r\n"
    "Host: " host "\r\n"
    "User-Agent: BuildFailureDetectorESP8266\r\n"

"Connection: close\r\n\r\n");

    Serial.println("Request enviada");
while(cliente.connected()){
    String line = cliente.readStringUntil('\n');
    if(line == "\r"){
        Serial.println("Cabeceras Recibidas");
        Serial.println(line);
    }
    break;
}

String line = cliente.readStringUntil('\n');
if(line.startsWith("{\"state\":\"success\"})){
    Serial.println("Éxito");
}else{
    Serial.println("El envío falló!");
}

    Serial.println("Respuesta:");
Serial.println("=====");
Serial.println(line);
Serial.println("=====");

```

```
Serial.println("¡Cerrando la conexión!");  
}
```

= = = = =

Webs de referencia:

<https://engenheirocaicara.com/explorando-o-nodemcu-envio-de-dados-para-google-sheets/>

<https://thekurks.net/blog/2018/3/20/nodemcu-data-logger>

https://www.hackster.io/nishant_sahay7/sensor-data-upload-to-google-sheets-through-nodemcu-632358