

Manual de Usuario – Mano Robótica Controlada por WiFi Wemos D1 mini (ESP8266)

1. Descripción General

Este sistema permite controlar una mano robótica con servomotores mediante una interfaz web inalámbrica, sin necesidad de conexión a Internet. El microcontrolador ESP8266 crea su propia red WiFi y un servidor web local al que el usuario puede acceder desde cualquier dispositivo (celular, notebook, tablet) para manipular los movimientos de los dedos.

2. Componentes Principales

Componente	Descripción
ESP8266 (NodeMCU o Wemos D1 Mini)	Controlador principal con WiFi integrado.
5 Servomotores	Controlan los dedos Pulgar, Índice, Medio, Anular y Meñique.
LED WiFi (D4)	Indica actividad del sistema (parpadea cada segundo).
Botón de Reset (D3)	Reinicia el sistema.
Fuente de alimentación 5V	Proporciona energía estable a los servos y al ESP8266.

3. Funcionamiento General

1. El ESP8266 crea una red WiFi local llamada 'ManoRobotica' con la contraseña '12345678'.
2. El usuario se conecta a esa red WiFi desde su dispositivo.
3. En el navegador, ingresar: <http://192.168.0.1>
4. Se abrirá una página web con sliders y botones interactivos para controlar los dedos y ejecutar gestos.

4. Interfaz Web

La interfaz cuenta con tres secciones: Movimiento fino (control individual de dedos), Acciones rápidas (gestos predefinidos), y Juguemos (formas Piedra, Papel o Tijera).

5. Detalles Técnicos

Función	Descripción
Modo WiFi	Access Point (AP) sin router externo.
IP fija	192.168.0.1
Puerto HTTP	80
Bibliotecas	ESP8266WiFi.h, ESP8266WebServer.h, Servo.h, ESP8266mDNS.h
Pins de servos	D0 (Pulgar), D8 (Índice), D5 (Medio), D6 (Anular), D7 (Meñique)
LED estado	D4 (GPIO2) – parpadea cada segundo
Botón Reset	D3 (GPIO0)

6. Encendido y Conexión

1. Conecte la alimentación de 5V a los servos y al ESP8266.
2. Espere que el LED parpadee.
3. Conéctese a la red WiFi 'ManoRobotica'.
4. Ingrese la contraseña '12345678'.
5. En el navegador, escriba `http://192.168.0.1` para acceder al panel.

7. Uso Básico

Use los sliders para mover los dedos individualmente. Los botones permiten ejecutar gestos rápidos, abrir la mano, realizar una demo o jugar Piedra, Papel o Tijera.

8. Recomendaciones

- No alimente los servos desde el pin 5V del ESP8266.
- Use una fuente externa de al menos 2A con masa común (GND).
- Evite mover los servos manualmente. • Si la web no carga, revise la conexión WiFi.

9. Posibles Mejoras

- Integración con WiFiManager.

- Control por voz o sensores.
- Adaptación a ESP32.

10. Créditos

Proyecto: Control de Mano Robótica con ESP8266 Wemos D1 Mini

Integrantes: Guadalupe Rivas, Sandra Barragan, Silvia Loro, Miguel Sanchez, Gonzalo Robles

Versión: 1.0

Lenguaje: C++ (Arduino IDE / PlatformIO).

11. Código.

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Servo.h>
#include <ESP8266mDNS.h>

#define RESET_PIN D3 // Botón físico (GPIO0)

#define LED_WIFI D4 // LED interno (GPIO2, activo en LOW)

Servo servoPulgar, servoIndice, servoMedio, servoAnular, servoMenique;

ESP8266WebServer server(80);

// Posiciones de cada dedo

int posPulgar = 0, posIndice = 0, posMedio = 0, posAnular = 0, posMenique = 0;

// ----- FUNCIONES DE SERVOS -----

void moverServo(int dedo, int valor) {

  switch (dedo) {

    case 1: servoPulgar.write(valor); break;

    case 2: servoIndice.write(180 - valor); break;

    case 3: servoMedio.write(180 - valor); break;

    case 4: servoAnular.write(valor); break;

    case 5: servoMenique.write(180 - valor); break;
```

```
}}  
void moverMano(int dedos[5]) {  
  
    moverServo(1, dedos[0]);  
  
    moverServo(2, dedos[1]);  
  
    moverServo(3, dedos[2]);  
  
    moverServo(4, dedos[3]);  
  
    moverServo(5, dedos[4]);  
  
    posPulgar = dedos[0];  
    posIndice = dedos[1];  
    posMedio = dedos[2];  
    posAnular = dedos[3];  
    posMenique = dedos[4];  
  
}  
  
// Formas rápidas  
  
void formaPiedra() {  
  
    int d[5] = { 180, 180, 180, 180, 180 };  
  
    moverMano(d);  
  
}  
  
void formaPapel() {  
  
    int d[5] = { 0, 0, 0, 0, 0 };  
  
    moverMano(d);  
  
}  
  
void formaTijera() {  
  
    int d[5] = { 180, 0, 0, 180, 180 };  
  
    moverMano(d);  
  
}  
  
void formaAleatoria() {
```

```
int n = random(3);

if (n == 0) {

    formaPiedra();

    Serial.println("□ Piedra");

}

if (n == 1) {

    formaPapel();

    Serial.println("□ Papel");

}

if (n == 2) {

    formaTijera();

    Serial.println("× Tijera");

}

}

// ----- PAGINA WEB RESPONSIVA -----

void handleRoot() {

    String html = R"rawliteral(

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Control Mano Robótica</title>

<style>

body { font-family: Arial; text-align:center; background:#f2f2f2; margin:0; padding:0;}

header { padding:10px; background:#2196F3; color:white; }

.container { max-width:600px; margin:20px auto; padding:20px; background:white; border-
radius:10px; box-shadow:0 0 10px #999; }
```

```
.slider { width: 100%; margin:10px 0; }

button { margin:5px; padding:12px 20px; font-size:16px; border:none; border-radius:8px;
cursor:pointer; }

.btn-group { display:flex; flex-wrap:wrap; justify-content:center; gap:10px; margin-
bottom:15px; }

.btn-group button { flex:1 1 100px; max-width:150px; }

.btn-accion{ background:#2196F3;color:white; }

.btn-abrir{ background:#4CAF50;color:white; }

.btn-juego{ background:#9C27B0;color:white; }

img.logo{ width:80%; max-width:300px; display:block; margin:0 auto 15px;}

@media(max-width:400px){

  .container { padding:15px; margin:10px; }

  button { font-size:14px; padding:8px 12px; }

  img.logo { width:70%; }

}

</style>

</head>

<body>

<header>

<h2>Control Mano Robótica</h2>

</header>

<div class="container">

<h3>Movimiento fino</h3>

<p>Pulgar: <input type="range" min="0" max="180" value="%P%" class="slider"
oninput="fetch('/servo?d=1&v='+this.value)"></p>

<p>Índice: <input type="range" min="0" max="180" value="%I%" class="slider"
oninput="fetch('/servo?d=2&v='+this.value)"></p>

<p>Medio: <input type="range" min="0" max="180" value="%M%" class="slider"
oninput="fetch('/servo?d=3&v='+this.value)"></p>
```

```
<p>Anular: <input type="range" min="0" max="180" value="%A%" class="slider"
oninput="fetch('/servo?d=4&v='+this.value)"></p>
```

```
<p>Meñique: <input type="range" min="0" max="180" value="%E%" class="slider"
oninput="fetch('/servo?d=5&v='+this.value)"></p>
```

```
<h3>Acciones rápidas</h3>
```

```
<div class="btn-group">
```

```
<button class="btn-accion" onclick="fetch('/cuenta?n=1')">1</button>
```

```
<button class="btn-accion" onclick="fetch('/cuenta?n=2')">2</button>
```

```
<button class="btn-accion" onclick="fetch('/cuenta?n=3')">3</button>
```

```
<button class="btn-accion" onclick="fetch('/cuenta?n=4')">4</button>
```

```
<button class="btn-accion" onclick="fetch('/cuenta?n=5')">5</button>
```

```
<button class="btn-abrir" onclick="fetch('/abrir')">Abrir Mano</button>
```

```
<button class="btn-abrir" onclick="demo()">Demo</button>
```

```
</div>
```

```
<h3>Juguemos!</h3>
```

```
<div class="btn-group">
```

```
<button class="btn-juego" onclick="fetch('/forma?t=piedra')">🔲 Piedra</button>
```

```
<button class="btn-juego" onclick="fetch('/forma?t=papel')">🔳 Papel</button>
```

```
<button class="btn-juego" onclick="fetch('/forma?t=tijera')">✂ Tijeras</button>
```

```
<button class="btn-juego" onclick="fetch('/aleatorio')">🎲 Aleatorio</button>
```

```
</div>
```

```
<script>
```

```
function demo(){
```

```
  let i=1;
```

```

function go(){

    fetch('/cuenta?n='+i);

    i++;

    if(i<=5) setTimeout(go,1000);

    else setTimeout(=>fetch('/abrir'),1000);

}

go();

}

</script>

</body>

</html>

)rawliteral";

    html.replace("%P%", String(posPulgar));

    html.replace("%I%", String(posIndice));

    html.replace("%M%", String(posMedio));

    html.replace("%A%", String(posAnular));

    html.replace("%E%", String(posMenique));

    server.send(200, "text/html", html);

}

// ----- HANDLERS -----

void handleServo() {

    int d = server.arg("d").toInt();

    int v = server.arg("v").toInt();

    moverServo(d, v);

    if (d == 1) posPulgar = v;

    if (d == 2) posIndice = v;

```



```
if (d == 3) posMedio = v;
if (d == 4) posAnular = v;
if (d == 5) posMenique = v;
server.send(200, "text/plain", "OK");
}

void handleCuenta() {
    int n = server.arg("n").toInt();
    int dedos[5] = { 0, 0, 0, 0, 0 };
    for (int i = 0; i < n; i++) dedos[i] = 180;
    moverMano(dedos);
    server.send(200, "text/plain", "Cuenta " + String(n));
}

void handleAbrir() {
    int d[5] = { 0, 0, 0, 0, 0 };
    moverMano(d);
    server.send(200, "text/plain", "Mano abierta");
}

void handleForma() {
    String tipo = server.arg("t");
    if (tipo == "piedra") formaPiedra();
    if (tipo == "papel") formaPapel();
    if (tipo == "tijera") formaTijera();
    server.send(200, "text/plain", "Forma: " + tipo);
}

void handleAleatorio() {
    formaAleatoria();
    server.send(200, "text/plain", "Aleatorio");
}
```

```
}

// ----- SETUP -----

void setup() {

  Serial.begin(115200);

  pinMode(RESET_PIN, INPUT_PULLUP);

  pinMode(LED_WIFI, OUTPUT);

  digitalWrite(LED_WIFI, HIGH);

  randomSeed(analogRead(A0));

  //  Modo AP con IP fija

  IPAddress local_ip(192, 168, 0, 1);

  IPAddress gateway(192, 168, 0, 1);

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

  WiFi.mode(WIFI_AP);

  WiFi.softAPConfig(local_ip, gateway, subnet);

  WiFi.softAP("ManoRobotica", "12345678");

  Serial.println("Red local creada: ManoRobotica");

  Serial.print("IP: ");

  Serial.println(WiFi.softAPIP());

  // Servos

  servoPulgar.attach(D0);

  servoIndice.attach(D8);

  servoMedio.attach(D5);

  servoAnular.attach(D6);
```

```
servoMenique.attach(D7);

// Handlers

server.on("/", handleRoot);

server.on("/servo", handleServo);

server.on("/cuenta", handleCuenta);

server.on("/abrir", handleAbrir);

server.on("/forma", handleForma);

server.on("/aleatorio", handleAleatorio);

server.begin();

Serial.println("🌐 Servidor web iniciado (192.168.0.1)");
}

// ----- LOOP -----

void loop() {

  server.handleClient();

  // LED parpadeante (estado activo)

  static unsigned long t0 = 0;

  if (millis() - t0 > 1000) {

    digitalWrite(LED_WIFI, !digitalRead(LED_WIFI));

    t0 = millis();

  }

}
```