

SERVIDOR WEB CON HIPERVÍNCULOS ASIGNANDO IP ESTÁTICA

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

//Este ejemplo usa IP estática

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

```
const char* ssid = "MEGACABLE-F79F";  
const char* password = "6edrW2323";
```

```
int ledPin = LED_BUILTIN;  
WiFiServer server(80);  
IPAddress ip(192, 168, 100, 15); // Es la IP fijada  
IPAddress gateway(192, 168, 100, 1); // establece la puerta de enlace  
//predeterminada de la red personal
```

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

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

```
  Serial.print(F("Estableciendo la IP estática a : "));  
  Serial.println(ip);
```

```
  // Connect to WiFi network  
  Serial.println();  
  Serial.println();  
  Serial.print("Conectando a ");  
  Serial.println(ssid);  
  IPAddress subnet(255, 255, 255, 0); // establece la máscara de subred personal  
  WiFi.config(ip, gateway, subnet);  
  WiFi.begin(ssid, password);
```

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

```

// Start the server
server.begin();
Serial.println("Server started");

// Print the IP address
Serial.print("Use this 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\">para</a> Apagar el LED de la Wemos D1 <br>");
client.println("Click <a href=\"/LED=encendido\">para</a> Encender el LED de la Wemos D1<br>");
client.println("</html>");

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

```

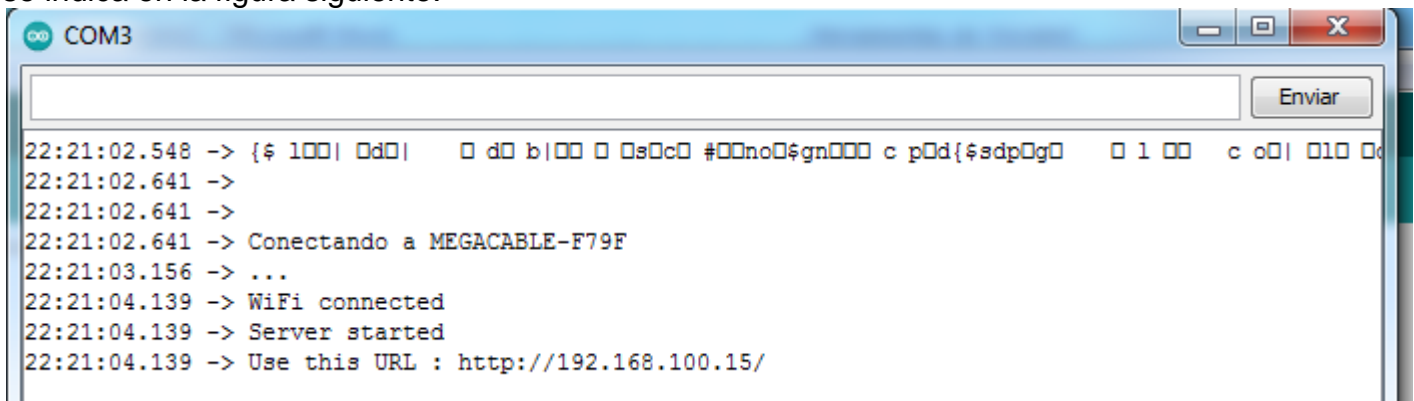
//-----

CONEXIONES:

No es necesario realizar conexiones en los GPIO.

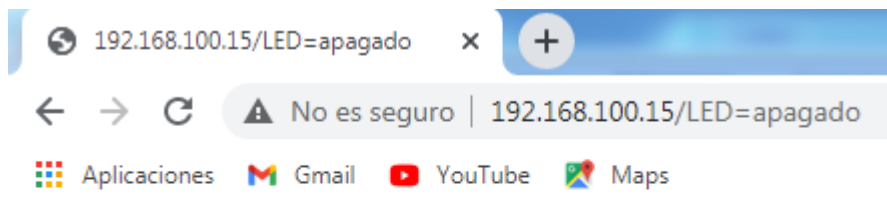
OBSERVACIONES PRÁCTICAS:

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



Se observa que se conectó con la IP que se le indicó mediante programación.

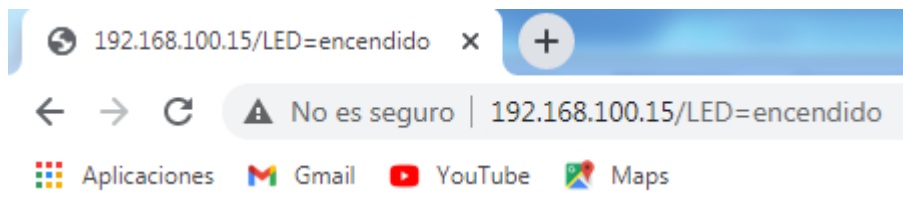
Abrir el navegador y pulsar la URL que se indica en el monitor serial. Pulsar en los hipervínculos y comprobar que se puede controlar el encendido y apagado del LED construido en la placa Wemos D1:



Led pin ahora en: APAGADO

Click [para](#) Apagar el LED de la Wemos D1

Click [para](#) Encender el LED de la Wemos D1



Led pin ahora en: ENCENDIDO

Click [para](#) Apagar el LED de la Wemos D1

Click [para](#) Encender 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:

```
22:34:49.405 ->
22:34:49.460 -> nuevo cliente
22:34:49.460 -> GET /favicon.ico HTTP/1.1
22:34:49.460 -> Cliente desconectado
22:34:49.460 ->
22:34:52.430 -> nuevo cliente
22:34:52.430 -> GET /LED=encendido HTTP/1.1
22:34:52.430 -> Cliente desconectado
22:34:52.430 ->
22:34:52.476 -> nuevo cliente
22:34:52.476 -> GET /favicon.ico HTTP/1.1
22:34:52.476 -> Cliente desconectado
22:34:52.476 ->
22:34:54.332 -> nuevo cliente
22:34:54.713 -> GET /LED=apagado HTTP/1.1
22:34:54.713 -> Cliente desconectado
```