NodeMCU: obteniendo datos desde una dirección HTTP con un JSON (II)

agrportfolioeducativo.blogspot.com/2019/07/nodemcu-obteniendo-datos-desde-una.html

Información en: https://circuits4you.com/2019/01/11/nodemcu-esp8266-arduino-json-parsing-example/

El siguiente código intentará obtener los resultados de los datos obtenidos online de un fichero JSON. Para ello, usaré dos bibliotecas:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
```

La primera permitirá la conexión de mi NodeMCU a internet. La segunda permitirá establecer conexiones HTTP como cliente.

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

Las siguientes dos líneas contienen la SSID o nombre de la conexión WIFI de mi casa o lugar de trabajo (palabra genérica) y la contraseña para acceder a la misma (palabra genérica).

= = = = = = =

SETUP()

El setup son las órdenes que se ejecutan al principio de arrancar el dispositivo

```
void setup()
{
 Serial.begin(115200);
 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED)
 {
  delay(1000);
  Serial.println("Connecting...");
 }
}
```

Empieza una conexión serie a 115200 baudios, típica de la conexión con este dispositivo

(supongo que tiene que concordar con el parámetro **Upload speed** que se encuentra en el menú **Herramientas del IDE**).

Se conecta a la Wifi con ese ssid y esa contraseña. Posteriormente entra en un bucle; si compreba que no está conectado, cada segundo, imprime "Conectando..."

Toda la información se visualiza en el monitor serie de Arduino.

= = = = = = =

LOOP()

Órdenes que se ejecutan constantemente en bucle...

```
void loop()
{
 if (WiFi.status() == WL_CONNECTED)
 {
  HTTPClient http; //Object of class HTTPClient
  http.begin("http://jsonplaceholder.typicode.com/users/2");
  int httpCode = http.GET();

  if (httpCode > 0)
  {

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

   Serial.print("Response Code:"); //200 is OK
   Serial.println(httpCode);   //Print HTTP return code

   Serial.print("Returned data from Server:");
   Serial.println(payload);    //Print request response payload

  }
  http.end(); //Close connection
 }
 delay(30000);
}
```

Una vez comprobado el status de conectado, crea un objeto http de la clase HTTPClient. Conecta con la dirección http://jsonplaceholder.typicode.com/users/2 que contiene información JSON, y en la variable hhtpCode almacena el resultado de hacer una petición tipo GET.

Si la respuesta es un código mayor que cero (debe ser 200 para OK), se obtiene la cadena de respuesta en la variable payload con el método **getString()** aplicado al objeto **http**.

http.end() destruye el objeto.

El proceso vuelve a empezar a los 30 segundos.

NOTA: si intentas conectar a una dirección HTTPS te dará una respuesta errónea, -1.