

Mini Estacion Meteorológica (con sensor de luz GUVA-S12D)



Integrantes

Andy Escobar - 21611024

Iván Carcamo - 21611288

Luis Ponce - 21711234

Mario Flores - 21711252



Introducción

Con el avance de la tecnología y desarrollo de nuevos los sensores podemos realizaras diversas dispositivos que nos ayuden de una manera precisa y automatizada para obtener datos, como por ejemplo la temparatura en lugar, la condiciones de luz en ambiente, la humedad en una superficie. Y gracias a estos podemos hacer uso del internet que da paso al **IoT** (*Internet Of Things / Español: Internet de la Cosas*), ya que nos proporciona grandes ventajas como ver los datos de nuestros dispositivos en cualquier lugar en que estemos siempre y cuando obtengamos acceso a internet.

A continuación estaremos mostrando el desarrollo de un dispositivo que es capaz de detectar la intensidad de luz en un ambiente y a través de un módulo Wi-Fi en el circuito trasmitirlo a través de internet en una página web, demostrando así un ejemplo aplicable para el **Internet de las Cosas**.

◆ Objetivos

- Implementar un circuito con un **sensor de luz** usando el microcontrolador **Arduino Uno** capaz de leer la intensidad de luz que recibe.
- Implementar el **sensor de Wi-Fi** en el circuito para que este envíe la información a través de internet.
- Comprender el principio del Internet de las Cosas al enviar la información recolectada por el dispositivo a internet.



Contenido

I. ♦ Software

● Driver

Los drivers para utilizar Arduino vienen incluidos en el IDE de [Arduino](#)

● IDE

Se utilizó el [Arduino IDE](#)

Arduino IDE:

Es una aplicación multiplataforma que se utiliza para escribir y cargar programas en placas compatibles con Arduino. Admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos.

● Diseño

Para el diseño o prototipo digital del dispositivo se utilizó la herramienta [Fritzing-Portable](#)

● Sensor

Para el sensor de luz, no se utilizó una libreria.

● Pantalla (Display LCD)

La libreria usada para el display fue [Liquid Crystal](#)

● Internet (Wifi)

El arduino WeMos D1 ya viene integrado con un módulo WiFi. La libreria a usar es [ESP8266WIFI](#)

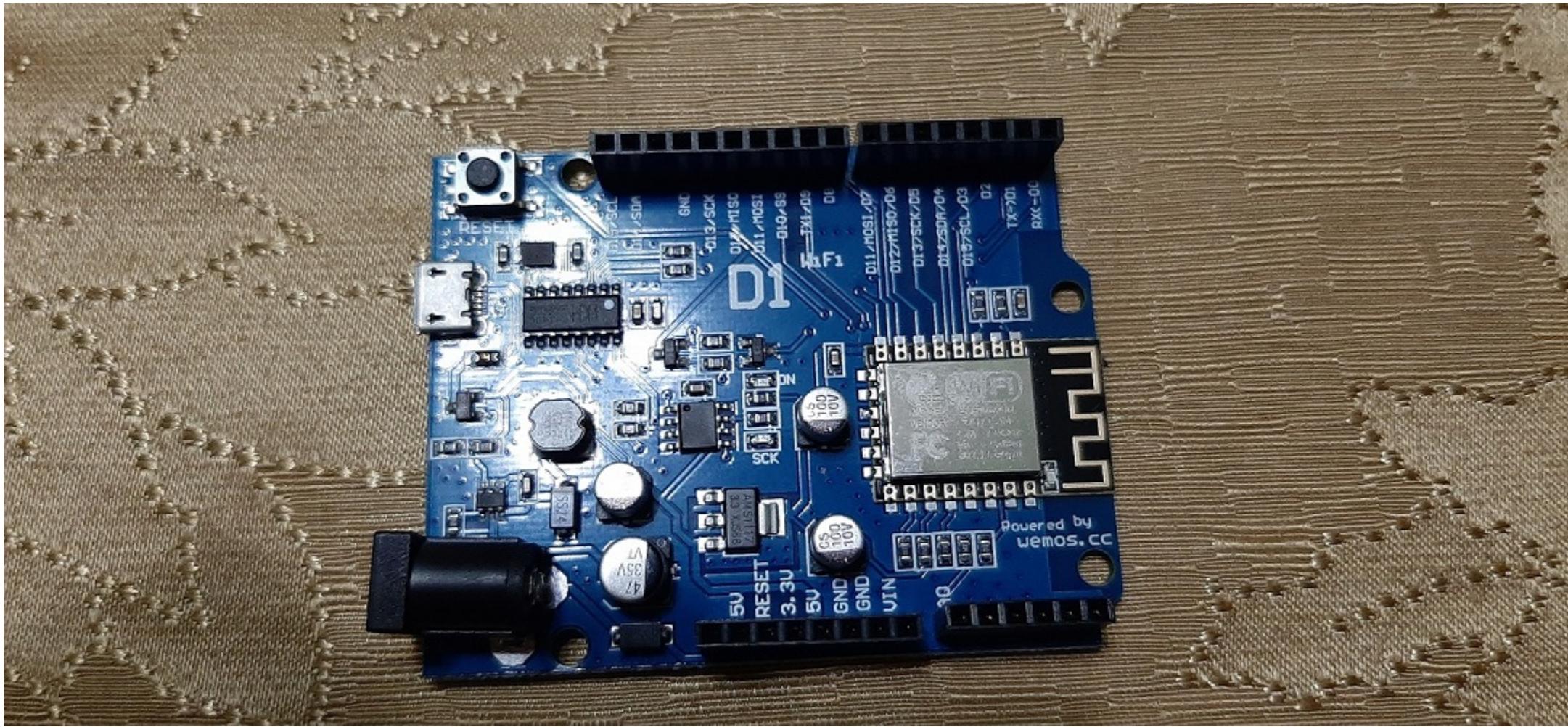
II. ♦ Hardware

● Microcontrolador

Microcontrolador:

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos.

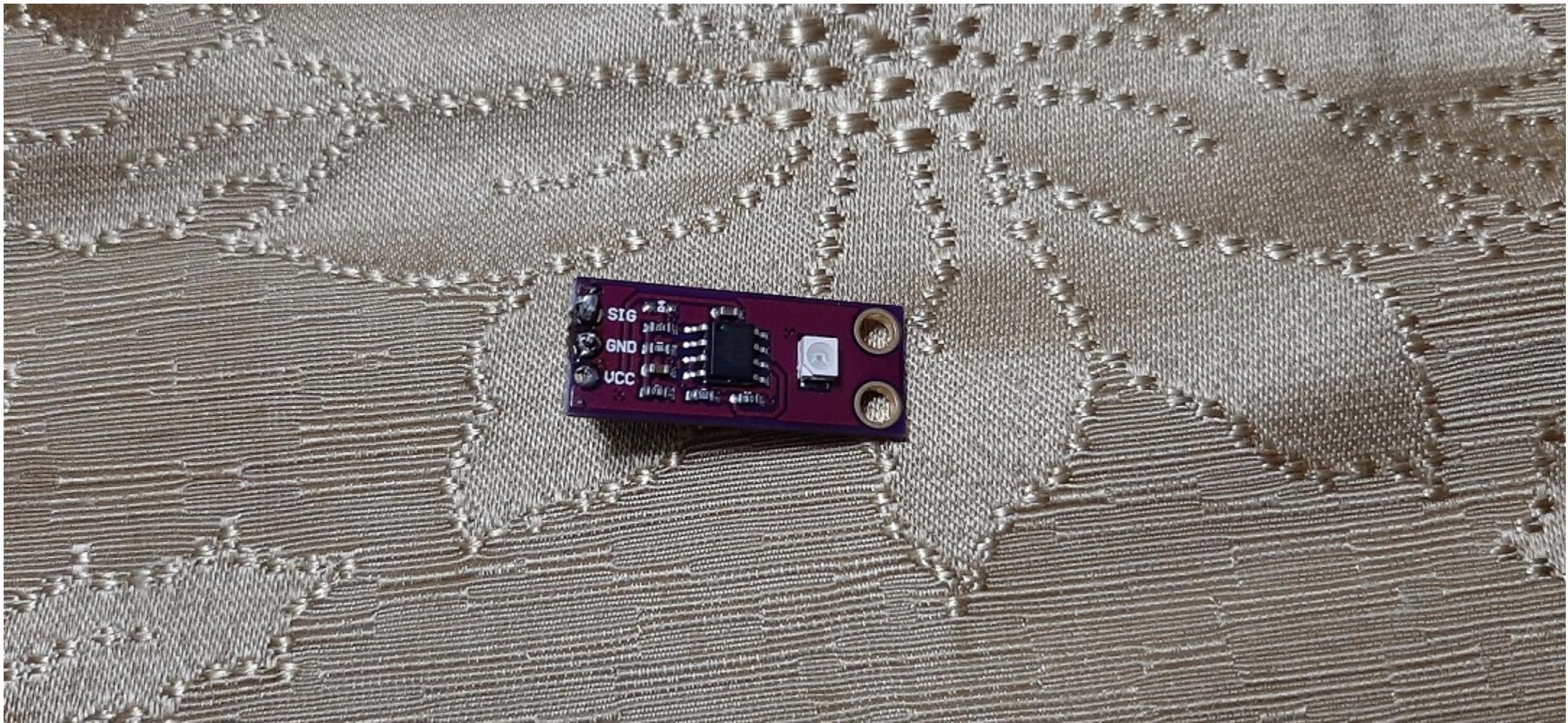
Arduino WeMos D1:



Se puede conseguir en [Amazon](#)

Sensor

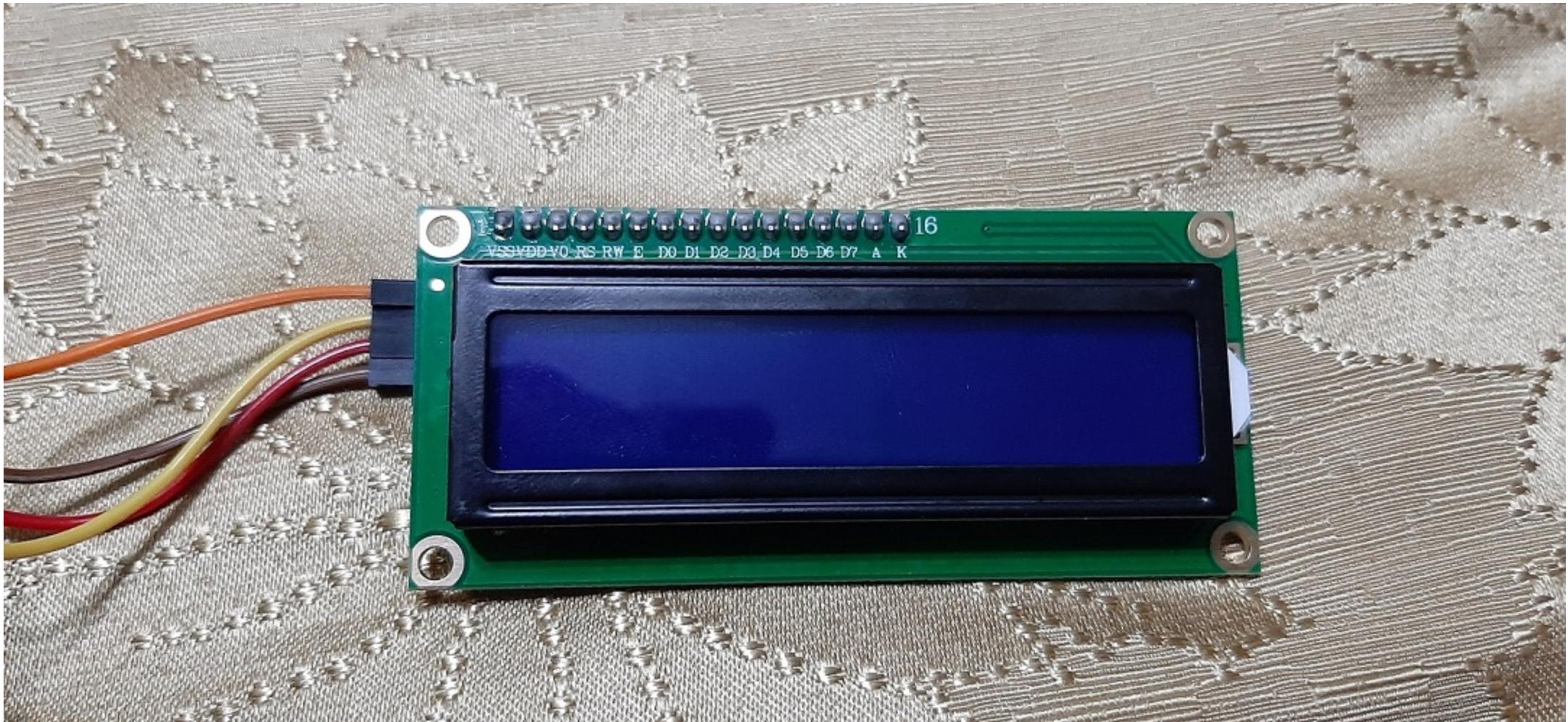
Sensor UV S12SD

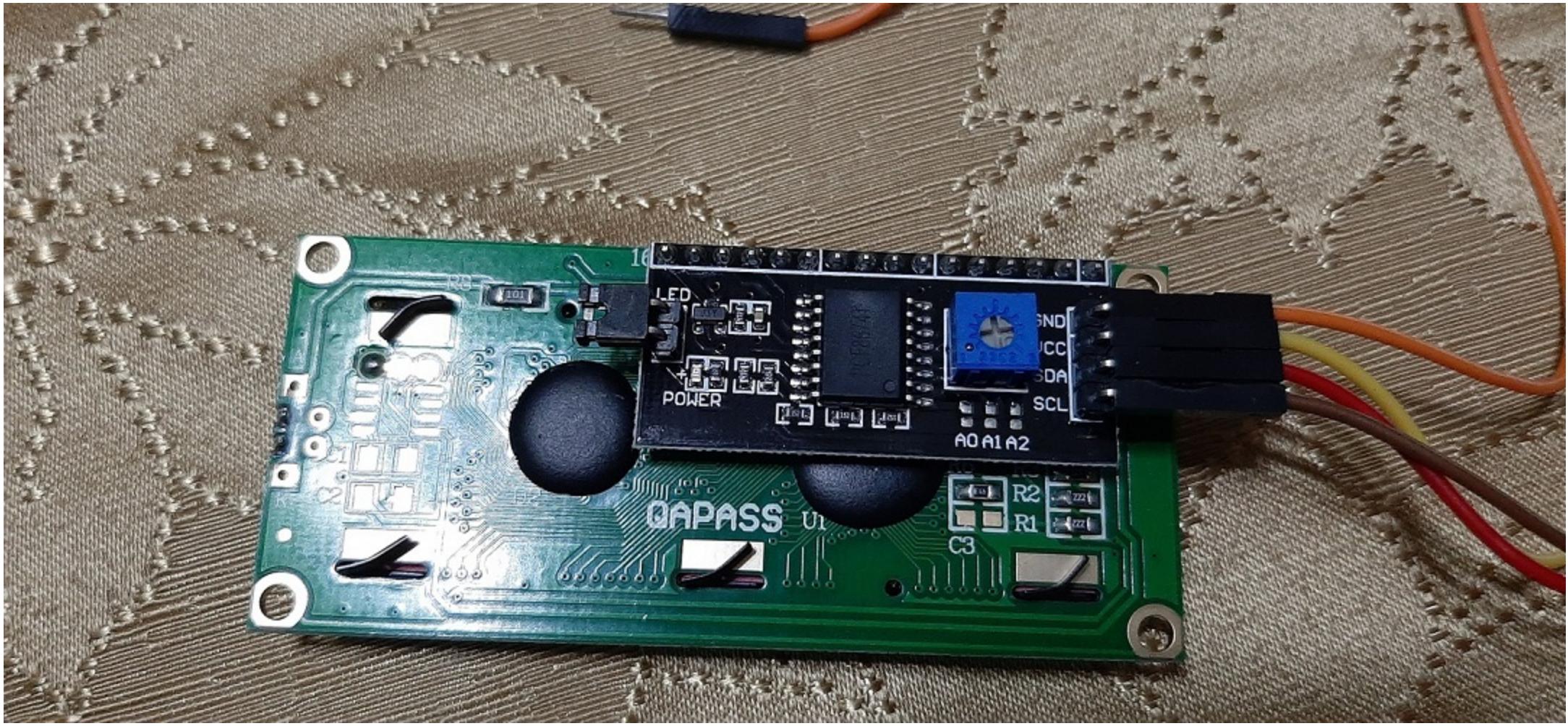


Comprado en [Think Making](#)

● Pantalla (Display LCD)

LCD 1602 + I2C

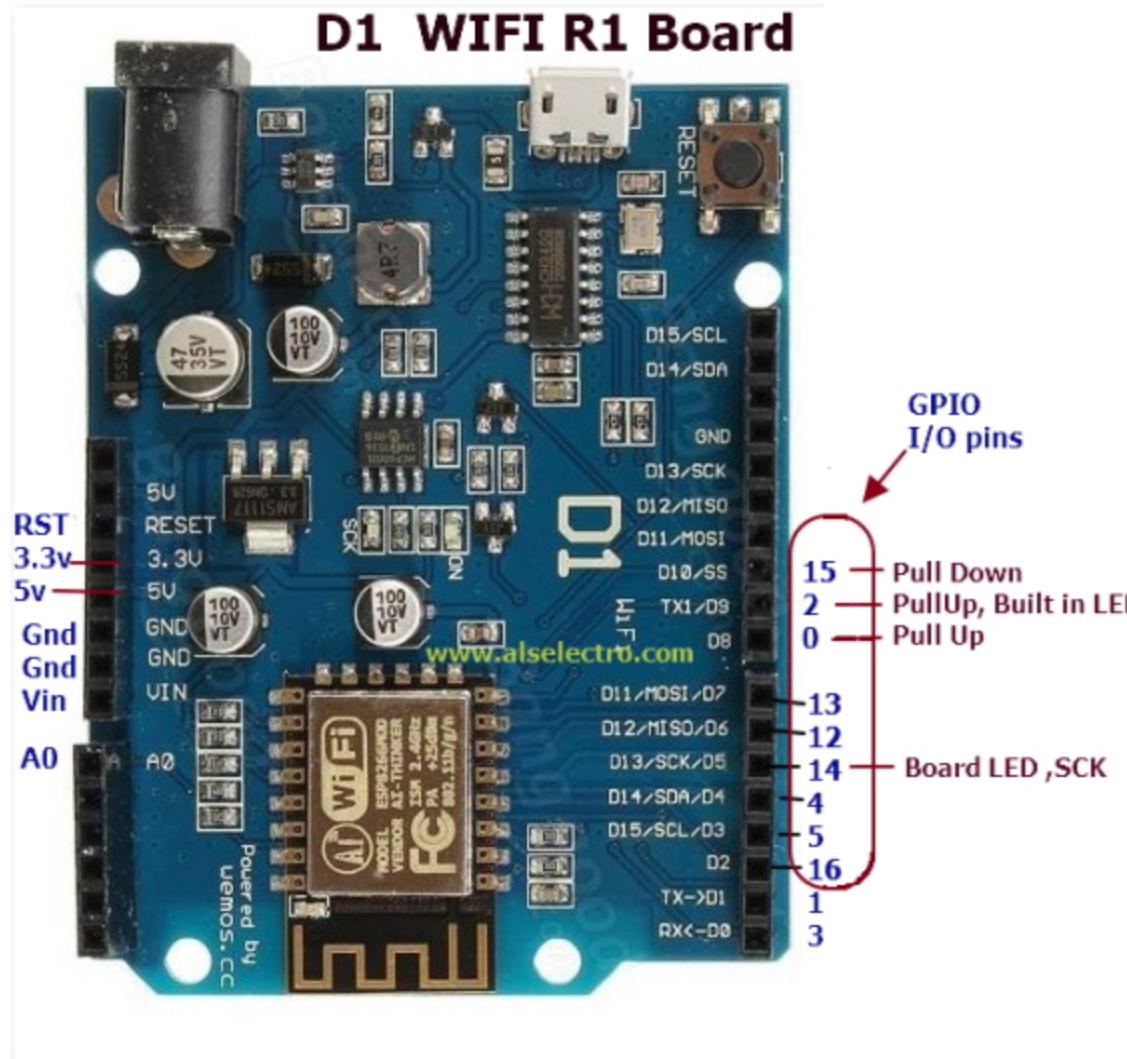




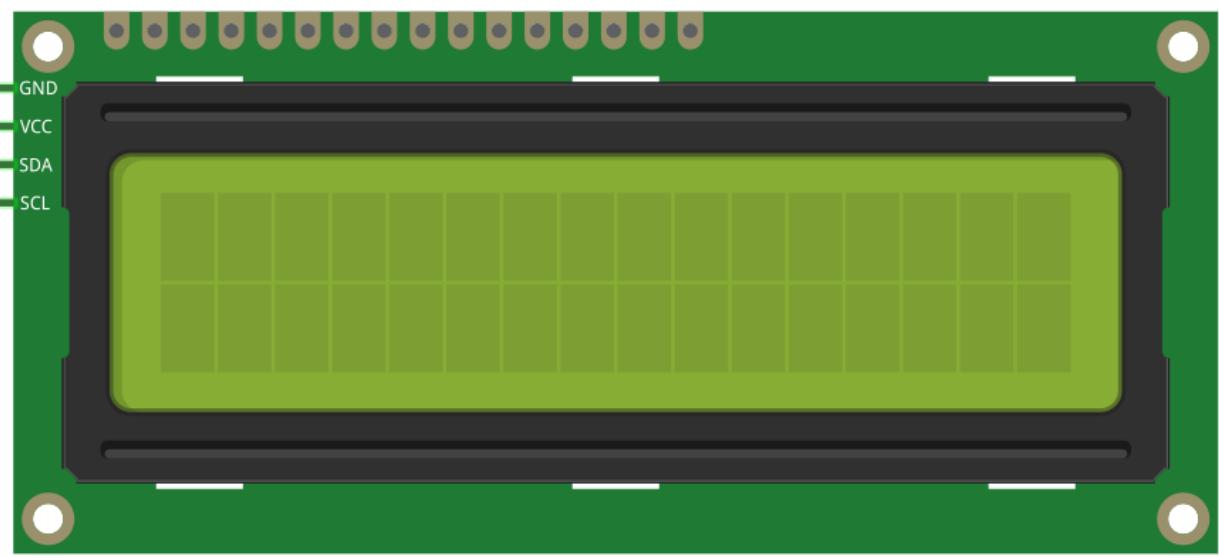
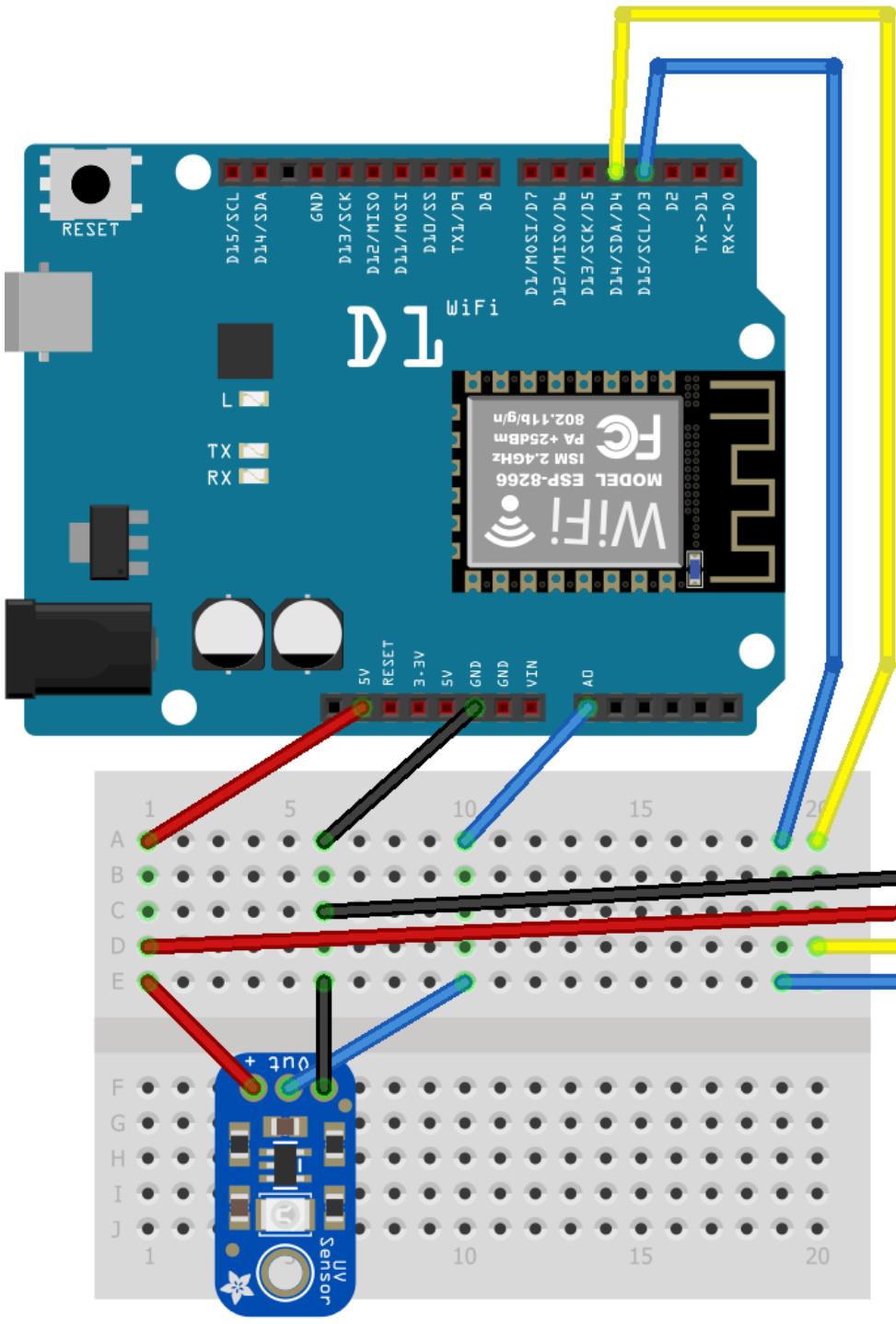
Comprado en [Think Making](#)

III. ♦ Diseño

● Pinout de Microcontrolador / Mapeo de pines



Arduino	LCD	Sensor
D14/SDA/D4	SDA	
D15/SCL/D3	SCL	\$12
5V	VCC	VCC
GND	GND	GND
A0		OUT



IV. ♦ Desarrollo

● Instalación de Tarjeta (Arduino)

En nuestro caso, no hubo necesidad de instalar una placa extra a nuestro arduino, solamente los componentes previamente listados (LCD, sensor)

● Instalación de Libreria(s)

◆ Sensor

No fue requerida una libreria para el sensor, el sensor simplemente enviaba una señal de entrada analoga a nuestro arduino

◆ Pantalla (Display LCD)

Para poder utilizar el LCD, fue necesario descargar una libreria afuera del arduino IDE. Se siguieron los mismos ***pasos de instalacion de la pagina oficial del creador de la libreria "Arduino-LiquidCrystal-I2C-library"***, la cual terminamos integrando a nuestro proyecto.



LiquidCrystal I2C

A library for I2C LCD displays.

Author	Frank de Brabander
Maintainer	Marco Schwartz
Website	https://github.com/marcoschwartz/LiquidCrystal_I2C
Category	Display
License	Unknown
Library Type	Contributed
Architectures	avr

The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.

Downloads

Filename	Release Date	File Size
LiquidCrystal_I2C-1.1.2.zip	2015-11-29	20.32 KiB

◆ Internet (Wifi)

Para la instalacion de las librerias del Wifi (libreria ***ESP8266HTTPClient*** y ***ESP8266WiFi***), simplemente fue cuestion de añadirla al proyecto ya que estas dos librerias ya vienen incluidas en las librerias que vienen con el arduino IDE.

Código de proyecto

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <LiquidCrystal_I2C.h>

int sensorPin = A0; // Pin de entrada del sensor
int sensorValue = 0; // Valor que captura el sensor
const char *port = "82"; // Puerto en el que se encuentra el default gateway
const char *url = "http://jsonplaceholder.typicode.com/users";
const char *id = "AndroidP0029"; // nombre de la red
const char *pass = "holal12345"; // password

int contador = 0; // verifica si tarda mucho en conectarse a la red
LiquidCrystal_I2C lcd(0x27,16,2); // si no te sale con esta direccion puedes usar (0x3f,16,2) || (0x27,16,2) ||(0x20,16,2)

void setup() {
  Serial.begin(115200);
  inicializarLCD();
  inicializarWifi();
  lcd.init();
  EscribirLCDfila1("Conexion");
  EscribirLCDfila2("Establecida");
  delay(3000);
  pinMode(sensorPin, INPUT);
}

void loop() {
  // toma el valor del sensor y lo asigna a sensorValue
  lcd.clear();
  sensorValue = analogRead(sensorPin);

  // muestra en el LCD
  EscribirLCDfila1("Lumenes: " + String(sensorValue));
  delay(3000); // espera 3 segundos hasta que haga el post
  lcd.clear();

  // hace el post
  hacerPost(String(sensorValue));
  lcd.clear();
  EscribirLCDfila1("Esperando...");
  EscribirLCDfila2("Segundos... ");
  delay(10000); // 10 segundos para hacer otro post
}

void EscribirLCDfila1(String texto){
  lcd.setCursor(0,0);
  lcd.print(texto);
  delay(100);
}

void EscribirLCDfila2(String texto){
  lcd.setCursor(0,1);
  lcd.print(texto);
  delay(100);
}

void inicializarLCD(){
  lcd.init();
  lcd.backlight();
}

void inicializarWifi(){
  WiFi.mode(WIFI_OFF);
  WiFi.begin(id,pass);
  WiFi.mode(WIFI_STA);

  while(WiFi.status() != WL_CONNECTED){
    delay(1000);
    EscribirLCDfila1("Esperando");
    EscribirLCDfila2("Conexion");
    lcd.clear();
    contador++;
    if(contador >10){
      break;
    }
  }
}

void hacerPost(String sensor){
  if(WiFi.status()== WL_CONNECTED){
    delay(1000);
    HTTPClient http;
    http.begin(post);
    http.setUrl(url);
    http.addHeader("Content-Type", "text/plain");
    int httpCode = http.POST(sensor);
    lcd.clear();
    EscribirLCDfila1("Respueta: ");
    if(httpCode == 201){ EscribirLCDfila2("OK");}else{EscribirLCDfila2("BAD REQ");}
    delay(3000);
    http.end();
  }
  while(WiFi.status()!= WL_CONNECTED){
    lcd.clear();
    EscribirLCDfila1("Esperando");
    EscribirLCDfila2("Conexion");
    delay(100);
  }
}
```

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <LiquidCrystal_I2C.h>

int sensorPin = A0;      // Pin de entrada del sensor
int sensorValue = 0;     // valor que capta el sensor
const char *port = "192.168.1.1"; // ip del default gateway
const char* url = "http://jsonplaceholder.typicode.com/users";
const char* id = "AndroidAP0D29"; // nombre de la red
const char* pass = "hola12345"; // password

int contador = 0; // verifica si tarda mucho en conectarse a la red
LiquidCrystal_I2C lcd(0x27,16,2); // si no te sale con esta direccion puedes usar (0x3f,16,2) || (0x27,16,2) ||(0x20,16,2)

void setup() {
  Serial.begin(115200);
  inicializarLCD();
  inicializarWifi();
  lcd.clear();
  EscribirLcdFila1("Conexion");
  EscribirLcdFila2("Establecida");
  delay(3000);
  pinMode(sensorPin, INPUT);
}

void loop() {
  // toma el valor del sensor y lo asigna a sensorValue
  lcd.clear();
  sensorValue = analogRead(sensorPin);

  // lo escribe en el LCD
  EscribirLcdFila1("Lumenes: "+ String(sensorValue));
  delay(3000); // espera 3 segundos hasta que haga el post
  lcd.clear();

  // hace el post
  hacerPost(String(sensorValue));
  lcd.clear();
  EscribirLcdFila1("Espere 10");
  EscribirLcdFila2("Segundos...");
  delay(10000); // 10 segundos para hacer otro post
}
```

```
void EscribirLcdFila1(String texto){  
    lcd.setCursor(0,0);  
    lcd.print(texto);  
    delay(100);  
}  
  
void EscribirLcdFila2(String texto){  
    lcd.setCursor(0,1);  
    lcd.print(texto);  
    delay(100);  
}  
  
void inicializarLCD(){  
    lcd.init();  
    lcd.backlight();  
}
```

```
void inicializarWifi(){
    WiFi.mode(WIFI_OFF);
    WiFi.begin(id, pass);
    WiFi.mode(WIFI_STA);

    while((WiFi.status() != WL_CONNECTED)){
        delay(1000); // 1 segundo
        EscribirLcdFila1("Esperando");
        EscribirLcdFila2("Conexion");
        lcd.clear();
        contador++;

        if(contador >=10){
            break;
        }
    }
}
```

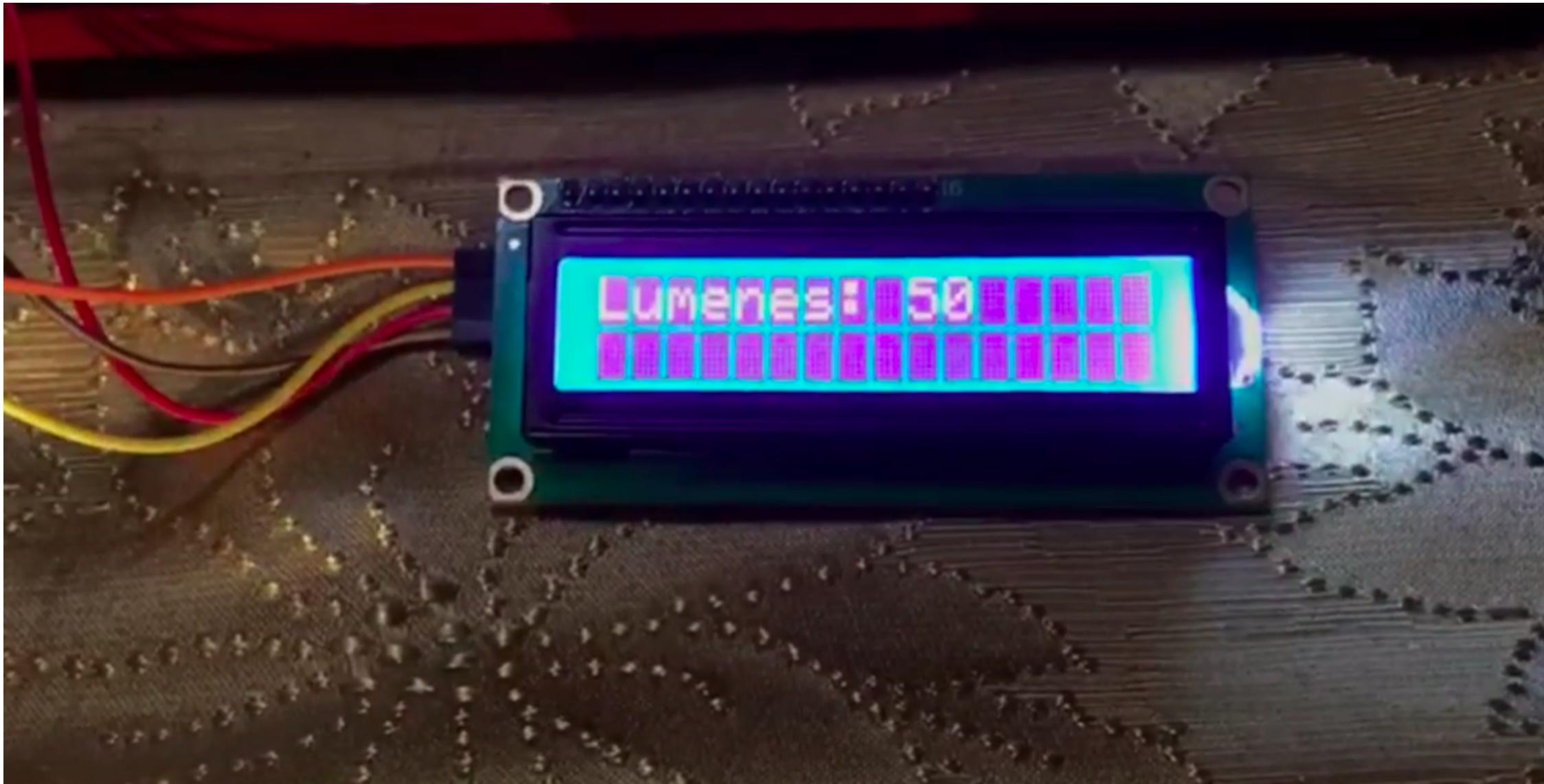
```
void hacerPost(String sensor){
    if(WiFi.status() == WL_CONNECTED){
        delay(1000);
        HTTPClient http;
        http.begin(port);
        http.begin(url);
        http.addHeader("Content-Type", "text/plain");
        int httpCode = http.POST(sensor);
        lcd.clear();
        EscribirLcdFila1("Respuesta: ");
        if(httpCode == 201){ EscribirLcdFila2("OK"); }else{EscribirLcdFila2("BAD REQ");}
        delay(3000);
        http.end();
    }else{
        while(WiFi.status() != WL_CONNECTED){
            lcd.clear();
            EscribirLcdFila1("Esperando");
            EscribirLcdFila2("Conexion");
            delay(100);
        }
    }
}
```

V. ♦ Funcionalidad

● Demostración (Video)

 [Video Demostrativo](#)

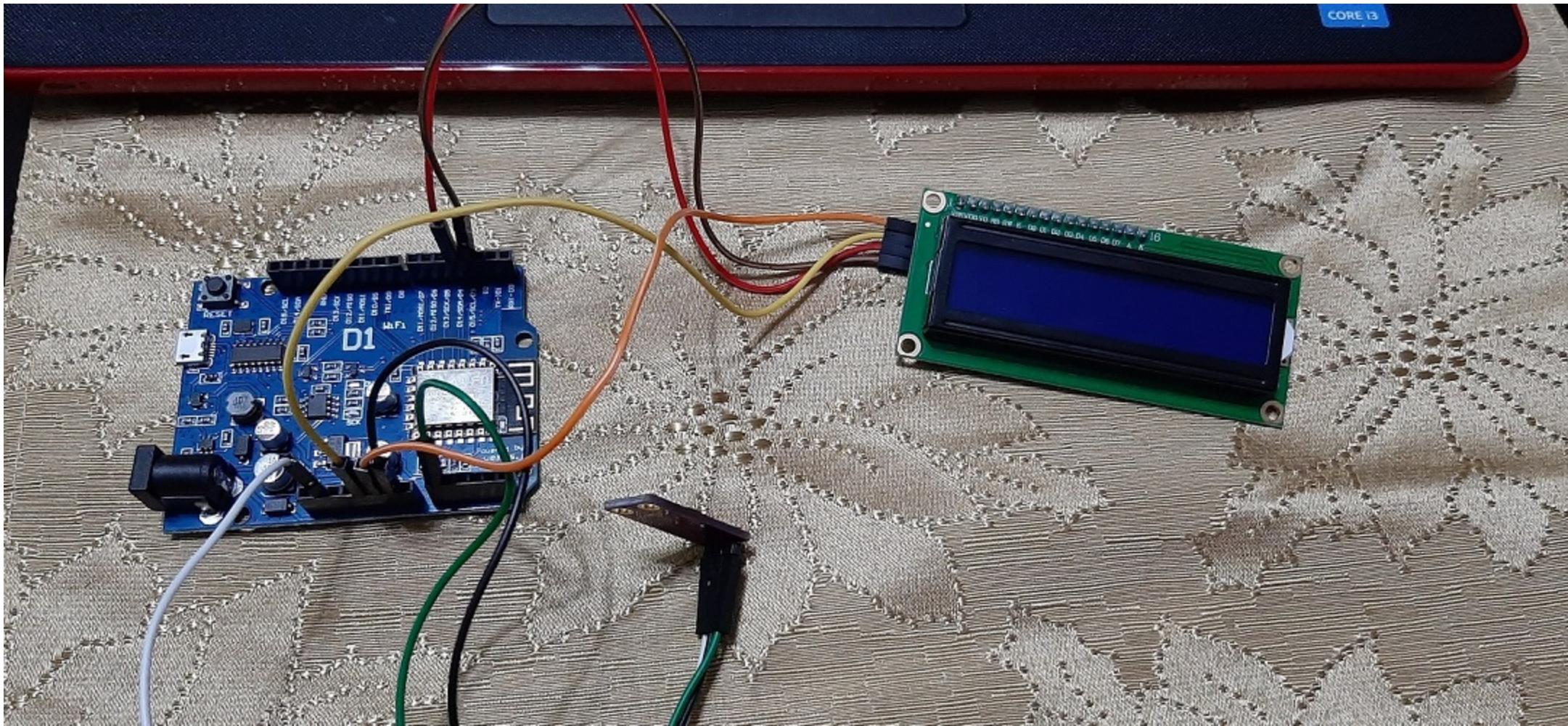
● Despliegue en pantalla



● Envío a servidor



● Todos los componentes del proyecto





◆ Conclusiones

- ✓ El Internet de las Cosas se plantea como una tecnología de enorme utilidad a la hora de mejorar la eficiencia de los dispositivos.
- ✓ Aprendimos que con Arduino podemos crear un dispositivo que funcione con los principios del Internet de la Cosas
- ✓ Se implementó un circuito capaz de realizar un dispositivo capaz de leer la intensidad de luz en un ambiente.

- ✓ Se logró extraer los datos recolectados por el sensor y enviarlos a través de internet y que estos sean proyectados en un sitio web.
- ✓ Aprendimos el uso de un arduino y las infinitas cosas que un microcontrolador puede proporcionar para llevar a cabo la innovación, la creatividad, entre muchas otras cosas.
- ✓ Aprendimos a aprovechar las librerías que ofrece Arduino y también a la instalación de librerías de terceros para llevar a cabo la integración de componentes que nos ayudaron a desarrollar el proyecto

◆ Referencias

 [circuits4you](#)

 [Arduino](#)

 [ESP8266 Learning](#)

 [ESP8266 Arduino Core](#)