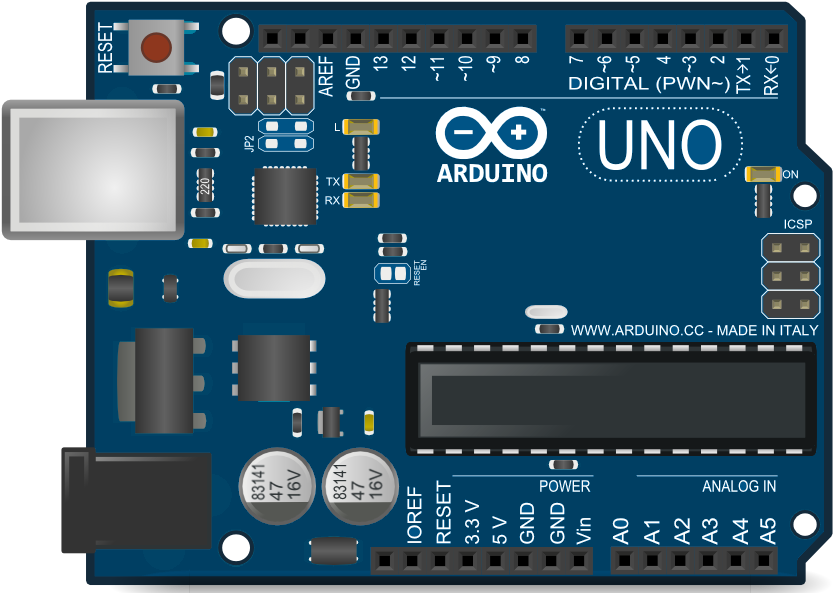
# REPORTES DE PRACTICAS ARDUINO



GRECIA GENESIS ROBLES PICAZO

ARDUINO M.C LINDA MARTINEZ

GRUPO 3 SISTEMAS

**Practica 7: Sensor infrarrojo**

**Concepto:**

En esta práctica, usaremos un sensor infrarrojo para detectar la presencia de objetos y visualizar el estado de detección en el monitor serial del microcontrolador.

**Materiales:**

1 Sensor infrarrojo (como un TCRT5000 o similar)

1 Resistencia (1kΩ o según especificaciones del sensor)

1 Protoboard

Cables

Microcontrolador (Arduino, ESP32, etc.)

**Codigo:**

int infrarrojo=8;

int valor=0;

void setup(){

Serial.begin(9600);

pinMode(infrarrojo, INPUT);

}

void loop(){

valor = digitalRead(infrarrojo);

if (valor == HIGH) {

Serial.println("no hay obstaculo");

}

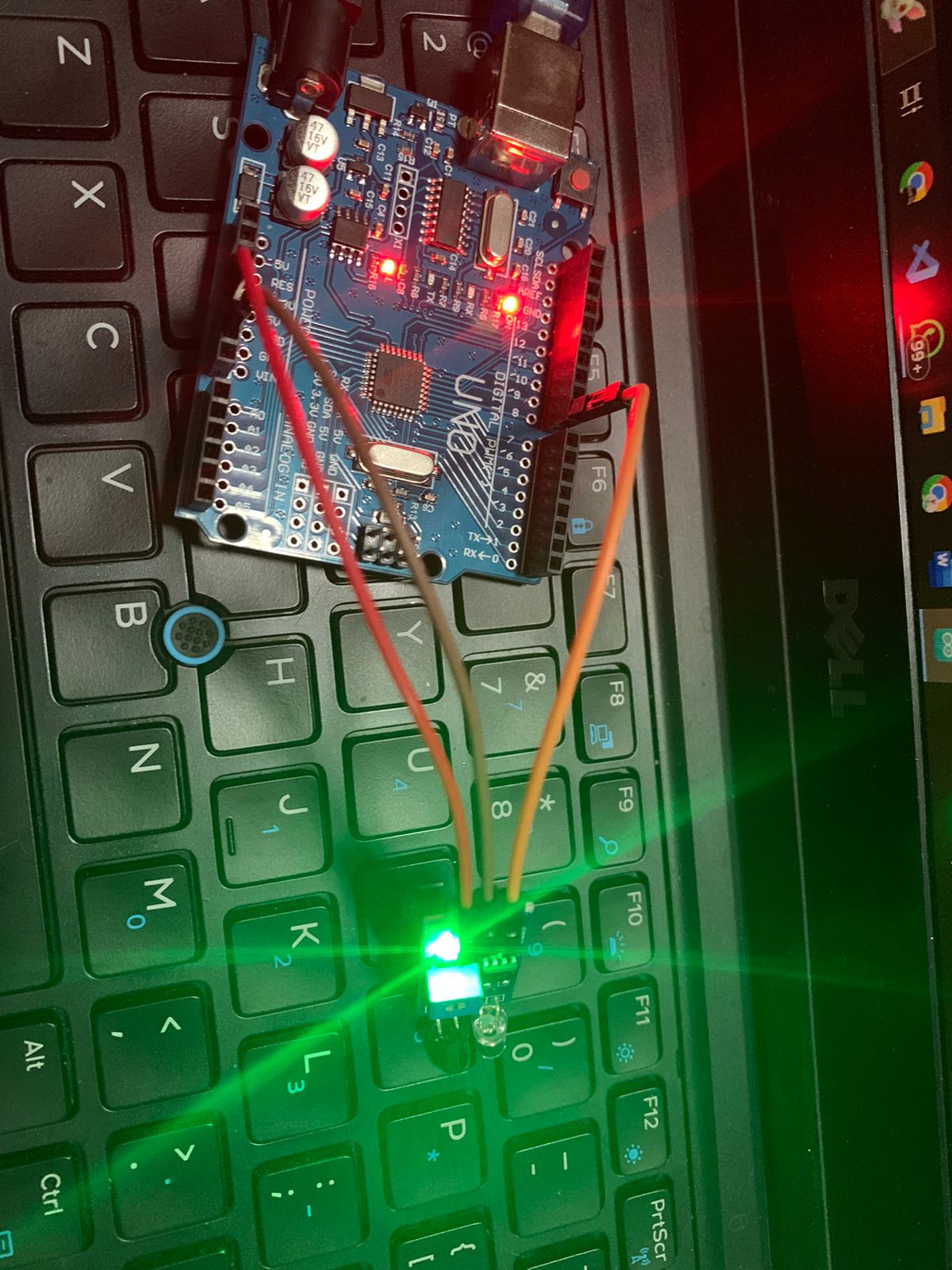
else{

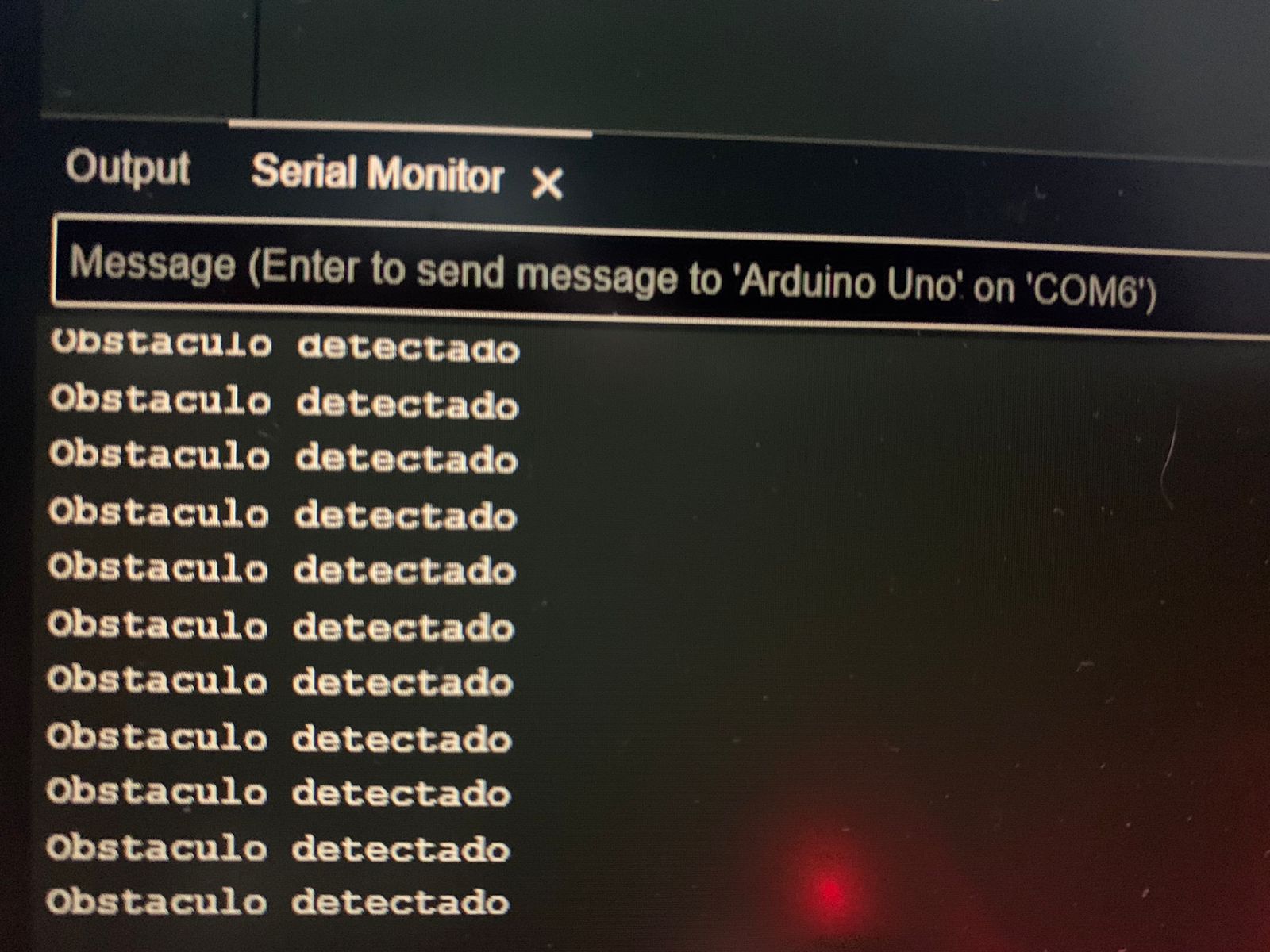
Serial.println("obstaculo detectado");

}

delay(500);

}





**Practica 8: Sensor Ultrasónico**

**Concepto:**

Esta práctica tiene como objetivo medir la distancia de un objeto utilizando un sensor ultrasónico y mostrar los resultados en el monitor serial.

**Materiales:**

1 Sensor ultrasónico

1 Protoboard

Cables

Microcontrolador (Arduino, ESP32, etc.)

**Código:**

int TRIG = 10;

int ECO = 9;

int DURACION;

int DISTANCIA;

void setup()

{

Serial.begin(9600);

pinMode(TRIG, OUTPUT);

pinMode(ECO, INPUT);

}

void accionUltra(int trigPin, int ecoPin) {

digitalWrite(trigPin, HIGH);

digitalWrite(trigPin, LOW);

DURACION = pulseIn(ecoPin, HIGH);

DISTANCIA = DURACION / 58.2;

Serial.println(DISTANCIA);

delay(200);

}

void loop()

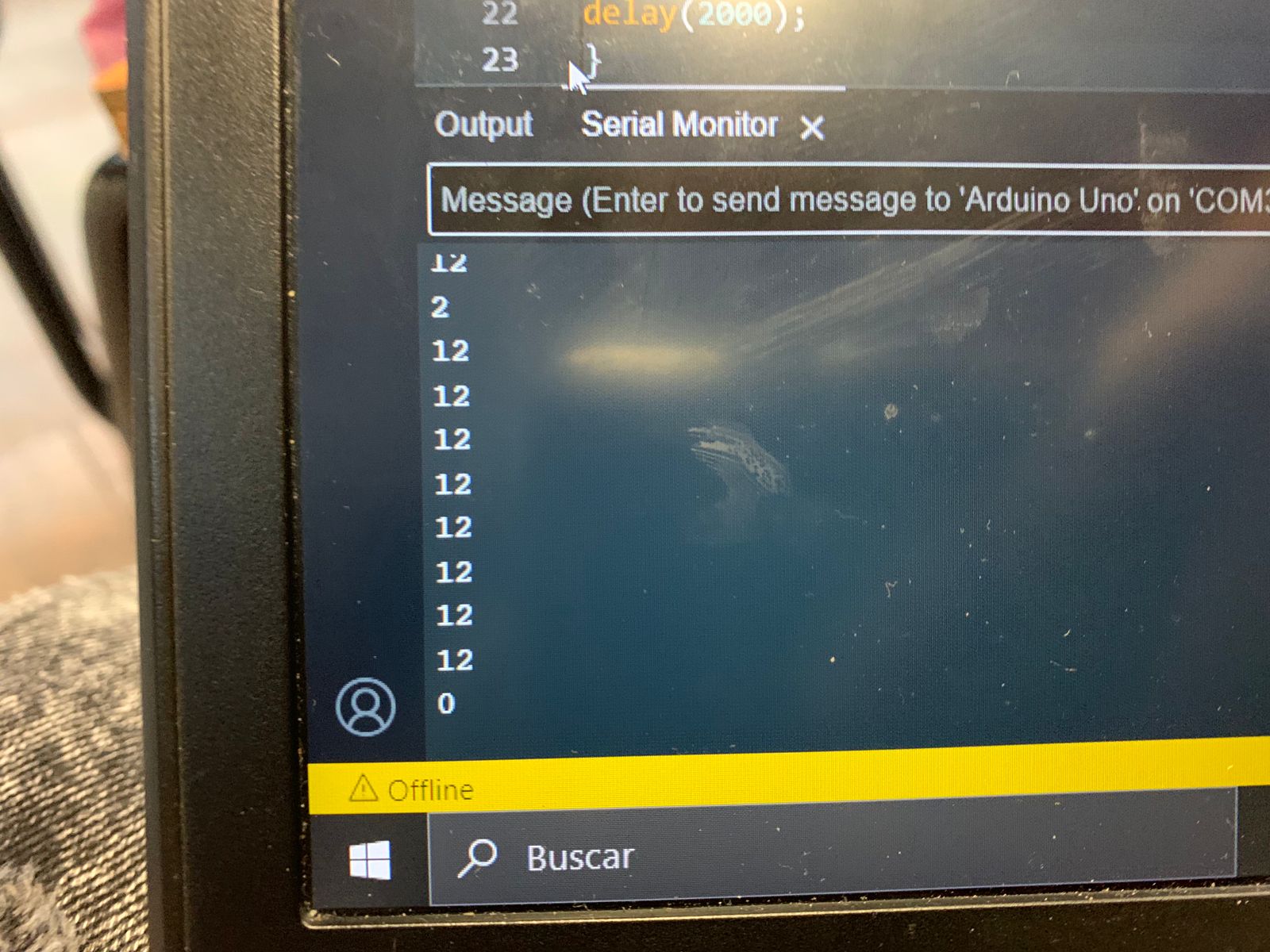
{

accionUltra(TRIG, ECO);

delay(2000);

**}**





**Practica 9: Encender/apagar un LED por medio del monitor serial**

**Concepto:**

En esta práctica, aprenderemos a controlar el encendido y apagado de un LED desde el monitor serial del microcontrolador.

**Materiales:**

1 LED

1 Resistencia (220Ω)

1 Protoboard

Cables

Microcontrolador (Arduino, ESP32, etc.)

**Código:**

// Configuración del pin del LED

const int ledPin = 9;

void setup() {

// Configuramos el pin del LED como salida

pinMode(ledPin, OUTPUT);

// Iniciamos la comunicación serial

Serial.begin(9600);

// Mensaje de inicio

Serial.println("Escribe 'ON' para encender el LED o 'OFF' para apagarlo.");

}

void loop() {

// Verificamos si hay datos disponibles en el Monitor Serial

if (Serial.available() > 0) {

// Leemos la entrada del usuario

String comando = Serial.readStringUntil('\n');

comando.trim(); // Eliminamos espacios o saltos de línea

// Verificamos el comando recibido

if (comando.equalsIgnoreCase("ON")) {

digitalWrite(ledPin, HIGH); // Encendemos el LED

Serial.println("LED encendido.");

} else if (comando.equalsIgnoreCase("OFF")) {

digitalWrite(ledPin, LOW); // Apagamos el LED

Serial.println("LED apagado.");

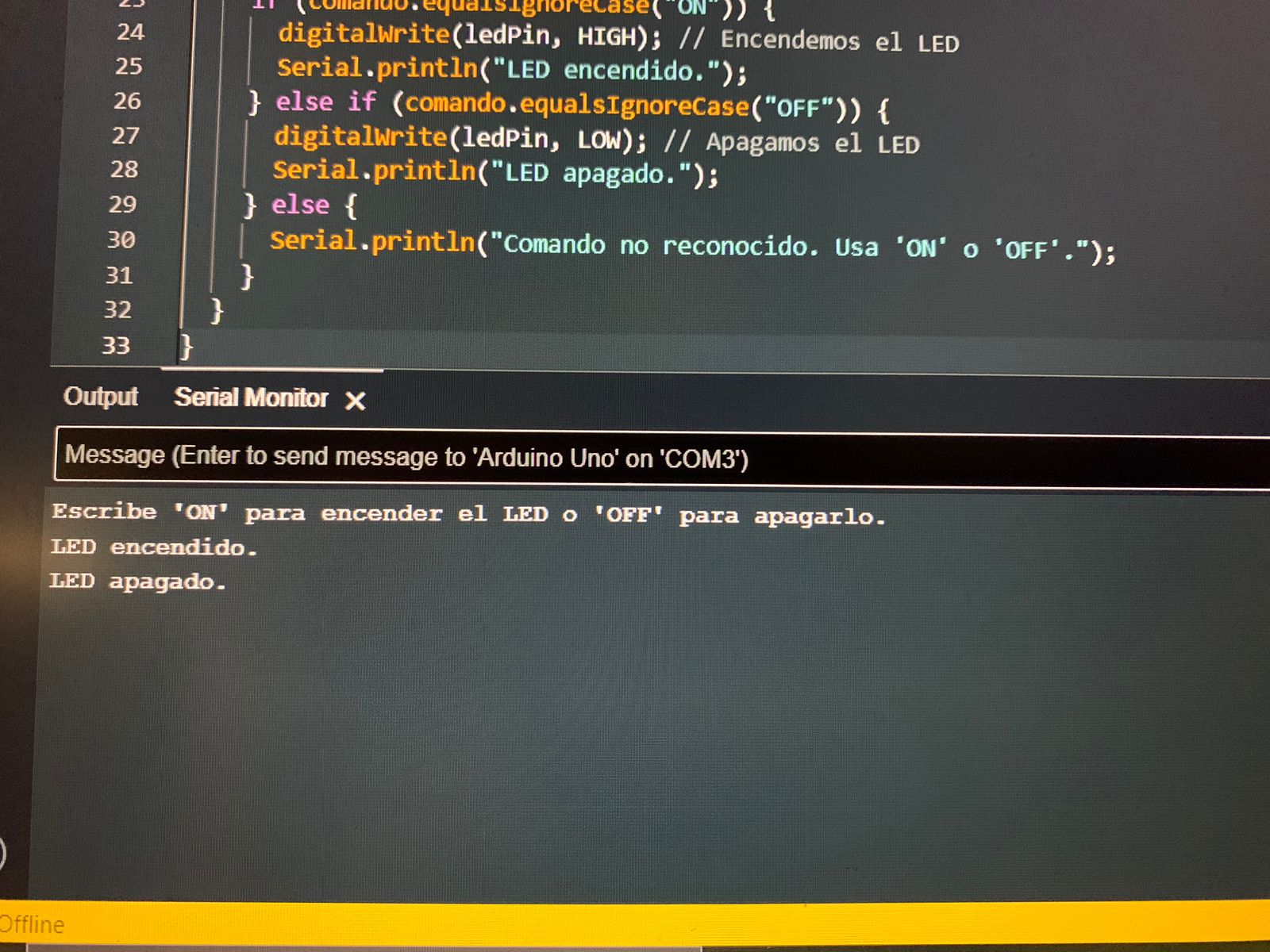
} else {

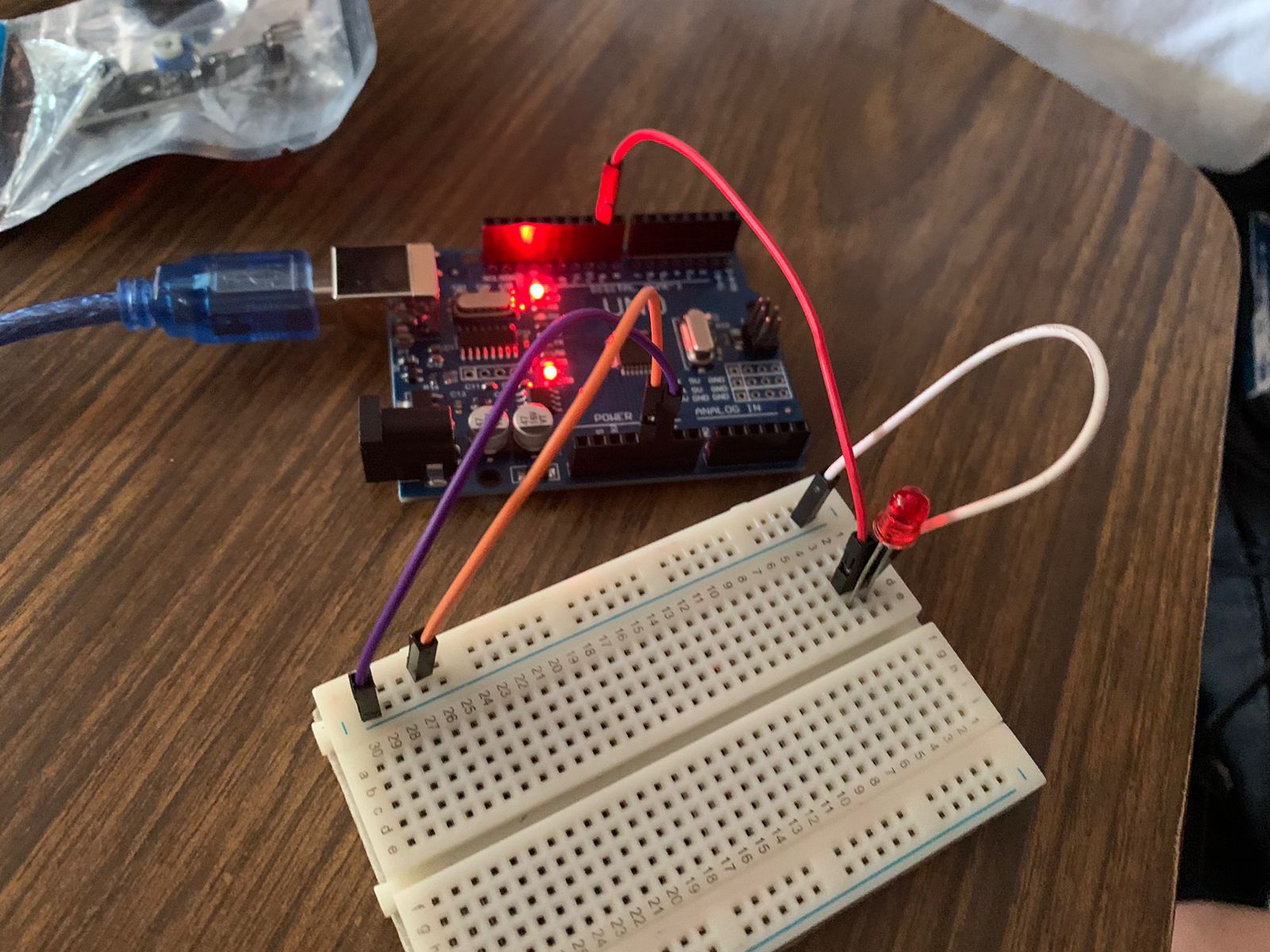
Serial.println("Comando no reconocido. Usa 'ON' o 'OFF'.");

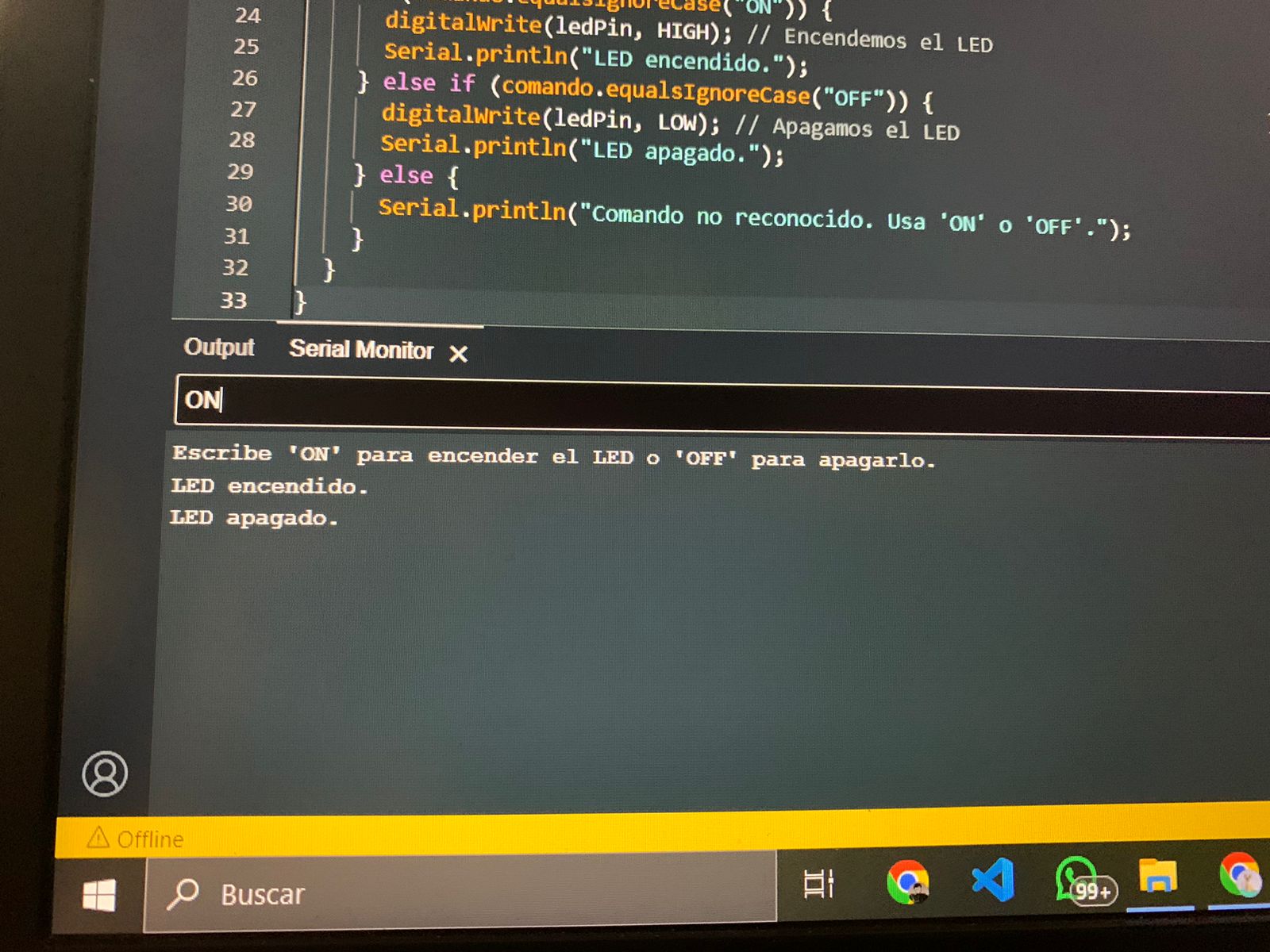
}

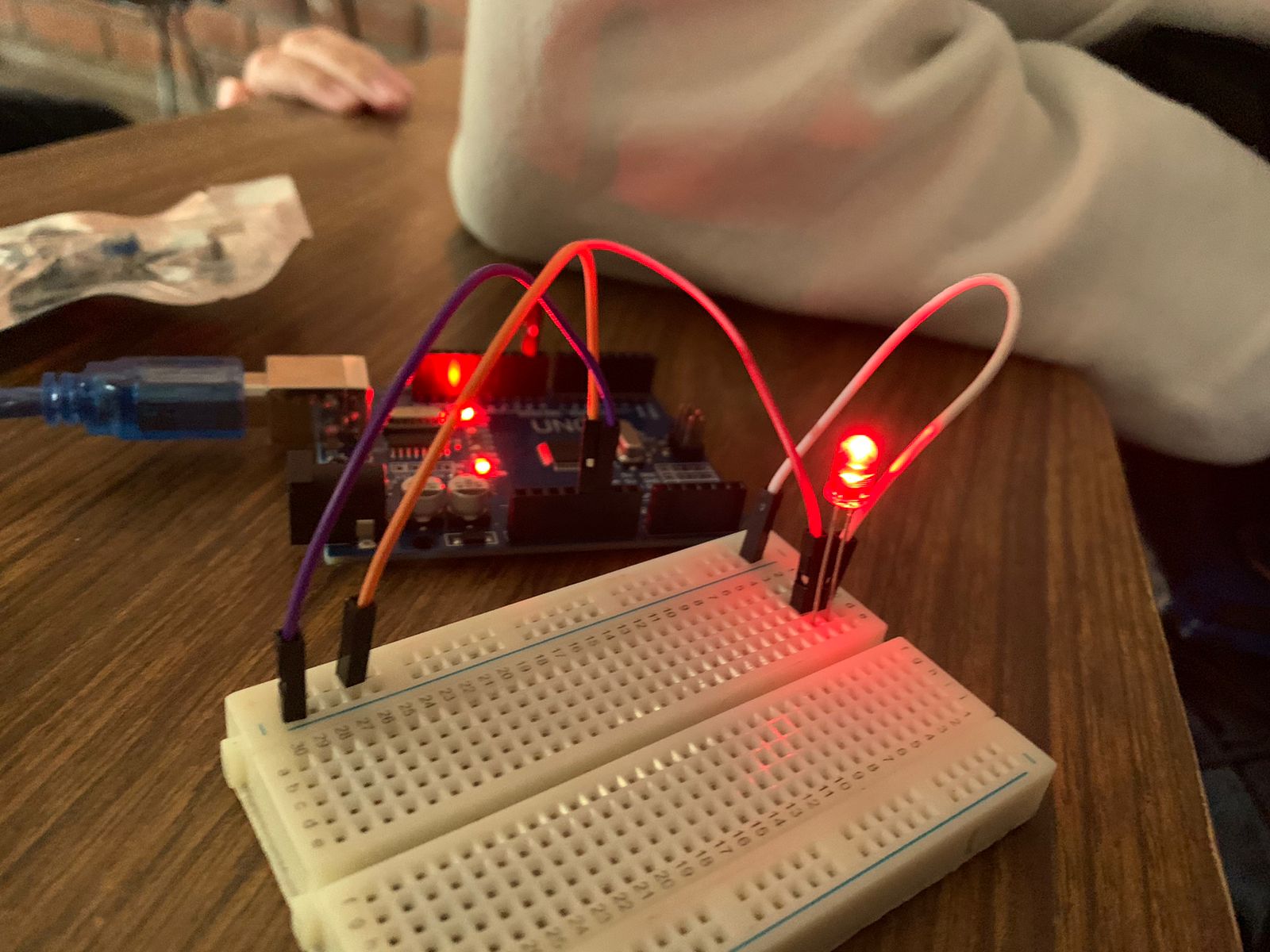
}

}









**Práctica 10: Aumentar/disminuir el brillo de un LED por medio del monitor serial**

**Concepto:**

Esta práctica nos permitirá controlar el brillo de un LED mediante comandos enviados desde el monitor serial, utilizando el principio de modulación por ancho de pulso (PWM).

**Materiales:**

1 LED

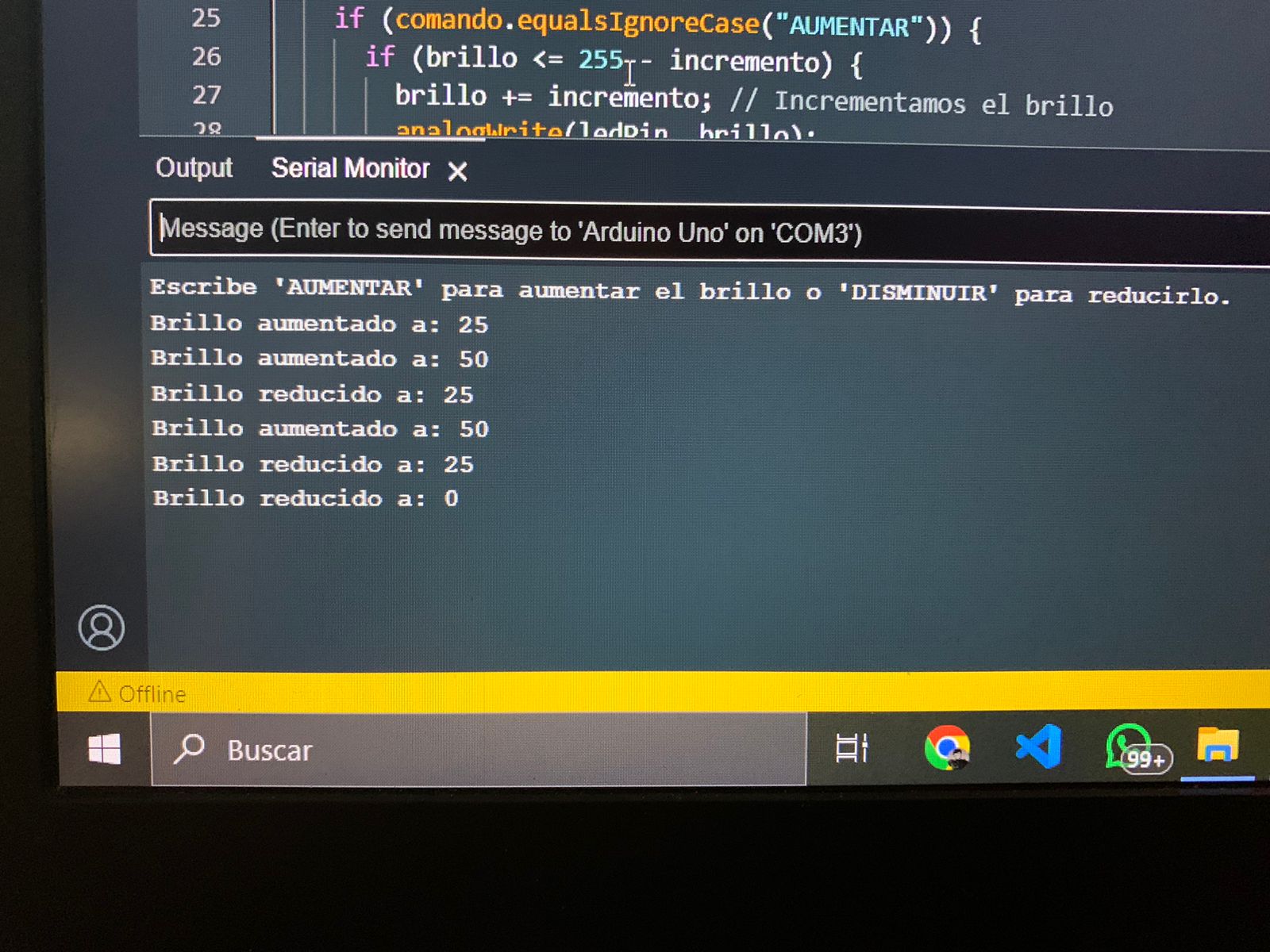
1 Resistencia (220Ω)

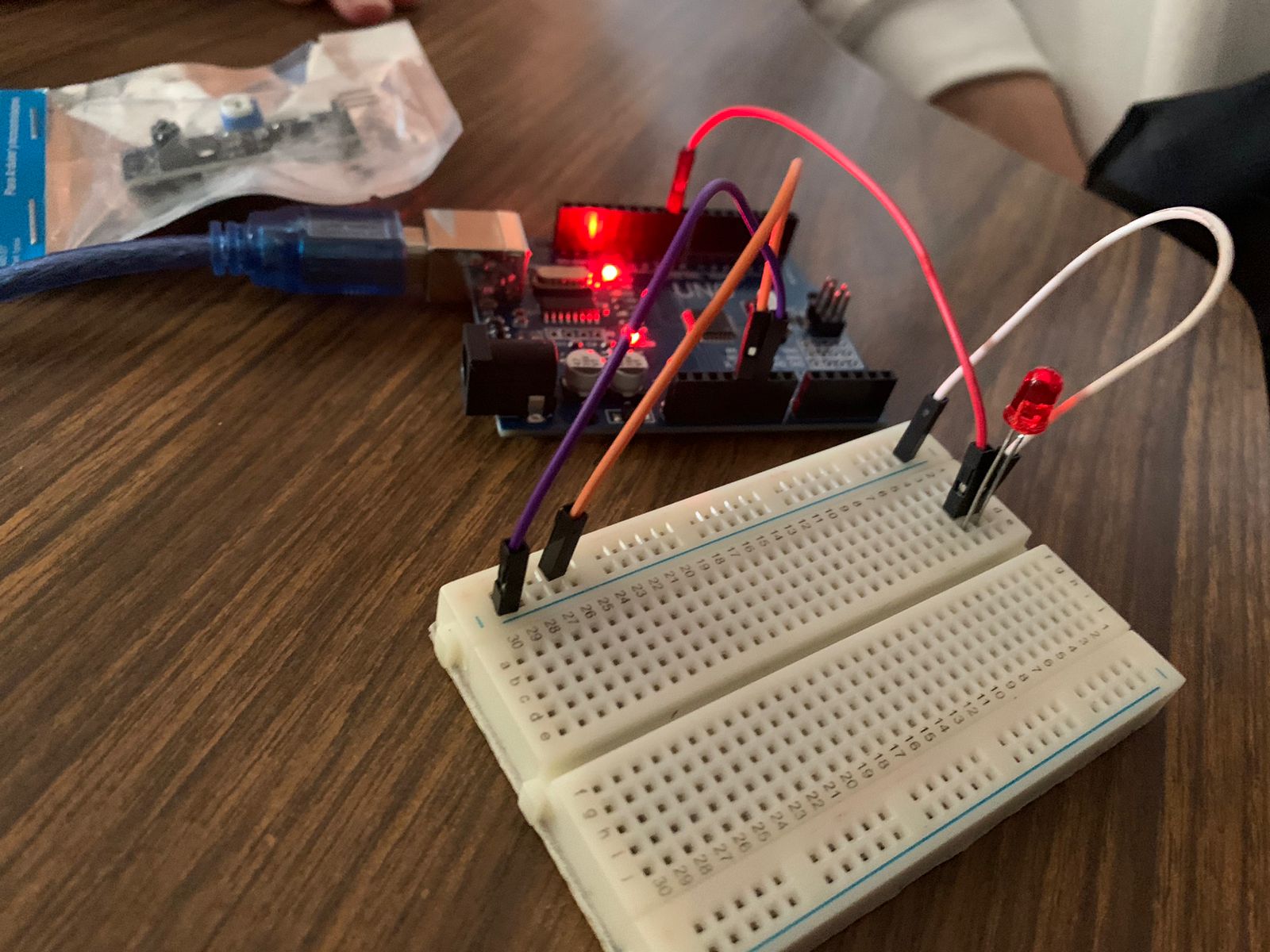
1 Protoboard

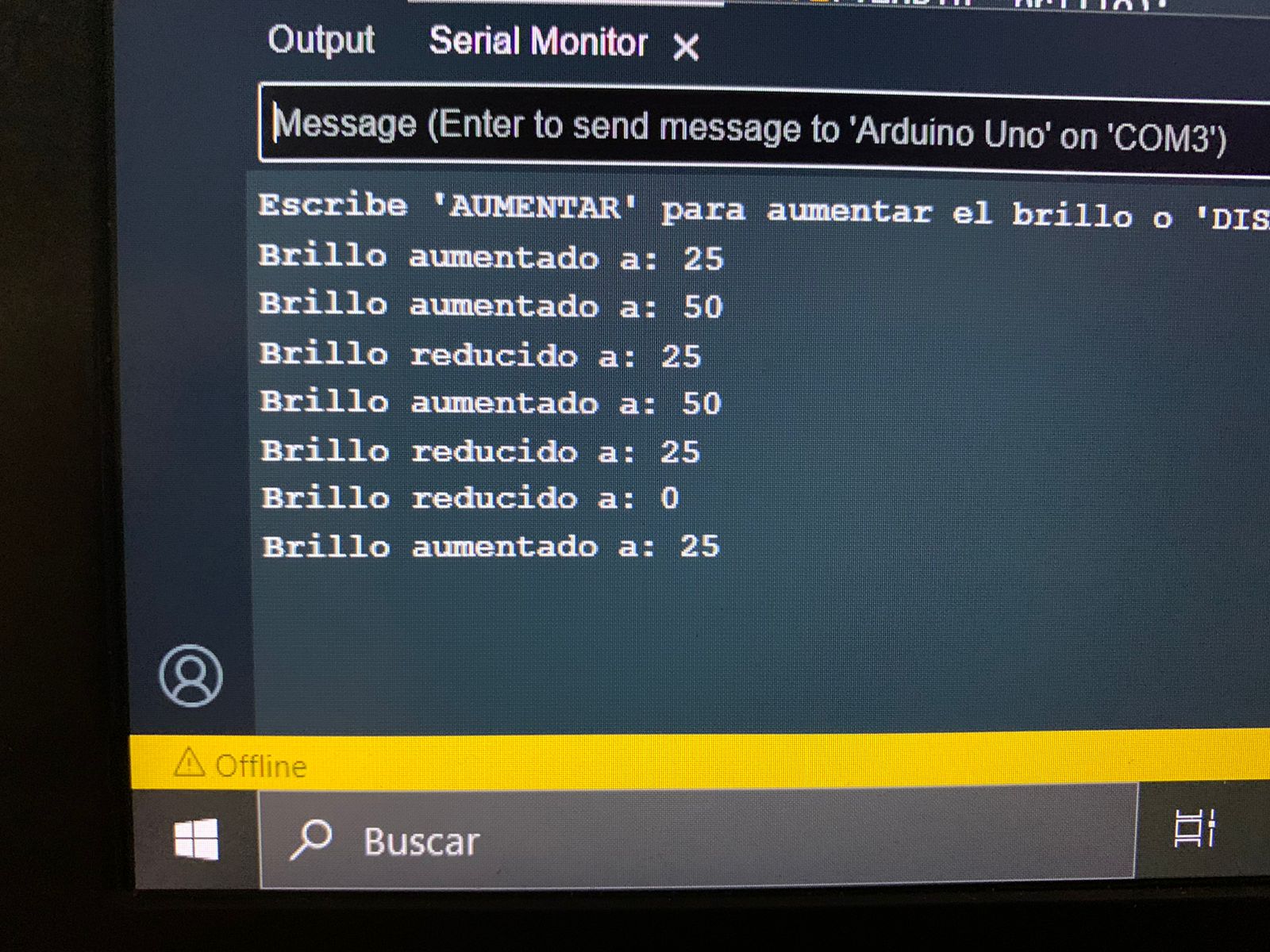
Cables

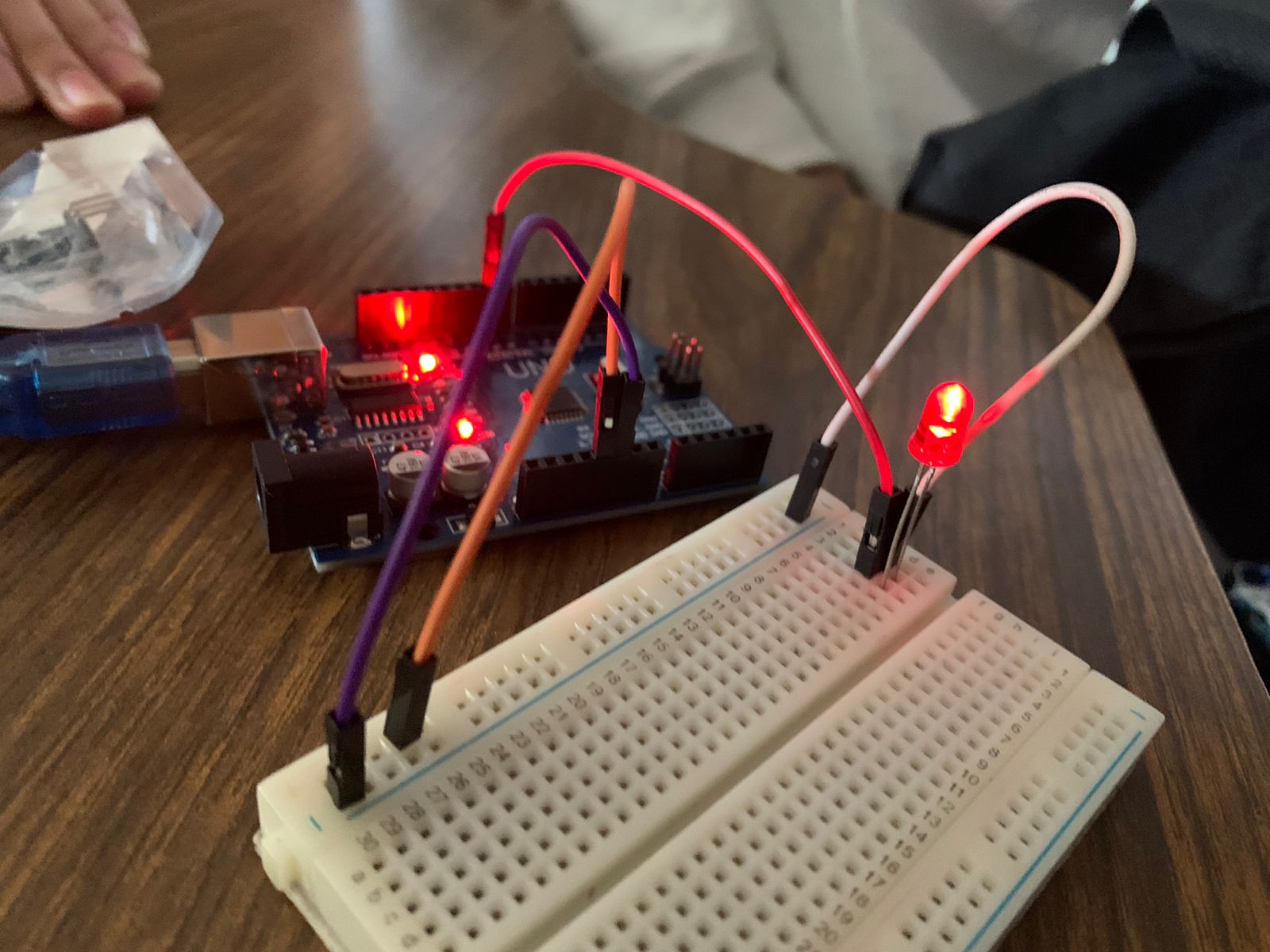
Microcontrolador (Arduino, ESP32, etc.)

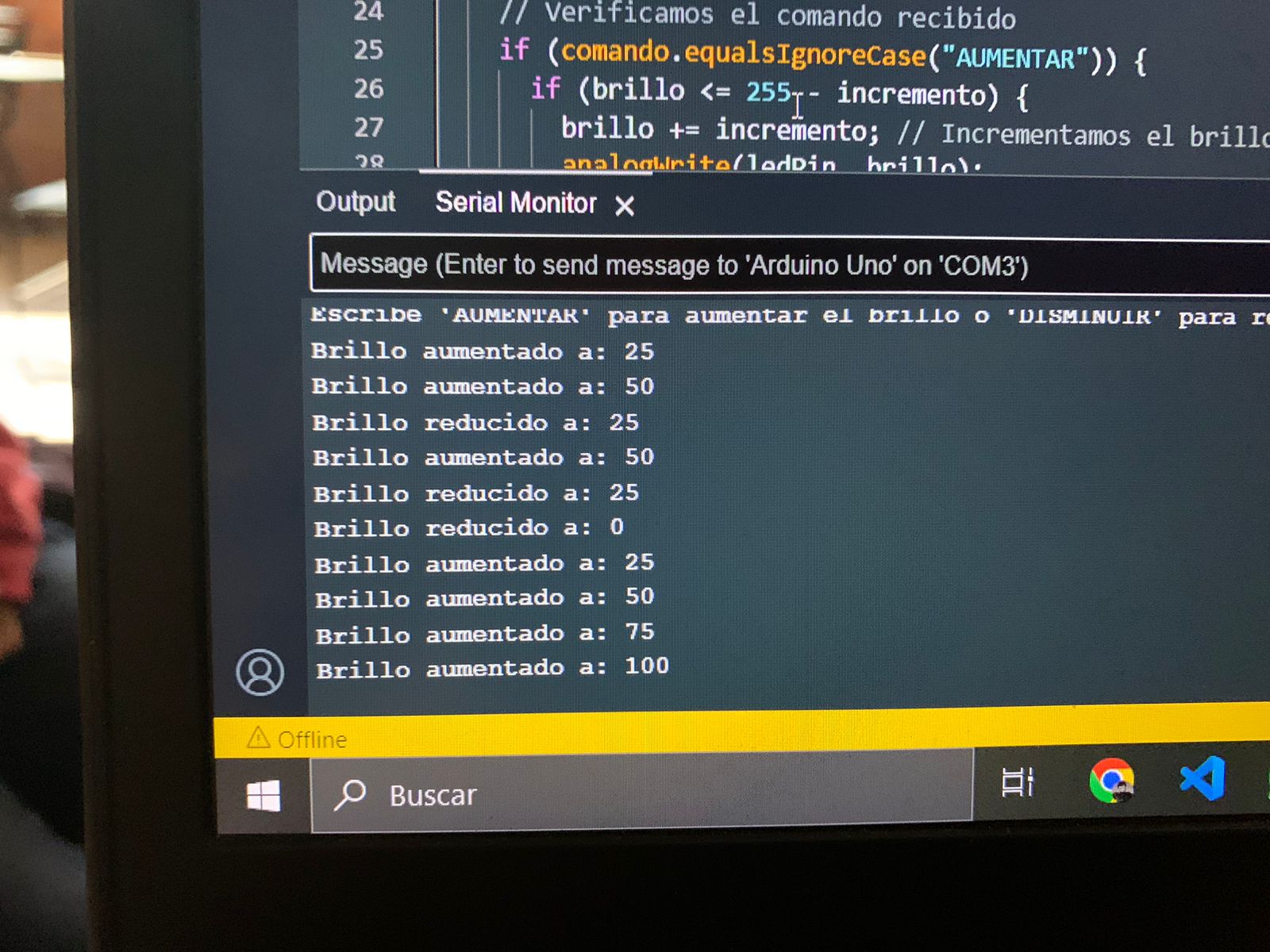
**Código:**

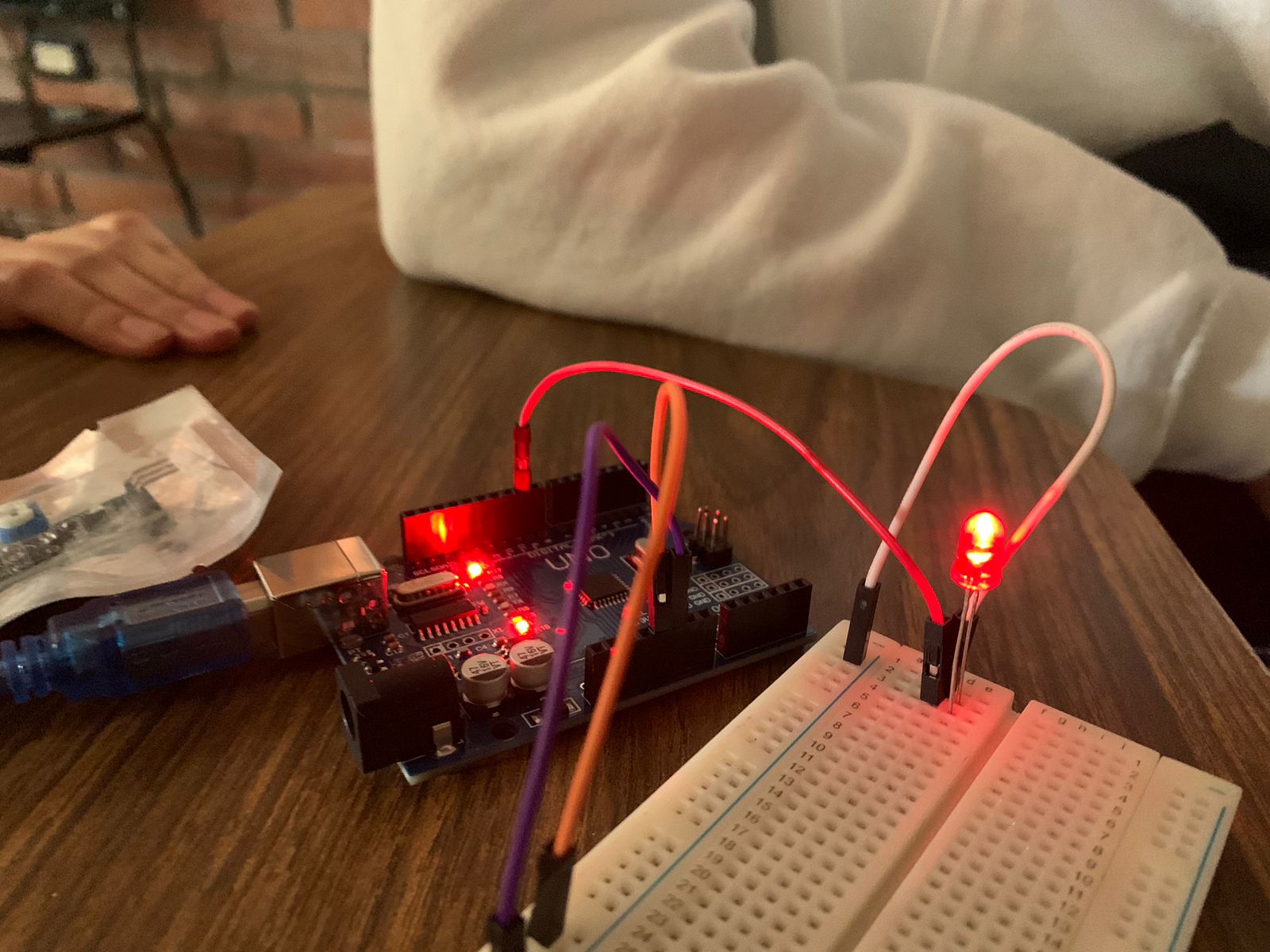












**Practica 11: Servomotores**

**¿Qué es un servomotor?**

Un servomotor es un tipo de motor eléctrico que permite un control preciso de posición, velocidad y aceleración. Tiene un sistema de control integrado que ajusta automáticamente su movimiento para alcanzar la posición deseada.

**¿De qué se compone?**

Motor DC o motor de corriente alterna: Proporciona la potencia.

Engranajes: Reducen la velocidad y aumentan el torque.

Sensor de posición: Detecta el ángulo del eje y envía retroalimentación al sistema de control.

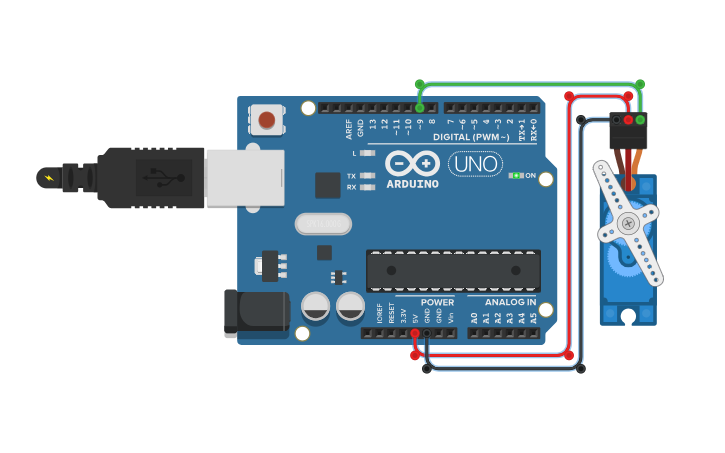
Controlador: Procesa la retroalimentación y ajusta la energía al motor para alcanzar la posición deseada.

Carcasa: Protege los componentes internos.

**Uso en Arduino**

**¿Cómo se usa un servomotor con Arduino?**

Para controlar un servomotor con Arduino, se utiliza la librería Servo.h, que simplifica el control del ángulo del eje. El servomotor recibe señales PWM (modulación por ancho de pulso) desde un pin digital del Arduino, lo que permite ajustar su posición de manera precisa.



**Ejemplos de aplicación en proyectos:**

Brazo robótico: Controlar los movimientos de cada articulación.

Puertas automáticas: Abrir y cerrar puertas mediante sensores o comandos.

Automodelismo: Control de dirección en autos a control remoto.

Cámaras de seguridad: Girar cámaras para seguimiento.

**Uso industrial de los servomotores**

En la industria, los servomotores son esenciales para aplicaciones que requieren precisión y repetibilidad:

**¿Dónde se aplican?**

Automatización de fábricas: Para mover brazos robóticos en líneas de ensamblaje.

Impresión 3D y CNC: Ajustan herramientas y materiales con alta precisión.

Manufactura textil: Manejan tensiones y movimientos en máquinas tejedoras.

Elevadores y transportadores: Controlan el posicionamiento en sistemas de carga y descarga.

Robótica: Para articulaciones y mecanismos de agarre.