void loop(){

WiFiClient client = server.available();

if (client) { // If a new client connects,

String currentLine = ""; // make a String to hold incoming data

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

header += c;

if (c == '\n') {

if (currentLine.length() == 0) {

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

if (header.indexOf("GET /led/on") >= 0) {

ledState = "on";

digitalWrite(led, HIGH);

Serial.println("led on");

}

if (header.indexOf("GET /led/off") >= 0)

{

ledState = "off";

digitalWrite(led, LOW);

Serial.println("led off");

}

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("**<head>**<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #555555;}</style>**</head>**");

// Web Page Heading

client.println("<body><h1>ESP32 Web Server</h1>");

// Display current led state

client.println("<p>State " + ledState + "</p>");

if (ledState=="off")

client.println("<p><a href=\"/led/on\"><button class=\"button\">ON</button></a></p>");

if (ledState=="on")

client.println("<p><a href=\"/led/off\"><button class=\"button button2\">OFF</button></a></p>");

client.println("</body></html>");

client.println();

break;

} else {

currentLine = "";

}

} else if (c != '\r') {

currentLine += c;

}

}

}

header = "";

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

CON IP FIJA

#include <WiFi.h>

const char\* ssid = "CLARO\_60FA";

const char\* password = "d2dACF";

IPAddress local\_IP(192, 168, 1, 184);

IPAddress gateway(192, 168, 1, 1);

IPAddress subnet(255, 255, 0, 0);

IPAddress primaryDNS(8, 8, 8, 8); //optional

IPAddress secondaryDNS(8, 8, 4, 4); //optional

WiFiServer server(80);

String header;

String ledState = "off";

const int led = 23;

void setup() {

Serial.begin(115200);

pinMode(led, OUTPUT);

digitalWrite(led, LOW);

Serial.print("Connecting");

// Configures static IP address

if (!WiFi.config(local\_IP, gateway, subnet, primaryDNS, secondaryDNS)) {

Serial.println("STA Failed to configure");

}

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

WiFiClient client = server.available(); // Listen for incoming clients

if (client) { // If a new client connects,

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

header += c;

if (c == '\n') {

if (currentLine.length() == 0) {

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

if (header.indexOf("GET /led/on") >= 0) {

ledState = "on";

digitalWrite(led, HIGH);

Serial.println("led on");

}

if (header.indexOf("GET /led/off") >= 0)

{

ledState = "off";

digitalWrite(led, LOW);

Serial.println("led off");

}

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #555555;}</style></head>");

// Web Page Heading

client.println("<body><h1>ESP32 Web Server</h1>");

// Display current led state

client.println("<p>State " + ledState + "</p>");

if (ledState=="off")

client.println("<p><a href=\"/led/on\"><button class=\"button\">ON</button></a></p>");

if (ledState=="on")

client.println("<p><a href=\"/led/off\"><button class=\"button button2\">OFF</button></a></p>");

client.println("</body></html>");

client.println();

break;

} else {

currentLine = "";

}

} else if (c != '\r') {

currentLine += c;

}

}

}

header = "";

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}

//= = = = =

Incluyee Bluettoth para conocer la IP

#include <WiFi.h>

#include "BluetoothSerial.h"

BluetoothSerial BT;

const char\* ssid = "CLARO\_608FFA";

const char\* password = "d2dACCD34F";

WiFiServer server(80);

String header;

String ledState = "off";

const int led = 23;

void setup() {

Serial.begin(115200);

pinMode(led, OUTPUT);

digitalWrite(led, LOW);

BT.begin("ESP32"); //Name of your Bluetooth to pair

Serial.print("Connecting");

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

}

void loop(){

//Check if we receive anything from Bluetooth

if (BT.available())   
  {

int income = BT.read();

Serial.println(income); // 0-> 48 1-> 49

if (income ==49) BT.println(WiFi.localIP());

}

//WIFI

WiFiClient client = server.available(); // Listen for incoming clients

if (client) { // If a new client connects,

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

header += c;

if (c == '\n') {

if (currentLine.length() == 0) {

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html");

client.println("Connection: close");

client.println();

if (header.indexOf("GET /led/on") >= 0) {

ledState = "on";

digitalWrite(led, HIGH);

Serial.println("led on");

}

if (header.indexOf("GET /led/off") >= 0)

{

ledState = "off";

digitalWrite(led, LOW);

Serial.println("led off");

}

// Display the HTML web page

client.println("<!DOCTYPE html><html>");

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #555555;}</style></head>");

// Web Page Heading

client.println("<body><h1>ESP32 Web Server</h1>");

// Display current led state

client.println("<p>State " + ledState + "</p>");

if (ledState=="off")

client.println("<p><a href=\"/led/on\"><button class=\"button\">ON</button></a></p>");

if (ledState=="on")

client.println("<p><a href=\"/led/off\"><button class=\"button button2\">OFF</button></a></p>");

client.println("</body></html>");

client.println();

break;

} else {

currentLine = "";

}

} else if (c != '\r') {

currentLine += c;

}

}

}

header = "";

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}