

SERVIDOR WEB ACCEDIDO MEDIANTE HIPERVÍNCULOS

MATERIALES NECESARIOS:

Tarjeta Wemos D1
Cable USB a micro.
Compilador de Arduino

INDICACIONES: Compilar el siguiente sketch en Arduino IDE y grabarlo en Wemos D1

```
//-----
```

```
// Una vez que se grabe en Wemos es posible que sea necesario pulsar botón RESET
```

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

```
const char* ssid = "MEGACABLE-F79F";  
const char* password = "x363RtMd";
```

```
int ledPin = LED_BUILTIN;  
WiFiServer server(80);
```

```
void setup() {  
  Serial.begin(115200);  
  delay(10);
```

```
  pinMode(ledPin, OUTPUT);  
  digitalWrite(ledPin, LOW);
```

```
  // Conectar a la red Wi fi  
  Serial.println();  
  Serial.println();  
  Serial.print("Conectando a ");  
  Serial.println(ssid);
```

```
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.println("");  
  Serial.println("WiFi conectada");
```

```
  // Start the server  
  server.begin();  
  Serial.println("Servidor iniciado ");
```

```

// Print the IP address
Serial.print("Use esta URL : ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");

}
// -----
void loop() {
    // Verifica si un cliente se ha conectado
    WiFiClient client = server.available();
    if (!client) {
        return;
    }

    // Espera hasta que el cliente envíe algún dato
    Serial.println("nuevo cliente ");
    while(!client.available()){
        delay(1);
    }

    // Lee la primera línea de respuesta
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush(); //Espera hasta que se hayan enviado todos
    // los caracteres salientes del buffer.

    // Acopla la respuesta

    int value = LOW;
    if (request.indexOf("/LED=apagado") != -1) { // El LED de la Wemos opera con
        digitalWrite(ledPin, HIGH);           // lógica invertida
        value = HIGH;
    }
    if (request.indexOf("/LED=encendido") != -1){
        digitalWrite(ledPin, LOW);
        value = LOW;
    }

    // Devuelve la respuesta
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println(""); // No olvidar esta
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");

    client.print("Led pin ahora en: ");

```

```

if(value == HIGH) {
  client.print("APAGADO"); // El LED de la Wemos opera con
  }                      // lógica invertida
else {
  client.print("ENCENDIDO");
  }
  client.println("<br><br>");
  // Lógica negativa en LED de Wemos D1
  client.println("Click <a href=\"/LED=apagado\">aqui</a> Apaga el LED de la Wemos D1 <br>");
  client.println("Click <a href=\"/LED=encendido\">aqui</a> Enciende el LED de la Wemos D1<br>");
  client.println("</html>");

  delay(1);
  Serial.println("Cliente desconectado");
  Serial.println("");

}
//-----

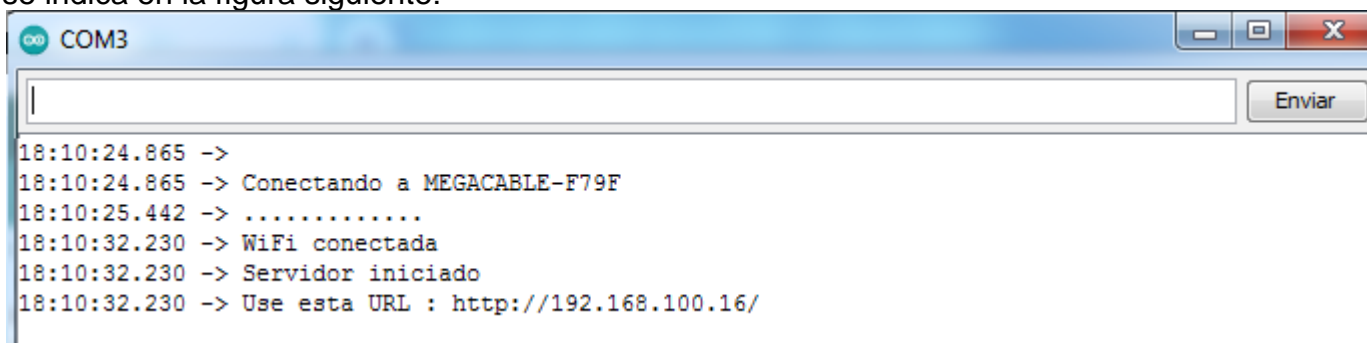
```

CONEXIONES:

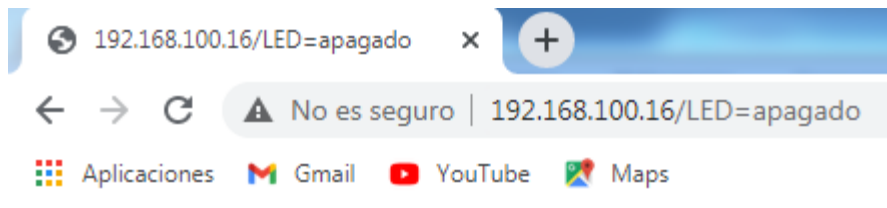
No es necesario realizar conexiones en los GPIO.

OBSERVACIONES PRÁCTICAS:

En el IDE de Arduino seleccionar herramientas y luego Monitor serie, debe aparecer similar a lo que se indica en la figura siguiente:



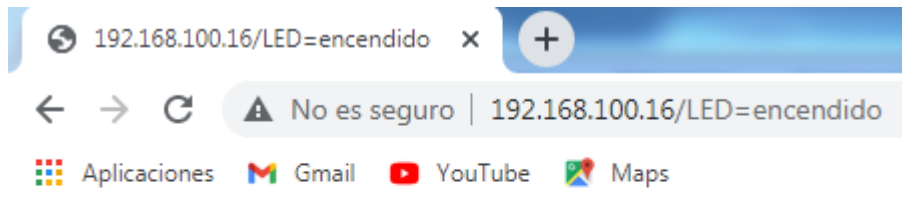
Abrir el navegador y pulsar la URL que se indica en el monitor serial. Pulsar en los hipervínculos y comprobar que se puede controlar el encendido y apagado del LED construido en la placa Wemos D1:



Led pin ahora en: APAGADO

Click [aqui](#) Apaga el LED de la Wemos D1

Click [aqui](#) Enciende el LED de la Wemos D1



Led pin ahora en: ENCENDIDO

Click [aqui](#) Apaga el LED de la Wemos D1

Click [aqui](#) Enciende el LED de la Wemos D1

A medida que se pulsán las opciones en los hipervínculos del servidor web, en el monitor serial aparece lo siguiente:

```
21:27:59.797 -> nuevo cliente
21:27:59.797 -> GET /LED=apagado HTTP/1.1
21:27:59.797 -> Cliente desconectado
21:27:59.797 ->
21:28:01.970 -> nuevo cliente
21:28:01.970 -> GET /LED=apagado HTTP/1.1
21:28:01.970 -> Cliente desconectado
21:28:01.970 ->
21:28:02.037 -> nuevo cliente
21:28:02.037 -> GET /favicon.ico HTTP/1.1
21:28:02.037 -> Cliente desconectado
21:28:02.037 ->
21:28:02.772 -> nuevo cliente
21:28:03.302 -> GET /LED=encendido HTTP/1.1
21:28:03.302 -> Cliente desconectado
... ..
```