“UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA”
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR
previo a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

“RECONOCIMIENTO DE GESTOS MANUALES BASADO EN APRENDIZAJE DE MÁQUINA A TRAVÉS DE DATOS EMG”

AUTOR:
SERPA TOLEDO DAVID MOISÉS

PROFESOR O TUTOR SUGERIDO:
ING. LUIS ENRIQUE CHUQUIMARCA JIMÉNEZ, MSC.

LA LIBERTAD - ECUADOR
2023 - 2024

AGRADECIMIENTO

En primer lugar, agradezco de manera especial al Divino Niño Jesús, como de cariño que le sé decir niño Manuelito, muchas gracias por tu guía divina, fortaleza y por haberme iluminado en cada etapa de este camino académico y personal.

Quiero agradecer sinceramente a mi querida madre, Nancy Serpa, por su amor incondicional y todo el apoyo que me ha brindado durante toda mi carrera. Sin ti, no hubiera sido posible llegar hasta aquí. ¡Muchas gracias de todo corazón!

Quiero agradecer a mi tía Elsa Serpa por su apoyo y aliento a lo largo de mi carrera académica. Sin usted, no hubiera podido culminar mis estudios. ¡Muchas gracias de corazón!

A mis primos, José Valencia y Paulina Valencia por brindarme su tiempo y así culminar con mi proyecto.

A mis amigos, Cesar Chancay y el Ing. Bryan Borbor, por ayudarme con la implementación de mi proyecto.

A mis tutores, el Ing. Luis Chuquimarca, Msc. Y el Ing. Óscar Gómez, Msc, por brindarme sus conocimientos y su notable paciencia entregada para culminar el proyecto.

David Moisés Serpa Toledo.

DEDICATORIA

Dedico el presente proyecto de titulación a mi querida madre, que con su amor y esfuerzo ha podido sacarme adelante.

A toda mi familia, por los consejos que me brindaron desde pequeño, por su comprensión y paciencia, y por estar siempre presentes en cada momento importante de mi vida. Este logro es tanto mío como suyo.

David Moisés Serpa Toledo.

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado: "Reconocimiento de gestos manuales basado en aprendizaje de máquina a través de datos EMG", elaborado por el estudiante Serpa Toledo David Moisés, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

La Libertad, 03 julio del 2024

Ing. Luis Enrique Chuquimarca Jiménez. Mgt.

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Rovira Jurado, PhD.
DIRECTOR DE LA CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

Ing. Óscar Gómez Morales, MBA.
DOCENTE ESPECIALISTA

Ing. Luis Chuquimarca Jiménez, MSc.
DOCENTE TUTOR
DOCENTE GUÍA UIC

Ing. Corina Gonzabay De La A, Mgt.
SECRETARIA

RESUMEN

Este trabajo de investigación se enfoca en el desarrollo de un sistema para el reconocimiento de gestos manuales basado en señales electromiográficas (EMG). Se diseñó y fabricó un sensor mioeléctrico específico para la adquisición precisa de señales EMG, complementado con un prototipo 3D de una prótesis de mano izquierda para el reemplazo de la extremidad. Utilizando una Raspberry Pi y un Arduino UNO, se implementaron algoritmos de aprendizaje de máquina basados en redes neuronales para procesar y clasificar los datos EMG capturados.

La metodología incluyó la captura de datos de tres gestos principales: mano cerrada/puño, pinza entre el dedo índice y el pulgar, y posición de descanso natural. Estos gestos fueron registrados mediante electrodos no invasivos ubicados en el antebrazo del usuario. Además, se llevó a cabo un exhaustivo proceso de procesamiento de señales y extracción de características relevantes para entrenar y validar el modelo de aprendizaje de máquina.

Los resultados experimentales demostraron una precisión en la clasificación de gestos del 97%, validando la viabilidad y eficacia del sistema propuesto. Este enfoque no solo representa un avance significativo en el campo del reconocimiento de gestos basado en EMG, sino que también promueve aplicaciones prácticas en rehabilitación médica, control de prótesis y tecnologías de interacción hombre-máquina.

Palabras claves: Sensor mioeléctrico, reconocimiento de gestos manuales, redes neuronales, extracción de características, aprendizaje de máquina.

ABSTRACT

This research focuses on developing a system for manual gesture recognition based on electromyographic (EMG) signals. A specific myoelectric sensor was designed and manufactured for precise EMG signal acquisition, complemented by a 3D prototype of a left-hand prosthesis for limb replacement. Using a Raspberry Pi and an Arduino UNO, machine learning algorithms based on neural networks were implemented to process and classify the captured EMG data.

The methodology included capturing data for three main gestures: closed hand/fist, pinch between the index finger and thumb, and natural rest position. These gestures were recorded using non-invasive electrodes placed on the user's forearm. Additionally, a thorough process of signal processing and feature extraction was conducted to train and validate the machine learning model.

Experimental results demonstrated a 97% accuracy in gesture classification, validating the feasibility and effectiveness of the proposed system. This approach not only represents a significant advancement in the field of EMG-based gesture recognition but also promotes practical applications in medical rehabilitation, prosthesis control, and human-machine interaction technologies.

Key word: myoelectric sensor, Hand gesture recognition, neural networks, feature extraction, machine learning.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

ÍNDICE DE ECUACIONES

INTRODUCCIÓN

En la actualidad, el reconocimiento de gestos manuales ha captado un gran interés en diversas aplicaciones, tales como la rehabilitación médica, la realidad virtual, la interacción hombre-máquina y el control de prótesis. Las técnicas convencionales para este propósito pueden enfrentar desafíos significativos en términos de precisión y adaptabilidad a diferentes condiciones ambientales. En este contexto, el uso de señales electromiográficas (EMG) emerge como una alternativa prometedora debido a su capacidad para captar la actividad eléctrica generada por los músculos durante los movimientos y gestos.

Las señales EMG ofrecen una representación directa de la intención muscular, lo que las hace especialmente útiles para el reconocimiento de gestos manuales. No obstante, la naturaleza compleja y ruidosa de estas señales plantea desafíos significativos en términos de procesamiento y clasificación.

El presente trabajo se centra en el desarrollo de un sensor mioeléctrico capaz de adquirir señales EMG. Además, se ha diseñado un prototipo de prótesis de mano izquierda para el reemplazo de la extremidad. Por último, se ha creado un sistema de reconocimiento de gestos manuales basado en el aprendizaje de máquina, utilizando los datos EMG obtenidos. La metodología propuesta incluye la adquisición de señales EMG mediante el sensor mioeléctrico, el procesamiento de estas señales, la extracción de características relevantes y la implementación de un modelo de aprendizaje de máquina para la clasificación de gestos.

El objetivo principal de este proyecto es diseñar y validar un sistema capaz de reconocer de manera precisa y eficiente distintos gestos manuales a partir de las señales EMG registradas. Para ello, se han recopilado datos de tres tipos de gestos: mano cerrada/puño, pinza entre el dedo indicé y pulgar, y posición de descanso natural, los cuales fueron capturados a través de electrodos no invasivos ubicados en el antebrazo del paciente.

CAPÍTULO I

1.1 Antecedentes

En los años recientes, se han llevado a cabo una serie de investigaciones en el ámbito de reconocimiento de gestos manuales empleando técnicas de aprendizaje de máquina, con la implementación de sensores EMG. Por otro lado, se aplicó distintas técnicas para la obtención de datos y procesamiento de señales eléctricas producidas por los músculos durante una contracción muscular, y finalmente se aplicaron distintas clases de algoritmos para el aprendizaje de máquina en función de las distintas señales obtenidas.

En primer lugar, el reconocimiento de gestos tiene diversas aplicaciones en los campos de medicina, ingeniería y robótica. Dicho esto, el reconocimiento de gestos de mano en tiempo real implica la capacidad de identificar sin demora perceptible un gesto particular realizado con la mano en cualquier momento. Para abordar este desafío, se presentó un modelo que utiliza señales electromiográficas (EMG – *Electromyography*) capturadas en el antebrazo mediante el uso de un sensor llamado Myo Armband. Como resultado, se obtuvo un codificador que permite extraer características automáticamente y una red neuronal artificial que se encarga de la clasificación de los gestos [1].

En segundo lugar, los gestos son uno de los agentes más importantes para la interacción humano-computadora. Se ha estudiado el reconocimiento de patrones de electromiografía (EMG - *Electromyography*) para la identificación de gestos durante muchos años para el control de prótesis y sistemas de rehabilitación [2].

En tercer lugar, el procesamiento de señales de electromiografía (EMG – *Electromyography*) se desarrolló para controlar los dispositivos médicos de asistencia para la rehabilitación clínica. El objetivo era determinar y comparar el rendimiento de diferentes algoritmos de aprendizaje de máquina (ML – *Machine Learning*) basados en redes neuronales artificiales en el procesamiento de señales EMG de superficie multicanal (sEMG – *Surface Electromyography*) [3].

Un motivo adicional para la importancia de este tipo de investigaciones es la necesidad de personalización en el ámbito de las prótesis, ya que cada

individuo presenta diferentes tipos de amputaciones con tamaños heterogéneos de miembros residuales, lo que implica que requieren niveles distintos de funcionalidad. Por lo tanto, es común que cada prótesis se fabrica de manera personalizada según las necesidades específicas de cada persona.

1.2 Descripción del proyecto

Esta propuesta tecnológica se enfoca hacia las personas que sufrieron una pérdida de una de las extremidades (manos); de hecho, el servicio médico en Ecuador no incluye en sus planes de salud la adaptación de una prótesis a una persona que haya perdido un miembro, es decir, la pérdida de un pie, una pierna o un brazo. Además, la mayoría de los empleos requieren el uso de ambas extremidades superiores y el alto costo de las prótesis dificultan que el sistema de salud satisfaga esta necesidad y las personas no puedan permitirse adquirirlas en el mercado, debido a la falta de prototipos de bajo costo.

Se planeó desarrollar un prototipo de una prótesis de miembro superior izquierdo (mano) que utilizará elementos electrónicos de bajo costo, pero que ofrecerá resultados comparables a los de una prótesis fabricada en el país.

El modelo del prototipo se realizará en el software Autodesk Fusion 360. Además, se diseñará un sensor mioeléctrico (EMG) que implementará amplificadores operacionales para amplificar la señal EMG. También se diseñarán diversos filtros para eliminar los armónicos y ruidos presentes en las señales EMG.

La señal EMG se adquirirá mediante un microcontrolador Arduino UNO, que se encargará de la extracción de características y enviará los datos procesados a una Raspberry Pi para su almacenamiento. El sensor EMG, junto con electrodos no invasivos colocados en el músculo del antebrazo del paciente, permitirá la obtención de las señales necesarias.

Finalmente, se realizará el entrenamiento del conjunto de señales utilizando redes neuronales en Python, todo este proceso lo realizará la Raspberry Pi. Los resultados de las predicciones se envían al Arduino UNO quien va a hacer el encargado de controlar los actuadores implementados en el prototipo de la prótesis de mano izquierda.

1.3 Objetivos del proyecto

1.3.1 Objetivo General

Aplicar el reconocimiento de gestos manuales basado en aprendizaje de máquina a través de toma de señales eléctricas del músculo del antebrazo por medio de sensores mioeléctricos EMG y electrodos no invasivos, para el movimiento de una prototipo 3D.

1.3.2 Objetivos Específicos

- Diseñar un prototipo de una prótesis de mano izquierda en el software Autodesk Fusion 360, para el reemplazo de la extremidad.
- Diseñar e implementar un sensor mioeléctrico EMG para la obtención de las señales eléctricas musculares.
- Desarrollar un algoritmo en el software Python para el aprendizaje de máquina de las señales obtenidas del músculo.
- Implementar el prototipo de la prótesis con la ayuda de una Raspberry PI para el control de los servos motores.

1.4 Justificación

Según los datos del Banco Mundial (DM), existen alrededor del 15% de la población mundial, personas que experimentan algún tipo de discapacidades, es decir, va entre 110 millones y 190 millones de personas, o sea la quinta parte de la población se ven afectadas por discapacidades importantes [4].

En Ecuador existen alrededor de 15.952.442 habitantes. De estas 563.515 personas poseen discapacidades físicas, psicosocial o sensorial, es decir, representa el 3.5% de la población nacional; asimismo, se encuentran distribuidas en su mayoría en la zona urbana con el 62.4% y el 37.6% restante se encuentra en la zona rural [5].

Según el Ministerio de Salud Pública del Ecuador (MPS), existe el 39.7% de personas con discapacidades físicas. Eso da a entender que son aquellas que ocurre al faltar o quedar muy poco de una parte del cuerpo, lo cual impide a la persona desenvolverse de la manera convencional [5].

El impacto será dar a conocer la propuesta investigativa acerca del reconocimiento de gestos manuales basados en aprendizaje de máquina, e implementarlo a un prototipo de una prótesis de miembro superior izquierda (mano), con el fin de proporcionar ayuda hacia las personas que han sufrido amputaciones traumáticas por consecuencia de algún accidente, por lo tanto, el objetivo es motivar para que tengan una mayor facilidad en sus actividades diarias y poder mejorar la calidad de vida.

1.5 Metodología

El presente proyecto se realizará utilizando los siguientes tipos de investigación:

1.5.1 Investigación Exploratoria.

Se aplicará este tipo de investigación para la recolección de las señales musculares del antebrazo y guardar en una base de datos, para poder realizar el aprendizaje de máquina.

1.5.2 Investigación Aplicativa

Este tipo de investigación se aplicará en el transcurso de la elaboración del proyecto, porque se utilizarán elementos electrónicos que podemos encontrar con mayor facilidad en el mercado, dando como resultado un dispositivo funcional.

1.5.3 Investigación Bibliográfica

Se aplicará este tipo de investigación para la recolección de datos de diferentes documentos como tesis y artículos científicos, relacionados con el tema de señales electromiográficas EMG con la aplicación de aprendizaje de máquina, gracias con la información obtenida poder aplicar los conocimientos adquiridos de los diferentes documentos examinados, sobre todo, para conocer y dar un correcto funcionamiento de los dispositivos que se van a emplear.

CAPÍTULO II

2.1 Marco Contextual

En Ecuador, existe más del 8% de la población tiene necesidades especiales en los miembros superiores y/o inferiores, debido a malformaciones, amputaciones médicas o accidentes, según lo detalla Alberto Larrea, coordinador del plan Jubilous3D [6]. Además, la mayoría de esta población no tiene los recursos suficientes para costearse unas prótesis producidas en el país, ya que el costo de una prótesis está alrededor de los 8 mil dólares [7]. Por otro lado, el acceso limitado a las prótesis pasivas no contribuye en gran parte a la solución del problema, ya que son más estéticas que funcionales, por lo que las personas optan por elegir las prótesis activas que son controladas por electromiografía, para poder realizar actividades que forman parte de la vida diaria.

Actualmente en Ecuador existen empresas dedicadas al diseño y fabricación de prótesis de miembros superiores, tales como: ROMP, Proteus, Ecuaprótesis3D, CRIE (*Centro de Rehabilitación Integral Especializado*), Ormedic entre otros [7]. Algunas de estas empresas producen prótesis pasivas que no satisfacen las necesidades básicas del paciente, ya que muchas priorizan la estética, dejando a un lado la funcionalidad y precisión.

Con la implementación de un prototipo de una prótesis de mano controlada por electromiografía, a su vez aplicando el aprendizaje de máquina de cada una de las señales eléctricas producidas del antebrazo, se espera generar un aporte significativo al mercado de prótesis activas. De esta manera, se podrá contribuir a la iniciativa de producirlas a un menor costo, facilitando el acceso a las personas que más lo necesitan.

2.2 Marco Conceptual

2.2.1 Señales Biológicas

Las señales biológicas, o bioseñales, son registros de espacio, tiempo o espacio-tiempo de un evento biológico, como un corazón que late o un músculo que se contrae, además, la actividad eléctrica, química y mecánica que ocurre durante este evento biológico a menudo produce señales que pueden medirse y analizarse; por lo tanto, contienen información útil que se puede utilizar para comprender los mecanismos fisiológicos subyacentes de un evento o sistema biológico específico y que puede ser útil para el diagnóstico médico [8].

Las señales biológicas se pueden adquirir de varias maneras, por ejemplo, por un médico que usa un estetoscopio para escuchar los sonidos del corazón de un paciente o con la ayuda de instrumentos biomédicos tecnológicamente avanzados [8].

2.2.2 Señales mioeléctricas

Las señales eléctricas dentro del cuerpo humano que estimulan el movimiento de los músculos. Esta señal es inferior a un milivoltio y tiene una frecuencia media de unos $100\ Hz$, por lo tanto, dichas señales mioeléctricas se utilizan para mover prótesis; asimismo, las señales débiles se detectan y amplifican para controlar los elementos móviles de las prótesis que funcionan con baterías. La palabra myo proviene del griego “myos” y significa “músculo” [9].

2.2.3 Obtención de las señales mioeléctricas

Para la obtención de las señales mioeléctricas se deben utilizar amplificadores de instrumentación y amplificadores operacionales diferenciales que reducen interferencias como ruido. Por lo tanto, la identificación de una señal EMG real que se origina en el músculo se pierde debido a la mezcla de varias señales de ruido o artefactos; además, una señal EMG depende de la estructura interna del sujeto, incluida la formación de piel individual, la velocidad del flujo sanguíneo, las temperaturas de la piel medidas, etc. Todo esto produce diferentes tipos de señales de ruido que se pueden encontrar dentro de las señales EMG [10].

La adquisición de señales mioeléctricas dependerá de tres factores: se inicia con la calidad del amplificador y electrodos que se utilice, continua con el manejo de la reducción de ruido ambiental; lo adecuado sería de 1 a 2 microvoltios y, por último, el estado de región muscular humana de donde se desea adquirir la señal; existe zonas más flexibles, sensitivas y más móviles en el cuerpo humano [11].

2.2.4 Amplificadores y electrodos

La función de los amplificadores operacionales (AD) es tener una diferencia entre sus dos entradas, entrada no inversora Vin(+) y entrada inversora Vin(-) para luego ser multiplicada por una ganancia del amplificador. La amplificación de señales analógicas ayuda notablemente en la reducción del ruido, ayudando a la adquisición de señales EMG [11].

$$Vout = Ad(Vin(+) - Vin(-))$$

Ecuación 1. Ecuación para calcular el voltaje de salida del Amplificador.

Donde:

- $\boldsymbol{Vout}$ es el voltaje de salida del amplificador.
- $\boldsymbol{Ad}$ es la ganancia diferencial del amplificador.
- $\boldsymbol{Vin}(+)$ es el voltaje en la entrada no inversora.
- $\boldsymbol{Vin}(-)$ es el voltaje en la entrada inversora.

Para registrar la señal EMG, el uso de electrodos no invasivos se aplica a la piel del sujeto; se ha encontrado que los electrodos hechos de plata/cloruro de plata (10 x 1 mm) dan una señal-ruido adecuada y son eléctricamente muy estables. Por esta razón, son ampliamente utilizados como electrodos de superficie. Cuando el tamaño del electrodo aumenta, la impedancia disminuye; sin embargo, el tamaño del electrodo no debe ser muy grande [10].

2.2.5 Electromiografía o EMG

La electromiografía o EMG se utiliza para medir los potenciales espontáneos o evocados de los músculos. Se pueden insertar electrodos de aguja en los músculos o se pueden usar electrodos de superficie; entonces, el paciente al realizar un movimiento provocar un potencial espontáneo, o se puede estimular el nervio que inerva el músculo. Los potenciales medidos van desde

menos de $50\,uV$ hasta $20-30\,mV$ y un ancho de banda amplificada tiene un rango de $50-5\,kHz$ [12].

2.2.6 Sistema muscular

El sistema muscular consiste en varios tipos de músculos, cada uno de los cuales juega un papel crucial en la función del cuerpo [13]. Además, el sistema muscular tiene más de 600 músculos que trabajan juntos para permitir el pleno funcionamiento del cuerpo [14].

La mayoría de los músculos están controlados a voluntad por el sistema nervioso central del individuo, aunque muchos otros responden de manera refleja, como en el caso de los músculos cardíacos que continúan contrayéndose mientras viva el organismo [15].

Tipos de músculos

El cuerpo humano está compuesto por tres tipos de músculos, estos son:

- **Músculo esquelético**

Es un tejido abundante que se encuentra en varias partes del cuerpo humano, y es el encargado de producir las contracciones; además, este tipo de músculos son los que en su mayoría se realizan de manera voluntaria cuando necesitamos movernos [16].

- **Músculo liso**

Están ubicados en las paredes de los órganos viscerales huecos (como el hígado, el páncreas y los intestinos), a excepción del corazón, tienen apariencia estriada y también están sujetos al control no intencional [17].

- **Músculo cardiaco**

Ubicado solo en el corazón, el músculo cardíaco bombea sangre por todo el cuerpo y estimula sus propias contracciones para crear los latidos cardíacos. Asimismo, las señales del sistema nervioso controlan la tasa de contracción, por lo que este tipo de actividad muscular es involuntaria [14].

2.2.7 Contracción muscular

Los músculos están compuestos en su interior principalmente por fibras, filamentos y por las proteínas actina, miosina, la tropomiosina y la mioglobina, todas necesarias para que la contracción muscular se dé adecuadamente [18].

La contracción muscular es un suceso organizado y natural que se produce a nivel muscular cuando las fibras y proteínas que contiene se combinan y crean tensión en la zona, provocando el movimiento del músculo esquelético o cardíaco [19].

2.2.8 Sistemas de aprendizaje de máquina

El aprendizaje de máquina consiste en una serie de técnicas de la Inteligencia Artificial (IA) que tienen como objetivo hacer predicciones basadas en datos a partir de datos de entrada que se introducen en modelos matemáticos definidos de manera supervisada o sin supervisar. El enfoque de este paradigma de implementación de predictores se basa en el uso de datos como fuente de conocimiento sobre las etiquetas a predecir, en lugar del conocimiento del sistema en sí, de tal manera que, teniendo un conjunto de datos de entrada y conociendo las características de estos, se implementará un modelo de aprendizaje que se adapte correctamente a los mismos [20].

A continuación, se van a describir algunos temas que forman parte del aprendizaje de máquina, estos son:

- Aprendizaje supervisado: consiste en proporcionar al modelo datos de entrada etiquetados con los correspondientes datos de salida, para que el modelo pueda comprobar si las predicciones que hace son correctas y corregirlas en caso contrario [20].
- Aprendizaje no supervisado: el modelo tiene datos sin etiquetar en la entrada, por lo que no tiene referencia sobre lo que ha de obtener. El algoritmo tiene que intentar entender por sí mismo [20].
- Regresión lineal: permite crear un modelo lineal; en su forma más simple ajusta un resultado de entrada a un resultado estimado mediante una línea recta de pendiente y offset determinado [20].

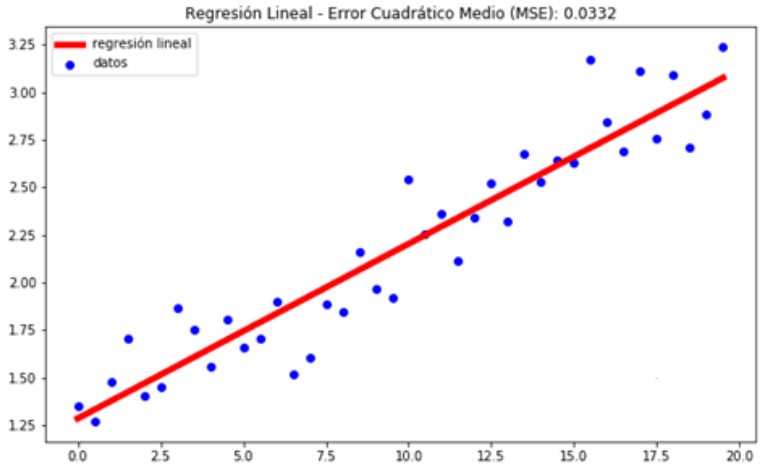


Figura 1. Regresor lineal.

Fuente. *[21]*

- Redes neuronales: el objetivo principal del modelo es aprender modificándose automáticamente, lo que permite realizar tareas complejas que la programación clásica basada en reglas no puede realizar. De esta manera, se pueden automatizar funciones que en un principio solo las personas podían realizar [22].

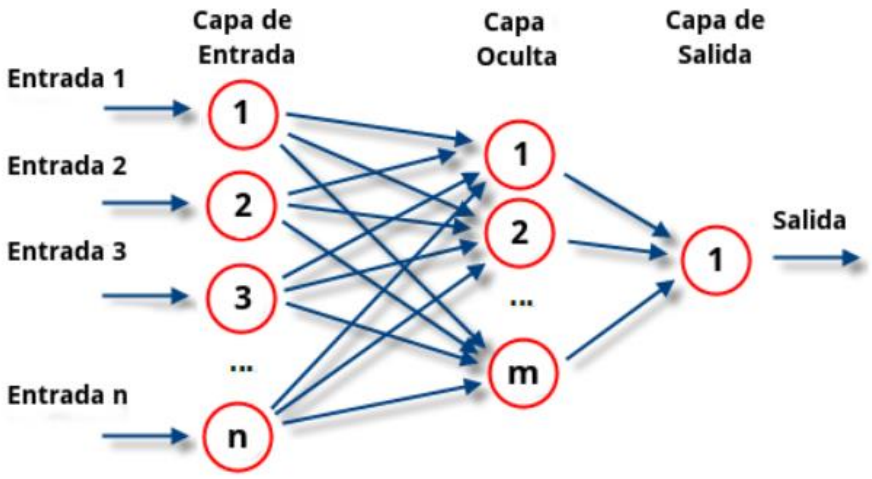


Figura 2. Red neuronal.

Fuente. [22]

- Perceptrón multicapa: es una red neuronal unidireccional constituida por múltiples capas; una capa de entrada, otra capa de salida y el resto de las capas intermedias llamadas capas ocultas [23].

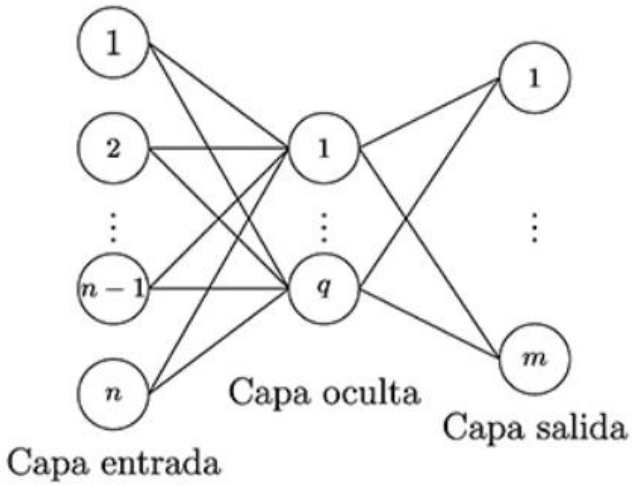


Figura 3. Perceptrón Multicapa (MLP).

Fuente. [23]

2.2.9 Software CAD

El software de diseño asistido por ordenador, mayormente conocido por las siglas CAD que provienen del inglés *Computer-Aided Design*, es un software para crear y editar modelos bidimensionales y tridimensionales de objetos físicos [24].

2.2.10 Microprocesador

Un microprocesador, también conocido como procesador o microchip, es un circuito electrónico que convierte la energía necesaria para operar el dispositivo electrónico que lo contiene mediante la ejecución de instrucciones y programas adecuadamente [25].

2.2.11 Lenguaje Python

Python es un lenguaje de programación de alto nivel, ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el Machine Learning. Además, contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible [26].

2.2.12 Protocolos de comunicación

En la electrónica integrada, se trata de interconectar circuitos (procesadores u otros circuitos integrados). Para que estos circuitos intercambien su información, deben compartir un protocolo de comunicación, por

lo que se han definido cientos de protocolos de comunicación para permitir este cambio de datos [27].

A continuación, se describe los protocolos de comunicación más utilizados que se puede realizar en una Rasberry Pi:

2.2.12.1 I2C

El protocolo de circuito integrado I2C es un protocolo destinado a permitir que varios circuitos integrados digitales “esclavos” (“chips”) se comuniquen con uno o más chips “maestros”. Está diseñada para comunicaciones de corta distancia dentro de un solo dispositivo; además, solo requiere dos cables de señal para intercambiar información [27].

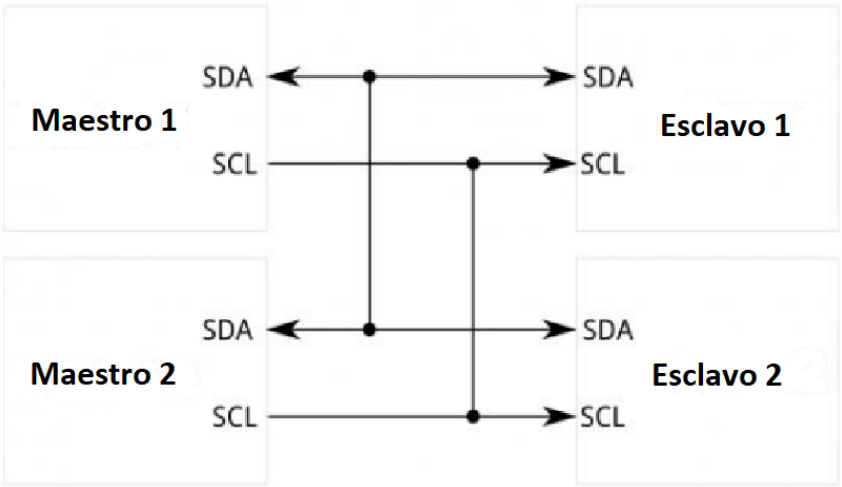


Figura 4. Esquema de comunicación I2C.

Fuente. [27]

2.2.12.2 SPI

Es un protocolo alternativo a I2C que utilizan muchos dispositivos. Es un protocolo serie, por lo que la dirección de los dispositivos no se transmite por el canal de datos, sino que se utilizan patas específicas para seleccionarlos. Hay un maestro que toma el rol de activo en la comunicación y uno o varios esclavos que toman el rol de pasivo [27].

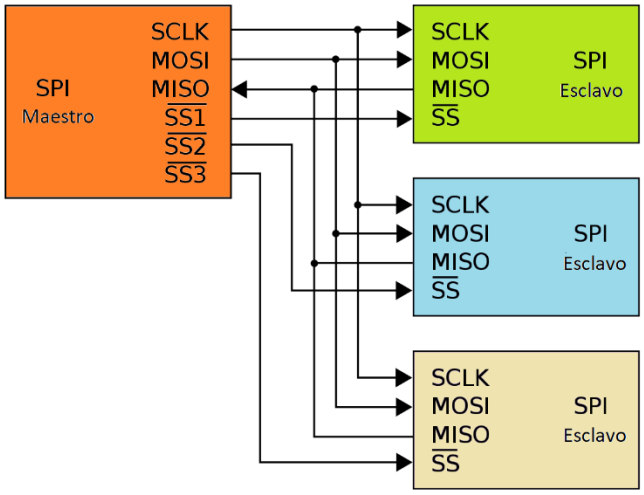


Figura 5. Esquema de comunicación SPI.

Fuente. [27]

2.2.13 Generalidades de los Mecanismos

Un mecanismo se define como un conjunto de dispositivos que, al recibir una energía de entrada, realizan un trabajo mediante un sistema de transmisión y transformación de movimientos. Para que algo sea considerado un mecanismo, debe estar compuesto por ciertos elementos clave: el eslabón, que es una pieza rígida que transfiere el movimiento de un lugar a otro y es esencial para la activación del mecanismo; el nodo, que conecta dos eslabones, permitiendo la transmisión del movimiento entre ellos; y la junta o par cinemático, que asegura el correcto funcionamiento entre el eslabón y el nodo, facilitando la unión de los eslabones dentro del nodo [28].

Los mecanismos se clasifican según la actividad que realizan:

- Acumulan energía.
- Reducen el esfuerzo.
- Transmisores de fuerzas (y tipos de palancas).
- Transmisores de movimiento y fuerzas de tracción.
- Reguladores del movimiento.

2.2.13.1 Definiciones básicas

Los eslabonamientos son los bloques de construcción básicos de todos los mecanismos. Es posible demostrar que todas las formas comunes de

mecanismos ya sean levas, engranajes, bandas o cadenas, son variaciones del tema común de eslabonamientos. Estos se componen de eslabones y juntas.

1. **Eslabones:** Un eslabón, como se muestra en la Figura 6, es un cuerpo rígido que mantiene su longitud en todo momento. Aunque en algunas situaciones se puede considerar a un resorte como un eslabón, generalmente un eslabón posee al menos dos nodos, que son puntos de unión con otros eslabones. De esta última característica, podemos mencionar que un eslabón binario es aquel que tiene dos nodos, uno ternario tiene tres nodos y uno cuaternario tiene cuatro nodos [29].

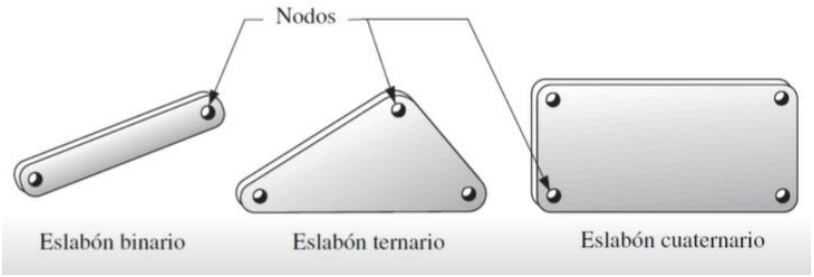


Figura 6. Eslabones de diferente orden.

Fuente: [29]

2. **Juntas:** Es una conexión entre dos o más eslabones (en sus nodos) que permite algún movimiento o movimiento potencial entre los eslabones conectados. Las juntas, también conocidas como pares cinemáticos, se pueden clasificar de varias maneras [29]:
   - **Por el tipo de contacto** entre los elementos, de línea, de punto o de superficie.
   - **Por el número de grados de libertad** permitidos en la junta.
   - Por el tipo **de cierre físico** de la junta: cerrada por fuerza o por forma.
   - Por el **número de eslabones** unidos (orden de la junta).

   Algunas de las juntas más comunes se ilustran en la Figura 7.

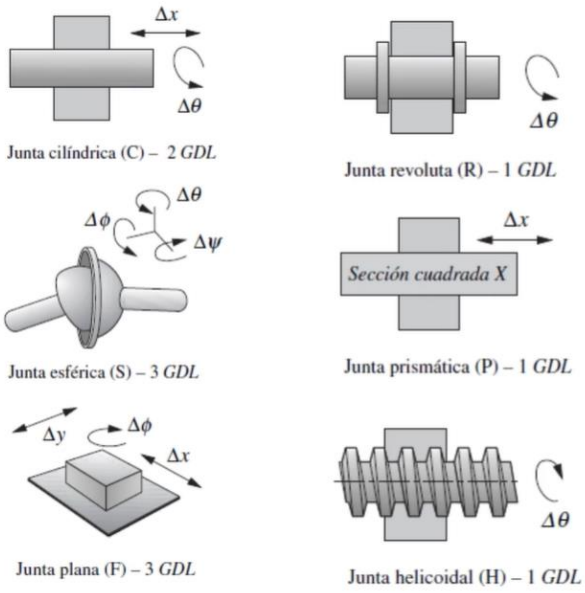


Figura 7. Seis pares inferiores.

Fuente: [29]

3. **Cadena cinemática:** Se define como un ensamble de eslabones y juntas interconectados, de modo que produzcan un movimiento controlado en respuesta a un movimiento suministrado.
4. **Mecanismo:** se define como una cadena cinemática en la cual por lo menos un eslabón se ha "fijado" o sujetado al marco de referencia (el cual por sí mismo puede estar en movimiento). Además, estos pueden ser abiertos o cerrados. En la Figura 8 muestra un mecanismo abierto y otro cerrado.

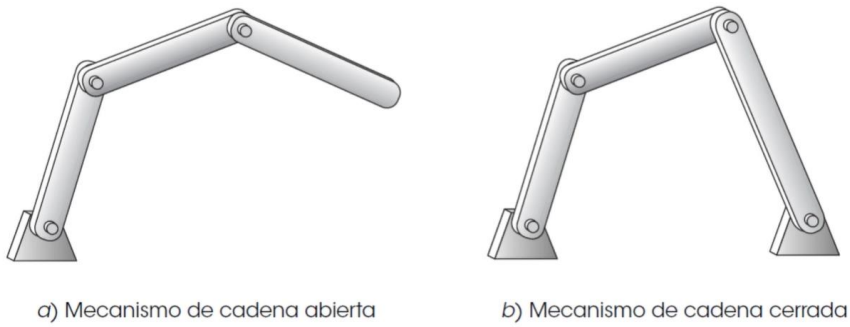


Figura 8. Tipos de mecanismos.

Fuente: [29]

Un **mecanismo cerrado** no tendrá puntos de fijación abiertos o nodos y puede tener uno o más grados de libertad. Como por ejemplo un dedo robótico.

Un **mecanismo abierto** con más de un eslabón siempre tendrá más de un grado de libertad, por lo que requieren tantos actuadores (motores) como grados de libertad tenga. Un ejemplo común de un mecanismo abierto es un robot industrial.

5. **Máquina:** se define como una combinación de cuerpos resistentes acomodados para hacer que las fuerzas mecánicas de la naturaleza realicen trabajo acompañadas por movimiento os determinados [29].

Para el mecanismo de cuatro barras, podemos definir:

6. **Manivela:** como un eslabón que realiza una revolución completa y está pivotada a la bancada [29].
7. **Balancín:** como un eslabón que tiene rotación oscilatoria (de vaivén) y está pivotado a la bancada [29].
8. **Acoplador o biela:** como un eslabón que tiene movimiento complejo y no está pivotado a la bancada [29].
9. **Bancada:** se define como cualquier eslabón o eslabones que están fijos (inmóviles) con respecto al marco de referencia. Hay que hacer notar que de hecho el marco de referencia puede estar en movimiento [29].
10. **Inversión:** Una inversión se crea al fijar un eslabón diferente en la cadena cinemática a un punto de referencia, conocido como tierra. Por lo tanto, el número de posibles inversiones en un eslabonamiento es igual al número de eslabones presentes. Cada inversión puede resultar en movimientos muy variados, aunque algunas pueden generar movimientos similares a los de otras inversiones dentro del mismo eslabonamiento. En situaciones donde los movimientos de las inversiones son completamente distintos, se consideran inversiones únicas. La Figura 9 muestra las cuatro inversiones del mecanismo de corredera-manivela de cuatro barras, cada una con movimientos distintos [30].

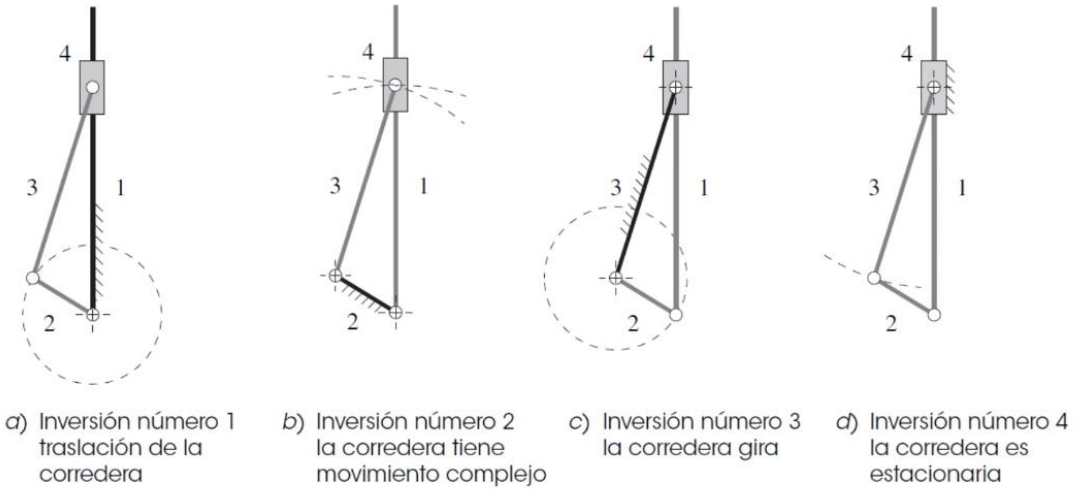


Figura 9. Tipos de Inversión.

Fuente: [31]

La **inversión número 1**, en la que el eslabón 1 actúa como base fija y la corredera realiza un movimiento de traslación pura, es la más común; por lo tanto, este tipo de inversión se emplea en motores de pistones y bombas de pistón [31].

La **inversión número 2** se obtiene al fijar el eslabón 2, produciendo el mecanismo de retorno rápido Whitworth o la limadora de manivelas, donde la corredera experimenta un movimiento complejo.

La **inversión número 3** se logra al fijar el eslabón 3, lo que permite a la corredera realizar una rotación pura.

La **inversión número 4** se obtiene al fijar el eslabón 4 y se utiliza en mecanismos manuales de bombas de pozo. En estos mecanismos, la manija actúa como el eslabón 2 extendido, mientras que el eslabón 1 desciende hasta la tubería del pozo, con un pistón montado en su extremo inferior (en la Figura 9, está invertido).

Estas inversiones demuestran que es posible crear diferentes dispositivos a partir de un mismo mecanismo simplemente variando un componente, en este caso, el eslabón fijo. Sin embargo, no es la única característica que se puede modificar para obtener resultados distintos [31].

2.2.13.2 Condición de Grashof

La condición de Grashof es una regla sencilla que permite predecir el comportamiento de rotación o rotabilidad de las inversiones en un mecanismo de cuatro barras, basándose únicamente en las longitudes de sus eslabones.

$$S + L \leq P + Q$$

Ecuación 2. Condición de Grashof.

Donde:

- S – longitud del eslabón más corto.
- L – longitud del eslabón más largo.
- P – longitud de un eslabón restante.
- Q – longitud de otro eslabón restante.

Si se cumple la igualdad de la Ecuación 2 el eslabonamiento es de Grashof, al menos uno de los eslabones podrá realizar una revolución completa en relación con la base fija. A esto se le denomina una cadena cinemática de clase I. Si la condición no se cumple, el eslabonamiento no es de Grashof y ningún eslabón podrá realizar una revolución completa en relación con los demás eslabones. Esto se conoce como una cadena cinemática de clase II [29].

Mecanismo manivela-balancín (de Grashof)

Este mecanismo se obtiene de una cadena cinemática de cuatro barras cuando la barra más corta (s) actúa como una manivela. En este caso, la barra más corta realiza giros completos, mientras que la otra barra conectada a la base fija ejecuta un movimiento de rotación alternativo, funcionando como un balancín [32], como podemos ver en la Figura 10.

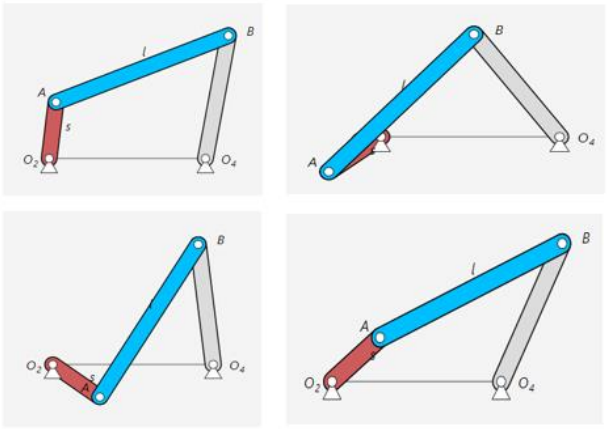


Figura 10. Mecanismo manivela-balancín (de Grashof).

Fuente: [32]

Mecanismo de doble manivela (de Grashof)

Este mecanismo se obtiene de la misma cadena cinemática de cuatro barras cuando la barra más corta (s) se fija en su posición. En este caso, las dos barras conectadas a la barra fija pueden realizar giros completos, actuando como manivelas [32], ver en la Figura 11.

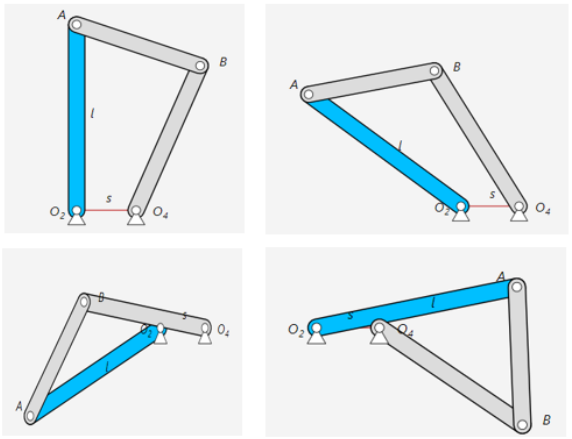


Figura 11. Mecanismo de doble manivela (de Grashof).

Fuente: [32]

Mecanismo de doble balancín (de Grashof)

Este mecanismo se deriva de la misma cadena cinemática de cuatro barras cuando la barra más corta (s) actúa como acoplador. En este caso, el mecanismo consiste en dos balancines conectados a la barra fija, y un acoplador que puede girar completamente [32], ver en la Figura 12.

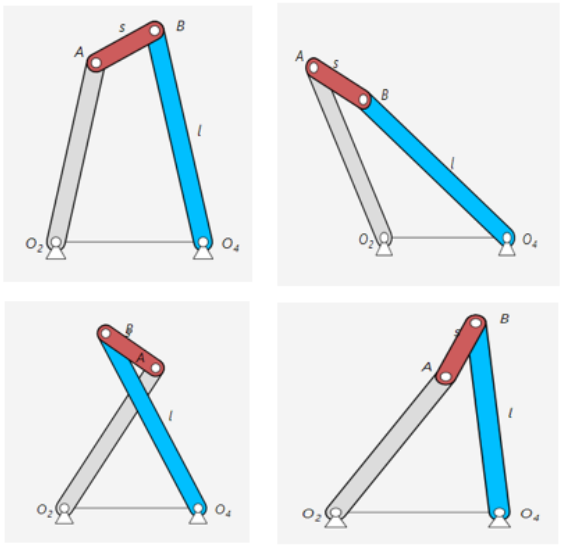


Figura 12. Mecanismo de doble balancín (de Grashof).

Fuente: [32]

Mecanismo de doble balancín (de no Grashof)

Cuando se cumple que $s + l > p + q$, ninguna inversión cinemática del cuadrilátero articulado permite que alguna de sus barras realice giros completos. Por lo tanto, todos los mecanismos que se pueden obtener en esta situación son triples balancines [32], ver Figura 13.

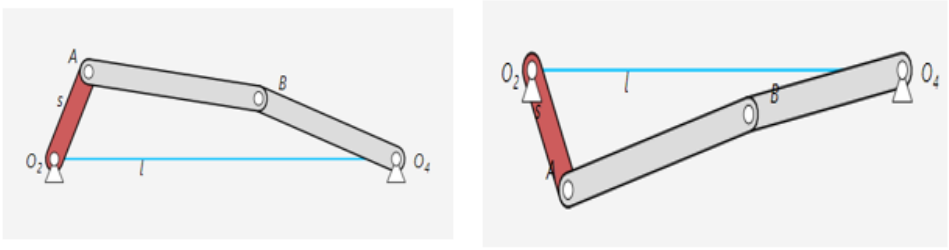


Figura 13. Mecanismo de doble balancín (de no Grashof).

Fuente: [32]

2.2.13.3 Clasificación de Barker

Barker desarrolló un esquema de clasificación que permite predecir el tipo de movimiento que se puede esperar de un mecanismo de cuatro barras, basándose en los valores de las relaciones entre sus eslabones [29].

Para esto, cada eslabón se designa con una letra según su tipo de movimiento al conectarse con los demás. Un eslabón que puede realizar una revolución completa se llama manivela (C), mientras que uno que no puede se

denomina balancín (R). El movimiento del mecanismo ensamblado, basado en su condición de Grashof y la inversión, se codifica con letras, como GCRR para un mecanismo de manivela-balancín de Grashof o GCCC para un mecanismo de doble manivela de Grashof [29].

La Tabla 1 muestra 14 tipos de mecanismos de cuatro barras según el esquema de Barker. Las primeras cuatro filas corresponden a inversiones de Grashof, las siguientes cuatro son balancines triples no Grashof, y las últimas seis representan casos especiales de eslabonamientos Grashof.

Tabla 1. Clasificación de Barker *[29]*.

Clasificación completa de Barker de mecanismos planos de cuatro barras

Tomada de la ref. (10). s = eslabón más corto, l = eslabón más largo, Gxxx = Grashof, RRRx = no Grashof, Sxx = Caso especial

Tipo	*s + l vs. p + q*	Inversión	Clase	Designación de Barker	Código	También conocido como
1	<	$L_1 = s$ = bancada	I-1	manivela-manivela-manivela de Grashof	GCCC	doble manivela
2	<	$L_2 = s$ = entrada	I-2	manivela-balancín-balancín de Grashof	GCRR	manivela-balancín
3	<	$L_3 = s$ = acoplador	I-3	balancín-manivela-balancín de Grashof	GRCR	doble balancín
4	<	$L_4 = s$ = salida	I-4	balancín-balancín-manivela de Grashof	GRRC	balancín-manivela
5	>	$L_1 = l$ = bancada	II-1	balancín-balancín-balancín clase 1	RRR1	triple balancín
6	>	$L_2 = l$ = entrada	II-2	balancín-balancín-balancín clase 2	RRR2	triple balancín
7	>	$L_3 = l$ = acoplador	II-3	balancín-balancín-balancín clase 3	RRR3	triple balancín
8	>	$L_4 = l$ = salida	II-4	balancín-balancín-balancín clase 4	RRR4	triple balancín
9	=	$L_1 = s$ = bancada	III-1	manivela-manivela-manivela con punto de cambio	SCCC	doble manivela SC*
10	=	$L_2 = s$ = entrada	III-2	manivela-balancín-balancín con punto de cambio	SCRR	manivela-balancín SC
11	=	$L_3 = s$ = acoplador	III-3	balancín-manivela-balancín con punto de cambio	SRCR	doble balancín SC
12	=	$L_4 = s$ = salida	III-4	balancín-balancín-manivela con punto de cambio	SRRC	balancín-manivela SC
13	=	dos pares iguales	III-5	punto de cambio doble	S2X	paralelogramo o deltoide
14	=	$L_1 = L_2 = L_3 = L_4$	III-6	punto de cambio triple	S3X	cuadrado

* SC = caso especial.

Por lo tanto, cada vez que diseñamos un mecanismo de cuatro barras, podemos definir las dimensiones de los eslabones según el proyecto y consultar esta tabla para prever el movimiento resultante. Si el movimiento no es el deseado, ajustamos las dimensiones hasta obtenerlo. Este enfoque nos ahorra tiempo y esfuerzo al evitar la creación de modelos 3D o prototipos físicos innecesarios, permitiéndonos enfocarnos en otros aspectos del diseño.

2.3 Marco Teórico

El trabajo de titulación, “Prótesis de antebrazo con control mioeléctrico mediante Machine Learning”, realizado en la Universidad de Valladolid, en el año 2020, por Maroto García y Miguel Ángel, crearon un prototipo funcional que podría aplicarse a personas después de una amputación, utilizando métodos de clasificación y algoritmos de aprendizaje de máquina, basados en el preprocesamiento y análisis de datos obtenidas de las señales mioeléctricas provenientes de los sensores EMG, tiene suficiente energía y habilidad para realizar actividades cotidianas como agarrar y llevar objetos ligeros [20].

El trabajo de titulación, “Desarrollo de un modelo de reconocimiento de gestos manuales de la mano utilizando señales EMG y Deep Learning”, realizado en la Escuela Politécnica Nacional, en el año 2021, por Francis Ferri Ripalda, creó un modelo de reconocimiento de la mano, basado en redes neuronales convolucionales. El dato de entrada del modelo es el espectro generado por la señal del electromiograma del antebrazo, se realizaron pruebas experimentales en 612 personas, los resultados de reconocimiento de gestos se obtuvieron en un 90.49% $\pm$ 9.70% [33].

El mismo año, en la Escuela 23 superior Politécnica del litoral, Karla Pavlova Avilés Mendoza y Neil George Gaibor León realizaron su trabajo de titulación, “Diseño e implementación de una prótesis robótica con señales EMG usando técnicas de Inteligencia Artificial”, tiene como objetivo el manejo de una prótesis de mano robótica para el campo de la terapia y la rehabilitación. La técnica utilizada fue machine learning y se usaron señales electromiográficas de los músculos para decidir si la prótesis debe abrir los dedos, cerrarlos o girar la muñeca, además, logró clasificar correctamente el 78.67% de las veces de acción deseada de la prótesis con un tiempo de respuesta bastante bajo (0.09 segundos) [7].

El trabajo de titulación, “Diseño e implementación de una prótesis biónica de mano, con capacidad de manipulación y rotación, controlada por inteligencia artificial a través de señales mioeléctricas, utilizando un sistema de entrenamiento en un entorno virtual para facilitar su adaptación”, realizado en la Universidad de las Fuerzas Armadas, en el año 2022, por Castro Vaca, Andrés

Sebastián y Lara Freire, implementaron un sistema virtual de entrenamiento para facilitar su acoplamiento y uso de la prótesis para el usuario, además, hicieron uso de sensores mioeléctricos colocados en el brazalete MYO ARM para registrar las señales del movimiento muscular del muñón en intervalos de tiempo para detectar patrones, estas señales se aplicaron en el desarrollo de la lógica de control a través de una red neuronal para la autonomía de la prótesis [11].

CAPÍTULO III

3.1 Componentes de la propuesta

En la siguiente sección, se llevó a cabo la elección de los componentes físicos y lógicos necesarios para la implementación de este proyecto. Se proporciona información detallada sobre las características técnicas más relevantes, junto con una breve explicación de su función.

3.1.1 Componentes físicos

En la implementación del prototipo de una prótesis de mano para el miembro superior izquierdo, se utiliza un enfoque basado en señales mioeléctricas (EMG) del antebrazo. Estas señales son captadas por un sensor EMG y enviadas a un microcontrolador, que las transmite a un microprocesador para su procesamiento y análisis mediante técnicas de aprendizaje automático.

Después del análisis, el microprocesador envía las predicciones al microcontrolador para controlar los servomotores integrados en el prototipo, permitiendo ejecutar los movimientos deseados en la prótesis. Para lograr esto, se emplean diversos equipos y componentes electrónicos, que se describen en detalle a continuación.

3.1.1.1 Raspberry PI 4

Fue desarrollada por la Fundación Raspberry Pi, es la cuarta generación de la serie de dispositivos Raspberry Pi, diseñada para ser asequible, de tamaño compacto y de bajo consumo de energía. Además, es una pequeña computadora de placa única con potentes capacidades de procesamiento, conectividad y salida de video, que se utiliza en una amplia variedad de proyectos y aplicaciones debido a su flexibilidad y bajo costo [34]. Este equipo será utilizado en el proyecto para el procesamiento y análisis mediante técnicas de aprendizaje de máquina. Los datos técnicos se pueden observar en la Tabla 2.

Tabla 2. Datos técnicos de la Raspberry Pi 4 model B *[34]*.

Datos Técnicos	
Modelo	Raspberry Pi 4 B
CPU	4 núcleos a 1,5GHz con brazo Cortex-A72
GPU	VideoCore VI

Memoria	8GB LPDDR4 RAM
Conectividad	802.11ac Wi-Fi/Bluetooth 5.0, Gigabit Ethernet
Vídeo y sonido	2 puertos micro-HDMI; salida estéreo de 4 polos
Puertos	2 puertos USB 3.0; 2 puertos USB 2.0
Alimentación	5V/3ª vía USB-C, vía cabezal GPIO
Expansión	Cabezal GPIO de 40 pines

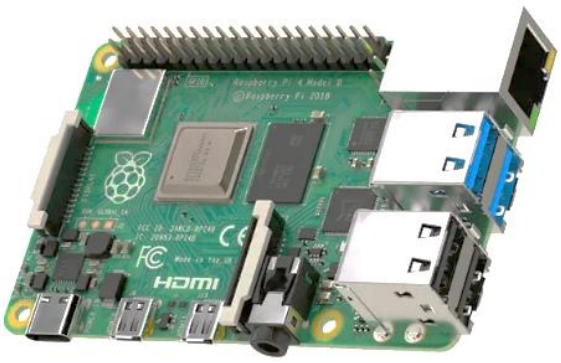


Figura 14. Raspberry Pi 4 model B.

Fuente. [35]

3.1.1.2 Arduino UNO

Es la plataforma de desarrollo de proyectos en electrónica y robótica más utilizada a nivel mundial, gracias a su facilidad de aprendizaje y uso, su abundante documentación y sus múltiples aplicaciones [36]. Los datos técnicos se pueden observar en la Tabla 3.

Tabla 3. Datos técnicos del Arduino UNO *[36]*.

Datos Técnicos	
Microcontrolador	ATmega328P (8-bit)
Chip USB	ATmega16U2
Voltaje de operación	5V
Voltaje de alimentación	6V - 20V DC (7-12V recomendado)
Pines digitales I/O	14 (6 salidas PWM)
Entradas analógicas	6 (ADC 10-bit)
Corriente entrada/salida por pin	40mA máx
Frecuencia de reloj	16MHz

Figura 15. Arduino UNO.

Fuente: [36]

3.1.1.3 Servo motor MG90S

Es un pequeño motor eléctrico utilizado en aplicaciones de control de movimiento que requieren precisión y respuesta rápida. Se caracteriza por su tamaño compacto, peso ligero y capacidad de giro de hasta 180 grados y se controla mediante señales de ancho de pulsos modulado (PWM). Además, es ampliamente utilizado en proyectos de robótica, aeromodelismo y otras aplicaciones donde se necesita un control preciso y ágil del movimiento [37]. Este componente será integrado en la mano del prototipo de la prótesis, desempeñando un papel fundamental al permitir los movimientos de los dedos, por lo tanto, se emplearán seis de estos servomotores para lograr una funcionalidad completa y controlada de la mano protésica. Los datos técnicos se pueden observar en la Tabla 4.

Tabla 4. Datos técnicos del servo motor MG90S *[37]*.

Datos Técnicos	
Voltaje de operación	3.0 – 7.2V DC
Torque reposo	2.2kg x cm (4.8V), 2.5kg (6.0V)
Velocidad	0.1 seg / 60 grados
Ancho del pulso	Entre 600uS y 240uS
Engranajes	Metal
Dimensiones	22*11.5*27mm
Peso	13.4 gramos

Figura 16. Servo motor MG90S.

Fuente. [37]

3.1.1.4 Pilas Recargables Maxday 3.7 Voltios

Las pilas recargables Maxday de 3.7 voltios son baterías de iones de litio diseñadas para proporcionar una fuente de energía confiable y duradera en una variedad de dispositivos electrónicos. Con una capacidad típica de 8800mAh, estas pilas son capaces de alimentar dispositivos de alta energía, como linternas LED, cámaras digitales, juguetes electrónicos y otros dispositivos portátiles [38]. En la Figura 17, se puede observar la pila que se va a utilizar en este proyecto.

Figura 17. Pila Maxday Li-ion.

Fuente. [38]

3.1.1.5 Porta Pilas

Es un dispositivo diseñado para contener y conectar una o varias pilas en serie o en paralelo, dependiendo de la configuración necesaria para alimentar un dispositivo electrónico. Además, su diseño modular y sus características de protección hacen que sea una solución conveniente y versátil para una variedad de aplicaciones electrónicas [39]. En la Figura 18, se puede observar el porta pilas que se va a utilizar en este proyecto.

Figura 18. Porta pilas.

Fuente. [39]

3.1.1.6 Electrodos de Plata/Cloruro de Plata

Son dispositivos utilizados en el campo de la electrofisiología, especialmente en aplicaciones médicas y de investigación donde se registran señales eléctricas del cuerpo humano, como en el caso de la electromiografía (EMG). Estos electrodos están compuestos por un canal conductor de plata, que permite una excelente conductividad eléctrica, y están recubiertos con cloruro de plata para mejorar la estabilidad del electrodo y reducir el ruido en las señales registradas [40]. En la Figura 19, se puede observar el tipo de electrodos Ag/AgCl.

Figura 19. Electrodo Superficial de Ag/AgCl.

Fuente. [41]

3.1.1.7 Cables para electrodos

Son conductores flexibles especialmente creados para unir electrodos a dispositivos de medición, como en el caso de los electrodos de plata/cloruro de plata utilizados en EMG. Estos cables transmiten las señales eléctricas captadas por los electrodos al dispositivo de registro o análisis, siendo vitales para obtener

datos precisos en pruebas médicas o investigaciones fisiológicas. En la Figura 20, se puede observar el tipo de cable donde van los electrodos.

Figura 20. Cable para electrodos superficiales.

Fuente. [42]

3.1.1.8 Gel conductor

Es un gel que mejora la conductividad eléctrica entre los electrodos y la piel del paciente durante las pruebas médicas como la electromiografía EMG. Esto permite una transmisión de señales más precisa y clara durante las pruebas musculares o procedimientos médicos. En la Figura 21, se puede observar el tipo de gel conductor.

Figura 21. Electro Gel.

Fuente. [43]

3.1.1.9 Osciloscopio DSO138

Es un osciloscopio digital de bajo costo y tamaño compacto, por lo tanto, lo hace atractivo para entusiastas de la electrónica y proyectos educativos. A pesar de su simplicidad y precio asequible, proporciona funcionalidades básicas

de visualización de señales analógicas que pueden resultar útiles en una variedad de proyectos [44], incluyendo las señales EMG, los datos técnicos se pueden observar en la Tabla 5.

Tabla 5. Datos técnicos del Osciloscopio DSO138 *[44]*.

Datos técnicos	
Frecuencia de muestreo máxima en tiempo real	1Msa/s
Ancho de banda analógico	0 – 200KHz
Rango de sensibilidad	10mV/div – 5V/div
Voltaje de entrada máximo	50 Vpk (sonda 1X)
Impedancia de entrada	1M ohm/20pF
Resolución	12 bits
Longitud del registro	1024 puntos
Rango de base de tiempo	500s/Div – 10us/Div
Modos de disparo	Auto, Normal, y Sencillo
Rango de posición de disparo	50%
Fuente de alimentación	9 VCC (8 – 12 V)
Consumo de corriente	~120mA
Dimensión	117 x 76 x 15mm
Peso	70 gramos (sin sonda)

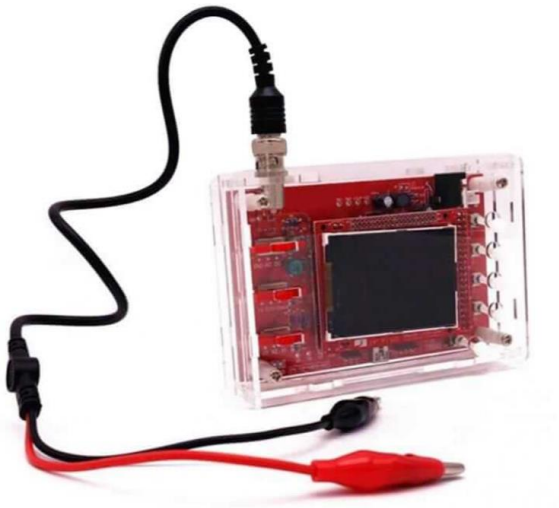


Figura 22. Osciloscopio DSO138.

Fuente. [45]

3.1.1.10 Amplificador de instrumentación AD623

El AD623 es un amplificador integrado de instrumentación que funciona con una o dos fuentes de alimentación, proporcionando una salida que se extiende hasta los límites de los rieles de alimentación, con voltajes que van desde 2,7 V hasta 12 V. Su flexibilidad permite al usuario ajustar la ganancia programando resistencias, y se adapta fácilmente a la distribución estándar de pines de 8 conductores de la industria. Por defecto, tiene una ganancia unitaria (G = 1), pero puede programarse con una resistencia externa para alcanzar ganancias de hasta 1000. La precisión del AD623 se ve mejorada por su CMRR "*Common Mode Rejection Ratio*" (relación de rechazo de modo común), que aumenta con la ganancia, permitiendo rechazar armónicos de ruido de línea hasta 200 Hz. Además, tiene un amplio rango de entrada de modo común, capaz de amplificar señales incluso con voltajes de modo común tan bajos como 150 mV por debajo del nivel de tierra. Su rendimiento óptimo se mantiene con fuentes de alimentación de polaridad simple o doble [46].

La Figura 23 y la Figura 24, se muestran los circuitos de conexión básicos para el AD623. Los terminales +VS y -VS están conectados a la fuente de alimentación, que puede ser bipolar (VS = ±2,5 V a ±6 V) o de suministro único ($-VS = 0\,V$ y $+VS = 2{,}7\,V\ a\ 12\,V$). Además, se desacopla capacitivamente las fuentes de alimentación cercanas a los pines del dispositivo para obtener resultados óptimos, se recomienda utilizar condensadores de chip cerámico de 0,1 µF de montaje superficial junto con condensadores de tantalio electrolíticos de 10 µF [46].

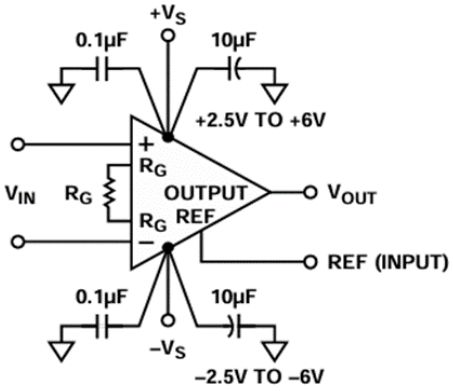


Figura 23. Conexión básica de suministro dual.

Fuente. [46]

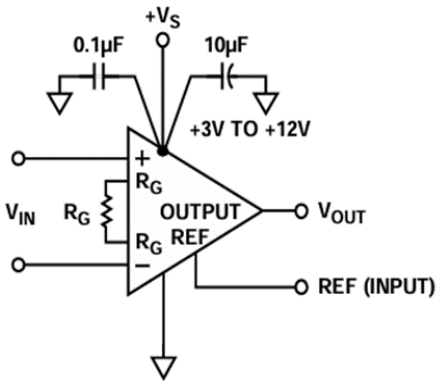


Figura 24. Conexión básica de suministro único.

Fuente. [46]

La ganancia del AD623 está programada por la resistencia RG, o más precisamente, por cualquier impedancia que aparezca entre el Pin 1 y el Pin 8. El AD623 ofrece ganancias precisas utilizando resistencias de tolerancia del 0,1% al 1% [46].

$$R_G = \frac{100\ k\Omega}{G - 1}$$

Ecuación 3. Resistencia para la ganancia del AD623.

Los datos técnicos del amplificador de instrumentación se pueden observar en la Tabla 6.

Tabla 6. Amplificador de instrumentación AD623 *[46]*.

Datos Técnicos	
Suministro de voltaje	± 2.7V a 12V, 2.5V a 6V
Corriente de reposo máxima	550 uA
Polarización de entrada de corriente	17 nA
Rango de ganancia	1 – 1000
Velocidad de respuesta	0.3 V / us
Ancho de banda de -3 dB	800 kHz

Figura 25. Circuito integrado del Amplificador de instrumentación AD623.

Fuente. [47]

3.1.1.11 Resistencia eléctrica

La resistencia eléctrica surge debido a la interacción entre los electrones (cargas eléctricas negativas) y los iones positivos en un material conductor. Cuando se aplica un campo eléctrico (voltaje) a través de un conductor, los electrones comienzan a moverse en respuesta a este campo. Sin embargo, su movimiento se ve obstaculizado por las colisiones con los átomos del material conductor. Estas colisiones dispersan la energía cinética de los electrones en forma de calor, lo que resulta en una resistencia al flujo de corriente [48]. En la Figura 26, se puede observar una representación de una resistencia eléctrica.

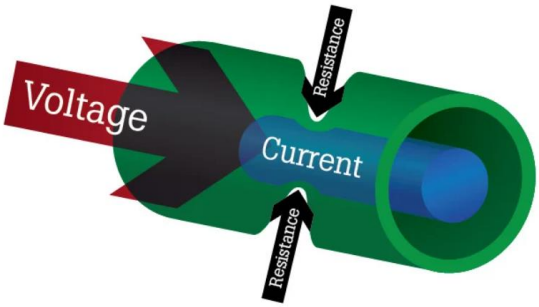


Figura 26. Resistencia eléctrica.

Fuente. [48]

La resistencia eléctrica tiene una estrecha relación con otros principios eléctricos, como la corriente (el flujo de carga eléctrica) y el voltaje (la diferencia de potencial eléctrico). Según la ley de Ohm, expresada como $\boldsymbol{V = I\,x\,R}$, se establece que la corriente ($\boldsymbol{I}$) que atraviesa un conductor es proporcional al voltaje ($\boldsymbol{V}$) aplicado e inversamente proporcional a la resistencia ($\boldsymbol{R}$) del conductor.

Además, la resistencia está vinculada con la potencia eléctrica (**P**) que se disipa en forma de calor. Esta potencia se calcula mediante las fórmulas $\boldsymbol{P =}$

$\boldsymbol{I^2\ x\ R}$ o $\boldsymbol{P = \frac{V^2}{R}}$. Estas relaciones son esenciales para diseñar circuitos eléctricos eficientes y para gestionar la energía de manera efectiva.

3.1.1.12 Capacitor cerámico

Un capacitor cerámico es un componente electrónico utilizado en circuitos eléctricos para almacenar y liberar carga eléctrica. Estos dispositivos están compuestos por materiales cerámicos y son ampliamente utilizados debido a sus características de rendimiento, fiabilidad y bajo costo.

La estructura de un capacitor cerámico consiste en dos placas metálicas, separadas por un material dieléctrico cerámico. Por lo tanto, este dieléctrico cerámico es un material no conductor que posibilita el almacenamiento de carga eléctrica entre las placas del capacitor. Al aplicar un voltaje a través del capacitor, las placas acumulan una carga eléctrica proporcional al voltaje aplicado y a la capacidad del capacitor [49]. En la Figura 27, se puede observar un capacitor cerámico y su estructura.

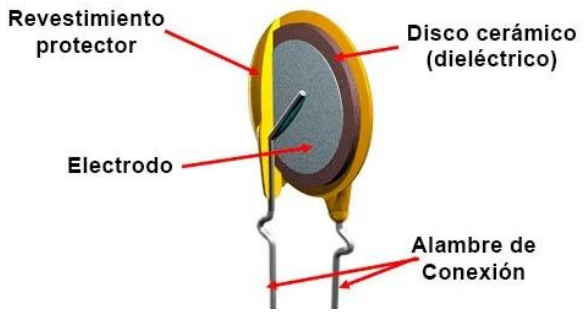


Figura 27. Estructura de un capacitor cerámico.

Fuente. [50]

3.1.1.13 Amplificador operacional LM358

El Amplificador Operacional LM358 es un circuito integrado (Ci) de 8 pines que contiene dos amplificadores operacionales independientes y con compensación de frecuencia interna, diseñado para operar con una única fuente de alimentación en un amplio rango de voltaje; además, puede funcionar con fuentes de alimentación divididas. Debido a su baja pérdida de corriente, es eficiente energéticamente, por lo que es ideal para aplicaciones como amplificadores de audio, filtros activos y convertidores de señal, destacándose

por su compacidad, facilidad de uso y versatilidad [51]. Los datos técnicos se pueden observar en la Tabla 7.

Tabla 7. Datos técnicos del amplificador operacional LM358 *[51]*.

Datos Técnicos	
Encapsulado	PDIP-8
Pines	8
Voltaje de alimentación min-max	3v-32v
Voltaje offset de entrada	7mV
Corriente de salida por canal	30mA
Corriente de suministro operativa	350uA
Velocidad de precesión	0.3 V/us
Numero de canales	2 canales
Frecuencia de ancho de banda (GBP)	700KHz
Ganancia de voltaje	100 dB

Figura 28. Amplificador operacional LM358.

Fuente. [51]

3.1.1.14 Convertidor A/D ADS1115

El Módulo ADS1115 ADC es un convertidor analógico a digital diseñado para microcontroladores sin un convertidor propio o cuando se requiere mayor precisión, proporcionando 16 bits a 860 muestras/segundo sobre I2C. El ADS1115 proporciona 4 ADC de 16 bits, 15 para la medición y un último para el signo, con 4 direcciones seleccionables mediante la conexión del pin ADDRESS. Este módulo se puede configurar con 4 canales en modo single-ended o 2 canales en modo diferencial; en modo single-ended dispone de cuatro canales de 15 bits, mientras que en modo diferencial se reduce a 2 canales, permitiendo medir tensiones negativas y ofreciendo mayor inmunidad al ruido. Además,

incluye un amplificador de ganancias programable (PGA) hasta x16, permitiendo aumentar señales más pequeñas dentro del rango de 6.144V a 0.256V. Funciona con fuentes de 2V a 5V, mide una variedad de señales y es fácil de usar. Su interfaz I2C admite hasta 4 módulos en un solo bus, proporcionando hasta 16 entradas single-ended [52]. Los datos técnicos se pueden observar en la Tabla 8.

Tabla 8. Datos técnicos del módulo A/D ADS1115 *[52]*.

Datos Técnicos	
Voltaje de entrada min-max	2v – 5v
Corriente de entrada en modo lectura	200 uA
Corriente de entrada en modo auto apagado	5 uA
Dimensiones	1.8 cm * 2.8 cm

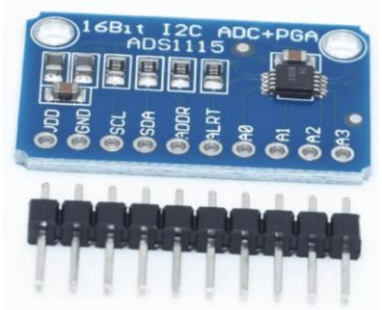


Figura 29. Módulo convertidor A/D ADS1115.

Fuente. [52]

3.1.1.15 Regulador de Voltaje DC/DC LM2596

El convertidor de voltaje DC-DC Step-Down 3A LM2596 tiene la función de proporcionar un voltaje de salida constante que sea inferior al voltaje de entrada, incluso ante variaciones en el voltaje de entrada o en la carga. Este dispositivo soporta corrientes de salida de hasta 3A, con un rango de voltaje de entrada entre 4.5V y 40V, y un rango de voltaje de salida entre 1.23V y 37V. El voltaje de salida se ajusta mediante un potenciómetro multivuelta [53]. Los datos técnicos se pueden ver en la Tabla 9.

Tabla 9. Datos técnicos LM2596 *[53]*.

Datos Técnicos	
Voltaje de entrada	4.5V a 40V DC

Voltaje de salida	1.23V a 37V DC
Corriente de Salida max	A, 2.5A recomendado
Potencia de salida	25W
Eficiencia de conversión	92%
Frecuencia de Trabajo	150KHz

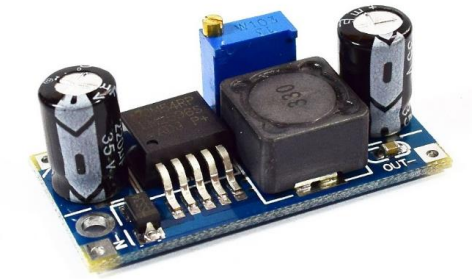


Figura 30. Regulador de voltaje LM2596.

Fuente: [53]

3.1.1.16 Regulador de voltaje lineal LM7805

El regulador de voltaje fijo 7805 es ideal para una variedad de aplicaciones, especialmente en la regulación en la tarjeta para reducir el ruido y resolver problemas de distribución de voltaje. Cuenta con limitación interna de corriente y apagado térmico, lo que lo protege contra sobrecargas. Además de su uso como regulador de voltaje fijo, puede combinarse con componentes externos para ajustar voltajes y corrientes de salida, y también puede servir como elemento de paso en reguladores de precisión [54], los datos técnicos se pueden observar en la Tabla 10.

Tabla 10. Regulador de voltaje LM7805 *[54]*.

Datos Técnicos	
Voltaje de entrada máximo	35 V
Corriente de salida	1.5 A
Voltaje de salida	5 V
Encapsulado	TO-220

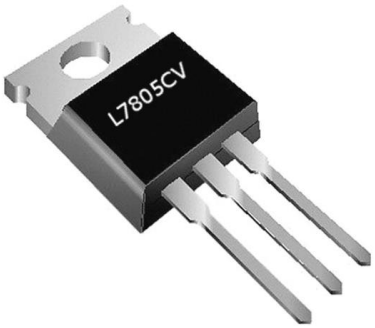


Figura 31. Regulador de voltaje LM7805.

Fuente. [54]

3.1.1.17 Regulador de voltaje lineal LM7905

El LM7905 es un regulador de voltaje negativo de 3 terminales que proporciona una salida fija de -5 V. Este dispositivo solo requiere un condensador de compensación en la salida como componente externo. Ofrece protección contra sobrecargas mediante limitación de corriente interna, protección de zona segura y apagado térmico. Además, su conector de baja corriente de tierra permite ajustar el voltaje de salida por encima del valor fijo utilizando un divisor de resistencia. La baja fuga de corriente de reposo y su estabilidad ante cambios en la línea y la carga aseguran una buena regulación del voltaje [55] Los datos técnicos se pueden observar en la Tabla 11.

Tabla 11. Regulación de voltaje negativo LM7905 *[55]*.

Datos Técnicos	
Tensión de entrada min-max	-30 V / --10 V
Tensión nominal de salida fija	-5 V
Corriente de salida	1 A
Encapsulado	TO-220

Figura 32. Regulador de voltaje negativo LM7905.

Fuente. [56]

3.1.2 Componentes lógicos

3.1.2.1 Autodesk Fusion 360

Autodesk Fusion 360 es una plataforma de software de modelado 3D, CAD, CAM, CAE y PCB basada en la nube, destinada al diseño y la fabricación de productos. Integra diseño paramétrico, modelado de superficies, renderizado, simulaciones, manufactura asistida por computadora (CAM) y análisis de elementos finitos (FEA) [57].

3.1.2.2 Multisim 14.0

Multisim es un software reconocido en la industria para el diseño y simulación de circuitos electrónicos, abarcando aplicaciones en electrónica de potencia, analógica y digital, utilizado ampliamente en entornos educativos y de investigación. Este programa combina la simulación SPICE con un entorno esquemático interactivo, lo que permite a los usuarios visualizar y analizar de manera inmediata el comportamiento de los circuitos electrónicos [58].

3.1.2.3 Raspberry PI Imager

Es una herramienta desarrollada para simplificar el proceso de instalación de sistemas operativos en tarjetas SD para dispositivos Raspberry Pi. Esta aplicación permite a los usuarios seleccionar y descargar diversas imágenes de sistemas operativos compatibles con Raspberry Pi, como Raspberry Pi OS, Ubuntu y otros [59].

3.1.2.4 Advanced IP Scanner

Es una herramienta de escaneo de redes que permite a los usuarios analizar y administrar dispositivos conectados a una red local. Sirve para detectar

y mostrar información detallada sobre todos los dispositivos presentes en la red, incluyendo direcciones IP, nombres de host, fabricantes de hardware y estados de conexión [60].

3.1.2.5 VNC Viewer

Es una aplicación que permite acceder de forma remota a equipos de escritorio desde cualquier lugar utilizando una conexión a Internet. Sirve para controlar y visualizar la pantalla de un ordenador de forma remota, lo que facilita el acceso y la gestión de dispositivos incluso cuando no están físicamente disponibles [61].

3.1.2.6 Python

Python es un lenguaje de programación ampliamente utilizado en aplicaciones web, desarrollo de software, ciencia de datos y machine learning (ML). Los desarrolladores prefieren Python porque es eficiente, fácil de aprender y se puede ejecutar en múltiples plataformas. Además, es gratuito para descargar, se integra bien con diversos sistemas y acelera la velocidad de desarrollo [62].

3.1.2.7 Tensorflow

Es una biblioteca de software de código abierto desarrollada por Google que se utiliza principalmente para el aprendizaje automático y la inteligencia artificial. Sirve como una plataforma completa para la construcción y entrenamiento de modelos de aprendizaje automático, incluyendo redes neuronales profundas y otros tipos de algoritmos de aprendizaje automático [63].

3.2 Diseño de la propuesta

En este capítulo se describen los pasos necesarios para alcanzar los objetivos planteados:

3.2.1 Diagrama de bloques del proyecto a realizar

El proyecto se divide en varias etapas funcionales, como se muestra a continuación en la Figura 33.

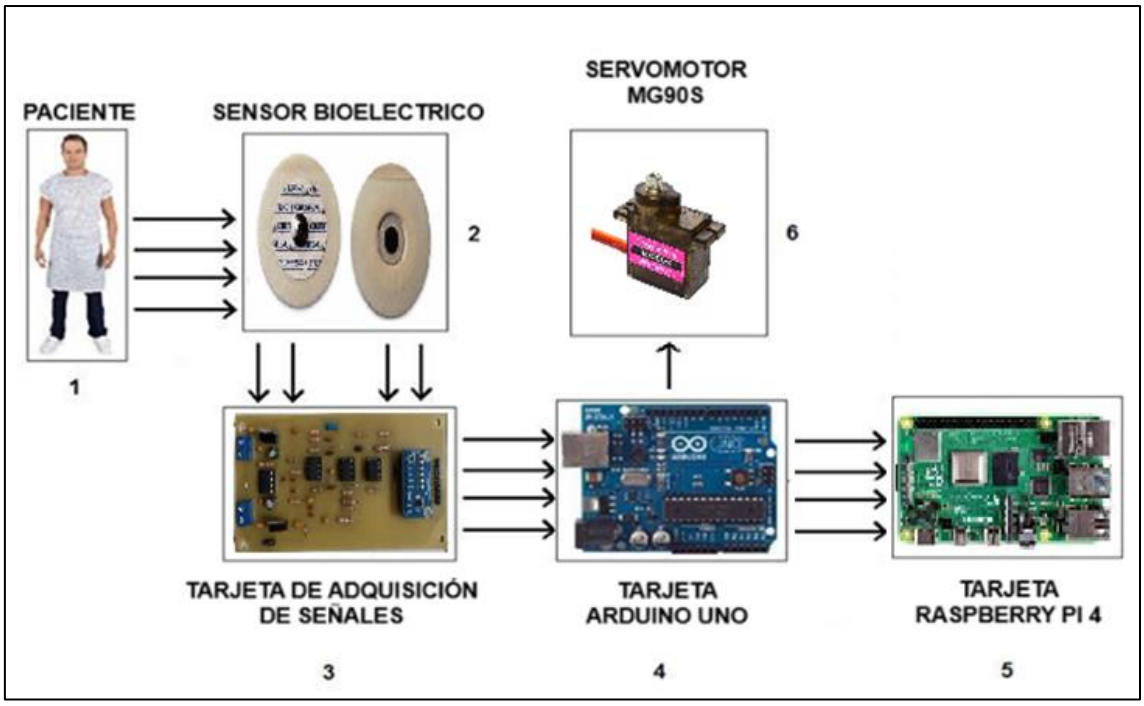


Figura 33. Diagrama de bloques general para la implementación del proyecto.

Fuente: Autoría propia.

En el primer y segundo bloque, el paciente tiene electrodos colocados en el antebrazo izquierdo para registrar las señales mioeléctricas del músculo durante una contracción muscular. A continuación, la señal pasa por etapas de amplificación inicial, prefiltrado y acondicionamiento para mejorar su calidad y eliminar interferencias.

Posteriormente, estas señales analógicas se convierten a digitales para obtener una mejor resolución. Este proceso se realiza en la tarjeta de adquisición de señales. El Arduino UNO extrae características de las señales correspondientes a diferentes gestos y las envía a la Raspberry Pi 4, que las guarda en una base de datos.

La Raspberry Pi 4 utiliza estas señales para entrenar un modelo de redes neuronales que permite el reconocimiento de gestos. Los resultados del modelo se envían de nuevo al Arduino UNO, que controla los servomotores en un prototipo de mano, facilitando así el control de la mano izquierda del paciente mediante los gestos reconocidos.

3.2.2 Implementación del EMG

Durante el desarrollo del EMG se adquirieron señales provenientes del antebrazo izquierdo del paciente utilizando electrodos, un amplificador de instrumentación, filtros analógicos y un amplificador de ganancia como se puede observar en los diagramas de bloque de la Figura 34 y en Anexo 1 se muestra el circuito de adquisición que permite que estas señales ingresen al osciloscopio DSO138 para su visualización.

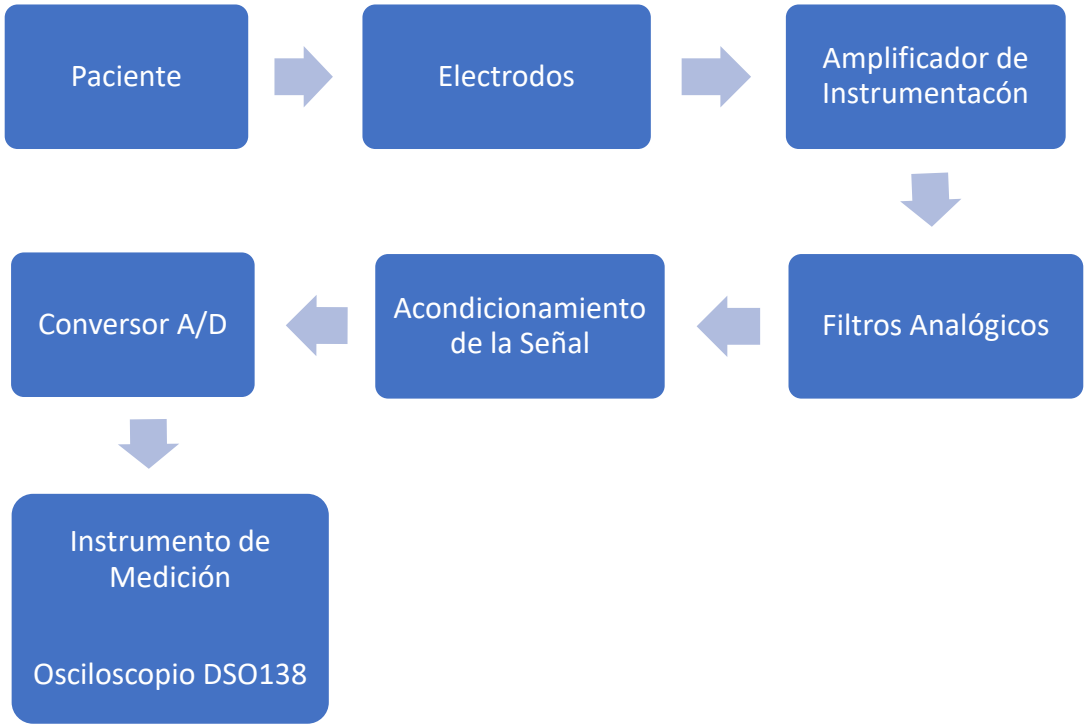


Figura 34. Diagrama de Bloques del EMG.

Fuente: Autoría propia.

3.2.2.1 Amplificación inicial

Una de las etapas fundamentales en el diseño del circuito para captar señales musculares es la amplificación inicial, el cual desempeña un papel crucial en la amplificación precisa de las débiles señales musculares captadas por los electrodos. Para esta tarea, se emplea el amplificador de instrumentación AD623 como se puede ver en la Figura 35, conocido por su alta precisión y bajo ruido.

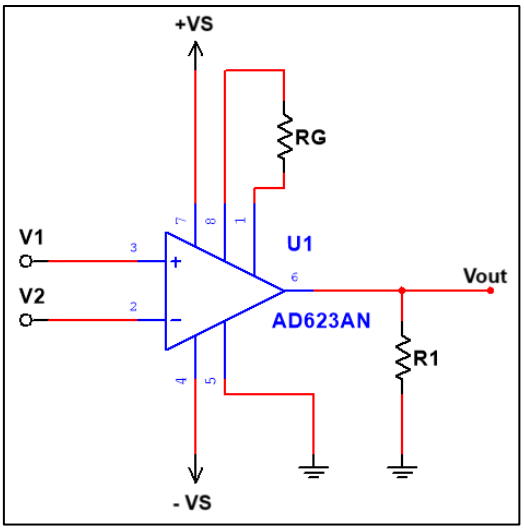


Figura 35. Diagrama del Amplificador de Instrumentación AD623.

Fuente: Autoría propia.

Este amplificador amplifica la diferencia entre dos señales de entrada y rechazan cualquier señal que sea común a ambas señales. También ofrece una ganancia ajustable, lo que permite amplificar selectivamente las señales de interés mientras minimiza la interferencia y el ruido externo. Además, el AD623 presenta una alta impedancia de entrada, asegurando una carga mínima en el circuito y una captación eficiente de las señales musculares.

Para esta primera etapa del circuito se dio una ganancia de 304 haciendo uso de la Ecuación 2, obtenemos el valor de resistencia de ganancia R_G del amplificador:

$$R_G = \frac{100\ k\Omega}{304 - 1} = 330\ \Omega$$

Al reemplazar los valores de la resistencia R_G calculado, y según el datasheet del AD623 [46] el valor de $R_1 = 10\ k\Omega$, se puede establecer el amplificador de instrumentación como se puede ver en la Figura 36.

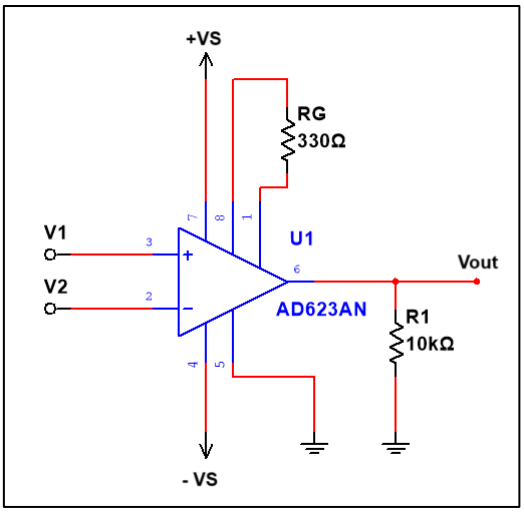


Figura 36. Amplificador de Instrumentación AD623 con los valores de diseño.

Fuente: Autoría propia.

3.2.2.2 Implementación de los Filtros Analógicos

El ruido ambiental puede afectar la señal EMG, por lo que se han desarrollado filtros específicos para obtener una señal aceptable. Para reducir las frecuencias indeseables y el ruido ambiental, se utilizan tres tipos de filtros: un filtro pasa alta que atenúa las frecuencias menores a 50 Hz (Figura 37), un filtro pasa baja que atenúa las frecuencias mayores a 300 Hz (Figura 39), ambos configurados en filtros de segundo orden Multiple Feedback con ganancias unitarias Butterworth, cubriendo un rango de 50 a 300 Hz. Además, se emplea un filtro de muesca (notch) de segundo orden en configuración Twin-T centrado en frecuencias de 60 Hz, para eliminar el ruido generado por la línea de potencia y otros dispositivos que emiten ruido a través de ondas de 60 Hz (Figura 41). Estos filtros en conjunto reducen el ruido y mejoran la calidad de la señal EMG.

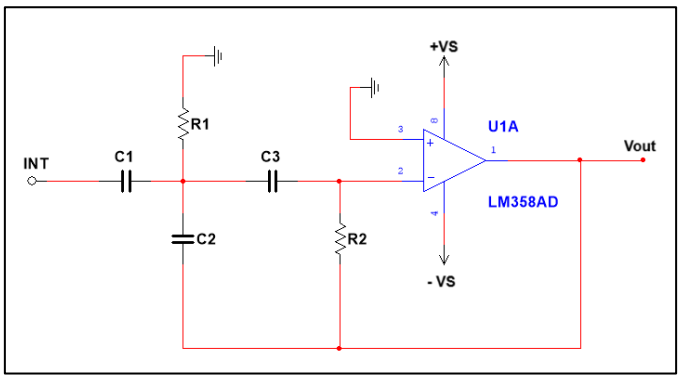


Figura 37. Filtro activo de segundo orden Pasa Alta.

Fuente: Autoría propia.

Ganancia Unitaria Butterworth – Filtro Pasa Alta

$$F_c = \frac{1}{2\pi RC}$$

Ecuación 4. Fórmula para hallar la frecuencia de corte del filtro pasa alta.

$$R_1 = 0.47R$$

Ecuación 5. Fórmula para hallar la resistencia R1 del filtro pasa alta.

$$R_2 = 2.1R$$

Ecuación 6. Fórmula para hallar la resistencia R2 del filtro pasa alta.

$$C_1 = C_2 = C_3 = C$$

Ecuación 7. Valores de los capacitores del filtro pasa alta.

Si $C = 0.01uF$ y como $F_c = 50Hz$

$$R = \frac{1}{2\pi F_c C} = \frac{1}{2\pi * 50Hz * 0.01uF} = 318.3k\Omega$$

$$R_1 = 0.47R = 0.47 * 318.3k\Omega = 149.6k\Omega$$

$\rightarrow \quad R_1 = 150k\Omega$

$$R_2 = 2.1R = 2.1 * 318.3k\Omega = 668.43k\Omega$$

$\rightarrow \quad R_2 = 680k\Omega$

$\rightarrow \quad C_1 = C_2 = C_3 = 0.01uF$

Una vez realizado los cálculos se reemplaza los valores de los capacitores y resistencias, se puede establecer el filtro pasa alta en la Figura 38.

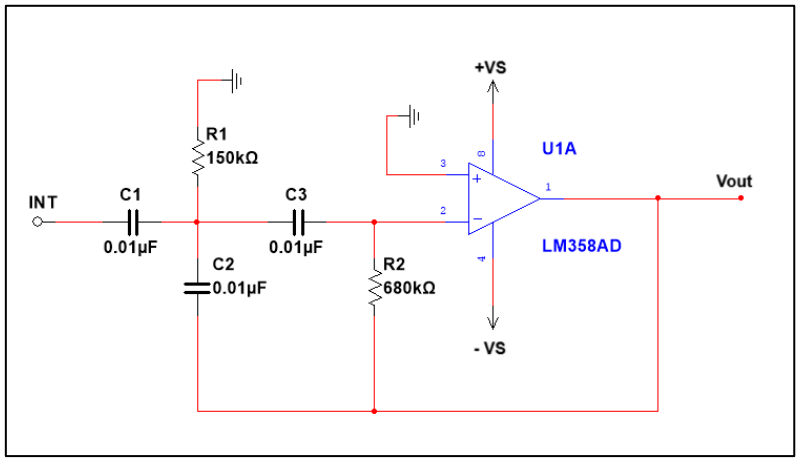


Figura 38. Filtro Pasa Alta con los valores del diseño.

Fuente: Autoría propia.

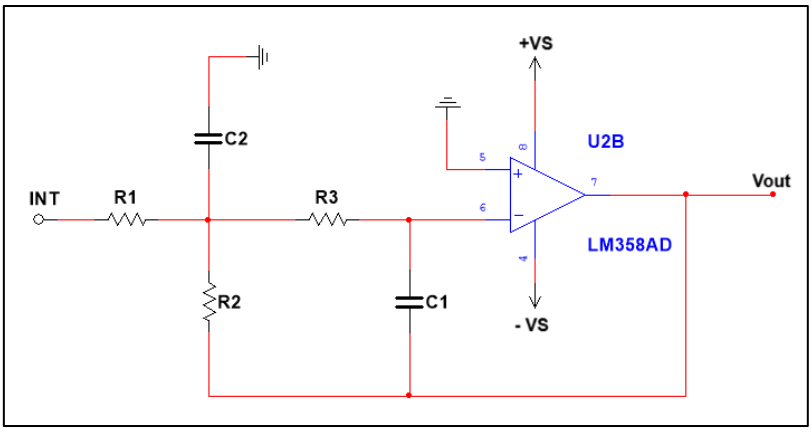


Figura 39. Filtro activo de segundo orden Pasa Baja.

Fuente: Autoría propia.

Ganancia Unitaria Butterworth – Filtro Pasa Baja

$$F_c = \frac{1}{2\pi RC}$$

Ecuación 8. Fórmula para hallar la frecuencia de corte del filtro pasa baja.

$$R_1 = R_2 = {}^{R}/_{\sqrt{2}}$$

Ecuación 9. Fórmula para hallar la resistencia R1 y R2 del filtro pasa baja.

$$R_3 = {}^{R}/_{2\sqrt{2}}$$

Ecuación 10. Fórmula para hallar la resistencia R3 del filtro pasa baja.

$$C_1 = C$$

Ecuación 11. Valor del capacitor C1 del filtro pasa baja.

$$C_2 = 4C$$

Ecuación 12. Valor del capacitor C2 del filtro pasa baja.

Si $C = 0.12uF$ y como $F_c = 300Hz$

$\rightarrow \quad C_1 = C = 0.12uF$

$$C_2 = 4C = 4 * 0.12uF = 0.48uF$$

$\rightarrow \quad C_2 = 0.47uF$

$$R = \frac{1}{2\pi F_c C} = \frac{1}{2\pi * 300Hz * 0.12uF} = 4421\ \Omega$$

$$R_1 = R_2 = {}^{R}/_{\sqrt{2}} = {}^{4421\Omega}/_{\sqrt{2}} = 3126\ \Omega$$

$\rightarrow \quad R_1 = R_2 = 3.3k\Omega$

$$R_3 = {R}/{2\sqrt{2}} = {4421\Omega}/{2\sqrt{2}} = 1563\ \Omega$$

$$\rightarrow \quad R_3 = 1.5k\Omega$$

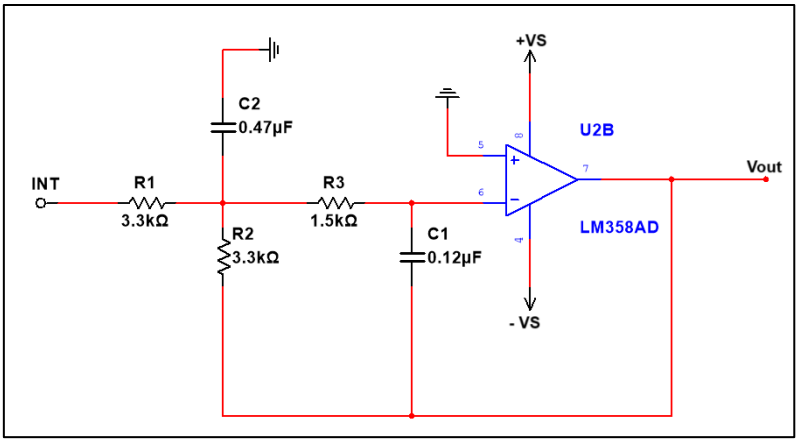


Figura 40. Filtro Pasa Alta con los valores del diseño.

Fuente: Autoría propia.

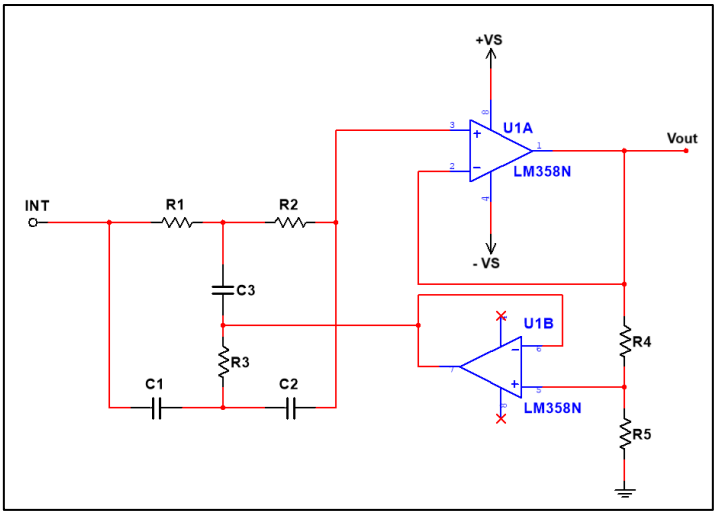


Figura 41. Filtro de muesca Twin-T.

Fuente: Autoría propia.

$$F_c = \frac{1}{2\pi RC}$$

Ecuación 13. Fórmula para hallar la frecuencia de corte del filtro de muesca.

$$R_1 = R_2 = R$$

Ecuación 14. Valores de las resistencias R1 y R2 del filtro de muesca.

$$R_3 = \frac{R}{2}$$

Ecuación 15. Fórmula para hallar la resistencia R3 del filtro de muesca.

$$C_1 = C_2 = C$$

Ecuación 16. Valores de los capacitores del filtro de muesca.

Q controlado por la proporción de R_4 y R_5

$$R_4 = 0.05 * R_5$$

Ecuación 17. Fórmula para hallar la resistencia R4 para un Q alto.

$$R_4 = 0.5 * R_5 : bajo\ Q$$

Ecuación 18. Fórmula para hallar la resistencia R4 para un Q bajo.

Si $C = 0.1uF$ y como $F_c = 60Hz$

$$R = \frac{1}{2\pi F_0 C} = \frac{1}{2\pi * 60Hz * 0.1uF} = 26.5K\Omega$$

$$R_1 = R_2 = R \rightarrow R_1 = R_2 = 26.5K\Omega$$

$$\rightarrow \quad R_1 = R_2 = 27k\Omega$$

$$C_1 = C_2 = C \rightarrow C_1 = C_2 = 0.1uF$$

$$\rightarrow \quad C_1 = C_2 = 0.1uF$$

$$R_3 = \frac{R}{2} = \frac{26.5k\Omega}{2} = 13.3K\Omega$$

$$\rightarrow \quad R_3 = 12k\Omega$$

$$C_3 = 2C = 2 * 0.1uF = 0.2uF$$

$$\rightarrow \quad C_3 = 0.2uF$$

Si $R_5 = 100k\Omega$

$$R_4 = 0.05 * R_5 : alto\ Q \rightarrow R_4 = 0.05 * 100K = 5K\Omega$$

$$\rightarrow \quad R_4 = 4.7k\Omega$$

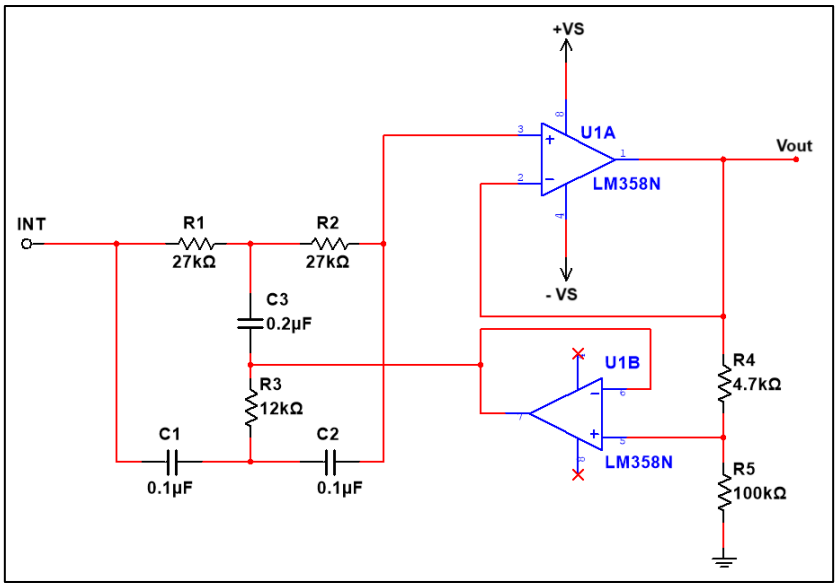


Figura 42. Filtro de muesca con los valores del diseño.

Fuente: Autoría propia.

Q hace referencia al ancho de banda (Figura 43) con lo que se desea trabajar para la atenuación de la frecuencia; para este caso se estableció una Q alto.

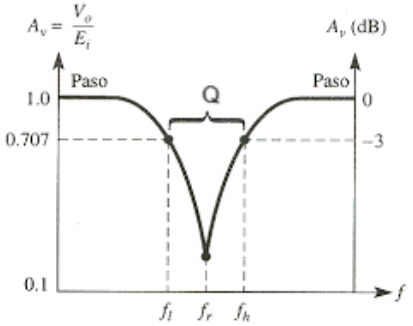


Figura 43. Ancho de banda Q para la atenuación de la frecuencia.

Fuente: Autoría propia.

3.2.2.3 Acondicionamiento de la señal

Una vez realizado el diseño del circuito de la amplificación inicial como los filtros analógicos y teniendo la información del valor de la señal EMG como se describió en (2.2.5). Tomando como referencia dicha señal EMG de $1mV_{ac}$, al darle una amplificación inicial de 304. Esta señal puede variar entre 0.1 y $0.3\ V_{ac}$, ya que los filtros analógicos tienen ganancias unitarias.

Para ello se debe amplificar alrededor de los $3V$. Se diseño un circuito para la amplificación final de la señal; por lo tanto, se realizó un circuito de

ganancia inversora, como también el uso de un diodo rectificador 1N4001 en configuración de media onda. Dicha configuración recorta las señales negativas y así obtenemos solo las señales positivas que es de nuestro interés. Por último, se utilizó un diodo Zener de $3.3V$ 1N4728A, como se puede ver en la Figura 44.

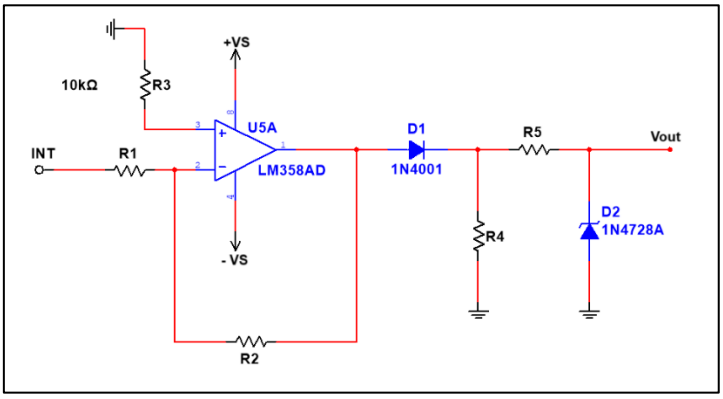


Figura 44. Etapa de acondicionamiento de la señal.

Fuente: Autoría propia.

$$Ganancia = -\frac{R_2}{R_1}$$

Ecuación 19. Fórmula para hallar la ganancia del circuito inversor.

$$R_3 = R_1 // R_2$$

Ecuación 20. Fórmula para hallar la resistencia R3 de la ganancia del circuito inversor.

Si $R_1 = 10k\Omega$ y una $Ganancia = 33$

$\rightarrow \quad R_2 = -Gain * R_1 = -33 * 10k\Omega = 330k\Omega$

$R_3 = R_1 // R_2 = 10k\Omega // 330k\Omega = 9.7k\Omega$

$\rightarrow \quad R_3 = 10k\Omega$

$\rightarrow \quad R_4 = 10k\Omega$

Si el voltaje es $V = 3.3V$ y la corriente $I = 20mA$

$$R_5 = \frac{V}{I} = \frac{3.3V}{20mA} = 165\Omega$$

$\rightarrow \quad R_5 = 220\Omega$

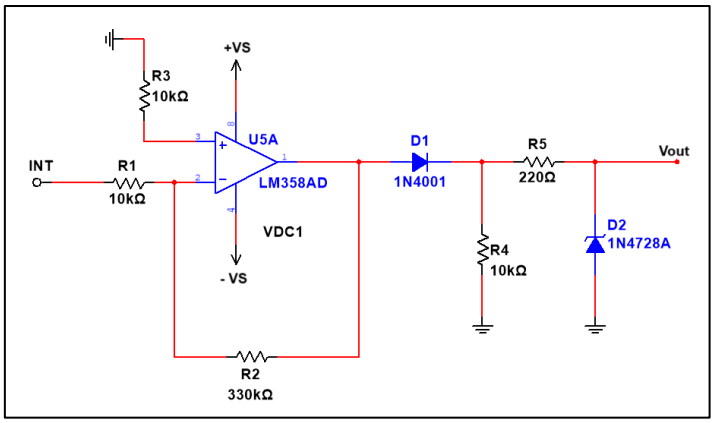


Figura 45. Etapa de acondicionamiento de la señal con los valores de diseño.

Fuente: Autoría propia.

La señal resultante tiene valores positivos que llegan alrededor de 0 hasta $3.04\ V$ al realizar la simulación en el software Multisim, como se puede ver en la siguiente Figura 46.

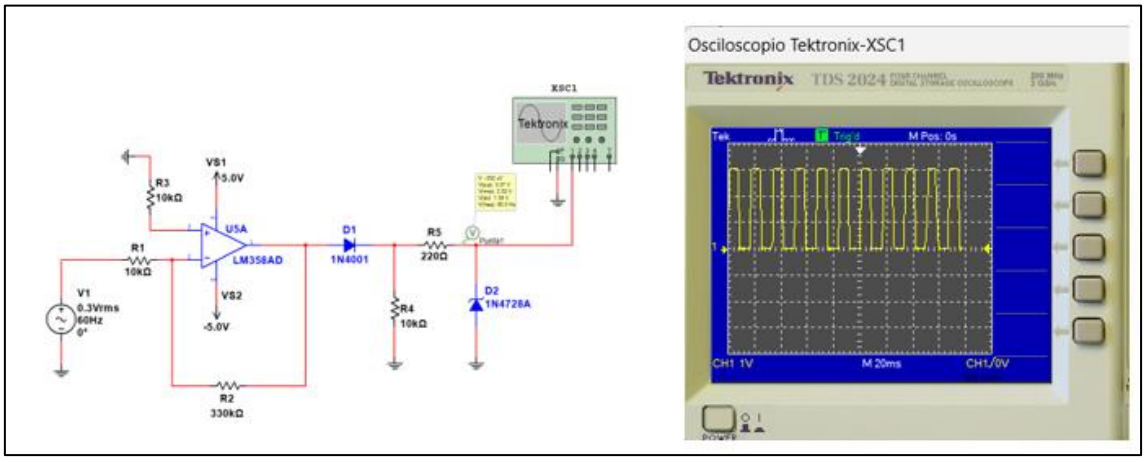


Figura 46. Simulación del circuito acondicionador de la señal.

Fuente: Autoría propia.

Una vez obtenido todo el diseño del circuito para la obtención de las señales musculares, se procedió a realizar la simulación en Multisim y corroborar que cumpla con todas las características que se detallan anteriormente con los cálculos realizados, como se puede ver en Anexo 2.

Cabe indicar que se omitió la primera etapa del circuito (amplificación inicial), ya que Multisim al simular un amplificador de instrumentación como el AD623 y al darle mucha ganancia, como en este caso de 304, este no nos permitirá realizar dicha simulación. Por estas razones se dio valores a las fuentes de voltaje Ac en serie de 0.1 Vrms y 0.15 Vrms como referencia, además se configuraron con frecuencias de 80 Hz y otra de 20 Hz respectivamente.

Podemos observar que las frecuencias por debajo de los 50 Hz se atenúan mientras que las frecuencias mayores a esta pasan a través del circuito, como podemos ver en la Figura 47 marcado con un rectángulo de color rojo. Cabe indicar que CH2 está conectado a la salida del circuito, mientras que el CH1 conectado en el inicio de dicho circuito.

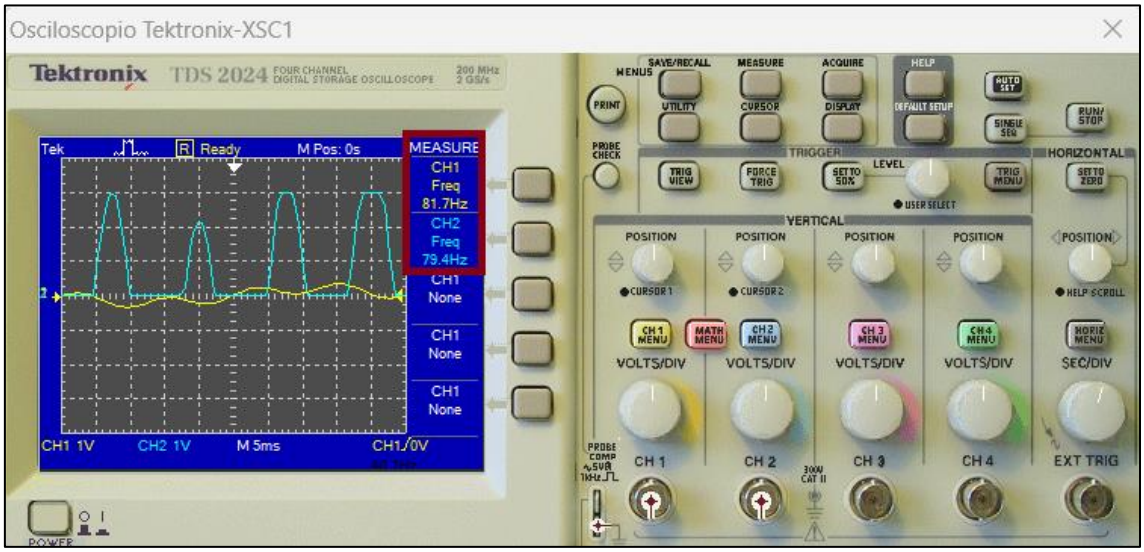


Figura 47. Visualización de la señal con frecuencias por debajo de 50 Hz.

Fuente: Autoría propia.

En Anexo 3, podemos observar el circuito configurado con frecuencias de 80 Hz y 400 Hz respectivamente; se observa que las frecuencias mayores a los 300 Hz se atenúan y deja pasar las frecuencias por debajo de estas, como podemos ver en la Figura 48.

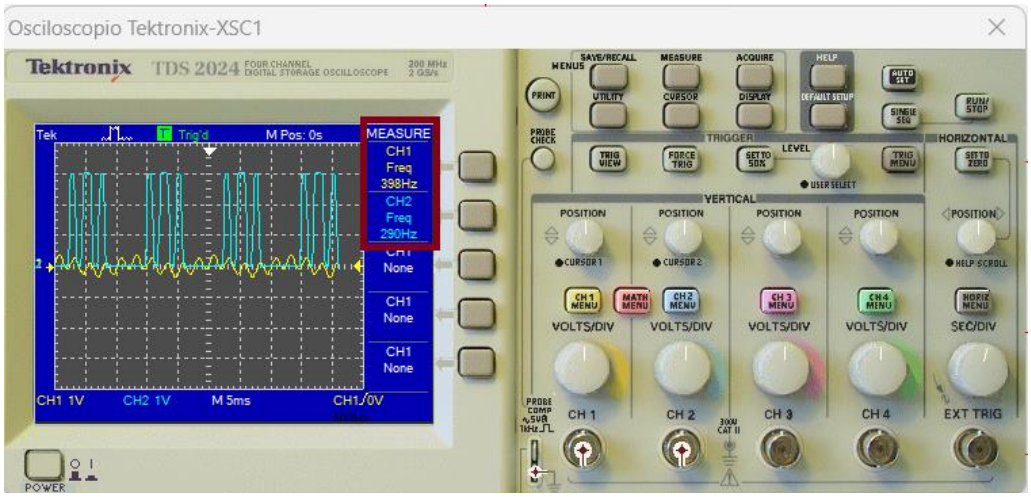


Figura 48. Visualización de la señal con frecuencia mayor a 300 Hz.

Fuente: Autoría propia.

Por último, se realiza la simulación con frecuencias de 50 Hz y 60 Hz respectivamente. Como podemos ver en Anexo 4, se puede observar en la Figura 49 cómo la frecuencia de 60 Hz se atenúa mientras que la frecuencia de 50 Hz pasa sin problema a través del circuito.

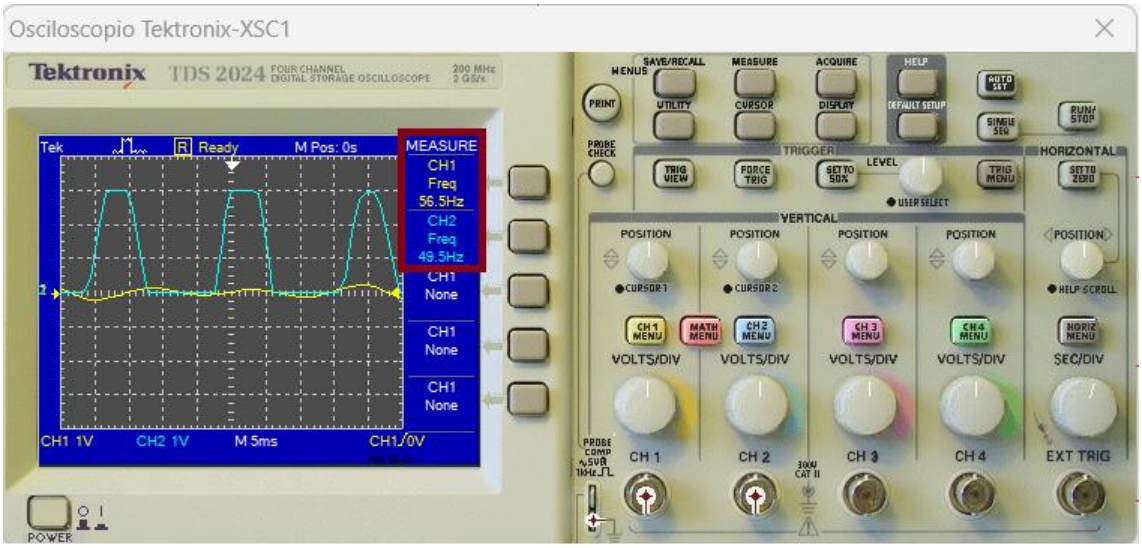


Figura 49. Visualización de la señal con frecuencias de 50 y 60 Hz.

Fuente: Autoría propia.

3.2.2.4 Visualización de la señal con el osciloscopio DSO138

La señal EMG se obtiene utilizando un circuito de acondicionamiento de señal, que fue diseñado y montado en una protoboard. Para optimizar la transferencia de señales entre la piel del usuario y la superficie de AgCl del electrodo, se usaron electrodos con gel conductor. El canal de señal EMG necesita tres electrodos: dos para captar la señal diferencial y uno que sirve como referencia a tierra. Este electrodo de referencia se posiciona en una zona del brazo con poca o ninguna musculatura, como el codo.

Los electrodos se colocaron sobre el flexor radial del carpo, como se muestra en la Figura 50, donde esta ubicación se indica en color azul. Por lo tanto, la disposición de los electrodos se realizó tal como se observa en la Figura 51.

Figura 50. Localización del flexor radial del carpo.

Fuente: [64].

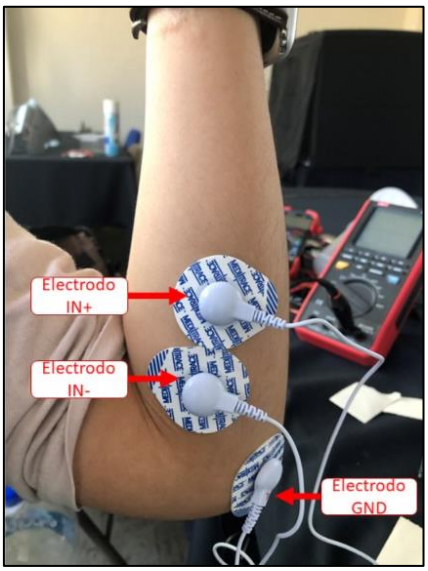


Figura 51. Colocación de electrodos sobre el flexor radial del carpo.

Fuente: Autoría propia.

Cada electrodo se conectó a la entrada del amplificador de instrumentación AD623 en sus respectivos pines, es decir, el electrodo IN+ en el pin número 3, el electrodo IN- en el pin número 2 y el electrodo GND va conectado al GND de la alimentación del circuito en general mediante cables para electrodos de broche, mientras que a la salida del circuito va conectado el osciloscopio, que viene incorporado dos cables con pinzas de cocodrilo de color rojo (Señal) y uno de color negro (GND), como podemos ver en la Figura 22. Entre la resistencia de 220Ω y el diodo Zener (Anexo 1) tiene que ir conectado el cable rojo, mientras que el cable negro va conectado al GND de la alimentación del circuito.

Cabe indicar que se realizaron modificaciones a estos cables de electrodos de broche para las respectivas pruebas, soldando cada cable por jumper macho, como podemos ver en la Figura 52.

Figura 52. Cable para electrodos modificado.

Fuente: Autoría propia.

Al realizar el gesto de puño, como se puede ver en la Figura 53, se visualiza la señal correspondiente a este movimiento en el osciloscopio DSO138 (Figura 54).

Figura 53. Gesto de puño.

Fuente: Autoría propia.

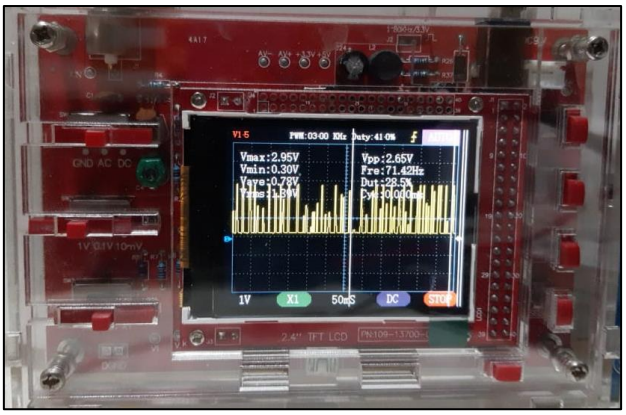


Figura 54. Visualización de la señal del gesto puño.

Fuente: Autoría propia.

Es importante mencionar que el osciloscopio se alimenta con una batería de 9V y está conectado a un broche de conector con plug DC (Figura 55) para garantizar la seguridad del paciente, evitando cualquier posible sobrecarga eléctrica.

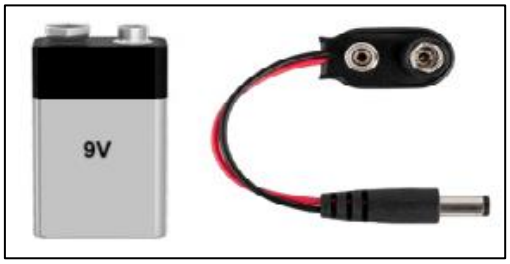


Figura 55. Broche conector para batería 9V con Plug DC.

Fuente: Autoría propia.

Finalmente, en la Figura 56 se puede observar la placa PCB terminal que se realizó con todos los componentes descritos anteriormente una vez completadas las pruebas. Adicional, se colocó el convertidor A/D ADS1115.

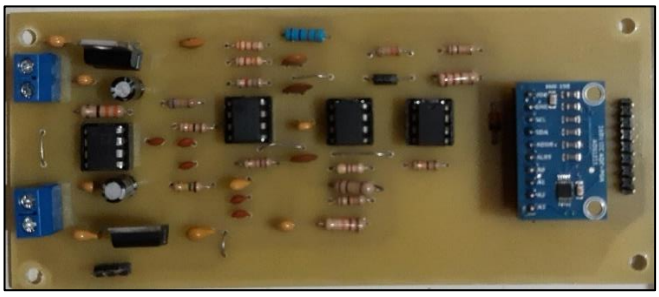


Figura 56. Placa final del EMG.

Fuente: Autoría propia.

3.2.3 Configuración de la Raspberry Pi 4

Para lograr adquirir la señal EMG a través de la Raspberry Pi 4, primero es necesario realizar algunas configuraciones que se detallan a continuación.

Inicialmente, la Raspberry Pi 4 se configura utilizando el programa "Raspberry Pi Imager". Una vez descargado e instalado, este software presenta una interfaz muy intuitiva. Este software es fundamental para instalar el sistema operativo en la Raspberry Pi 4, y requiere que una tarjeta SD esté conectada a la computadora para que el programa la reconozca automáticamente. La interfaz del programa puede verse en la Figura 57.

Figura 57. Interfaz Raspberry Pi Imager.

Fuente: Autoría propia.

Para cargar el sistema operativo en la tarjeta SD, se debe hacer clic en "CHOOSE OS". Esto desplegará varias opciones disponibles. Para el desarrollo de la propuesta, se seleccionó la primera opción, como se muestra en la Figura 58.

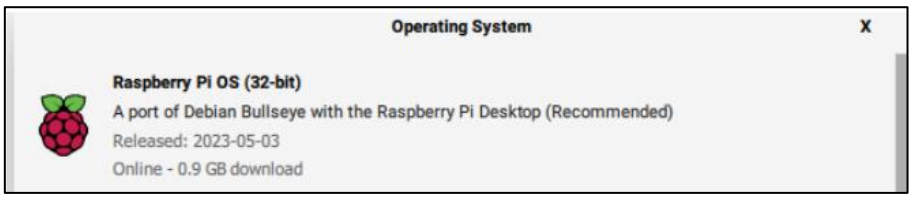


Figura 58. Sistema operativo.

Fuente: Autoría propia.

Después de completar los pasos anteriores y cargar el sistema operativo en la tarjeta SD, el siguiente paso es realizar la configuración inicial de la

Raspberry Pi. Para esto, se conecta la Raspberry Pi a un monitor externo, y al encenderla se mostrará una pantalla como se ilustra en la Figura 59.

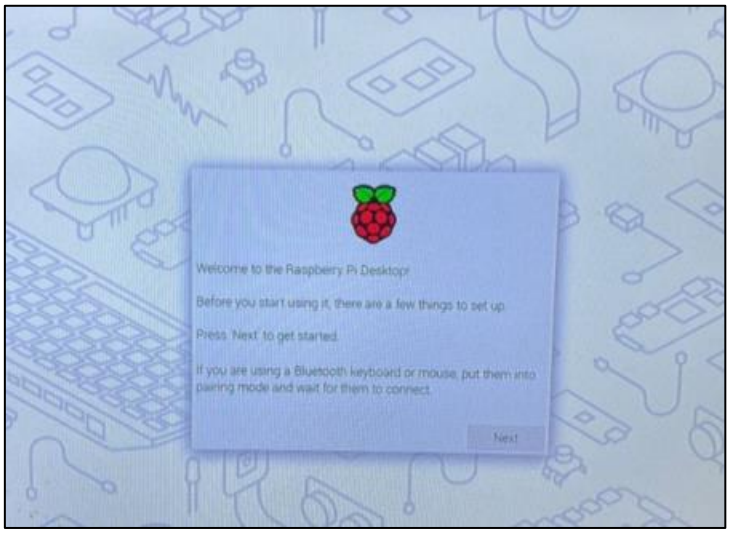


Figura 59. Configuración inicial de Raspberry.

Fuente: Autoría propia.

Después de avanzar a la siguiente ventana, se procede a seleccionar el país, idioma y zona horaria correspondientes. La configuración final se visualiza como se representa en la Figura 60.

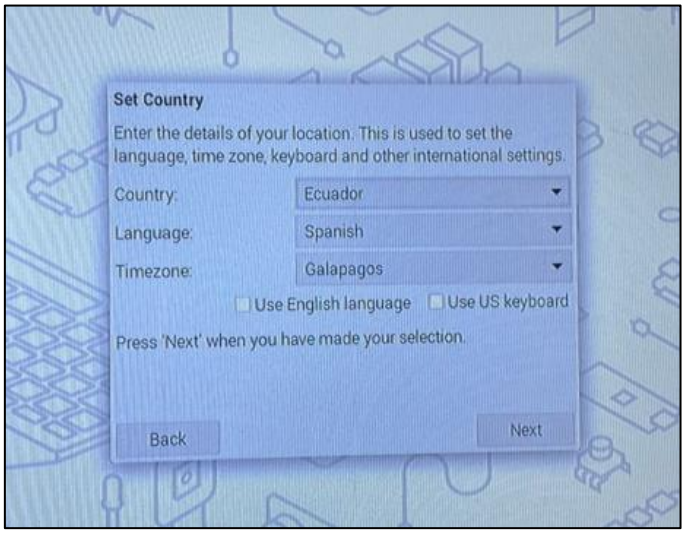


Figura 60. Selección de país, idioma, zona horaria en la Raspberry.

Fuente: Autoría propia.

Después de completar el paso anterior, en la siguiente ventana se debe crear un usuario y una contraseña específicos para la Raspberry Pi. Esta configuración se ilustra en la Figura 61.

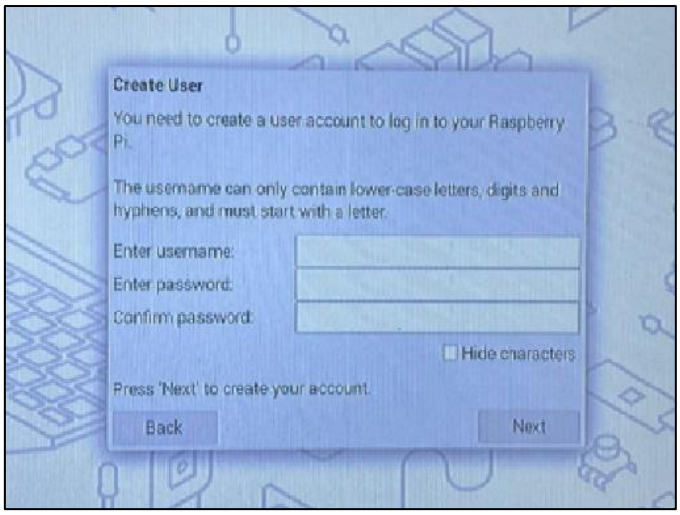


Figura 61. Usuario y contraseña de Raspberry.

Fuente: Autoría propia.

En esta propuesta es crucial que la Raspberry Pi funcione de manera remota al computador, por lo cual es necesario conectarla a una red Wifi. Este proceso se realiza siguiendo los pasos ilustrados en la Figura 62.

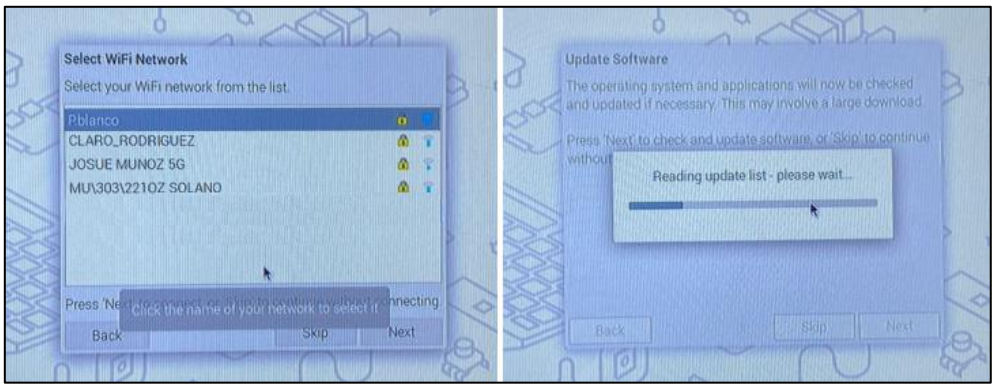


Figura 62. Configuración de la red Wifi en la Raspberry.

Fuente: Autoría propia.

Después de completar la configuración básica, se podrá visualizar la pantalla de inicio, tal como se muestra en la Figura 63.

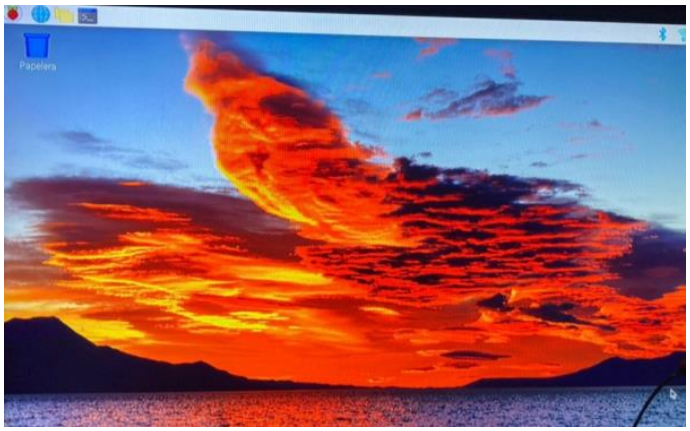


Figura 63. Pantalla de inicio de la Raspberry.

Fuente: Autoría propia.

Para poder realizar la comunicación remota, es necesario utilizar el programa RealVNC Viewer. Este programa permite conectar la Raspberry Pi a la computadora. A continuación, se explicarán las indicaciones necesarias para configurar esta conexión.

Para encontrar la dirección IP de la Raspberry Pi, es necesario abrir el símbolo del sistema y ejecutar las instrucciones que se muestran en la Figura 64.

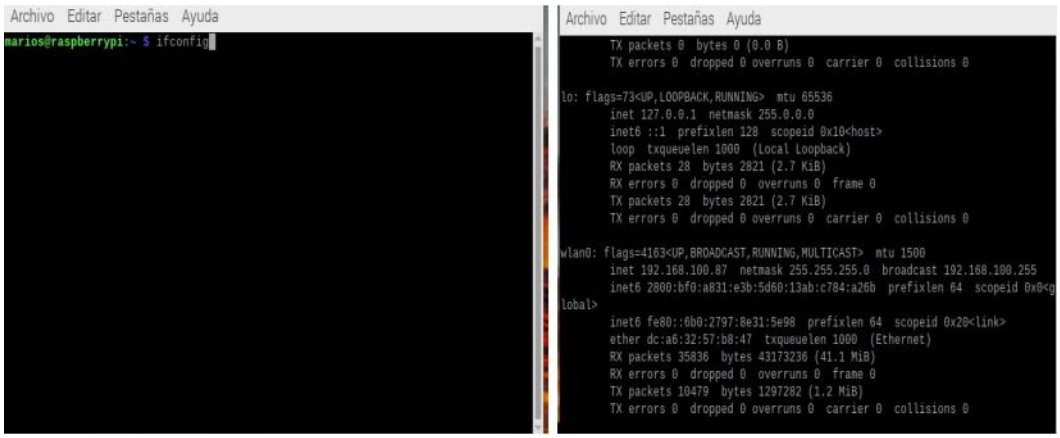


Figura 64. Ip de la Raspberry Pi.

Fuente: Autoría propia.

Una vez que se obtiene la dirección IP de la Raspberry Pi, es necesario activar las opciones de SSH y VNC en la configuración de la Raspberry Pi. Estas opciones se pueden encontrar y activar como se muestra en la Figura 65.

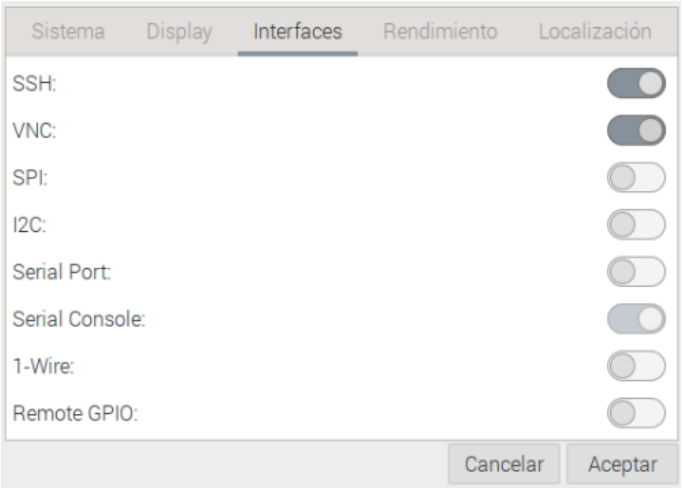


Figura 65. Activación de opciones para conexión remota.

Fuente: Autoría propia.

Una vez completado lo anterior, es el momento de utilizar RealVNC Viewer para establecer la conexión entre los dos dispositivos. Para ello, se debe abrir el programa y en el apartado de "RVNC CONNECT" digitar la Ip obtenida anteriormente, como se muestra en la Figura 66.

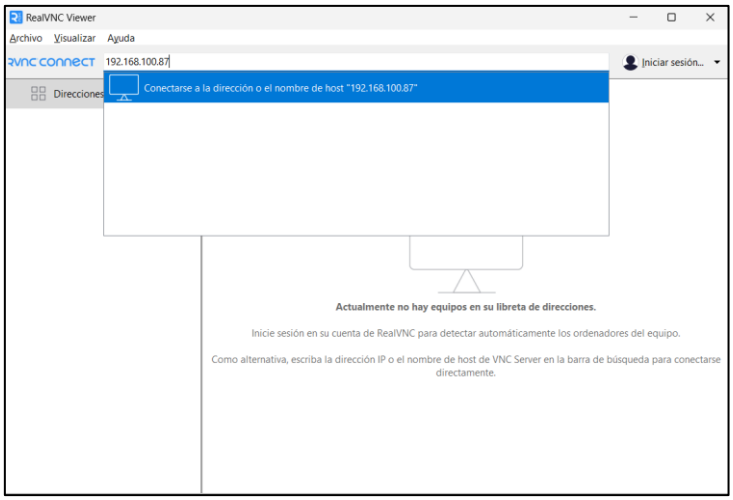


Figura 66. Interfaz RealVNC Viewer y colocación de la Ip.

Fuente: Autoría propia.

Para conectarse mediante este programa, es necesario establecer una nueva conexión donde se deben ingresar el nombre de usuario y la contraseña registrados en la Raspberry, como se muestra en la Figura 67.

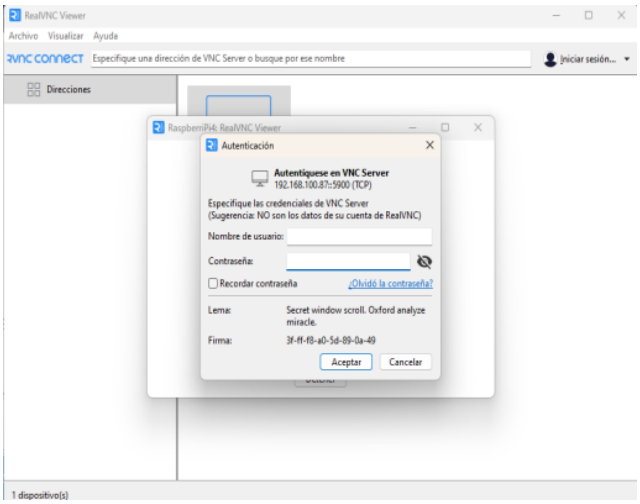


Figura 67. Usuario y contraseña para RealVNC.

Fuente: Autoría propia.

Finalmente, al realizar la conexión, es importante mencionar que, al ser la primera vez, el programa indicará que no se puede verificar la identidad del dispositivo. Sin embargo, se debe permitir la conexión al reconocer que es el dispositivo correcto, como se muestra en la Figura 68.

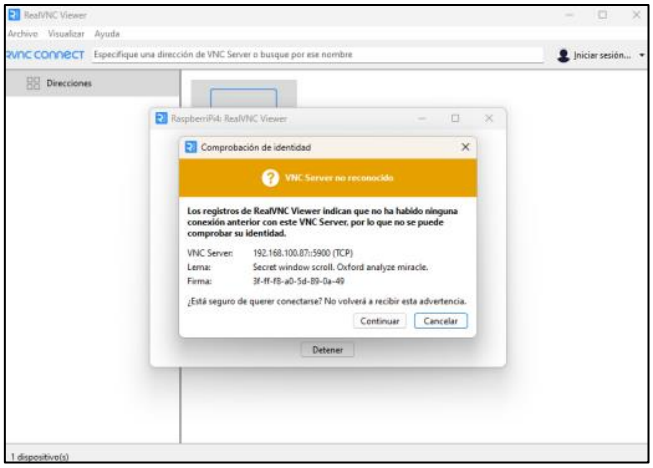


Figura 68. Permiso de conexión RealVNC.

Fuente: Autoría propia.

3.2.3.1 Instalación de Librerías

Para configurar todas las librerías necesarias, es fundamental utilizar varios comandos e instalarlos a través de la terminal de la tarjeta Raspberry Pi 4, los cuales son explicados a continuación.

Para instalar Python en la Raspberry Pi, es necesario ingresar el comando “sudo apt-get install python3” en la terminal, como podemos observar en la Figura 69.

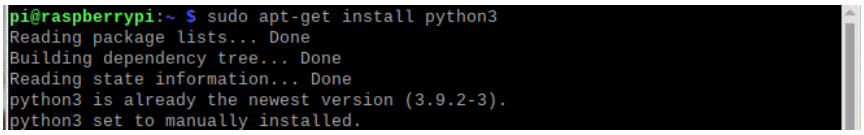


Figura 69. Instalación de Python.

Fuente: Autoría propia.

Una vez que Python está instalado en la Raspberry Pi, también es necesario instalar Python IDLE. Esto se realiza ingresando el comando “sudo apt-get install idle” en la terminal. Durante la instalación, se descargarán ciertos requisitos necesarios para su funcionamiento, como se muestra en la Figura 70.

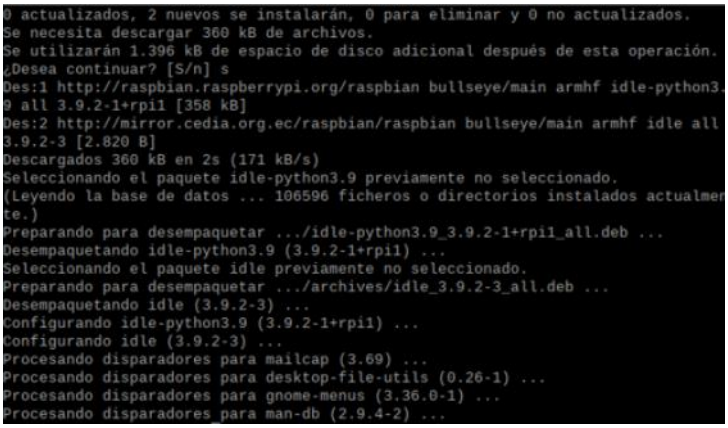


Figura 70. Instalación Python IDLE.

Fuente: Autoría propia.

Para instalar la librería Numpy, se debe utilizar el comando “sudo pip3 install -upgrade numpy”. Dado que esta librería puede presentar problemas para ser encontrada, es importante asegurarse de cuál versión de Python está instalada antes de proceder con la instalación. La Figura 71 muestra cuando ya está instalada la librería.

Figura 71. Instalación de la librería Numpy.

Fuente: Autoría propia.

Para instalar la librería TensorFlow, se debe utilizar el comando “sudo pip3 install -upgrade tensorflow”, como podemos ver en la Figura 72. Es una de las herramientas más populares y poderosas en el ámbito del aprendizaje automático y la inteligencia artificial.

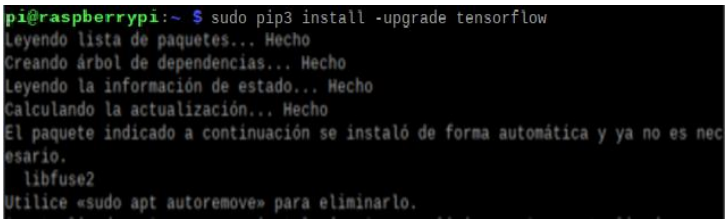


Figura 72. Instalación de la librería Tensorflow.

Fuente: Autoría propia.

Para instalar la librería pyserial, se debe utilizar el comando “sudo pip3 install -upgrade pyserial”, ver en la Figura 73. Es una biblioteca muy útil para interactuar con dispositivos serie (como puertos COM en Windows o puertos UART en sistemas embebidos como la Raspberry Pi).

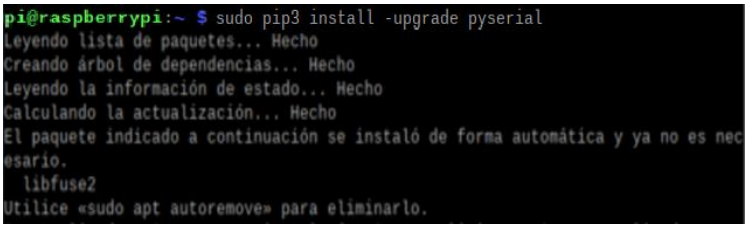


Figura 73. Instalación de la librería Pyserial.

Fuente: Autoría propia.

Para instalar Arduino IDE en la raspberry pi 4, se debe utilizar el comando “sudo apt-get install arduino”, ver en la Figura 74. Este método instalará la versión de Arduino IDE que está disponible en los repositorios de distribución de Raspberry Pi OS.

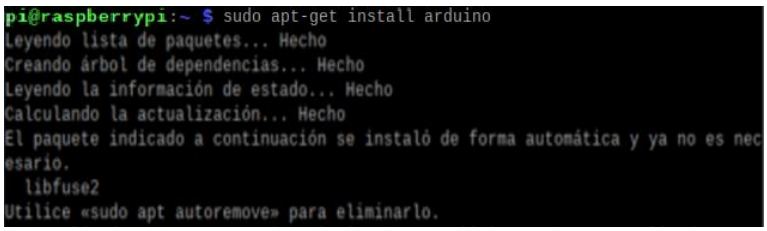


Figura 74. Instalación de Arduino IDE.

Fuente: Autoría propia.

Finalmente, es necesario ejecutar los comandos sudo apt update y sudo apt upgrade para asegurarse de que todo esté actualizado, como se muestra en la Figura 75.

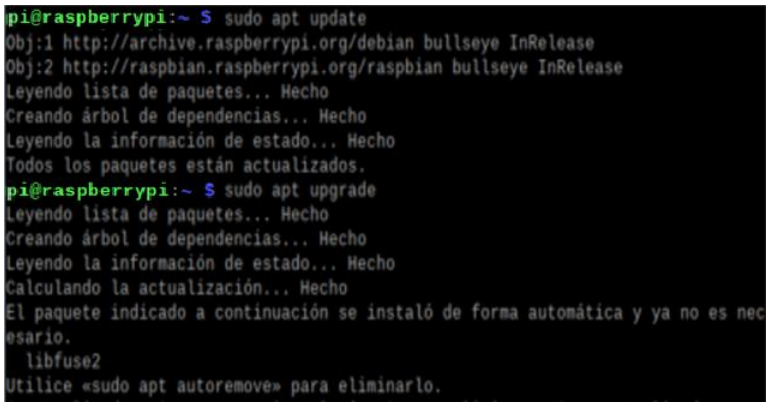


Figura 75. Actualización de todas las librerías.

Fuente: Autoría propia.

3.2.4 Comunicación entre la Raspberry Pi 4 y Arduino IDE

En este apartado se describe la comunicación serial entre el Arduino y la Raspberry Pi 4 a través de Python. Se desarrollaron los códigos necesarios para enviar datos de los diferentes gestos desde el Arduino y luego guardar estos datos en archivos con extensión*.npy* para su posterior entrenamiento. Esta comunicación se realiza de forma serial, es decir, a través de la conexión por USB, donde el Arduino es conectado en cualquier entrada USB que viene incorporada en la Rasberry Pi 4. A continuación, se detallarán los gestos y el proceso de guardado de los datos.

Para realizar dicha comunicación es necesario activar en la Raspberry Pi 4 la opción “Serial Port” como se puede ver en la siguiente figura 76.

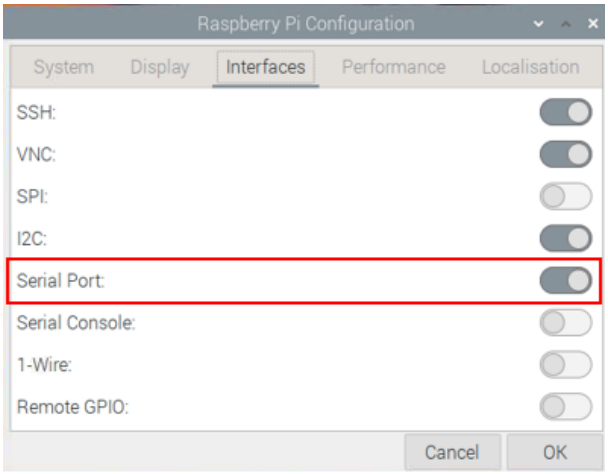


Figura 76. Activación Serial Port.

Fuente: Autoría propia.

Una vez completado este paso, abrimos el programa Arduino IDE, seleccionamos la placa con la que vamos a trabajar y verificamos el puerto al que está conectado. Esta información es necesaria para establecer la comunicación entre el Arduino IDE y Python, como se muestra en la Figura 77.

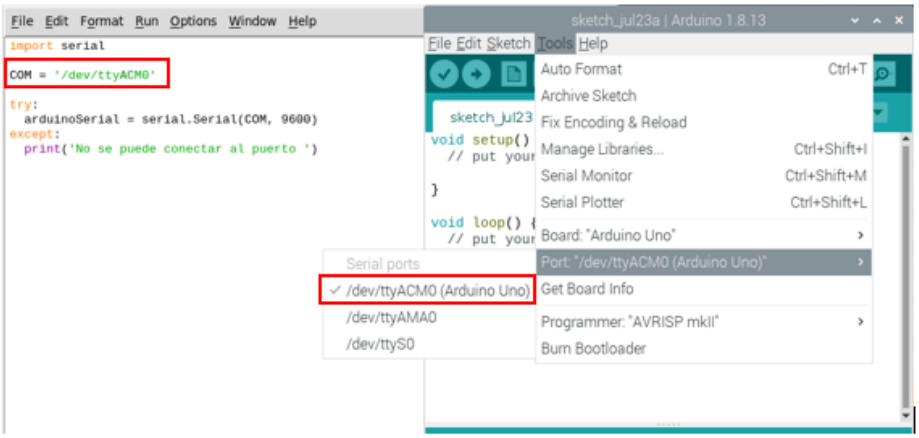


Figura 77. Comunicación Arduino IDE y Python.

Fuente: Autoría propia.

3.2.4.1 Extracción de características

La extracción de características es un paso crucial en el procesamiento de señales EMG para convertir las señales crudas en un conjunto de características más manejable y significativo. En el contexto de la EMG, las características comúnmente utilizadas como MAV (Valor Absoluto Medio), RMS (Raíz Cuadrada Media) y WL (Longitud de la onda) son fundamentales para analizar y clasificar las señales musculares.

1. **MAV (Mean Absolute Value – Valor Absoluto Medio)**

El MAV es una medida que proporciona una estimación de la amplitud media de la señal EMG. Se calcula promediando los valores absolutos de la señal. Esta característica es útil para evaluar el nivel general de actividad muscular [65].

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i|$$

Ecuación 21. Cálculo del MAV.

Donde:

- N es el número de muestras.

- x_i es el valor de la señal EMG en la muestra i.

2. RMS (Root Mean Square – Raiz Cuadrada Media)

El RMS es una medida de la magnitud de la señal EMG, que refleja tanto la amplitud como la energía de la señal. Es especialmente útil para analizar la intensidad de la contracción muscular [65].

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

Ecuación 22. Cálculo del RMS.

Donde:

- N es el número de muestras.
- x_i es el valor de la señal EMG en la muestra i.

3. WL (Waveform Length – Longitud de la Onda)

La WL mide la complejidad de la señal EMG y es una indicación de la cantidad total de actividad muscular durante el periodo de tiempo analizado. Refleja los cambios totales en la amplitud de la señal [65].

$$WL = \sum_{i=1}^{N} |x_{i+1} - x_i|$$

Ecuación 23. Cálculo de la WL.

Donde:

- N es el número de muestras.
- x_i y x_{i+1} son los valores de la señal EMG en las muestras i e x_{i+1} respectivamente.

Antes de calcular las características, es importante normalizar la señal EMG para que los valores estén en un rango de 0 a 1. Esto ayuda a reducir el impacto de las variaciones de amplitud entre diferentes muestras.

$$señal_{normalizada} = \frac{señal - \min(señal)}{\max(señal) - \min(señal)}$$

Ecuación 24. Fórmula para la normalización de la señal.

Donde:

- $señal$ es el vector de valores de la señal EMG
- $\min(señal)$ y $\max(señal)$ son los valores mínimo y máximo de la señal EMG.

Se seleccionaron tres clases de salida, en este caso gestos con las manos, en función de su utilidad y ubicuidad en aplicaciones prácticas. Estas clases son:

1. Clase 1: Meno cerrada/puño, como se muestra en la Figura 78.

Figura 78. Gesto mano cerrada/ puño.

Fuente: Autoría propia.

2. Clase 2: Pinza entre el dedo indicé y dedo pulgar, como se muestra en la Figura 79.

Figura 79. Gesto pinza.

Fuente: Autoría propia.

3. Clase 3: Mano abierta/descanso, como se muestra en la Figura 80.

Figura 80. Gesto mano abierta/descanso.

Fuente: Autoría propia.

3.2.4.2 Adquisición y procesamiento de la señal EMG

La señal EMG procesada se adquirió en intervalos de 1 segundo, con muestras tomadas cada 100 ms para cada gesto. Se desarrolló un código en el Arduino IDE (Anexo 4) para la extracción de características y se implementó un pulsador para facilitar la adquisición de la señal.

En Python se desarrolló el código necesario para guardar las señales en archivos con extensión.npy y para visualizar las características extraídas (Anexo 5). Se adquirieron 100 señales de cada gesto para su posterior uso en el entrenamiento del modelo.

En la Figura 81, 82 y 83, se puede observar las características MAV, RMS y WL de las 100 señales tomada del gesto mano cerrada/puño.

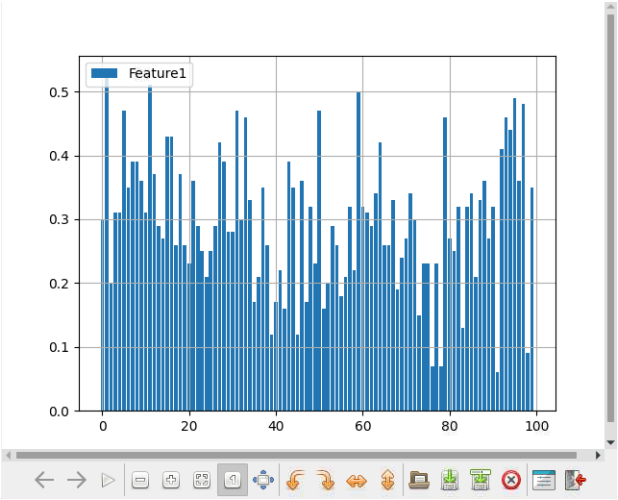


Figura 81. Características MAV de mano cerrada/puño.

Fuente: Autoría propia.

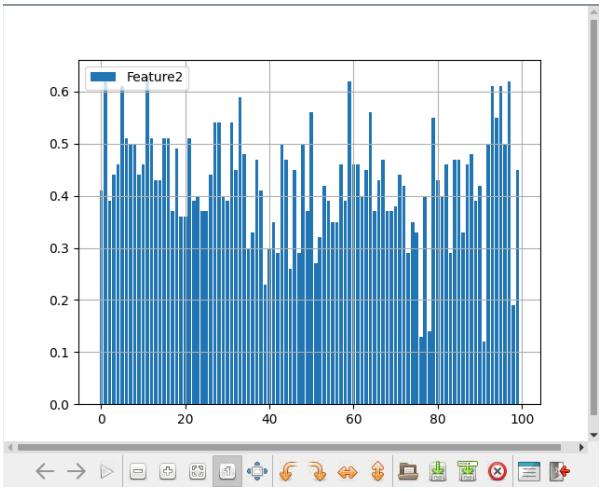


Figura 82. Características RMS de mano cerrada/puño.

Fuente: Autoría propia.

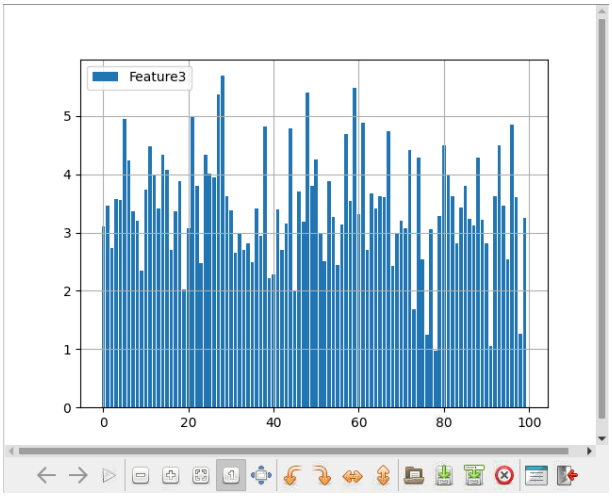


Figura 83. Características WL de mano cerrada/puño.

Fuente: Autoría propia.

En la Figura 84, 85 y 86, se puede observar las características MAV, RMS y WL de las 100 señales tomada del gesto pinza entre el dedo índice y dedo pulgar.

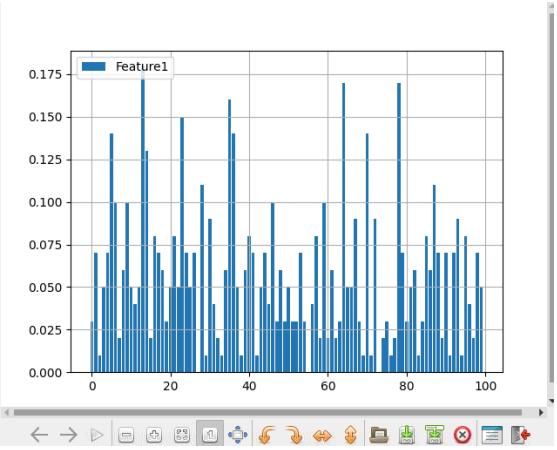


Figura 84. Características MAV del gesto pinza.

Fuente: Autoría propia.

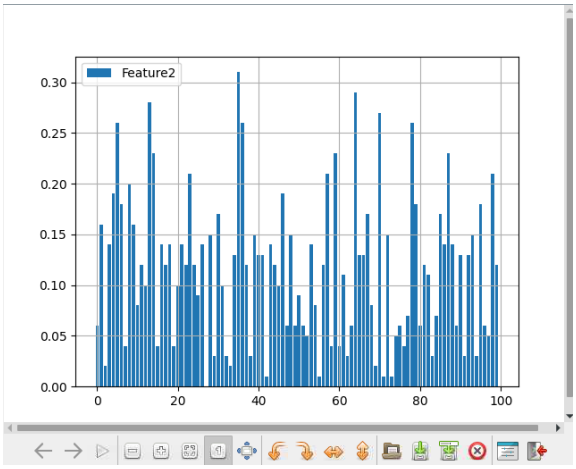


Figura 85. Características RMS del gesto pinza.

Fuente: Autoría propia.

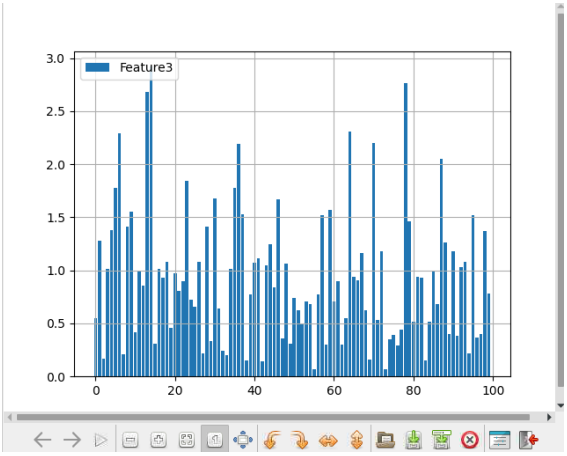


Figura 86. Característica WL del gesto pinza.

Fuente: Autoría propia.

En la Figura 87, 88 y 89 se puede observar las características MAV, RMS y WL de las 100 señales tomada del gesto mano cerrada/descanso.

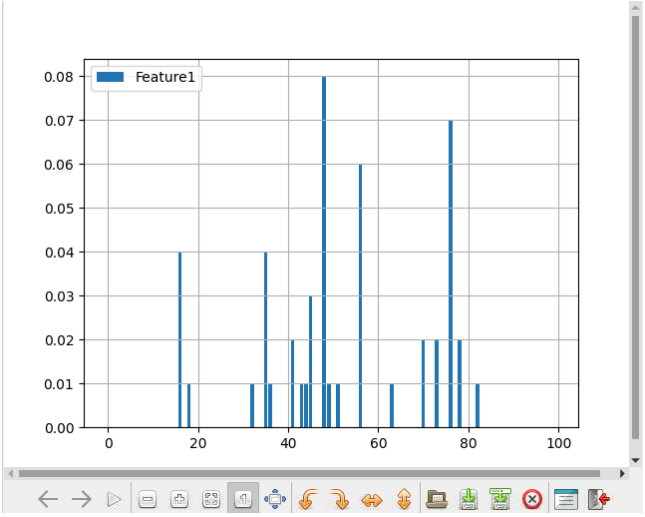


Figura 87. Características MAV de mano abierta/descanso.

Fuente: Autoría propia.

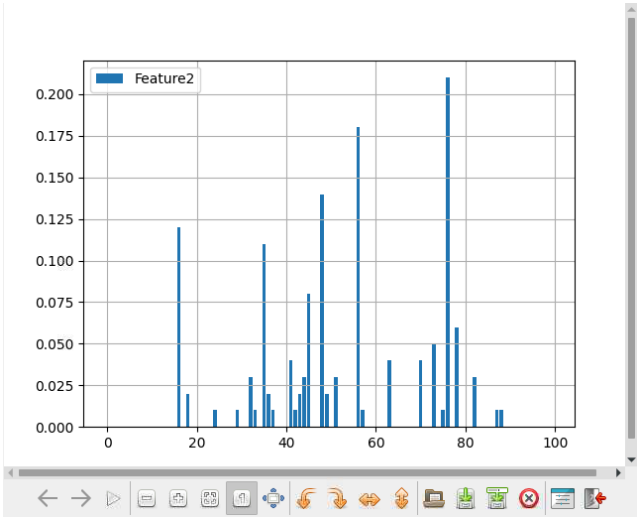


Figura 88. Características RMS de mano abierta/descanso.

Fuente: Autoría propia.

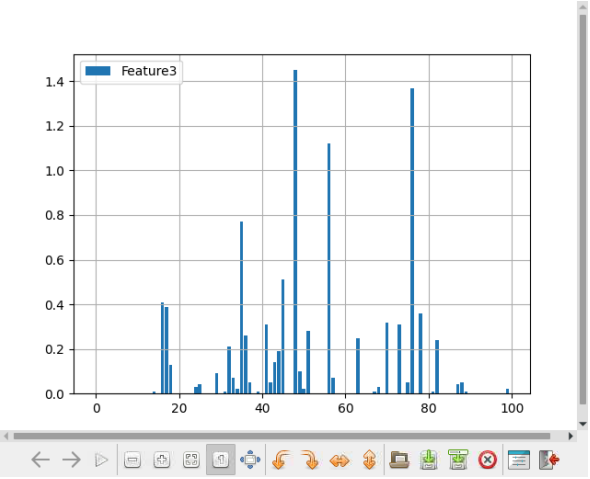


Figura 89. Característica WL de mano abierta/descanso.

Fuente: Autoría propia.

Una vez obtenidas las señales correspondientes a los tres tipos de gestos, se procedió a generar el conjunto de datos de entrada para el entrenamiento de las señales. Para ello, se concatenaron todas las señales en un único archivo P.npy. Además, se creó un archivo adicional para etiquetar los datos T.npy, con las siguientes clases:

- Clase 0: Gesto de mano cerrada.
- Clase 1: Gesto de pinza
- Clase 2: Gesto de mano abierta.

En la Figura 90, se puede observar los datos de los tres gestos con las tres extracciones de características, el MAV, RMS y WL. De 0 a 100 se tiene los datos del gesto mano cerrada, 100 a 200 se tiene los gestos de piza y los últimos 100 de mano abierta. Todo este proceso se realizó en Python (Anexo 6).

```
File  Edit  Shell  Debug  Options  Window  Help
= RESTART: /home/pi/Arduino/Extraccion_de_caracteristicas/generar_datos_de_entra
da.py
Size Input Data: (300, 3)
Size Class Data 1: 100
Size Class Data 2: 100
Size Class Data 3: 100
Size Output Data: (300, 1)
>>> T
array([[0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
```

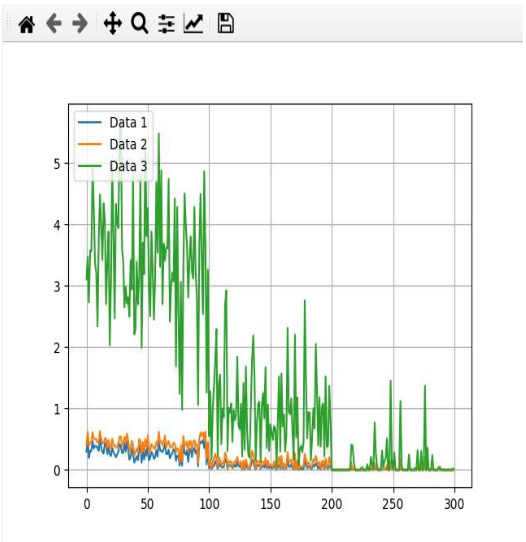


Figura 90. Conjunto de datos.

Fuente: Autoría propia.

3.2.5 Clasificación Multiclase con Redes Neuronales

En esta parte del proyecto se centra en el desarrollo y entrenamiento de un modelo de red neuronal para la clasificación de señales EMG en tres categorías distintas. El proceso incluye la carga de datos, preprocesamiento, construcción del modelo, entrenamiento, evaluación y almacenamiento del modelo entrenado. A continuación, se describen los pasos detalladamente.

3.2.5.1 Carga y Preprocesamiento de datos

En la Figura 91, se visualiza el código donde importamos las librerías correspondientes para realizar el entrenamiento de las señales EMG. Además, cargamos los archivos “P.npy” y “T.npy” que contienen el conjunto de datos y etiquetas respectivamente.

```
File  Edit  Format  Run  Options  Window  Help
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
import joblib
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Importar o generar conjuntos de datos
P = np.load('P.npy')
T = np.load('T.npy')
```

Figura 91. Importación de librerías y carga de los datos.

Fuente: Autoría propia.

3.2.5.2 Preprocesamiento de datos

El siguiente código de la Figura 92, se utiliza “StandarScaler” de “sklearn” para normalizar las características, asegurando que tenga una media de 0 y una desviación estándar de 1, lo que es crucial para la convergencia eficiente del modelo.

```
# Preprocesamiento de datos
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(P)
P = scaler.transform(P)
```

Figura 92. Preprocesamiento de los datos.

Fuente: Autoría propia.

3.2.5.3 Codificación de etiquetas

La siguiente línea de código de la Figura 93, las etiquetas se convierten en formato one-hot, necesario para la clasificación multiclase.

```
one_hot_labels = to_categorical(T, num_classes=3)
```

Figura 93. Codificación de etiquetas.

Fuente: Autoría propia.

3.2.5.4 División de datos

El conjunto de datos se divide en conjuntos de entrenamiento y prueba, reservando el 20% de los datos para pruebas. Para ello se utiliza el siguiente código, como se puede observar en la Figura 94.

```
# Dividir el conjunto de datos en conjuntos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
P_train, P_test, T_train, T_test = train_test_split(P, one_hot_labels, test_size=0.20, random_state=42)
```

Figura 94. Dividir el conjunto de datos en entrenamiento y prueba.

Fuente: Autoría propia.

3.2.5.5 Hiperparámetros

El código que se visualiza en la Figura 95, se definen los hiperparámetros clave como el número de épocas y la cantidad de nodos en las capas ocultas.

```
# Establecer hiperparámetros de algoritmo
epochs = 1000
hiddenNodes = 8
```

Figura 95. Hiperparámetros.

Fuente: Autoría propia.

3.2.5.6 Definición del modelo

Se construye un modelo secuencial con una capa de entrada con 3 características, con dos capas ocultas cada una con 8 nodos o neuronas y la función de activación ReLU, además con una capa de salida con 3 neuronas. La función de activación Softmax, adecuada para la clasificación en 3 clases. El código de la Figura 96 realiza todo este proceso.

```
# Definir la arquitectura de la red neuronal
model = Sequential()
model.add(Dense(hiddenNodes, activation='relu', input_dim=3))
model.add(Dense(hiddenNodes, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.summary()
```

Figura 96. Definir la arquitectura de la red neuronal.

Fuente: Autoría propia.

3.2.5.7 Compilación del modelo

La función de perdida que el modelo usará para evaluar cuán bien se está ajustando sus predicciones a las etiquetas verdaderas durante el entrenamiento.

El optimizador que se utilizará para ajustar los pesos del modelo durante el entrenamiento mediante el algoritmo Adam, que es una elección popular debido a su capacidad para adaptarse bien a una variedad de problemas de aprendizaje.

Finalmente, se compila el modelo con la función de perdida y el optimizador definidos, y además especifica que la métrica será la precisión ‘accuracy’. Esto prepara el modelo para el entrenamiento. En la Figura 97 se puede visualizar el código que realiza todo este proceso.

```
# Declarar la función de pérdida y el optimizador
loss = 'categorical_crossentropy'
optimizer = tf.keras.optimizers.Adam()

model.compile(loss=loss, optimizer=optimizer, metrics=['accuracy'])
```

Figura 97. Declarar la función de pérdida y el optimizador.

Fuente: Autoría propia.

3.2.5.8 Entrenamiento del modelo

En esta parte del código de la Figura 98, nos permite entrenar una red neuronal, utilizando una porción de datos de entrenamiento como el conjunto de validación. Después, genera un gráfico que muestra cómo cambia la pérdida del modelo tanto en el conjunto de entrenamiento como en el de validación a lo largo

de las épocas, permitiendo evaluar visualmente el rendimiento del modelo durante el entrenamiento.

```
# Entrenar el modelo
history = model.fit(P_train, T_train, epochs=epochs, verbose=1, validation_split=0.1)
plt.xlabel('Epocas')
plt.ylabel('Pérdida')
plt.plot(history.epoch, np.array(history.history['loss']), 'b', label='Train Loss')
plt.plot(history.epoch, np.array(history.history['val_loss']), 'r', label='Val Loss')
plt.legend()
plt.grid()
```

Figura 98. Entrenar el modelo.

Fuente: Autoría propia.

3.2.5.9 Evaluación del modelo

El siguiente código de la Figura 99 nos permite evalúar el modelo con el conjunto de los datos de prueba para determinar su precisión y pérdida.

```
# Evaluar el modelo
test_loss, test_acc = model.evaluate(P_test, T_test, verbose=1)
print('Exactitud de prueba: ', test_acc)
```

Figura 99. Evaluar el modelo.

Fuente: Autoría propia.

3.2.5.10 Almacenamiento del modelo

El modelo entrenado y el escalador se guardan para uso futuro. Para ello se utilizan las siguientes líneas de código que estan en la Figura 100.

```
# Guardar el modelo entrenado
model.save('modelo_entrenado.keras')

# Guardar el scaler
joblib.dump(scaler, 'scaler.pkl')
```

Figura 100. Almacenamiento del modelo y escalador.

Fuente: Autoría propia.

3.2.5.11 Visualización del modelo

En la Figura 101 está el código que nos permite visualizar la pérdida del entrenamiento y la validación para analizar el desempeño del modelo.

```
# Mostrar la gráfica de pérdida
plt.xlabel('Epocas')
plt.ylabel('Pérdida')
plt.plot(history.epoch, np.array(history.history['loss']), 'b', label='Train Loss')
plt.plot(history.epoch, np.array(history.history['val_loss']), 'r', label='Val Loss')
plt.legend()
plt.grid()
plt.show()
```

Figura 101. Visualización del modelo.

Fuente: Autoría propia.

3.2.6 Implementación del modelo entrenado.

El siguiente código en Python, está diseñado para leer datos del sensor EMG a través del Arduino, predecir un gesto utilizando un modelo de red neuronal previamente entrenado, y enviar la predicción de vuelta al Arduino.

En la Figura 102, importamos las librerías necesarias y además la configuración del puerto serie conectado con el Arduino. Cargar el modelo y el scaler.

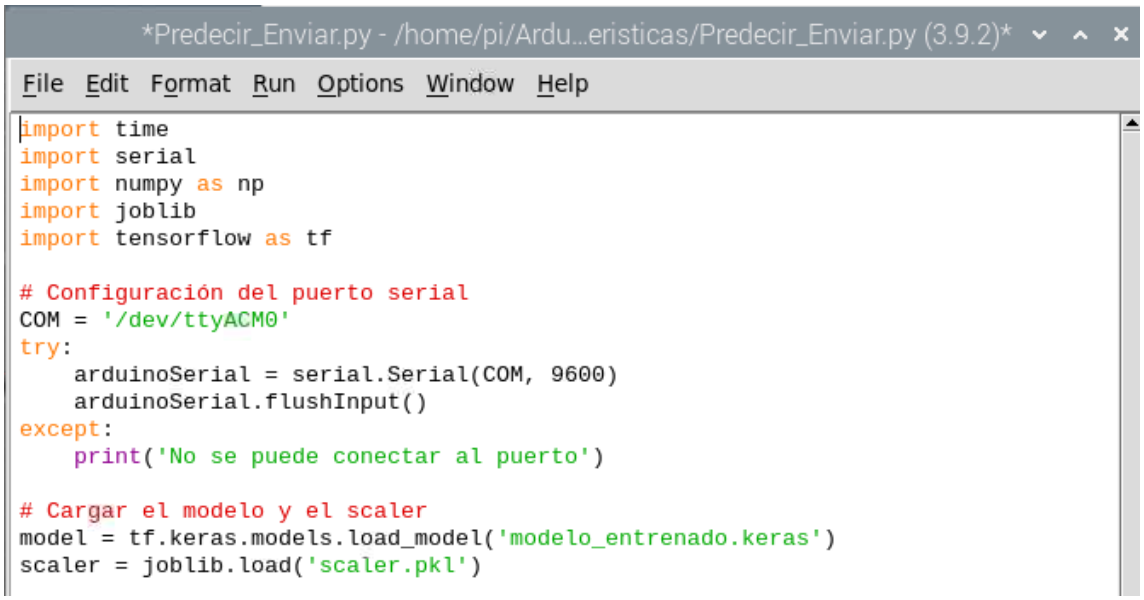


Figura 102. Configuración del puerto y carga de los archivos.

Fuente: Autoría propia.

En la Figura 103, se puede observar la función para leer las muestras provenientes desde el Arduino como el MAV, RMS y WL.

```
# Función para leer las muestras desde el Arduino
def leer_muestras():
    muestras = []
    for _ in range(3):  # Leer 3 características (MAV, RMS, WL)
        while arduinoSerial.in_waiting == 0:
            pass
        try:
            muestra = float(arduinoSerial.readline().decode('utf-8').strip())
            muestras.append(muestra)
        except ValueError:
            continue
    return np.array(muestras)
```

Figura 103. Función para leer las muestras desde el Arduino UNO.

Fuente: Autoría propia.

En la Figura 104 se observa el bucle principal asegurando que se han extraído las tres características, en base a esa característica realiza las

predicciones y envía de nuevo al Arduino para el control de los servos motores implementados en el prototipo.

```
# Bucle principal
try:
    while True:
        muestras = leer_muestras()
        if muestras.size == 3:  # Asegurarse de que se leyeron las 3 caracterist
            muestras = muestras.reshape(1, -1)
            muestras_normalizadas = scaler.transform(muestras)

            # Realizar predicción
            prediccion = model.predict(muestras_normalizadas)
            clase_predicha = np.argmax(prediccion, axis=1)
            print(f"Predicción: {clase_predicha[0]}")

            # Enviar la predicción al Arduino
            arduinoSerial.write(f"{clase_predicha[0]}\n".encode())

        #time.sleep(1)  # Esperar 1 segundo antes de la siguiente adquisición

except KeyboardInterrupt:
    pass

finally:
    if arduinoSerial.is_open:
        arduinoSerial.close()
```

Figura 104. Bucle principal para la predicción de los gestos.

Fuente: Autoría propia.

3.2.7 Diseño de la Prótesis de Mano Izquierda en Autodesk Fusion 360.

El diseño del prototipo de una mano izquierda en Autodesk Fusion 360 se inicia con la construcción de un dedo robótico que emplea un mecanismo de 4 barras para su movimiento. Este dedo robótico se compone de una falange, enlaces superior e interno, y una carcasa para el servomotor MG90S, que actúa como la cuarta barra, ver en la Figura 105.

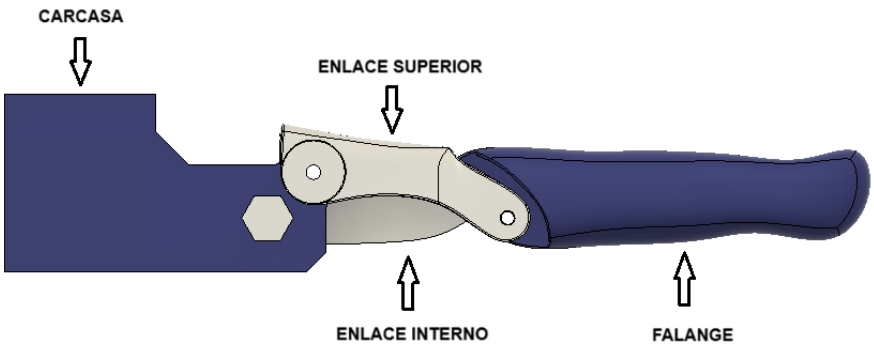


Figura 105. Partes del dedo robótico.

Fuente: Autoría propia.

Como se mencionó anteriormente en la sección de marco teórico, el diseño del dedo robótico se basa en la condición de Grashof y la condición de Barker. Para el diseño de la carcasa, que es el primer eslabón de la bancada, se

tomaron como referencia las medidas del servomotor MG90S, como se muestra en la Figura 106.

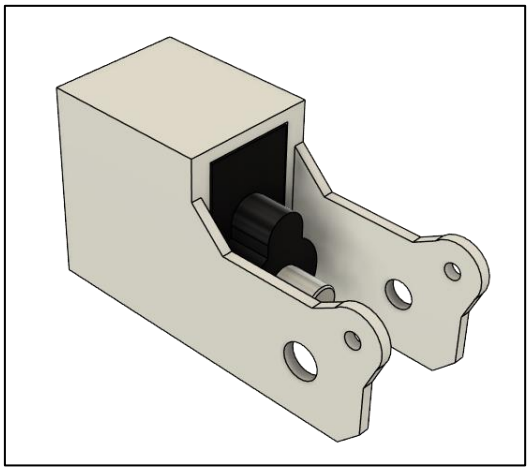


Figura 106. Carcasa del servo motor MG90S.

Fuente: Autoría propia.

Eslabón motriz o enlace inferior, a través de esta pieza se va a transmitir el movimiento a las demás partes, ver en la Figura 107.

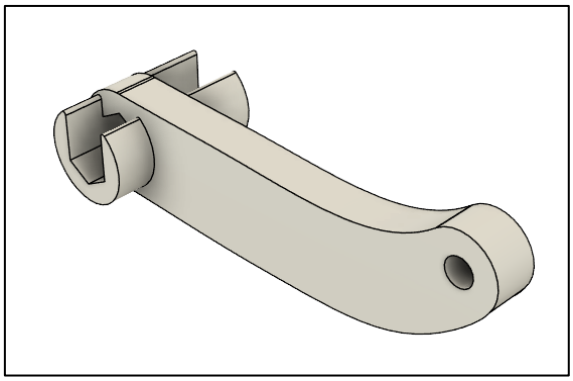


Figura 107. Enlace inferior.

Fuente: Autoría propia.

El eslabón acoplado consistirá en el cuerpo de la falange distal y una porción de la falange media, diseñado para que tenga una apariencia más parecida a un dedo humano real, como se ilustra en la Figura 108.

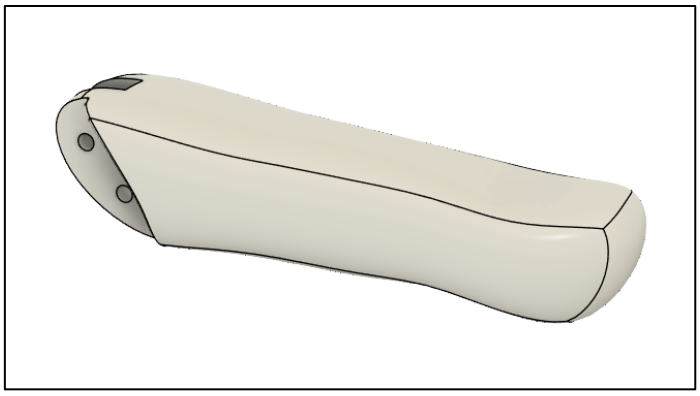


Figura 108. Falange.

Fuente: Autoría propia.

Y finalmente el eslabón superior el cual va a tener un movimiento de balancín, recordemos que este mecanismo es de Grashof y de acuerdo con la clasificación de Barker vamos a tener un movimiento de doble balancín ya que en esta ocasión el eslabón más pequeño el eslabón S va a hacer el acoplador, ver en la Figura 109.

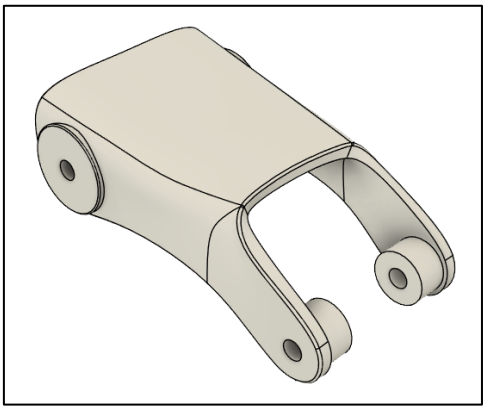


Figura 109. Enlace superior.

Fuente: Autoría propia.

Aparte de estos 4 eslabones se realiza los engranes cónicos, que el primer engrane va a ir al servomotor y el otro va a estar transmitiendo el movimiento precisamente del servomotor MG90S hacia los demás eslabones, como podemos ver en la siguiente Figura 110.

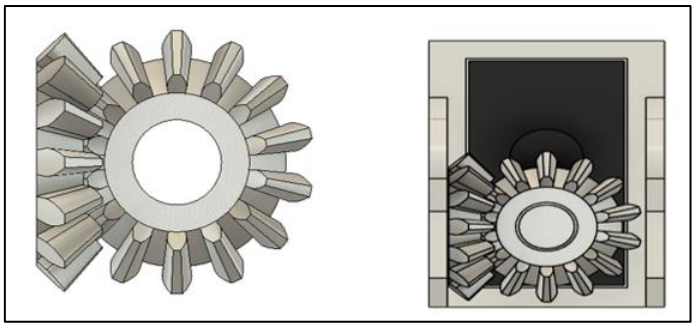


Figura 110. Engranes cónicos

Fuente: Autoría propia.

Además, se realizó una sexta pieza el cual es el eje, con este eje se va a generar la rotación y a transmitir el movimiento para que pueda rotar el eslabón motriz (enlace inferior), como podemos ver en la siguiente Figura 111.

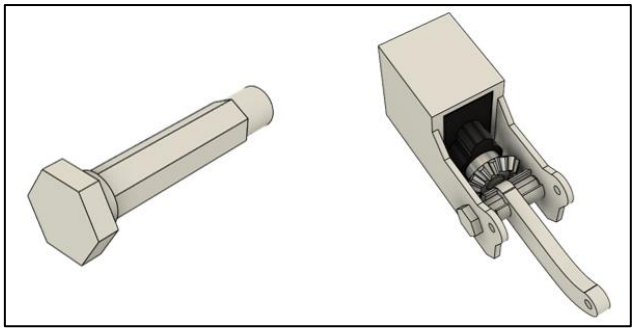


Figura 111. Eje motriz.

Fuente: Autoría propia.

Una vez realizado las piezas que forman parte del dedo robótico, se procedió a realizar una simulación en el software Fusion 360 y observar los movimientos que realiza, como podemos observar en la Figura 112.

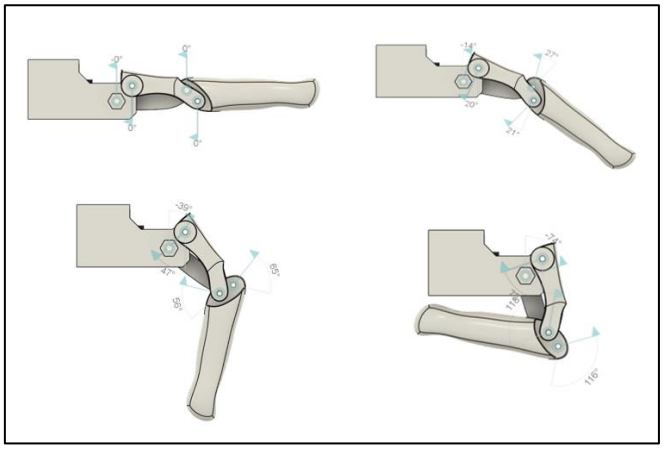


Figura 112. Movimientos del dedo robótico.

Fuente: Autoría propia.

Una vez finalizado el diseño del dedo robótico, se procedió a crear el dedo pulgar (Figura 113). Este diseño incluye dos movimientos: uno para la rotación de la falange y otro para el movimiento lateral de todo el dedo, permitiendo así el agarre en conjunto con el dedo índice, como se muestra en la Figura 114.

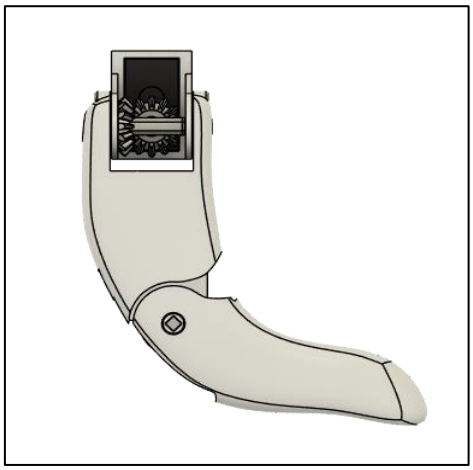


Figura 113. Dedo pulgar.

Fuente: Autoría propia.

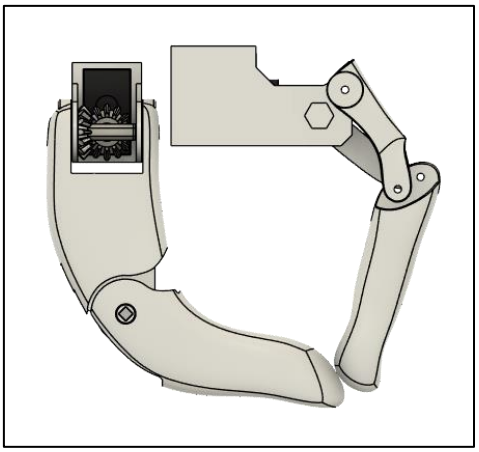


Figura 114. Agarre entre el dedo pulgar e índice.

Fuente: Autoría propia.

Es importante señalar que el dedo pulgar utiliza los mismos engranajes que el dedo robótico, con la única diferencia de que se realizaron modificaciones en la carcasa (bancada) utilizada en el dedo robótico. Adicionalmente, se ajustó el eje de rotación (Figura 115) para que encaje correctamente con los otros elementos del dedo pulgar, permitiendo así un movimiento adecuado, como se puede observar en la Figura 116.

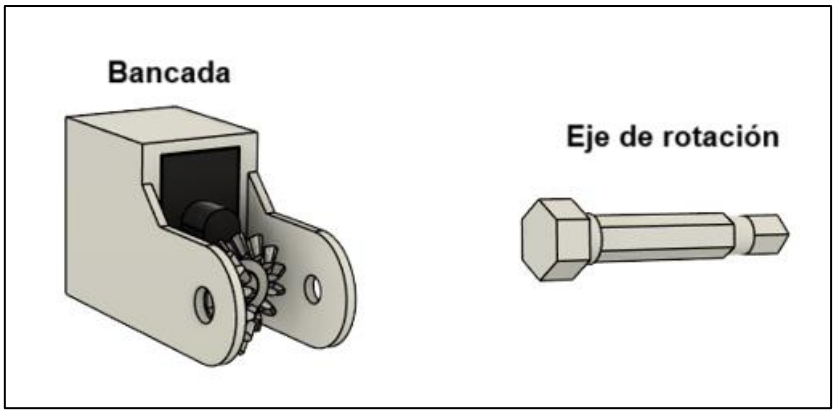


Figura 115. Bancada y Eje de rotación del dedo pulgar.

Fuente: Autoría propia.

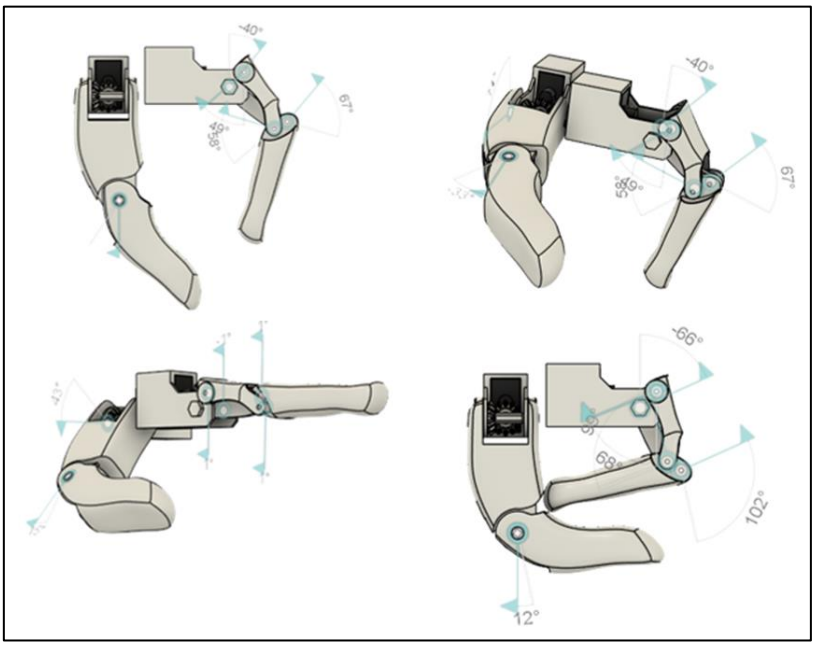


Figura 116. Movimientos del dedo pulgar.

Fuente: Autoría propia.

Para la construcción de la palma de la mano robótica, se utilizarán los mismos diseños del dedo robótico, específicamente el dedo índice, medio, anular y meñique, además del dedo pulgar. Cada uno de ellos estará ubicado a una distancia que se asemeje a la de una mano humana real, como se puede ver en la siguiente Figura 117.

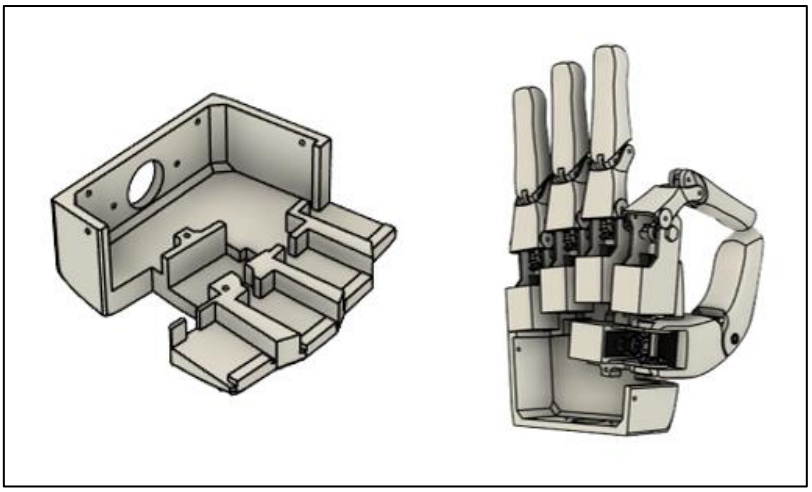


Figura 117. Palma de la Mano Robótica.

Fuente: Autoría propia.

Una vez realizada la construcción de la palma, se procedió a la construcción del dorso, como se puede ver en la Figura 118.

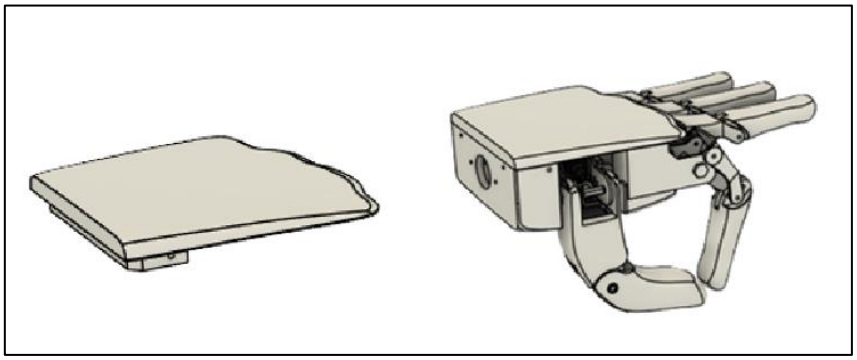


Figura 118. Dorso mano robótica.

Fuente: Autoría propia.

Como podemos ver en la Figura 119, Se realizó la construcción de un acople que va al final de la mano robótica. Esta pieza servirá para acoplar con la pieza del antebrazo y base, que se detallará más adelante.

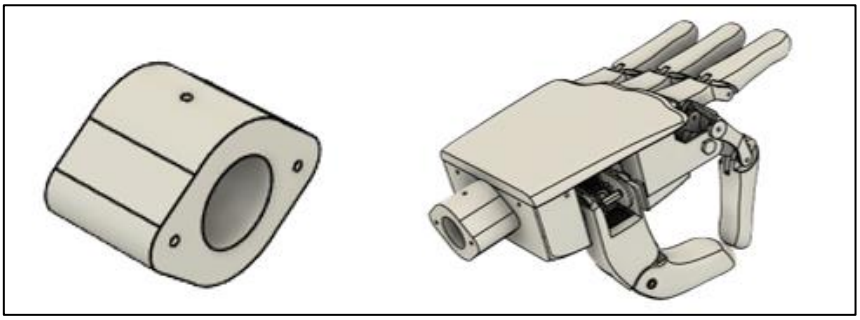


Figura 119. Acople mano robótica.

Fuente: Autoría propia.

Adicionalmente se realizó una base donde ira la mano robótica, como podemos ver en la Figura 120. Esta base fue diseñada para tener una mayor facilidad a la hora de realizar pruebas.

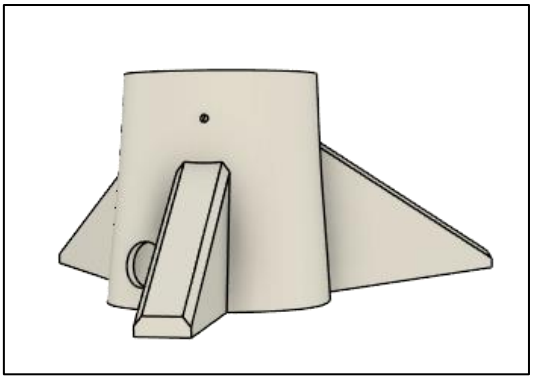


Figura 120. Base mano robótica.

Fuente: Autoría propia.

En la Figura 121 se puede observar el diseño terminado del prototipo de la prótesis de mano izquierda.

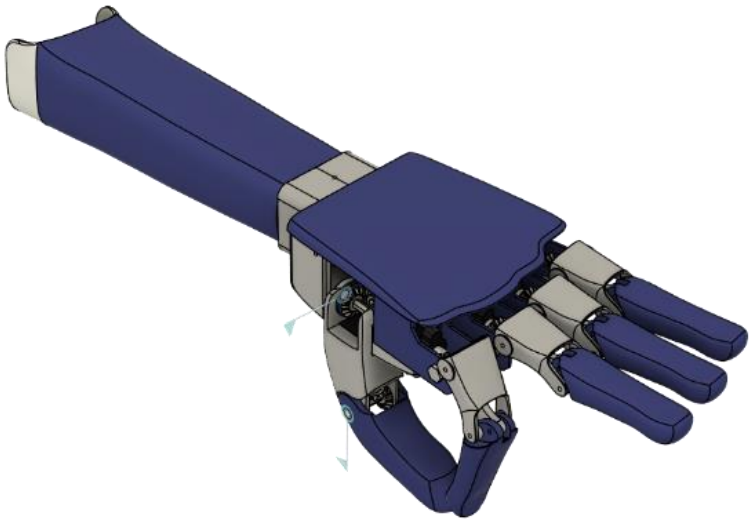


Figura 121. Prototipo 3D.

Fuente: Autoría propia.

3.3 Estudio de factibilidad

Para determinar el costo de la implementación del prototipo de una prótesis de mano izquierda mediante aprendizaje de máquina, que permite el reconocimiento de gestos manuales a través de un sensor mioeléctrico EMG con electrodos no invasivos, se detallan los costos en la Tabla 12.

Tabla 12. Valores para la implementación de la propuesta.

Cantidad	Descripción	Costo
1	Raspberry Pi 4	$250,00
1	Arduino Uno	$20,00
1	Gel Conductor	$9,00
50	Electrodos de Plata/Cloruro de Plata	$8.70
1	Osciloscopio DSO138	$50,00
1	Amplificador de Instrumentación AD623	$15,00
4	Amplificadores de Operación LM358	1,80
1	Diodo de Rectificación 1N4001	$0,10
1	Diodo Zener	$0,10
1	Cables para Electrodos de Broche	$15,00
17	Resistencias	$3,40
12	Capacitores	$3,90
1	Reguradores LM7805	$0,75
1	Regulador LM7905	$0,75
6	Pilas Recargables de 3.7V 8.800mAh	$23,40
3	Porta Pilas 18650 3.7V	$10,20
3	Modulo LM2596 - Regulador Step Down	$8,10
6	Servomotor MG90S	$30,00
1	Placa PCB	$35,00
1	Impresión 3D	$60,00
TOTAL		546,7

3.4 Pruebas y Resultados

La Figura 122 ilustra la evolución de la pérdida durante el proceso de entrenamiento y validación del modelo. Este gráfico es crucial para evaluar cómo se desempeña el modelo y para determinar si está ajustado adecuadamente a los datos.

- **Línea azul:** Representa la perdida en el conjunto de datos de entrenamiento.
- **Línea roja:** Representa la perdida en el conjunto de datos de validación.

Al analizar la pérdida a lo largo de las épocas, podemos obtener una comprensión más profunda del rendimiento del modelo. Si hay una diferencia significativa entre la pérdida del entrenamiento y la pérdida de validación, esto puede indicar que el modelo está sobreajustado a los datos de entrenamiento y no generaliza bien a nuevos datos. Por otro lado, si las pérdidas de entrenamiento y validación son similares, sugiere que el modelo se ajusta adecuadamente a los datos y está bien generalizado.

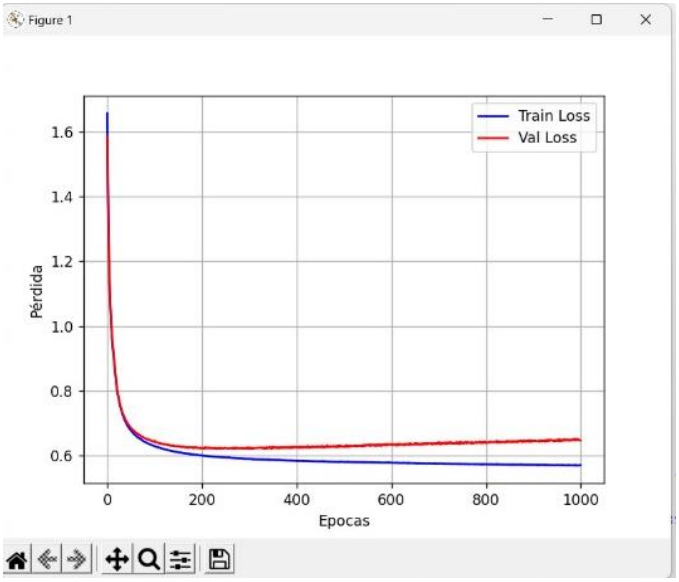


Figura 122. Visualización de la pérdida de entrenamiento y la validación.

Fuente: Autoría propia.

En la Figura 123 se puede visualizar los resultados del entrenamiento, obteniendo el 97%

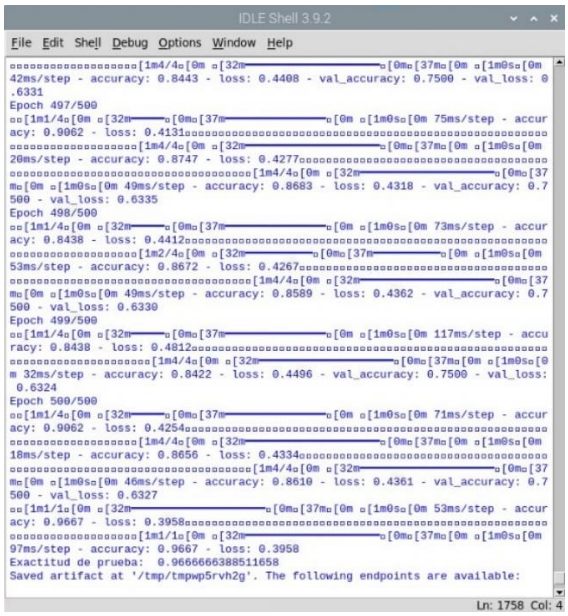


Figura 123. Resultados del entrenamiento.

Fuente: Autoría propia.

Durante las pruebas en tiempo real, el programa registro constantemente la señal EMG y a su vez la extracción de características en el Arduino y enviándolas como entrada a la red neuronal. Luego en base al modelo_entrenado.keras y el scarlet.pkl clasificó las predicciones Clase 0 – gesto de mano cerrada/puño, Clase 1 – gesto de pinza entre el dedo indicé y pulgar, Clase 2 – gesto de mano cerrada/descanso como podemos ver en la Figura 124.

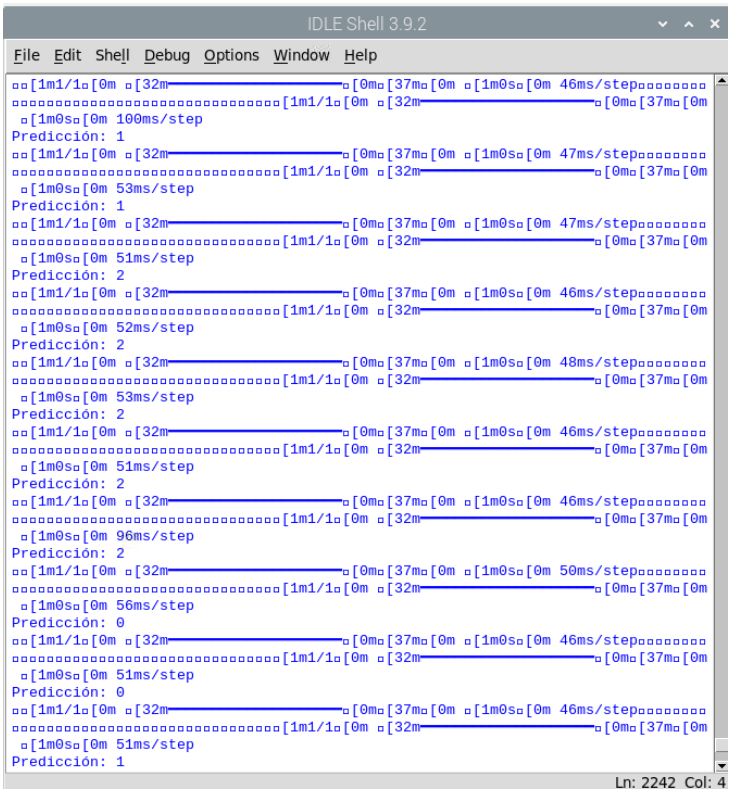


Figura 124. Predicciones en la consola de Python.

Fuente: Autoría propia.

CONCLUSIONES

El sensor EMG se ha destacado por su capacidad para captar con precisión la actividad eléctrica muscular, facilitando avances significativos en áreas como la medicina, la rehabilitación y la ingeniería robótica. Su habilidad para distinguir entre diferentes patrones musculares promete mejorar el diseño de prótesis y dispositivos de asistencia, permitiendo una integración más natural y eficaz con el movimiento humano. Para maximizar su utilidad, es esencial seguir desarrollando técnicas avanzadas de procesamiento de señales y mejorar la ergonomía de los electrodos. Estas mejoras podrían ampliar aún más su aplicación en contextos clínicos y prácticos.

El diseño de la mano robótica, desarrollado en Fusion 360 y utilizando servomotores MG90S para los movimientos de los dedos, representa un avance significativo en la integración de tecnologías accesibles para replicar movimientos precisos y naturales. La estructura robusta y los actuadores seleccionados han permitido crear una mano capaz de ejecutar una variedad de movimientos con eficacia. Para mejorar aún más su funcionalidad, se recomienda continuar refinando el diseño mecánico para optimizar la durabilidad y la ergonomía, así como explorar opciones para mejorar la precisión y la respuesta de los servomotores en diferentes condiciones operativas. Estos pasos podrían fortalecer su aplicación en áreas como la robótica educativa, la rehabilitación y la asistencia personalizada.

El entrenamiento de redes neuronales con señales EMG ha demostrado ser una técnica efectiva para la clasificación y predicción de los tres gestos de la mano. A lo largo de este proyecto, hemos desarrollado y entrenado un modelo que logra identificar los gestos a partir de señales EMG.

RECOMENDACIONES

Para realizar pruebas del sensor diseñado en una protoboard, es recomendable mantener el circuito lo más compacto posible. Además, es importante asegurarse de que los cables que conectan cada elemento del sensor no se crucen entre sí. Esto se debe a que los tipos de alambres utilizados en este tipo de pruebas pueden actuar como antenas, introduciendo ruido en la señal.

Al diseñar y realizar una amplificación inicial para la adquisición de señales musculares, es recomendable evitar una ganancia excesiva. Un aumento exagerado podría amplificar señales no relacionadas con la contracción muscular, introduciendo ruido y distorsionando los datos.

Al realizar mediciones con el sensor EMG, es crucial utilizar baterías para la alimentación. Si se desea conectar el sensor al computador para adquirir las señales, también es importante que el computador y cualquier otro microcontrolador o microprocesador como en este caso la Raspberry Pi no esté conectado a la fuente de voltaje de la red eléctrica para evitar sobrecargas eléctricas que puedan afectar al paciente.

Para el diseño del prototipo en este proyecto, se recomienda utilizar engranajes de metal en lugar de plástico. Con el tiempo, los engranajes de plástico pueden aflojarse, lo que puede causar un desequilibrio en los dedos y hacer que no se mantengan firmes.

Aumentar la cantidad de datos de entrenamiento puede mejorar la precisión del modelo y su capacidad para generalizar. Es recomendable recolectar más muestras en diferentes condiciones y variaciones de los gestos.

BIBLIOGRAFÍA

[1] E. A. C. a. M. E. Benalcázar, «Real-Time Hand Gesture Recognition Model Using Deep Learning Techniques and EMG Signals,» *2019 27th European Signal Processing Conference (EUSIPCO),* 2019.

[2] M. S. B. Zehra KARAPINAR SENTURK, «Machine Learning-Based Hand Gesture Recognition via EMG Data,» *Advances in Distributed Computing and Artificial Intelligece Journal,* 2021.

[3] C. C. J. N. M. L. G. X. J. C. M. C. S. L. Yang Zhou, «EMG Signal Processing for Hand Motion Pattern,» *Archives of Orthopaedics,* 2020.

[4] T. l. D. s. Reservados, «Discapacitados,» Banco Mundial, 03 2021. [En línea]. Available: https://www.bancomundial.org/es/topic/disability#:~:text=Entre%20110%20millones%20y%20190,ven%20afectadas%20por%20discapacidades%20importantes.. [Último acceso: 08 12 2022].

[5] T. l. d. s. reservados, «CARACTERIZACIÓN DEL SERVICIO DE PERSONAS CON DISCAPACIDAD,» *Coordinación General de Investigacion y Datos de Inclusión,* pp. 6-9, 2017.

[6] T. l. d. reservados, «Ecuador fabrica prótesis externas con orientación social,» La Hora, 31 12 2019. [En línea]. Available: https://www.lahora.com.ec/secciones/ecuador-fabrica-protesis-externas-con-orientacion-social/#:~:text=Con%20el%20fin%20de%20mejorar,los%20requerimientos%20de%20cada%20persona.. [Último acceso: 03 01 2023].

[7] N. G. G. L. Karla Pavlova Avilés Mendoza, «Diseño e implementación de una prótesis robótica con señales EMG usando técnicas de Inteligencia Artificial,» *PROYECTO INTEGRADOR ,* 2021.

[8] P. Escabí Monty, «Biosignal Processing,» *Introduction to Biomedical Engineering (Third Edition),* 2012.

[9] T. l. d. reservados, «myoelectric,» PCMag.com, 2023. [En línea]. Available: https://www.pcmag.com/encyclopedia/term/myoelectric#:~:text=The%20electrical%20signals%20within%20the,used%20to%20move%20prosthetic%20limbs..

[10] M. B. I. R. M. A. B. M. A. A. A. B. K. C. y. T. G. C. Rubana H. Chowdhury, «Surface Electromyography Signal Processing and Classification Techniques,» *Sensors,* 2013.

[11] A. S. L. F. y. J. C. Castro Vaca, «Diseño e implementación de una prótesis biónica de mano, muñeca y antebrazo, con capacidad de manipulación y rotación, controlada por inteligencia artificial a través de señales mioeléctricas, utilizando un sistema de entrenamiento en un entorno virtual p,» *Trabajo de Titulación,,* 2022.

[12] B. G. Zoe Brown, «Biological Signals and their Measurement,» *Update in Anaesthesia,* 2003.

[13] T. l. d. reservados, «What are the main functions of the muscular system,» MedicalNewsToday, 25 04 2018. [En línea]. Available: https://www.medicalnewstoday.com/articles/321617. [Último acceso: 03 02 2023].

[14] L. Burgess, «WHAT ARE THE MAIN FUNCTIONS OF THE MUSCULAR SYSTEM,» NEW MEXICO ORTHOPAEDICS, [En línea]. Available: https://nmortho.com/what-are-the-main-functions-of-the-muscular-system/#:~:text=Muscles%20allow%20a%20person%20to,rely%20on%20the%20muscular%20system.. [Último acceso: 05 02 2023].

[15] E. Equipo editorial, «Sistema muscular,» ENCICLOPEDIA HUMANIDADES, 10 08 2018. [En línea]. Available: https://humanidades.com/sistema-muscular/. [Último acceso: 07 02 2023].

[16] T. l. d. reservados, «MÚSCULO ESQUELÉTICO,» fisioonline, [En línea]. Available: https://www.fisioterapia-online.com/glosario/musculo-esqueletico. [Último acceso: 10 02 2023].

[17] I. DrTango, «Tipos de tejido muscular,» MedlinePlus, 28 08 2019. [En línea]. Available: https://medlineplus.gov/spanish/acercade/uso/citar/. [Último acceso: 12 02 2023].

[18] T. l. d. reservados, «Que es músculo,» fisioonline, [En línea]. Available: https://www.fisioterapia-online.com/glosario/musculo. [Último acceso: 15 02 2023].

[19] T. l. d. reservados, «QUÉ ES CONTRACCIÓN MUSCULAR,» fisioonline, [En línea]. Available: https://www.fisioterapia-online.com/glosario/contraccion-muscular#:~:text=La%20contracci%C3%B3n%20muscular%2C%20es%20un,esquel%C3%A9tico%20o%20del%20m%C3%BAsculo%20card%C3%ADaco.. [Último acceso: 16 02 2023].

[20] M. Á. Maroto García, «Protesis de antebrazo con control mioeléctrico mediante Machine Learning,» *Escuela de ingenierías industriales ,* 2020.

[21] J. M. Heras, «Regresión Lineal: teoría y ejemplos en Python,» IArtificial.net, 02 10 2020. [En línea]. Available: https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/. [Último acceso: 18 02 2023].

[22] «Qué son las redes neuronales y sus funciones,» ATRIA INNOVATION, 22 10 2019. [En línea]. Available: https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/. [Último acceso: 18 02 2023].

[23] H. J. M. R. P. Hevert Vivas, «Método secante estructurado para el entrenamiento del perceptrón multicapa,» Revistas de Ciencias, 22 08 2014. [En línea]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-19352014000200010#:~:text=El%20perceptr%C3%B3n%20multicapa%20es%20una,capas%20intermedias%20denominadas%20capas%20ocultas.. [Último acceso: 20 02 2023].

[24] T. l. d. reservados, «Qué es un software CAD y para qué sirve,» Integral innovation experts, 20 08 2019. [En línea]. Available: https://integralplm.com/blog/2019/08/20/que-es-

cad/#:~:text=El%20software%20de%20dise%C3%B1o%20asistido,Computer%2DAided%20Design%20%26%20Drafting.. [Último acceso: 20 002 2023].

[25] «Microprocesador,» Significados.com, 27 02 2023. [En línea]. Available: https://www.significados.com/microprocesador/. [Último acceso: 20 02 2023].

[26] Y. D.-R. R. A. B.-G. Ivet Challenger-Pérez, «El lenguaje de programación Python/The programming language Python,» *Ciencias Holguín,* 2014.

[27] V. L. Gaston, «Comunicaciones,» [En línea]. Available: http://cdr.ing.unlp.edu.ar/files/presentaciones/014_Comunicaciones.pdf. [Último acceso: 18 02 2023].

[28] R. G. Reyna, «Generalidades de Los Mecanismos,» PDFCOFFEE, [En línea]. Available: https://pdfcoffee.com/generalidades-de-los-mecanismos-4-pdf-free.html. [Último acceso: 12 06 2024].

[29] D. I. Á. F. V. Reyna, «Analisis cinematico de mecanismos unidad 2,» Dlideshare a Scribd company, 23 06 2016. [En línea]. Available: https://www.slideshare.net/slideshow/analisis-cinematico-de-mecanismos-unidad-2/57411566. [Último acceso: 15 06 2024].

[30] A. Pérez, «Inversión cinemática,» Mecapedia, 01 01 2000. [En línea]. Available: https://www.mecapedia.uji.es/pages/inversion_cinematica.html#:~:text=Se%20denomina%20inversi%C3%B3n%20cinem%C3%A1tica%20de,como%20eslab%C3%B3n%20fijo%20del%20mecanismo.. [Último acceso: 20 06 2024].

[31] «MECANISMOS (TOPOLOGÍA),» 13 09 2015. [En línea]. Available: https://pdfcoffee.com/mecanismos-topologia-pdf-free.html. [Último acceso: 21 06 2024].

[32] J. Andrés, «Ley de Grashof,» Mecapedia, 17 09 2021. [En línea]. Available: https://www.mecapedia.uji.es/pages/ley_de_Grashof.html. [Último acceso: 17 06 2024].

[33] F. F. Ripalda, «Desarrollo de un modelo de reconocimiento de gestos manuales de la mano utilizando señales EMG y Deep Learning,» *Trabajo de Titulación,* 2021.

[34] L. T. ojeda, «¿Que es Raspberry Pi?,» MCI electronics, [En línea]. Available: https://raspberrypi.cl/que-es-raspberry/. [Último acceso: 15 03 2024].

[35] A. y. r. Ingeniería, «Raspberry Pi,» Novatronic, [En línea]. Available: https://novatronicec.com/index.php/product/raspberry-pi-4-4g-ram/. [Último acceso: 15 03 2024].

[36] «ARDUBOARD UNO R3,» NAYLAMP MECHATRONICS, [En línea]. Available: https://naylampmechatronics.com/ardusystem-tarjetas/8-uno-r3.html. [Último acceso: 25 06 2024].

[37] T. l. d. reservados, «MICRO SERVO MG90S,» NAYLAMP MECHATRONICS, [En línea]. Available: https://naylampmechatronics.com/servomotores/246-micro-servo-

mg90s.html#:~:text=Servomotor%20de%20tama%C3%B1o%20peque%C3%B1o%20ideal,90%C2%B0%20en%20cada%20direcci%C3%B3n.. [Último acceso: 16 03 2024].

[38] T. l. d. reservados., «Batería Recargable Maxday LIR-18650.,» Digital box, [En línea]. Available: https://digitalbox.com.ec/home/993-bateria-recargable-maxday-lir-18650.html. [Último acceso: 16 03 2024].

[39] T. l. d. reservados., «Porta Pilas 2 Pilas 18650,» ROBÓTICA, ELECTRÓNICA COMPUTACIÓN INGENIERÍA., [En línea]. Available: https://megatronica.cc/producto/porta-pilas-2-pilas-18650/. [Último acceso: 16 03 2024].

[40] A. M. A. D. y. J. P. Estrella, «Construcción de electrodos de Plata/Cloruro de plata para medición de Potenciales Eléctricos en Estructuras Sumergidas.,» [En línea]. Available: https://www.dspace.espol.edu.ec/bitstream/123456789/24440/1/TESIS%20ALEJANDRO%20DAVID%20AGAMA%20MOSQUERA.pdf. [Último acceso: 20 03 2024].

[41] T. l. d. reservados, «Electrodos de espuma Kendall serie 200,» Quadmed, Inc, [En línea]. Available: https://quadmed.com/kendall-200-series-foam-electrodes/. [Último acceso: 16 03 2024].

[42] «Cable para Electrodo muscular,» Electronica Estudio, [En línea]. Available: https://www.estudioelectronica.com/tienda/sensores/cable-para-electrodo-muscular/. [Último acceso: 17 03 2024].

[43] «Jalea Conductiva Electro Gel 250g,» LANCETAHG, [En línea]. Available: https://www.lancetahg.com.mx/productos/30/jalea-conductiva-electro-gel-250-g. [Último acceso: 21 03 2024].

[44] «Osciloscopio DSO138 con Carcasa y Display TFT,» UNIT ELECTRONICS, [En línea]. Available: https://uelectronics.com/producto/osciloscopio-dso138-con-carcasa-y-display-tft/. [Último acceso: 26 03 2024].

[45] «Osciloscopio DSO138, Mini osciloscopio de bricolaje DSO 138, Kit de osciloscopio digital TFT 1MSPS de 2.4 "Con tamaño de bolsillo portátil 13803K (Versión soldada),» Ubuy, [En línea]. Available: https://www.ubuy.ec/es/product/49TE7OU-dso138-oscilloscope-dso-138-diy-mini-oscilloscope-2-4-tft-1msps-digital-oscilloscope-kit-handheld-pocket-sized-13803k-welded-version. [Último acceso: 22 03 2024].

[46] A. Devices, «AD623,» ANALOG DEVICES, [En línea]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad623.pdf. [Último acceso: 28 03 2024].

[47] «AD623 IC – Instrumentation Amplifier IC,» SHARVI ELECTRONICS, [En línea]. Available: https://sharvielectronics.com/product/ad623-ic-instrumentation-amplifier-ic/. [Último acceso: 2 04 2024].

[48] «¿Qué es la resistencia?,» FLUKE CORPORATION, [En línea]. Available: https://www.fluke.com/es-ec/informacion/blog/electrica/que-es-la-resistencia. [Último acceso: 24 03 2024].

[49] «Condensador Cerámico.,» OnubaElectrónica., [En línea]. Available: https://onubaelectronica.es/condensador-ceramico/. [Último acceso: 25 03 2024].

[50] «Condensador Cerámico,» ELECTRÓNICAONLINE, [En línea]. Available: https://electronicaonline.net/componentes-electronicos/condensador/condensador-ceramico/. [Último acceso: 25 03 2024].

[51] «LM358N Amplificador Operacional PDIP−8,» UNIT ELECTRONICS, [En línea]. Available: https://uelectronics.com/producto/lm358n-amplificador-operacional-pdip%e2%88%928/. [Último acceso: 27 03 2024].

[52] «Modulo ADS1115 ADC Amplificador de Ganancia Programable,» UNIT ELECTRONICS, [En línea]. Available: https://uelectronics.com/producto/modulo-ads1115-adc/. [Último acceso: 27 03 2024].

[53] «CONVERTIDOR VOLTAJE DC-DC STEP-DOWN 3A LM2596,» NAYLAMPMECHATRONICS, [En línea]. Available: https://naylampmechatronics.com/conversores-dc-dc/196-convertidor-voltaje-dc-dc-step-down-3a-lm2596.html. [Último acceso: 16 05 2024].

[54] «7805 REGULADOR DE VOLTAJE 5V,» TECmikro, [En línea]. Available: https://tecmikro.com/reguladores-de-voltaje/118-7805-regulador-de-voltaje-5v.html. [Último acceso: 05 04 2024].

[55] «LM7905 Regulador de voltaje negativo -5V,» DIY ELECTRÓNICA, [En línea]. Available: https://www.electronicadiy.com/products/lm7905-regulador-de-voltaje-negativo-5v. [Último acceso: 6 04 2024].

[56] «Regulador De Voltaje Negativo De 5v Lm7905 L7905,» MV ELECTRÓNICA, [En línea]. Available: https://mvelectronica.com/producto/regulador-de-voltaje-negativo-de-5v-lm7905-l7905. [Último acceso: 17 04 2024].

[57] «Autodesk Fusion: el futuro del diseño y la fabricación más allá de CAD,» AUTODESK, [En línea]. Available: https://www.autodesk.es/products/fusion-360/overview?term=1-YEAR&tab=subscription. [Último acceso: 15 04 2024].

[58] «Multisim,» EMERSON, [En línea]. Available: https://www.ni.com/es/support/downloads/software-products/download.multisim.html#452133. [Último acceso: 13 04 2024].

[59] «Sistema operativo Raspberry Pi,» Raspberry Pi, [En línea]. Available: https://www.raspberrypi.com/software/. [Último acceso: 17 04 2024].

[60] «Escanee una red en cuestión de segundos,» Advanced IP Sanner, [En línea]. Available: https://www.advanced-ip-scanner.com/es/. [Último acceso: 17 04 2024].

[61] «VNC Viewer, una herramienta robusta y fiable para acceder a escritorios de manera remota,» Softonic, [En línea]. Available: https://vnc-viewer.softonic.com/. [Último acceso: 17 04 2024].

[62] «¿Qué es Python?,» AWS Amazon, [En línea]. Available: https://aws.amazon.com/es/what-is/python/. [Último acceso: 10 04 2024].

[63] «TensorFlow para aprendizaje automático,» Blog, [En línea]. Available: https://iartificial.blog/desarrollo/tensorflow-para-aprendizaje-automatico/. [Último acceso: 18 04 2024].

[64] C.-E. S. f. t. C. o. M. I. u. S. E. a. A. N. Networks, «ResearchGate,» 04 2017. [En línea]. Available: https://www.researchgate.net/publication/321987246_Cost-effective_system_for_the_classification_of_muscular_intent_using_surface_electromyography_and_artificial_neural_networks. [Último acceso: 25 04 2024].

[65] I. A. S. R. A. A. Zainal Arief, «Comparison of Five Time Series EMG Features Extractions Using Myo Armband,» International Electronics Symposium (IES), Indonesia, 2015.

[66] A. Camarillo, «Que es un control PID,» 330ohms, 02 06 2021. [En línea]. Available: https://blog.330ohms.com/2021/06/02/que-es-un-control-pid/. [Último acceso: 21 02 2023].

[67] «Descargar Logotipo vertical de Python PNG transparente,» Stick PNG, [En línea]. Available: https://www.stickpng.com/es/img/dibujos-animados/los-numtums/logotipo-vertical-de-python. [Último acceso: 11 04 2024].

[68] «Autodesk Fusion 360 Logo - PNG Logo Vector Brand Downloads (SVG, EPS),» Pinterest, [En línea]. Available: https://in.pinterest.com/pin/autodesk-fusion-360-logo--268456827780030537/. [Último acceso: 15 04 2024].

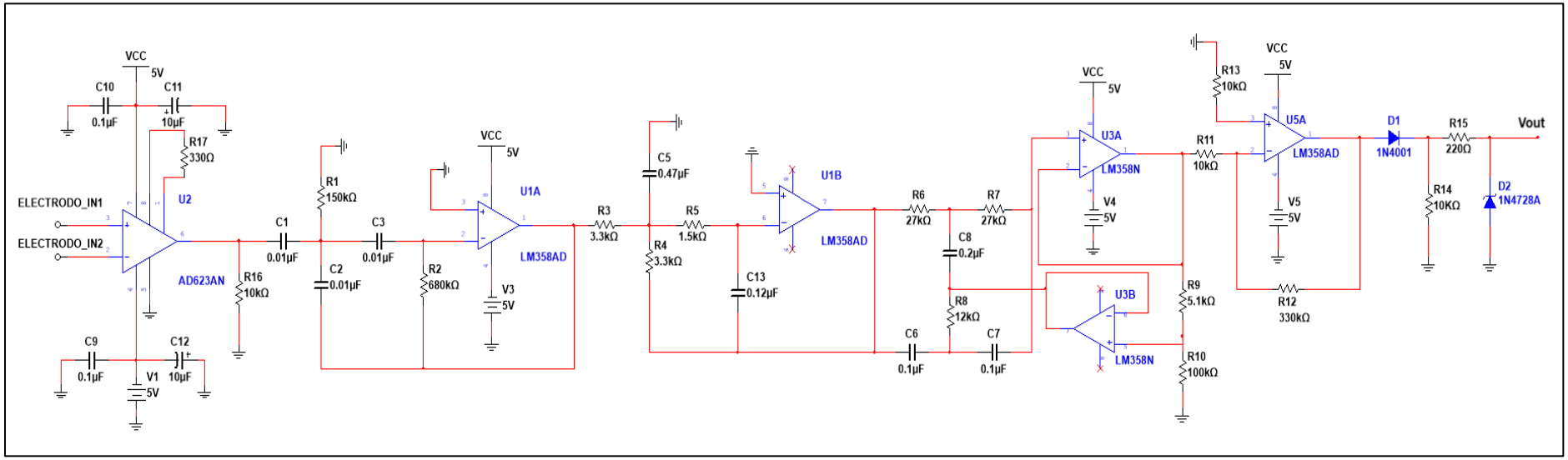
[69] «How to install TensorFlow on Raspberry Pi,» The MagPi Magazine, [En línea]. Available: https://magpi.raspberrypi.com/articles/tensorflow-ai-raspberry-pi. [Último acceso: 18 04 2024].

[70] «BROCHE CONECTOR PARA BATERIA 9V CON PLUG DC,» MEGATRONICA ROBÓTICA ELECTRÓNICA COMPUTACION INGENIERÍA, [En línea]. Available: https://megatronica.cc/producto/broche-conector-para-bateria-9v-con-plug-dc/. [Último acceso: 05 04 2024].

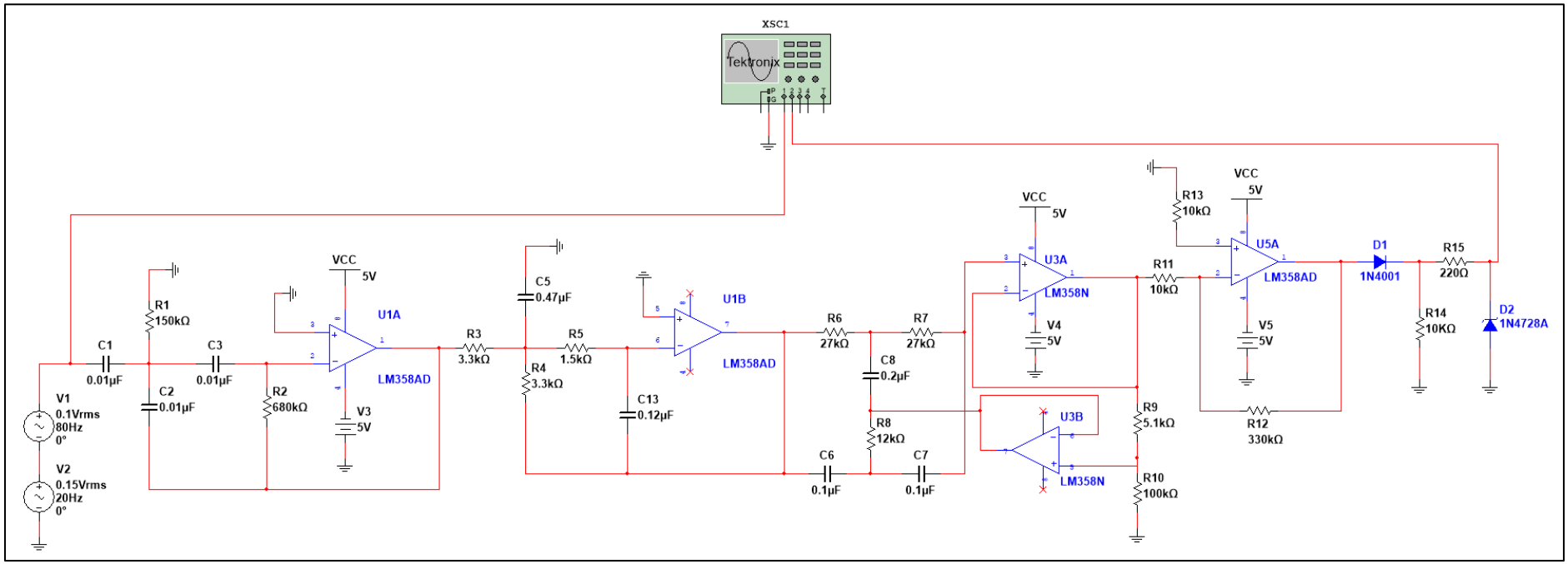
[71] «ARDUBOARD UNO R3,» NAYLAMP, [En línea]. Available: https://naylampmechatronics.com/ardusystem-tarjetas/8-uno-r3.html. [Último acceso: 01 07 2014].

ANEXOS

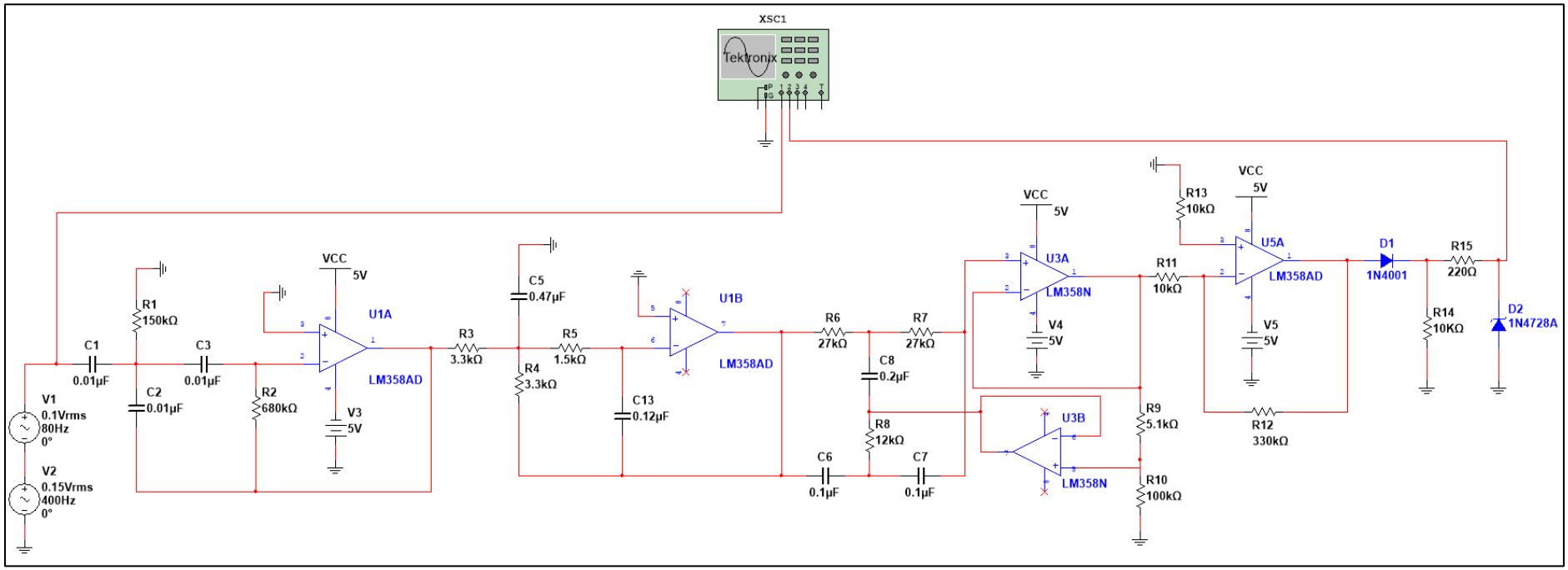
Anexo 1. Diagrama Esquemático de la adquisición de la señal EMG.



Anexo 2. Simulación del circuito con frecuencia menor a 50 Hz.



Anexo 3. Simulación del circuito con frecuencia mayor a 300 Hz



Anexo 4. Código en Arduino UNO para la adquisición de la señal y la extracción de características.

```
#include <Wire.h>

#include <Adafruit_ADS1X15.h>


// Crear un objeto para el ADS1115

Adafruit_ADS1115 ads;


const int timeData = 1;

double mavEMG = 0;

double rmsEMG = 0;

double wlEMG = 0;

double wlaEMG = 0;


const int pinStart = 3;

boolean flag = false;

unsigned long debounce = 0;


void setup(void) {

 pinMode(LED_BUILTIN, OUTPUT);

 pinMode(pinStart,INPUT_PULLUP);

 digitalWrite(LED_BUILTIN, LOW);


 Serial.begin(9600);  // Inicializar la comunicación serie


 // Inicializar el ADS1115

 while(!ads.begin()) {

  delay(500);

 }


 // Configurar la tasa de muestreo a 860 SPS
```

```
  ads.setDataRate(RATE_ADS1115_860SPS);

  // Configurar el rango de voltaje a ±4.096V (ganancia = 1)
  ads.setGain(GAIN_ONE);
}

void loop(void) {
  if(!digitalRead(pinStart)){
    pulse();
  }
  if(flag){
    digitalWrite(LED_BUILTIN,HIGH);
    featuresExtraction();
    digitalWrite(LED_BUILTIN,LOW);

    Serial.println(mavEMG);
    Serial.println(rmsEMG);
    Serial.println(wlEMG);
    flag = false;
  }

}
void featuresExtraction(){
  int sizeSample = timeData*10;
  mavEMG = 0;
  rmsEMG = 0;
  wlEMG = 0;
  for (int k = 0; k < sizeSample; k++){
      // Leer la señal del canal 0
      int16_t adcValue = ads.readADC_SingleEnded(0);
```

```
    // Convertir a voltaje (considerando GAIN_ONE que tiene resolución de 0.125 mV por bit)
    float voltaje = adcValue * 0.125 / 1000.0;  // Convertir a voltios

    // Normaliza el valor ADC en el rango de 0 a 1
    float normalizedValue = voltaje / 3.0;

    mavEMG = mavEMG + abs(normalizedValue);
    rmsEMG = rmsEMG + normalizedValue * normalizedValue;
    wlEMG = wlEMG + abs(normalizedValue - wlaEMG);

    wlaEMG = normalizedValue;

    delay(100); //Esperar 100ms entre cada lectura

  }
   mavEMG = mavEMG/(double)sizeSample;
   rmsEMG = sqrt(rmsEMG/(double)sizeSample);
}
void pulse(){
  if(!digitalRead(pinStart) && (millis()-debounce > 500)){
    debounce = millis();
    flag = true;
    delay(2000);
  }
}
```

Anexo 5. Código en Python para la lectura de las señales desde el Arduino y guardado en archivo .npy

```
import time

import serial

import numpy as np

import matplotlib.pyplot as plt


Examples = 3


COM = '/dev/ttyACM0'


try:

  arduinoSerial = serial.Serial(COM, 9600)

except:

  print('Cannot conect to the port')


numFeatures = 3


P =  np.zeros((Examples,numFeatures))


time.sleep(3)


print('Recolectando Datos')


with arduinoSerial:

  for k in range(Examples):

    for n in range(numFeatures):

      P[k,n] = arduinoSerial.readline().strip()

    print(k+1)


print('Datos Recolectados')
```

```
np.save('descan',P)


for i in range(numFeatures):
  fig = plt.figure()
  plt.bar(np.arange(0,Examples),P[:,i],label = 'Feature'+str(i+1))
  plt.legend(loc="upper left")
  plt.grid()

plt.show()
```

Anexo 6. Código en Python para generar el conjunto de datos de entrada.

```
from readDataset import *
import matplotlib.pyplot as plt
import numpy as np

sizeClass = 3  # Cantidad de clases a clasificar

d = dataset(sizeClass)
d.read()

print(f"Size Input Data: {d.datasetInput.shape}") #datasetInput [samples,input]

for i in range(sizeClass):
    print(f"Size Class Data {i+1}: {d.sizeSubClass[i]}")

T = np.concatenate((0*np.ones((d.sizeSubClass[0],1)),
                1*np.ones((d.sizeSubClass[1],1)),
                2*np.ones((d.sizeSubClass[2],1))), axis=0)

print(f"Size Output Data: {T.shape}") #datasetOutput[samples,output]

# Plot input
for i in range(d.numInputs):
    plt.plot(d.datasetInput[:,i], label='Data '+str(i+1))
    plt.legend(loc="upper left")

plt.grid()
plt.show()

np.save('P', d.datasetInput)
np.save('T', T)
```

Anexo 7. Código para el entrenamiento de las señales EMG.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
import joblib
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Importar o generar conjuntos de datos
P = np.load('P.npy')
T = np.load('T.npy')

# Preprocesamiento de datos
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(P)
P = scaler.transform(P)

one_hot_labels = to_categorical(T, num_classes=3)

# Dividir el conjunto de datos en conjuntos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
P_train, P_test, T_train, T_test = train_test_split(P, one_hot_labels, test_size=0.20,
random_state=42)

# Establecer hiperparámetros de algoritmo
epochs = 700
hiddenNodes = 8
```

```
# Definir la arquitectura de la red neuronal
model = Sequential()
model.add(Dense(hiddenNodes, activation='relu', input_dim=3))
model.add(Dense(hiddenNodes, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.summary()

# Declarar la función de pérdida y el optimizador
loss = 'categorical_crossentropy'
optimizer = tf.keras.optimizers.Adam()

model.compile(loss=loss, optimizer=optimizer, metrics=['accuracy'])

# Entrenar el modelo
history = model.fit(P_train, T_train, epochs=epochs, verbose=1, validation_split=0.1)
plt.xlabel('Epocas')
plt.ylabel('Pérdida')
plt.plot(history.epoch, np.array(history.history['loss']), 'b', label='Train Loss')
plt.plot(history.epoch, np.array(history.history['val_loss']), 'r', label='Val Loss')
plt.legend()
plt.grid()

# Evaluar el modelo
test_loss, test_acc = model.evaluate(P_test, T_test, verbose=1)
print('Exactitud de prueba: ', test_acc)

# Guardar el modelo entrenado
model.save('modelo_entrenado.keras')

# Guardar el scaler
joblib.dump(scaler, 'scaler.pkl')
```

```
# Mostrar la gráfica de pérdida
plt.xlabel('Epocas')
plt.ylabel('Pérdida')
plt.plot(history.epoch, np.array(history.history['loss']), 'b', label='Train Loss')
plt.plot(history.epoch, np.array(history.history['val_loss']), 'r', label='Val Loss')
plt.legend()
plt.grid()
plt.show()
```

Anexo 8. Código en Arduino para la predicción de los gestos y control de los servomotores.

```
#include <Wire.h>

#include <Adafruit_ADS1X15.h>

#include <Servo.h>


// Crear un objeto para el ADS1115

Adafruit_ADS1115 ads;


const int timeData = 1;

double mavEMG = 0;

double rmsEMG = 0;

double wlEMG = 0;

double wlaEMG = 0;



Servo PulgarBase;

Servo PulgarFalange;

Servo Indice;

Servo Medio;

Servo Anular;

Servo Pinky;


int Ini_pulgarBase=110;

int Ini_pulgarFalange=115;

int Ini_Indice=80;

int Ini_Medio=145;

int Ini_Anular=152;

int Ini_Pinky=142;


int Fin_pulgarBase=30;
```

```
int Fin_pulgarFalange=25;
int Fin_Indice=37;
int Fin_Medio=80;
int Fin_Anular=75;
int Fin_Pinky=75;

void setup(void) {
  pinMode(LED_BUILTIN, OUTPUT);

  PulgarFalange.attach(10);
  PulgarBase.attach(11);
  Indice.attach(3);
  Medio.attach(5);
  Anular.attach(6);
  Pinky.attach(9);

  PulgarBase.write(Ini_pulgarBase);
  PulgarFalange.write(Ini_pulgarFalange);
  Indice.write(Ini_Indice);
  Medio.write(Ini_Medio);
  Anular.write(Ini_Anular);
  Pinky.write(Ini_Pinky);

  Serial.begin(9600);  // Inicializar la comunicación serie

  // Inicializar el ADS1115
  while (!ads.begin()) {
    delay(500);
  }

  // Configurar la tasa de muestreo a 860 SPS
```

```
  ads.setDataRate(RATE_ADS1115_860SPS);

  // Configurar el rango de voltaje a ±4.096V (ganancia = 1)
  ads.setGain(GAIN_ONE);
}

void loop(void) {
  // Ejecutar la extracción de características
  featuresExtraction();

  // Enviar las características al puerto serial
  Serial.println(mavEMG);
  Serial.println(rmsEMG);
  Serial.println(wlEMG);

  // Leer y procesar la predicción recibida del Python
  if (Serial.available() > 0) {
    int prediccion = Serial.parseInt();  // Leer la predicción del puerto serial



    // Procesar la predicción recibida (ejemplo: encender un LED según la clase predicha)
    if (prediccion == 0) {
      PulgarBase.write(Fin_pulgarBase);
      PulgarFalange.write(Fin_pulgarFalange);
      Medio.write(Fin_Medio);
      Indice.write(Fin_Indice);
      Anular.write(Fin_Anular);
      Pinky.write(Fin_Pinky);
    } else if (prediccion == 1) {
      PulgarBase.write(Fin_pulgarBase);
```

```
    PulgarFalange.write(Fin_pulgarFalange);
    Indice.write(Fin_Indice-30);
    Medio.write(Ini_Medio);
    Anular.write(Ini_Anular);
    Pinky.write(Ini_Pinky);

  } else if (prediccion == 2) {
    PulgarBase.write(Ini_pulgarBase);
    PulgarFalange.write(Ini_pulgarFalange);
    Indice.write(Ini_Indice);
    Medio.write(Ini_Medio);
    Anular.write(Ini_Anular);
    Pinky.write(Ini_Pinky);
  }
 }

 // Esperar un segundo antes de la siguiente adquisición
 //delay(1000);
}

void featuresExtraction() {
 int sizeSample = timeData * 10;
 mavEMG = 0;
 rmsEMG = 0;
 wlEMG = 0;
 for (int k = 0; k < sizeSample; k++) {
  // Leer la señal del canal 0
  int16_t adcValue = ads.readADC_SingleEnded(0);

  // Convertir a voltaje (considerando GAIN_ONE que tiene resolución de 0.125 mV por bit)
  float voltaje = adcValue * 0.125 / 1000.0;  // Convertir a voltios
```

```
    // Normaliza el valor ADC en el rango de 0 a 1
    float normalizedValue = voltaje / 3.0;

    mavEMG = mavEMG + abs(normalizedValue);
    rmsEMG = rmsEMG + normalizedValue * normalizedValue;
    wlEMG = wlEMG + abs(normalizedValue - wlaEMG);

    wlaEMG = normalizedValue;

    delay(100); // Esperar 100ms entre cada lectura
  }
  mavEMG = mavEMG / (double)sizeSample;
  rmsEMG = sqrt(rmsEMG / (double)sizeSample);
}
```