



# **Carpeta de Campo**

# **Proyecto Midas**

## **Integrantes:**

**Espandrio Santiago**  
**Horn Lautaro**  
**Jurado Gabriel**  
**Ledesma Kevin**  
**Tapia Federico**

## **Carpeta de Campo**

Exoesqueleto para mano con pérdida de movilidad/sin movilidad:

El título es ese debido a que el proyecto en sí está centrado en crear un tipo de ortesis para poder ayudar a esas personas que perdieron la movilidad o parte de ella en la mano/s

### **Integrantes:**

- Horn Lautaro
- Ledesma Kevin
- Espandrio Santiago Emanuel
- Tapia Federico Nicolás
- Jurado Gabriel

### **Objetivo del proyecto:**

El objetivo de este proyecto es recuperar el movimiento de un brazo orgánico, cuyas capacidades de movimiento están reducidas por cualquier problema previo (desde un accidente hasta una enfermedad o discapacidad de nacimiento).

### **Utilidades de el proyecto:**

En el mundo 1710 millones de personas padecen de trastornos músculo esqueléticos que los hace tener pérdida o reducción de movilidad en determinadas zonas, y dicho proyecto ayuda al sector médico a asistir a esas personas ya que aún tienen ciertas dificultades para implementar rehabilitaciones/recuperaciones de movilidad y es una buena alternativa ya que puede ser una alternativa notablemente más económica que otras, innovadora, simple y cumple la función deseada.

### **Descripción del funcionamiento:**

Lo que ocurre en este sistema es que se llevará a cabo a partir de señales musculares/nerviosas captadas a partir de una zona funcional del cuerpo se hará que este exoesqueleto lleve a cabo los movimientos con la mayor movilidad normal posible, por lo menos aspirando a un 75% de normalidad en

comparación a las personas que no padecen de esta pérdida de movilidad, esas señales pasarán por una placa que sensa las señales musculares y las procesa junto con un acelerómetro que será el que se encarga de calcular y dirigir correctamente las señales deseadas al movimiento que se quiere realizar el cual se llevará a cabo por unos servomecanismos articulados en los dedos a partir de un guante con agregados impresos en 3D solo en zonas necesarias para que no pierda la flexibilidad que deseamos que tenga este dispositivo el cual aumenta el porcentaje de movilidad que deseamos para que la persona pueda realizar sus actividades con la mayor normalidad posible.

### **Análisis de factibilidad:**

Para comenzar ya tenemos vistos la mayoría de componentes necesarios, ya tenemos un código base para ir arreglando y modificando a medida que avance el proyecto, y una idea de cómo serán las conexiones ensamblaje y base de cómo encajar el Exoesqueleto en la mano porque una mano cuando pierde la movilidad no queda abierta y de forma que entre como si nada en el Exoesqueleto.

Ya estuvimos poniéndonos en contacto para concretar en unos días una charla con un cirujano y un kinesiólogo para hablar sobre el proyecto, hacerle ciertas preguntas que ayudarían al proyecto y dirían que tan factible es que funcione.

- **Diseño:**

- Espandrio Emanuel Santiago junto con Tapia Federico Nicolás serán los encargados de la parte del diseño estético y graficado para llevar a cabo de tal manera el Exoesqueleto.

- Dibujarán diferentes croquis con el fin de construir escalonadamente el modelo más adecuado para los requerimientos de la persona o mano en cuestión.

- **La estructura:**

- Horn Lautaro diseñará la base con un guante y algunos agregados con impresora 3D para más estabilidad, agarre y solidez pero a la vez que no pierda esa flexibilidad que lo hace tan útil, también se encargará de llevar a cabo de pcb y soldadura

•  
○ Se colocará un elemento de unión como algunas barritas de metal entre partes de impresión 3D y el guante en la parte de las articulaciones enganchados de una manera que no entorpezcan el movimiento deseado.

° Jurado Gabriel se encargará de tareas que se vayan asignando y en un inicio en la impresión 3D u obtención de planos para piezas necesarias

- Sistema de propulsión: El diseño del sistema está pensado para ser propulsado/movido por motores eléctricos que impulsen unos enganches que va a haber en un guante a la altura de las articulaciones de los dedos para un movimiento más complejo y más del rango que uno desee. Y eso último se logrará con la ayuda de Horn Lautaro en la parte de programación/software del Exoesqueleto.

- Kevin Joel Ledesma se encargará de parte hardware y composición del circuito en cuanto a simulaciones, pruebas y calculados de valores para evitar fallos, quemaduras del circuito o cortocircuito

## **Marzo:**

Charla introductoria con los profesores Medina y Bianco.

Donde explicaron cómo se llevaría a cabo el trabajo en el año para organizar todo de manera eficiente pudiendo llevar a cabo el proyecto de una mejor forma de avanza y más prolija en cuanto a formas de manejarse

Empezamos a trabajar en el anteproyecto.

Terminamos la primera versión del anteproyecto.

Trabajamos en las historias de usuario principales:

Que son básicamente los casos y situaciones que se pueden dar y los modos de afrontarlos en situaciones hipotéticas planteadas para enfocarlo en un ambiente más profesional y en el caso de que el producto se haga conocido pueda ser entendido en conjunto con sus funciones y utilidades fácilmente a partir de ejemplos y sus explicaciones con protocolos de trabajo y uso

Corregimos las historias erróneas:

Ya que algunas necesitaban ser más específicas, completas y cubrir ciertos contenidos para cumplir con el requerimiento técnico

Charla con el profesor Medina sobre los sprints/backlog.

Esto nos ayudó a organizarnos más de modo que fuera un trabajo más eficiente a cada uno tener su tarea propuesta en un sistema de una aplicación y se avise cuando se haya terminado de modo que pueda proponerse otra tarea o que asista en las que se requiera ayuda

Armamos las historias desglosadas en tareas.

De modo que a partir de esos casos y utilidades asignadas en las historias de usuario se deben cumplir esas funciones y para ello se crearon tareas que se llevarán a cabo para que el funcionamiento del dispositivo sea como se indica en esas historias y en el principio de funcionamiento del proyecto

Terminamos la versión final del anteproyecto.

Lo cual nos habilitó principalmente a dirigir mas nuestra atencion al proyecto en cuestiones de que empezamos a hacer

## **Abril:**

Se corrigieron algunos errores de diseño del plan original.

Ya que había ciertos componentes que no se podían obtener o el modo de construirlo que habíamos pensado no era del todo eficiente ya que tenia ciertos vacios en la idea y eso podría complicar las cosas más a largo plazo.

Se establece un principio de funcionamiento.

Que esto se basó en un modo de que el circuito funcionara en conjunto a un sistema que lleve a cabo el movimiento de los dedos y como iria adaptado a la mano este sistema de modo que no complique ni se pierda la naturalidad del movimiento y se llegue a el mayor porcentaje de movimiento posible estimando a un 75%.

## **Mayo:**

Creación de Redes Sociales

Básicamente creamos un Instagram y un LinkedIn para compartir los datos sobre el proyecto y que se conozca de su existencia

Pedido de componentes:

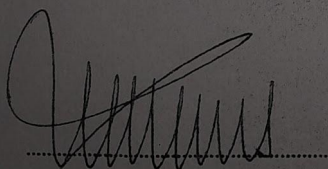
# PRÁCTICAS PROFECIONALIZANTES – SOLICITUD DE MATERIALES

NÚMERO DE PEDIDO: 4      FECHA: 4/8/23  
 NOMBRE DEL PROYECTO: Proyecto Hids      ESPECIALIDAD: Informática

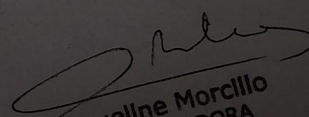
NOMBRE DE INTEGRANTES	DATOS DEL PROVEEDOR
Arnau Lautaro	EMPRESA: Starware
Kevin Joel Ledesma	TELÉFONO: +54-9-46020-6473
Espandio Emanuel Santiago	E-MAIL: vesitar@starware.com
Tapia Hissolar Federico	FAX:
Jurado Alejandro Gabriel	WEB: https://hids.starware.com.ar
	DIRECCIÓN:

MATERIALES NECESARIOS			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO PARCIAL
Placa desarrollo HDSB-berry Pi Pico PIP2040	1	\$2669	

PRECIO TOTAL: ~~\$6783~~ \$2669



FIRMA DEL RESPONSABLE  
 Profa CARLOS BIANCO

  
 Jacqueline Morcillo  
 ADMINISTRADORA  
 C.A. E.S.T. T.R.2.

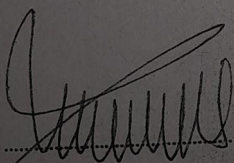
**PRÁCTICAS PROFECIONALIZANTES – SOLICITUD DE MATERIALES**

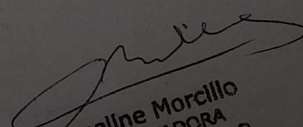
NÚMERO DE PEDIDO: .....2.....      FECHA: 4/1/23  
 NOMBRE DEL PROYECTO: Proyecto Nidas.....      ESPECIALIDAD: Avionica.....

NOMBRE DE INTEGRANTES	DATOS DEL PROVEEDOR
Hora Lautaro	EMPRESA: Tecnolive Usa
Kevin Joel Ledesma	TELÉFONO: +54 9 358 513 - 3333
Espandio Emanuel Santiago	E-MAIL: ventas@tecnoliveusa.com
Tapia Nicolas Federico	FAX:
Jurado Alejandro Gabriel	WEB: https://tecnoliveusa.com/#
	DIRECCIÓN: Paso 270, Rio Cuarto, Córdoba

MATERIALES NECESARIOS			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO PARCIAL
Acelerometro Adxl345	5	\$7.289,82	\$36.450 +
3 ejes Sensor xyz Ardu			\$2320 (envio)

PRECIO TOTAL: \$38770

  
 FIRMA DEL RESPONSABLE  
 Prof CARLOS BIANCO

  
 Jacqueline Morcillo  
 ADMINISTRADORA  
 Asoc. Coop. E.E.T. T.R.Q.  
 04/08/23



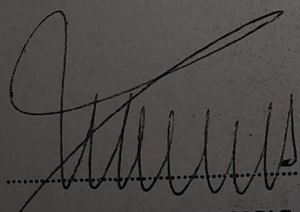
# PRÁCTICAS PROFECIONALIZANTES – SOLICITUD DE MATERIALES

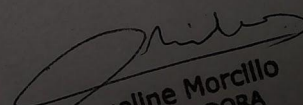
NÚMERO DE PEDIDO: 3 FECHA: 4/1/2013 ESPECIALIDAD: Automatización  
 NOMBRE DEL PROYECTO: Proyecto Mides

NOMBRE DE INTEGRANTES	DATOS DEL PROVEEDOR
Hernán Lautam	EMPRESA: Todo Micro
Kevin Joel Ledesma	TELÉFONO: +54 11 5263 9793
Espandrio Emanuel Santiago	E-MAIL: INFO@TodoMicro.com.ar
Tapia Nicolas Federico	FAX:
Jorda Alejandro Gabriel	WEB: <a href="https://www.TodoMicro.com.ar">https://www.TodoMicro.com.ar</a>
	DIRECCIÓN: <del>Paseo</del> <del>7000</del> Galería central Av. corrientes
	640 Piso 4 De 7
	Ciudad de Bs As

MATERIALES NECESARIOS			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO PARCIAL
Servo Motor S3003 3,2Kg Para robotica DOMOTICA	5	\$3.066,80	\$15.334,02

PRECIO TOTAL: \$15.334,02

  
 FIRMA DEL RESPONSABLE  
 Prof. CARLOS BIANCO

  
 Jacqueline Morcillo  
 ADMINISTRADORA  
 S.E.T. T.R.Q.



# PRÁCTICAS PROFECIONALIZANTES – SOLICITUD DE MATERIALES

NÚMERO DE PEDIDO: N1

FECHA: 03/05 /2023

NOMBRE DEL PROYECTO: MIDAS

ESPECIALIDAD: Aviónica

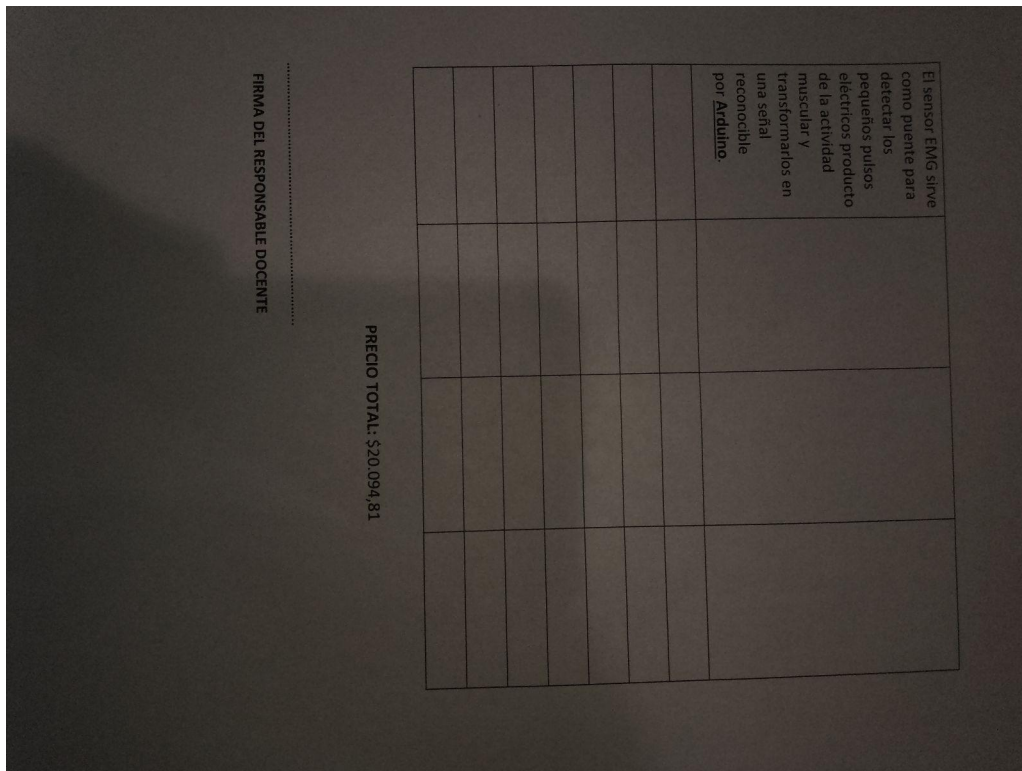
NOMBRE DE INTEGRANTES	DATOS DEL PROVEEDOR
Kevin Joel Ledesma	EMPRESA: Analog Devices
Federico Nicolás Tapia	TELÉFONO: 523336127301
Gabriel Alejandro Jurado	E-MAIL: argentina@mouser.com
Lautaro Horn	FAX: n/a
Santiago Emanuel Espandrio	WEB: ADA4666-2: <a href="https://ar.mouser.com/ProductDetail/Analog-Devices/ADA4666-2ARMZ?qs=pqQDoY22quY1e9pRg1Z%252B6Q%3D%3D">https://ar.mouser.com/ProductDetail/Analog-Devices/ADA4666-2ARMZ?qs=pqQDoY22quY1e9pRg1Z%252B6Q%3D%3D</a> INA129p: <a href="https://ar.mouser.com/ProductDetail/Texas-Instruments/INA129P?qs=pt%2FlvSr0EPeetMaOMVBEGQ%3D%3D">https://ar.mouser.com/ProductDetail/Texas-Instruments/INA129P?qs=pt%2FlvSr0EPeetMaOMVBEGQ%3D%3D</a>
	DIRECCIÓN: Av. Moctezuma 3515, M1 Zapopan, JAL – CP 45050 México

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO PARCIAL
ADA4666-2: un amplificador de entrada/salida duales y de riel optimizado para aplicaciones de rango de voltaje del suministro de bajo consumo, ancho de banda alto y de amplio funcionamiento	Necesitaríamos un total de 4 ADA4666-2	\$551	\$5.204
INA129p: este es un amplificador de	Simplemente se necesita 1	\$1.701,63	\$4.701,63

Instrumentación ideal para las aplicaciones biomédicas	amplificador utilizado para la etapa de la primera amplificación		

NOMBRE DE INTEGRANTES	DATOS DEL PROVEEDOR
	EMPRESA: <u>OEM</u>
	TELÉFONO: +34 661 378 025
	E-MAIL: <u>compras@solectroshop.com</u>
	FAX:
	Web: <a href="https://solectroshop.com/es/sensores-medicos/1100-emg-sensor-de-senal-muscular-electromiografia-signal.html">https://solectroshop.com/es/sensores-medicos/1100-emg-sensor-de-senal-muscular-electromiografia-signal.html</a>
	DIRECCIÓN: Av. Aragón 30 46021 , España

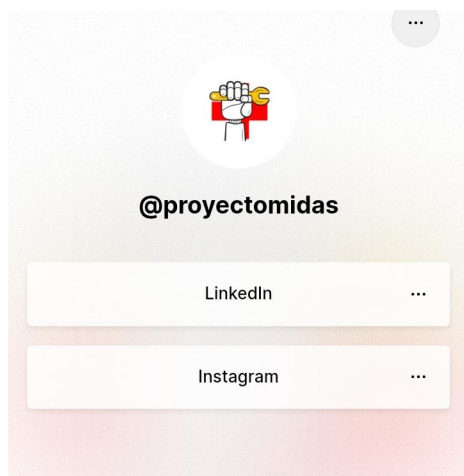
MATERIALES NECESARIOS			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO PARCIAL
EMG Kit de módulo sensor de señal muscular con 6 electrodos desechables.	1	\$9201,68	\$10.189,18



## Junio:

Cambio del funcionamiento del proyecto

Ya que no parece posible ni cómodo el anterior y además no se lograria el estimado que queríamos de movilidad y además nos podríamos haber quedado estancados queriendo llegar a llevar a cabo tal sistema



## Julio:

Creación de las piezas en 3D de la mano

Se buscaron y editaron modelos de mano de exoesqueleto para poder empezar a trabajar sobre eso y que nos sea de mayor facilidad explicar el proyecto

## Reposo de invierno

En este reposo de invierno se llevaron a cabo la creación de PCBs de forma digital para que en la vuelta a clases se realicen las correcciones necesarias para poder imprimir los PCBs y pasarlos a una plancha de cobre y que podamos pasar a la etapa de perforado y soldado de componentes. También se avanza en la producción de software de la mano para que funcione correctamente el movimiento de los servos en conjunto con los acelerómetros.

También ambientamos el fin de este mes en parte de desarrollo de software para ir teniendo un código base el cual cambiar, perfeccionar o usar según las circunstancias el cual es el siguiente que se encargara de lo procesado mandarlo a los servomecanismos, lo procesado seria la señal electro muscular amplificada y pasada a digital por un ADC incluido en la raspberry pi pico para que por los pines gpio pase a los acelerómetros y esos dirijan el movimiento deseado

```
import array
import machine
import math
import time
import utime

# Configurar los pines GPIO para los acelerómetros y los servos
accel_x_pin = machine.ADC(0) # Asegúrate de usar los pines correctos
accel_y_pin = machine.ADC(1)
servo_pin = machine.PWM(machine.Pin(2)) # Ejemplo de configuración de
un servo en el pin GPIO 2

# Función para obtener lecturas del acelerómetro en el eje X
def read_accel_x():
    return accel_x_pin.read_u16()

# Función para obtener lecturas del acelerómetro en el eje Y
def read_accel_y():
```

```

    return accel_y_pin.read_u16()

# Función para procesar la señal EMG (usando un filtro de media móvil)
def process_emg_signal(emg_samples, window_size=5):
    return sum(emg_samples[-window_size:]) / window_size

# Configurar la frecuencia de muestreo de los acelerómetros
accel_sampling_frequency = 100 # 100 muestras por segundo

# Configurar la frecuencia de actualización de los servos
servo_update_frequency = 50 # 50 Hz (50 actualizaciones por segundo)

# Lista para almacenar las muestras EMG
emg_samples = []

# Configurar la duración de la adquisición (en segundos)
acquisition_duration = 10

# Tiempo inicial de la adquisición
start_time = utime.ticks_ms()

# Bucle principal de adquisición y procesamiento
while utime.ticks_diff(utime.ticks_ms(), start_time) <
acquisition_duration * 1000:
    # Leer señales EMG
    emg_value = read_emg()
    emg_samples.append(emg_value)

    # Leer señales del acelerómetro
    accel_x = read_accel_x()
    accel_y = read_accel_y()

    # Aplicar filtrado a la señal EMG
    filtered_emg_value = process_emg_signal(emg_samples)

    # Realizar cálculos para controlar los servos según la intensidad
    de la señal EMG

    # Aquí puedes implementar tu lógica para controlar los servos según
    las señales EMG y los valores de los acelerómetros

    # Por ejemplo, si la señal EMG es alta, podrías mover el servo
    hacia una posición específica

```

```
# y si la señal es baja, podrías mover el servo hacia otra
posición.

# Puedes utilizar el método duty() para configurar el ciclo de
trabajo del PWM y controlar la posición del servo
# Por ejemplo, si servo_pin es el objeto que representa al servo,
puedes hacer lo siguiente:
# servo_pin.duty(500) # Configura el ciclo de trabajo para una
posición específica

utime.sleep_ms(1000 // accel_sampling_frequency) # Esperar para
mantener la frecuencia de muestreo de los acelerómetros

# Detener los servos al finalizar el bucle
servo_pin.duty(0)

# Configurar los pines GPIO para las entradas analógicas
emg_pin = machine.ADC(26)
reference_pin = machine.ADC(27)
ground_pin = machine.ADC(28)

# Configurar la frecuencia de muestreo
sampling_frequency = 1000 # 1000 muestras por segundo

# Configurar el tamaño de la ventana del filtro
filter_window_size = 5

# Lista para almacenar las muestras EMG
emg_samples = []

# Configurar la duración de la adquisición (en segundos)
acquisition_duration = 10

# Tiempo inicial de la adquisición
start_time = utime.ticks_ms()

# Bucle principal de adquisición
while utime.ticks_diff(utime.ticks_ms(), start_time) <
acquisition_duration * 1000:
    emg_value = read_emg()
    emg_samples.append(emg_value)
```



```

    utime.sleep_us(1000000 // sampling_frequency) # Esperar para
mantener la frecuencia de muestreo

    # Aplicar filtrado a la lista de muestras
    filtered_emg_value = apply_filter(emg_samples, filter_window_size)

    # Aquí puedes agregar más procesamiento y análisis de las señales
    EMG, según tus necesidades específicas.

# Puedes mostrar los resultados en la consola o enviarlos a un
dispositivo externo para su visualización.
print(emg_samples)

# Función para obtener lecturas de los electrodos
def read_emg():
    return emg_pin.read_ul6()

# Función para aplicar un filtro de media móvil para suavizar la señal
def apply_filter(signal, window_size=5):
    return sum(signal[-window_size:]) / window_size

# Configuración de los pines de entrada de la señal EMG
emg_pin = machine.Pin(0, machine.Pin.IN, machine.Pin.PULL_UP)

# Creación de un array para guardar los datos EMG
buffer_size = 1000
emg_buffer = array.array('H', [0] * buffer_size)

# Función para leer la señal EMG
def read_emg_signal(samples):
    for i in range(samples):
        emg_buffer[i] = emg_pin.value()
    emg_data = emg_buffer[:samples]
    return emg_data

# Configurar los pines GPIO para los electrodos EMG
emg_pin = machine.ADC(0) # Asegúrate de usar el pin correcto

# Configurar los pines GPIO para los acelerómetros
accel_x_pin = machine.ADC(1)
accel_y_pin = machine.ADC(2)

```

```

# Configurar el pin GPIO para el servo
servo_pin = machine.PWM(machine.Pin(3)) # Ejemplo de configuración de
un servo en el pin GPIO 3

# Función para obtener la lectura del electrodo EMG
def read_emg():
    return emg_pin.read_u16()

# Función para obtener las lecturas del acelerómetro en los ejes X e Y
def read_accel():
    accel_x = accel_x_pin.read_u16()
    accel_y = accel_y_pin.read_u16()
    return accel_x, accel_y

# Función para procesar la señal EMG (usando un filtro de media móvil)
def process_emg_signal(emg_samples, window_size=5):
    return sum(emg_samples[-window_size:]) / window_size

# Configurar la frecuencia de muestreo de los electrodos EMG y los
acelerómetros
emg_sampling_frequency = 500 # 500 muestras por segundo
accel_sampling_frequency = 100 # 100 muestras por segundo

# Lista para almacenar las muestras EMG
emg_samples = []

# Configurar la duración de la adquisición (en segundos)
acquisition_duration = 10

# Tiempo inicial de la adquisición
start_time = utime.ticks_ms()

# Bucle principal de adquisición y procesamiento
while utime.ticks_diff(utime.ticks_ms(), start_time) <
acquisition_duration * 1000:
    # Leer señales EMG
    emg_value = read_emg()
    emg_samples.append(emg_value)

    # Leer señales de los acelerómetros
    accel_x, accel_y = read_accel()

```

```

# Aplicar filtrado a la señal EMG
filtered_emg_value = process_emg_signal(emg_samples)

# Calcular la magnitud de la aceleración (se puede usar la suma de
los valores de los ejes X e Y)
accel_magnitude = accel_x + accel_y

# Realizar cálculos para controlar los servos según la intensidad
de la señal EMG y la aceleración
# Aquí puedes implementar tu lógica para controlar los servos según
las señales EMG y los valores de los acelerómetros

# Por ejemplo, si la señal EMG es alta y la aceleración es baja,
podrías mover el servo hacia una posición específica
# Si la señal EMG es baja y la aceleración es alta, podrías mover
el servo hacia otra posición.

# Puedes utilizar el método duty() para configurar el ciclo de
trabajo del PWM y controlar la posición del servo
# Por ejemplo, si servo_pin es el objeto que representa al servo,
puedes hacer lo siguiente:
# servo_pin.duty(500) # Configura el ciclo de trabajo para una
posición específica

# Puedes utilizar las variables filtered_emg_value y
accel_magnitude para definir tu lógica de control

utime.sleep_ms(1000 // emg_sampling_frequency) # Esperar para
mantener la frecuencia de muestreo de los electrodos EMG

# Detener los servos al finalizar el bucle
servo_pin.duty(0)

# Configuración del acelerómetro
i2c = machine.I2C(0, scl=machine.Pin(9), sda=machine.Pin(8),
freq=100000)
i2c.writeto(0x18, bytes([0x20, 0x0F])) # Configuración del acelerómetro
i2c.writeto(0x18, bytes([0x23, 0x80])) # Configuración del acelerómetro

# Leer los datos del acelerómetro
data = bytearray(6)
i2c.readfrom_mem_into(0x18, 0x28 | 0x80, data)

```

```

x = (data[1] << 8 | data[0]) / 16384.0
y = (data[3] << 8 | data[2]) / 16384.0
z = (data[5] << 8 | data[4]) / 16384.0

# Procesar los datos y obtener la inclinación del dispositivo
inclinacion = math.atan2(x, math.sqrt(y*y + z*z)) * 180 / math.pi
print("Inclinación: %.2f grados" % inclinacion)

# Configuración de los pines de los servos
servo1 = machine.PWM(machine.Pin(0))
servo2 = machine.PWM(machine.Pin(1))
servo3 = machine.PWM(machine.Pin(2))
servo4 = machine.PWM(machine.Pin(3))
servo5 = machine.PWM(machine.Pin(4))
servo1.freq(50)
servo2.freq(50)
servo3.freq(50)
servo4.freq(50)
servo5.freq(50)

# Función para calcular el ángulo del servo a partir de la inclinación
def calcular_angulo(inclinacion):
    # Ajustar los valores de acuerdo a los requerimientos de cada servo
    # específico
    return int(90 + inclinacion * 2)

# Bucle infinito para controlar los servos
while True:
    # Leer los datos del acelerómetro
    # Código para leer los datos del acelerómetro

    # Procesar los datos y obtener la inclinación del dispositivo
    inclinacion = math.atan2(x, math.sqrt(y*y + z*z)) * 180 / math.pi

    # Calcular el ángulo correspondiente para cada servo
    angulo1 = calcular_angulo(inclinacion)
    angulo2 = calcular_angulo(inclinacion)
    angulo3 = calcular_angulo(inclinacion)
    angulo4 = calcular_angulo(inclinacion)
    angulo5 = calcular_angulo(inclinacion)

    # Ajustar los ángulos de los servos
    servo1.duty_ns(500000 + angulo1 * 1000000 // 180)

```

```
servo2.duty_ns(500000 + angulo2 * 1000000 // 180)
servo3.duty_ns(500000 + angulo3 * 1000000 // 180)
servo4.duty_ns(500000 + angulo4 * 1000000 // 180)
servo5.duty_ns(500000 + angulo5 * 1000000 // 180)

# Esperar un tiempo antes de volver a medir la inclinación y
ajustar los servos
time.sleep(0.5)
```

## **Agosto:**

Configurado del software de impresión 3D

Se configuró la impresora 3D ya que había estado en desuso y al calibrar y ajustar en la calidad y temperatura requerida se mejoraría la velocidad y calidad en la impresión

Impresión de las piezas 3D

Se inició la impresión 3D de la base para comenzar a realizar pruebas sobre una mano y armar el sistema de funcionamiento sobre esta misma

## **Septiembre:**

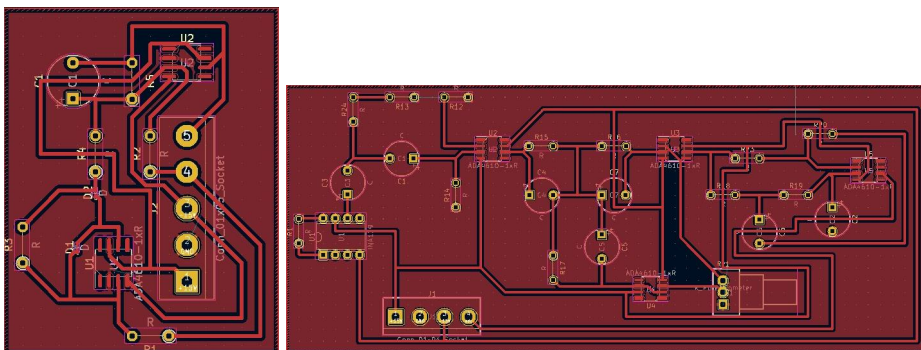
Soldado de Pines de la Raspberry Pi Pico

Se realizó el soldado de pines debido a que el microprocesador no trae pines incluidos en este modelo adquirido y para comenzar con la identificación de salida y pines necesarios para el proyecto



## Corrección de PCB

Debido a que el tamaño de las pistas perjudicaba la integridad de la placa y facilitaba el que haya puentes, cortocircuitos o errores varios, además los pines no serían eficientes

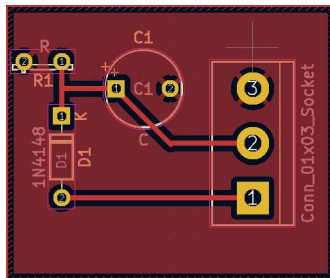
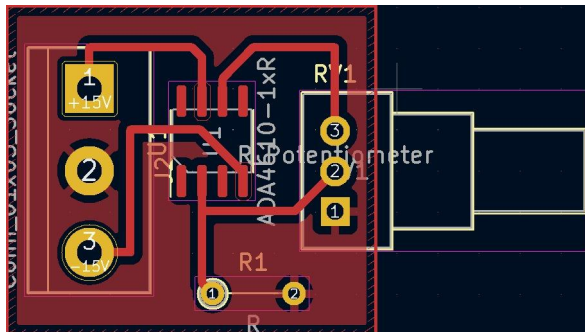


**Octubre:**

Última corrección de PCB's



Ya que en varios casos el grosor de las pistas no resultaba eficiente ya que si eran muy finas podrían ocasionar problemas más adelante o podría ser más propensa a fallos y además también debíamos corregir el tamaño de ciertos componentes con sus pines



Impresión de los 4 PCB

Para proceder a pasarlos a la plancha para comenzar a trabajar correctamente

Pasado 4 PCB a plancha

De modo que ya podemos empezar a trabajar con los componentes en esas pistas ya que también se llevó a cabo el perforado de pines



### Simulaciones acabadas

De modo que ya estamos seguros de que los valores son los conseguidos y que el circuito va a funcionar con la composición que tiene y sentirnos más seguros con el armado de PCB, pero con el tiempo es probable que surjan imprevistos que haya que arreglar sobre el pcb ya que es algo armado espontáneamente debido al tiempo y a los imprevistos que hubo a lo largo del recorrido y ya con los valores que son los que conseguimos en si despues ya estaríamos mas encaminados a que el circuito sea menos probable de fallar

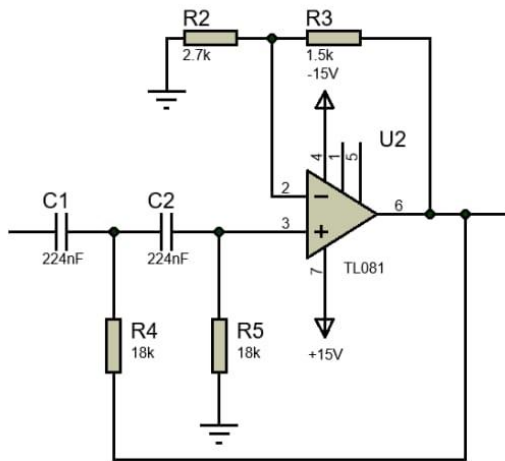


Figura 67: filtro paso alto simulado

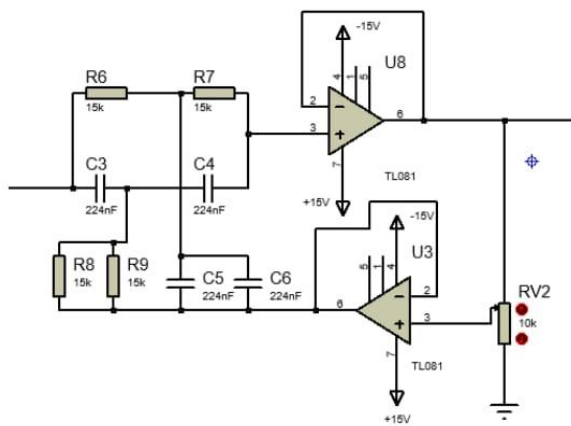


Figura 71: filtro Notch simulado

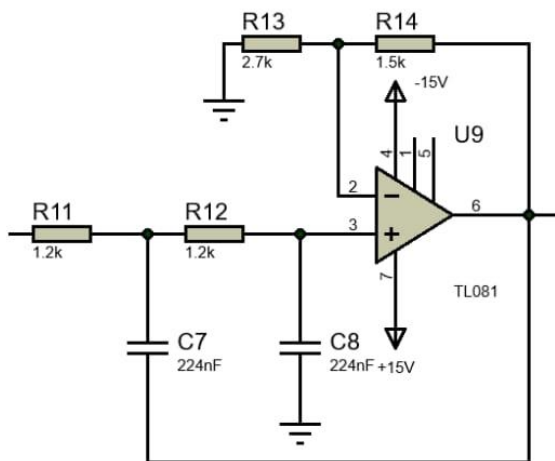


Figura 75: filtro paso bajo simulado

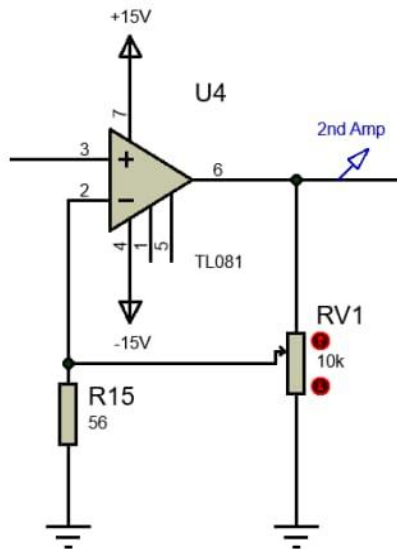


Figura 79: segunda amplificación simulada

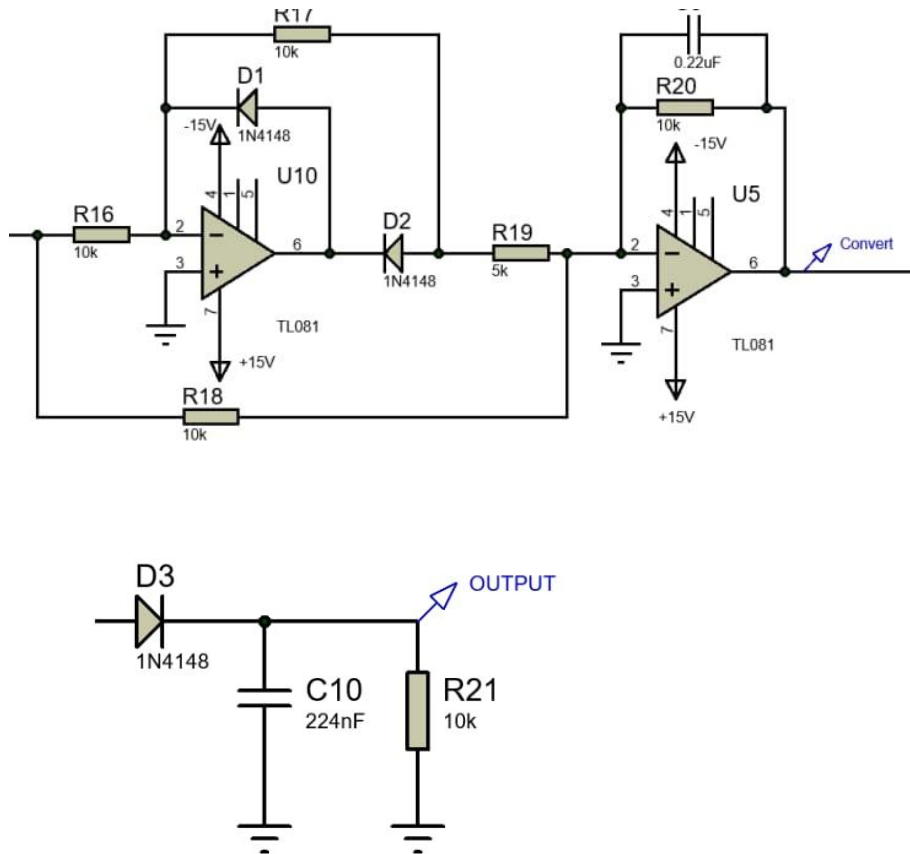


Figura 84: detector de envoltura simulado

Comprobar continuidad de las placas

Esto para que nos aseguremos de que no ocurra un cortocircuito ni haya pistas que estén dañadas ni muy juntas de modo que se confirme la integridad de la placa.

## Noviembre:

### Ubicado y Soldado de Pines

Se llevó a cabo de modo que ya quede todo para su pronta prueba de funcionamiento y aún faltan unos detalles para que quede completo

### Verificación de Continuidad de la Placa

Esto se realiza para comprobar que las pistas no se toquen y haya continuidad en las pistas requeridas para que se vean concretadas las funciones deseadas del modo correcto

### Revisión, Corrección y Terminado de Ensamblado y Conectado

Básicamente nos pusimos a soldar y conectar componentes, corregir por encima errores en las placas y enmendarlos con conexiones por sobre el pcb como divisores de voltaje, cables para ciertos componentes que no pudimos conseguir de tal medida y soldar componentes que previamente no teníamos y recién sobre la fecha se pudieron conseguir para poder estar más cerca de concretar la función del proyecto

