

**Trabajo 11 NodeMCU**

**Sección: D03**

**Clave: I5909**

**Profesor: Lopez Franco, Michel Emmanuel**

**Christopher Antonio Velasco Gutierrez**

**Código: 218867133**

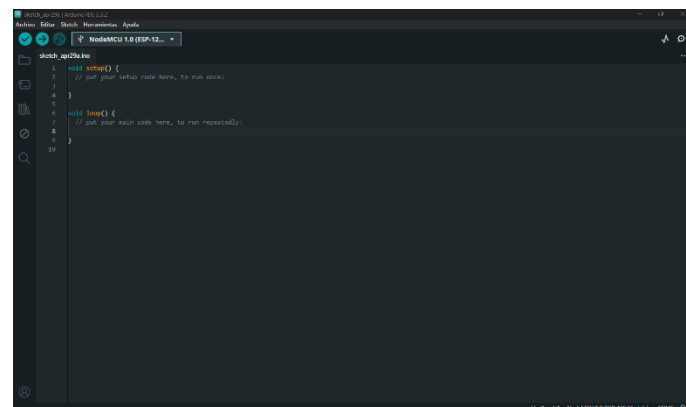
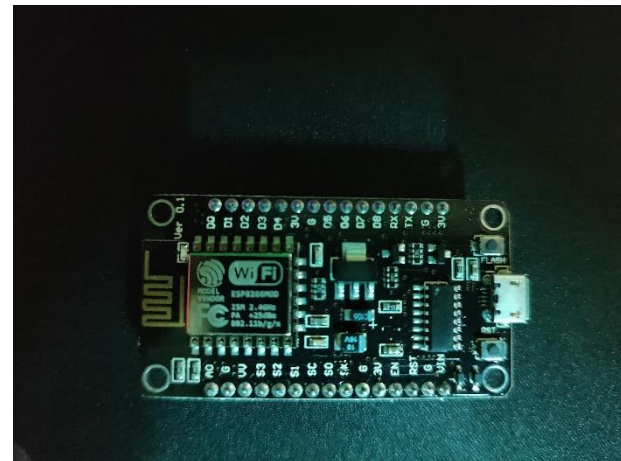


## Introducción

En la presente era digital, la interacción entre dispositivos inalámbricos y la Internet de las Cosas (IoT) está transformando el enfoque tradicional de los sistemas electrónicos. Un ejemplo claro de esta evolución se observa en el uso de módulos WiFi para controlar dispositivos a través de interfaces web. Este reporte describe el desarrollo de una aplicación basada en un módulo WiFi, específicamente diseñada para controlar el estado (encendido y apagado) de un LED integrado. Mediante la implementación de un servidor web dentro del módulo, los usuarios pueden interactuar directamente con el hardware mediante una página web sencilla y funcional. Esta solución no solo demuestra la capacidad de control remoto vía inalámbrica, sino que también sirve como una excelente herramienta pedagógica y prototipo para futuras expansiones en proyectos relacionados con la automatización del hogar y sistemas interactivos.

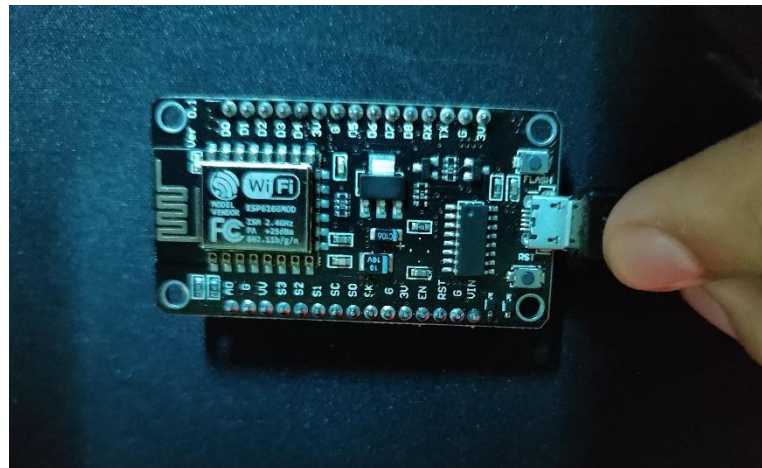
En el presente proyecto, se ha seleccionado el módulo WiFi NodeMCU ESP8266E V1.0 como el componente central. Este módulo ha sido elegido por su robustez, flexibilidad y bajo costo, lo que lo hace ideal para aplicaciones de IoT a pequeña escala. El NodeMCU ESP8266E V1.0 integra funcionalidades de WiFi que permiten desarrollar aplicaciones inalámbricas eficientemente. Además, su plataforma es compatible con diversos entornos de programación, facilitando la implementación de un servidor web que maneje las funcionalidades del LED integrado.

Para la programación del NodeMCU ESP8266E V1.0, se utilizará el Arduino IDE, un entorno de desarrollo ampliamente reconocido por su simplicidad y eficacia en la programación de microcontroladores. Este software facilita la escritura, compilación y carga de código a



diversos módulos y placas. Para la compatibilidad específica con nuestro módulo WiFi, he incorporado el paquete para ESP8266, disponible en el repositorio oficial en GitHub. Este paquete añade soporte para el ESP8266 al Arduino IDE, permitiendo aprovechar todas las funcionalidades del módulo mediante una interfaz familiar y accesible.

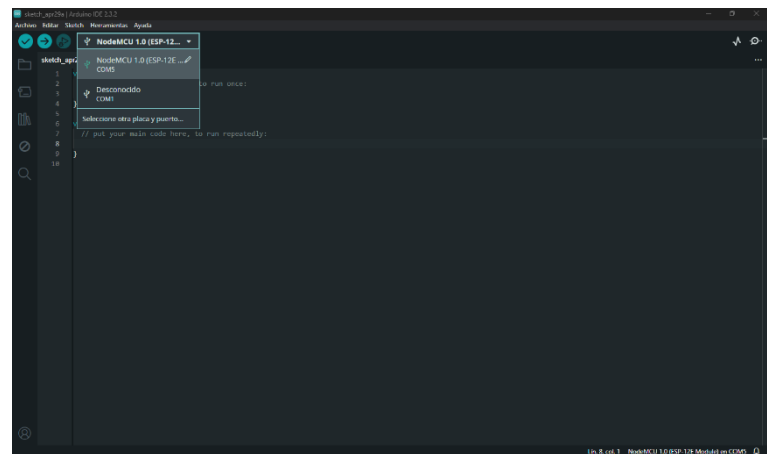
Inicialmente, para comenzar la configuración del módulo WiFi, es necesario conectar el NodeMCU ESP8266E V1.0 a la computadora utilizando su puerto USB integrado. Este paso es fundamental para establecer una comunicación entre el módulo y el Arduino IDE, permitiendo la transferencia de código y la depuración del mismo.



Asegúrese de que la conexión sea segura para evitar interrupciones durante el proceso de programación.

Una vez que el PC haya reconocido el módulo WiFi NodeMCU ESP8266E V1.0, el

siguiente paso es configurar adecuadamente el Arduino IDE para asegurar la correcta programación del dispositivo. Para ello, se debe seleccionar el modelo exacto de la placa en el menú de herramientas bajo la opción “Placa:” y elegir “NodeMCU 1.0 (ESP-12E Module)”. Posteriormente, es necesario asignar el



puerto correcto al cual está conectado el módulo, en este caso “COM5”. Además, ajustaremos el tamaño del flash en “4M (1M SPIFFS)” y la velocidad de subida a 115200 bps en las configuraciones adicionales. Estos ajustes son cruciales para garantizar una comunicación fluida y sin errores durante la carga del código al módulo.

El código proporcionado gestiona un servidor web en el módulo WiFi NodeMCU ESP8266E V1.0 para controlar un LED incorporado. Comienza con la inclusión de la librería ESP8266WiFi, necesaria para manejar las funcionalidades WiFi del módulo.

### Configuración Inicial:

- Se configura el PIN donde está conectado el LED como salida y se inicializa apagado.
- Se establece una conexión WiFi utilizando el SSID y contraseña especificados.
- Tras conectar exitosamente a la red WiFi, se inicia el servidor en el puerto 80 y se imprime la dirección IP local para acceder al servidor.

### Bucle Principal (loop):

- El servidor espera conexiones entrantes. Al recibir una, verifica si hay datos disponibles del cliente.
- Lee la petición del cliente hasta que encuentra un retorno de carro (\r), y procesa la petición:
  - Si la petición contiene /LED=ON, se cambia el estado del LED a apagado (LOW).
  - Si contiene /LED=OFF, se cambia el estado del LED a encendido (HIGH).
- Según el estado del LED, se actualiza el pin correspondiente.

```
WIFI_LED_CONTROL.ino
1  #include <ESP8266WiFi.h>
2
3  const char* ssid = "MEGACABLE-37F0";
4  const char* password = " ";
5
6  WiFiServer server(80);
7
8  int PinLED = LED_BUILTIN;
9  int estado = LOW;
10
11 void setup() {
12     Serial.begin(115200);
13
14     pinMode(PinLED, OUTPUT);
15     digitalWrite(PinLED, LOW);
16
17     WiFi.begin(ssid, password);
18     Serial.printf("\n\nConectando a la red: %s\n", WiFi.SSID().c_str());
19     while (WiFi.status() != WL_CONNECTED) {
20         delay(500);
21         Serial.print(".");
22     }
23     Serial.println("");
24     Serial.println("WiFi conectada");
25
26     server.begin();
27     Serial.println("Servidor inicializado");
28
29     Serial.printf("\n\nUtiliza esta URL para conectar: http://%s/\n", WiFi.localIP().toString().c_str());
30 }
31
32 void loop()
33 {
34
35     WiFiClient client = server.available();
36     if (!client) {
37         return;
38     }
39
40     Serial.println("nuevo cliente");
41     while(!client.available()){
42         delay(1);
43     }
44
45     String peticion = client.readStringUntil('\r');
46     Serial.println(peticion);
47     client.flush();
48
49     if (peticion.indexOf('/LED=ON') != -1)
50     {estado = LOW;}
51     if (peticion.indexOf('/LED=OFF') != -1)
52     {estado = HIGH;}
53
54     digitalWrite(PinLED, estado);
55
56     client.println("HTTP/1.1 200 OK");
57     client.println("Content-Type: text/html");
58     client.println("Connection: close");
59     client.println();
60
61     client.println("<!DOCTYPE HTML>");
62     client.println("<html>");
63     client.println("<head>");
64     client.println("<title>Control Remoto LED</title>");
65     client.println("<style>");
66
67     Salida Monitor Serie
```

[illegible]

Con un título y dos botones en forma de triángulo, se muestra el botón de encendido.

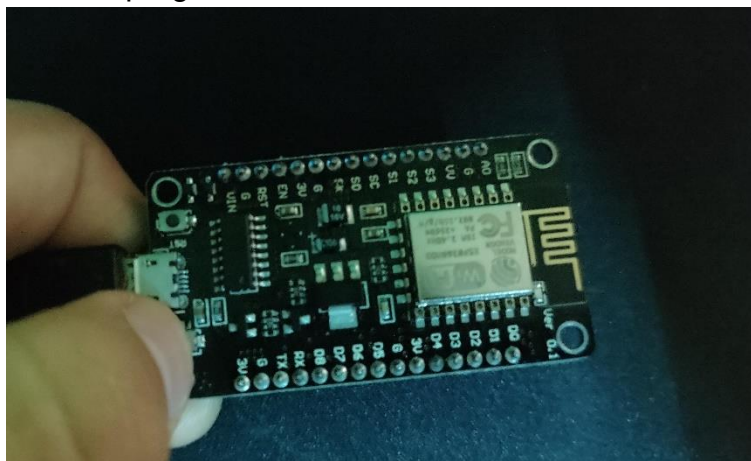
- Acceso remoto con el hardware conectado al sistema
- Interfaz de usuario sencilla y efectiva
- Navegador web conectado a la misma red

The screenshot shows the Arduino IDE interface with the 'NodeMCU 1.0 (ESP-12...)' board selected. The 'Serial Monitor' window is open, displaying the boot sequence and the successful completion of the firmware upload. The progress bar at the top indicates that the upload is 100% complete.

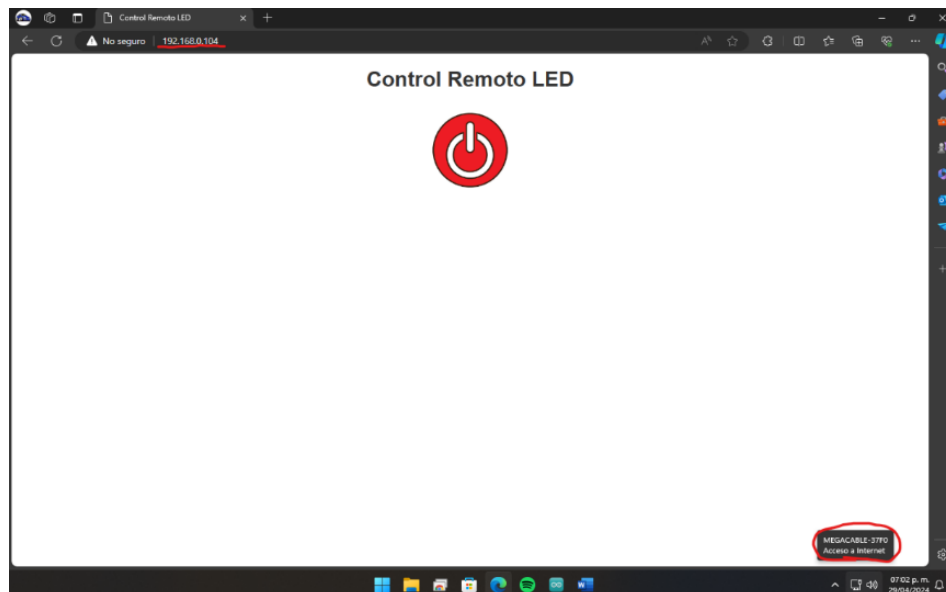
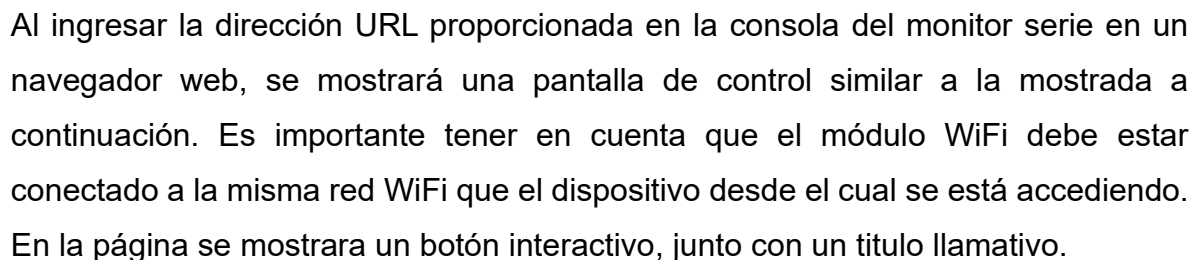
```

code in flash (default, ICACHE_FLASH_ATTR), used 240220 / 1048576 bytes (23%)
|-----| 8115 |-----| DESCRIPTION
|-----| ESP8266 |-----| code in flash
upload.py v3.0
Serial port COM6
Connecting...
chip is ESP8266
Features: WiFi
Crystal is 20MHz
MCU: 4437-S01-Rev3.177
uploading stub...
running stub...
Stub running...
Configuring flash size...
Auto-detected flash size: 0M
Compressed 240220 bytes to 208396...
writing at 0x00000000... (7 %)
writing at 0x00000000... (15 %)
writing at 0x00000000... (21 %)
writing at 0x00000000... (30 %)
writing at 0x00000000... (40 %)
writing at 0x00000000... (46 %)
writing at 0x00000000... (51 %)
writing at 0x00000000... (57 %)
writing at 0x00000000... (63 %)
writing at 0x00000000... (69 %)
writing at 0x00000000... (76 %)
writing at 0x00000000... (84 %)
writing at 0x00000000... (92 %)
writing at 0x00000000... (100 %)
write 240220 bytes (208396 compressed) at 0x00000000 in 59.9 seconds (effective 114.0 kbit/s)...
hash of data verified.

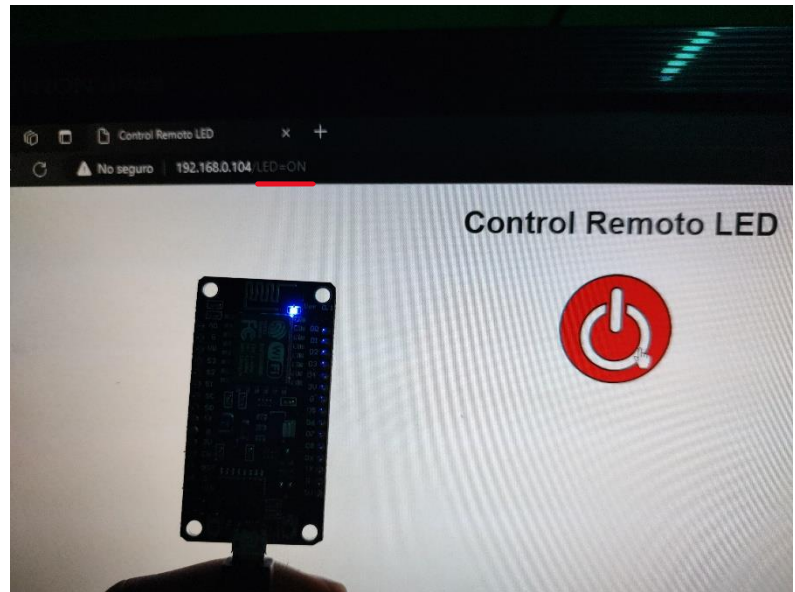
leaving...
Hard resetting via RTS pin...
  
```



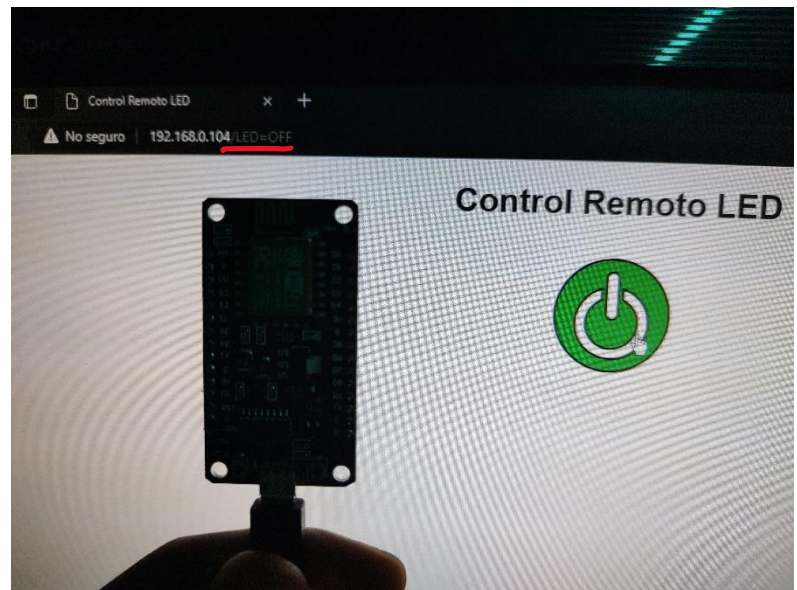




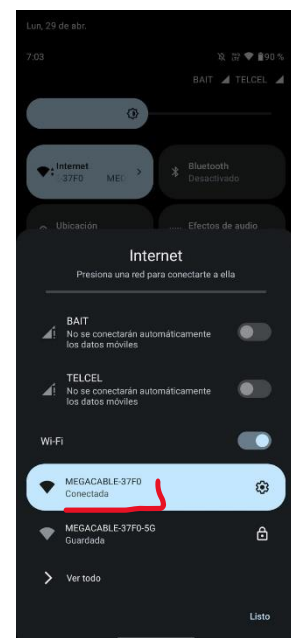
Al presionar el botón interactivo en la página web, el LED integrado en nuestro módulo WiFi se encenderá. El botón se reducirá ligeramente en tamaño, simulando estar presionado, y cambiará su color a rojo. Además, la URL en la barra del navegador se modificará automáticamente a /LED=ON, indicando que se ha enviado una solicitud para encender el LED.



Al presionar nuevamente el botón en la página web, este volverá a su tamaño original, simulando no estar presionado, y cambiará su color a verde. Además, se apagará el LED integrado en nuestro módulo WiFi. La URL en la barra del navegador se actualizará a /LED=OFF, indicando que se ha enviado una solicitud para apagar el LED.



Para realizar la prueba desde un dispositivo móvil, primero nos conectaremos a la misma red WiFi a la que está conectado el módulo WiFi. Esto garantizará que el dispositivo móvil y el módulo WiFi estén en la misma red local, permitiendo una comunicación efectiva entre ellos.



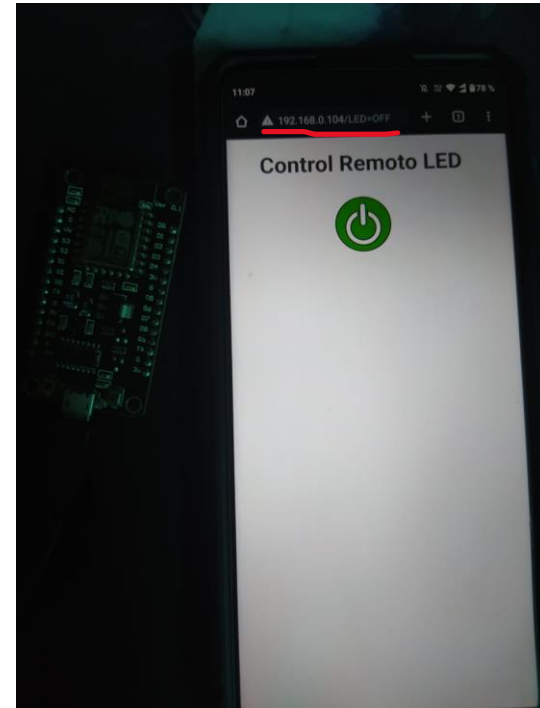


Control Remoto LED

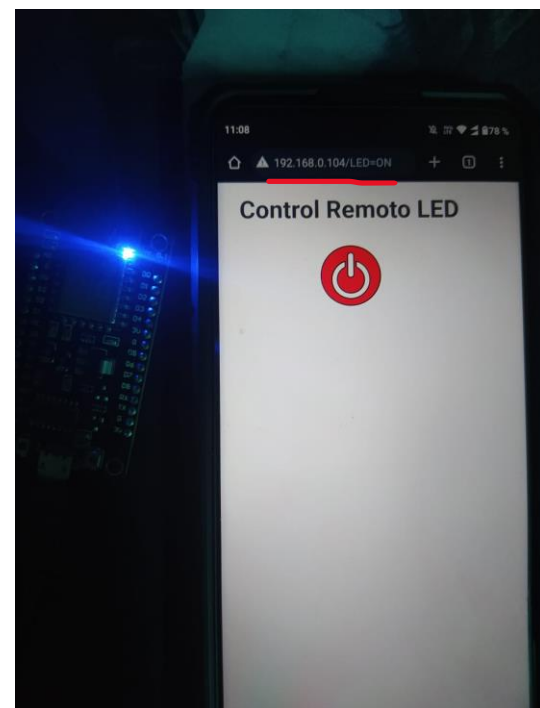


Luego de conectarnos a la misma red WiFi que el módulo WiFi, ingresaremos la dirección URL proporcionada por el monitor serie del Arduino IDE en el navegador web de nuestro dispositivo móvil. Esto nos permitirá acceder a la misma página web de control que vimos anteriormente desde el navegador de la computadora.

Al presionar el botón en la página web desde el dispositivo móvil, este cambiará su apariencia a verde y la URL en la barra del navegador se actualizará automáticamente a /LED=OFF. Además, el LED integrado en nuestro módulo WiFi se apagará, proporcionando un control remoto efectivo del LED desde el dispositivo móvil.



Al presionar el botón en la página web desde el dispositivo móvil, el LED del módulo WiFi se encenderá y el botón cambiará su color a rojo. Además, la URL en la barra del navegador se actualizará automáticamente a /LED=ON. Este proceso confirma el correcto funcionamiento del código, la página web y la finalización exitosa de la actividad planificada.





## **Conclusión**

En este proyecto, hemos desarrollado y probado con éxito un sistema de control remoto para un LED integrado utilizando el módulo WiFi NodeMCU ESP8266E V1.0. Este sistema permite a los usuarios encender y apagar el LED desde cualquier dispositivo conectado a la misma red WiFi que el módulo.

A través del código proporcionado, hemos creado un servidor web que proporciona una interfaz intuitiva y fácil de usar, accesible desde cualquier navegador web. La interacción con la página web se traduce directamente en acciones físicas en el módulo, demostrando así la capacidad de control remoto del hardware a través de Internet.

Este proyecto no solo destaca las capacidades del módulo WiFi NodeMCU ESP8266E V1.0 y su integración con el entorno de desarrollo Arduino IDE, sino que también sirve como una excelente introducción al mundo de la Internet de las Cosas (IoT) y la programación de dispositivos conectados. Además, la modularidad y la versatilidad de esta solución permiten su aplicación en una amplia gama de proyectos futuros relacionados con la automatización del hogar, el monitoreo remoto y la creación de sistemas interactivos.

Para concluir decir que el éxito de este proyecto resalta la importancia y el potencial de la integración de la tecnología WiFi en dispositivos electrónicos, abriendo un mundo de posibilidades para la creación de soluciones innovadoras y conectadas en el ámbito de la electrónica y la informática.