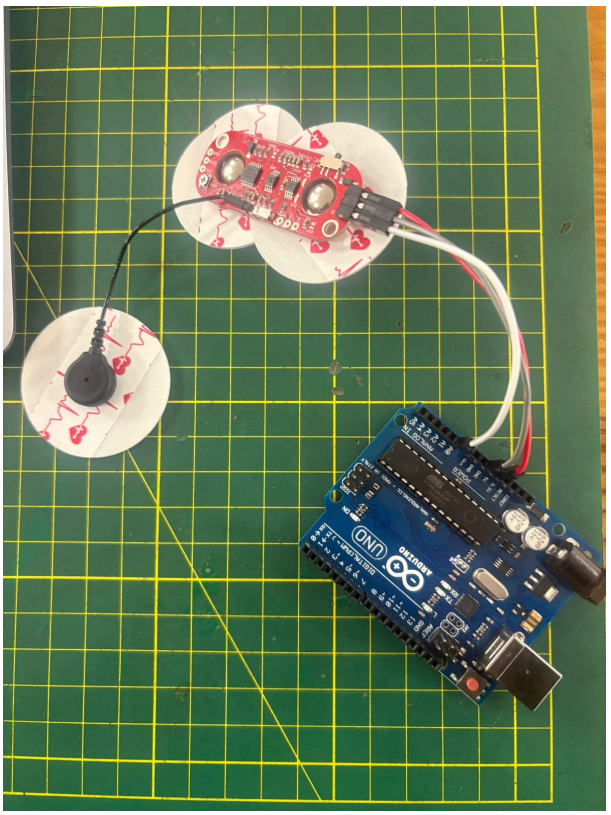
Taller 4 - Sensores

1. EMG
   Video: https://youtube.com/shorts/V_NVIQxtwuE

Código:

```
/*
  Myoware Muscle Sensor Test
  myoware-test.ino
  Demo of Myoware Muscle Sensor
  Output to Serial Monitor
  Use battery-powered computer or USB isolator for safety!
  DroneBot Workshop 2021
  https://dronebotworkshop.com
*/

// Connection to Myoware sensor
#define SENSOR_PIN 0

// Integer for sensor value
int sensorValue;

void setup() {
  // Set up serial port
  Serial.begin(9600);
}

void loop() {
  // Read sensor value
  sensorValue = analogRead(SENSOR_PIN);

  // Print value to Serial Monitor
  Serial.println(sensorValue);

  delay(10); // Pequeña pausa para no saturar el monitor
}
```



2. Sensor para caida

Video: https://youtube.com/shorts/9M1Kc93TrT4?feature=share

Código: #include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

#define BUZZER_PIN 8  // Pin del buzzer

void setup() {
  Serial.begin(115200);

  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);

  Wire.begin();
  delay(100);

  mpu.initialize();

  if (!mpu.testConnection()) {
    Serial.println("MPU6050 no conectado");
    while (1);
  }

  Serial.println("MPU6050 conectado correctamente");
}

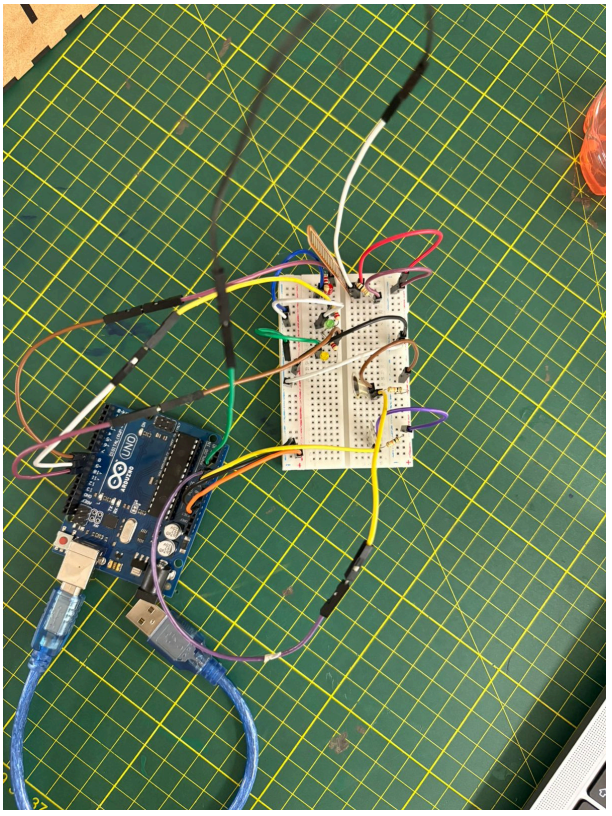
void loop() {
  int16_t ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);

  // Pasar a g
  float ax_g = ax / 16384.0;
  float ay_g = ay / 16384.0;
  float az_g = az / 16384.0;

  // Magnitud total
  float a_total = sqrt(ax_g * ax_g + ay_g * ay_g + az_g * az_g);

  Serial.print("Aceleracion_g: ");
  Serial.println(a_total);

  // Encender buzzer si supera 2g



```
  if (a_total > 2.0) {
    Serial.println(">>> ¡CAÍDA DETECTADA! <<<");
    tone(BUZZER_PIN, 1000);
    delay(2000);   // Suena mientras esté >2g
  } else {
    noTone(BUZZER_PIN);
    digitalWrite(8,HIGH);
        // Se apaga si baja a ≤2g
  }
```

Actividad 2.
Video: https://youtube.com/shorts/FNQsxX2g074

Código: / Pines de sensores

```
const int FLEX_PIN = A0;   // Flex sensor
const int FSR1_PIN = A1;   // FSR 1
const int FSR2_PIN = A2;   // FSR 2

// Pines de LEDs
const int LED_VERDE = 8;   // PWM -> Flex
const int LED_ROJO  = 9;   // ON/OFF -> FSR1
const int LED_AMAR  = 10;   // ON/OFF -> FSR2

// Rango esperado del Flex (ajusta con tu prueba real)
int flexMin = 250;
int flexMax = 800;
bool invertFlex = false;   // cambia a true si el brillo va al revés

// Umbrales de activación para los FSR (ajusta tras pruebas)
int umbralFSR1 = 450;
int umbralFSR2 = 450;

void setup() {
  pinMode(LED_VERDE, OUTPUT);
  pinMode(LED_ROJO, OUTPUT);
  pinMode(LED_AMAR, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // Lecturas analógicas
  int vFlex = analogRead(FLEX_PIN);
  int vFSR1 = analogRead(FSR1_PIN);
  int vFSR2 = analogRead(FSR2_PIN);
```

```
  // -------- LED verde proporcional al Flex --------
  vFlex = constrain(vFlex, flexMin, flexMax);
  int pwmV = invertFlex
           ? map(vFlex, flexMin, flexMax, 255, 0)
           : map(vFlex, flexMin, flexMax,   0, 255);
  pwmV = constrain(pwmV, 0, 255);
  analogWrite(LED_VERDE, pwmV);

  // -------- LEDs rojo y amarillo según FSR --------
  digitalWrite(LED_ROJO,  (vFSR1 >= umbralFSR1) ? HIGH : LOW);
  digitalWrite(LED_AMAR,  (vFSR2 >= umbralFSR2) ? HIGH : LOW);

  // -------- Monitor serial --------
  Serial.print("Flex: ");   Serial.print(vFlex);
  Serial.print(" -> Verde PWM: "); Serial.print(pwmV);

  Serial.print(" | FSR1: "); Serial.print(vFSR1);
  Serial.print(vFSR1 >= umbralFSR1 ? " (ROJO ON)" : " (ROJO off)");

  Serial.print(" | FSR2: "); Serial.print(vFSR2);
  Serial.println(vFSR2 >= umbralFSR2 ? " (AMARILLO ON)" : " (AMARILLO off)");

  delay(50);
}
```