

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

// --- Configura tu WiFi ---
const char* ssid = "Pacha";
const char* password =
"mechupas";

// --- LED ---
const int ledPin = 2;

// --- Servidor ---
WebServer server(80);

// --- Página HTML ---
String htmlPage =
R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Control LED ESP32</tit
le>
  <style>
    body { font-family: Arial;
text-align: center; background-
color: #f2f2f2; }
    h1 { color: #333; }
    .btn {
display: inline-block;
padding: 15px 25px;
font-size: 20px;
text-decoration: none;
color: white;
border-radius: 8px;
margin: 10px;
}
    .on { background-color: gre
en; }
    .off { background-color: re
d; }
  </style>
</head>
<body>
  <h1>Control del LED ESP32</h1
>
  <a class="btn on" href="/pren
der">Prender LED</a>
  <a class="btn off" href="/apa
gar">Apagar LED</a>
</body>
</html>
)"rawliteral";

// --- Manejo de rutas ---
void handleRoot() {
  server.send(200,
"text/html", htmlPage);
}

void handlePrender() {
  digitalWrite(ledPin, HIGH);
  server.send(200,
"text/html", htmlPage +
"<p>LED Encendido</p>" );
}

void handleApagar() {
  digitalWrite(ledPin, LOW);
  server.send(200,
"text/html", htmlPage +
"<p>LED Apagado</p>" );
}

void setup() {
  Serial.begin(115200);

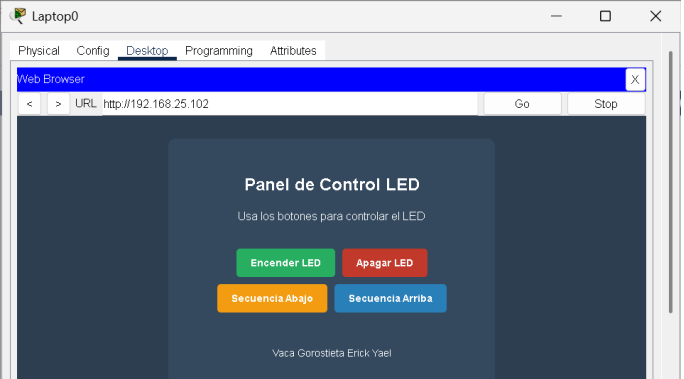
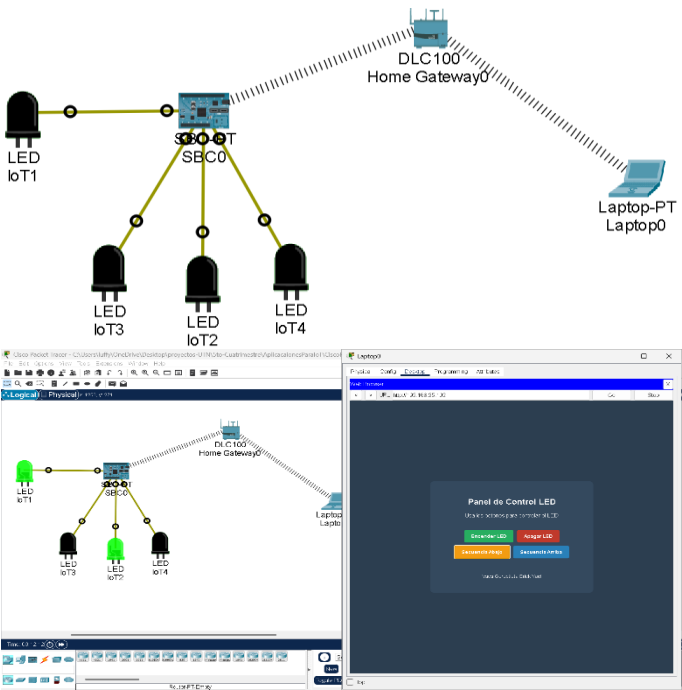
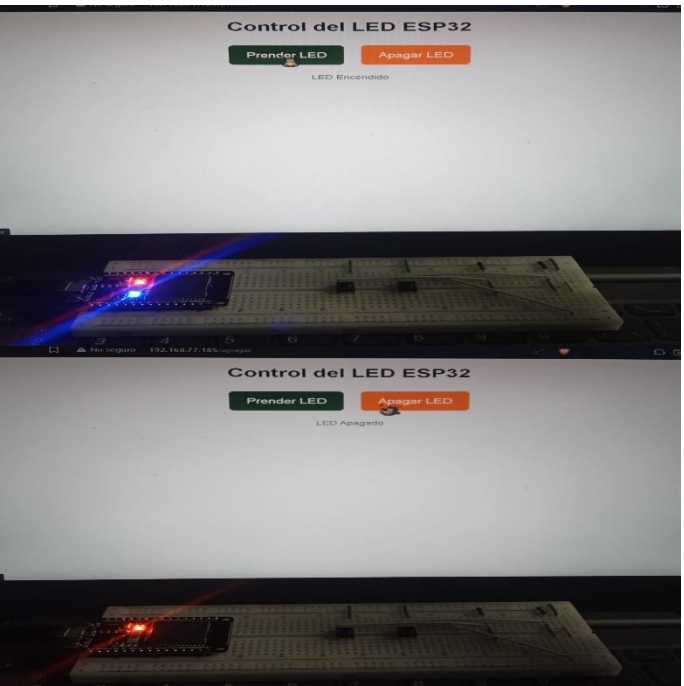
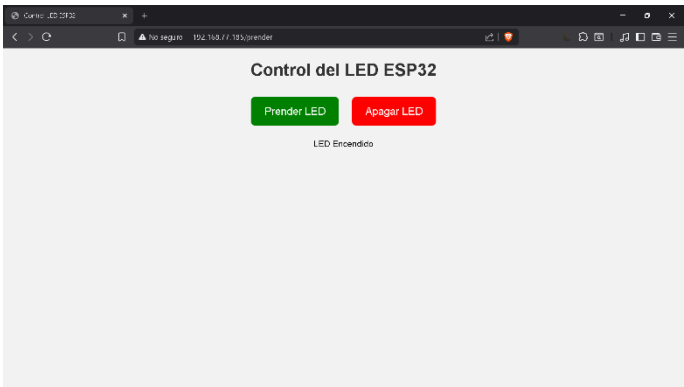
  // Configurar pin LED
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Conexión WiFi
  WiFi.begin(ssid, password);
  Serial.print(
"Conectando a WiFi..." );
  while (WiFi.status() !=
WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\n
WiFi conectado");
  Serial.print(
"IP del ESP32: " );
  Serial.println(WiFi.localIP
());

  // Rutas
  server.on("/", handleRoot);
  server.on("/prender",
handlePrender);
  server.on("/apagar",
handleApagar);

  // Iniciar servidor
  server.begin();
  Serial.println(
"Servidor HTTP iniciado" );
}

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



```
function setup() {
  // Página HTML con botones para el LED

  var htmlPage = `<DOCTYPE html>
  \n' +
  '<html lang="es"> \n' +
  '<head> \n' +
  '  <meta charset="UTF-8"> \n' +
  '  <title>Control LED</title> \n' +
  '  <style> \n' +
  '    body { \n' +
  '      font-family: Arial, sans-serif;
  \n' +
  '        background: #2c3e50; \n' +
  '
  '        display: flex; \n' +
  '        justify-content: center;
  \n' +
  '
  '        align-items: center; \n' +
  '
  '        height: 100vh; \n' +
  '        color: white; \n' +
  '        margin: 0; \n' +
  '      } \n' +
  '      .caja { \n' +
  '        display: flex; \n' +
  '        flex-direction: column;
  \n' +
  '        align-items: center; \n' +
  '
  '        width: 90%; \n' +
  '        max-width: 400px; \n' +
  '        padding: 30px; \n' +
  '        background: #34495e; \n' +
  '
  '        border-radius: 10px; \n' +
  '      } \n' +
  '      .titulo { \n' +
  '        margin-bottom: 20px; \n' +
  '
  '        text-align: center; \n' +
  '      } \n' +
  '      .botones { \n' +
  '        display: flex; \n' +
  '        flex-wrap: wrap; \n' +
  '        gap: 10px; \n' +
  '        margin-bottom: 20px; \n' +
  '
  '        justify-content: center;
  \n' +
  '
  '      } \n' +
  '      button { \n' +
  '        padding: 12px 20px; \n' +
  '
  '        border: none; \n' +
  '        border-radius: 5px; \n' +
  '
  '        cursor: pointer; \n' +
  '        font-size: 14px; \n' +
  '        font-weight: bold; \n' +
  '        min-width: 120px; \n' +
  '      } \n' +
  '      #encender { background: #27ae60; color: white; }
  \n' +
  '      #apagar { background: #c0392b; color: white; }
  \n' +
  '      #arriba { background: #2980b9; color: white; }
  \n' +
  '      #abajo { background: #f39c12; color: white; }
  \n' +
  '      .alumno { \n' +
  '        margin-top: 15px; \n' +
  '        font-size: 14px; \n' +
  '      } \n' +
  '    </style> \n' +
  '  </head> \n' +
  '  <body> \n' +
  '    <div class="caja"> \n' +
  '      <section class="titulo"> \n' +
  '
  '        <h1>Panel de Control LED
  </h1>
  \n' +
  '
  '        <p>Usa los botones para controlar el LED</p>
  \n' +
  '
  '      </section> \n' +
  '      <section class="botones"> \n' +
  '
  '        <button id="encender">Encender LED</button>
  \n' +
  '
  '        <button id="apagar">Apagar LED</button>
  \n' +
  '
  '        <button id="abajo">Secuencia Abajo</button>
  \n' +
  '
  '        <button id="arriba">Secuencia Arriba</button>
  \n' +
  '
  '      </section> \n' +
  '      <section class="alumno"> \n' +
  '
  '        <p>Vaca Gorostieta Erick
  `;

```

```
Yael</p>
  \n' +
  '
  </section> \n' +
  '
  </div> \n' +
  '
  <script> \n' +
  '
  document.getElementById( 'encender').addEventListener( 'click', () => { \n' +
  '
  fetch( ' /prender' )
  .then(r => alert("LED Encendido"));
  \n' +
  '
  }); \n' +
  '
  document.getElementById( 'apagar').addEventListener( 'click', () => { \n' +
  '
  fetch( ' /apagar' )
  .then(r => alert("LED Apagado"));
  \n' +
  '
  }); \n' +
  '
  document.getElementById( 'arriba').addEventListener( 'click', () => { \n' +
  '
  fetch( ' /arriba' )
  .then(r => alert("Secuencia Arriba"));
  \n' +
  '
  }); \n' +
  '
  document.getElementById( 'abajo').addEventListener( 'click', () => { \n' +
  '
  fetch( ' /abajo' )
  .then(r => alert("Secuencia Abajo"));
  \n' +
  '
  }); \n' +
  '
  </script> \n' +
  '
  </body> \n' +
  '
  </html>';

  // Rutas del servidor
  HTTPServer.route("/", function (url, res) {
    res.setContentType( "text/html");
    res.send( htmlPage );
  });

  HTTPServer.route("/personal", function (url, res) {
    res.send( "Bienvenido a mi web personal" );
  });

  HTTPServer.route("/prender", function (url, res) {
    digitalWrite(0, HIGH);
    // pin D0 ON
    res.send( "LED Encendido" );
  });

  HTTPServer.route("/apagar", function (url, res) {
    digitalWrite(0, LOW);
    // pin D0 OFF
    res.send( "LED Apagado" );
  });

  // Ruta para la secuencia "arriba" (1 -> 2 -> 3)
  HTTPServer.route("/arriba", function (url, res) {
    digitalWrite(1, HIGH);
    delay(500);
    digitalWrite(1, LOW);
    digitalWrite(2, HIGH);
    delay(500);
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    delay(500);
    digitalWrite(3, LOW);
    res.send("Secuencia Arriba");
  });

  // Ruta para la secuencia "abajo" (3 -> 2 -> 1)
  HTTPServer.route("/abajo", function (url, res) {
    digitalWrite(3, HIGH);
    delay(500);
    digitalWrite(3, LOW);
    digitalWrite(2, HIGH);
    delay(500);
    digitalWrite(2, LOW);
    digitalWrite(1, HIGH);
    delay(500);
    digitalWrite(1, LOW);
    res.send("Secuencia Abajo");
  });

  // Ruta por defecto
  HTTPServer.route("/", function (url, res) {
    res.send( "404: Recurso no encontrado" );
  });

  // Inicializa servidor
  HTTPServer.start(80);
  Serial.println( "Servidor HTTP IoT iniciado en puerto 80" );
}

```

