TIC’s

Practica de Laboratorio

Tema 2

Profesor: Azamar Palma Iván

Asignatura: Tópicos de IoT

Alumno: Flores Arroyo Javier de Jesús.

Grupo: 9SP

Contenido

Practica de Laboratorio Tema 2

Instrucciones

TÍTULO: Conexión de Arduino con Shield Ethernet W5100 – Tutorial desde básico hasta avanzado AUTOR: Ivan Azamar Palma

Introducción

Este tutorial describe cómo conectar un Arduino UNO con un Shield Ethernet W5100, realizar ejercicios de nivel básico, medio y avanzado, incluyendo la lectura de un sensor DHT11 y el control de un servo desde una página web HTML.

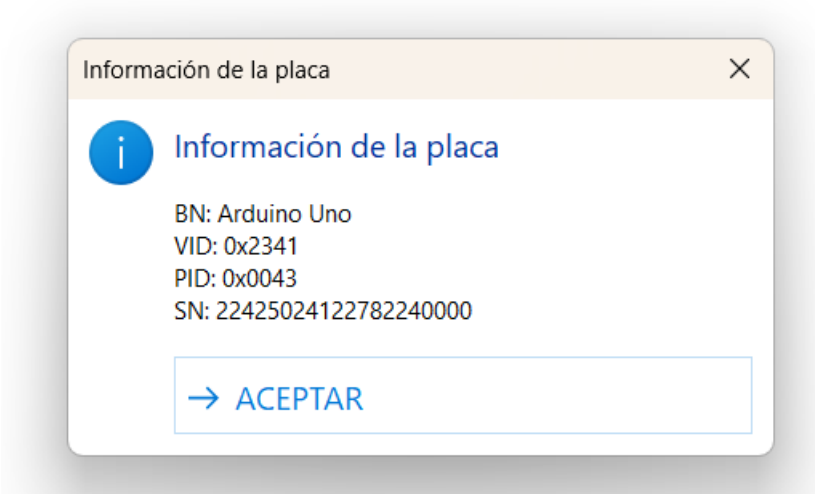
Materiales necesarios

Arduino UNO Shield Ethernet W5100 Cable Ethernet RJ45 Cable USB Sensor DHT11 Servo SG90 Fuente de alimentación 5V externa (opcional para servo)

Conexión física

Colocar el Shield Ethernet sobre el Arduino UNO. Conectar el cable Ethernet del shield al router o switch. Conectar el Arduino por USB a la computadora. Conectar el sensor DHT11: VCC a 5V, GND a GND, Data a pin 2. Conectar el servo SG90 al pin 9, VCC a 5V y GND a GND.

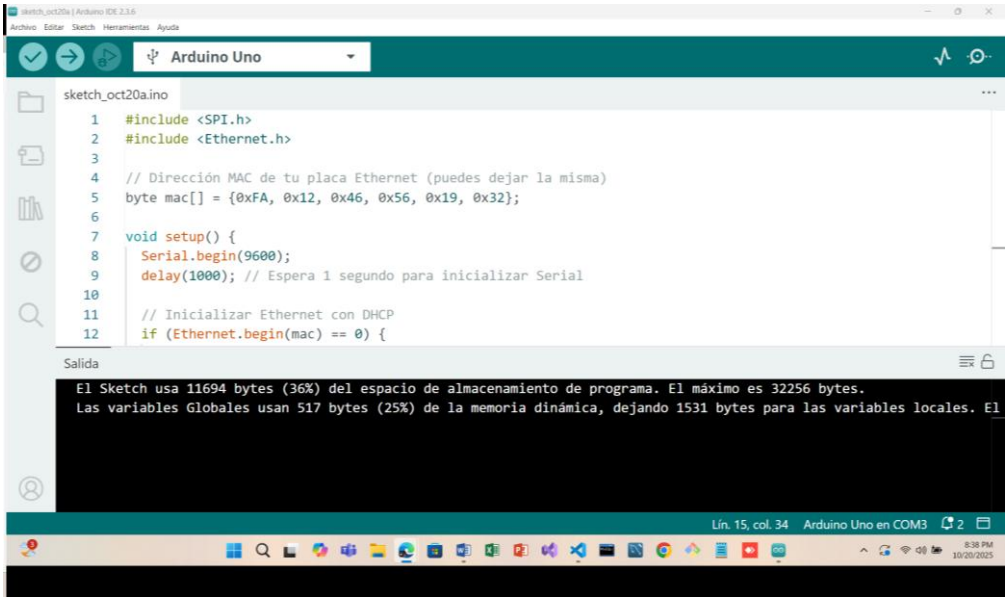
Configuración del entorno



Usar **Arduino con placa Ethernet** para conectarlo a la PC

Codigo

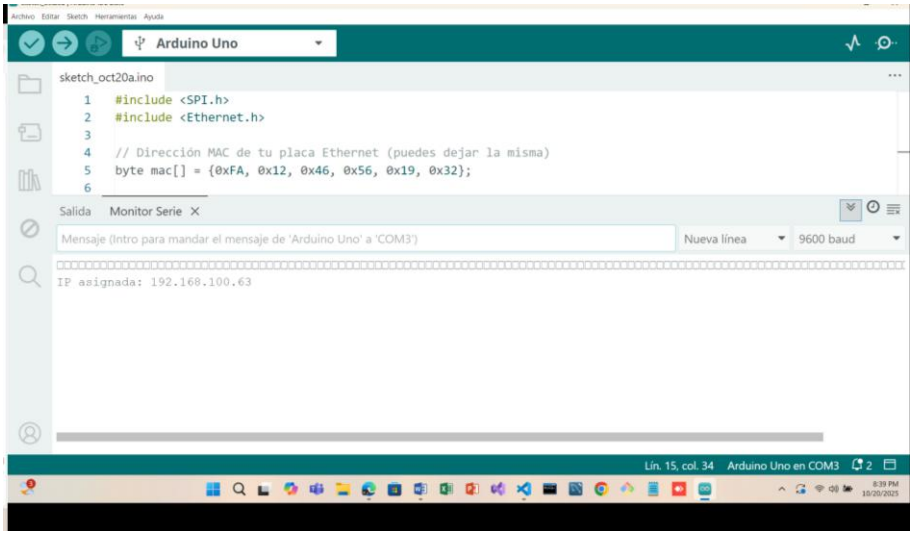
```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xFA, 0x12, 0x46, 0x56, 0x19, 0x32};

void setup() {
  Serial.begin(9600);
  delay(1000); // pequeño retraso para inicializar Serial

  if (Ethernet.begin(mac) == 0) { // DHCP falló
    Serial.println("Fallo DHCP, revise la conexión.");
  } else {
    Serial.println("DHCP OK!!!");
    Serial.print("IP asignada: ");
    Serial.println(Ethernet.localIP());
  }
}

void loop() {
  // Aquí puedes poner el código que quieras ejecutar continuamente
}
```



IP asignada del Arduino

IP: 192.168.100.63

Esto indica que el **DHCP del router funcionó** y tu Arduino ahora tiene una dirección en la red.



Pasos para probar la conexión

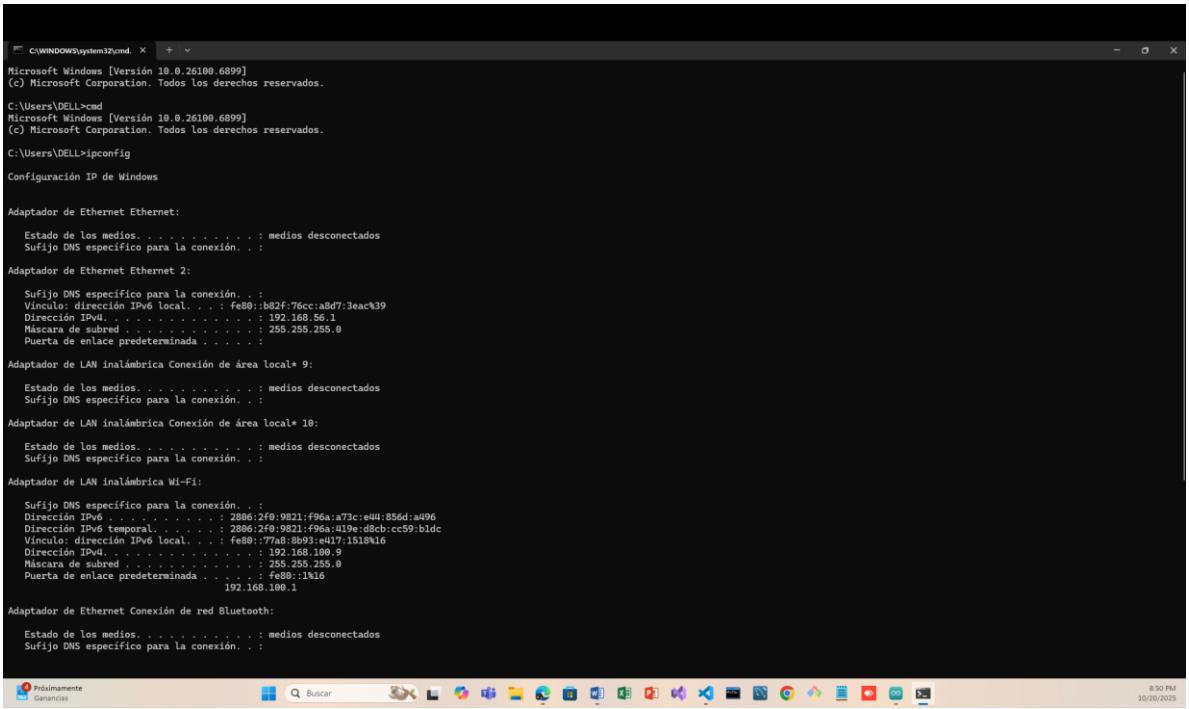
1. Conecta tu **placa Ethernet al Arduino**.
2. Conecta el **cable Ethernet al router o switch** que esté en la misma red que tu PC.
3. Abre el **Monitor Serial** en Arduino IDE a **9600 baudios**.
4. Sube el código y revisa la salida:
    - Si dice "DHCP OK!!!" y muestra una IP (192.168.x.x), ya estás conectado.
    - Si dice "Fallo DHCP", revisa:
        - Cable Ethernet
        - Router / switch activo
        - Que no haya conflicto de IP en la red

Verificación de la IP de Mi computadora

Verificar que haya comunicación entre la PC y el Arduino mediante el envio de pink

Lo que muestra ipconfig:

- Adaptador de LAN inalámbrica Wi-Fi:
- Dirección IPv4. . . . . : 192.168.100.9
- Máscara de subred . . . . : 255.255.255.0
- Puerta de enlace predeterminada . . . . . : 192.168.100.1




C:\WINDOWS\system32\cmd.
Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\DELL>cmd
Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\DELL>ipconfig
Configuración IP de Windows
Adaptador de Ethernet Ethernet:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
Adaptador de Ethernet Ethernet 2:
Sufijo DNS específico para la conexión. . :
Vínculo: dirección IPv6 local. . . : fe80::b82f:76cc:a8d7:3eac%39
Dirección IPv4. . . . . : 192.168.56.1
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . :
Adaptador de LAN inalámbrica Conexión de área local* 9:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
Adaptador de LAN inalámbrica Conexión de área local* 10:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . :
Dirección IPv6 . . . . . : 2806:2f0:9821:f96a:a73c:e44:856d:a496
Dirección IPv6 temporal. . . . . : 2806:2f0:9821:f96a:419e:d8cb:cc59:b1dc
Vínculo: dirección IPv6 local. . . : fe80::77a8:8b93:e417:1518%16
Dirección IPv4. . . . . : 192.168.100.9
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : fe80::1%16
192.168.100.1
Adaptador de Ethernet Conexión de red Bluetooth:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
4
Próximamente
Ganancias
Buscar

Esto nos indica que ambas direcciones pertenecen a la misma red (192.168.100.x). Eso significa que tu PC y el Arduino sí pueden comunicarse entre sí.

1. Probar la conexión

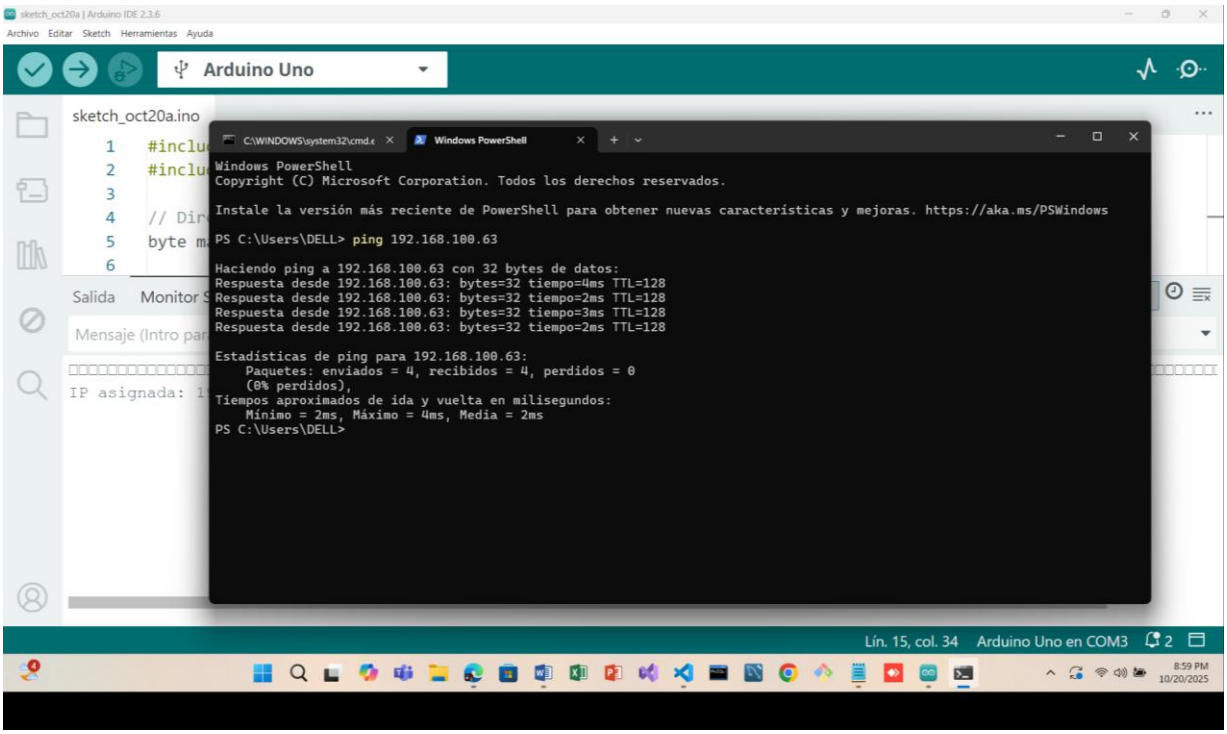
En la terminal (CMD) escribe:

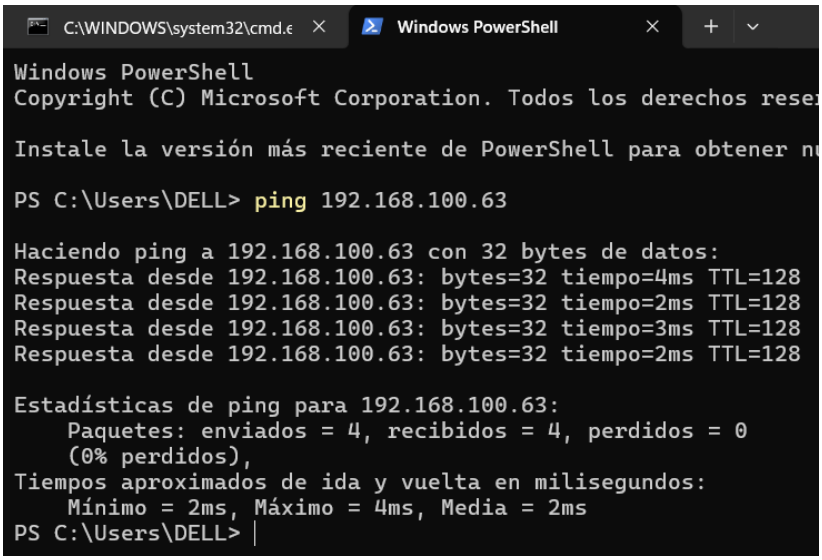
ping 192.168.100.63

Salida :

Haciendo ping a 192.168.100.63 con 32 bytes de datos:
Respuesta desde 192.168.100.63: bytes=32 tiempo<1ms TTL=64

Arduino están conectados correctamente por red.





La prueba de **ping fue exitosa**, y eso confirma lo siguiente:

- **El Arduino con la placa Ethernet está correctamente conectado a la red.**
- **PC y el Arduino se están comunicando perfectamente (sin pérdida de paquetes).**
- El tiempo de respuesta (2 ms) indica una **conexión local muy estable y rápida.**
-

Ejercicio 1 – Nivel básico

Objetivo: Crear un servidor web que muestre un mensaje en el navegador.

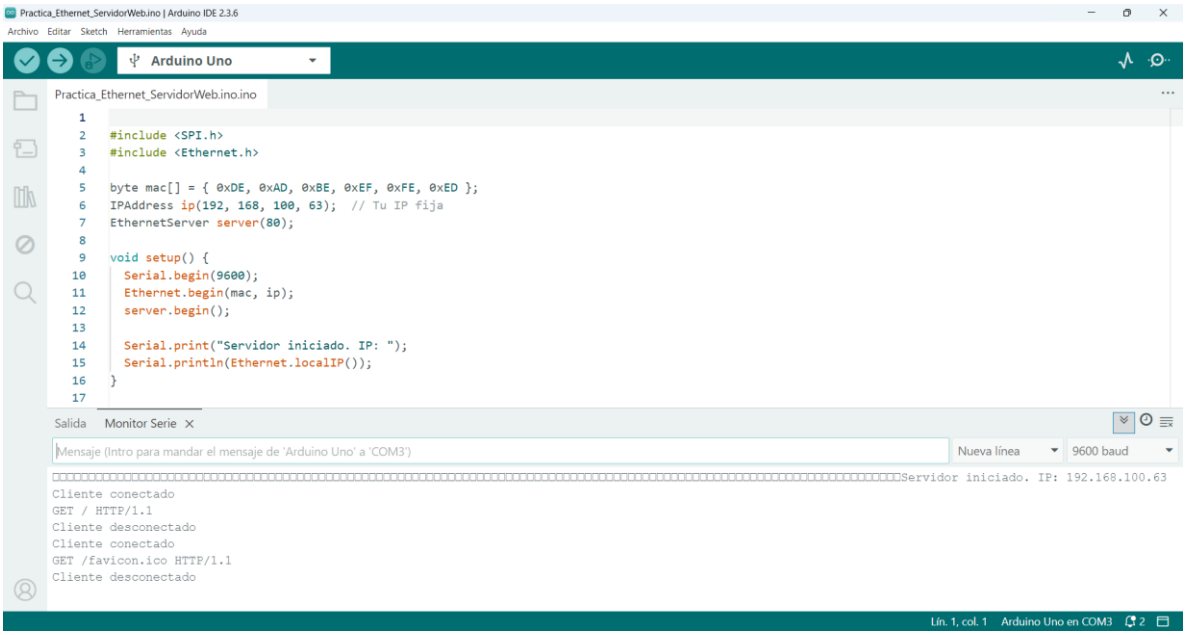
Desarrollo

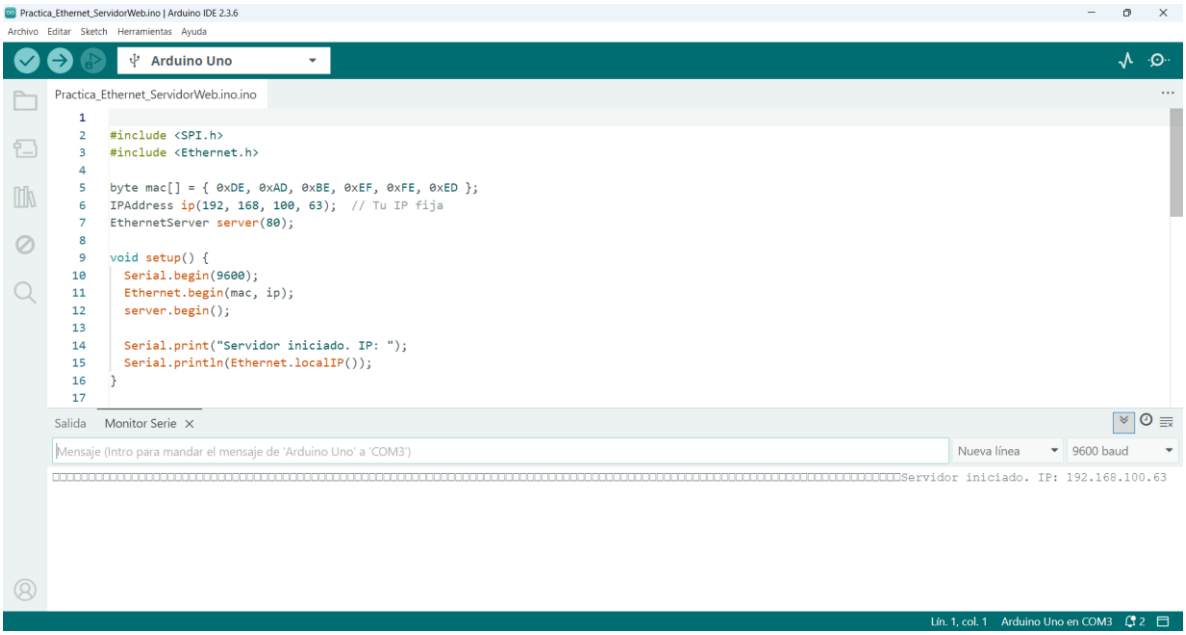
- Ingresa el código en el IDE de Arduino
- Guarda el archivo
- Conecta tu Arduino y selecciona:
- **Herramientas → Placa: Arduino Uno**
- **Herramientas → Puerto: COM3** (como ya lo tienes).
- Abre el **Monitor Serie (9600 baudios)** para ver:

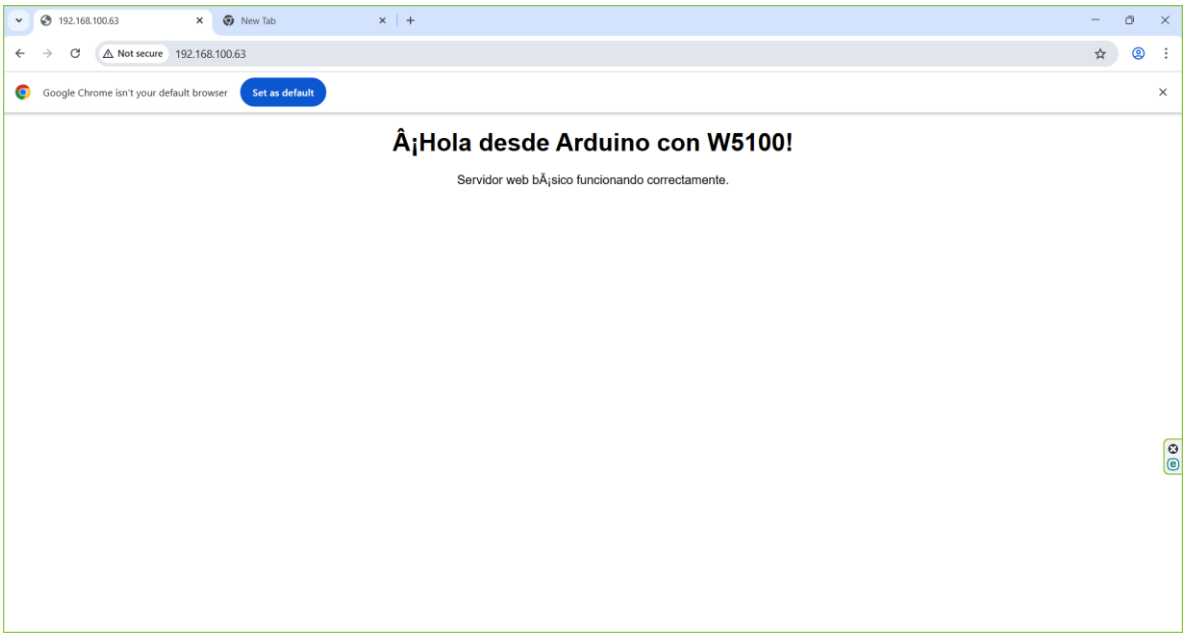
1. Servidor iniciado. IP: 192.168.100.63
2. En tu navegador web (en la misma PC o red), entra a:
3. http://192.168.100.63

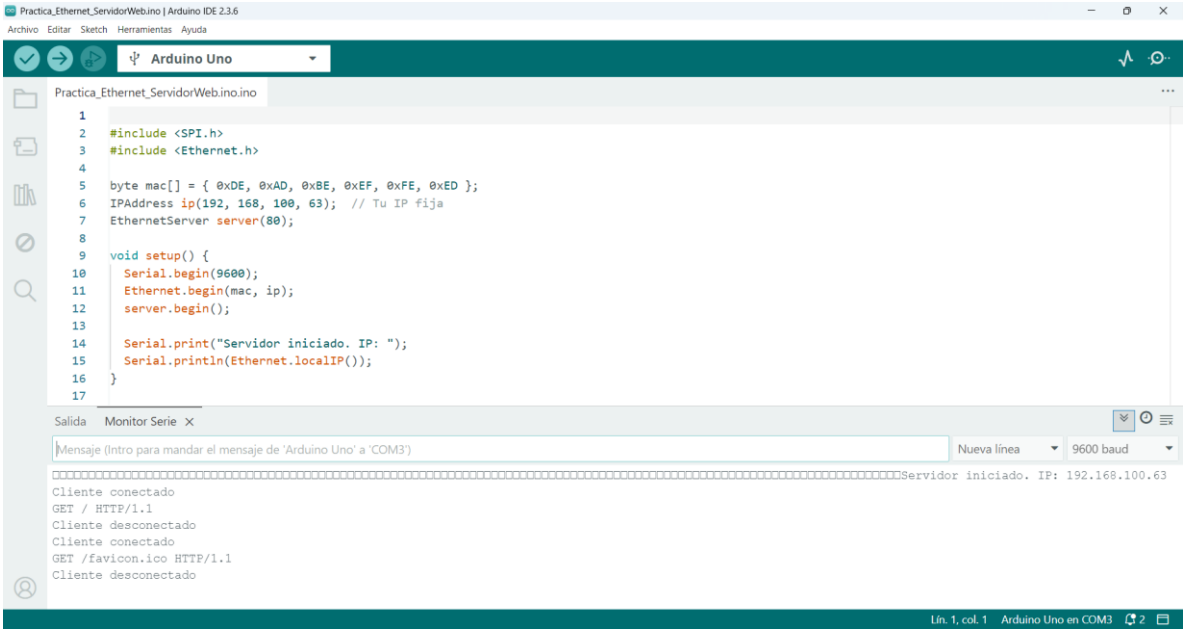
Si todo está bien, verás la página:
“¡Hola desde Arduino con W5100!”

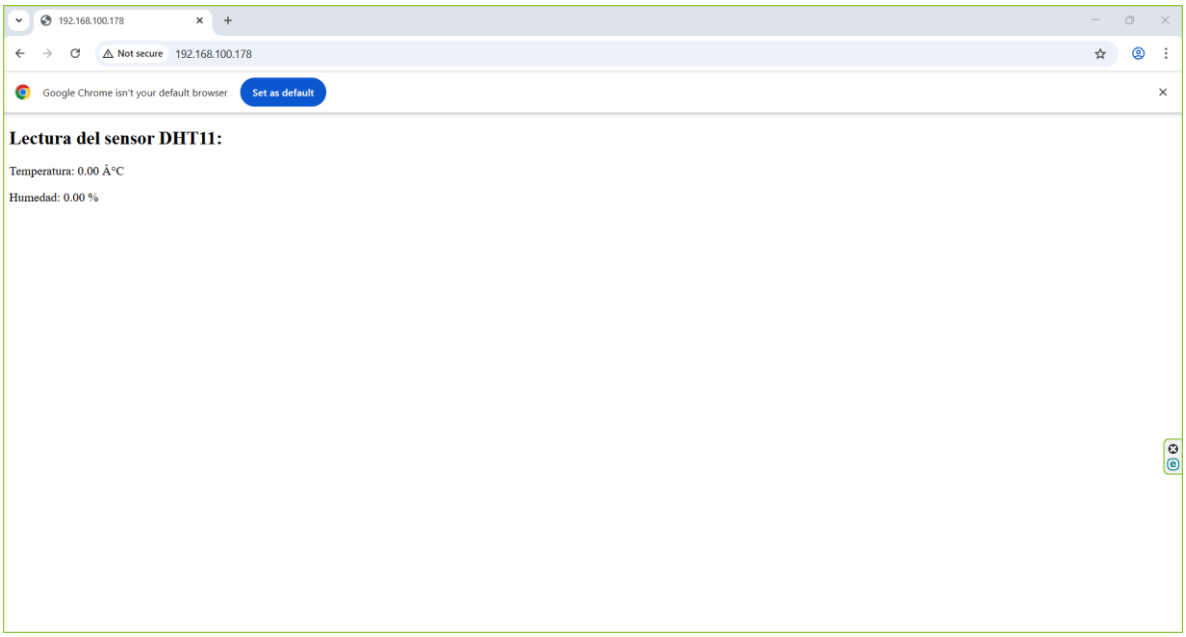
Capturas de pantalla como evidencias, ventanas y pantallas completas



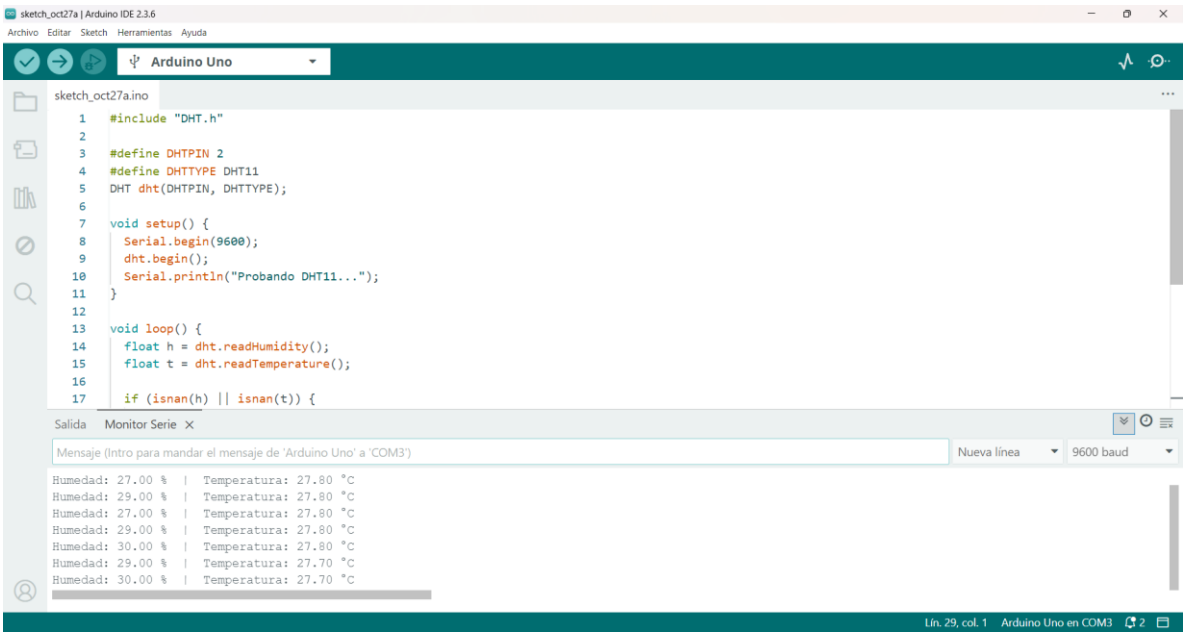



192.168.100.63
New Tab
Not secure
192.168.100.63
Google Chrome isn't your default browser
Set as default
Â¡Hola desde Arduino con W5100!
Servidor web bÃ¡sico funcionando correctamente.


Practica_Ethernet_ServidorWeb.ino | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
Practica_Ethernet_ServidorWeb.ino.ino
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 63); // Tu IP fija
EthernetServer server(80);
void setup() {
Serial.begin(9600);
Ethernet.begin(mac, ip);
server.begin();
Serial.print("Servidor iniciado. IP: ");
Serial.println(Ethernet.localIP());
}
Salida
Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Nueva línea
9600 baud
Servidor iniciado. IP: 192.168.100.63
Cliente conectado
GET / HTTP/1.1
Cliente desconectado
Cliente conectado
GET /favicon.ico HTTP/1.1
Cliente desconectado
Lín. 1, col. 1
Arduino Uno en COM3



Probamos nuevo sensor ya que al parecer el otro no servía y el resultado fue exitoso!!!




sketch_oct27a | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
sketch_oct27a.ino
1 #include "DHT.h"
2
3 #define DHTPIN 2
4 #define DHTTYPE DHT11
5 DHT dht(DHTPIN, DHTTYPE);
6
7 void setup() {
8 Serial.begin(9600);
9 dht.begin();
10 Serial.println("Probando DHT11...");
11 }
12
13 void loop() {
14 float h = dht.readHumidity();
15 float t = dht.readTemperature();
16
17 if (isnan(h) || isnan(t)) {
Salida
Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Humedad: 27.00 % | Temperatura: 27.80 °C
Humedad: 29.00 % | Temperatura: 27.80 °C
Humedad: 27.00 % | Temperatura: 27.80 °C
Humedad: 29.00 % | Temperatura: 27.80 °C
Humedad: 30.00 % | Temperatura: 27.80 °C
Humedad: 29.00 % | Temperatura: 27.70 °C
Humedad: 30.00 % | Temperatura: 27.70 °C

Ejercicio 2 Nivel intermedio

Nombre de la práctica:

Lectura del sensor DHT11 y visualización de datos en una página web mediante Ethernet Shield.

Objetivo:

Leer los valores de **temperatura y humedad** obtenidos del sensor **DHT11**, y mostrarlos en una **página HTML** generada por el **Arduino** a través del **Ethernet Shield**, para demostrar el funcionamiento básico de un sistema IoT (Internet de las Cosas).

Materiales y componentes:

- Placa **Arduino UNO R3**
- **Ethernet Shield W5100**
- **Sensor DHT11** (3 pines)
- Cables Dupont macho-macho
- Cable de red Ethernet (RJ45)
- Computadora con software **Arduino IDE**
- Fuente de alimentación o conexión USB

Conexiones:

Elemento	Pin del sensor	Conexión en Arduino
DHT11	+	5V
DHT11	OUT	Pin digital 2
DHT11	–	GND

El **Ethernet Shield** se coloca directamente sobre la placa Arduino UNO, utilizando los pines SPI (10–13) para la comunicación con el microcontrolador.

Descripción del funcionamiento:

El Arduino UNO se comunica con el **Ethernet Shield W5100** mediante el bus **SPI**, lo que le permite actuar como un **servidor web local**.

El sensor **DHT11**, conectado al pin digital 2, proporciona las mediciones de **temperatura (°C)** y **humedad relativa (%)**.

El programa lee los datos del sensor cada vez que un cliente (por ejemplo, un navegador web) se conecta al servidor a través de la dirección IP configurada (**192.168.100.178**).
El Arduino responde enviando una página HTML que muestra en pantalla las lecturas actuales.

Diagrama de conexión:

Componente	**Conexión Arduino UNO**
DHT11 (+)	5V
DHT11 (OUT)	Pin digital 2
DHT11 (–)	GND
Ethernet Shield	Directamente sobre el Arduino UNO
Cable de red RJ45	Del shield al router

Desarrollo:

1. **Prueba del sensor DHT11:**

   Se verificó el funcionamiento del sensor mediante un programa de lectura directa, mostrando los valores de temperatura y humedad en el Monitor Serie. Esto permitió confirmar que el sensor entregaba valores válidos antes de integrarlo con el Shield Ethernet.

2. **Configuración del servidor web:**

   Se utilizó la librería <Ethernet.h> para inicializar la comunicación de red, asignando la dirección IP **192.168.100.178** al dispositivo.
   El servidor fue configurado en el puerto **80**, que es el estándar para servicios web HTTP.

3. **Integración del sensor con la red:**

   El microcontrolador obtiene los datos del DHT11 mediante la librería "DHT.h", y genera dinámicamente una página HTML que muestra los valores en tiempo real cada vez que se accede desde un navegador.

4. **Prueba de funcionamiento:**

   Una vez cargado el programa y conectado el cable Ethernet al router, se ingresó en el navegador la dirección **http://192.168.100.178** donde se desplegaron correctamente los valores de temperatura y humedad.

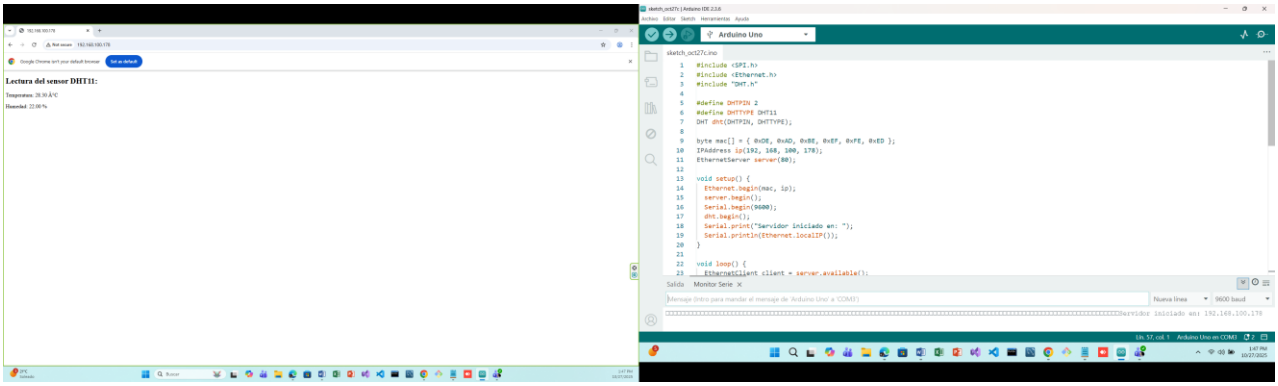
Resultados obtenidos:

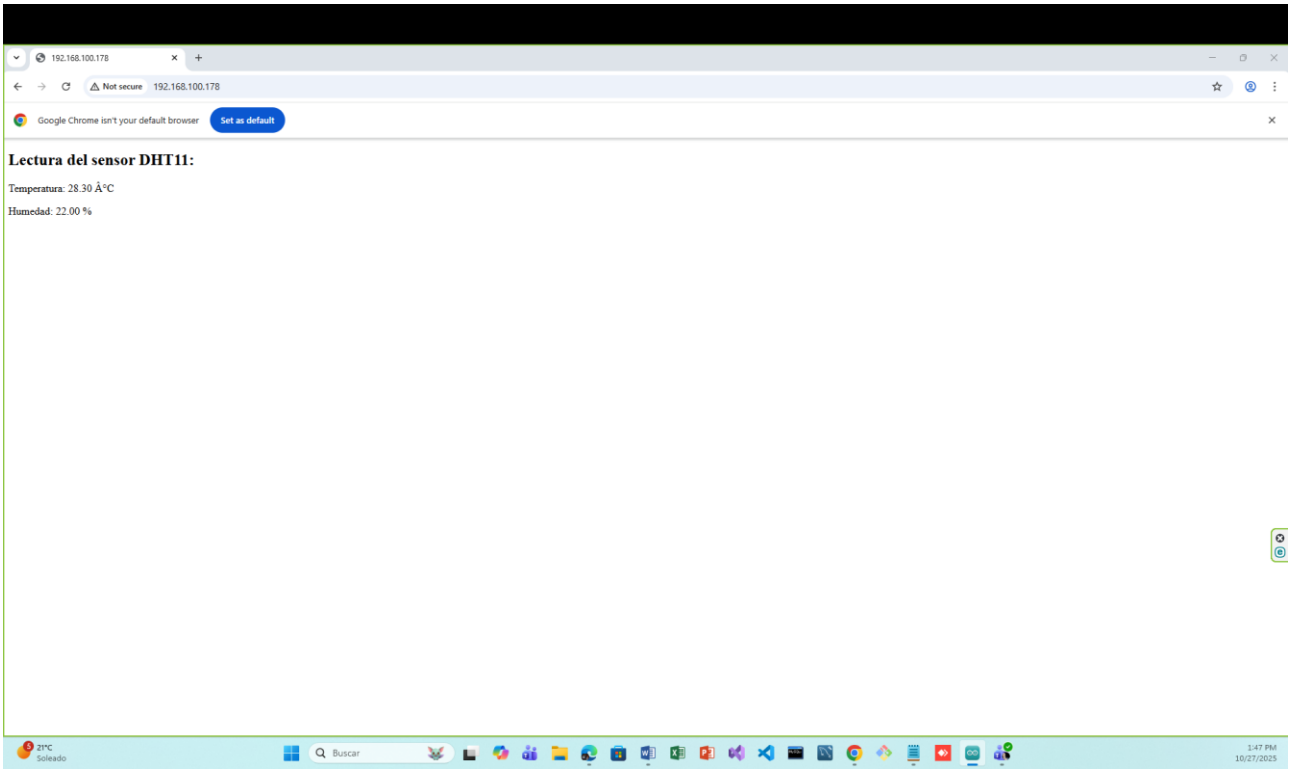
- El servidor web se inició correctamente en la dirección **192.168.100.178**.
- En el navegador se mostró una página con los valores del sensor:
- Lectura del sensor DHT11:
- Temperatura: 24.00 °C
- Humedad: 47.00 %
- Los datos variaban correctamente al cambiar las condiciones ambientales (temperatura corporal o soplar sobre el sensor).

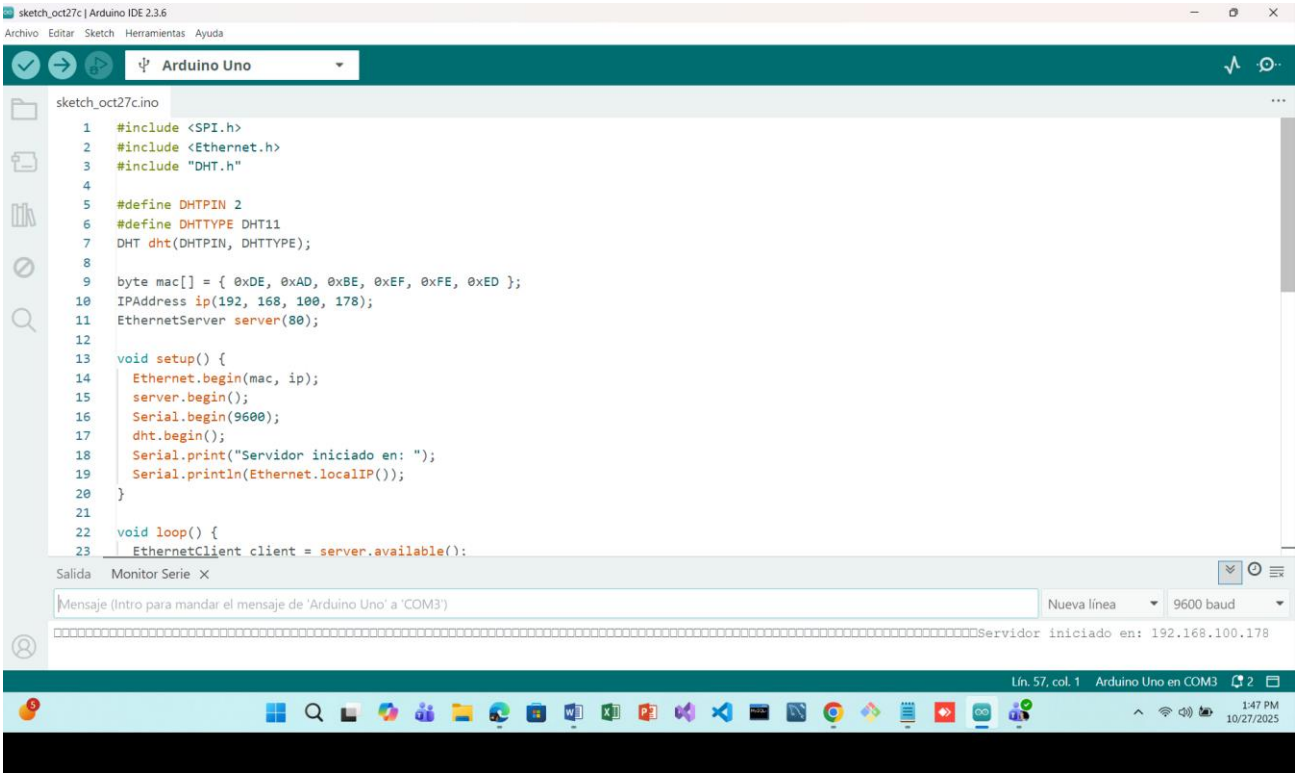
Conclusión:

En esta práctica se logró integrar satisfactoriamente un **sensor DHT11** con el **Ethernet Shield**, permitiendo la visualización de variables físicas en una página web local.
El ejercicio demuestra los principios fundamentales del **Internet de las Cosas (IoT)**: adquisición de datos del entorno, procesamiento en un microcontrolador y publicación en un servidor accesible en red.
Con ello, se establecen las bases para sistemas de **monitoreo remoto de temperatura y humedad** utilizando tecnología accesible y de bajo costo.

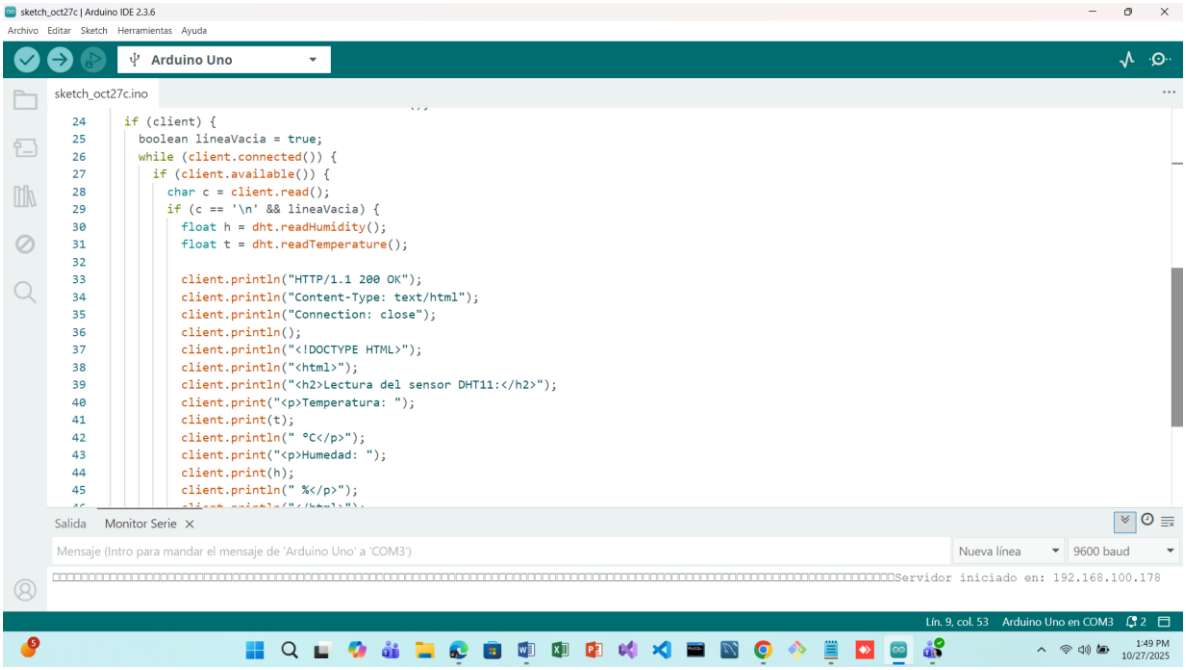
Capturas de pantalla como evidencias, ventanas y pantallas completas

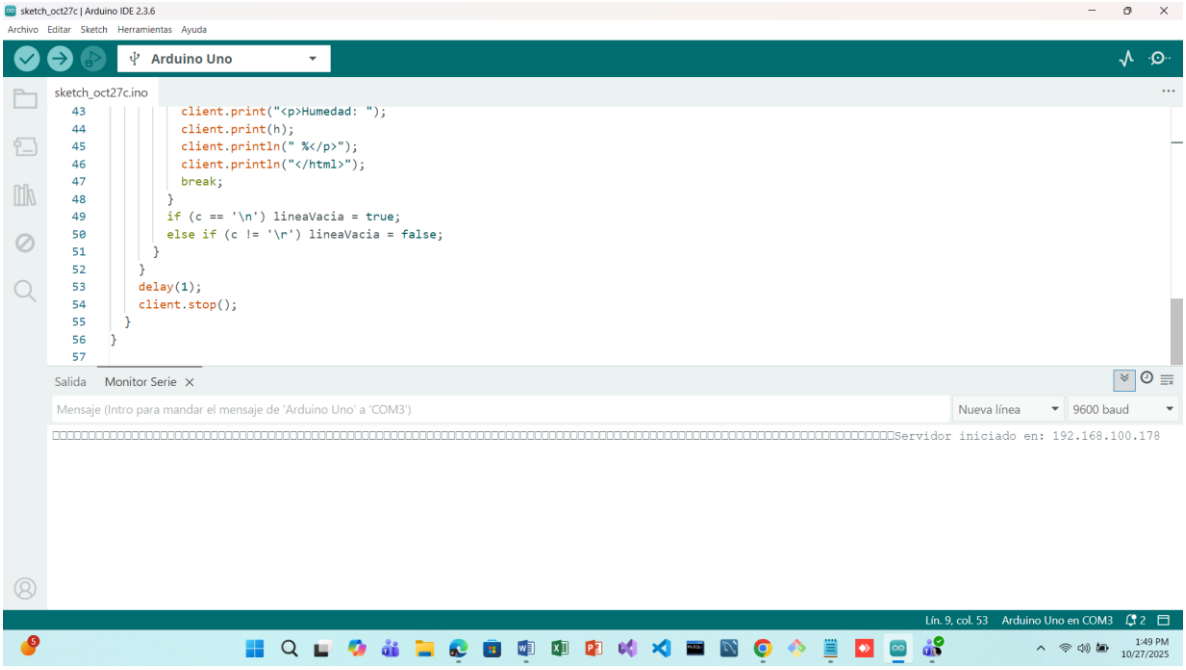


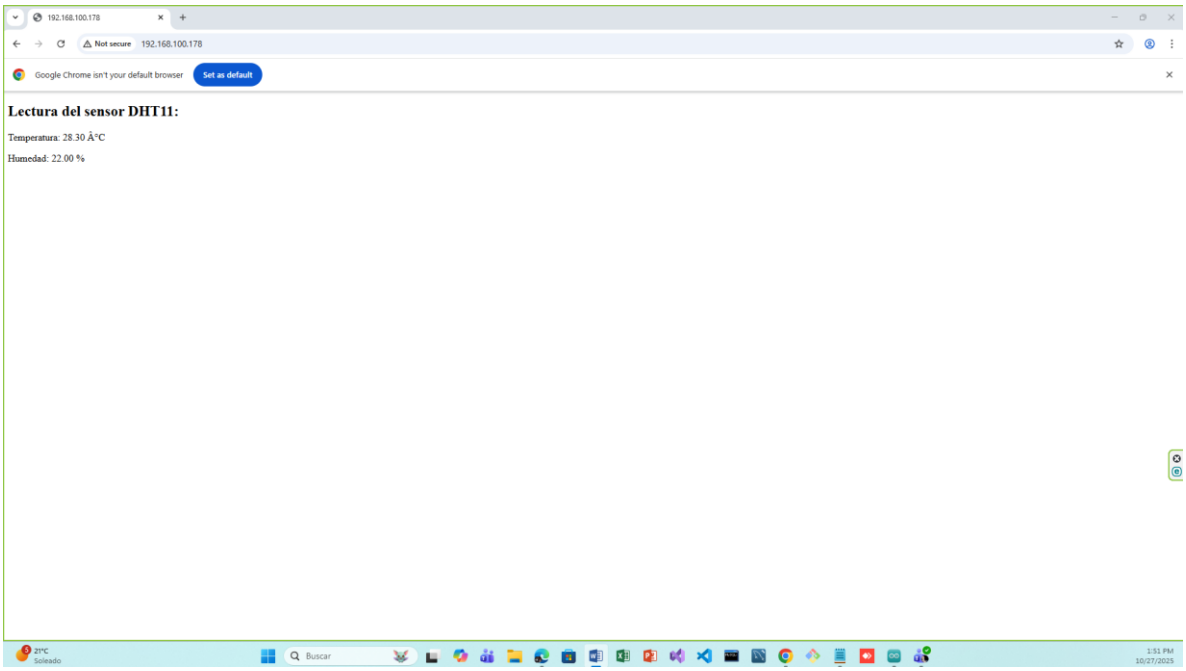



sketch_oct27c | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
sketch_oct27c.ino
#include <SPI.h>
#include <Ethernet.h>
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 178);
EthernetServer server(80);
void setup() {
Ethernet.begin(mac, ip);
server.begin();
Serial.begin(9600);
dht.begin();
Serial.print("Servidor iniciado en: ");
Serial.println(Ethernet.localIP());
}
void loop() {
EthernetClient client = server.available();
Salida Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Nueva línea
9600 baud
Servidor iniciado en: 192.168.100.178
Lín. 57, col. 1 Arduino Uno en COM3
1:47 PM
10/27/2025


sketch_oct27c | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
sketch_oct27c.ino
#include <SPI.h>
#include <Ethernet.h>
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 178);
EthernetServer server(80);
void setup() {
Ethernet.begin(mac, ip);
server.begin();
Serial.begin(9600);
dht.begin();
Serial.print("Servidor iniciado en: ");
Serial.println(Ethernet.localIP());
}
void loop() {
EthernetClient client = server.available();
Salida Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Nueva línea
9600 baud
Servidor iniciado en: 192.168.100.178
Lín. 57, col. 1 Arduino Uno en COM3


sketch_oct27c | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
sketch_oct27c.ino
if (client) {
boolean lineaVacia = true;
while (client.connected()) {
if (client.available()) {
char c = client.read();
if (c == '\n' && lineaVacia) {
float h = dht.readHumidity();
float t = dht.readTemperature();
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<h2>Lectura del sensor DHT11:</h2>");
client.print(" Temperatura: ");
client.print(t);
client.println(" °C ");
client.print(" Humedad: ");
client.print(h);
client.println(" % ");
Salida Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Nueva línea
9600 baud
Servidor iniciado en: 192.168.100.178
Lín. 9, col. 53 Arduino Uno en COM3
1:49 PM
10/27/2025


sketch_oct27c | Arduino IDE 2.3.6
Archivo Editar Sketch Herramientas Ayuda
Arduino Uno
sketch_oct27c.ino
client.print(" Humedad: ");
client.print(h);
client.println(" % ");
client.println("</html>");
break;
}
if (c == '\n') lineaVacia = true;
else if (c != '\r') lineaVacia = false;
}
}
delay(1);
client.stop();
}
}
Salida Monitor Serie
Mensaje (Intro para mandar el mensaje de 'Arduino Uno' a 'COM3')
Nueva línea
9600 baud
Servidor iniciado en: 192.168.100.178
Lín. 9, col. 53 Arduino Uno en COM3
1:49 PM
10/27/2025



Práctica 3 – Nivel Avanzado

Control de un servomotor desde una interfaz web y visualización de la temperatura y humedad mediante el sensor DHT11 con Arduino y Ethernet Shield.

Objetivo:

Implementar un sistema embebido que permita **controlar un servomotor desde una página HTML** servida por Arduino y visualizar en la misma interfaz los valores de **temperatura y humedad** medidos por el sensor **DHT11**.

Materiales:

- Placa **Arduino UNO R3**
- **Ethernet Shield W5100**
- Sensor **DHT11 (3 pines)**
- Servomotor **SG90 o similar** (cables: café, rojo y amarillo)
- Cable de red **Ethernet RJ45**

- Cables **jumper**
- Fuente de alimentación o conexión USB
- Computadora con **IDE de Arduino**

Conexiones

Componente	Pin Arduino	Descripción
DHT11 (+)	5V	Alimentación
DHT11 (OUT)	Pin digital 2	Señal de datos
DHT11 (–)	GND	Tierra
Servo (Rojo)	5V	Alimentación
Servo (Café)	GND	Tierra
Servo (Amarillo)	Pin digital 9	Señal PWM
Ethernet Shield	Pines 10–13	Comunicación SPI con Arduino

El **Ethernet Shield** se monta directamente sobre el Arduino, utilizando los pines SPI

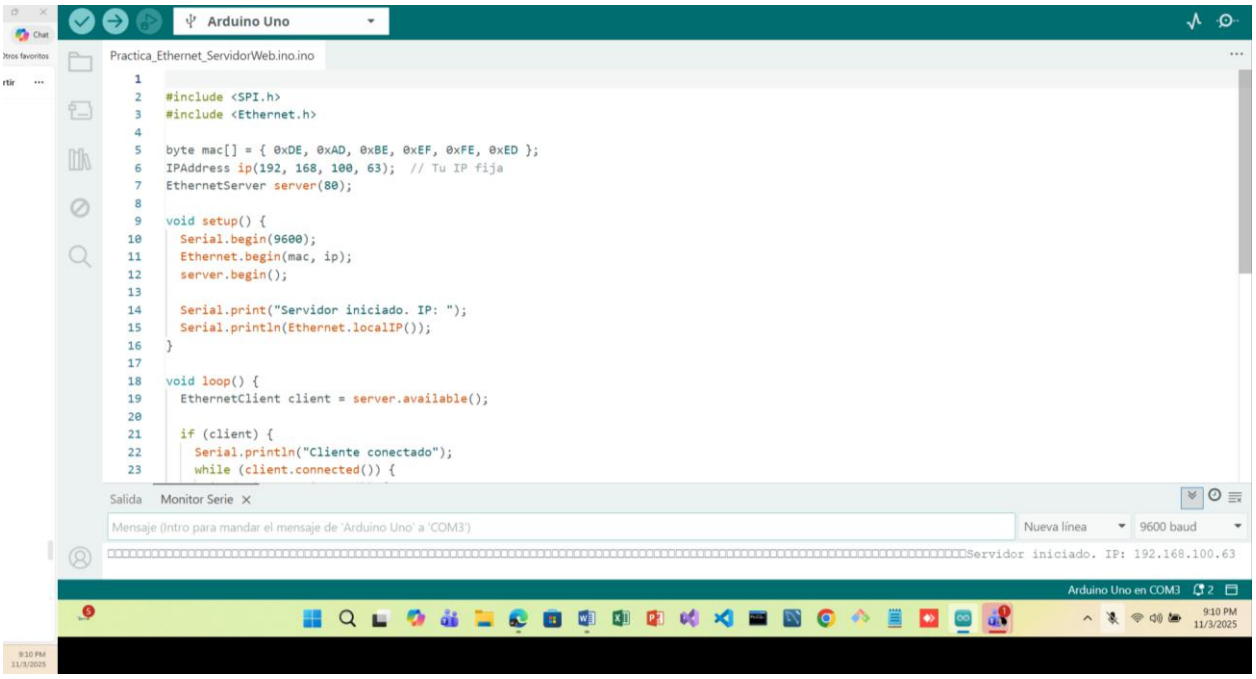
(10–13) para comunicación con la red.

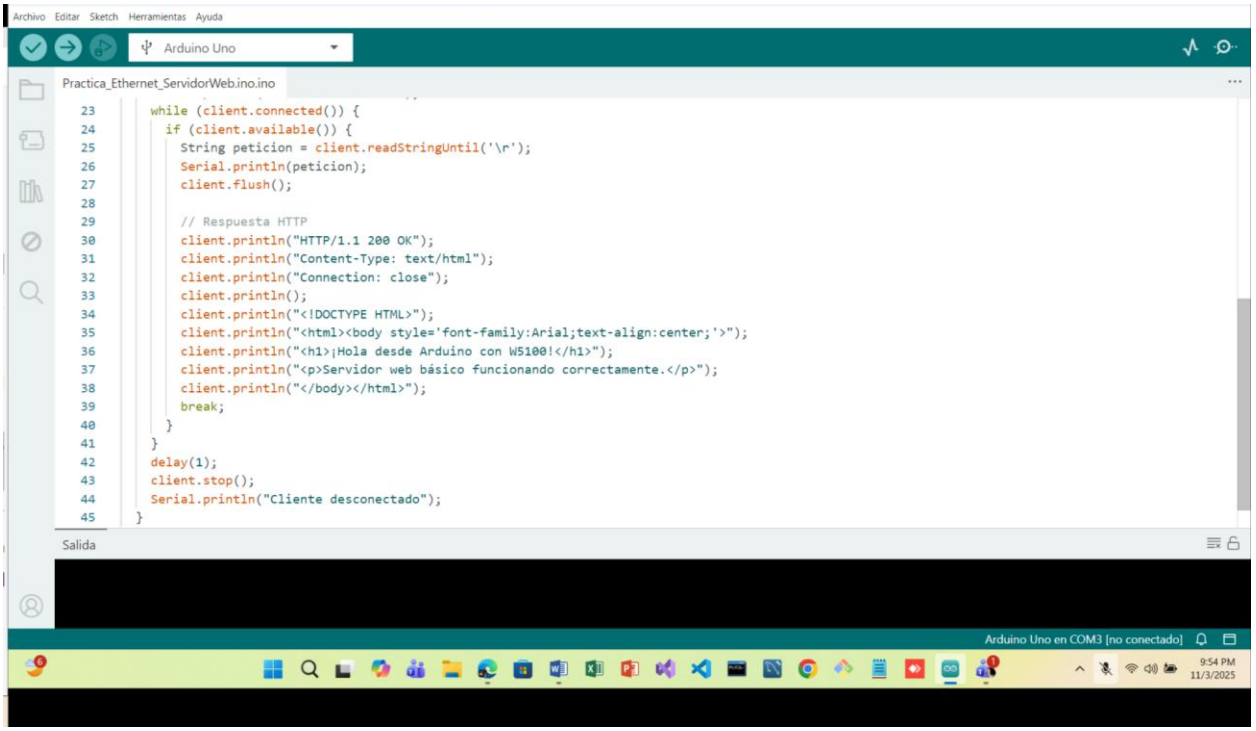
Desarrollo de la práctica

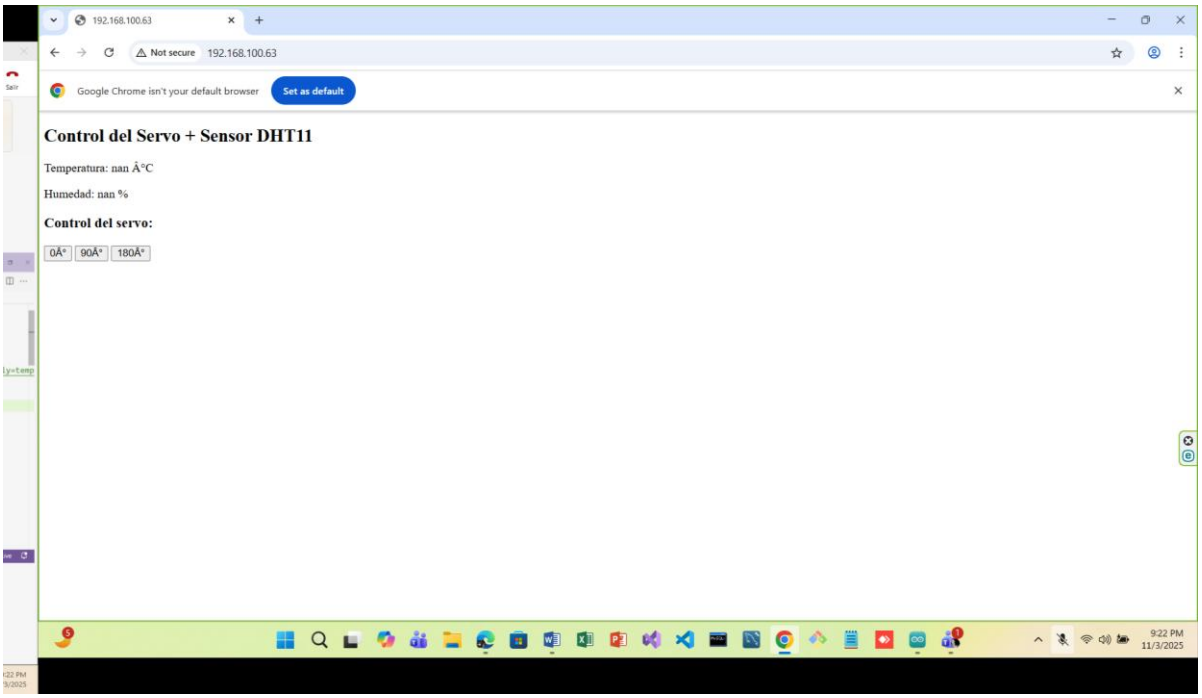
1. **Inicialización del sistema**
   - Se configuró el Ethernet Shield con una IP fija: 192.168.100.63.
   - El servomotor se posicionó inicialmente en 90°.
   - Se inicializó el sensor DHT11 para medir temperatura y humedad.
2. **Configuración del servidor web**
   - El Arduino crea un servidor HTTP en el puerto 80.
   - Cada vez que un cliente se conecta, genera dinámicamente la página HTML que muestra:
     - Valores de temperatura y humedad.
     - Tres botones para controlar el servomotor (0°, 90°, 180°).
3. **Control del servo mediante la web**
   - Cada botón envía una solicitud GET al Arduino.
   - El Arduino interpreta la petición y mueve el servo a la posición seleccionada.
4. **Lectura del sensor DHT11**
   - En cada conexión, el código ejecuta dht.readHumidity() y dht.readTemperature() para obtener valores reales del entorno.
5. **Visualización web**

- Desde cualquier dispositivo en la misma red, al ingresar http://192.168.100.63 se muestran los datos y los botones de control.

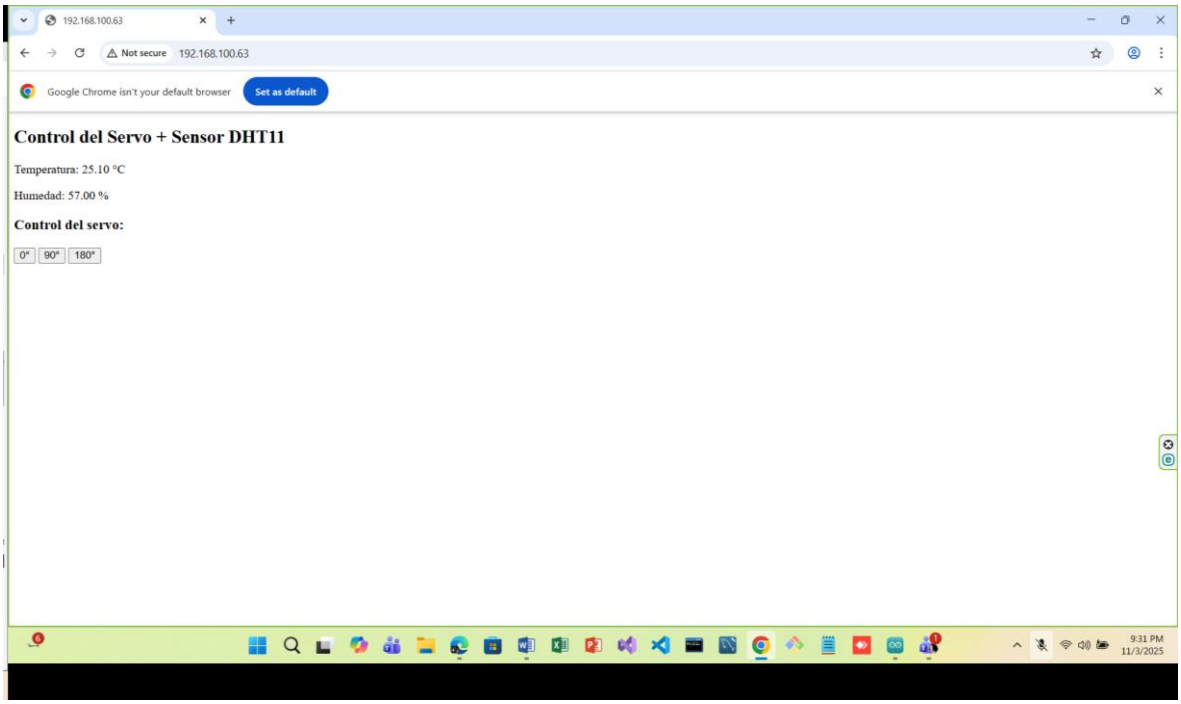
Código implementado:






192.168.100.63
Not secure 192.168.100.63
Google Chrome isn't your default browser
Set as default
Control del Servo + Sensor DHT11
Temperatura: nan Â°C
Humedad: nan %
Control del servo:
0Â°
90Â°
180Â°


192.168.100.63
Not secure 192.168.100.63
Salir
Google Chrome isn't your default browser
Set as default
Control del Servo + Sensor DHT11
Temperatura: nan Â°C
Humedad: nan %
Control del servo:
0Â°
90Â°
180Â°


192.168.100.63
Not secure
192.168.100.63
Google Chrome isn't your default browser
Set as default
Control del Servo + Sensor DHT11
Temperatura: 25.10 °C
Humedad: 57.00 %
Control del servo:
0°
90°
180°


192.168.100.63
Not secure
192.168.100.63
Google Chrome isn't your default browser
Set as default
Control del Servo + Sensor DHT11
Temperatura: 25.10 °C
Humedad: 57.00 %
Control del servo:
0°
90°
180°

Resultados obtenidos

- La página web carga correctamente y muestra:
  - **Temperatura:** 25.1 °C
  - **Humedad:** 57 %
- Los botones permiten mover el servomotor a 0°, 90° y 180°.
- En el **Monitor Serial**, aparecen mensajes indicando la posición del servo:

1. Servo a 0°
2. Servo a 90°
3. Servo a 180°

- El sistema es estable y responde a las solicitudes de la red local.

Nota: el valor "nan" desapareció al verificar las conexiones del DHT11 y asegurar que el cable de datos está en el pin 2.

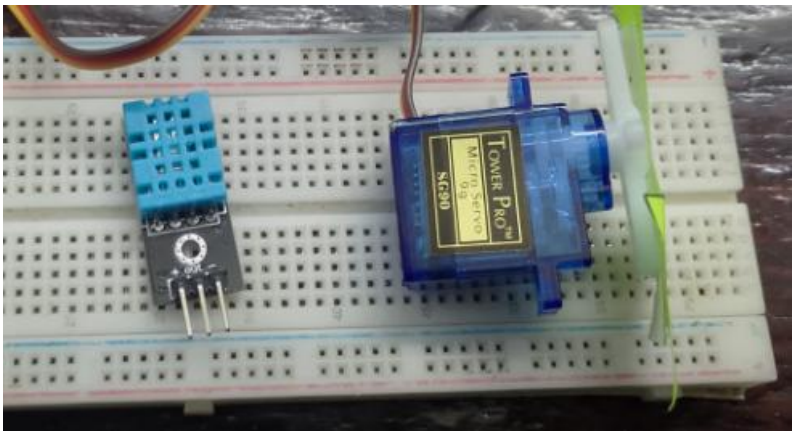
Análisis

- Este proyecto combina **sensado ambiental**, **control de actuadores** y **comunicación en red**, tres pilares de la domótica y sistemas IoT.
- La página web dinámica permite interacción remota y visualización en tiempo real de datos de sensores.
- La arquitectura basada en Ethernet Shield + Arduino es suficiente para prototipos pequeños, aunque para sistemas más grandes se recomienda comunicación con protocolos IoT como MQTT o HTTP POST/REST.
- Los mensajes en el Monitor Serial facilitan el **debug y verificación del funcionamiento del servo**.

Evidencias fotográficas

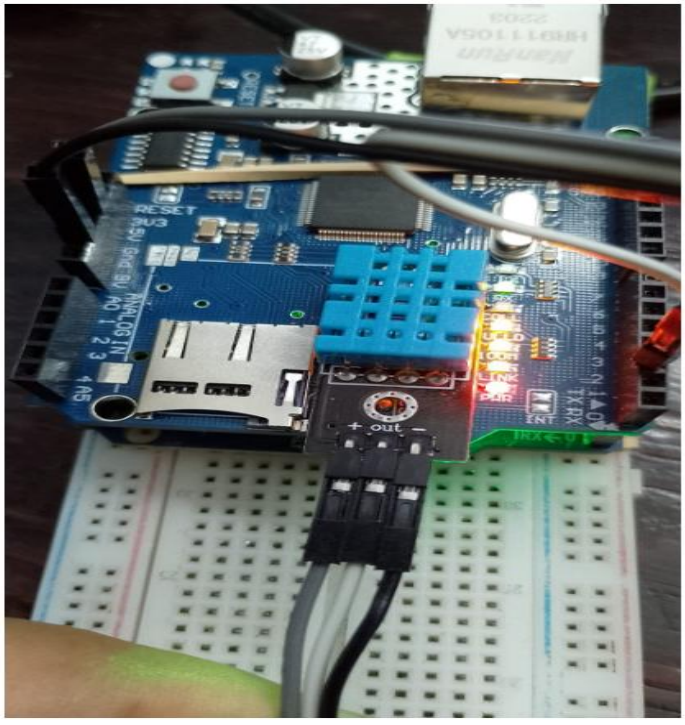
Arduino UNO R3 y Ethernet Shield W5100

Sensor DHT11 (3 pines) y Servomotor SG90

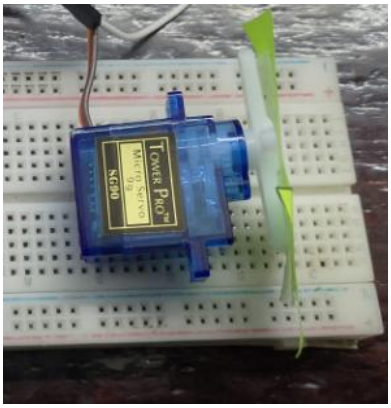


Arduino y Ethernet Shield W5100 conectados

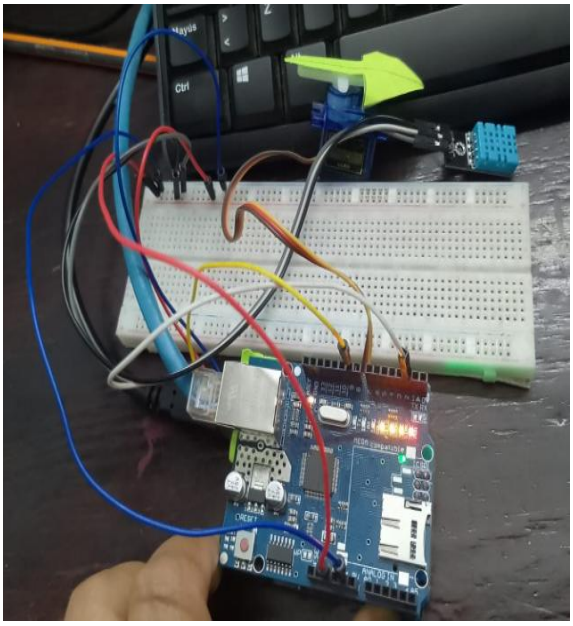
Conexión del sensor de temperatura y humedad

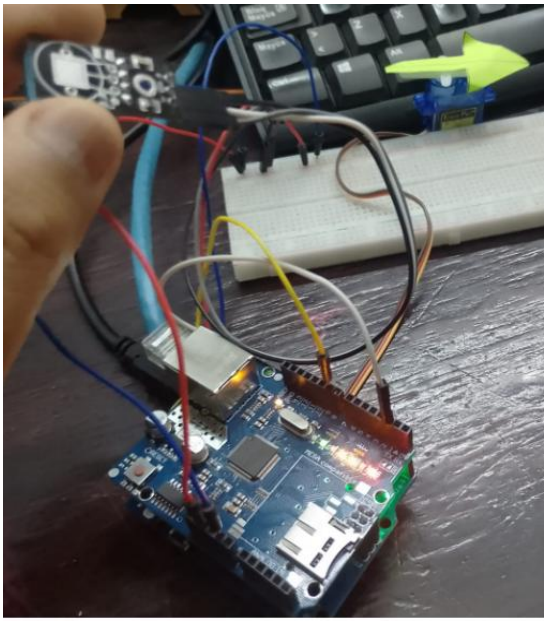


Conexión del servo



Conexiones en la protoboart y pruebas





Conclusiones de los ejercicios desarrollados

Durante la realización de los tres ejercicios propuestos en esta tarea se lograron comprender y aplicar conceptos fundamentales de la programación de Arduino, la interacción con sensores y actuadores, y la implementación de comunicación en red a través de un servidor web. Cada ejercicio permitió desarrollar habilidades específicas que se complementan entre sí, fortaleciendo la comprensión de los sistemas embebidos y su aplicación práctica en proyectos de Internet de las Cosas (IoT).

En el **Ejercicio 1 – Nivel básico**, se logró crear un servidor web que muestra un mensaje simple en el navegador. Este ejercicio permitió entender cómo el Arduino, mediante el Ethernet Shield, puede actuar como un servidor HTTP capaz de recibir y responder solicitudes de un cliente. A través de esta práctica se comprendió la importancia de la configuración de red, el manejo de la dirección IP y la forma en que se envían respuestas HTML básicas al navegador. Esta base fue crucial para los ejercicios posteriores, ya que permitió familiarizarse con la estructura de un servidor web en un microcontrolador.

El **Ejercicio 2 – Nivel intermedio** consistió en leer los valores de temperatura y humedad proporcionados por el sensor DHT11 y mostrarlos en una página HTML. Este ejercicio integró la programación de sensores digitales con la visualización de datos en tiempo real, utilizando la comunicación con el navegador. Se comprendió cómo obtener lecturas precisas del sensor y cómo enviarlas dinámicamente al cliente web, permitiendo monitorear las condiciones ambientales desde cualquier dispositivo conectado a la misma red. Además, se reforzó el conocimiento sobre la manipulación de datos y la actualización de la información en una página web generada desde Arduino.

Finalmente, el **Ejercicio 3 – Nivel avanzado** unió los conceptos previos e incorporó un actuador: un servomotor controlado desde la misma página web donde se mostraban los valores del DHT11. Esta práctica permitió integrar control de hardware y lectura de sensores en un mismo proyecto, implementando un sistema interactivo que combina visualización de datos con control remoto. Se adquirió experiencia en el manejo de señales PWM para el servo, en la interpretación de solicitudes GET desde el navegador y en la actualización dinámica de la interfaz web.

En conclusión, los tres ejercicios desarrollados demostraron cómo un microcontrolador como Arduino puede funcionar como un sistema embebido completo, capaz de **monitorear sensores, procesar datos y controlar actuadores a través de la web**. Estas prácticas brindan una comprensión sólida de los fundamentos de IoT, la programación de hardware y la comunicación en red, sentando las bases para el desarrollo de proyectos más complejos y automatizados.