

Trabajo 1:

Enlace al video: https://youtube.com/shorts/_da-cblvmNc

Código:

```
// === Pines ===
const int potPin = A0;      // Potenciómetro
const int buttonPin = 12;   // Pulsador con resistencia pull-up
const int enA = 10;         // Pin PWM -> ENA (L298N)
const int in1 = 9;          // IN1 (L298N)
const int in2 = 8;          // IN2 (L298N)

// === Variables ===
int potValue = 0;
int pwmValue = 0;
bool buttonState = 0;

void setup() {
  pinMode(potPin, INPUT);
  pinMode(buttonPin, INPUT_PULLUP); // Pulsador activo en LOW
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);

  Serial.begin(9600);
}

void loop() {
  // Lectura del potenciómetro (0 - 1023)
  potValue = analogRead(potPin);

  // Escalar a rango PWM (0 - 255)
  pwmValue = map(potValue, 0, 1023, 0, 255);

  // Leer pulsador
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    // === MODO NORMAL ===
    // Giro horario
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);

    // Velocidad controlada por potenciómetro
    analogWrite(enA, pwmValue);

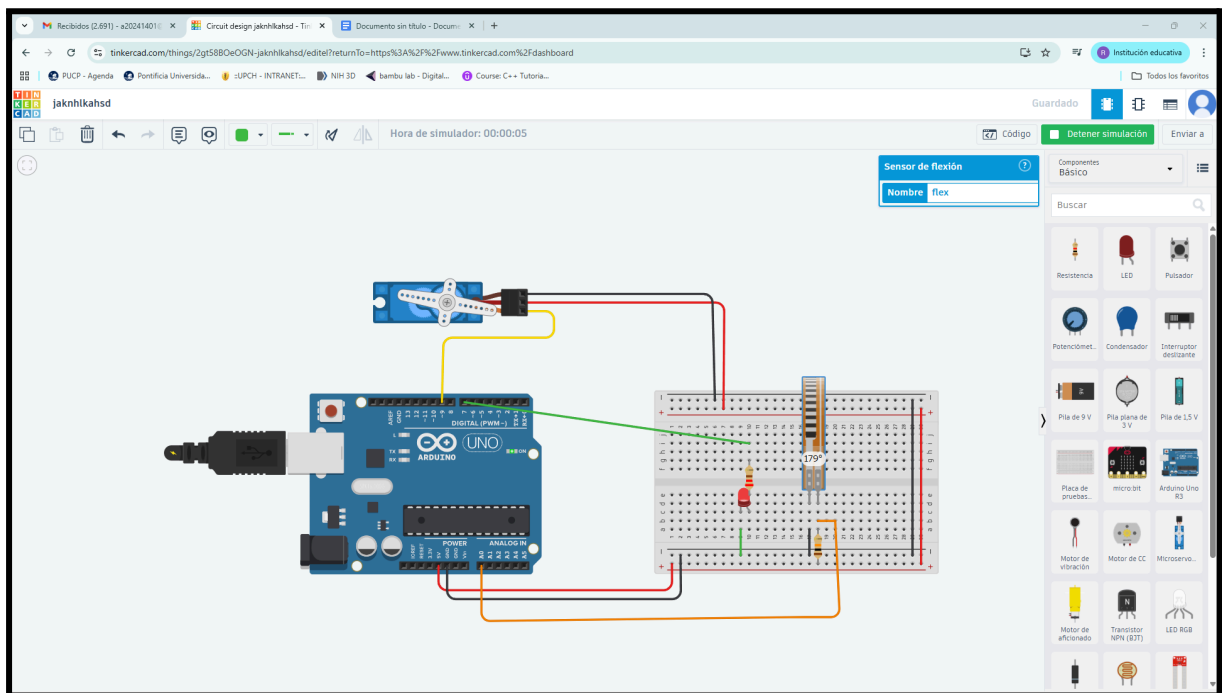
    Serial.print("Modo: Normal, PWM: ");
    Serial.print(pwmValue);
```

```
Serial.print(", Pot: ");
Serial.println(potValue);

} else {
  // === MODO ASISTENCIA ===
  // Giro antihorario
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);

  // Aceleración suave en rampa
  for (int i = 0; i <= 255; i += 5) {
    analogWrite(enA, i);
    Serial.print("Modo: Asistencia, PWM: ");
    Serial.println(i);
    delay(30); // Ajusta la velocidad de rampa
    if (digitalRead(buttonPin) == HIGH) break; // salir si se suelta el botón
  }
}
}
```

Trabajo 2: (Simulado en Tinkercad)



Código:

```
#include <Servo.h>
```

```
Servo miServo;
```

```
int pinFlex = A0;
```

```
int pinLed = 7;
```

```
int pinServo = 9;
```

```
void setup() {
```

```
    miServo.attach(pinServo);
```

```
    pinMode(pinLed, OUTPUT);
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    int valorFlex = analogRead(pinFlex);
```

```
    // Mapea el rango medido a 0–180
```

```
    int angulo = map(valorFlex, 767, 964, 0, 180);
```

```
    angulo = constrain(angulo, 0, 180);
```

```
    // Mueve el servo al ángulo detectado
```

```
    miServo.write(angulo);
```

```
// ● LED se enciende si el flex está muy doblado (cerca al máximo)
if (valorFlex > 950) {
    digitalWrite(pinLed, HIGH);
} else {
    digitalWrite(pinLed, LOW);
}

// Debug
Serial.print("Flex: ");
Serial.print(valorFlex);
Serial.print(" Ángulo: ");
Serial.println(angulo);

delay(50);
}
```

Trabajo 3:

Link: <https://youtube.com/shorts/9TG8R5W-HwA>

Código:

```
#include "DHT.h"

// === Configuración de pines ===
#define DHTPIN 6      // Pin de datos del DHT11
#define DHTTYPE DHT11 // Tipo de sensor
#define motorPin 3    // Pin PWM para controlar el motor (con transistor)

// === Inicialización del sensor ===
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(motorPin, OUTPUT);
  dht.begin();
  Serial.begin(9600);
}

void loop() {
  // Leer temperatura (en °C)
  float temp = dht.readTemperature();

  // Validar lectura correcta
  if (isnan(temp)) {
    Serial.println("Error al leer el DHT11");
    return;
  }

  int pwmValue = 0;

  // === Condiciones de vibración ===
  if (temp >= 24 && temp < 28) {
    pwmValue = 0; // Sin vibración
  }
  else if (temp >= 28 && temp < 31) {
    pwmValue = 128; // Vibración media (50% PWM)
  }
  else if (temp >= 31 && temp <= 34) {
    pwmValue = 255; // Vibración alta (100% PWM)
  }
}
```

```
// Aplicar PWM al motor
analogWrite(motorPin, pwmValue);

// Mostrar en Serial Monitor
Serial.print("Temperatura: ");
Serial.print(temp);
Serial.print(" °C | PWM: ");
Serial.println(pwmValue);

delay(1000); // Lectura cada 1 segundo
}
```