# **Informe Técnico: Implementación de Actuadores con Arduino**

****

**Fundamentos de diseño**

**Docentes:** Renzo Chan Ríos / Lewis De La Cruz

**Integrantes:**

* Ordinola Zapata Karla Maybe
* Gomez Rodriguez Adim Orfel
* Elvis Deiwis Pilco Cruz
* Luis Martin Valenzuela Valer

1. **Introducción**

Los actuadores son elementos clave en el diseño de sistemas electrónicos interactivos, pues convierten señales eléctricas en movimientos o acciones físicas esenciales para interfaces biomédicas y dispositivos de asistencia. Arduino presenta diferentes implementaciones con con servomotores, motores de corriente continua y micromotores de vibración, enfocándose en el control mediante señales digitales y PWM, así como en la lectura analógica con sensores como potenciómetros para lograr una interacción proporcional y dinámica. También se usó componentes eléctricas esenciales que incluyen el uso de transistores, resistencia y drivers para manejar cargas, junto con ejemplos prácticos que demuestran aplicaciones reales en rehabilitación, retroalimentación háptica y sistemas de control de posición y velocidad. De esta manera, el trabajo busca fortalecer competencias en electrónica aplicada y fomentar el diseño de soluciones tecnológicas centradas en el usuario para mejorar la calidad de vida en contextos biomédicos y de automatización.

**2. Objetivos**

- Comprender el funcionamiento de diferentes tipos de actuadores usados con Arduino.

- Implementar circuitos de control para servomotores, motores DC y micromotores.

- Aplicar conceptos de señales digitales y PWM para controlar movimiento y posición.

- Explorar posibles aplicaciones reales de cada tipo de actuador.

**3. Materiales y Herramientas**

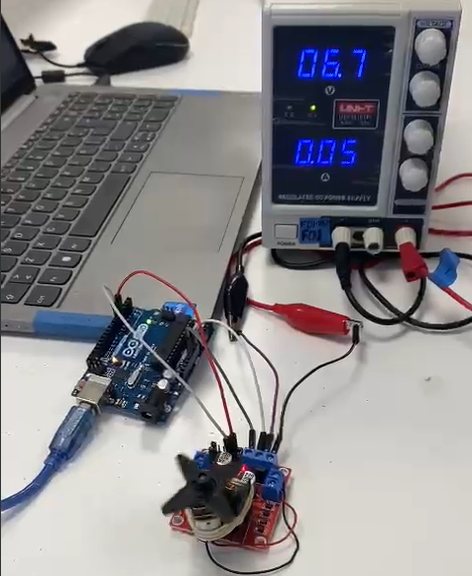
* Placa Arduino UNO o similar
* Protoboard y cables de conexión
* Servomotor DS3235
* Motor DC (3-6V)
* Micromotor de vibración
* Transistores NPN
* Potenciómetro de 10kΩ
* Diodo 1N4007
* Módulo driver L298N
* Fuente de alimentación externa (6-12V)

**4. Desarrollo y Ejemplos Implementados**

### **4.1 Control de Dirección del Motor GA12-N20 con Arduino y L298N**

En este ejemplo se conecta el motor GA12-N20 a un módulo L298N para controlar únicamente el sentido de giro del motor. Usando dos pines digitales del Arduino, es posible alternar entre giro horario y antihorario cada 5 segundos. Este tipo de control simple es ideal para entender cómo funciona un puente H y cómo se puede aplicar en proyectos reales que requieren invertir el movimiento. El motor GA12-N20 destaca por ser pequeño, preciso y tener un buen nivel de torque para su tamaño, lo que lo hace útil en proyectos biomédicos o robóticos portátiles.

En este caso, no se utiliza control de velocidad (PWM), solo dirección, por lo que el pin de habilitación del L298N (ENA) se mantiene activo conectado a una fuente de 5V.



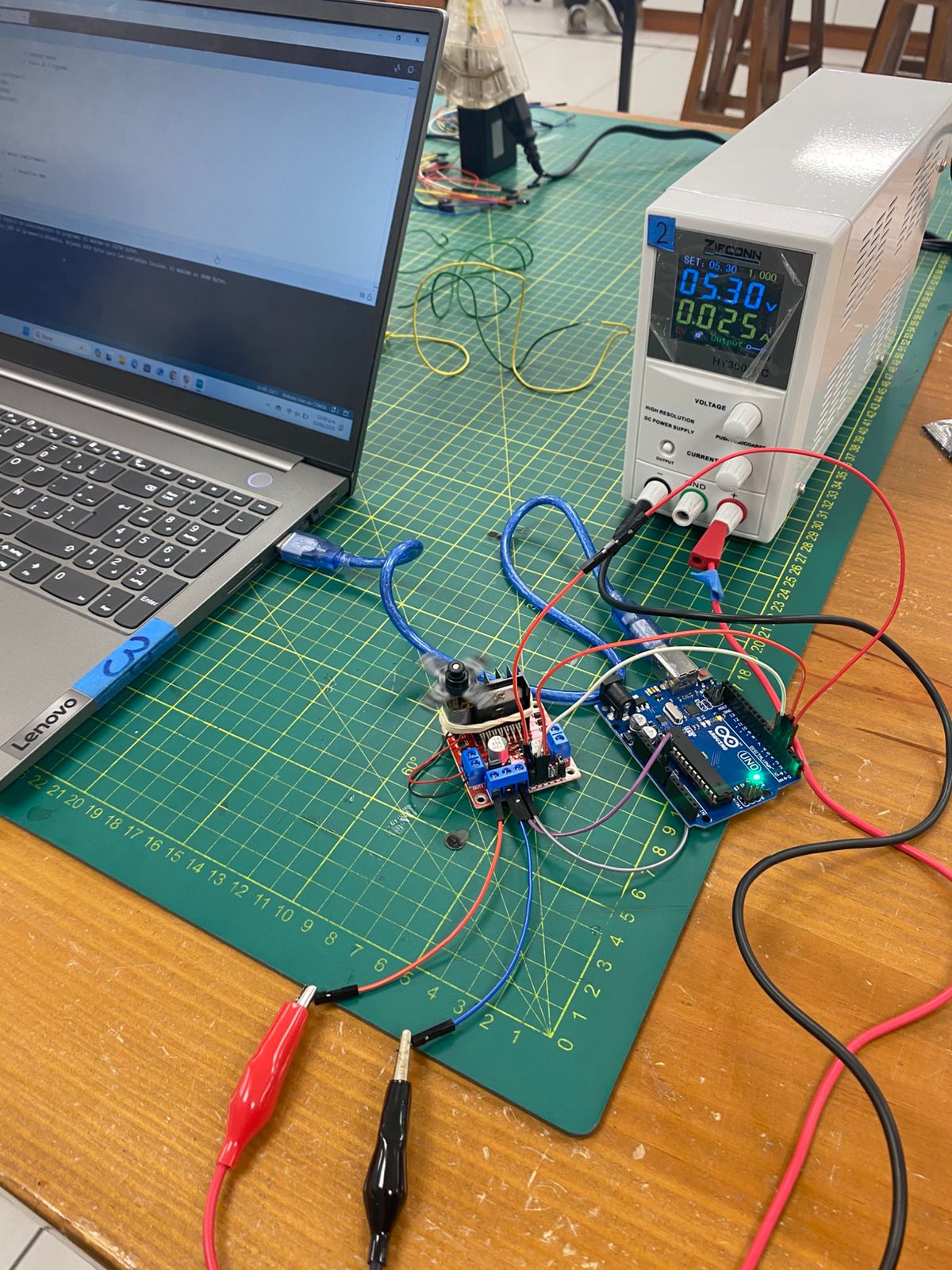
Motor GA12-N20 + Driver + Arduino + Fuente de alimentación

### 

### **4.2 Control de Dirección y Velocidad del Motor GA12-N20 con Arduino y L298N**

En esta implementación se utiliza un motor GA12-N20 junto con un módulo L298N para controlar tanto su dirección como su velocidad. El sentido de giro se determina mediante dos pines digitales (IN1 e IN2) conectados a pines digitales de Arduino (pines 4 y 5), mientras que la velocidad se ajusta con una señal PWM enviada al pin ENA del módulo. El motor alterna entre giros en sentido horario y antihorario cada 5 segundos, implementado en el código de Arduino IDE, lo que permite simular sistemas que requieren ciclos de movimiento repetitivo, como bombas peristálticas o mecanismos de asistencia en rehabilitación.

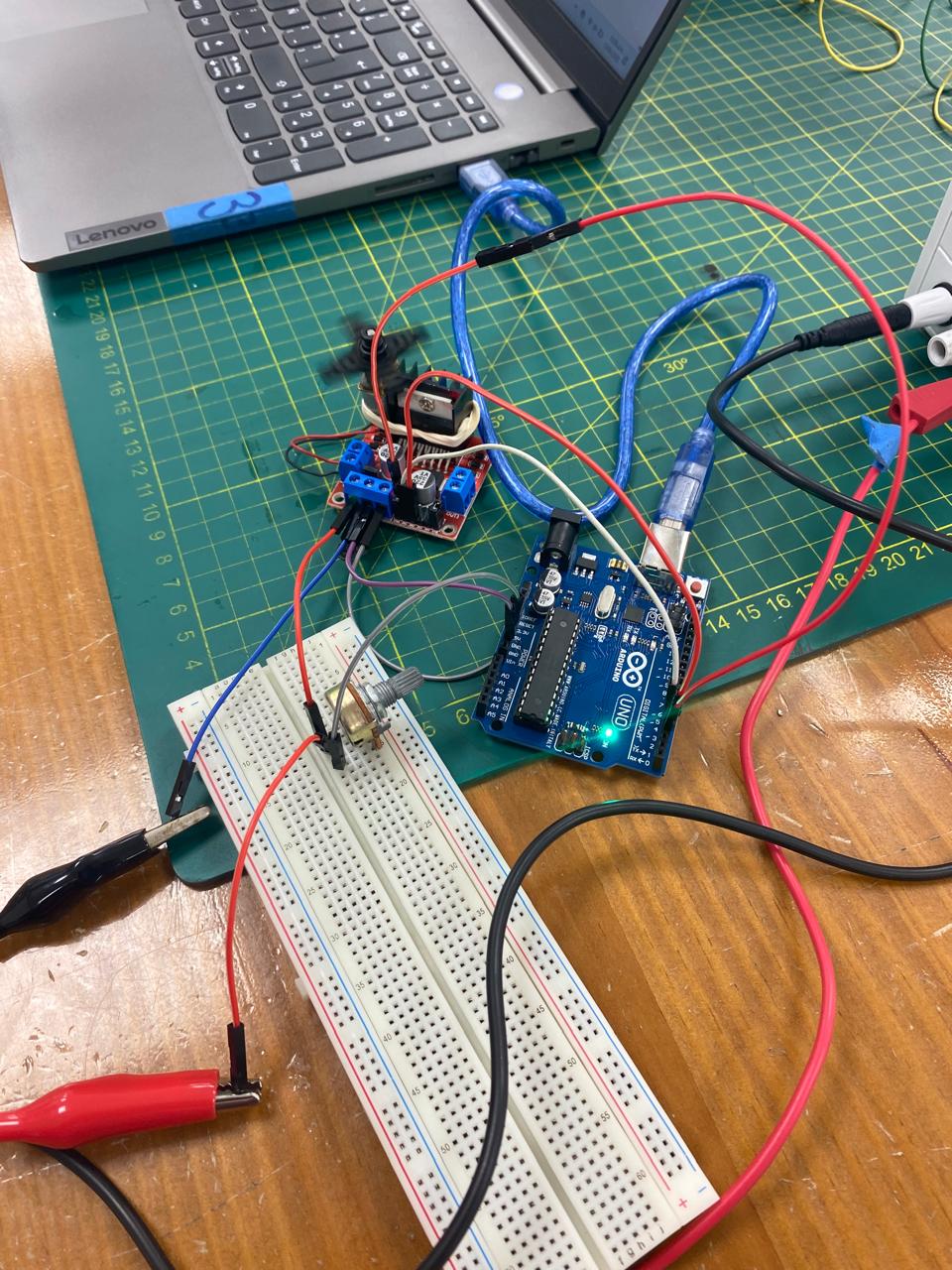
Este tipo de control es ideal para comprender cómo se integran señales digitales y PWM en sistemas de movimiento. Además, permite experimentar con distintos niveles de velocidad y su efecto en el comportamiento del motor. Al emplear una fuente de alimentación externa adecuada, se asegura un funcionamiento estable sin sobrecargar la placa Arduino.



### **4.3 Control de Velocidad del Motor GA12-N20 con Potenciómetro y Arduino**

En este experimento se utiliza un potenciómetro para controlar manualmente la velocidad de un motor GA12-N20, mediante un módulo L298N. El Arduino lee la posición del potenciómetro a través de una entrada analógica y convierte ese valor en una señal PWM, que regula la velocidad del motor. La dirección de giro se mantiene fija en sentido horario, permitiendo enfocarse únicamente en la variación de velocidad.

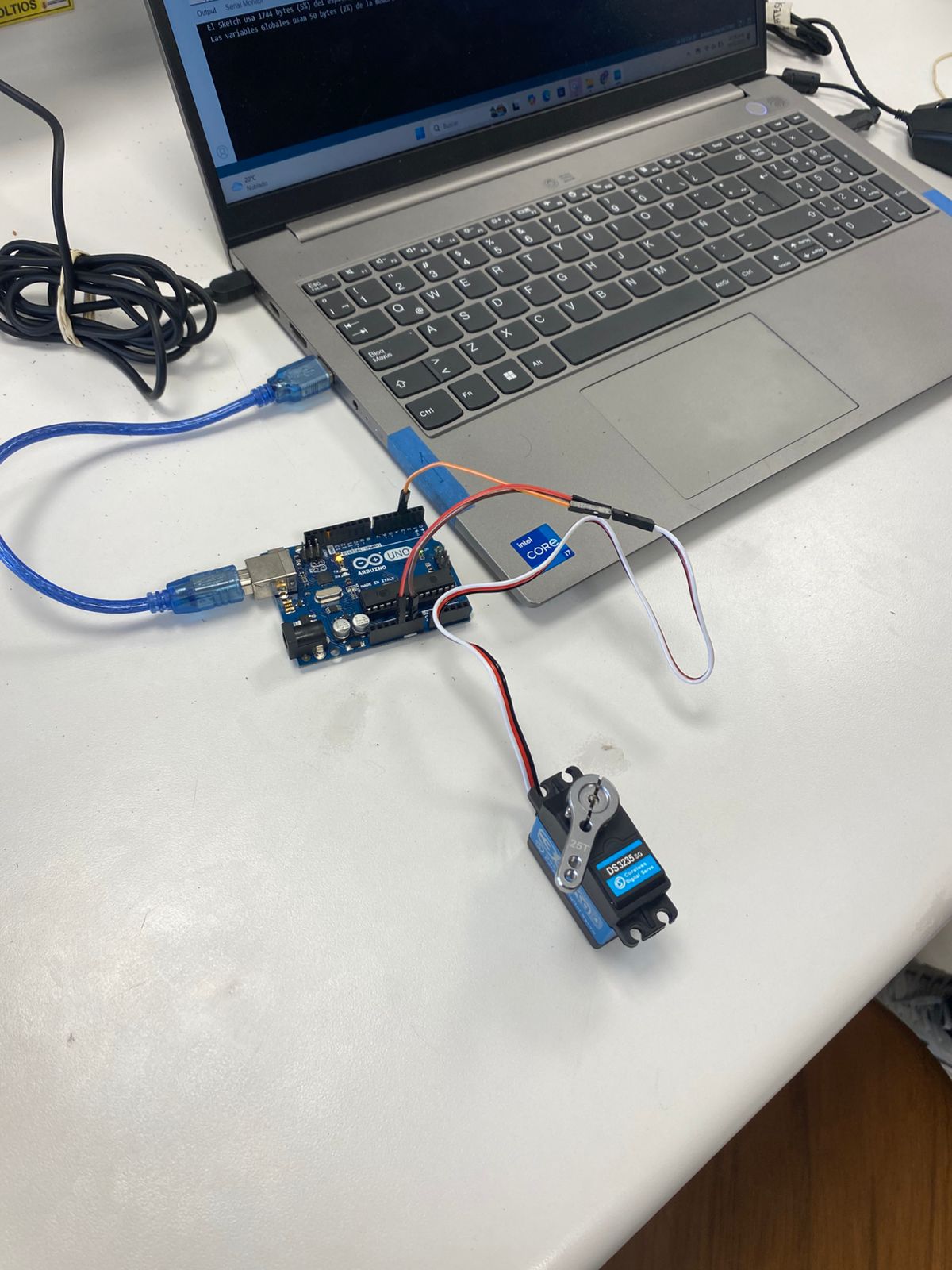
Este tipo de control resulta muy útil para simular interfaces manuales en sistemas reales, como dispositivos de terapia física, ajustes de flujo en bombas, o ejercicios de estimulación pasiva. Además, los valores analógicos y PWM se imprimen en el monitor serial, lo que permite visualizar en tiempo real cómo la entrada del usuario afecta el comportamiento del actuador.



### **4.4 Control de Posición del Servomotor DS3235 con Arduino**

En este experimento se controla la posición de un servomotor DS3235 moviéndolo entre cuatro ángulos definidos: 0°, 90°, 180° y 270°. Para lograrlo, se utiliza la función “writeMicroseconds()” de la librería Servo, que permite enviar pulsos personalizados más allá del rango típico de servos estándar como el SG90. Este tipo de servo sirve cuando se necesita un alto torque y un rango angular extendido.

Cada posición se mantiene durante 5 segundos para observar el movimiento. El control por pulsos controlados permite una mejor precisión en sistemas que requieren un posicionamiento preciso y repetible, como prótesis, brazos mecánicos o dispositivos de rehabilitación asistida.

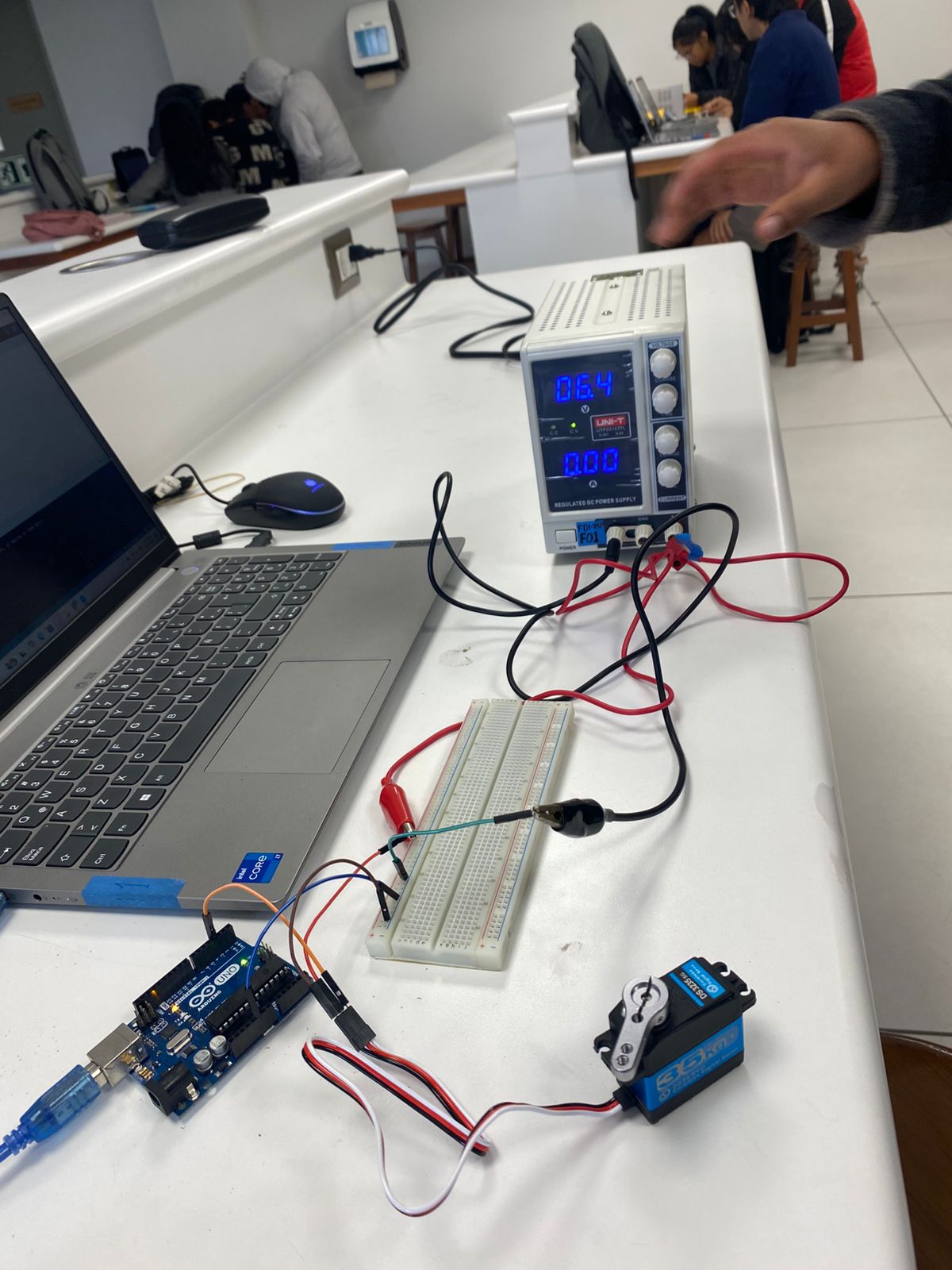


Arduino + Servomotor DS3235

### **4.5 Control del Servomotor DS3235 con Potenciómetro (PWM en Microsegundos)**

En esta práctica se utiliza un potenciómetro para controlar la posición de un servomotor DS3235, aprovechando su amplio rango de giro (hasta 265°) y de alta potencia. El valor leído del potenciómetro se convierte directamente en un pulso PWM expresado en microsegundos, que se envía al servo usando la función “writeMicroseconds()”.

Aunque el cálculo del ángulo no es necesario para el control, se incluye en el monitor serial como referencia visual, lo que resulta útil para verificar la respuesta del servo en tiempo real. Este tipo de control proporcional es común en interfaces manuales de sistemas biomédicos, como ortesis activas o simuladores de movimiento.

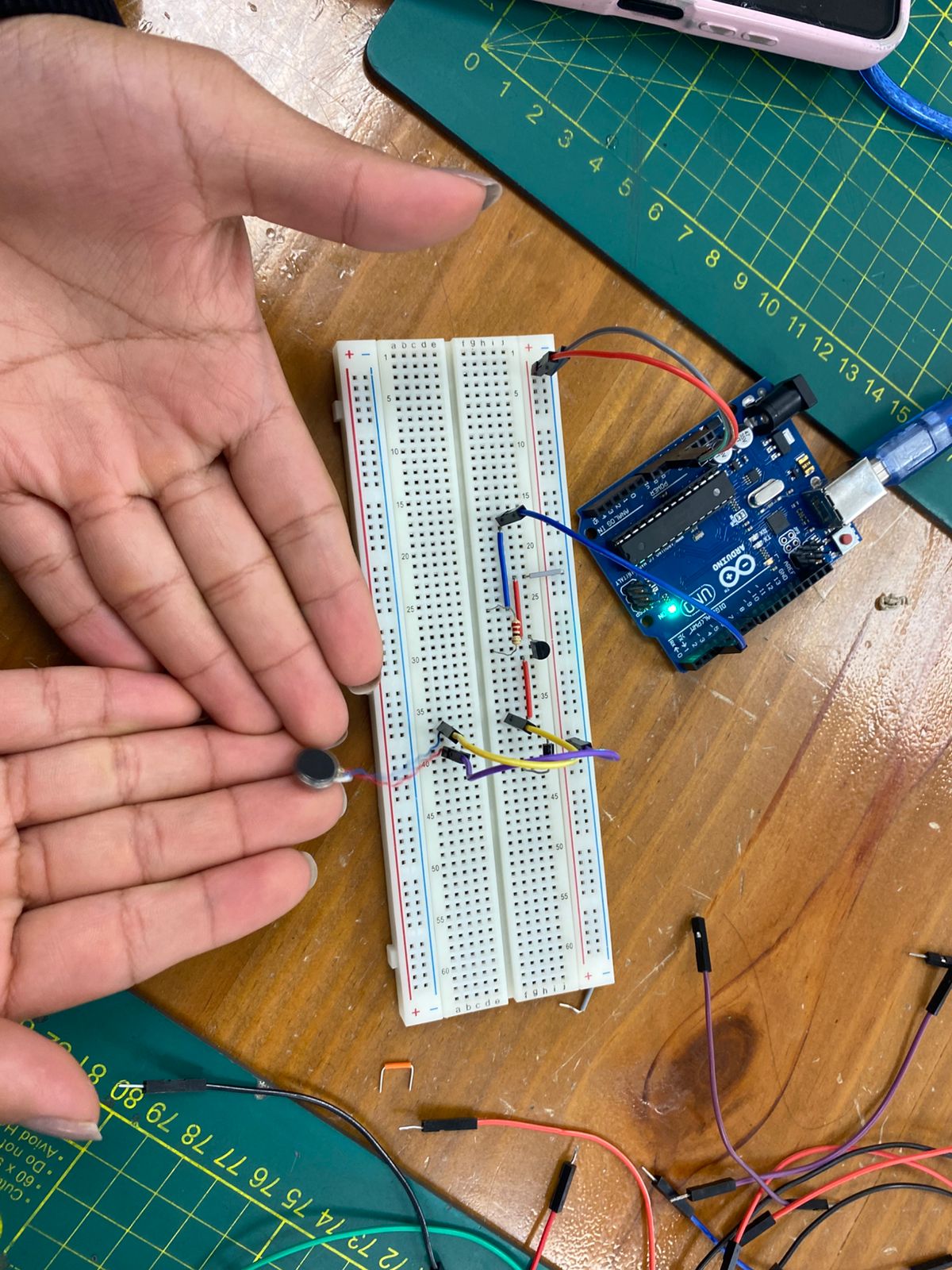


Aclaración: No se encontró la foto con el potenciómetro

### **4.6 Activación Básica de un Micromotor de Vibración con Arduino**

En este ejemplo se muestra cómo encender un micromotor de vibración utilizando Arduino y un transistor como interruptor. Se trata de una implementación sencilla pero muy útil para dispositivos biomédicos portátiles o en interfaces silenciosas que emiten señales táctiles.

El Arduino envía una señal digital al transistor NPN, que actúa como un interruptor. Cuando el pin digital se activa, el transistor permite el paso de corriente hacia el motor, generando una vibración constante durante unos segundos. Este tipo de vibración puede servir como una alerta sin sonido, o como una señal que acompaña visualizaciones o acciones físicas en terapias de rehabilitación.



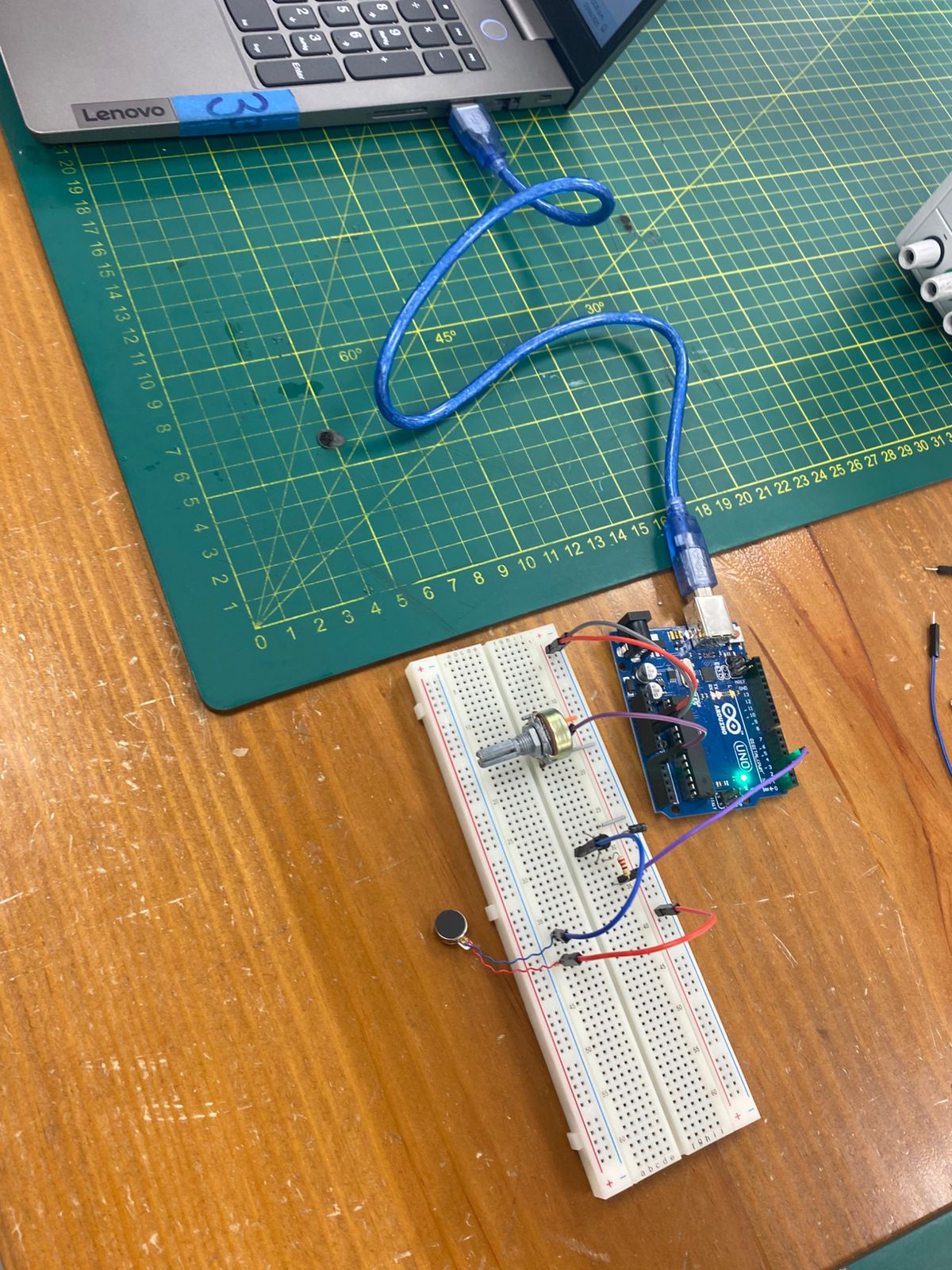
Arduino + Protoboard + Vibrador + Resistencia + Diodo + Transistor

### **4.7 Control de Intensidad de Vibración con Potenciómetro y Arduino**

En este experimento se logra controlar de manera proporcional la intensidad de vibración de un micromotor tipo *coin cell* utilizando un potenciómetro como entrada de usuario. Este tipo de implementación es muy útil para simular sistemas de retroalimentación háptica, como los que se emplean en órtesis, prótesis o interfaces sensoriales.

El funcionamiento es simple pero poderoso: al girar el potenciómetro, se genera una señal analógica que Arduino lee y convierte en una señal PWM. Esta señal PWM regula la corriente que pasa a través de un transistor hacia el motor vibrador, controlando así la intensidad de vibración. El usuario puede "sentir" ese cambio, lo que convierte a este sistema en una interfaz táctil interactiva.

Además, el monitor serial muestra en tiempo real tanto el valor leído del potenciómetro como el nivel de PWM aplicado. Esto permite visualizar y entender la relación entre la entrada del usuario y la respuesta física del sistema, una herramienta clave para el diseño centrado en el usuario.



Vibrador controlado con potenciómetro

**5. Conclusión**

La experiencia de trabajar con actuadores y Arduino nos demuestra cómo la electrónica puede dar vida a proyectos que interactúan con el entorno y con las personas. Cada tipo de actuador aporta algo distinto: los servos ofrecen precisión, los motores DC fuerza y movimiento continuo, y los motores de vibración una comunicación táctil discreta pero efectiva. Aprender a controlar estos dispositivos con transistores, PWM y sensores abre un mundo de posibilidades para crear soluciones biomédicas y tecnológicas que respondan a necesidades reales. En resumen, este taller no solo fortalece habilidades técnicas, sino que también inspira a diseñar dispositivos más humanos, accesibles y útiles en la vida diaria.

**6. Referencia**

Chan Ríos, R., & De La Cruz, L. (2025). *Taller de Actuadores Electromecánicos con Arduino*. Universidad Peruana Cayetano Heredia (UPCH). Repositorio GitHub:<https://github.com/devRenzo/fundbio-actuadores>