

# **Practica 6: Practica de arduino y shield Ethernet**

**Unidad II: Interfaces de comunicación Tema 2.1**

**Sistemas Embebidos II 18MPEDS0729**

**Ago-Dic 2025**

**Centro de Enseñanza Tecnica Industrial Plantel Colomos**

**Tgo. en Desarrollo de Software**

**Academia: Sistemas Digitales**

**Profesor: Antonio Lozano Gonzáles**

EMMANUEL BUENROSTRO 22300891 7F1

EMILIANO ARZATE 22300929 7F1

19 de Octubre de 2025



## §1 Objetivo

Controlar relevadores por medio de internet, para manipular cualquier tipo de carga y encender y/o apagar aparatos a distancia.

## §2 Desarrollo de la Práctica

### §2.1 Condiciones de la Práctica

Consiste en usar dos relevadores, los cuales serán controlados desde una pagina web. Se deberá utilizar un Arduino y una shield ethernet. Por lo tanto se deberá tener en dicha pagina mínimo dos botones, uno dirá relevador 1 ON/OFF; el otro botón dirá relevador 2 ON/OFF. Se pueden utilizar más botones si así lo desean. O en vez de botones perillas etc. usen su imaginación. Un relevador sera para controlar un foco incandescente y el otro relevador controlara un ventilador

### §2.2 Algoritmo o Diagrama de Flujo

- Inicializa comunicación serie, pines de relés como salida y los pone en estado HIGH (apagados).
- Configura la interfaz Ethernet con dirección MAC e IP y arranca un servidor en el puerto 80.
- En el bucle principal espera clientes entrantes del servidor.
- Cuando llega un cliente, lee la petición HTTP carácter por carácter y la acumula en una cadena.
- Si detecta rutas con parámetros (?rel1=on/off, ?rel2=on/off) actualiza los pines correspondientes y variables de estado.
- Cuando termina de leer la cabecera HTTP responde con una página HTML que muestra botones para encender/apagar según el estado actual.
- Cierra la conexión con el cliente y vuelve a esperar nuevos clientes.

### §2.3 Código C

```
1 #include <SPI.h>
2 #include <Ethernet.h>
3
4 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
5 IPAddress ip(192, 168, 1, 177);
6 EthernetServer server(80);
7
8 const int RELE1_PIN = 7;
9 const int RELE2_PIN = 8;
10
11 boolean rele1_estado = false;
12 boolean rele2_estado = false;
13
14 void setup() {
15   Serial.begin(9600);
16 }
```

```
17 pinMode(RELE1_PIN, OUTPUT);
18 pinMode(RELE2_PIN, OUTPUT);
19
20 digitalWrite(RELE1_PIN, HIGH);
21 digitalWrite(RELE2_PIN, HIGH);
22
23 Ethernet.begin(mac, ip);
24 server.begin();
25
26 Serial.print("Servidor iniciado en la IP: ");
27 Serial.println(Ethernet.localIP());
28 }
29
30 void loop() {
31   EthernetClient client = server.available();
32   if (client) {
33     Serial.println("Nuevo cliente conectado.");
34     boolean currentLineIsBlank = true;
35     String request = "";
36
37     while (client.connected()) {
38       if (client.available()) {
39         char c = client.read();
40         request += c;
41
42         if (c == '\n' && currentLineIsBlank) {
43           client.println("HTTP/1.1 200 OK");
44           client.println("Content-Type: text/html");
45           client.println("Connection: close");
46           client.println();
47
48           client.println("<!DOCTYPE HTML>");
49           client.println("<html>");
50           client.println("<head><title>Control de Relevadores</title>");
51           client.println("<meta name='viewport'");
52             content='width=device-width, initial-scale=1'>");
53           client.println("<style>");
54           client.println("body { font-family: Arial, sans-serif;");
55             text-align: center; background-color: #282c34; color: white;");
56           client.println("h1 { color: #61dafb; }");
57           client.println(".boton { display: block; width: 80%; padding:");
58             20px; margin: 20px auto; font-size: 24px; border: none;");
59           client.println(".on { background-color: #28a745; color: white;");
60             }");
61           client.println(".off { background-color: #dc3545; color: white;");
62             }");
63           client.println("</style></head>");
64           client.println("<body>");
65           client.println("<h1>Control de Dispositivos con Arduino</h1>");
66
67           client.println("<h2>Foco Incandescente</h2>");
68           if (rele1_estado) {
69             client.println("<a href='\"/?rele1=off\"'><button class='boton");
70               off'>APAGAR</button></a>");
71           } else {
```

```

66         client.println("<a href='\"/?rele1=on\"'><button class='boton
           on'>ENCENDER</button></a>");
67     }
68
69     client.println("<h2>Ventilador</h2>");
70     if (rele2_estado) {
71         client.println("<a href='\"/?rele2=off\"'><button class='boton
           off'>APAGAR</button></a>");
72     } else {
73         client.println("<a href='\"/?rele2=on\"'><button class='boton
           on'>ENCENDER</button></a>");
74     }
75
76     client.println("</body></html>");
77     break;
78 }
79 if (c == '\n') {
80     currentLineIsBlank = true;
81 } else if (c != '\r') {
82     currentLineIsBlank = false;
83 }
84
85 if(request.indexOf("GET /?rele1=on") != -1) {
86     Serial.println("Petición para encender Rele 1");
87     digitalWrite(RELE1_PIN, LOW);
88     rele1_estado = true;
89 }
90 if(request.indexOf("GET /?rele1=off") != -1) {
91     Serial.println("Petición para apagar Rele 1");
92     digitalWrite(RELE1_PIN, HIGH);
93     rele1_estado = false;
94 }
95 if(request.indexOf("GET /?rele2=on") != -1) {
96     Serial.println("Petición para encender Rele 2");
97     digitalWrite(RELE2_PIN, LOW);
98     rele2_estado = true;
99 }
100 if(request.indexOf("GET /?rele2=off") != -1) {
101     Serial.println("Petición para apagar Rele 2");
102     digitalWrite(RELE2_PIN, HIGH);
103     rele2_estado = false;
104 }
105 }
106 }
107
108 delay(1);
109
110 client.stop();
111 Serial.println("Cliente desconectado.");
112 }
113 }

```

### §3 Observaciones y Conclusiones

- Usamos por primera vez el arduino como local para poder generar una pagina web.

- Esto le dio una IP al arduino.

## §4 Imagen

