

# PRACTICA 3 : WIFI y BLUETOOTH

---

## A: GENERACIÓN DE UNA PÁGINA WEB

En esta práctica, deseamos realizar la conexión de nuestra placa ESP 32 a una wifi existente y posteriormente generar una página Web.

### Código

```
#include <WiFi.h>
#include <WebServer.h>

// SSID & Password
const char* ssid = "VI-203-101-001-003"; // Enter your SSID here
const char* password = "Eo0Ep3ii"; //Enter your Password here

WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

void handle_root(void);

void setup() {
  Serial.begin(115200);
  Serial.println("Try Connecting to ");
  Serial.println(ssid);

  // Connect to your wi-fi modem
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected successfully");
  Serial.print("Got IP: ");
  Serial.println(WiFi.localIP()); //Show ESP32 IP on serial

  server.on("/", handle_root);

  server.begin();
  Serial.println("HTTP server started");
  delay(100);
}

void loop() {
  server.handleClient();
}

// HTML & CSS contents which display on web server
```

```
String HTML = "<!DOCTYPE html>\n
<html>\n
<body>\n
<h1>My Página WEB con ESP32 &#128522;</h1>\n
</body>\n
</html>";

// Handle root url (/)
void handle_root() {
  server.send(200, "text/html", HTML);
}
```

## Funcionamiento del programa

En primer lugar, deberemos declarar dos librerías que son necesarias para nuestra práctica. En primer lugar, necesitamos la librería "WiFi.h" para conectar nuestra placa ESP32 a una red WiFi. En segundo lugar, también necesitamos la librería "WebServer.h" para generar un servidor web. Ahora vamos a declarar las variables globales que contienen el nombre y contraseña de la red wifi a la que nos vamos a conectar.

```
#include <WiFi.h>
#include <WebServer.h>

// SSID & Password
const char* ssid = "VI-203-101-001-003"; // Enter your SSID here
const char* password = "Eo0Ep3ii"; //Enter your Password here

WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

void handle_root(void);
```

Ahora encontramos la función Setup que tiene como objetivo iniciar la conexión WiFi, una vez se haya realizado nos lo mostrará por pantalla, posteriormente ejecuta la función "handle\_root" y crea el servidor web en la IP indicada.

```
void setup() {
  Serial.begin(115200);
  Serial.println("Try Connecting to ");
  Serial.println(ssid);

  // Connect to your wi-fi modem
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
}
```

```
Serial.println("WiFi connected successfully");
Serial.print("Got IP: ");
Serial.println(WiFi.localIP()); //Show ESP32 IP on serial

server.on("/", handle_root);

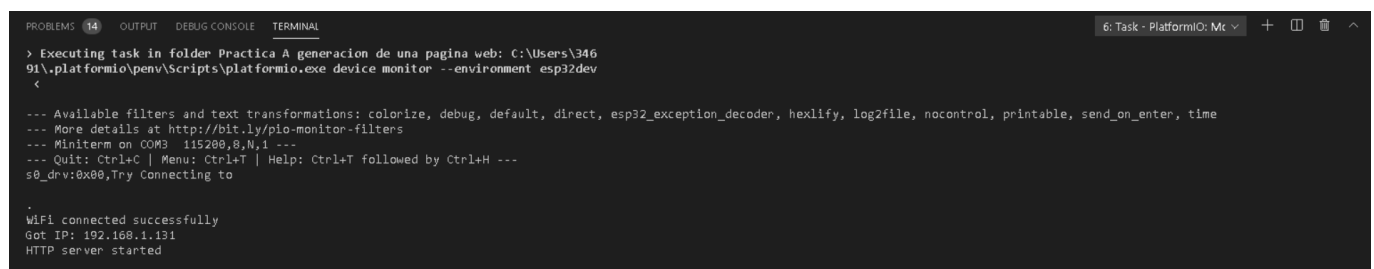
server.begin();
Serial.println("HTTP server started");
delay(100);
}
```

Finalmente crearemos una variable tipo string donde guardaremos el texto que deseamos mostrar en el Servidor Web.

```
// HTML & CSS contents which display on web server
String HTML = "<!DOCTYPE html>\n
<html>\n
<body>\n
<h1>My Página WEB con ESP32 &#128522;</h1>\n
</body>\n
</html>";

// Handle root url (/)
void handle_root() {
  server.send(200, "text/html", HTML);
}
```

## Salida por el Terminal



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL 6: Task - PlatformIO: Mc + [] ☒ ^
> Executing task in folder Practica A generacion de una pagina web: C:\Users\346
91\platformio\penv\Scripts\platformio.exe device monitor --environment esp32dev
<

--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at http://bit.ly/pio-monitor-filters
--- Miniterm on COM3 115200,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
s0_drv:0x00, Try Connecting to

WiFi connected successfully
Got IP: 192.168.1.131
HTTP server started
```